PracticalR Final Presentation

AO

December 18, 2018

Libraries called for analysis(called in the setup chunk)

```
#library(affy)
#library(affycomp)
#library(affydata)
#library(tools)
#library(tidyverse)
#library(biomaRt)
#library(ggplot2)
#library(reshape2)
#library(dplyr)
```

import data from desktop

```
data.NKrest<- ReadAffy(filenames="D:/Users/Steve Fitzgerald/Desktop/GSM198942.CEL/GSM198942.
```

convert AffyBatch Object into expressionset.

- exprs returns a matrix containing the expression values

```
eset <- mas5(data.NKrest)
```

```
## background correction: mas
## PM/MM correction : mas
## expression values: mas
## background correcting...done.
## 54675 ids to be processed
## |                    |
## |####################|
```

```
exp.NKrest <-  exprs(eset)
```

Check for missing data

```
any(is.na(exp.NKrest))
```

```
## [1] FALSE
```

Change the column name so that it is named after the sample

```
colnames(exp.NKrest) <- c('NK_Rest')
```

Importing data sets for other samples in experiment

```
data.NK2 <- ReadAffy(filenames="D:/Users/Steve Fitzgerald/Desktop/GSM198943.CEL/GSM198943.CE
data.NK8 <- ReadAffy(filenames="D:/Users/Steve Fitzgerald/Desktop/GSM198944.CEL/GSM198944.CE
data.NK24 <- ReadAffy(filenames="D:/Users/Steve Fitzgerald/Desktop/GSM198945.CEL/GSM198945.C
data.T <- ReadAffy(filenames="D:/Users/Steve Fitzgerald/Desktop/GSM198958.CEL/GSM198958.CEL"
data.lymph <- ReadAffy(filenames="D:/Users/Steve Fitzgerald/Desktop/GSM198959.CEL/GSM198959.
```

Creating a function for converting AffyBatch to expression set: AffytoExpFun

```
AffytoExpFun <- function(x){
  eset <- mas5(x)
  exp.NK <<-  exprs(eset)
}
```

Applying AffytoExpFun to next data file

```
AffytoExpFun(data.NK2)
```

```
## background correction: mas
## PM/MM correction : mas
## expression values: mas
## background correcting...done.
## 54675 ids to be processed
## |                    |
## |####################|
```

- it worked!

Renaming recent data import

```
exp.NK2 <- exp.NK
colnames(exp.NK2) <- c("NK_2hours")
```

Putting it all together and applying to other samples

```
AffytoExpFun(data.NK8)
```

```
## background correction: mas
## PM/MM correction : mas
## expression values: mas
## background correcting...done.
## 54675 ids to be processed
## |                    |
## |####################|
```

```
exp.NK8 <- exp.NK
colnames(exp.NK8) <- c('NK_8hours')

AffytoExpFun(data.NK24)
```

```
## background correction: mas
## PM/MM correction : mas
```

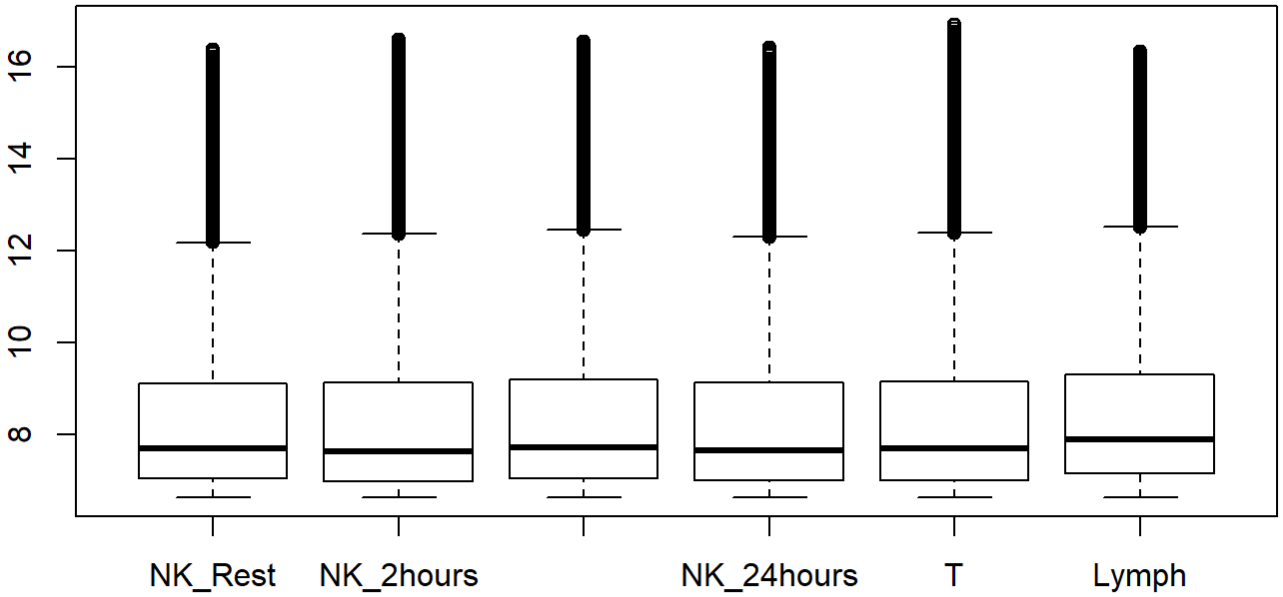
Combining all data into one matrix with sample as the columns and gene ids as rows using cbind

- wanted to use left_join by row names but couldn't figure it out

```
exp.combined <- cbind(exp.NKrest, exp.NK2, exp.NK8, exp.NK24, exp.T, exp.lymph)
```

Plot distribution of intensity values (log2) before quantile normalization

```
boxplot(exp.combined %>% {log2(.+100)})
```



Identifying affyid associated with protein of interest: FAP

```
#affyIDs <- rownames(exp.lymph)
#ensemblHumanData <- useMart("ensembl",dataset="hsapiens_gene_ensembl")
#genenames <-getBM(attributes = c('affy_hg_u133_plus_2','hgnc_symbol'),filters = "affy_hg_u1

#I tried annot.exp2 <- merge(genenames, exp.combined) to add the gene names to the data but

#subset(genenames, subset = hgnc_symbol == 'FAP')
```

Extracting FAP data based on affy id

```
FAPdata <- exp.combined['209955_s_at',] %>%
  as.data.frame() %>%
  rownames_to_column()
names(FAPdata) <- c('sample', 'expression')
FAPdata$sample <- factor(FAPdata$sample, levels=c("NK_Rest", "NK_2hours", "NK_8hours", "NK_2
class(FAPdata)
```

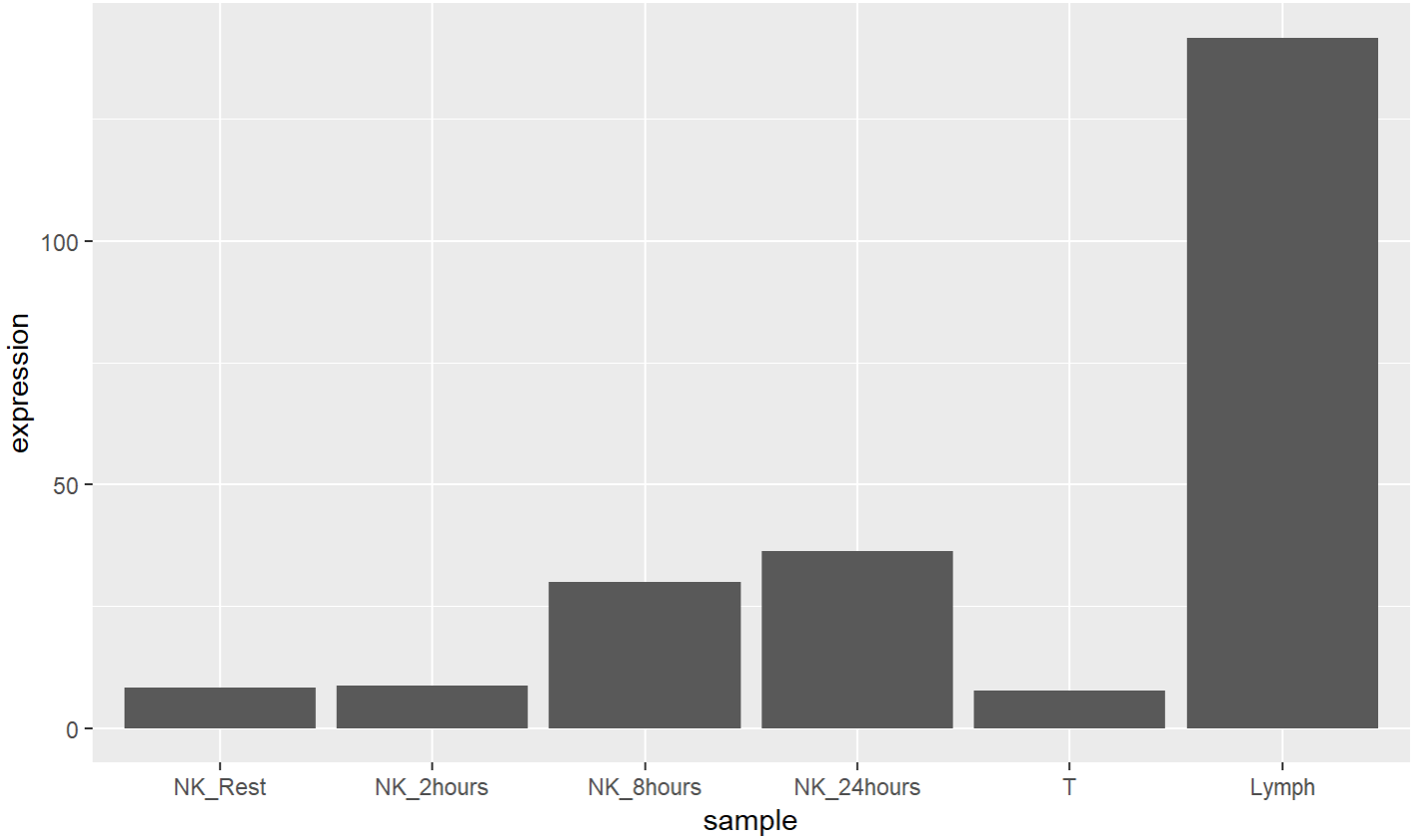
```
## [1] "data.frame"
```

```
FAPdata
```

```
##       sample expression
## 1    NK_Rest   8.259636
## 2  NK_2hours   8.763853
## 3  NK_8hours  30.012646
## 4 NK_24hours  36.243880
## 5          T   7.783959
## 6      Lymph 141.663739
```

Plotting FAP expression by sample

```
FAPdata %>% ggplot(aes(sample,expression))+geom_col()
```



Making a plot without the lymph

```
FAPminuslymph <- FAPdata[1:5,]
FAPminuslymph
```

```
##       sample expression
## 1    NK_Rest   8.259636
## 2  NK_2hours   8.763853
## 3  NK_8hours  30.012646
## 4 NK_24hours  36.243880
## 5          T   7.783959
```

```
FAPminuslymph %>% ggplot(aes(sample, expression))+geom_col()+theme_classic()
```

Determining count of GAPDH to see what counts for housekeeping genes are like

```
#subset(genenames, subset = hgnc_symbol == 'GAPDH')
GAPDHdata <- exp.combined['212581_x_at',] %>%
  as.data.frame() %>%
  rownames_to_column()
names(GAPDHdata) <- c('sample', 'expression')
GAPDHdata$sample <- factor(GAPDHdata$sample, levels=c("NK_Rest", "NK_2hours", "NK_8hours", "
GAPDHdata
```

```
##       sample expression
## 1    NK_Rest   27324.95
## 2  NK_2hours   26446.89
## 3  NK_8hours   38554.13
## 4 NK_24hours   44799.84
## 5          T   33166.65
## 6      Lymph   56288.69
```

```
GAPDHdata %>% ggplot(aes(sample, expression))+geom_col()+theme_classic()
```

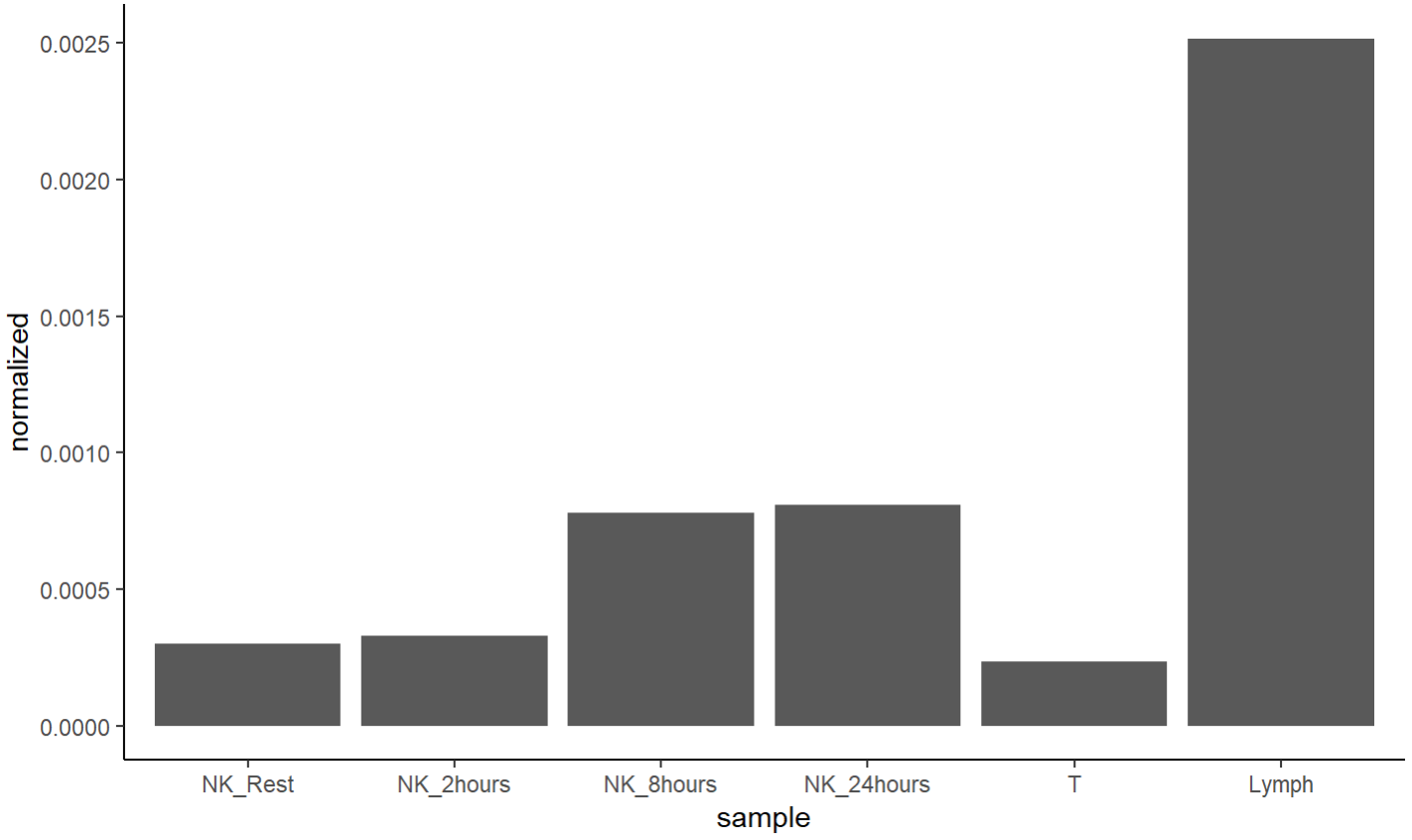
Normalize FAP Data to GAPDH

```
FAPdata2 <- FAPdata %>%
  mutate(normalized = FAPdata$expression / GAPDHdata$expression)
FAPdata2
```

```
##       sample expression   normalized
## 1    NK_Rest   8.259636 0.0003022745
## 2  NK_2hours   8.763853 0.0003313756
## 3  NK_8hours  30.012646 0.0007784548
## 4 NK_24hours  36.243880 0.0008090180
## 5          T   7.783959 0.0002346923
## 6      Lymph 141.663739 0.0025167354
```

Creating a graph with GAPDH normalized values

```
FAPdata2 %>% ggplot(aes(sample, normalized))+geom_col()+theme_classic()
```



Looking at CD107 as positive control (Gene Name = LAMP1)

```
#subset(genenames, subset = hgnc_symbol == 'LAMP1')
LAMP1data <- exp.combined[c('201551_s_at','201553_s_at', '201552_at', '213728_at'),]
LAMP1data
```

```
##                NK_Rest  NK_2hours  NK_8hours NK_24hours          T
## 201551_s_at  1665.8764  1318.0414  1341.5973  1266.6598  1524.9696
## 201553_s_at 19464.0189 19876.9711 19604.6061 21445.2759 18827.5733
## 201552_at    5596.8437  6439.0492  5710.2939  6644.9255  5972.9261
## 213728_at     268.2253   146.8866   371.8361   355.6982   500.6912
##                 Lymph
## 201551_s_at  680.7097
## 201553_s_at 9764.3251
## 201552_at   1970.4581
## 213728_at    143.5302
```

Melt Data

```
LAMP1data2 <- LAMP1data %>%
  melt()
LAMP1data2
```

```
##          Var1       Var2      value
## 1  201551_s_at    NK_Rest  1665.8764
## 2  201553_s_at    NK_Rest 19464.0189
## 3    201552_at    NK_Rest  5596.8437
## 4    213728_at    NK_Rest   268.2253
## 5  201551_s_at  NK_2hours  1318.0414
## 6  201553_s_at  NK_2hours 19876.9711
## 7    201552_at  NK_2hours  6439.0492
## 8    213728_at  NK_2hours   146.8866
## 9  201551_s_at  NK_8hours  1341.5973
## 10 201553_s_at  NK_8hours 19604.6061
## 11   201552_at  NK_8hours  5710.2939
## 12   213728_at  NK_8hours   371.8361
## 13 201551_s_at NK_24hours  1266.6598
## 14 201553_s_at NK_24hours 21445.2759
## 15   201552_at NK_24hours  6644.9255
```

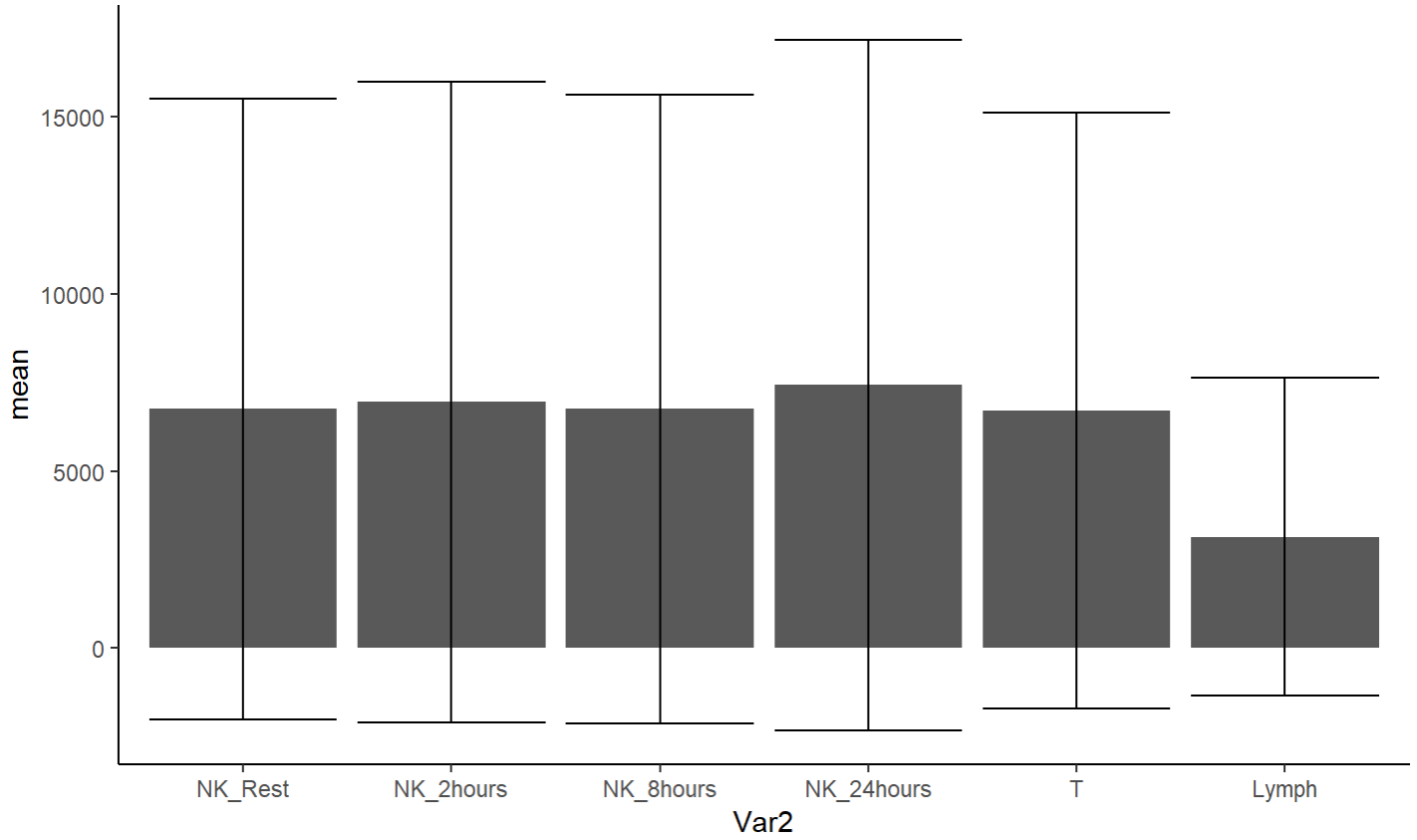
Group Data by sample, get average and sd expression

```
LAMP1data3 <- LAMP1data2 %>%
  group_by(Var2) %>%
  summarise(mean = mean(value), sd = sd(value))
LAMP1data3
```

```
## # A tibble: 6 x 3
##   Var2        mean    sd
##   <fct>      <dbl> <dbl>
## 1 NK_Rest    6749. 8772.
## 2 NK_2hours  6945. 9044.
## 3 NK_8hours  6757. 8874.
## 4 NK_24hours 7428. 9748.
## 5 T          6707. 8423.
## 6 Lymph      3140. 4482.
```

Make plot

```
LAMP1data3 %>% ggplot(aes(Var2, mean)) + geom_col() +
  geom_errorbar(aes(ymin= mean+sd, ymax= mean-sd))+ theme_classic()
```



Did Shapiro test, got error because sample size is not between 3 and 5000

```
#shapiro.test(exp.T)
```

Did one-way ANOVA

```
res.aov <- aov(value ~ Var2, data = LAMP1data2)
summary(res.aov)
```

```
##             Df    Sum Sq  Mean Sq F value Pr(>F)
## Var2         5 4.900e+07  9800654   0.139  0.981
## Residuals   18 1.271e+09 70590679
```