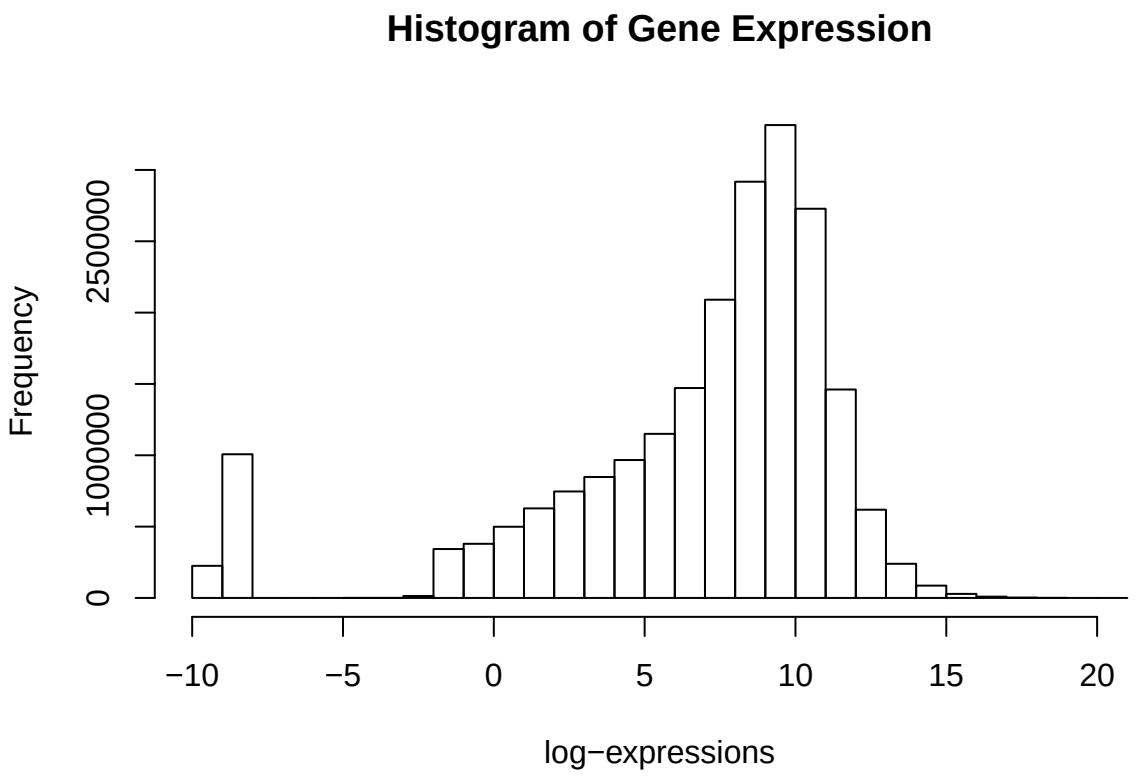
RNAseq DCM Analysis

Kevin O'Connor

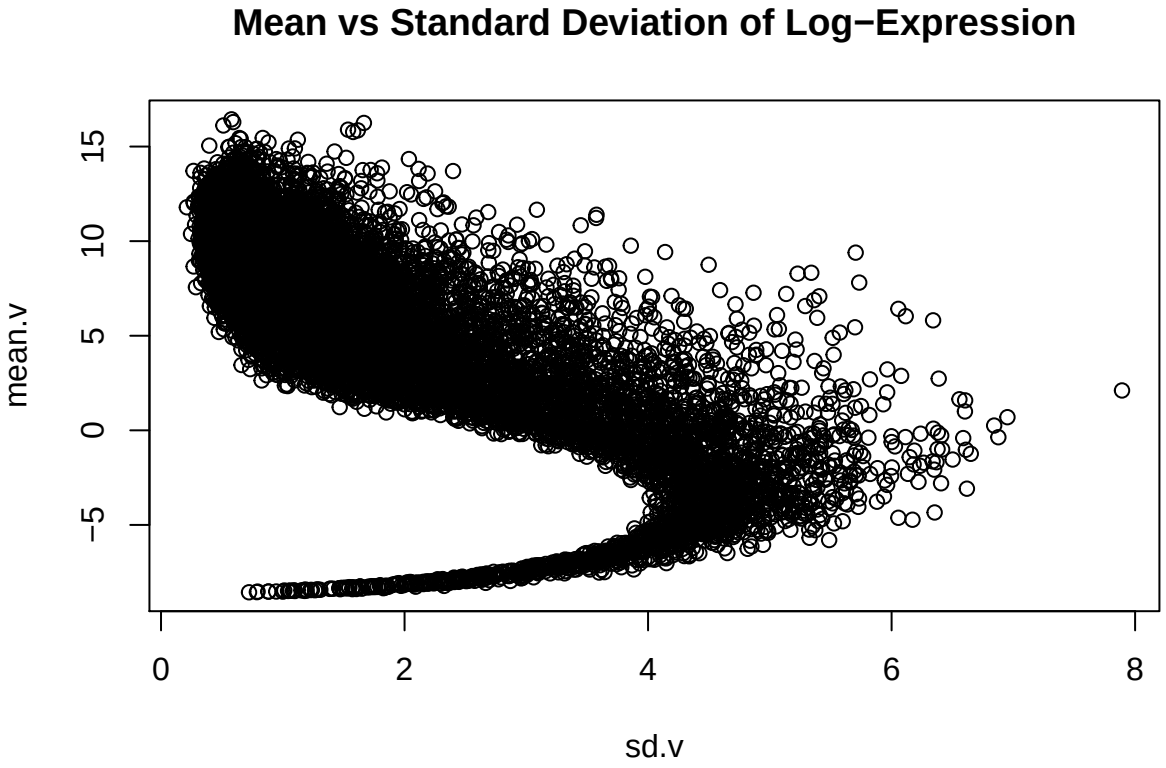
8/28/2018

In this document, we perform some exploratory data anlysis on the RNAseq data. We begin by filtering and log-transforming the data.

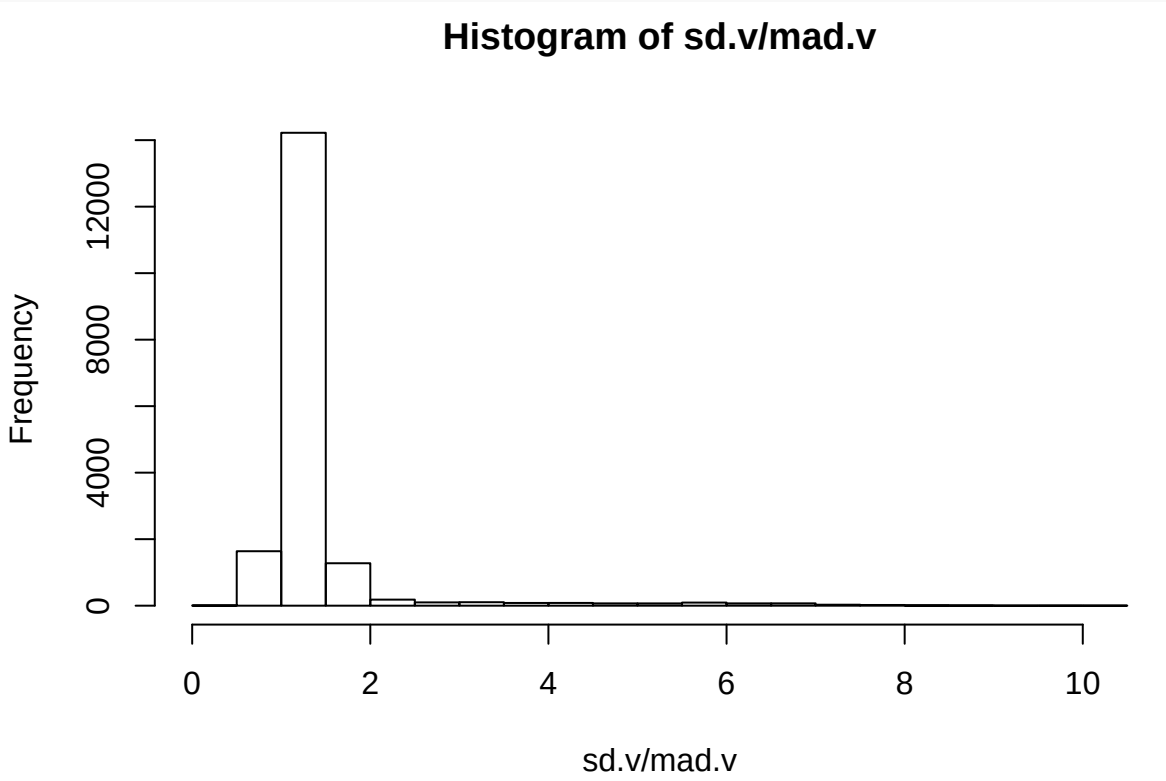
```
# Filter and transform the data.
dataM <- filter_data(dataM0, min.var=10, max.var=1e24)
hist(dataM, xlab="log-expressions", main="Histogram of Gene Expression")
```




```
# Investigate mean vs standard deviation.
mean.v <- apply(dataM, 1, mean)
sd.v   <- apply(dataM, 1, sd)

plot(sd.v, mean.v, main="Mean vs Standard Deviation of Log-Expression")
```

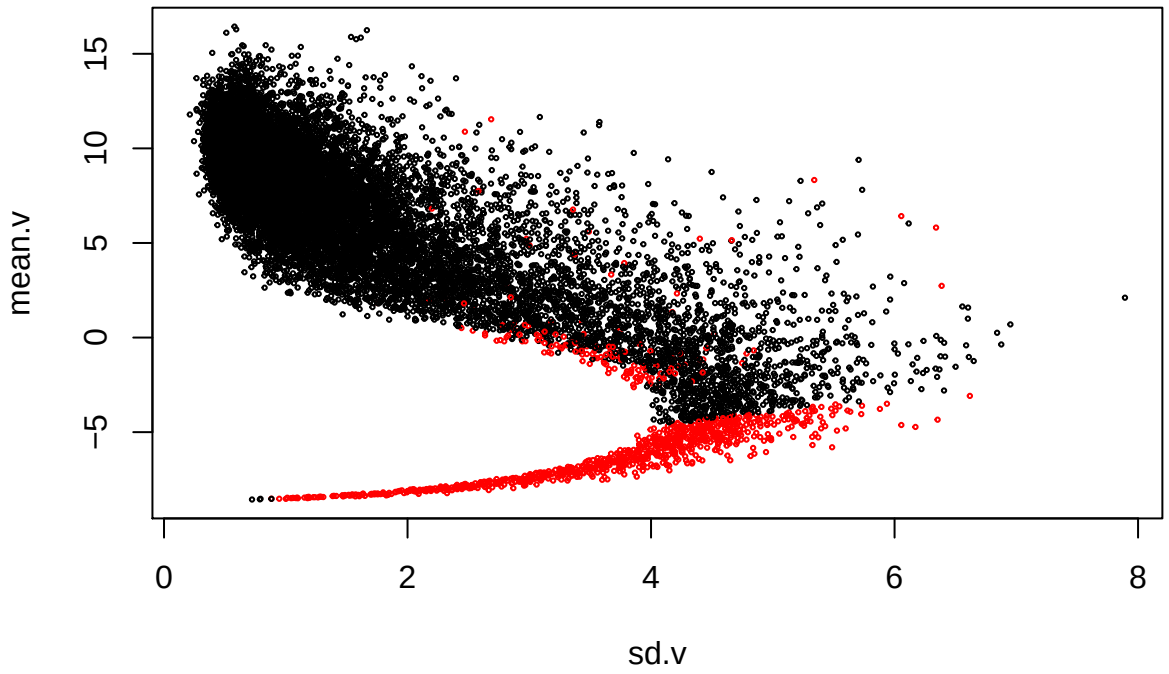



```
# Investigate mean absolute deviation and compare to standard deviation.
mad.v <- apply(dataM, 1, mad)
hist(sd.v/mad.v)
```



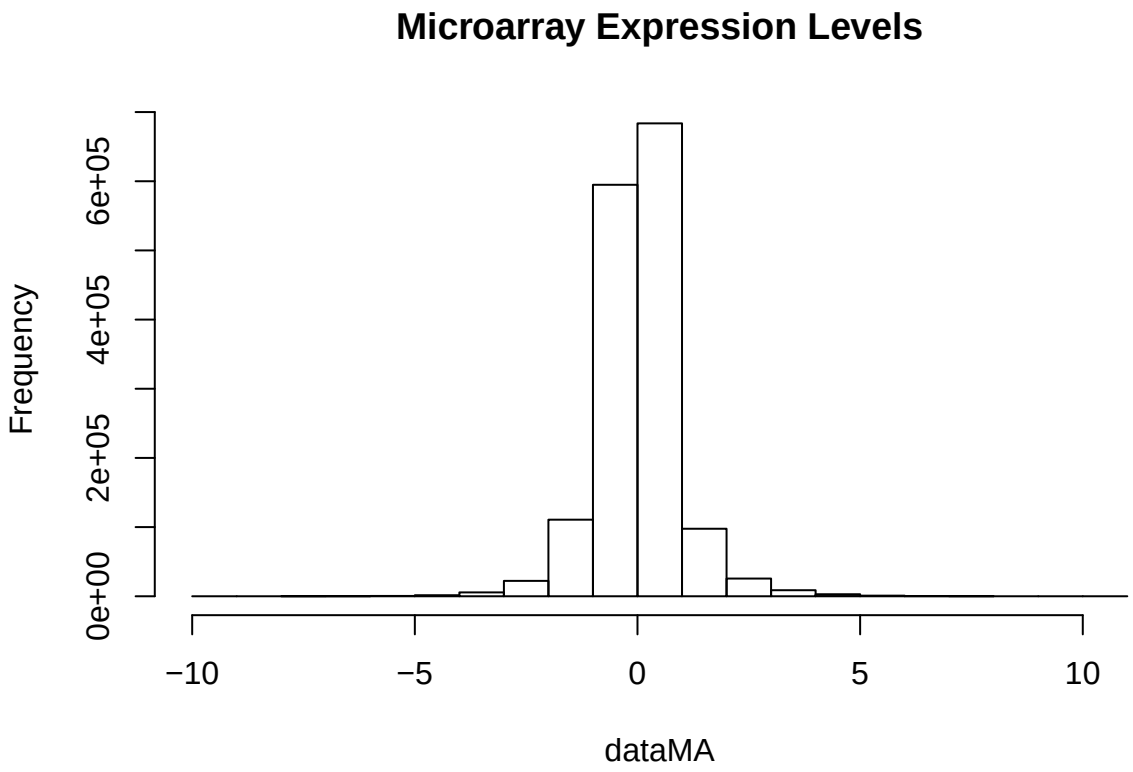
So some genes have small MAD's compared to their standard deviations. As the MAD is more robust to outliers, genes with a low MAD/SD ratio likely contain serious outliers.

```
# Set sd/mad threshold.
th <- 2

# Find genes with mad/sd exceeding threshold.
bad.genes <- sd.v/mad.v > th
plot(sd.v, mean.v, col=ifelse(bad.genes, "red", "black"), cex=0.3)
```

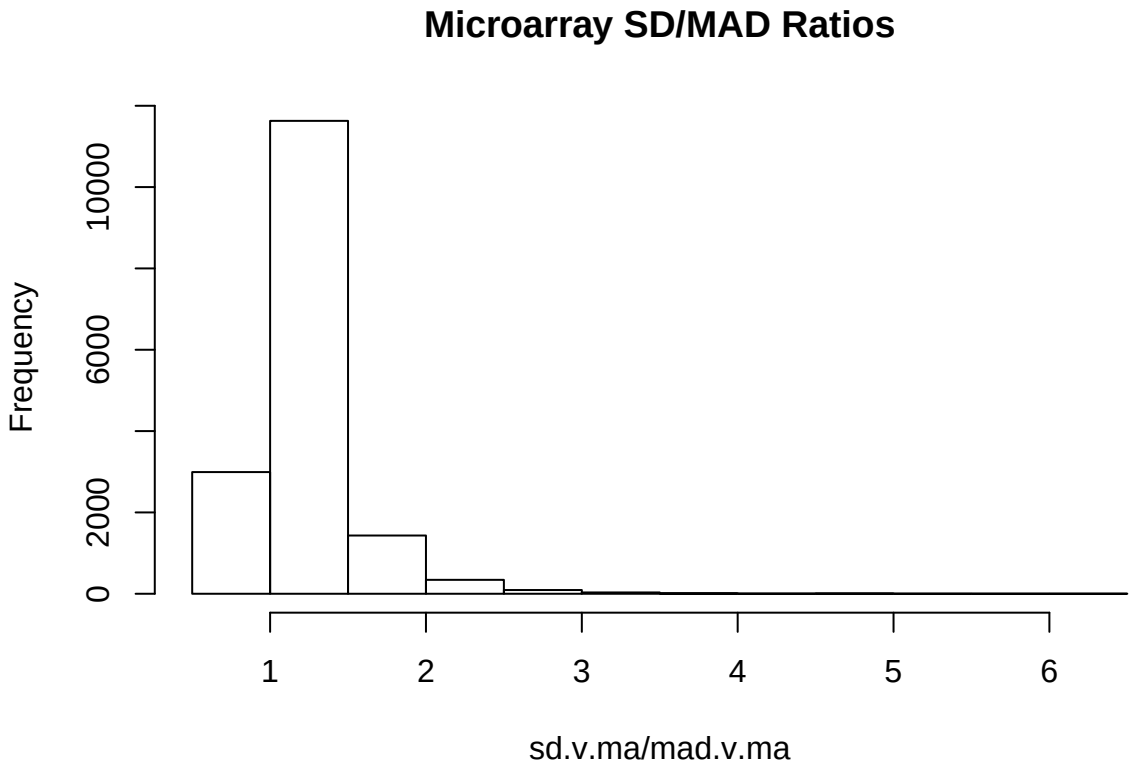


The plot above highlights the "bad genes" in red. The proportion of genes which are bad is 0.0544402. We can compare this to the microarray data which was observed to converge.

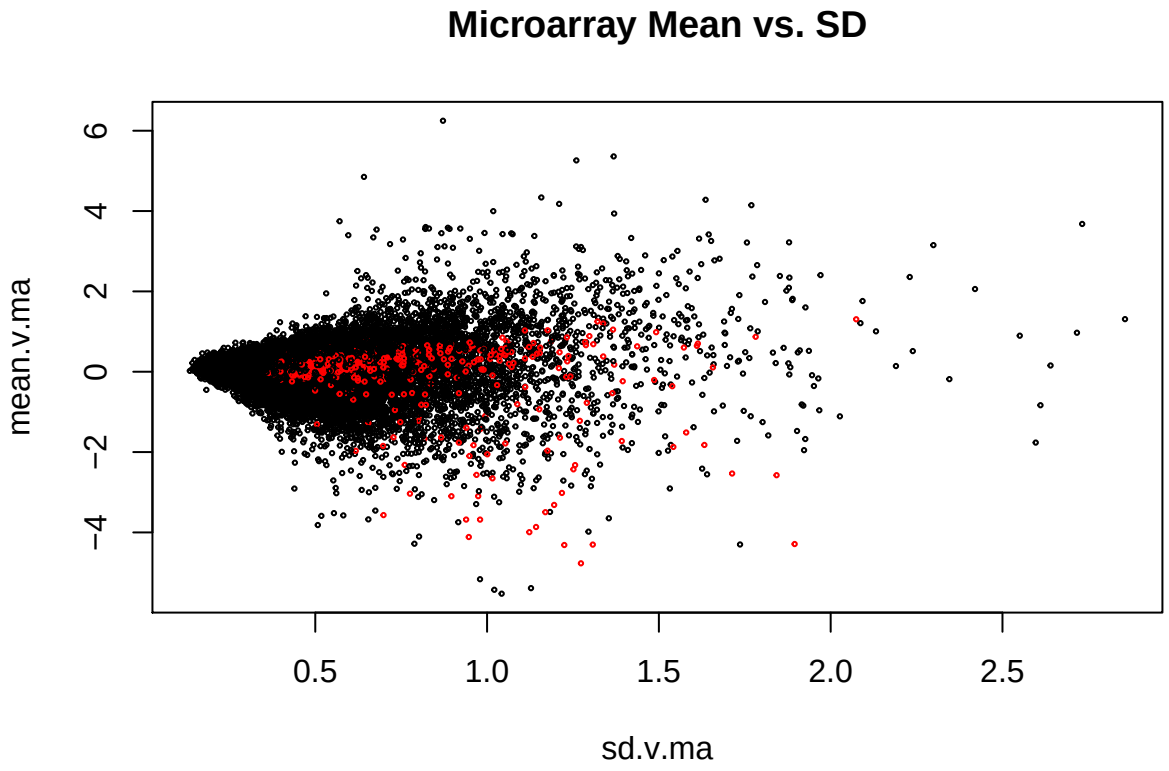
```
load(file=file.path(data.dir, "microarray_filtered.RData"))
dataMA <- microarray.dat
hist(dataMA, main="Microarray Expression Levels")
```




```
mean.v.ma <- apply(dataMA, 1, mean)
sd.v.ma   <- apply(dataMA, 1, sd)
mad.v.ma  <- apply(dataMA, 1, mad)
hist(sd.v.ma/mad.v.ma, main="Microarray SD/MAD Ratios")
```

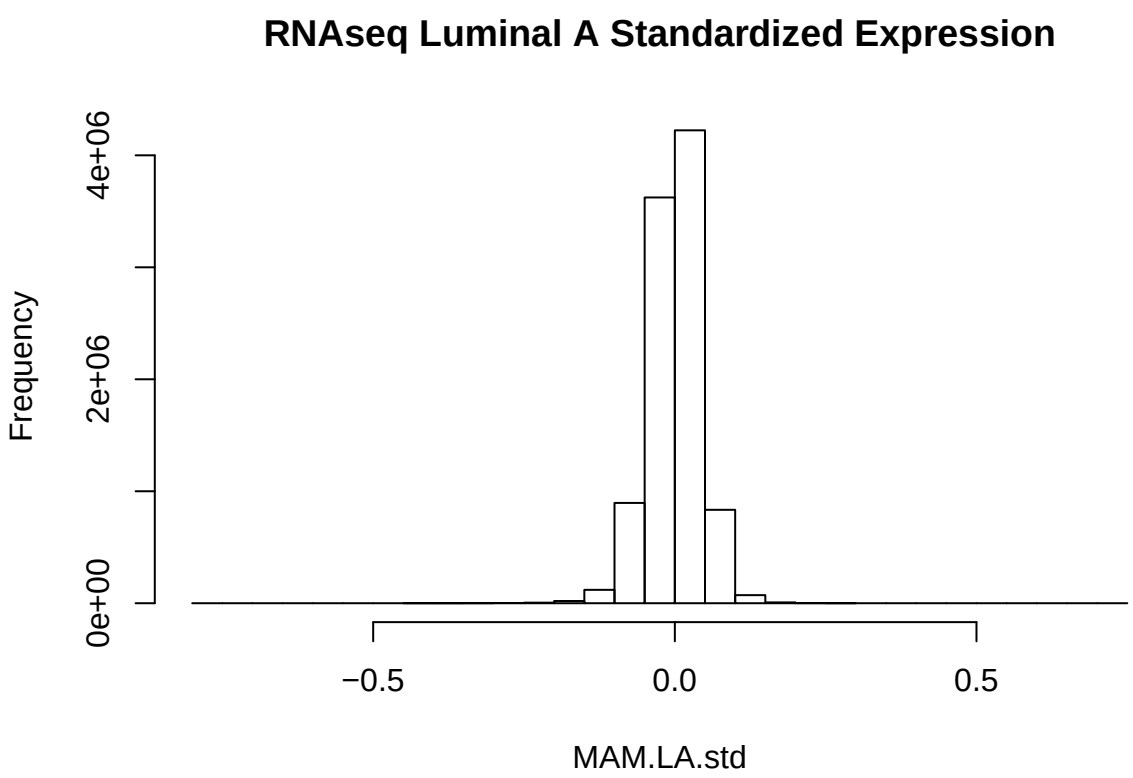


```
bad.genes.ma <- sd.v.ma/mad.v.ma > th
plot(sd.v.ma, mean.v.ma, col=ifelse(bad.genes.ma, "red", "black"), cex=0.3, main="Microarray Mean vs. SD
```

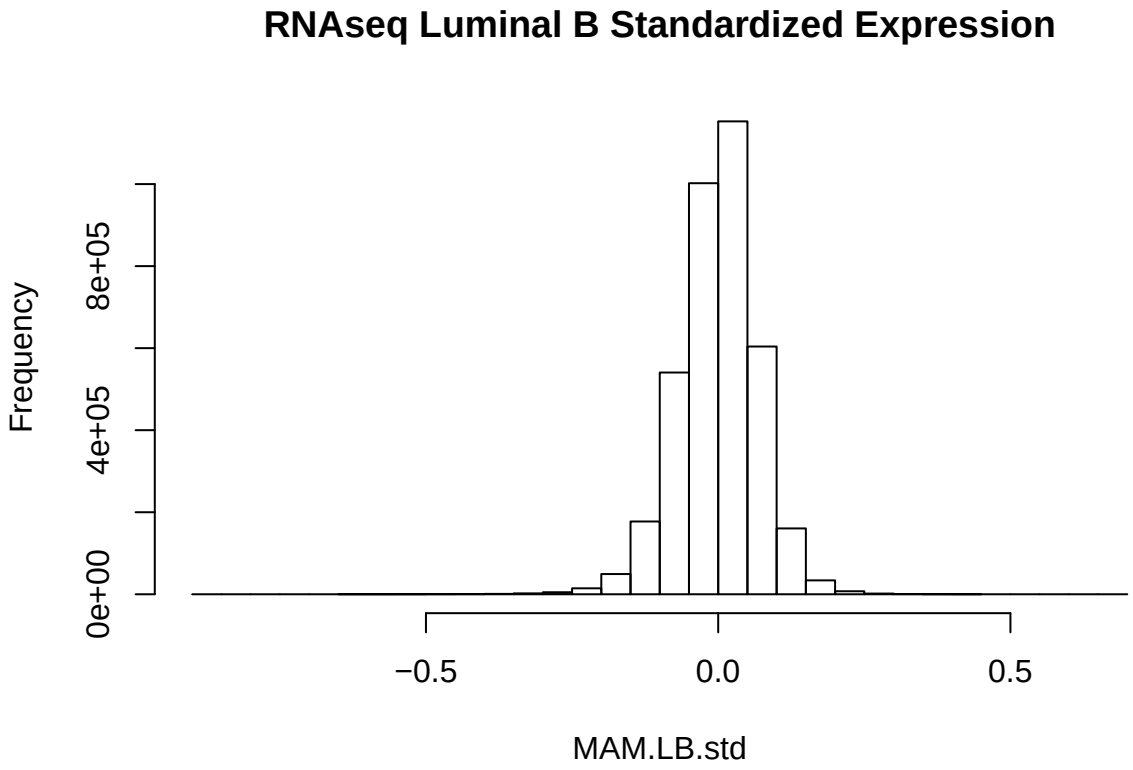



```
# Filter out bad genes.
dataM.good <- dataM[!bad.genes,]

# Create matrix for each group.
tar <- samTab0$Call
wb  <- which(tar=="Basal")
wla <- which(tar=="LumA")
wlb <- which(tar=="LumB")
MAM.basal <- dataM.good[, wb]
MAM.LA    <- dataM.good[, wla]
MAM.LB    <- dataM.good[, wlb]

# Standardize row within each group.
stdize <- function(gene){
  # Make zero mean.
  gene <- gene - mean(gene)

  # Make sum of squares one.
  gene <- gene/sqrt(sum(gene^2))

  return(gene)
}

MAM.LA.std <- MAM.LA
for(r in 1:nrow(MAM.LA)){
  MAM.LA.std[r,] <- stdize(MAM.LA[r,])
```

```
}

MAM.LB.std <- MAM.LB
for(r in 1:nrow(MAM.LB)){
  MAM.LB.std[r,] <- stdize(MAM.LB[r,])
}

MAM.basal.std <- MAM.basal
for(r in 1:nrow(MAM.basal)){
  MAM.basal.std[r,] <- stdize(MAM.basal[r,])
}

# Histogram of standardized RNAseq expression values.
hist(MAM.LA.std, main="RNAseq Luminal A Standardized Expression")
```

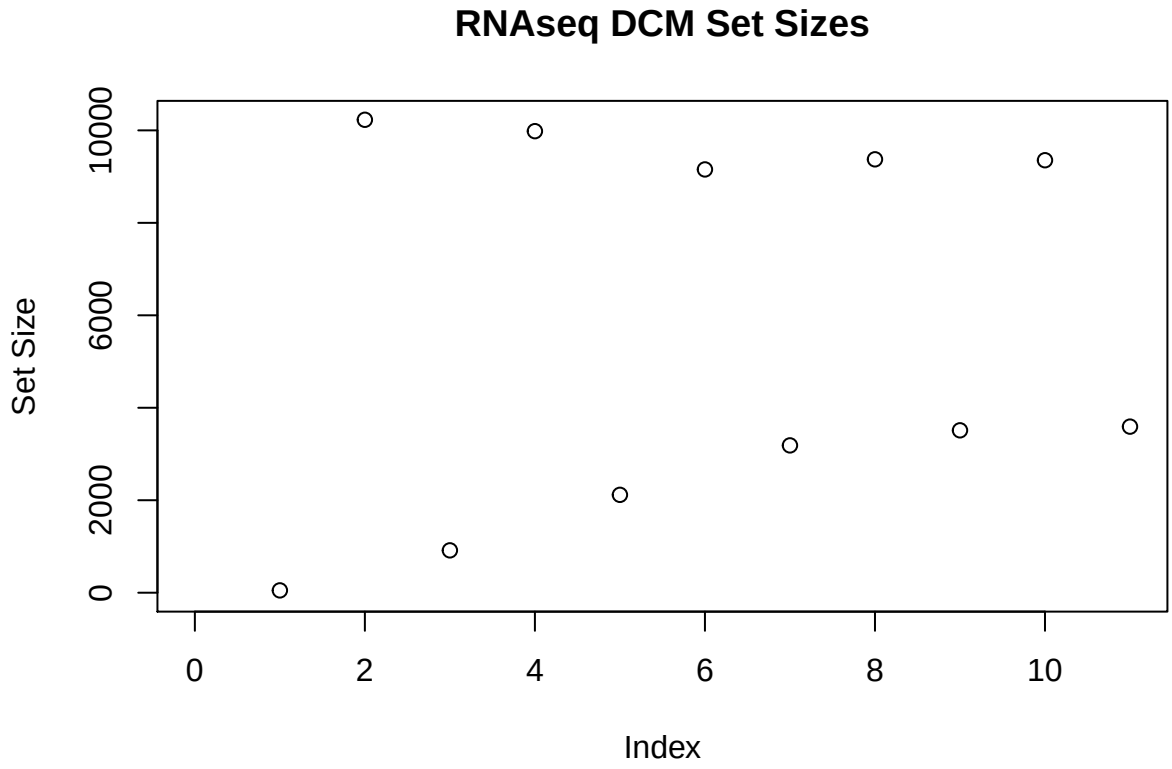



```
hist(MAM.LB.std, main="RNAseq Luminal B Standardized Expression")
```



Running DCM on the LumA vs LumB data, we find that it does not converge in 10 iterations. In fact, even the set sizes failed to converge. A plot of the set sizes can be found below.

```
load("/Users/kevinoconnor/Documents/Research/DCM/Test/2018_08_29 11:33:06/No QR/Debug_Output/DCM_1.RDat
plot(sapply(DCM$it_sets, length), xlim=c(0, 11), main="RNAseq DCM Set Sizes", ylab="Set Size")
```



We can compare this to the row-normalized microarray data which also did not converge.

```
load("/Users/kevinoconnor/Documents/Research/DCM/Test/2018_08_29 11:33:06/Microarray No QR Filtered by S
plot(sapply(DCM$it_sets, length), xlim=c(0, 11), main="RNAseq DCM Set Sizes", ylab="Set Size")
```