rnaSeq analysis

Christin M. Hong

Last updated 2018-06

Background

Objective: Analyzing processed microarray data

Team: Connie Cepko, Harvard Medical School

This data was provided as counts(?) in Excel spreadsheets.

```
#### Set seed for reproducibility ####
set.seed(200)


#### Set variables ####
# What's the working directory?  Preferably in same location as the data and Rproject.
getwd()
```

```
[1] "/home/christin/aaa_present_CepkoLab/lab_cepko-code/rnaSeq"
```

```
fname <- "dw_microarray_rpe-innateImmunity"


#### Import data ####
d <- read.csv("output/innate immune genes tissue microarraysgrant ed dw copy.csv")
# I ended up manually editing the sample names in this file because it was the easiest way to generate usable
# annotations...thankfully there were only 10 samples.  I'm sure there are more elegant ways of doing it,
# though (what if there were 100s of samples?!).


#### * File name * ####
out.name <-
    paste0("output/",
           fname,
           "_",
           "rnaSeq-analysis_",
           Sys.Date())

print(out.name)
```

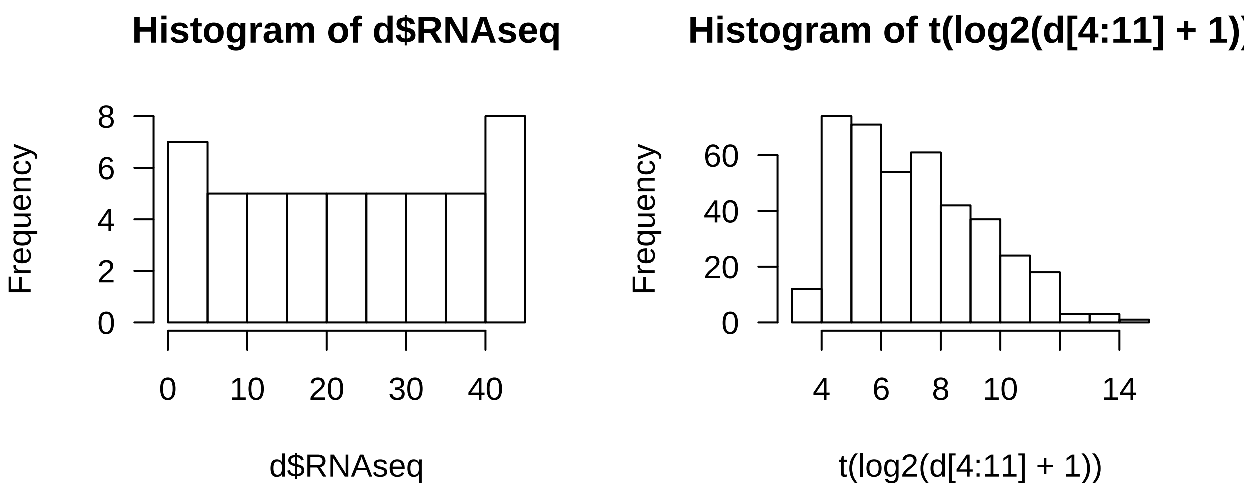
```
[1] "output/dw_microarray_rpe-innateImmunity_rnaSeq-analysis_2018-06-19"
```

```
#### START ANALYSIS ####

# look at number/distribution of reads, including only complete cases
sapply(d, class)
```

```
              n        Unigene           Name  C57Bl6_na_5w_a  C57Bl6_na_5w_b  C57Bl6_na_8w_a
      "integer"       "factor"       "factor"       "integer"       "integer"       "integer"
 C57Bl6_na_8w_b C57Bl6_RPE_6w_a    FVB_na_5w_a     FVB_na_5w_b    FVB_RPE_6w_a       gene.name
      "integer"      "integer"      "integer"       "integer"       "integer"        "factor"
         RNAseq
       "factor"
```

```
d <- transform(d, RNAseq = as.integer(RNAseq))

par(mfrow=c(1,2), las=1)

hist(d$RNAseq) # This is odd.  Are these p-values or read counts?
```

```r
# Graph log2(x+1) RNAseq values (log = natural log)
hist(t(log2(d[4:11] + 1)))
```



```r
par(mfrow=c(1,1), las=1)


#### GUESSES #### Are these raw reads...?  I hope not.  I guess I'll assume that these have at least been
#scaled to library size, since I don't have the full library to scale with anyway.  (Not sure if it's
#different for microarray data, but I'm still sure I'm only working with a subset of the data, so assuming
#that it comes scaled is still reasonable.)

# Will also assume data has already been filtered.

# It looks like there are 2 mouse strains, 3 timepoints, and 2 types of tissue (whole retina vs. RPE?).  There
# are replicates at the timepoints for the unlabeled tissue type, which I'm going to assume is whole retina.

# which ones are most useful for testing differential expression...?


#### Editing mislabeled values ####
# duplicate tnfaip8 in gene.name?
dplyr::filter(d, gene.name == "tnfaip8")
```

```
   n  Unigene                                          Name C57Bl6_na_5w_a C57Bl6_na_5w_b C57Bl6_na_8w_a
1 19 Mm.27740 tumor necrosis factor, alpha-induced protein 8             92             78             78
2 23 Mm.31403 tumor necrosis factor, alpha-induced protein 9             14             14             15
  C57Bl6_na_8w_b C57Bl6_RPE_6w_a FVB_na_5w_a FVB_na_5w_b FVB_RPE_6w_a gene.name RNAseq
1             62            1325          58          65          910   tnfaip8     42
2             13             667          14          29         1015   tnfaip8     42
```

```r
# Oh, I see, it's mislabeled in one row---should be tnfaip9 in row with Unigene Mm.31403.

dplyr::filter(d, grepl("26103", gene.name)) # filtering for partial match with grep
```

```
  n  Unigene                             Name C57Bl6_na_5w_a C57Bl6_na_5w_b C57Bl6_na_8w_a C57Bl6_na_8w_b
1 7 Mm.45995 RIKEN cDNA 2610307O08 gene sting             76             62             62             77
  C57Bl6_RPE_6w_a FVB_na_5w_a FVB_na_5w_b FVB_RPE_6w_a      gene.name RNAseq
1             247          86          77          234 2610307O08Rik     43
```

```r
# correcting
d$gene.name <- as.character(d$gene.name)

# subset for correct values (AND to keep values that are correct for all counts)
d.ok <- dplyr::filter(d, Unigene != "Mm.31403" & Unigene != "Mm.45995")

# subset for incorrect (OR to get all errors)
d.error <- dplyr::filter(d, Unigene == "Mm.31403" | Unigene == "Mm.45995")
```

```
d.error$gene.name[1] <- "sting"
d.error$gene.name[2] <- "tnfaip9"
d.error
```

```
   n  Unigene                                          Name C57Bl6_na_5w_a C57Bl6_na_5w_b C57Bl6_na_8w_a
1  7 Mm.45995               RIKEN cDNA 2610307O08 gene sting             76             62             62
2 23 Mm.31403 tumor necrosis factor, alpha-induced protein 9             14             14             15
  C57Bl6_na_8w_b C57Bl6_RPE_6w_a FVB_na_5w_a FVB_na_5w_b FVB_RPE_6w_a gene.name RNAseq
1             77             247          86          77          234     sting     43
2             13             667          14          29         1015   tnfaip9     42
```

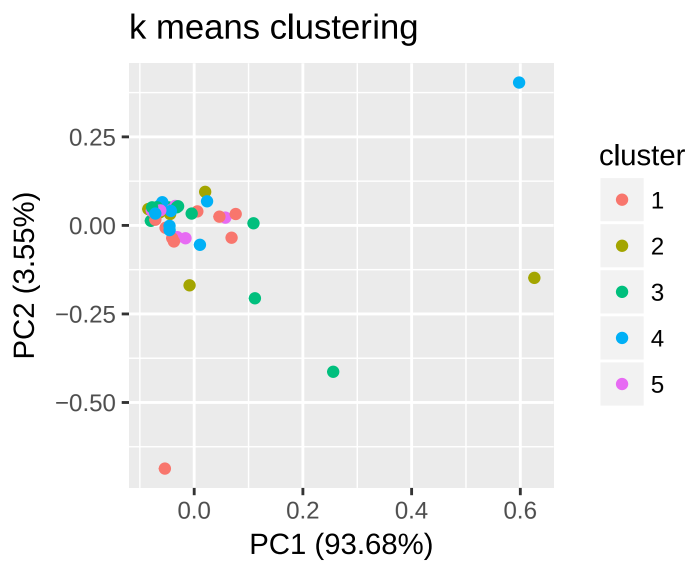
```
# merge
d.corrected <- rbind(d.ok, d.error)

# check
dplyr::filter(d.corrected, gene.name == "tnfaip8")
```

```
   n  Unigene                                          Name C57Bl6_na_5w_a C57Bl6_na_5w_b C57Bl6_na_8w_a
1 19 Mm.27740 tumor necrosis factor, alpha-induced protein 8             92             78             78
  C57Bl6_na_8w_b C57Bl6_RPE_6w_a FVB_na_5w_a FVB_na_5w_b FVB_RPE_6w_a gene.name RNAseq
1             62            1325          58          65          910   tnfaip8     42
```

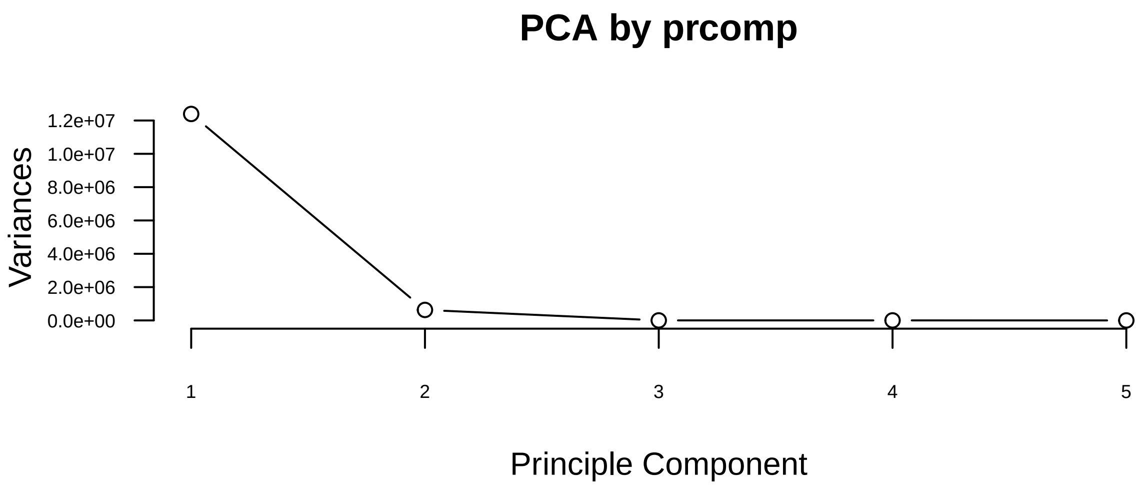
```
tail(d.corrected, 2)
```

```
    n  Unigene                                          Name C57Bl6_na_5w_a C57Bl6_na_5w_b C57Bl6_na_8w_a
49  7 Mm.45995               RIKEN cDNA 2610307O08 gene sting             76             62             62
50 23 Mm.31403 tumor necrosis factor, alpha-induced protein 9             14             14             15
   C57Bl6_na_8w_b C57Bl6_RPE_6w_a FVB_na_5w_a FVB_na_5w_b FVB_RPE_6w_a gene.name RNAseq
49             77             247          86          77          234     sting     43
50             13             667          14          29         1015   tnfaip9     42
```

```
#### tidying ####
# Renaming rows by gene name
d2 <- d.corrected
rownames(d2) <- d.corrected[, 12]

# log2(x+1) scaling
d.val <- log2(d2[4:11] + 1)


#### PCA and k means ####
autoplot(
    kmeans(
        d.val,
        centers = 5,
        iter.max = 100,
        nstart = 20,
        algorithm = "Lloyd"
    ),
    main = paste0("k means clustering"),
    data = d,
    asp = 1
)
```



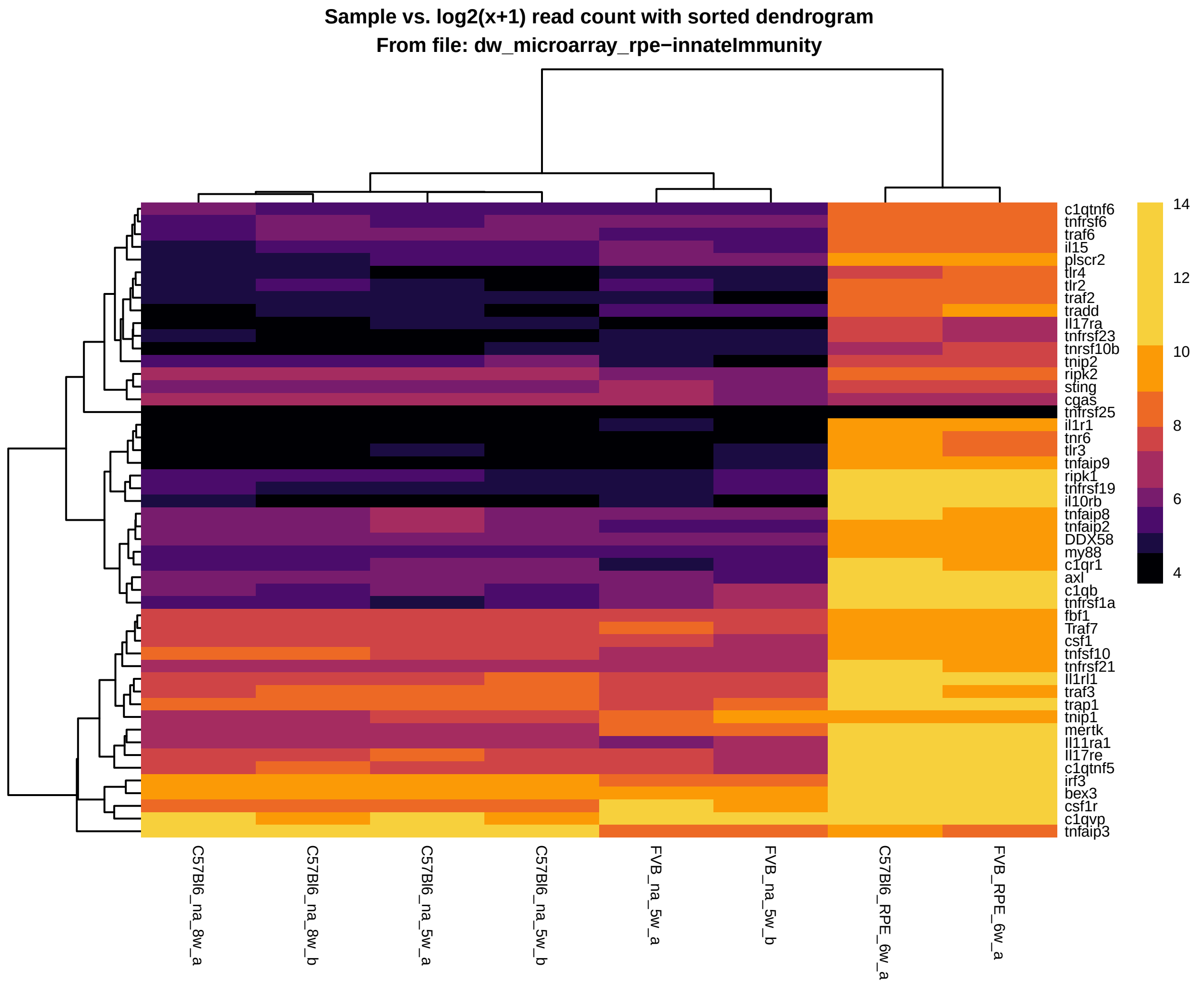
```r
flow_pcaPr <- prcomp(d[, c(4:8)], center = F, scale. = F)
par(cex.axis = 0.6)
plot(flow_pcaPr, type = "lines", main = "PCA by prcomp")
title(xlab = "Principle Component")
```



```r
par(cex.axis = 1)


#### Heatmap ####
# From Kam's tutorial: http://slowkow.com/notes/heatmap-tutorial/

# Setting coloring scheme based on quantile breaks so coloring will represent equal proportion of data
quantile_breaks <- function(xs, n = 10) {
    breaks <- quantile(xs, probs = seq(0 , 1, length.out = n))
    breaks[!duplicated(breaks)]
}

# Quantile function dislikes data.frames -> coerce into matrix
flowHM_breaks <- quantile_breaks(as.matrix(d.val), n = 11))


## Sorting the dendrograms for easier interpretation
# Cluster by column
flowHM_cluster_cols <- hclust(dist(t(d.val)))
```

```r
# Sort clustering by column
sort_hclust <- function(...) as.hclust(dendsort(as.dendrogram(...)))

flowHM_cluster_cols <- sort_hclust(flowHM_cluster_cols)

# Sorting for rows
flowHM_cluster_rows <- sort_hclust(hclust(dist(d.val)))
```

```r
## Plotting sorted heatmap
pheatmap(
    mat = d.val,
    color = inferno(10),
    border_color = NA,
    cellheight = 5,
    show_colnames = T,
    show_rownames = T,
    drop_levels = T,
    fontsize = 6,
    main = paste0("Sample vs. log2(x+1) read count with sorted dendrogram\nFrom file: ",
                  fname),
    kmeans_k = NA,
    breaks = flowHM_breaks,
    cluster_cols = flowHM_cluster_cols,
    cluster_rows = flowHM_cluster_rows
)
```


Sample vs. log2(x+1) read count with sorted dendrogram
From file: dw_microarray_rpe−innateImmunity
c1qtnf6
tnfrsf6
traf6
il15
plscr2
tlr4
tlr2
traf2
tradd
Il17ra
tnfrsf23
tnrsf10b
tnip2
ripk2
sting
cgas
tnfrsf25
il1r1
tnr6
tlr3
tnfaip9
ripk1
tnfrsf19
il10rb
tnfaip8
tnfaip2
DDX58
my88
c1qr1
axl
c1qb
tnfrsf1a
fbf1
Traf7
csf1
tnfsf10
tnfrsf21
Il1rl1
traf3
trap1
tnip1
mertk
Il11ra1
Il17re
c1qtnf5
irf3
bex3
csf1r
c1qvp
tnfaip3
C57Bl6_na_8w_a
C57Bl6_na_8w_b
C57Bl6_na_5w_a
C57Bl6_na_5w_b
FVB_na_5w_a
FVB_na_5w_b
C57Bl6_RPE_6w_a
FVB_RPE_6w_a