RNA-seq_Plotting

MTM

10/10/2020

Loading Souce Data

Loaded human log2FC data, normalized counts and sample guide from Posfai's Plos Pathogens paper, and lncRNA plus gene neighbor lists from Kaessman's Nature paper.

```
Hs_Log2FC <- read.csv("source_data/Hs_Log2FC.csv", header = TRUE)
lncRNAs_paper <- read.csv("source_data/lncRNAs_filtered_data_Kaessmann2019.csv", header= TRUE)
norm_counts_all <- read.csv("source_data/Norm_counts_all.csv", header = TRUE)
sample_guide <- read.csv("source_data/SampleGuide_data_dds.csv", header = TRUE)
```

Check data

```
str(Hs_Log2FC)
```

```
## 'data.frame':    196428 obs. of  7 variables:
##  $ Ensembl.ID      : Factor w/ 32738 levels "ENSG00000000003",..: 9245 2666 11831 7874 11519 12533 3
##  $ Gene.name       : Factor w/ 32641 levels "1-Dec","1-Mar",..: 14980 29377 5108 4174 3172 12073 656
##  $ log2FoldChange  : num  4.7 3.99 -2.73 3.47 2.42 ...
##  $ pvalue          : num  1.07e-14 3.44e-13 1.89e-12 1.87e-11 1.90e-11 ...
##  $ adjusted.p.value: num  1.76e-10 7.74e-09 4.02e-08 2.10e-07 1.55e-07 6.46e-07 3.43e-07 2.08e-06 2.
##  $ Cell.line       : Factor w/ 2 levels "HepG2","HuH7": 1 1 1 1 1 2 1 1 2 2 ...
##  $ time.point      : Factor w/ 3 levels "early","late",..: 3 2 1 2 3 1 2 3 1 1 ...
```

```
str(lncRNAs_paper)
```

```
## 'data.frame':    182544 obs. of  10 variables:
##  $ X                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ XLOC.id            : Factor w/ 182544 levels "Chi_XLOC_000004",..: 16840 16841 16842 16843 16844
##  $ Species            : Factor w/ 7 levels "Chicken","Human",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ chromosome         : Factor w/ 1023 levels "1","1.09921E+12",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Dynamic            : logi  FALSE FALSE FALSE FALSE TRUE TRUE ...
##  $ ENSEMBL75.id       : Factor w/ 10893 levels "ENSG00000031544",..: 3604 NA 3114 3198 1509 NA 389 NA
##  $ LncRNA.name        : Factor w/ 10305 levels "0610005C13Rik",..: 5132 NA 9632 6338 5023 NA 7623 NA
##  $ Minimum.age        : Factor w/ 11 levels "180Mya","20Mya",..: 10 10 8 8 8 8 8 8 8 8 ...
##  $ Genomic.class      : Factor w/ 8 levels "antisense","divergent_BT",..: 8 7 8 3 7 8 4 1 2 1 ...
##  $ Nearest.coding.gene: Factor w/ 21739 levels "ENSG00000000005",..: 9216 10696 10667 11091 11074 11
```

```
str(norm_counts_all)
```

```
## 'data.frame':    32738 obs. of  41 variables:
##  $ X         : Factor w/ 32738 levels "ENSG00000000003",..: 24279 23552 15844 23678 23859 23572 2385
##  $ X50001_N_1: num  0 0 0 0 0 ...
##  $ X50000_N_1: num  0 0 0 2.65 0 ...
##  $ X50002_N_1: num  0 0 0 1.73 0 ...
```

```
##  $ X50003_N_1: num  0 0 0 0 0 ...
##  $ X50005_N_1: num  0 0 0 2.2 0 ...
##  $ X50006_N_1: num  0 0 1.65 5.79 0 ...
##  $ X50007_N_1: num  0 0 0 4.64 0 ...
##  $ X50008_N_1: num  0 0 0 0 0 ...
##  $ X50009_N_1: num  0 0 0 4.89 0 ...
##  $ X50011_N_1: num  0 0 0 3.29 0 ...
##  $ X50013_N_1: num  0 0 0 1.58 0 ...
##  $ X50014_N_1: num  0.639 0 0 2.557 0 ...
##  $ X50015_N_1: num  0 0 0 6.11 0 ...
##  $ X50018_N_1: num  0 0 0 4.86 0 ...
##  $ X50019_N_1: num  0 0 1.35 2.02 0 ...
##  $ X50021_N_1: num  0 0 0 1.93 0 ...
##  $ X50022_N_1: num  0 0 0 1.74 0 ...
##  $ X50023_N_1: num  0 0 0.657 5.257 0 ...
##  $ X50024_N_1: num  0 0 0 5.25 0 ...
##  $ X50025_N_1: num  0 0 0 0 0 ...
##  $ X50026_N_1: num  0 0 1.67 11.69 0 ...
##  $ X50027_N_1: num  0 0 0 2.33 0 ...
##  $ X50030_N_1: num  0 0 0 0 0 ...
##  $ X50031_N_1: num  0 0 0 0 0 ...
##  $ X50033_N_1: num  0 0 0 1.81 0 ...
##  $ X50034_N_1: num  0 0 0 5.11 0 ...
##  $ X50035_N_1: num  0 0 0 7.58 0 ...
##  $ X50036_N_1: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ X50038_N_1: num  0 0 0 0 0 ...
##  $ X50039_N_1: num  0 0 0 0 0 ...
##  $ X50040_N_1: num  0 0 0 0 0 ...
##  $ X50041_N_1: num  0 0 0 4.78 0 ...
##  $ X50042_N_1: num  0 0 0 0 0 ...
##  $ X50043_N_1: num  0 0 0 0 0 ...
##  $ X50044_N_1: num  0 0 0 0 0 ...
##  $ X50045_N_1: num  0 0 0 0 0 ...
##  $ X50046_N_1: num  0 0 0 0 0 ...
##  $ X50047_N_1: num  0 0 0 0 0 ...
##  $ X50048_N_1: num  0.852 0 0 0 0 ...
##  $ X50049_N_1: num  0 0 0 0 0 ...
```

```
str(sample_guide)
```

```
## 'data.frame':    40 obs. of  6 variables:
##  $ X              : Factor w/ 40 levels "50000_N_1","50001_N_1",..: 2 1 3 4 5 6 7 8 9 10 ...
##  $ count          : num  59.1 72.9 68 64.5 60.9 ...
##  $ Cell.line      : Factor w/ 2 levels "HepG2","HuH7": 2 2 2 2 2 2 2 2 2 2 ...
##  $ time.groups    : Factor w/ 3 levels "early","late",..: 1 NA 1 1 1 3 2 2 NA 1 ...
##  $ analysis.groups: Factor w/ 8 levels "HepG2.control",..: 6 5 6 6 6 8 7 7 5 6 ...
##  $ time.groups2   : Factor w/ 4 levels "control","early",..: 2 1 2 2 2 4 3 3 1 2 ...
```

Clean up and transform dataframes

1. Transform categorical data into factors

```
Hs_Log2FC$Cell.line <- as.factor(Hs_Log2FC$Cell.line)
Hs_Log2FC$time.point <- as.factor(Hs_Log2FC$time.point)
sample_guide$Cell.line <- as.factor(sample_guide$Cell.line)
sample_guide$time.groups2 <- as.factor(sample_guide$time.groups2)
```

2. Rename some columns for simplicity.

```
lncRNAs_paper <- dplyr::rename(lncRNAs_paper, Ensembl.ID=ENSEMBL75.id)
norm_counts_all <- dplyr::rename(norm_counts_all, Ensembl.ID=X )
sample_guide <- dplyr::rename(sample_guide, sample.ID=X)
```

Plotting DEGs

1. Create variables that contain threshold criteria. The lfc.cutoff is set to 0.58; remember that we are working with log2 fold changes so this translates to an actual fold change of 1.5.

```
### Set thresholds

padj.cutoff <- 0.05
logfc.cutoff <- 0.58
```

2. Create vector that helps us identify the genes that meet our criteria:

```
threshold<-Hs_Log2FC$adjusted.p.value<padj.cutoff & abs(Hs_Log2FC$log2FoldChange)>logfc.cutoff
```

We now have a logical vector of values that has a length which is equal to the total number of genes in the dataset. The elements that have a TRUE value correspond to genes that meet the criteria (and FALSE means it fails). How many genes are differentially expressed in infected compared to Control, given our criteria specified above? Does this reduce our results?

```
length(which(threshold))
```

```
## [1] 264
```

To add this vector to our results table we can use the $ notation to create the column on the left hand side of the assignment operator, and the assign the vector to it instead of using cbind():

```
Hs_Log2FC$threshold<-threshold
```

Now we can easily subset the results table to only include those that are significant using the subset() function:

```
sig_DE_all<-data.frame(subset(Hs_Log2FC,threshold==TRUE))
```

Subset lncRNAs from sig_DE_all

1. Need to clean up lncRNA table

```
#check how many lncRNAs are in table from paper

length(lncRNAs_paper$Ensembl.ID)
```

```
## [1] 182544
```

```
#remove na in r - remove rows - na.omit function / option

lncRNAs_paper <-na.omit(lncRNAs_paper)
```

NA values have been removed. Here I am not sure if I removed IDs that did had an NAs in some other column, but I think it doesn't matter.

2.Subset lncRNAs

```
sig_DE_lncRNA <- sig_DE_all[sig_DE_all$Ensembl.ID %in% lncRNAs_paper$Ensembl.ID,]
print(sig_DE_lncRNA)
```

```
##          Ensembl.ID      Gene.name log2FoldChange      pvalue adjusted.p.value
## 7   ENSG00000267934   CTB-176F20.3       3.009213 4.58000e-11      0.000000343
## 46  ENSG00000233246   RP11-415J8.5      -2.985132 9.03000e-07      0.004375660
## 50  ENSG00000224189       HOXD-AS1      -1.381031 1.03000e-06      0.004375660
## 59  ENSG00000245648 RP11-277P12.20       1.907498 1.83000e-06      0.002072254
## 65  ENSG00000260710  RP11-616M22.7       2.475895 2.96000e-06      0.002546216
## 79  ENSG00000264569  RP13-650J16.1      -1.563193 4.36000e-06      0.003363469
## 82  ENSG00000267934   CTB-176F20.3       1.682409 4.73000e-06      0.029937621
## 107 ENSG00000227857    RP4-533D7.5       2.592566 1.13000e-05      0.017175348
## 109 ENSG00000240498     CDKN2B-AS1       1.600369 1.17000e-05      0.006606007
## 127 ENSG00000224093  RP5-1033H22.2      -2.178604 2.14000e-05      0.025271331
## 128 ENSG00000237686  RP5-1120P11.1      -1.477782 2.22000e-05      0.011062147
## 136 ENSG00000266903    CTB-171A8.1      -1.656562 2.57000e-05      0.027349680
## 148 ENSG00000233255     AC019181.2      -1.612433 3.05000e-05      0.014382847
## 161 ENSG00000231721      LINC-PINT      -1.704573 3.81000e-05      0.031147379
## 163 ENSG00000245648 RP11-277P12.20       1.800723 3.85000e-05      0.034405348
## 181 ENSG00000229656   RP11-462L8.1       2.028474 5.59000e-05      0.018641326
## 186 ENSG00000267308     AC004510.3       2.804807 5.97000e-05      0.045838478
## 188 ENSG00000245532          NEAT1       1.461839 6.12000e-05      0.045838478
## 244 ENSG00000259347   RP11-798K3.2      -1.195173 1.08638e-04      0.030713876
## 277 ENSG00000259867  RP11-525K10.3      -1.857503 1.56850e-04      0.036119672
## 307 ENSG00000224272     AC114730.3      -1.155861 2.01106e-04      0.045898901
##     Cell.line time.point threshold
## 7       HepG2       late      TRUE
## 46      HepG2      early      TRUE
## 50      HepG2      early      TRUE
## 59       HuH7      early      TRUE
## 65      HepG2        mid      TRUE
## 79       HuH7      early      TRUE
## 82       HuH7       late      TRUE
## 107     HepG2      early      TRUE
## 109      HuH7      early      TRUE
## 127     HepG2      early      TRUE
## 128      HuH7      early      TRUE
## 136     HepG2      early      TRUE
## 148      HuH7      early      TRUE
## 161     HepG2      early      TRUE
## 163      HuH7        mid      TRUE
## 181     HepG2        mid      TRUE
## 186     HepG2       late      TRUE
## 188     HepG2       late      TRUE
## 244      HuH7      early      TRUE
## 277     HepG2        mid      TRUE
## 307      HuH7      early      TRUE
```

Visualizing DE lncRNAs

Load packages that will be needed

```
# Load libraries

library(reshape)
library(ggplot2)
library(ggrepel)
library(RColorBrewer)
library(pheatmap)
```

Before it gets more complicated, I am going to split my data per cell line.

```
sig_DE_lncRNA_hepg2<-sig_DE_lncRNA[sig_DE_lncRNA$Cell.line=="HepG2",]
sig_DE_lncRNA_huh7<-sig_DE_lncRNA[sig_DE_lncRNA$Cell.line=="HuH7",]
```

I am going to focus on HepG2 for now

Extract the normalized count values for these genes by subsetting.

```
sig_DE_lncRNA_hepg2_counts <- norm_counts_all[norm_counts_all$Ensembl.ID %in% sig_DE_lncRNA_hepg2$Ensembl

# use first column for row names

sig_DE_lncRNA_hepg2_counts <- data.frame(sig_DE_lncRNA_hepg2_counts, row.names = 1)
```

Backtrack and get only HepG2 samples

```
#transform analysis.groups to factors

sample_guide$analysis.groups <- as.factor(sample_guide$analysis.groups)

#extract HepG2 samples

HepG2samples<-sample_guide[sample_guide$Cell.line=="HepG2",c("sample.ID","analysis.groups")]
```

```
remove(siglncHepG2_counts)
```

```
## Warning in remove(siglncHepG2_counts): object 'siglncHepG2_counts' not found
```

```
remove(counts_sigLncHepG2)
```

```
## Warning in remove(counts_sigLncHepG2): object 'counts_sigLncHepG2' not found
```

```
library(data.table)

# get data

  data(sig_DE_lncRNA_hepg2_counts)
```

```
## Warning in data(sig_DE_lncRNA_hepg2_counts): data set
## 'sig_DE_lncRNA_hepg2_counts' not found
```

```
# transpose

  t_sigHepG2counts <- transpose(sig_DE_lncRNA_hepg2_counts)

colnames(t_sigHepG2counts) <- rownames(sig_DE_lncRNA_hepg2_counts)
  rownames(t_sigHepG2counts) <- colnames(sig_DE_lncRNA_hepg2_counts)
```

```
#make rownames a new column

  setDT(t_sigHepG2counts, keep.rownames = "sample.ID")
```

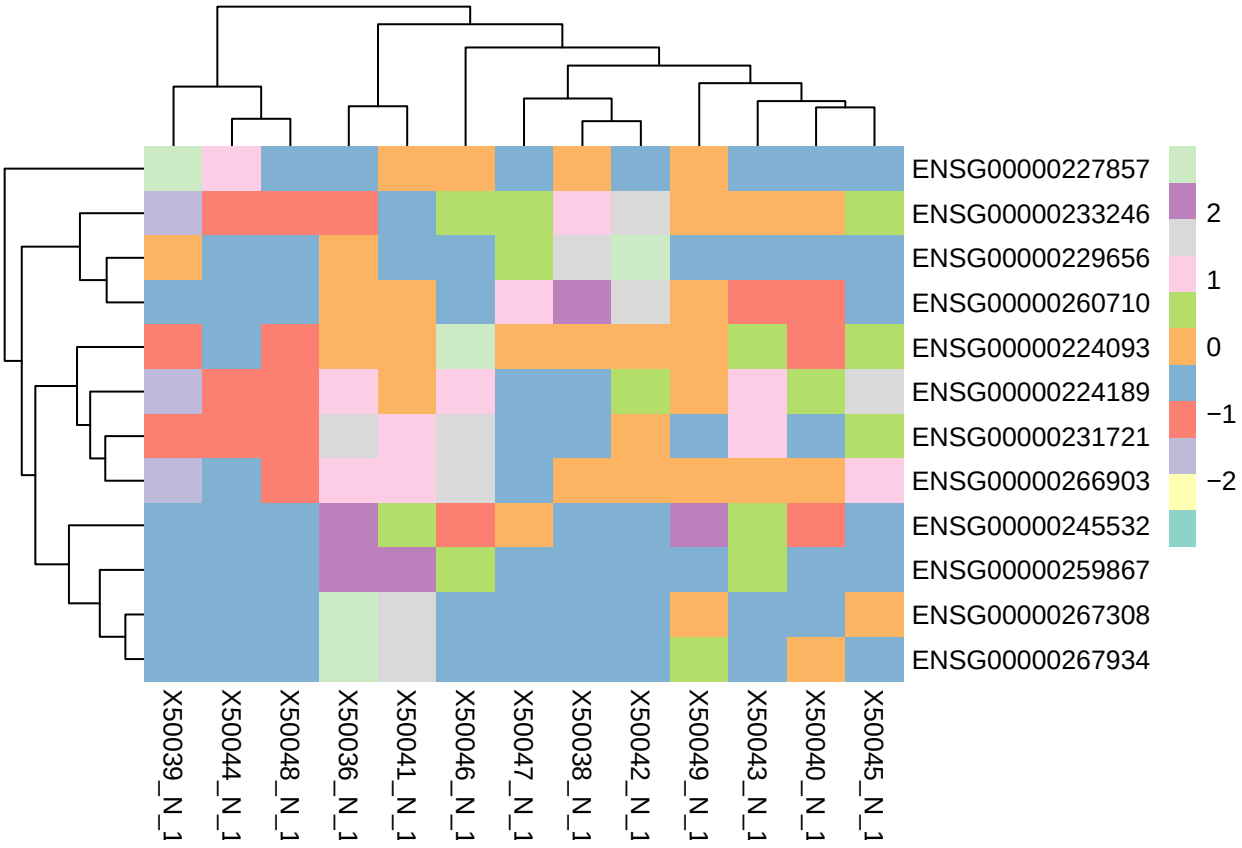
Now subset

THERE IS AN ERROR IN THE SIG_HEPG2 COUNTS SAMPLE IDs. An X was added at some point in front of the name.

```
#subset by indexing columns

sigHepG2counts<-sig_DE_lncRNA_hepg2_counts[,c(28:40)]
```

Plot

```
#load packages

library(pheatmap)

### Set a color palette

heat.colors <- brewer.pal(11, "Set3")

### Run pheatmap

pheatmap(sigHepG2counts, color = heat.colors, cluster_rows = T, show_rownames=T,
 border_color=NA, fontsize = 10, scale="row", fontsize_row = 10, height=5, width = 5)
```


ENSG00000227857
ENSG00000233246
ENSG00000229656
ENSG00000260710
ENSG00000224093
ENSG00000224189
ENSG00000231721
ENSG00000266903
ENSG00000245532
ENSG00000259867
ENSG00000267308
ENSG00000267934
2
1
0
−1
−2
X50039_N_1
X50044_N_1
X50048_N_1
X50036_N_1
X50041_N_1
X50046_N_1
X50047_N_1
X50038_N_1
X50042_N_1
X50049_N_1
X50043_N_1
X50040_N_1
X50045_N_1

‘‘