Julie Garcia
January 31, 2018

HW 1
Advanced Genomics and Genetics Analyses

In this lab, we will be analyzing exon array data from a set of drug-treated cells from GEO (accession GSE19891). The summary and design for this study are both copied below.

Summary:
Alternative splicing analysis after treatment with three clinically approved drugs
Bioactive compounds have been invaluable for dissecting the mechanisms, regulation and functions of cellular processes. However, very few such reagents have been described for pre-mRNA splicing. To facilitate their systematic discovery, we developed a high throughput cell-based assay that measures pre-mRNA splicing utilizing a quantitative reporter system with advantageous features. The reporter, consisting of a destabilized, intron-containing luciferase expressed from a short-lived mRNA, allows rapid screens (<4 hr) thereby obviating potential toxicity of splicing inhibitors. We describe three inhibitors (out of >23,000 screened), all pharmacologically active: clotrimazole, flunarizine and chlorhexidine. Interestingly, none was a general splicing inhibitor. Rather, each caused distinct splicing changes of numerous genes.

Overall design:
Treated HeLa cells with clotrimazole, flunarizine, chlorhexidine or DMSO for 6 hours followed by analysis of total RNA on human exon 1.0 ST platform. Three biological replicates were used for each treatment.

Sample annotation information

GSM497231	Control DMSO 6hrs 1
GSM497232	Control DMSO 6hrs 2
GSM497233	Control DMSO 6hrs 3
GSM497234	Treated Clotrimazole 6hrs 1
GSM497235	Treated Clotrimazole 6hrs 2
GSM497236	Treated Clotrimazole 6hrs 3
GSM497237	Treated Flunarizine 6hrs 1
GSM497238	Treated Flunarizine 6hrs 2
GSM497239	Treated Flunarizine 6hrs 3
GSM497240	Control for chlorhexidine DMSO 6hrs 1
GSM497241	Control for chlorhexidine DMSO 6hrs 2
GSM497242	Control for chlorhexidine DMSO 6hrs 3
GSM497243	Treated Chlorhexidine 6hrs 1
GSM497244	Treated Chlorhexidine 6hrs 2
GSM497245	Treated Chlorhexidine 6hrs 3

1.) Download the GSE19891_RAW.tar file and decompress it. This file contains the CEL files that we will be working with. Next, run the apt-probeset-summarize function to obtain gene, exon, and dabg values for these CEL files. Make sure to use rma-sketch normalization and only the core probeset (or metaprobeset) library files.

```
apt-probeset-summarize \
  -p ~/datasets/apt/HuEx-1_0-st-v2.r2.pgf \
  -c ~/datasets/apt/HuEx-1_0-st-v2.r2.clf \
  -b ~/datasets/apt/HuEx-1_0-st-v2.r2.antigenomic.bgp \
  --qc-probesets ~/datasets/apt/HuEx-1_0-st-v2.r2.qcc \
  -m ~/datasets/apt/HuEx-1_0-st-v2.r2.dt1.hg18.core.mps \
  -a rma-sketch \
  -o output-dir/gene \
  /cels/exon/keep/*.CEL

apt-probeset-summarize \
  -p ~/datasets/apt/HuEx-1_0-st-v2.r2.pgf \
  -c ~/datasets/apt/HuEx-1_0-st-v2.r2.clf \
  -b ~/datasets/apt/HuEx-1_0-st-v2.r2.antigenomic.bgp \
  --qc-probesets ~/datasets/apt/HuEx-1_0-st-v2.r2.qcc \
  -s ~/datasets/apt/HuEx-1_0-st-v2.r2.dt1.hg18.core.ps \
  -a rma-sketch \
  -o ~/datasets/output/exon \
  /cels/exon/keep/*.CEL

apt-probeset-summarize \
  -p ~/datasets/apt/HuEx-1_0-st-v2.r2.pgf \
  -c ~/datasets/apt/HuEx-1_0-st-v2.r2.clf \
  -b ~/datasets/apt/HuEx-1_0-st-v2.r2.antigenomic.bgp \
  --qc-probesets ~/datasets/apt/HuEx-1_0-st-v2.r2.qcc \
  -s ~/datasets/apt/HuEx-1_0-st-v2.r2.dt1.hg18.core.ps \
  -a dabg \
  -o ~/datasets/output/exon \
  /cels/exon/keep/*.CEL
```

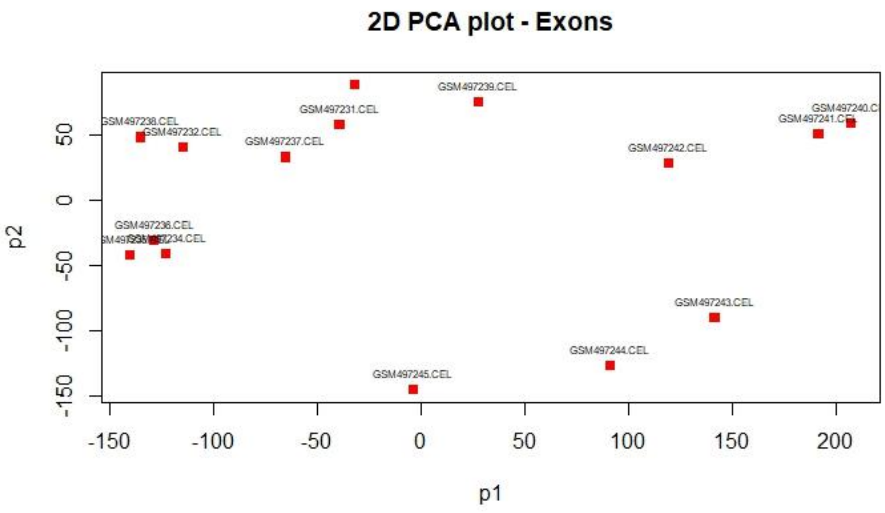
2.) Read in the Exon array annotation file (from the course website), and the exon matrix, and gene matrix. There are so many groups here that we'll not filter the probesets for this assignment and let the statistical significance hopefully extract the most robust differences. **Make sure to print out the dimensions of each data matrix to verify you skipped the appropriate number of lines.**

```
e = read.table("c:\\temp\\datasets\\homework\\exon-rma-
sketch.summary.txt", header=T, row.names=1)
dim(e)

  [1] 287329     15
```

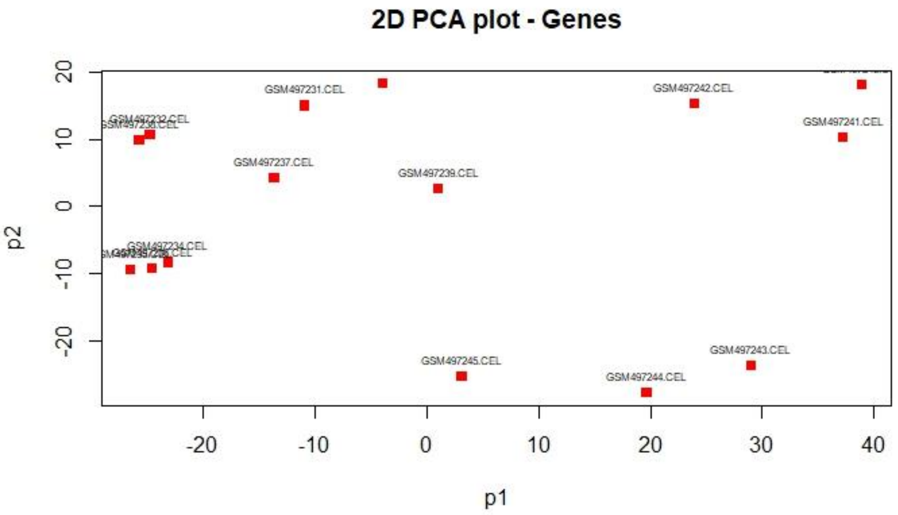
```
g = read.table("c:\\temp\\datasets\\homework\\gene-rma-
sketch.summary.txt", header=T, row.names=1)
dim(g)

  [1] 22011    15


p = read.table("c:\\temp\\datasets\\homework\\dabg.summary.txt",
header=T, row.names=1)
dim(p)

  [1] 287329     15


map = read.csv("c:\\temp\\datasets\\homework\\HuEx-1_0-st-
v2.na24.hg18.probeset_abbr.csv", header=T, row.names=1)
dim(map)

  [1] 1425647       7
```

3.) Use PCA to identify any outliers in the gene and/or exon matrices. Create a plot of the exon values as well as the gene values (i.e. 2 plots). Label the plots appropriately. **Do any of the treatment conditions have more variability between replicates than others?**

```
e.pca <- prcomp(t(e))
e.loads <- e.pca$x[,1:2]
plot(e.loads[,1],e.loads[,2],main="2D PCA plot - Exons",
     xlab="p1",ylab="p2",col='red',cex=1,pch=15)
text(e.loads,label=dimnames(e)[[2]],pos=3,cex=0.5)
```



```
g.pca <- prcomp(t(g))
g.loads <- g.pca$x[,1:2]
plot(g.loads[,1],g.loads[,2],main="2D PCA plot - Genes",
     xlab="p1",ylab="p2",col='red',cex=1,pch=15)
text(g.loads,label=dimnames(g)[[2]],pos=3,cex=0.5)
```



```
GSM497243, GSM497244, GSM497245 all seem to have more variability
than the others.
```

4.) We want to next compare the differentially expressed exons between the antifungal drug Clotrimazole and its control, then between the anti-bacterial drug Chlorhexidine and its respective control. Use the code already provided to create NI values and calculate normalized fold changes with associated p-values from the t-test for each of the comparisons. You will have to modify the t.two() function from the lecture notes to extract the $\log_2$ fold changes.

Be sure to first intersect the exon probes in common between the annotation file and the exon matrix. Then subset the annotation file and exon matrices by these common probes (we are doing no filtering, remember). Next, get the unique transcript cluster IDs for the loop. Be sure to print out dimensions of the matrices throughout each of these steps to make it easier to debug issues. Also note that the loop for each treatment comparison took >2 hours to run, so plan accordingly.

```
#intersect matching probes between annotation file and data
matrix
x <- intersect(dimnames(e)[[1]],dimnames(map)[[1]])
legnth(x)

  [1] 287220

e.dat <- e[x,]
```

```
dim(e.dat)

  [1] 287220     15

p.dat <- p[x,]
dim(p.dat)

  [1] 287220     15

map.dat <- map[x,]
dim(map.dat)

  [1] 287220     7

#get unique transcript cluster IDs (gene IDs) from annotation
file
u <- unique(as.character(map.dat$transcript_cluster_id))
u <- intersect(u,dimnames(g)[[1]])
length(u)

  [1] 17881

# two t-test called by exon ni function below
t.two <- function(x,sam,v=F) {
  x <- as.numeric(x)
  out <-
t.test(as.numeric(x[sam]),as.numeric(x[!sam]),alternative="two.si
ded",var.equal=v)
  control <- mean(log2(x[sam]), 1)
  test <- mean(log2(x[!sam]), 1)
  fold <- control - test
  o <-
as.numeric(c(out$statistic,out$p.value,out$conf.int[1],out$conf.i
nt[2], fold))
  names(o) <- c("test_statistic","pv","lower_ci","upper_ci",
"fold change")
  return(o)
}

# exon ni function
exon.ni <- function(genex,exonx,rx) {
  ni <- t(t(exonx)-genex)
  ttest <- t(apply(ni,1,t.two,sam=as.logical(rx),v=F))
  return(ttest)
}

# Clotrimazole vs. control
pvalues.clotrim <- data.frame(test_statistic=as.numeric(),
pv=as.numeric(),
                      lower_ci=as.numeric(),
upper_ci=as.numeric(),
                      fold=as.numeric())
r.clotrim = c(0,0,0,1,1,1)

for (uniqId in u) {
  uniqId = u[1]
  ex <- dimnames(map.dat[map.dat$transcript_cluster_id %in%
uniqId,])[[1]]
  d.exon <- e.dat[ex,1:6]
  d.gene <- g[uniqId,1:6]
  if(dim(d.exon)[[1]] > 2) {
    ni.out <-
exon.ni(genex=as.numeric(d.gene),exonx=d.exon,rx=r.clotrim)
    pvalues <- rbind(pvalues, ni.out)
  }
}

# Chlorhexidine vs. control
pvalues.clotrim <- data.frame(test_statistic=as.numeric(),
pv=as.numeric(),
                              lower_ci=as.numeric(),
upper_ci=as.numeric(),
                              fold=as.numeric())
r.chlorhex = c(0,0,0,1,1,1)

for (uniqId in u) {
  uniqId = u[1]
  ex <- dimnames(map.dat[map.dat$transcript_cluster_id %in%
uniqId,])[[1]]
  d.exon <- e.dat[ex,10:15]
  d.gene <- g[uniqId,10:15]
  if(dim(d.exon)[[1]] > 2) {
    ni.out <-
exon.ni(genex=as.numeric(d.gene),exonx=d.exon,rx=r.chlorhex)
    pvalues <- rbind(pvalues, ni.out)
  }
}
```

5.) **How many exon probes are significantly enriched or depleted using a threshold of p<.01 and fold change (both directions) greater than 1.5? How many unique significant transcript clusters (using our threshold above) are in common between the 2 drugs?**

6.) We now want to see that first, there is significant evidence to support the fact that these identified exon probes are associated with known splicing events. Second, we want to see if there is any common biological function that supports the known mechanism of these two compounds.

Write out the transcript cluster IDs (keep duplicates) to a file for each of the compounds and use the Functional Annotation Chart tools in NCBI's DAVID to see what the top functional categories are for each of these compounds**. Do these enriched functions seem to support what is known about these compounds?**

7.) Now let's take a look at some of the splice events in the most significantly enriched/depleted exons within specific transcript clusters. Use a threshold of p<.001 and |FC|>2 and get the intersecting transcript clusters between the 2 compounds. **Print these transcript cluster IDs to the screen.**

8.) Plot the boxplots using the plot.exons() function for each of the intersecting transcript clusters within both compounds (i.e. 2 plots per cluster ID). You should modify the plot.exons() function so the title of each plot indicates which treatment you are plotting for a transcript ID.