

Looking and visualizing individual genes: PIN1

date: June 25, 2014

The goal of this report is to look at PIN1 genes in the LCM data.

```
library(rmarkdown)
render("curatedGenes.Rmd", "pdf_document")
```

```
library(reshape2)
library(ggplot2)
library(plyr)
```

Read in Data

```
countData <- read.csv("../data/normalized_read_count.csv")
geneList <- read.csv("pnas.1402835111.sd06.csv")
```

Set up the dataframe for analysis and visualization

```
#Melt Data
countData <- melt(countData)
```

```
## Using X as id variables
```

```
colnames(countData) <- c("gene", "sample", "count")
```

```
#set genotype
```

```
countData$genotype <- ifelse(grepl("wt", countData$sample, ignore.case = T), "wt",
                             ifelse(grepl("tf2", countData$sample, ignore.case = T), "tf2", "unknown"))
```

```
#set type
```

```
countData$tissue <- ifelse(grepl("other", countData$sample, ignore.case = T), "other",
                           ifelse(grepl("mbr", countData$sample, ignore.case = T), "mbr", "unknown"))
```

```
#Set Region
```

```
countData$region <- ifelse(grepl("a", countData$sample, ignore.case = T), "A",
                           ifelse(grepl("c", countData$sample, ignore.case = T), "C", "B"))
```

```
#Set tissueRegion
```

```
countData$type <- paste(countData$region, countData$tissue, sep = "")
```

Subset for each gene

First I take the 384 genes and make them into a characters, so that I can find them and loop through.

```
genes <- geneList[,1]
genesOfInterest <- levels(genes)
```

The loop:

```
for(GENE in genesOfInterest[4:5]) {

  if(length(grep(GENE, countData$gene)) < 1){
    next;
  }

  geneData <- subset(countData, grepl(GENE, countData$gene))

  sumGraph <- ddply(geneData, c("type", "genotype"), summarise,
    N      = length(count),
    mean   = mean(count),
    sd     = sd(count),
    se     = sd / sqrt(N) )

  #Plot 1
  plot1 <- ggplot(sumGraph, aes(type, mean, fill = genotype)) +
    geom_bar(stat = "identity", position = "dodge") +
    geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
      width=.2, # Width of the error bars,
      position=position_dodge(.9)) +
    theme(text = element_text(size=20),
      axis.text.x = element_text(angle=90, vjust=1)) +
    ggtitle(GENE)

  print(plot1)

  #Plot 2
  plot2 <- ggplot(sumGraph, aes(genotype, mean, fill = type)) +
    geom_bar(stat = "identity", position = "dodge") +
    geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
      width=.2, # Width of the error bars,
      position=position_dodge(.9)) +
    theme(text = element_text(size=20),
      axis.text.x = element_text(angle=0, vjust=1)) +
    ggtitle(GENE)

  print(plot2)
}
```

