data exploration

Marie Moriarty

2022-12-14

Import MAGeCK data set

```r
# Import MAGeCK data set
library(readr)
library(ggplot2)

mageck <- read_delim("mageckRRA.gene_summary.txt",
    delim = "\t", escape_double = FALSE,
    trim_ws = TRUE)
```

```
## Rows: 19672 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (1): id
## dbl (13): num, neg|score, neg|p-value, neg|fdr, neg|rank, neg|goodsgrna, neg...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

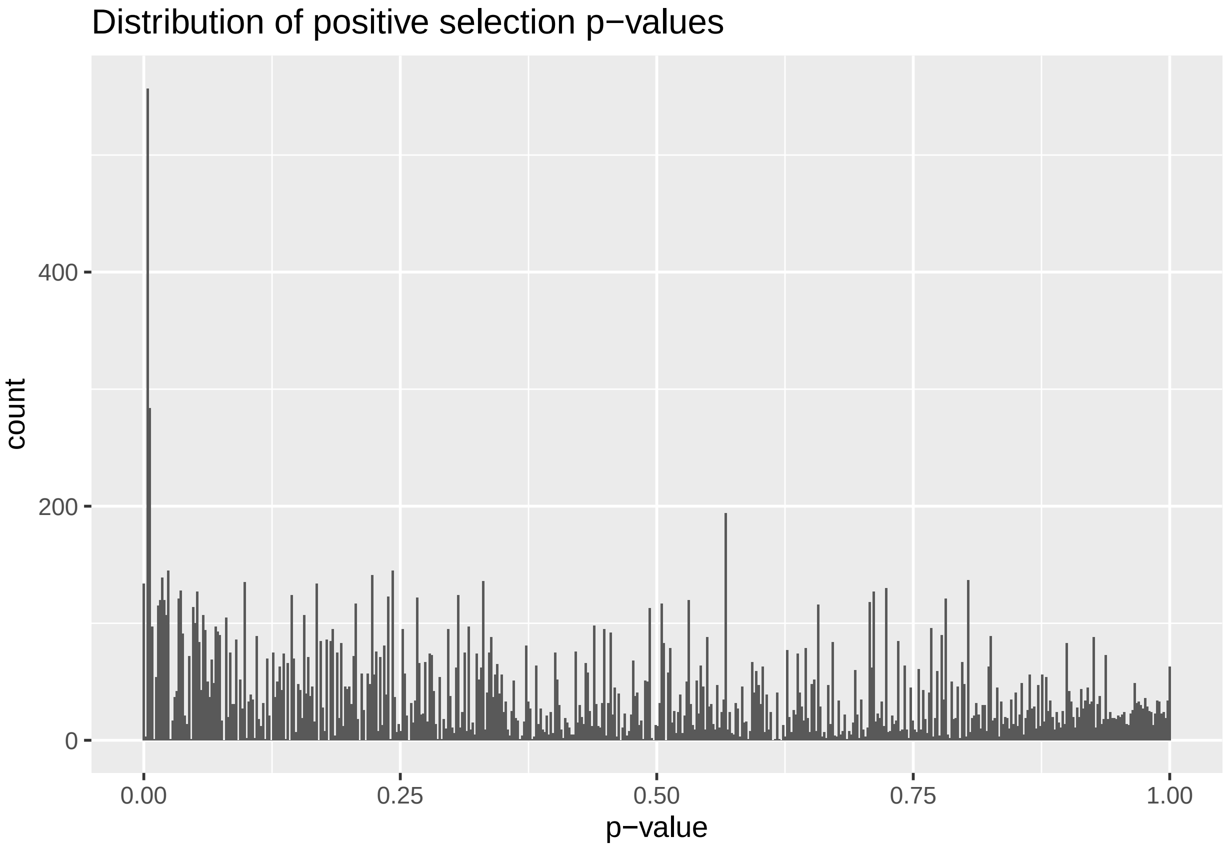
```r
# Reassign column names
colnames(mageck) <- c("id",
                      "num",
                      "neg.score",
                      "neg.p_value",
                      "neg.fdr",
                      "neg.rank",
                      "neg.goodsgrna",
                      "neg.lfc",
                      "pos.score",
                      "pos.p_value",
                      "pos.fdr",
                      "pos.rank",
                      "pos.goodsgrna",
                      "pos.lfc"
                      )

# Convert goodsgrna to factor
mageck$num <- as.factor(mageck$num)
mageck$neg.goodsgrna <- as.factor(mageck$neg.goodsgrna)
mageck$pos.goodsgrna <- as.factor(mageck$pos.goodsgrna)
```

```
# view data summary
summary(mageck)
```

```
##       id            num         neg.score       neg.p_value
##  Length:19672       1:   11   Min.   :0.0000   Min.   :0.0000002
##  Class :character   2:   10   1st Qu.:0.1466   1st Qu.:0.3637900
##  Mode  :character   3:  159   Median :0.4151   Median :0.6424600
##                     4:19490   Mean   :0.4624   Mean   :0.6012542
##                     8:    2   3rd Qu.:0.7843   3rd Qu.:0.8687800
##                               Max.   :1.0000   Max.   :1.0000000
##     neg.fdr            neg.rank     neg.goodsgrna    neg.lfc
##  Min.   :0.000381   Min.   :    1   0:5337        Min.   :-1.781400
##  1st Qu.:1.000000   1st Qu.: 4919   1:7395        1st Qu.:-0.151500
##  Median :1.000000   Median : 9836   2:4928        Median :-0.004924
##  Mean   :0.993202   Mean   : 9836   3:1693        Mean   : 0.026433
##  3rd Qu.:1.000000   3rd Qu.:14754   4: 319        3rd Qu.: 0.154968
##  Max.   :1.000000   Max.   :19672                 Max.   : 3.622500
##    pos.score       pos.p_value           pos.fdr            pos.rank
##  Min.   :0.0000   Min.   :0.0000048   Min.   :0.001763   Min.   :    1
##  1st Qu.:0.1667   1st Qu.:0.1433900   1st Qu.:0.591756   1st Qu.: 4919
##  Median :0.4388   Median :0.3493300   Median :0.731480   Median : 9836
##  Mean   :0.4731   Mean   :0.4113567   Mean   :0.714705   Mean   : 9836
##  3rd Qu.:0.7905   3rd Qu.:0.6704100   3rd Qu.:0.935922   3rd Qu.:14754
##  Max.   :1.0000   Max.   :1.0000000   Max.   :1.000000   Max.   :19672
##  pos.goodsgrna    pos.lfc
##  0:5157        Min.   :-1.781400
##  1:7318        1st Qu.:-0.151500
##  2:4732        Median :-0.004924
##  3:1810        Mean   : 0.026433
##  4: 655        3rd Qu.: 0.154968
##                Max.   : 3.622500
```

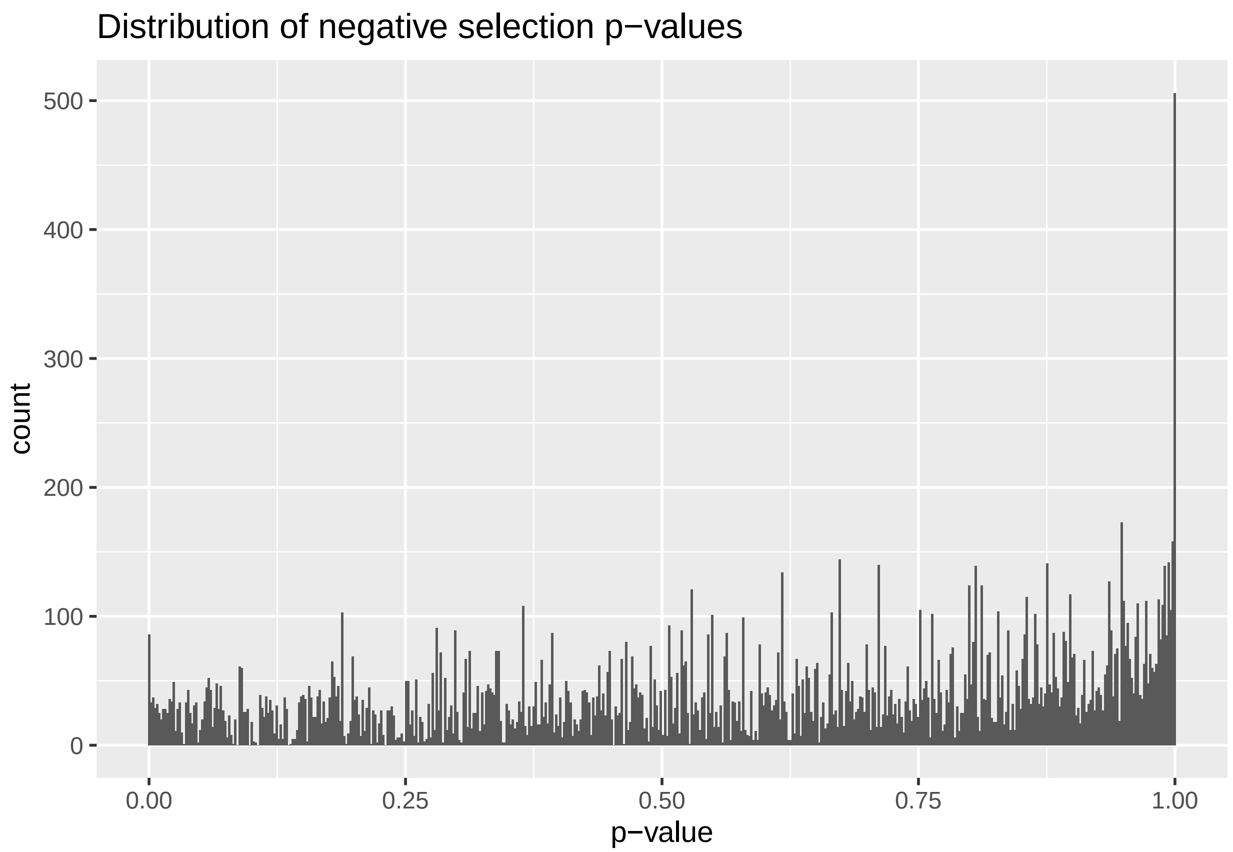
Since the data was already clean, after importing I only switched to more easily referenced variable names. I converted the sgRNA related columns to factor-type variables. I then printed the summary of the data set. One thing that I noticed was that there is a factor level in the `num` column saying that there were two observations that had 8 sgRNAs. These seem to be outliers, so they may need to be excluded before beginning the analysis, but I will consult Dr. Ge beforehand.

```
#### p-values

# Distribution of positive selection p-values
ggplot(mageck, aes(x = pos.p_value)) +
  geom_histogram(bins = 500) +
  xlab("p-value") +
  ggtitle("Distribution of positive selection p-values")
```

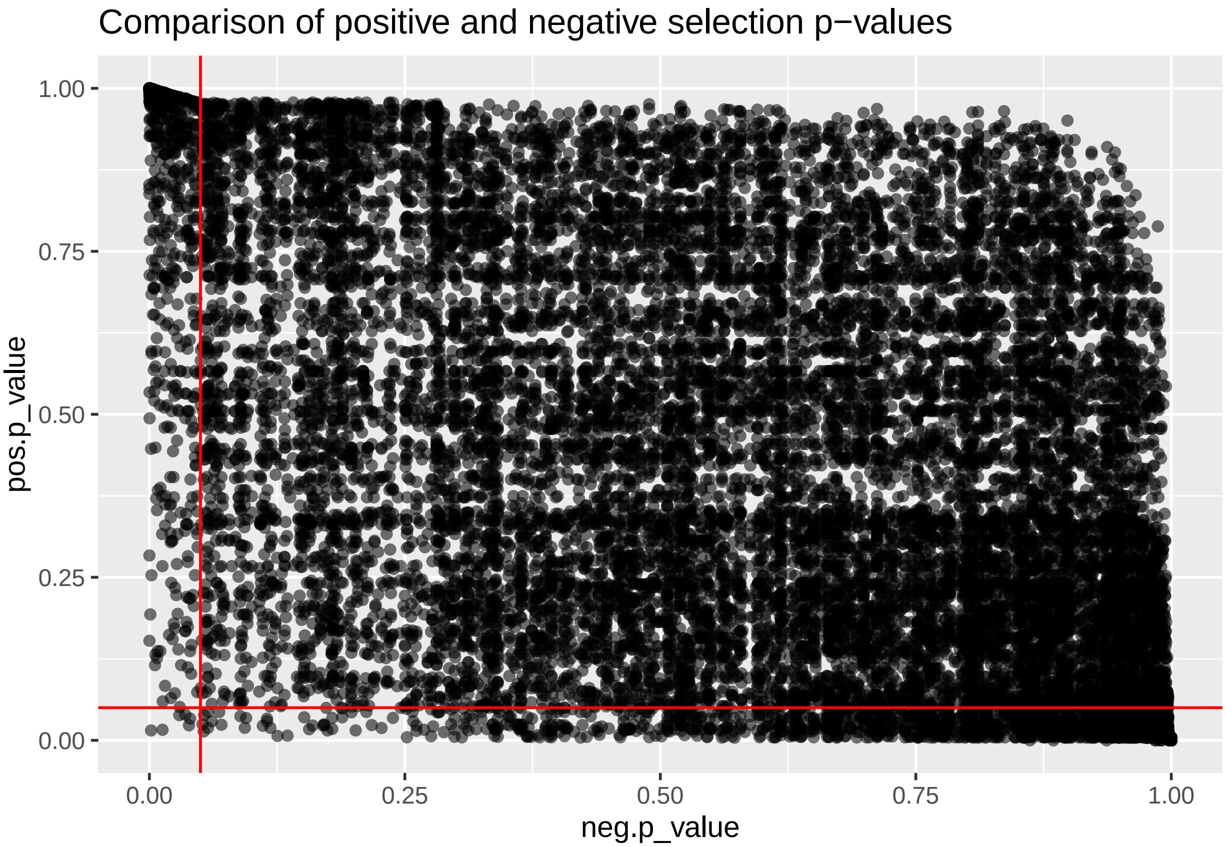


Initial plots

```
# Distribution of negative selection p-values
ggplot(mageck, aes(x = neg.p_value)) +
  geom_histogram(bins = 500) +
  xlab("p-value") +
  ggtitle("Distribution of negative selection p-values")
```

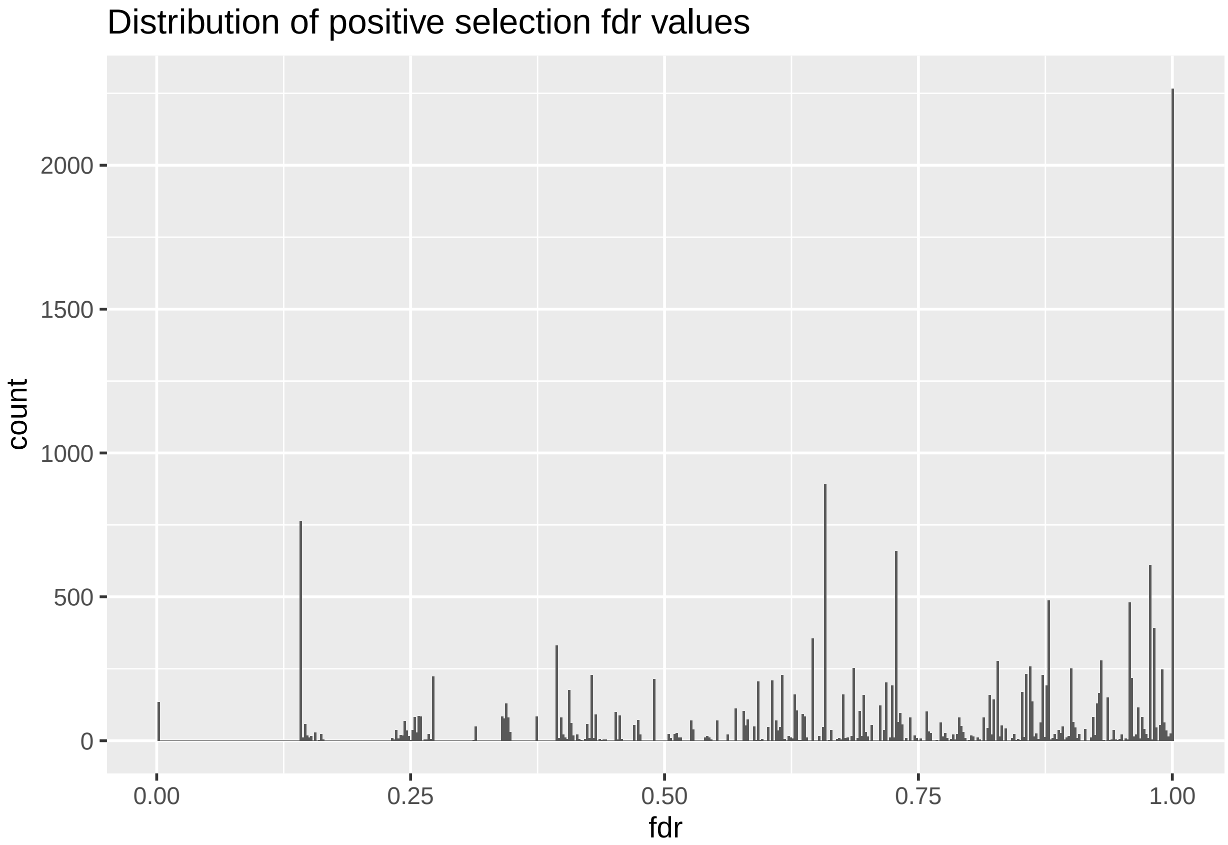


```
# Compare positive and negative p-values
ggplot(mageck, aes(x = neg.p_value,
                   y = pos.p_value,
                   alpha = 0.01)) +
  geom_point() +
  geom_vline(xintercept = 0.05,
             color = "red") +
  geom_hline(yintercept = 0.05,
             color = "red") +
  ggtitle("Comparison of positive and negative selection p-values") +
  theme(legend.position="none")
```

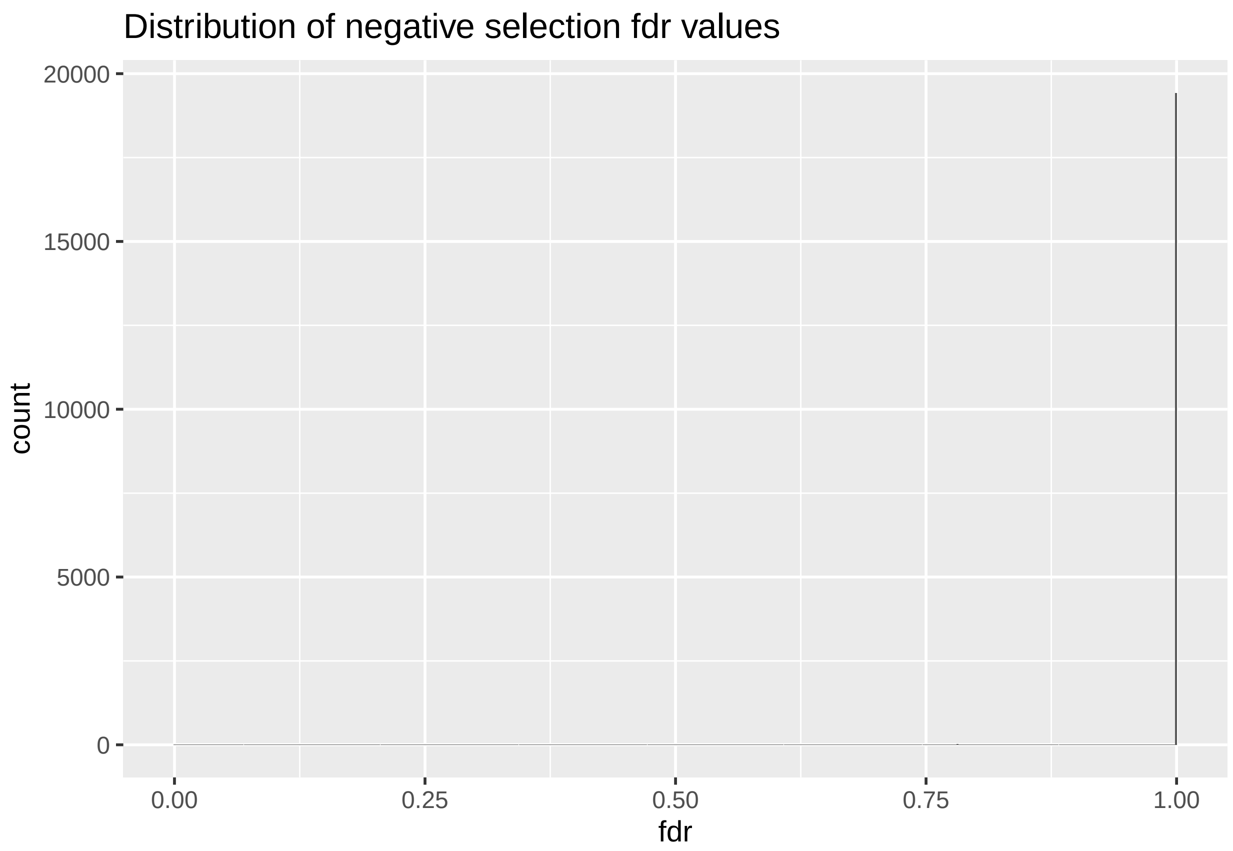


Viewing the histogram of positive selection p-values, the data does not seem to show any obvious patterns, except for some higher frequencies as the values approach zero. For the negative selection p-values, we see more observations with p-values at or close to one. The scatter plot, showing both variables along with red lines marking a significance level of 0.05, shows far more significant p-values for the positive selection than the negative, with a large cluster of data points with both very low positive p-values and very high negative p-values.

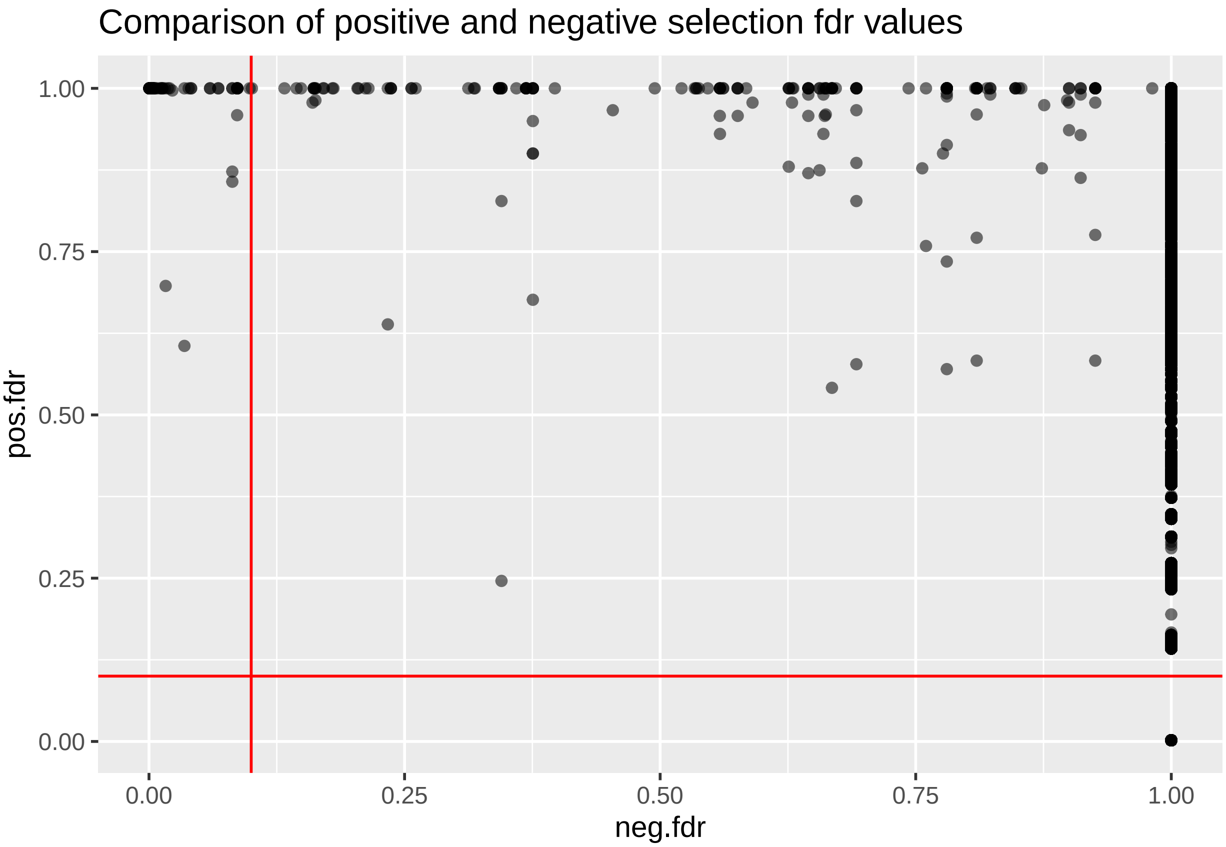
```
####  False Discovery Rates

# Distribution of positive selection fdr
ggplot(mageck, aes(x = pos.fdr)) +
  geom_histogram(bins = 500) +
  xlab("fdr") +
  ggtitle("Distribution of positive selection fdr values")
```



```
# Distribution of negative selection p-values
ggplot(mageck, aes(x = neg.fdr)) +
  geom_histogram(bins = 500) +
  xlab("fdr") +
  ggtitle("Distribution of negative selection fdr values")
```

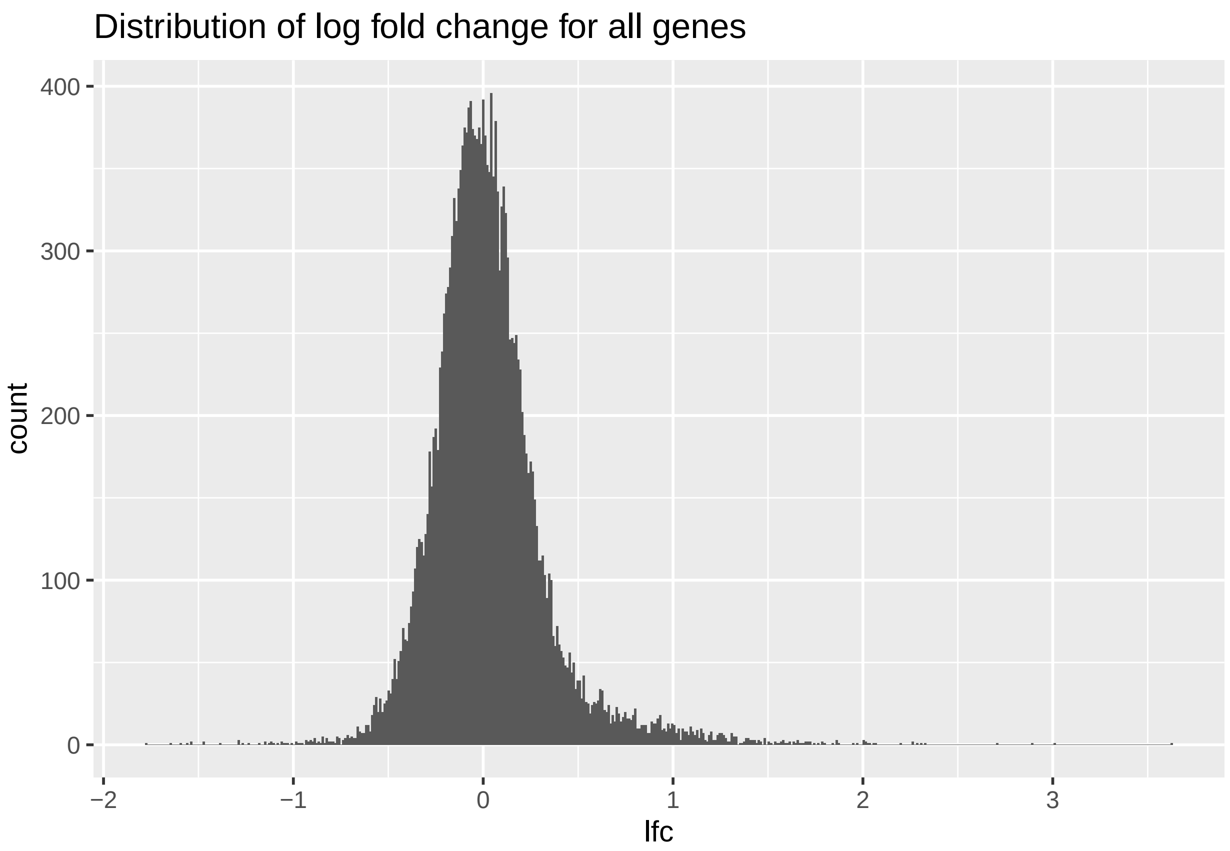


```
# Compare positive and negative fdr's
ggplot(mageck, aes(x = neg.fdr,
                   y = pos.fdr,
                   alpha = 0.01)) +
  geom_point() +
  geom_vline(xintercept = 0.1,
             color = "red") +
  geom_hline(yintercept = 0.1,
             color = "red") +
  ggtitle("Comparison of positive and negative selection fdr values") +
  theme(legend.position="none")
```

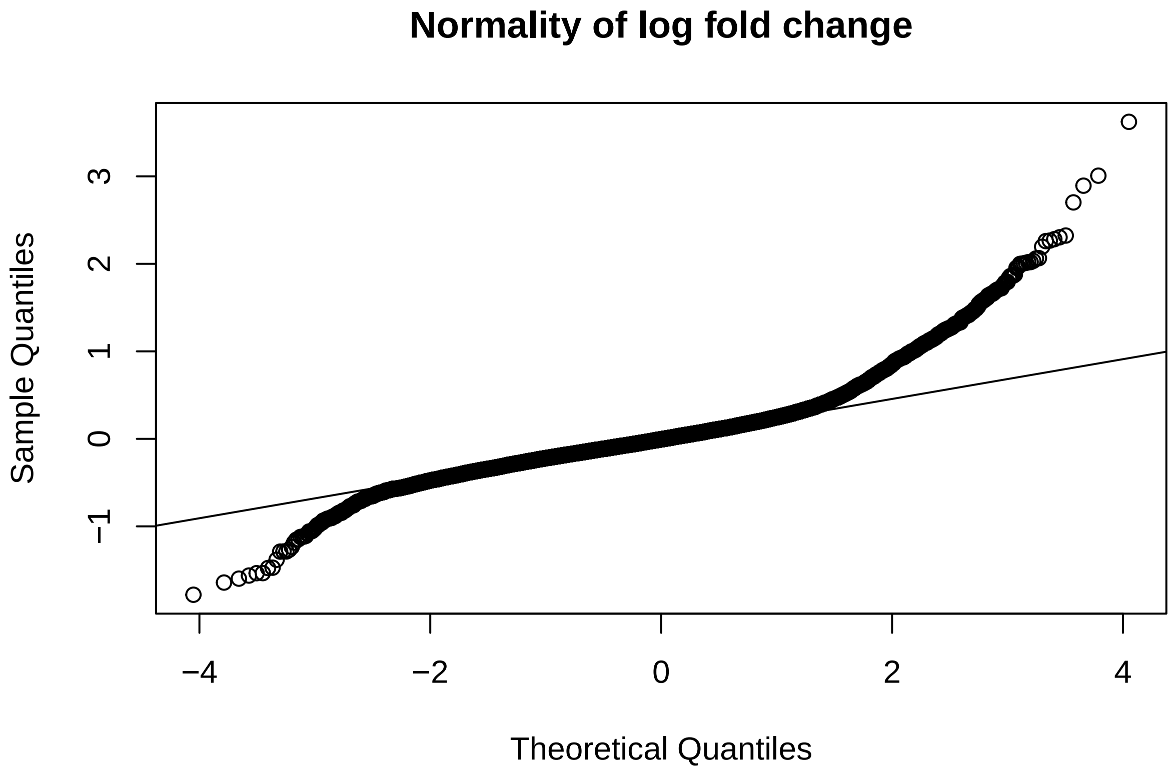


Looking at the positive and negative selection false discovery rate values, we see the majority of the positive values are closer to one, and surprisingly all but a handful of values for the negative selection are equal to one. We see this more clearly in the scatter plot, which includes significance lines at FDR = 0.1.

```
#### log fold change

# Distribution of lfc (positive and negative values the same)
ggplot(mageck, aes(x = pos.lfc)) +
  geom_histogram(bins = 500) +
  xlab("lfc") +
  ggtitle("Distribution of log fold change for all genes")
```

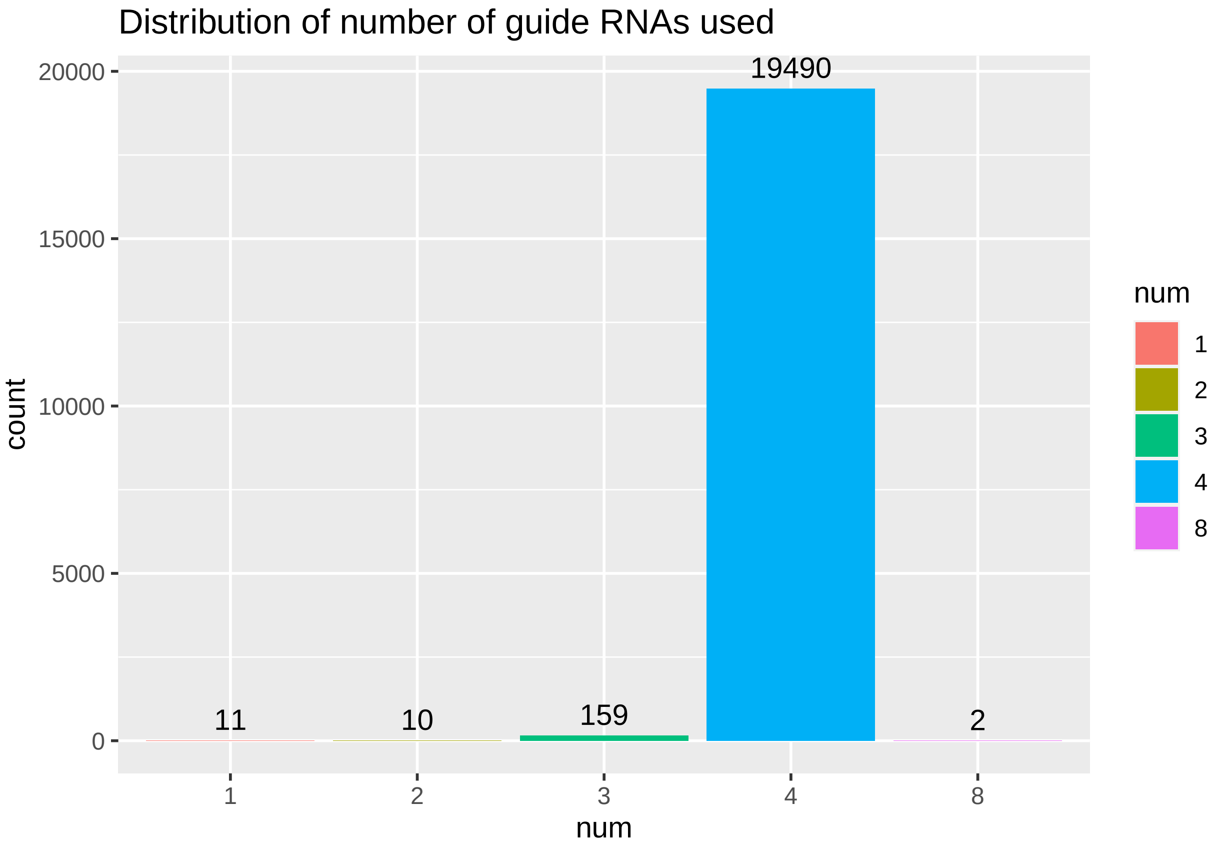


```r
# Check normality of variable
qqnorm(mageck$pos.lfc, main = "Normality of log fold change")
qqline(mageck$pos.lfc)
```

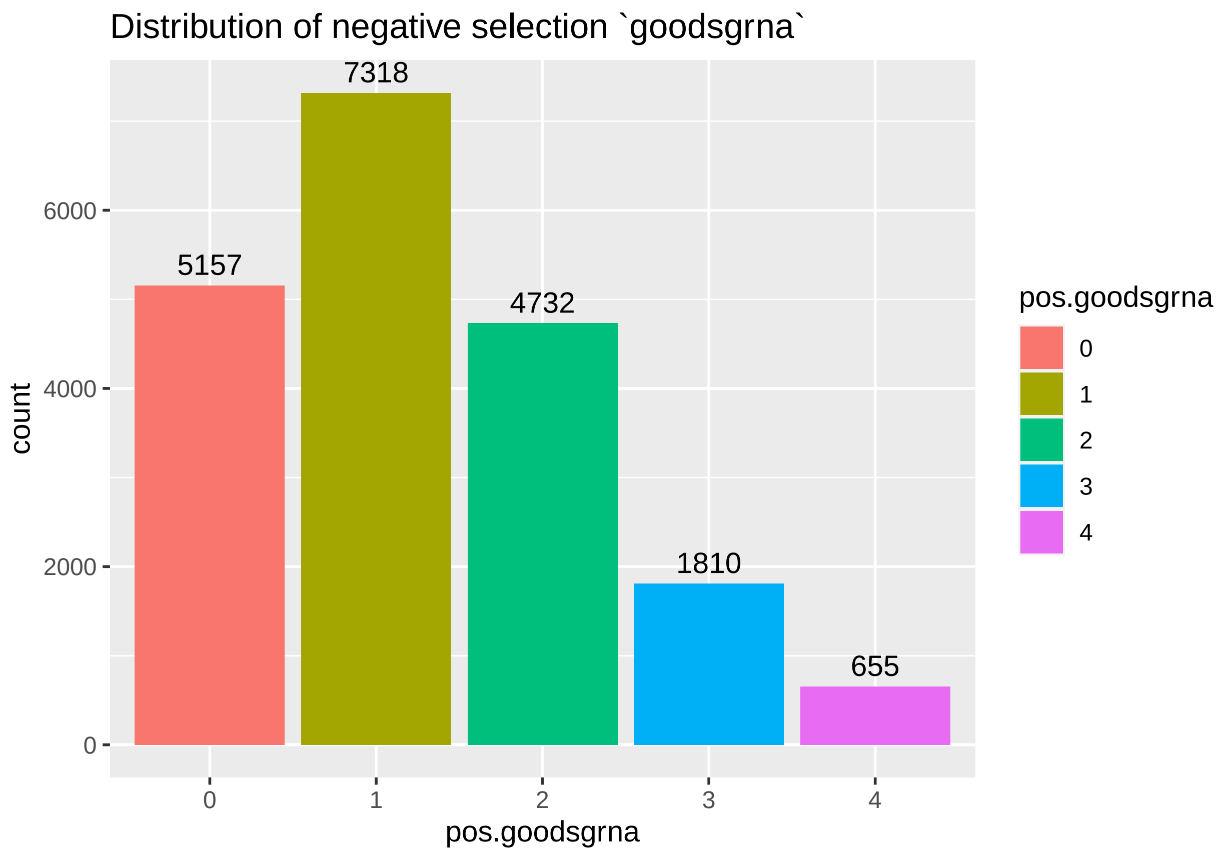


The log fold change variable (same for both positive and negative selections) appears to follow an approximately normal distribution, judging by the histogram. We validated this result using the Q-Q plot in the next figure.

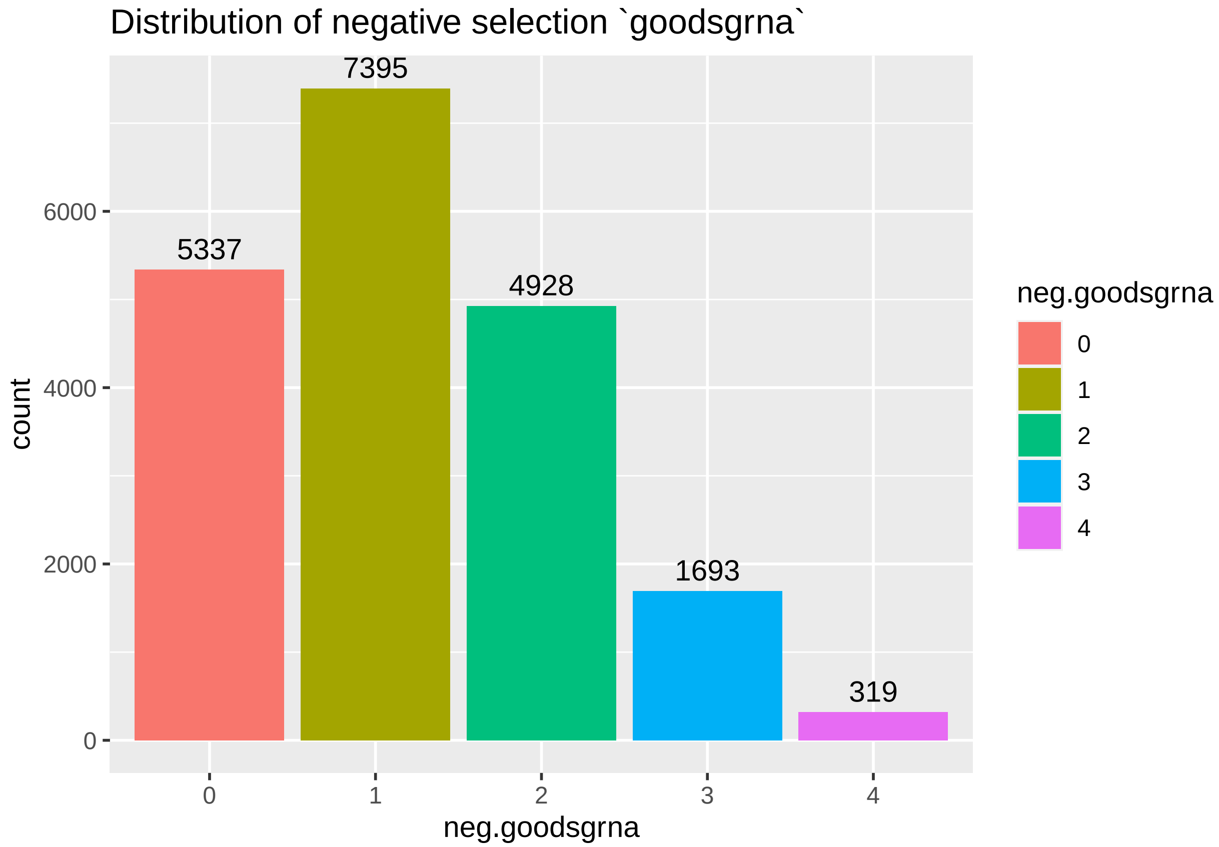
```
### number of sgRNA's

ggplot(mageck, aes(num)) +
  geom_bar(aes(fill = num)) +
  geom_text(stat = 'count',
            aes(label = after_stat(count)),
            vjust = -0.5) +
  ggtitle("Distribution of number of guide RNAs used")
```



```
### goodsgrna

# positive selection
ggplot(mageck, aes(pos.goodsgrna)) +
  geom_bar(aes(fill = pos.goodsgrna)) +
  geom_text(stat = 'count',
            aes(label = after_stat(count)),
            vjust = -0.5) +
  ggtitle("Distribution of negative selection `goodsgrna`")
```



```r
# negative selection
ggplot(mageck, aes(neg.goodsgrna)) +
  geom_bar(aes(fill = neg.goodsgrna)) +
  geom_text(stat = 'count',
            aes(label = after_stat(count)),
            vjust = -0.5) +
  ggtitle("Distribution of negative selection `goodsgrna`")
```



I created bar plots showing the `num` column (the number of targeting sgRNAs for each gene) and the positive and negative selection `goodsgrna` columns (the number of "good" sgRNAs, i.e. those whose ranking fell below a set FDR cutoff). We see that for almost every gene four sgRNA's were used (perhaps the rest could even be considered outliers). The number of "good" sgRNA's followed similar distributions for both positive and negative, with the majority of genes having 1.

```
# Code adapted from https://stackoverflow.com/questions/6602881/text-file-to-list-in-r

# Read in pathways data as list and split elements into strings
gmt <- scan("m2.cp.v2022.1.Mm.symbols.gmt", what = "", sep = "\n")
pathways <- strsplit(gmt, "[[:space:]]+")

# Assign first entry to names of each list element
names(pathways) <- sapply(pathways, `[[`, 1)

# save urls to separate list for reference
source <- sapply(pathways, `[[`, 2)

# Remove two beginning reference rows
pathways <- lapply(pathways, `[`, -c(1:2))

# Preview data set
head(pathways)
```

Create list of all gene sets

```
## $BIOCARTA_RELA_PATHWAY
##  [1] "Tnfrsf1a" "Tnf"      "Chuk"     "Fadd"     "Ikbkg"    "Crebbp"
##  [7] "Nfkbia"   "Tnfrsf1b" "Rela"     "Nfkb1"    "Ep300"    "Tradd"
## [13] "Ikbkb"    "Traf6"    "Ripk1"
##
## $BIOCARTA_CSK_PATHWAY
##  [1] "Cd4"     "Cd3d"    "Zap70"   "Prkacb"  "Csk"     "Prkar2b" "Prkar1a"
##  [8] "Crebbp"  "Cd3e"    "Cd247"   "Prkar2a" "Adcy1"   "Lck"     "Prkar1b"
## [15] "Cd3g"    "Ptprc"
##
## $BIOCARTA_SRCRPTP_PATHWAY
##  [1] "Csk"    "Cdc25b" "Prkcb"  "Prkca"  "Ptpra"  "Ccnb1"  "Cdk1"   "Cdc25c"
##  [9] "Cdc25a" "Grb2"
##
## $BIOCARTA_ARAP_PATHWAY
##  [1] "Arfgap1" "Cyth1"   "Arfgap3" "Gbf1"    "Cyth2"   "Asap1"   "Arap1"
##  [8] "Cyth3"   "Gpld1"   "Clta"    "Chmp4c"  "Arf1"
##
## $BIOCARTA_AGR_PATHWAY
##  [1] "Cdc42"   "Rapsn"   "Dvl1"    "Chrna1"  "Sp1"     "Dag1"    "Mapk3"
##  [8] "Egfr"    "Musk"    "Mapk8"   "Pak4"    "Pak3"    "Lama3"   "Git2"
## [15] "Mapk1"   "Cttn"    "Acta1"   "Pak2"    "Chrm1"   "Lama2"   "Lama4"
## [22] "Nrg3"    "Pak1"    "Arhgef6" "Itgb1"   "Agrn"    "Jun"     "Dmd"
## [29] "Lama1"   "Itga1"   "Utrn"
##
## $BIOCARTA_AKAP95_PATHWAY
##  [1] "Prkag1"  "Prkacb"  "Prkar2b" "Ddx5"    "Prkar2a" "Ncapd2"  "Ccnb1"
##  [8] "Ppp2ca"  "Cdk1"    "Akap8"
```

This code chunk imports a file containing a collection of mouse gene sets which will be used for the GSEA analysis. The data needed to be reformatted so that we could separate the gene set names and source urls from the actual list of genes for each set. Finally we ended up with a list object where each element contains a gene set, with a list of all the genes in that set in order.