MMID_workshop_Feb23

MacKenzie Wilke

02/02/2022

Load in the libraries

install.packages () first before loading them into the library

```
library(tinytex)#install.packages("tinytex") #To make the rmarkdown PDF file
library(dplyr) #install.packages("dyplr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readxl) #install.packages("readxl")
library(snakecase) #install.packages("snakecase")
library(ggplot2) #install.packages("ggplot2")
library(cowplot) #install.packages("cowplot")
library(plotly)#install.packages("plotly")
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

Step 1. Read in Excel File

CLUE output via Josset et al. supplementary table 5

```
clue_results<-read_excel("/Users/msarvis/Desktop/MMID/mmid_workshop.xls", sheet=1) %>%
  rename_with(to_snake_case)

#See what it looks like
glimpse(clue_results)
```

```
## Rows: 14
## Columns: 5
## $ perturbagen               <chr> "IKZF2", "GSTP1", "fluconazole", "F2R", "SRC~
## $ median_connectivity_score <dbl> 0.80, 0.79, 0.78, 0.78, 0.78, 0.77, 0.77, 0.~
## $ filter_type               <chr> "antiviral_article_results", "antiviral_arti~
## $ perturbagen_type          <chr> "Over-expression", "shRNA loss of function",~
## $ gene_expression_profile   <chr> "Similar", "Similar", "Similar", "Similar", ~
```

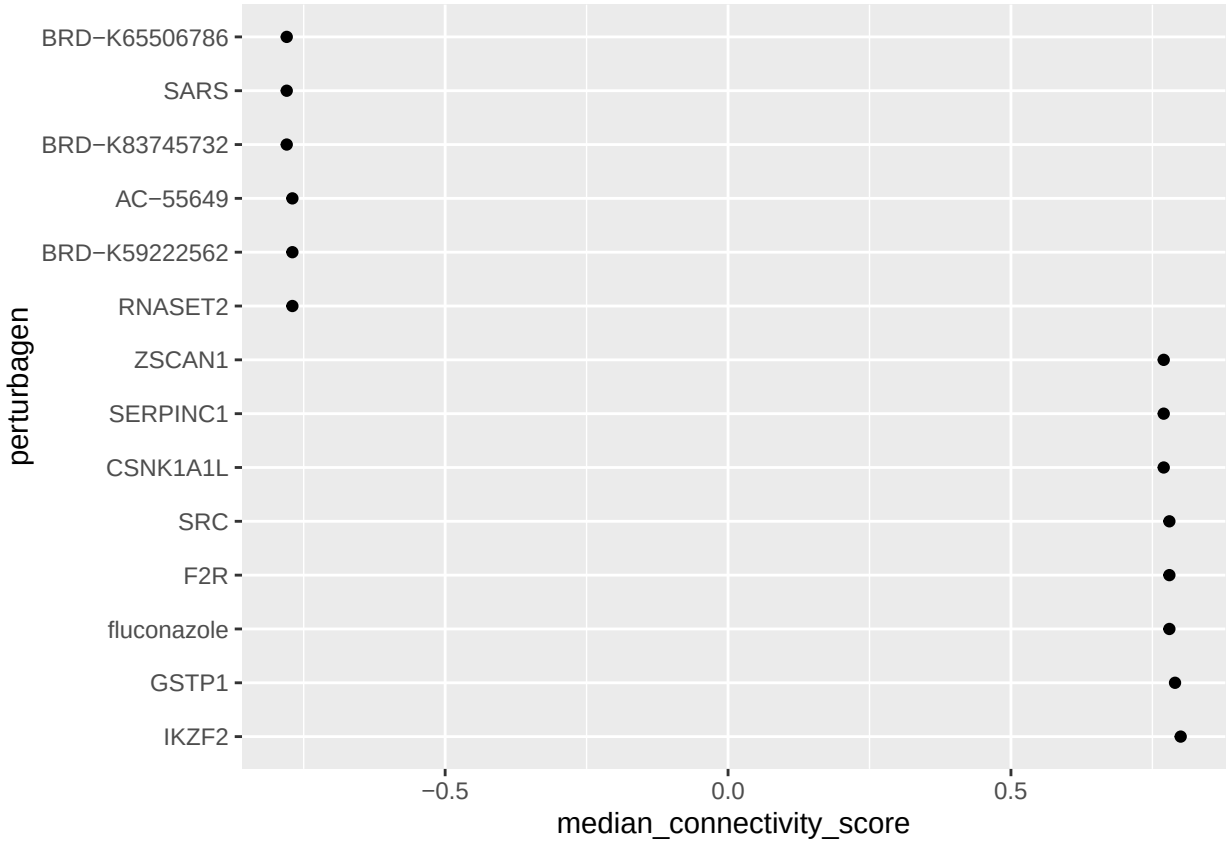
Step 2. Change the perturbagen names into a factor and also gives the factor levels

This is to sort the perturbagens in a way that isn't alphabetic

```
clue_results$perturbagen <- factor(clue_results$perturbagen, levels = clue_results$perturbagen)
```

Step 3. Create a basic plot

```
basic_plot<-ggplot(clue_results, aes(x=median_connectivity_score, y=perturbagen)) +
  geom_point()

#To view the plot
basic_plot
```


BRD−K65506786
SARS
BRD−K83745732
AC−55649
BRD−K59222562
RNASET2
ZSCAN1
SERPINC1
CSNK1A1L
SRC
F2R
fluconazole
GSTP1
IKZF2
perturbagen
−0.5
0.0
0.5
median_connectivity_score

Step 4. Change the colour of the perturbagens based on their connectivity score

- Blue is when the score is less than 0 and red is when it is greater than 0
- Saves this as a value

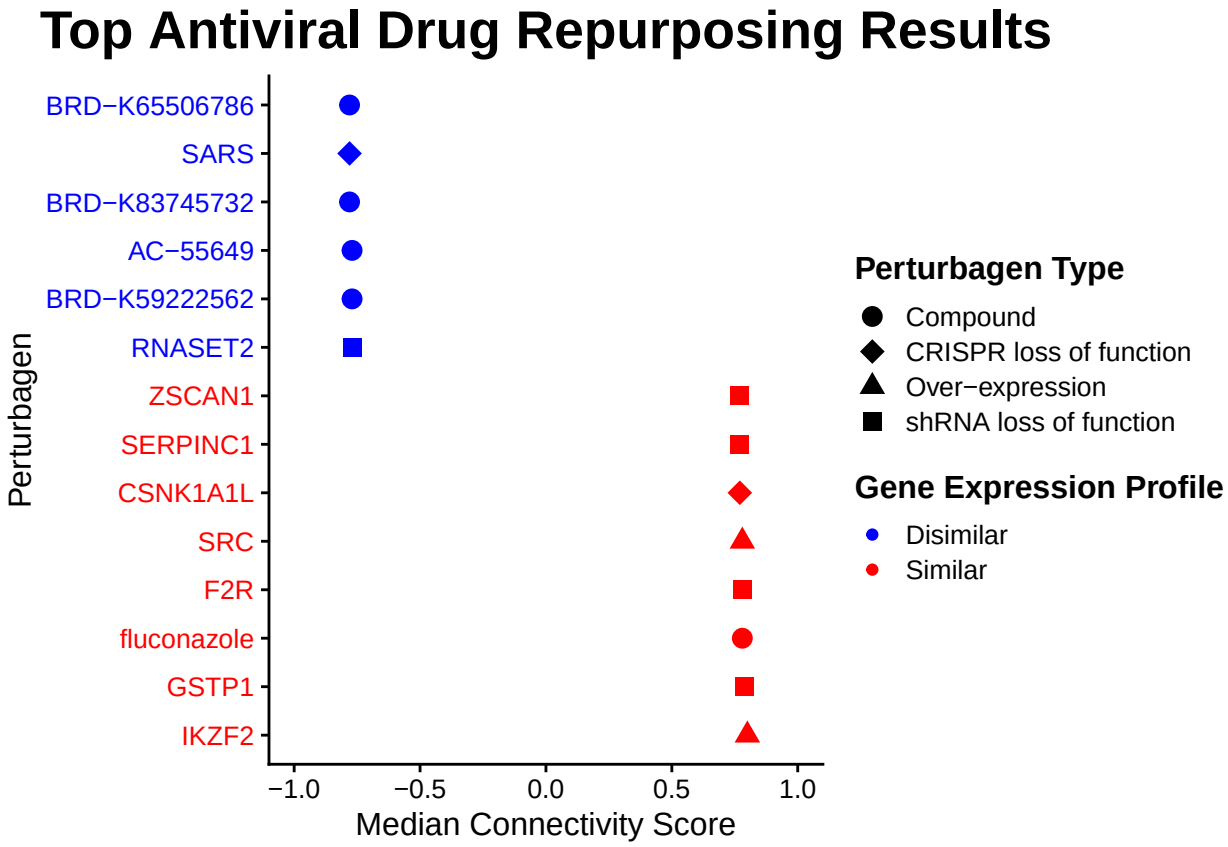
```
colours <- ifelse(clue_results$median_connectivity_score < 0, "blue", "red")
```

Step 5. Adding in extra details to our plot (i.e. colour, shape and size)

```
drug_repurposing_plot<-basic_plot + geom_point(aes(colour=gene_expression_profile, shape= perturbagen_ty
  theme_cowplot(12) + ggtitle("Top Antiviral Drug Repurposing Results")+
    xlab("Median Connectivity Score") +ylab ("Perturbagen") + xlim(-1,1) +
      theme(plot.title = element_text(size=20, hjust = 0.5), axis.text.y = element_text(hjust = 1, colou
            legend.title = element_text(face = "bold")) +
               scale_shape_manual(values= c(19,18,17,15))+
                  scale_color_manual(values = c("Disimilar" = "blue", "Similar" = "red")) +
                      scale_size_manual(values=c(3,4,3,3)) + guides(shape=guide_legend("Perturbagen Type
                        guides(size=guide_legend("Perturbagen Type")) +
                         guides(color=guide_legend("Gene Expression Profile"))
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
#To view the plot
drug_repurposing_plot
```

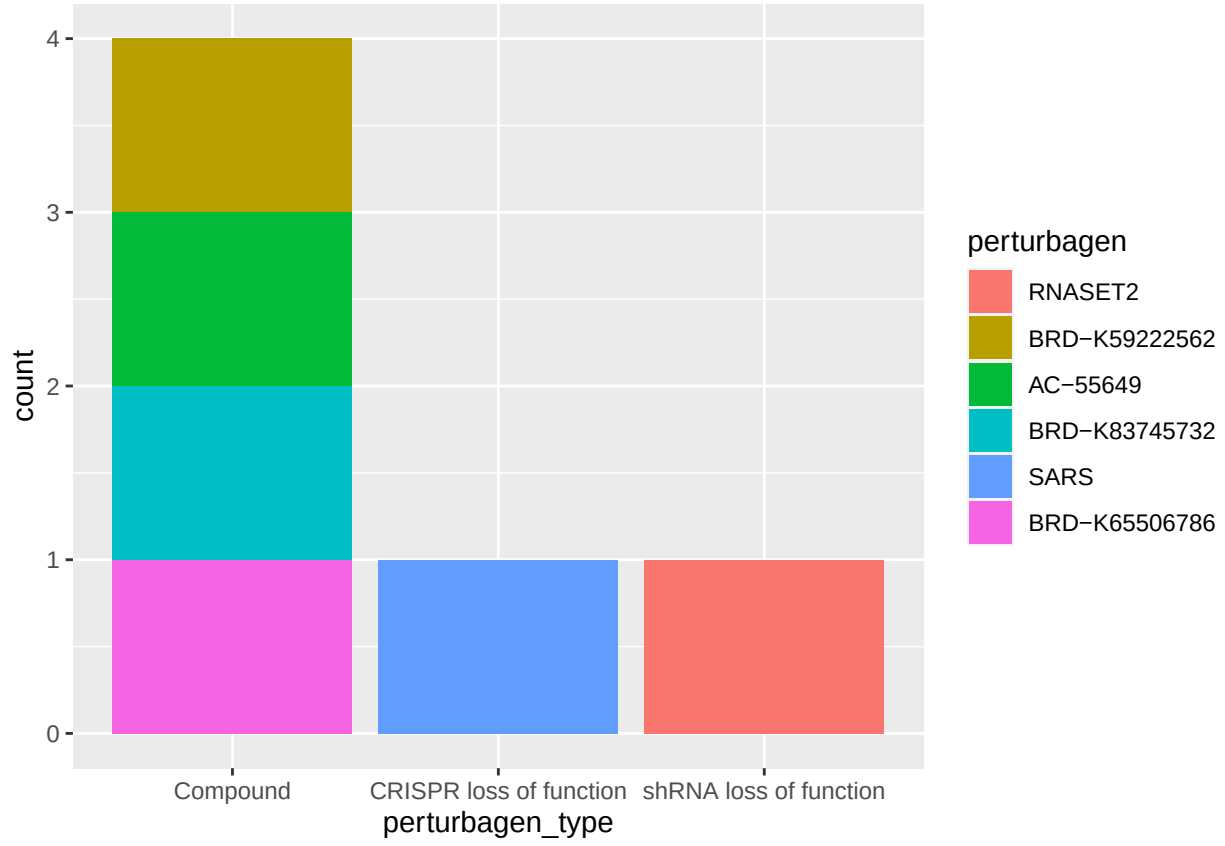



```
#Bonus:View the plot interactively, not included in RMarkdown file output
#ggplotly(drug_repurposing_plot)
```

Step 6. Add in another plot so that we can use the cow_plot package to combine them

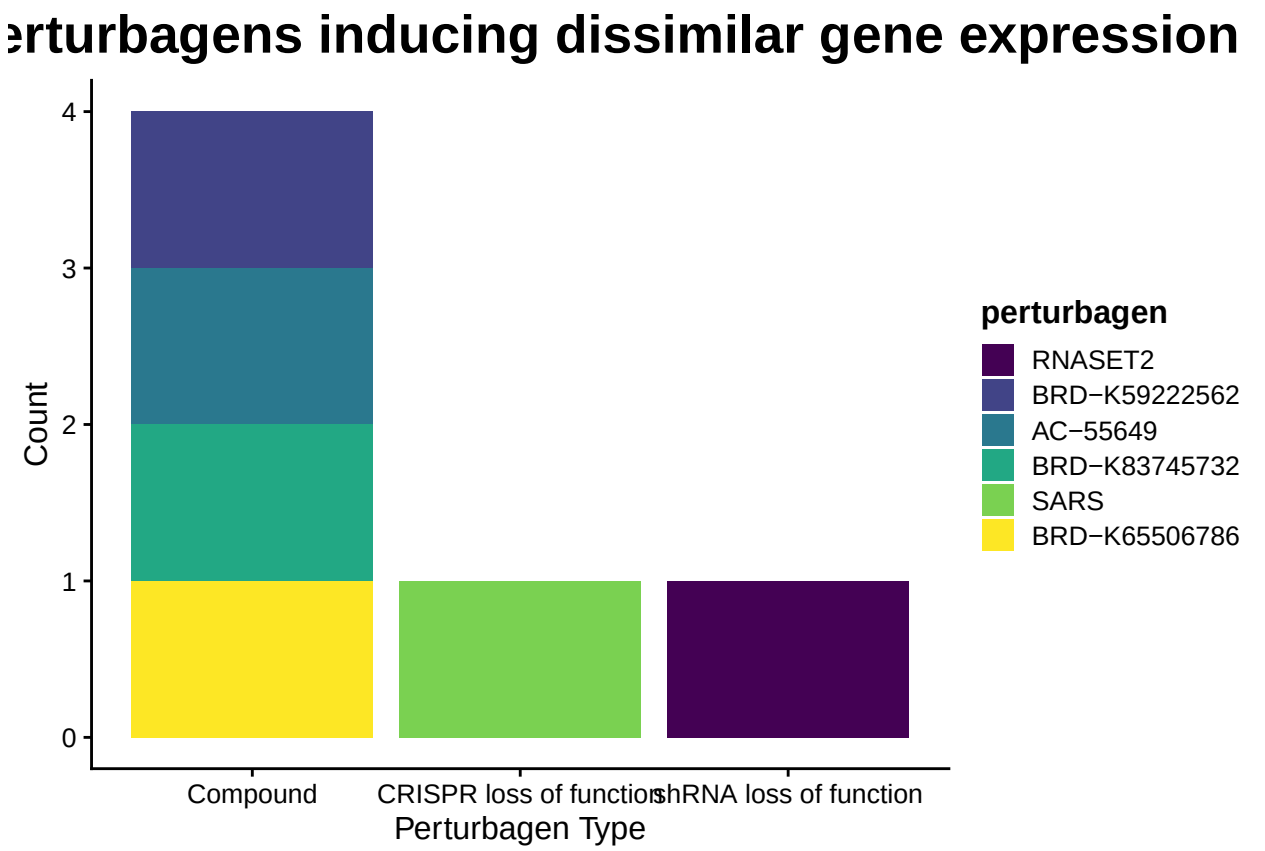
We are going to make a bar plot comparing the gene expression profiles and their respective perturbagens

```
#6a. Filter results to perturbagens inducing dissimilar (and similar) gene expression profiles (i.e. bl
#Can also filter during the ggplot step if you want with a pipe

disimilar_perturbagens<-clue_results %>% filter(median_connectivity_score<0)
similar_perturbagens<-clue_results %>% filter(median_connectivity_score>0)

#6b. Make a basic bar graph
basic_bar_graph<-ggplot(disimilar_perturbagens, aes(x=perturbagen_type, fill=perturbagen)) +
  geom_bar()

basic_bar_graph
```

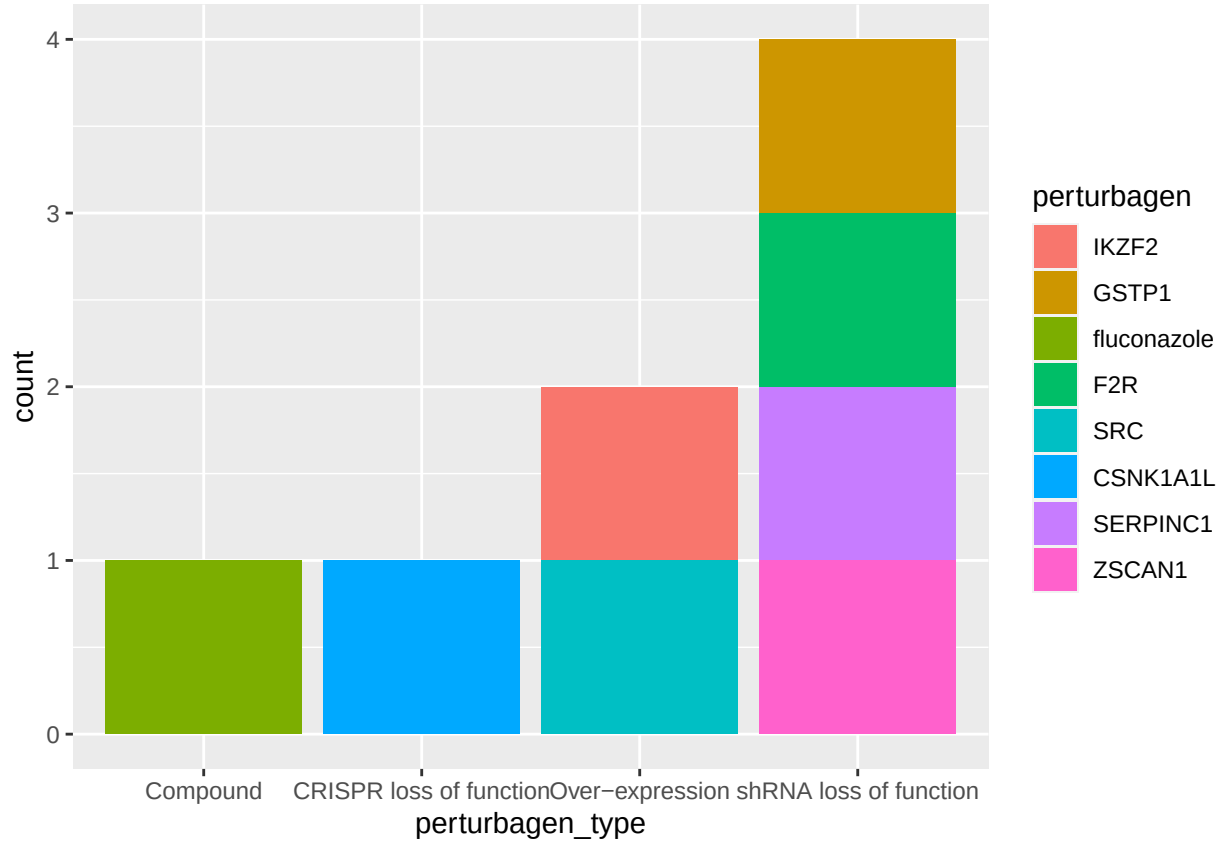



```
#6c. Add in different elements you want to have
drug_repurposing_bar_plot_dissimilar<-basic_bar_graph + geom_bar() + theme_cowplot(12) +
                                ggtitle("Breakdown of perturbagens inducing dissimilar gene expression pr
                                    xlab("Perturbagen Type") + ylab ("Count") +
                                      theme (plot.title = element_text(size=20,hjust = 0.5),
```

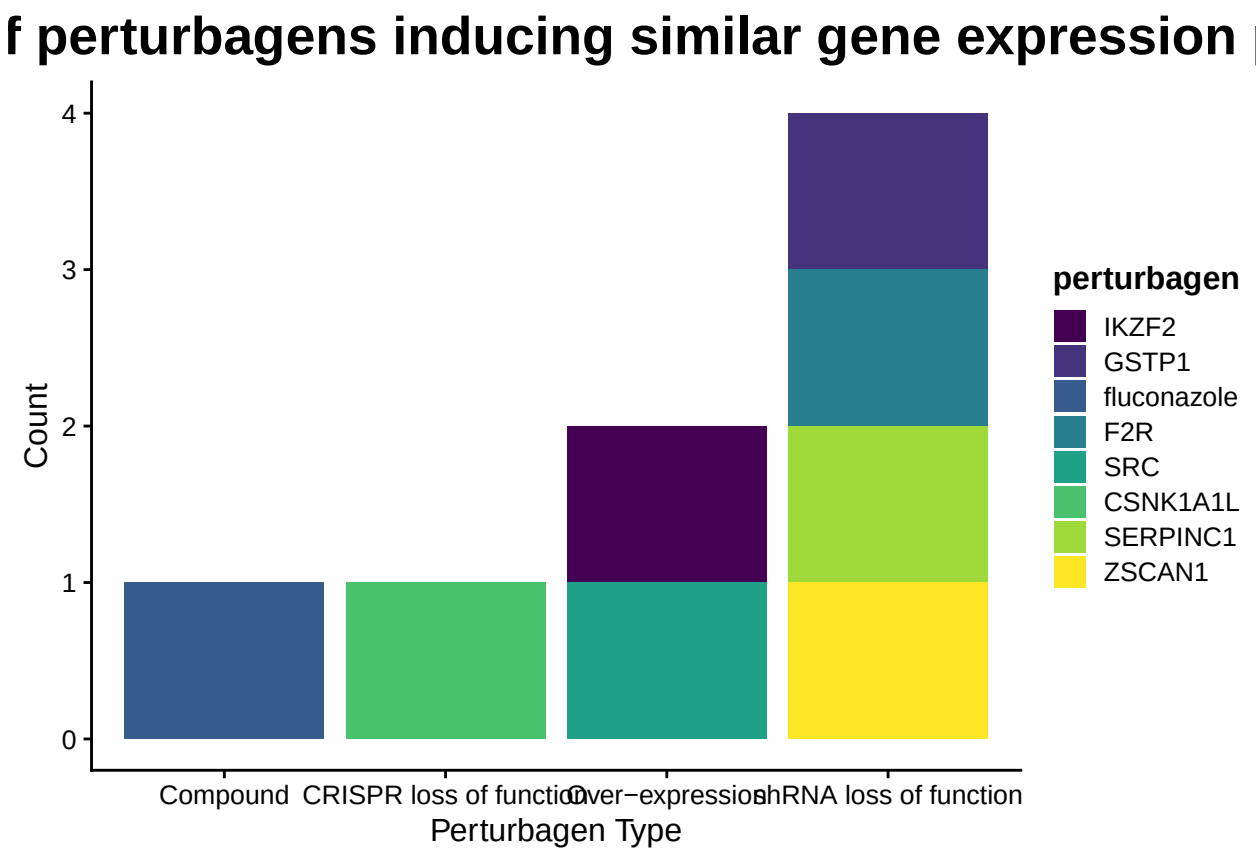
```
                                        legend.title = element_text(face = "bold")) +
                                            scale_fill_viridis_d()

drug_repurposing_bar_plot_dissimilar
```




```
#Bonus: See it interactively
#ggplotly(drug_repurposing_bar_plot_dissimilar)

##Now repeat steps 6a-c for the similar expression perturbagens

basic_bar_graph_similar<-ggplot(similar_perturbagens, aes(x=perturbagen_type, fill=perturbagen)) +
  geom_bar()

basic_bar_graph_similar
```




```
drug_repurposing_bar_plot_similar<-basic_bar_graph_similar + geom_bar() + theme_cowplot(12) +
  ggtitle("Breakdown of perturbagens inducing similar gene expression profiles") +
  xlab("Perturbagen Type") + ylab ("Count") +
  theme (plot.title = element_text(size=20,hjust = 0.5), legend.title = element_text(face = "bold")) +
  scale_fill_viridis_d()

drug_repurposing_bar_plot_similar
```

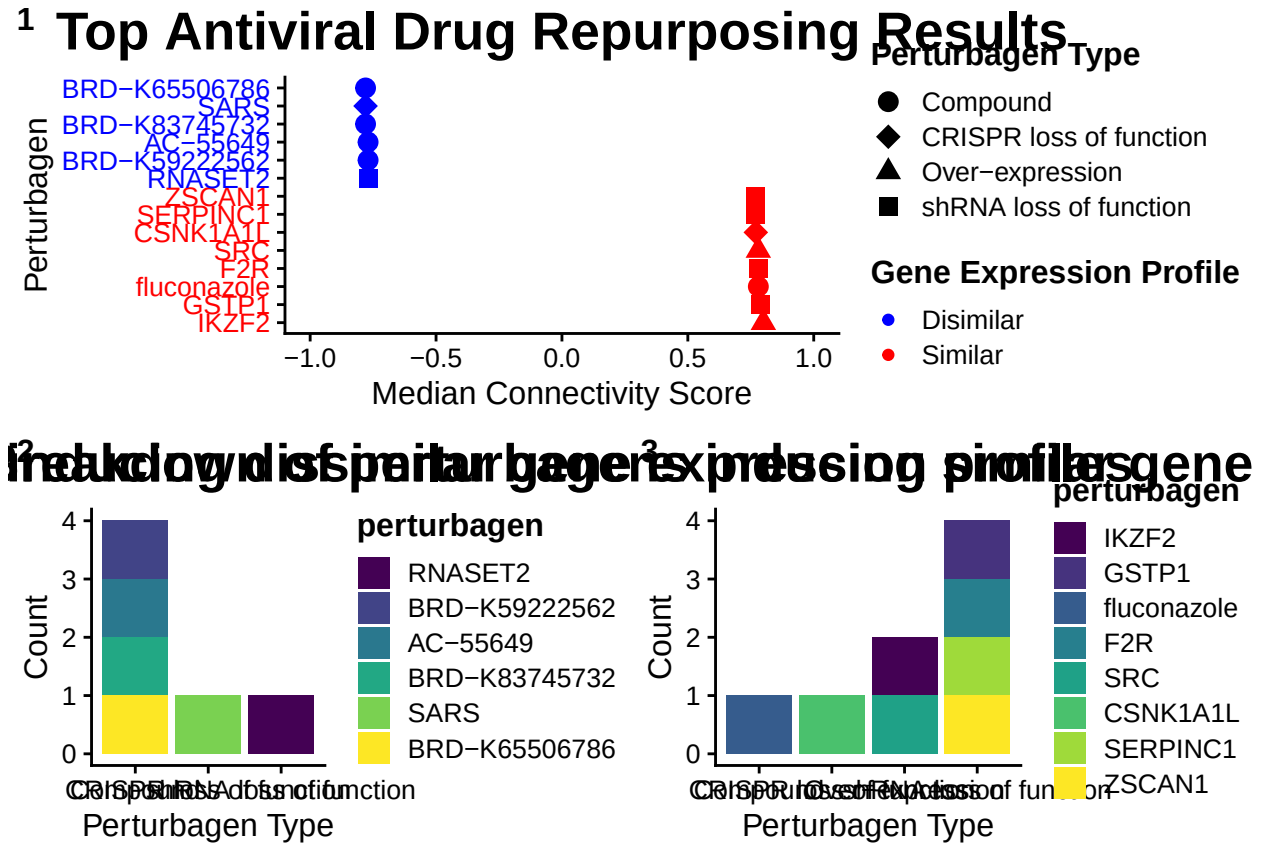



```
#ggplotly(drug_repurposing_bar_plot_similar)
```

Step 7. Use cow_plot (via plot_grid) to add the two plots together and produce a pdf image to save on your desktop

- Can also do ncol=1 to have them on the same column instead of row
- Other types of files you can save like .png, .tiff
- Can also change the dpi based on if it's for a poster, online presentation, etc

```
#We are creating the bottom row that we want to have, formatting how we want the layout to look
#We want the two bar charts to be on the bottom and the scatterplot on the top

bottom_row <- plot_grid(drug_repurposing_bar_plot_dissimilar, drug_repurposing_bar_plot_similar,
                        labels = c('2', '3'), label_size = 12)

plot_grid(drug_repurposing_plot, bottom_row,
          labels = c('1', ''), label_size=12, ncol = 1)
```




```
ggsave("/Users/msarvis/Desktop/MMID/drug_repurposing.pdf", height=15, width=25, units='in', dpi=300)
```

The End!