Mageck

Dennis

2024-04-04

1) INSTALL AND LOAD THE LIBRARIES

```
##

##

## ########################################################################################
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications
## or products. For details, do citation("pathview") within R.
##
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## ########################################################################################

## Warning: package 'clusterProfiler' was built under R version 4.4.2

## clusterProfiler v4.14.6 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## G Yu. Thirteen years of clusterProfiler. The Innovation. 2024,
## 5(6):100722

##
## Attaching package: 'clusterProfiler'

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

## Warning: package 'ggplot2' was built under R version 4.4.3
```

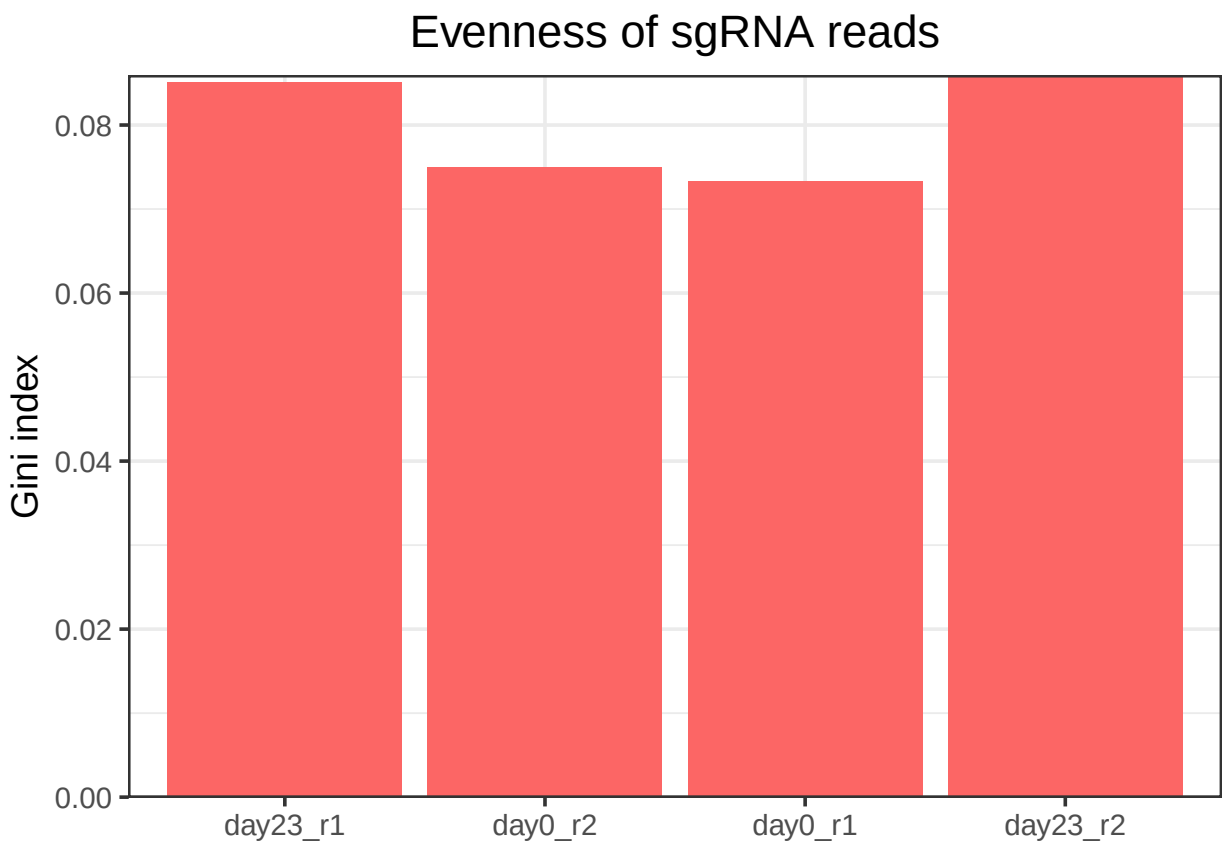
2) LOAD THE DATA FOR QC. DATA COMES FROM OPHIR SHALEM PAPER (2014) "Genome-Scale CRISPR-Cas9 Knockout Screening in Human Cells"

QAULITY CONTROL

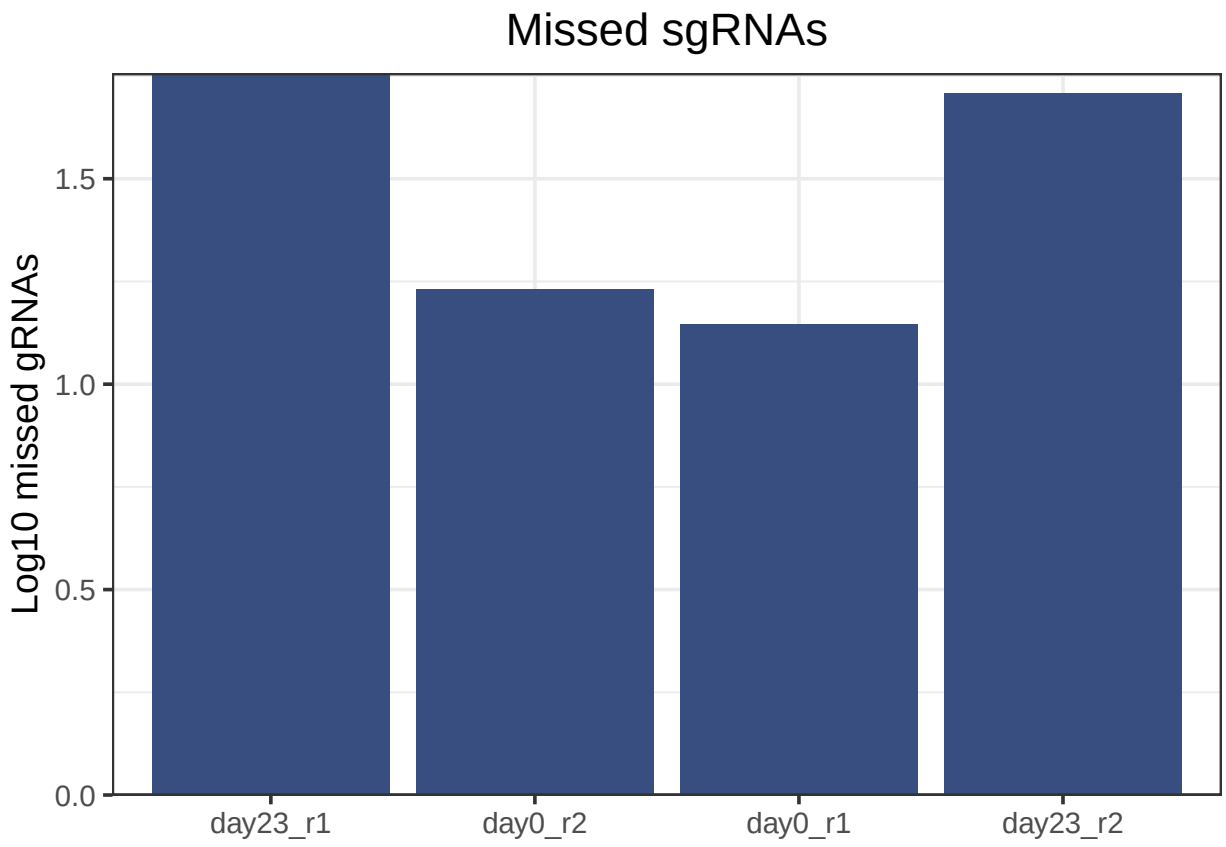
```
file4 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/countsummary.txt")
countsummary = read.delim(file4, check.names = FALSE)
head(countsummary)
```

```
##                                File    Label    Reads   Mapped Percentage
## 1 ../data/GSC_0131_Day23_Rep1.fastq.gz day23_r1 62818064 39992777     0.6366
## 2  ../data/GSC_0131_Day0_Rep2.fastq.gz  day0_r2 47289074 31709075     0.6705
## 3  ../data/GSC_0131_Day0_Rep1.fastq.gz  day0_r1 51190401 34729858     0.6784
## 4 ../data/GSC_0131_Day23_Rep2.fastq.gz day23_r2 58686580 37836392     0.6447
##   TotalsgRNAs Zerocounts GiniIndex NegSelQC NegSelQCPval
## 1       64076         57   0.08510        0            1
## 2       64076         17   0.07496        0            1
## 3       64076         14   0.07335        0            1
## 4       64076         51   0.08587        0            1
##   NegSelQCPvalPermutation NegSelQCPvalPermutationFDR NegSelQCGene
## 1                       1                          1            0
## 2                       1                          1            0
## 3                       1                          1            0
## 4                       1                          1            0
```

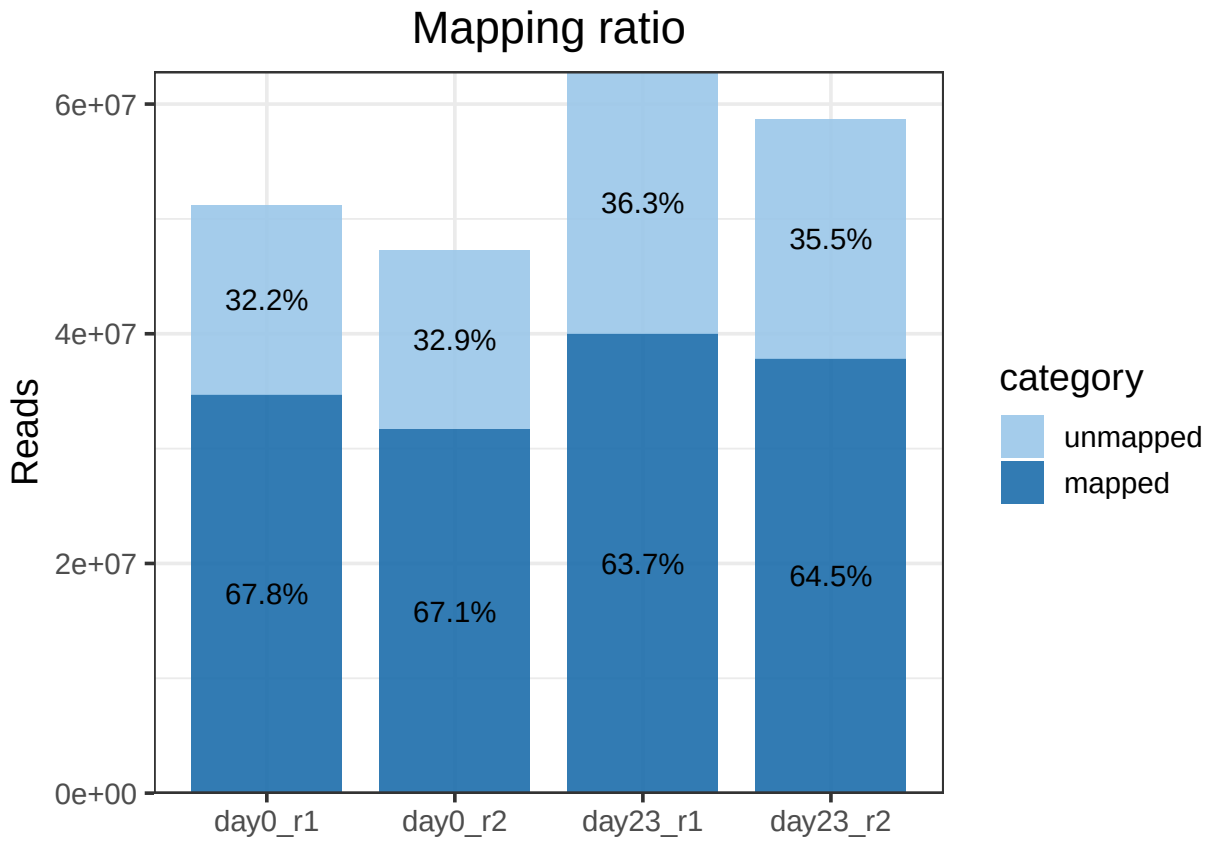
```
# Gini index
BarView(countsummary, x = "Label", y = "GiniIndex",
        ylab = "Gini index", main = "Evenness of sgRNA reads")
```



```
# Missed sgRNAs
countsummary$Missed = log10(countsummary$Zerocounts)
BarView(countsummary, x = "Label", y = "Missed", fill = "#394E80",
        ylab = "Log10 missed gRNAs", main = "Missed sgRNAs")
```




```
# Read mapping
MapRatesView(countsummary)
```



3) ANALYSIS OF MAGECK-RRA FOR IDENTIFICATION OF ESSENTIAL GENES

MAGECK-RRA

For CRISPR/Cas9 screens with two experimental conditions, MAGeCK-RRA is available for identification of essential genes. In MAGeCK-RRA results, the sgRNA summary and gene summary file summarizes the statistical significance of positive selections and negative selections at sgRNA level and gene level.

Gene Level:

```
file1 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/rra.gene_summary.txt")
gdatar = ReadRRA(file1)
head(gdatar)
```

```
##       id   Score      FDR
## 1  Cd274 -3.4698 0.002475
## 2  Psmb8 -3.7260 0.002475
## 3   Rela -3.2413 0.008251
## 4 Otulin -3.6018 0.008663
## 5  Ikbkb -3.4256 0.010891
## 6  Tcea1 -3.4236 0.042079
```

sgRNA Level:

```
file2 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/rra.sgrna_summary.txt")
sdatar = ReadsgRRA(file2)
```

```
head(sdatar)
```

```
##                   sgrna  Gene     LFC FDR
## 1 AAACATATAGTGTACCTCTA  Jak1 10.8910   0
## 2 TCCGAACCGAATCATCACTG  Jak1 10.9170   0
## 3 TGAATAAATCCATCAGACAG  Jak1 10.5970   0
## 4 GGATAGACGCCCAGCCACTG Stat1  9.9921   0
## 5 TTAATGACGAGCTCGTGGAG Stat1  9.2728   0
## 6 GAAAAGCAAGCGTAATCTCC Stat1  8.7931   0
```

To incorporate depmap data that are profiled in human cell lines, we will convert mouse gene names to homologous human genes for this dataset. Depmap is a cancer dataset profiling of genes.

```
gdatar$HumanGene = TransGeneID(gdatar$id, fromType = "symbol", toType = "symbol",
                               fromOrg = "mmu", toOrg = "hsa")
sdatar$HumanGene = TransGeneID(sdatar$Gene, fromType = "symbol", toType = "symbol",
                               fromOrg = "mmu", toOrg = "hsa")
```

Remove Duplicated Genes

```
idx = duplicated(gdatar$HumanGene)|is.na(gdatar$HumanGene)
gdata = gdatar[!idx, ]
head(gdatar)
```

```
##       id   Score      FDR HumanGene
## 1  Cd274 -3.4698 0.002475     CD274
## 2  Psmb8 -3.7260 0.002475     PSMB8
## 3   Rela -3.2413 0.008251      RELA
## 4 Otulin -3.6018 0.008663    OTULIN
## 5  Ikbkb -3.4256 0.010891     IKBKB
## 6  Tcea1 -3.4236 0.042079     TCEA1
```

Omit essential genes from the data, as these might be false positives because their importance in cell viability.

```
gdatar = OmitCommonEssential(gdatar, symbol = "HumanGene")
```

```
## see ?depmap and browseVignettes('depmap') for documentation
```

```
## loading from cache
```

```
## see ?depmap and browseVignettes('depmap') for documentation
```

```
## loading from cache
```

```
sdatar = OmitCommonEssential(sdatar, symbol = "HumanGene")
```

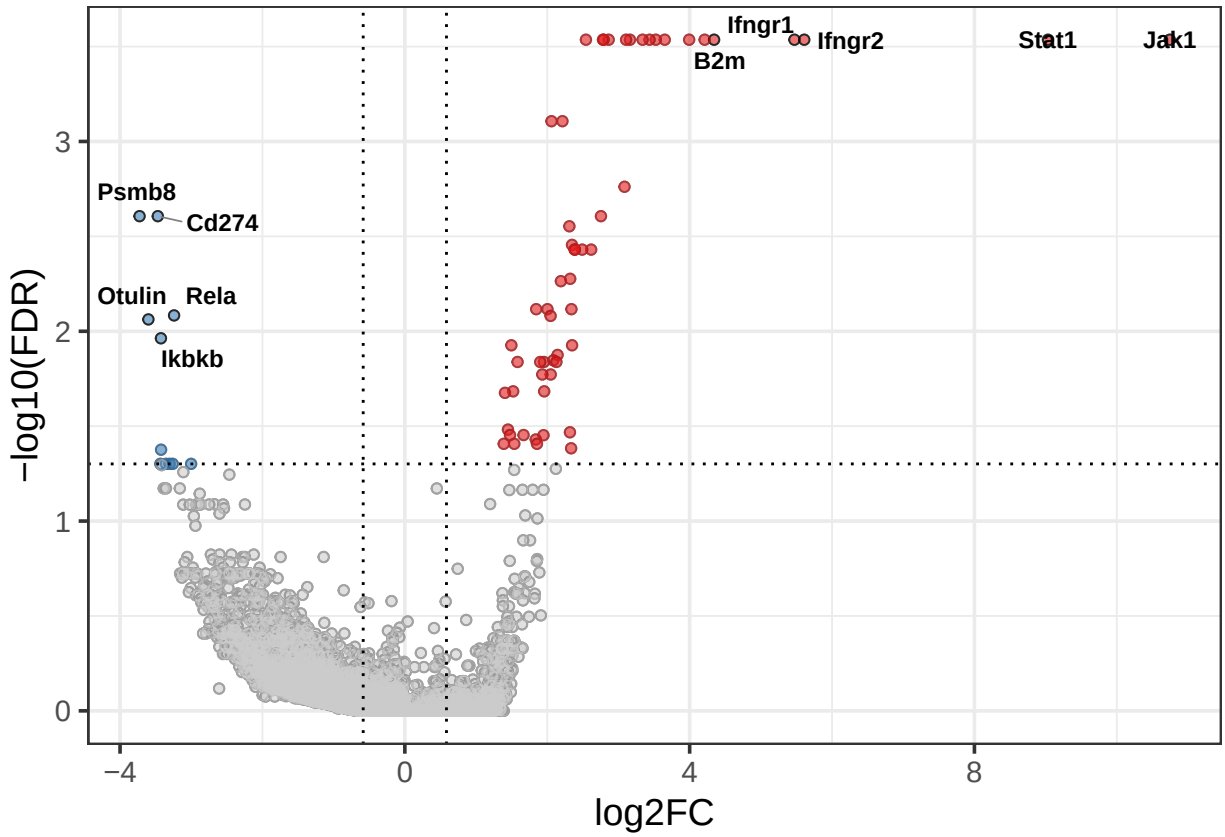
```
## see ?depmap and browseVignettes('depmap') for documentation
## loading from cache
```

```
## see ?depmap and browseVignettes('depmap') for documentation

## loading from cache
```

Visualization of negative selections and positive selections

```
p2r = VolcanoView(gdatar, x = "Score", y = "FDR", Label = "id")
print(p2r)
```

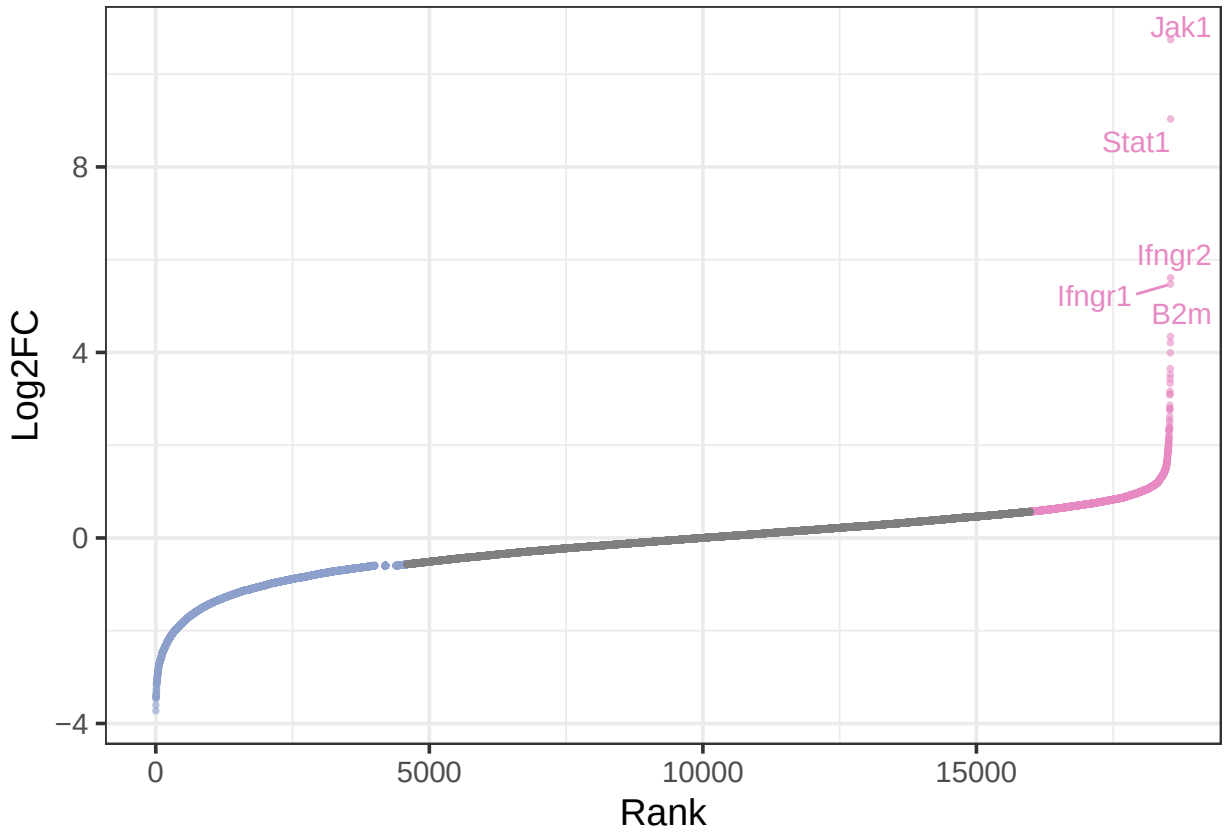


RANK PLOT

Rank all the genes based on their scores and label genes in the rank plot.

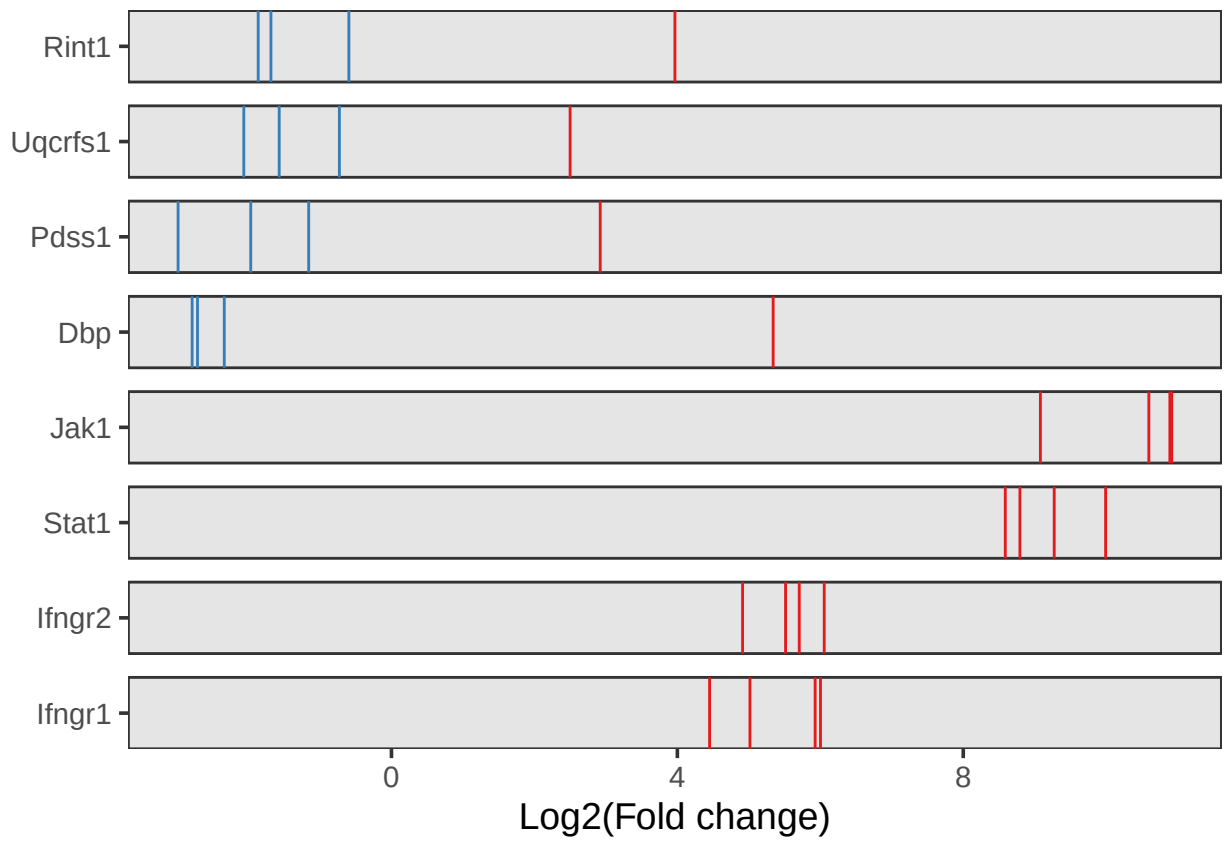
```
gdatar$Rank = rank(gdatar$Score)
p1r = ScatterView(gdatar, x = "Rank", y = "Score", label = "id",
                  top = 5, auto_cut_y = TRUE, ylab = "Log2FC",
                  groups = c("top", "bottom"))
print(p1r)
```

```
## Warning: ggrepel: 5 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To Visualize top selected genes in Rank Format

```
p2 = sgRankView(sdatar, top = 4, bottom = 4)
print(p2)
```



ANALYSIS USING MAGECK MLE

The MAGeCK-VISPR (mageck mle) utilizes a maximum likelihood estimation (MLE) for robust identification of CRISPR screen hits. It outputs beta scores and the associated statistics in multiple conditions. The beta score describes how a gene is selected: a positive beta score indicates positive selection, and a negative beta score indicates negative selection. Using mageck mle, we removed the baseline effect (plasmid sample) from all the three samples, including Pmel1_Input (B16F10 cells without T cell co-culture), Pmel1_Ctrl (B16F10 cells co-cultured with control T cells), and Pmel1 (B16F10 cells co-cultured with antigen specific T cells).

```
file3 = file.path(system.file("extdata", package = "MAGeCKFlute"),
                  "testdata/mle.gene_summary.txt")
# Read and visualize the file format
gdata = ReadBeta(file3)
head(gdata)
```

```
##      Gene Pmel1_Input Pmel1_Ctrl     Pmel1
## 1  Defb34    0.087884  -0.010034 -0.068015
## 2   Mndal    0.282860   0.033850 -0.161790
## 3   Cox8c    0.212140  -0.031618 -0.491360
## 4 Poldip3    0.125260  -0.123200 -0.450270
## 5   Bcas2   -0.196670   0.031254 -0.457530
## 6  Klk1b9   -0.220270  -0.019697 -0.443300
```

#NORMALIZATION OF BETA SCORES:

Control all samples with a consistent cell cycle, using information about essential genes, which are those that are indespensible for cell survival.

```
ctrlname = "Pmel1_Ctrl"
treatname = "Pmel1"
gdata$HumanGene = TransGeneID(gdata$Gene, fromType = "symbol", toType = "symbol",
                              fromOrg = "mmu", toOrg = "hsa")

gdata_cc = NormalizeBeta(gdata, id = "HumanGene", samples=c(ctrlname, treatname),
                         method="cell_cycle")
head(gdata_cc)
```

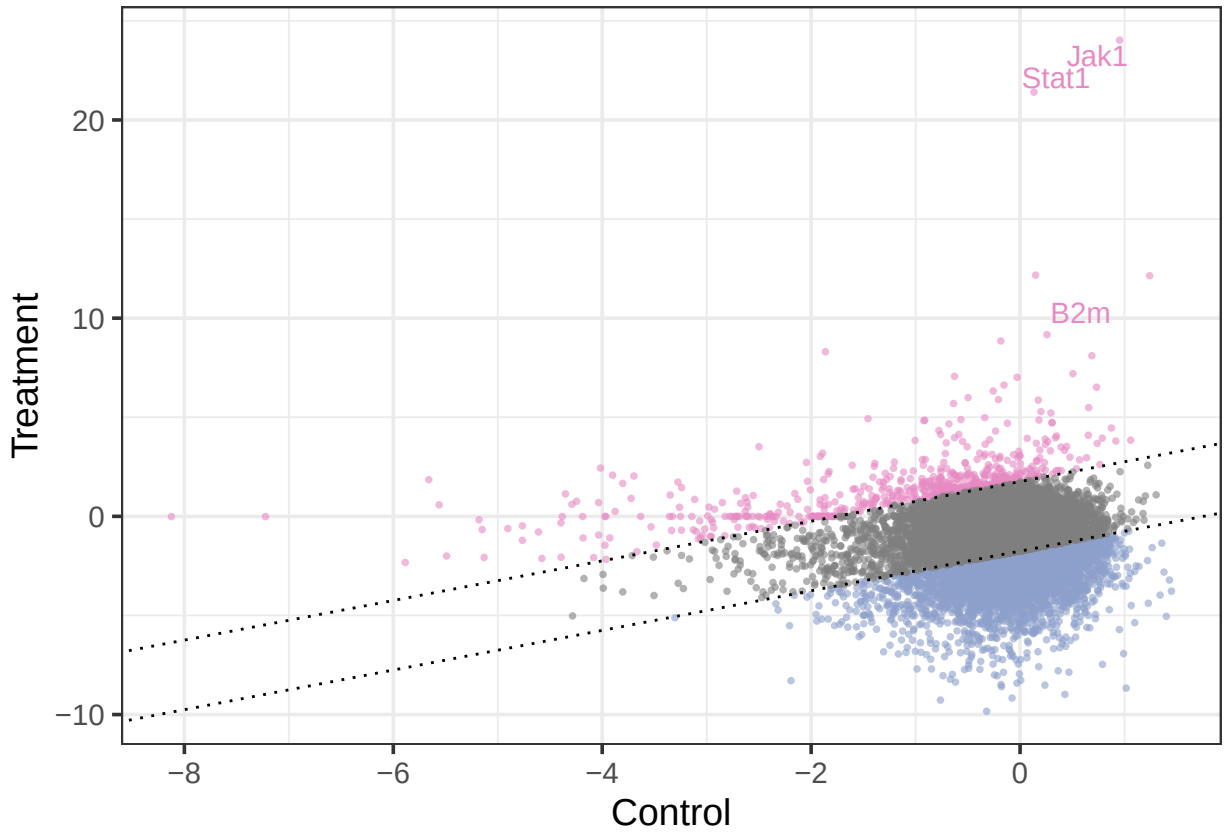
```
##      Gene Pmel1_Input  Pmel1_Ctrl      Pmel1 HumanGene
## 1  Defb34    0.087884 -0.06263323 -0.2425685  DEFB106A
## 2   Mndal    0.282860  0.21129508 -0.5770074     IFI16
## 3   Cox8c    0.212140 -0.19736271 -1.7523850     COX8C
## 4 Poldip3    0.125260 -0.76902670 -1.6058418   POLDIP3
## 5   Bcas2   -0.196670  0.19509059 -1.6317338     BCAS2
## 6  Klk1b9   -0.220270 -0.12295064 -1.5809840      KLK3
```

#Positive selection and negative selection Rank based on the difference between treatment and control Beta's

```
gdata_cc$Control = rowMeans(gdata_cc[,ctrlname, drop = FALSE])
gdata_cc$Treatment = rowMeans(gdata_cc[,treatname, drop = FALSE])

p1 = ScatterView(gdata_cc, "Control", "Treatment", label = "Gene",
                 auto_cut_diag = TRUE, display_cut = TRUE,
                 groups = c("top", "bottom"),
                 toplabels = c("Pbrm1", "Brd7", "Arid2", "Jak1", "Stat1", "B2m", "Bcl11a"))
print(p1)
```

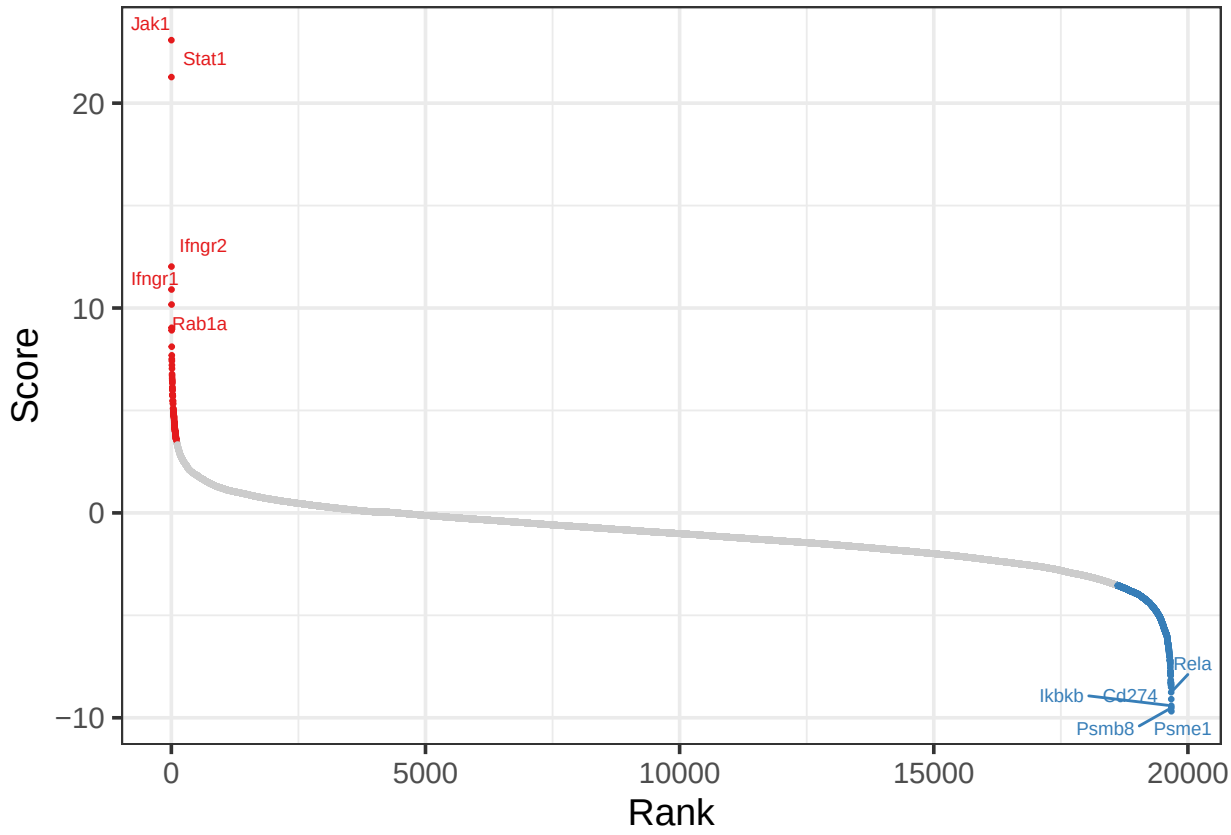
```
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



We can compare this with the ranks from MageCK-RRA

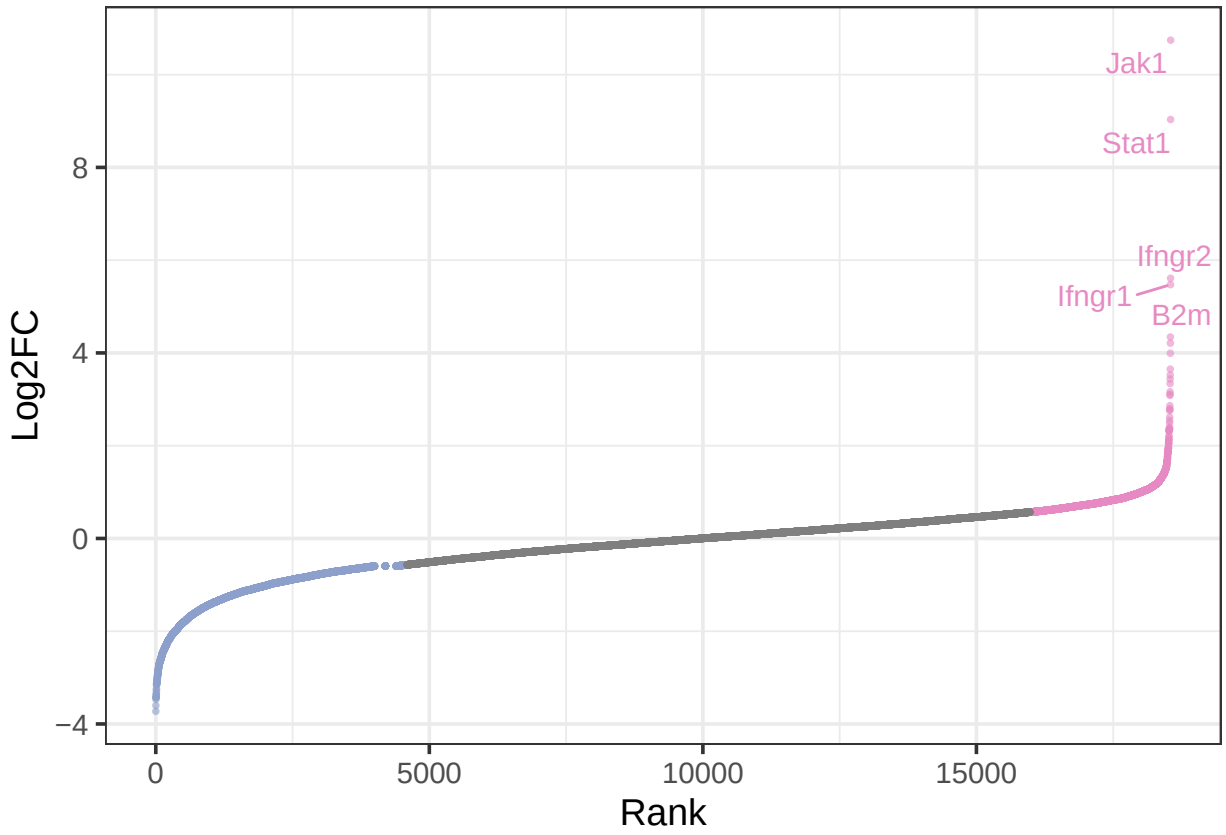
```
rankdata = gdata_cc$Treatment - gdata_cc$Control
names(rankdata) = gdata_cc$Gene
RankView(rankdata)
```

```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s fill values.
```




```
print(p1r)
```

```
## Warning: ggrepel: 5 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Identifying treatment specific genes from Mageck-MLE results

```
p1 = ScatterView(gdata_cc, x="Pmel1_Ctrl", y="Pmel1", label = "Gene",
                 model = "ninesquare", display_cut = TRUE, force = 2)
```