Assigment 3 - Aman kumar - 17025

Nef (Negative Regulatory Factor) is a small 27 35 kDa protein that is coded by the HIV genome.This gene performs several different tasks during infection. However, despite being studied for more than two decades, it was not known how this protein increases the infectivity of new virus particles released from the cell. In the year 2015, two groups of scientists independently solved this 20 year old mystery and shed light on a pair of new genes that could restrict HIV in absence of the Nef protein (Rosa et al. 2015; Usami et al. 2015).

I have use this following paper to replicate results from the paper:

Rosa A, Chande A, Ziglio S, De Sanctis V, Bertorelli R, Goh SL, McCauley SM, Nowosielska A, Antonarakis SE, Luban J, et al. 2015. HIV 1 Nef promotes infection by excluding SERINC5 from virion incorporation. Nature 526:212—217.217.

They measured infectivity ratio of these same cell lines with Nef+ and Nef HIV. Using this data, they looked at correlations between expression levels of each gene with the infectivity ratio's. Genes whose expression level showed a strong correlation with the infectivity ratio were chosen as candidate genes that were involved in restricting the virus in the absence of the Nef gene . Further experimental validation of these candidate genes lead to interesting new discovery that.

```
getwd()
```

```
## [1] "D:/IISER BHOPAL/SEM 8/DSE 401/Assignment 3"
```

```
#setwd("D:/IISER BHOPAL/SEM 8/DSE 401/Assignment 3")
#getwd()
```

Loading the libraries

```
#install.packages("psych")
```

```
library(tidyverse)
library(dplyr)
library(tidyr)
library(Hmisc)
library(ggplot2)
library(psych)
library(factoextra)
library(devtools)
```

Reading files

Loading Read-count files

```
load_htseq <- list.files(pattern = "*.htseq")
all_samples <- lapply(load_htseq, read.delim, header=F)
all_samples <- as.data.frame(all_samples)
colnames(all_samples)<- c("MT4","MT4_count","HSB2","HSB2_count","HT1080","HT1080_count","RAJI",
"RAJI_count","CEM SS","CEM SS_count","DAUDI","DAUDI_count","C8166","C8166_count","RAMOS","RAMOS_
count","IMR90","IMR90_count","CEM_A_301","CEM_A_301_count","CEM_X_174","CEM_X_174_count","WI38",
"WI38_count","JURKAT_E6.1","JURKAT_E6.1_count","bl41","bl41_count","Jurkat_tag","Jurkat_tag_coun
t")
```

Extracting only count columns

```
samples <- all_samples[c(1, seq(2, 30, 2))]
colnames(samples)[1] <- c("Gene_ID")
```

Loading Inactivity ratio file

```
infect <- read.delim("infect.txt", sep="\t", header= F)
colnames(infect) <- c("accession_id", "infectivity_ratio", "cell_lines")
infect <- arrange(infect,accession_id)
```

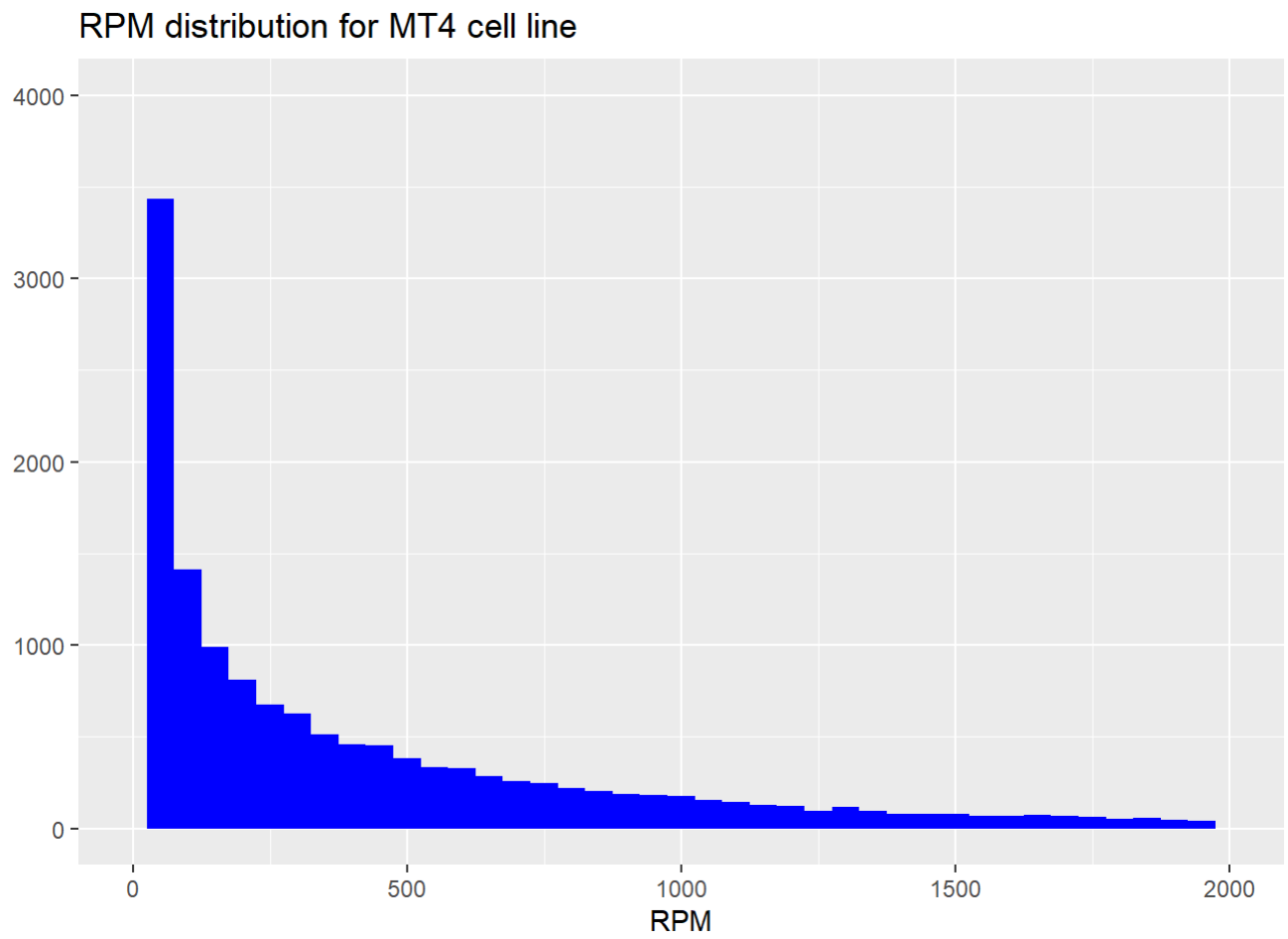
Data Visualization

```
MT4 = read.delim("SRR2166624.htseq", header=F)
MT4 <- MT4[MT4$V2 != 0,]
MT4 <- MT4[-c(58303, 58304, 58305, 58306, 58307),]
```

```
qplot(MT4$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for MT4 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2000),
      ylim=c(0,4000))
```

```
## Warning: Removed 1291 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

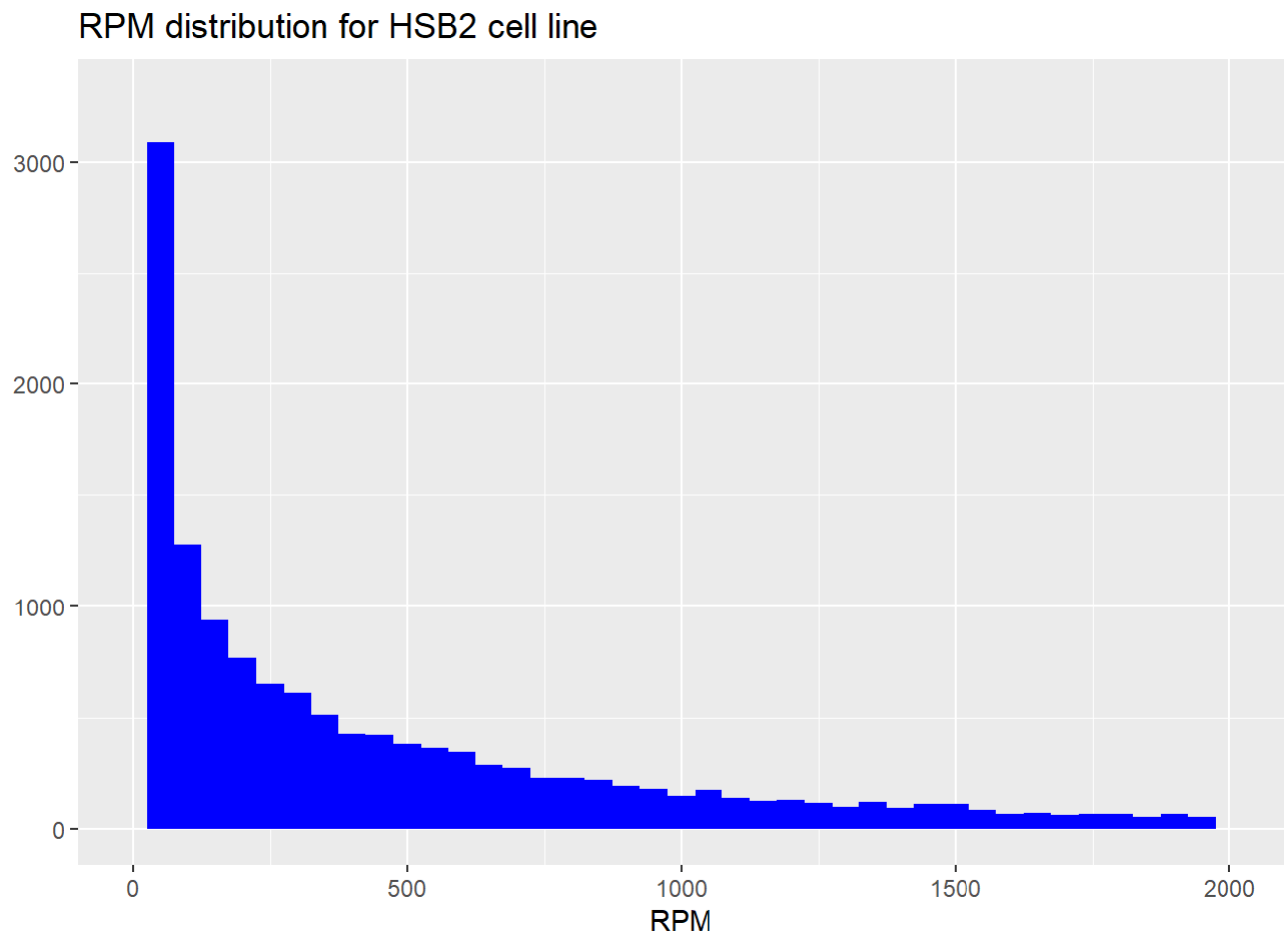



```
HSB2 = read.delim("SRR2166625.htseq", header=F)
HSB2 <- HSB2[HSB2$V2 != 0,]
```

```
qplot(HSB2$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for HSB2 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2000),
      ylim=c(0,3300))
```

```
## Warning: Removed 1380 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

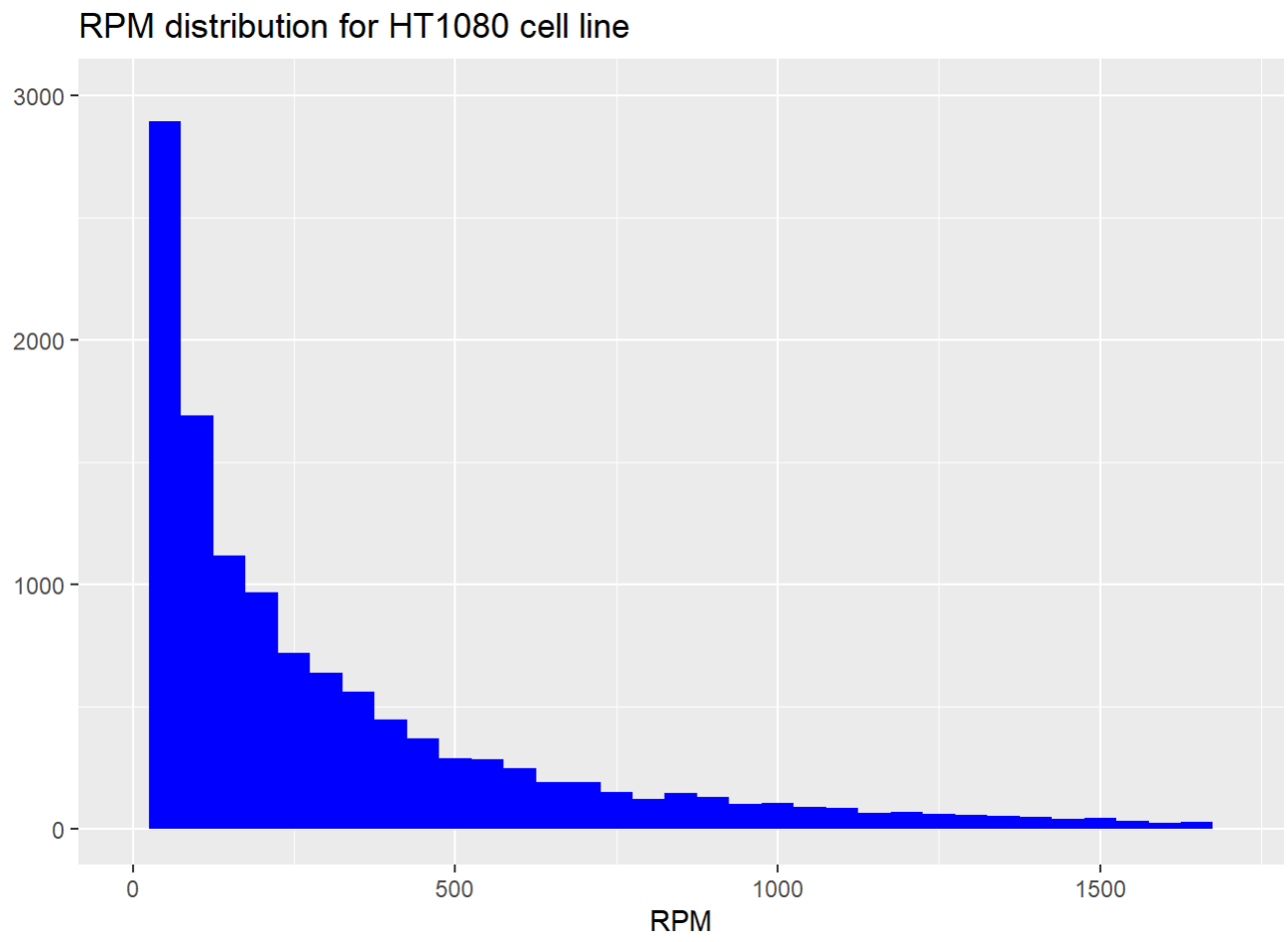



```
HT1080 = read.delim("SRR2166626.htseq", header=F)
HT1080 <- HT1080[HT1080$V2 != 0,]
```

```
qplot(HT1080$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for HT1080 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,1700),
      ylim=c(0,3000))
```

```
## Warning: Removed 688 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

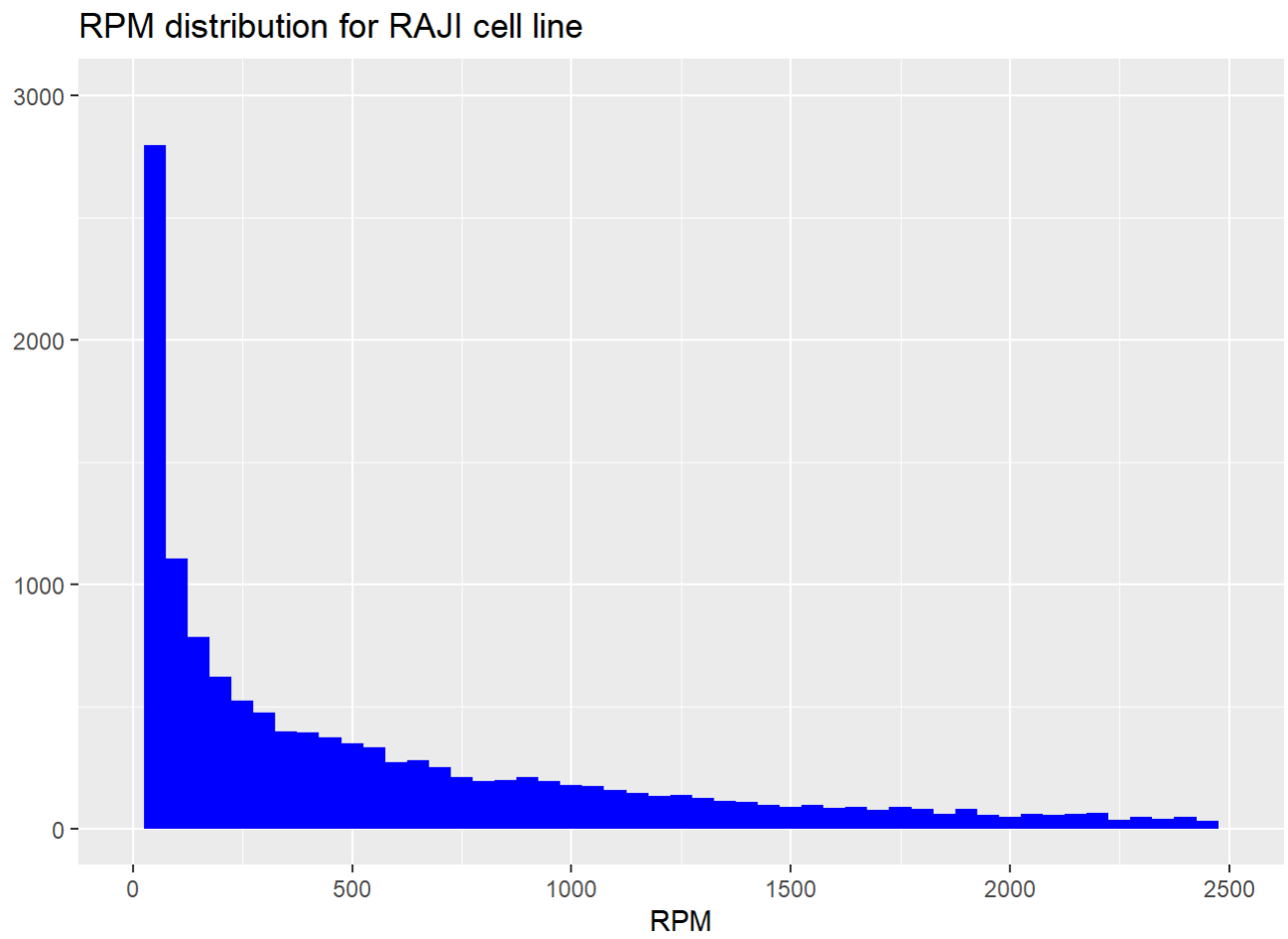



```
RAJI = read.delim("SRR2166627.htseq", header=F)
RAJI <- RAJI[RAJI$V2 != 0,]
```

```
qplot(RAJI$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for RAJI cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2500),
      ylim=c(0,3000))
```

```
## Warning: Removed 1486 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

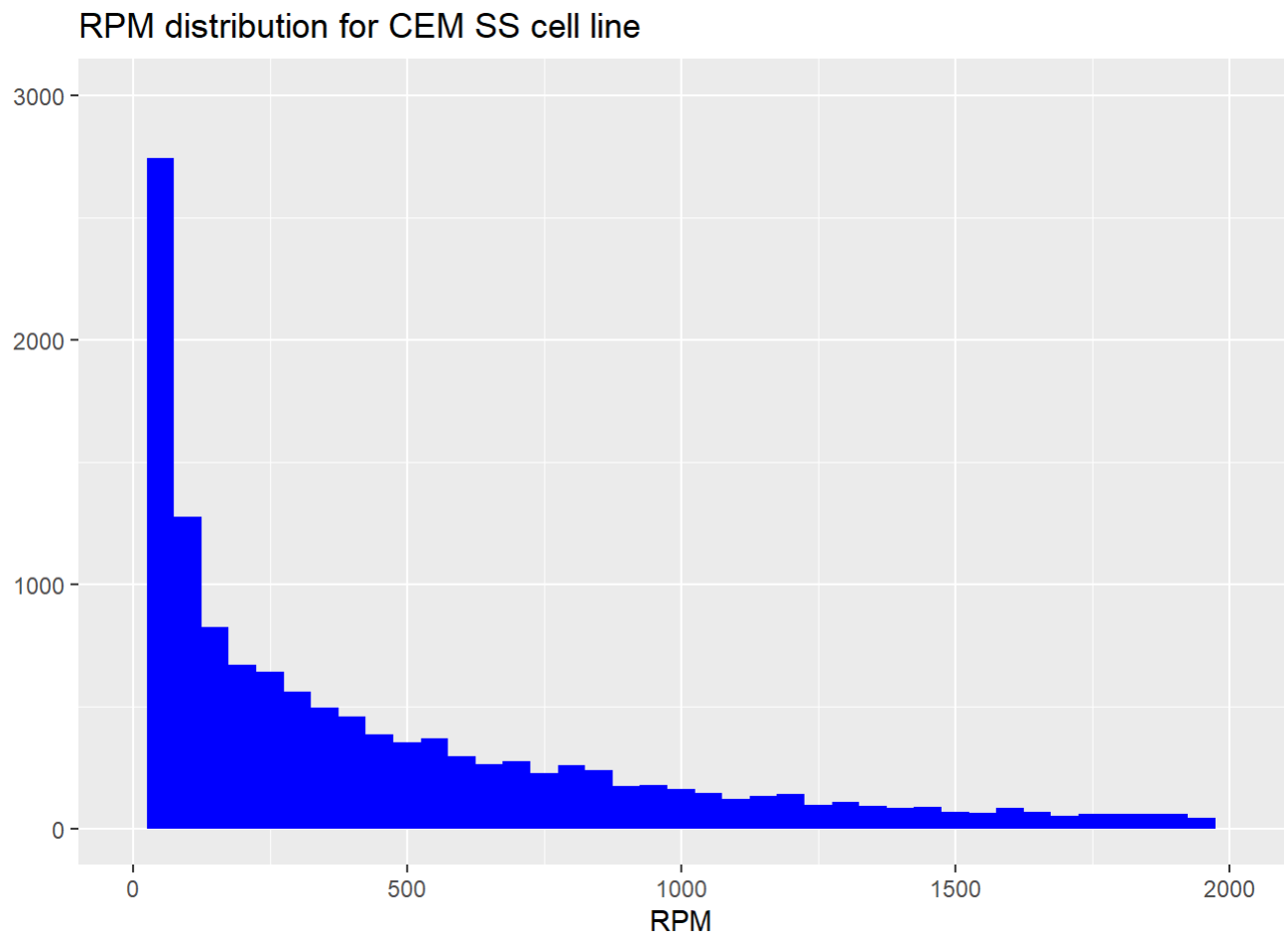



```
cell_line = read.delim("SRR2166628.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for CEM SS cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2000),
      ylim=c(0,3000))
```

```
## Warning: Removed 1155 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

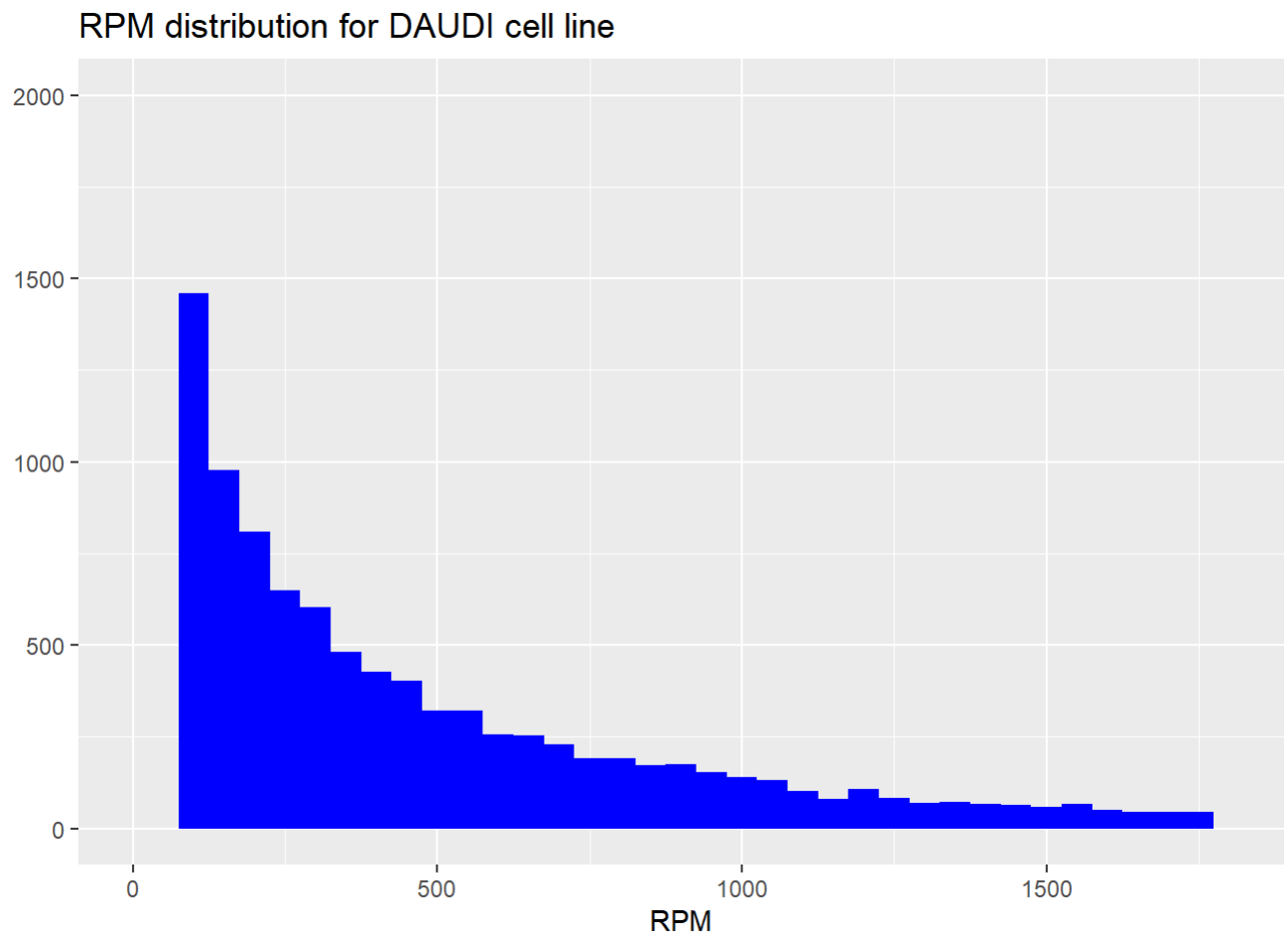



```
cell_line = read.delim("SRR2166629.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for DAUDI cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,1800),
      ylim=c(0,2000))
```

```
## Warning: Removed 1070 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```

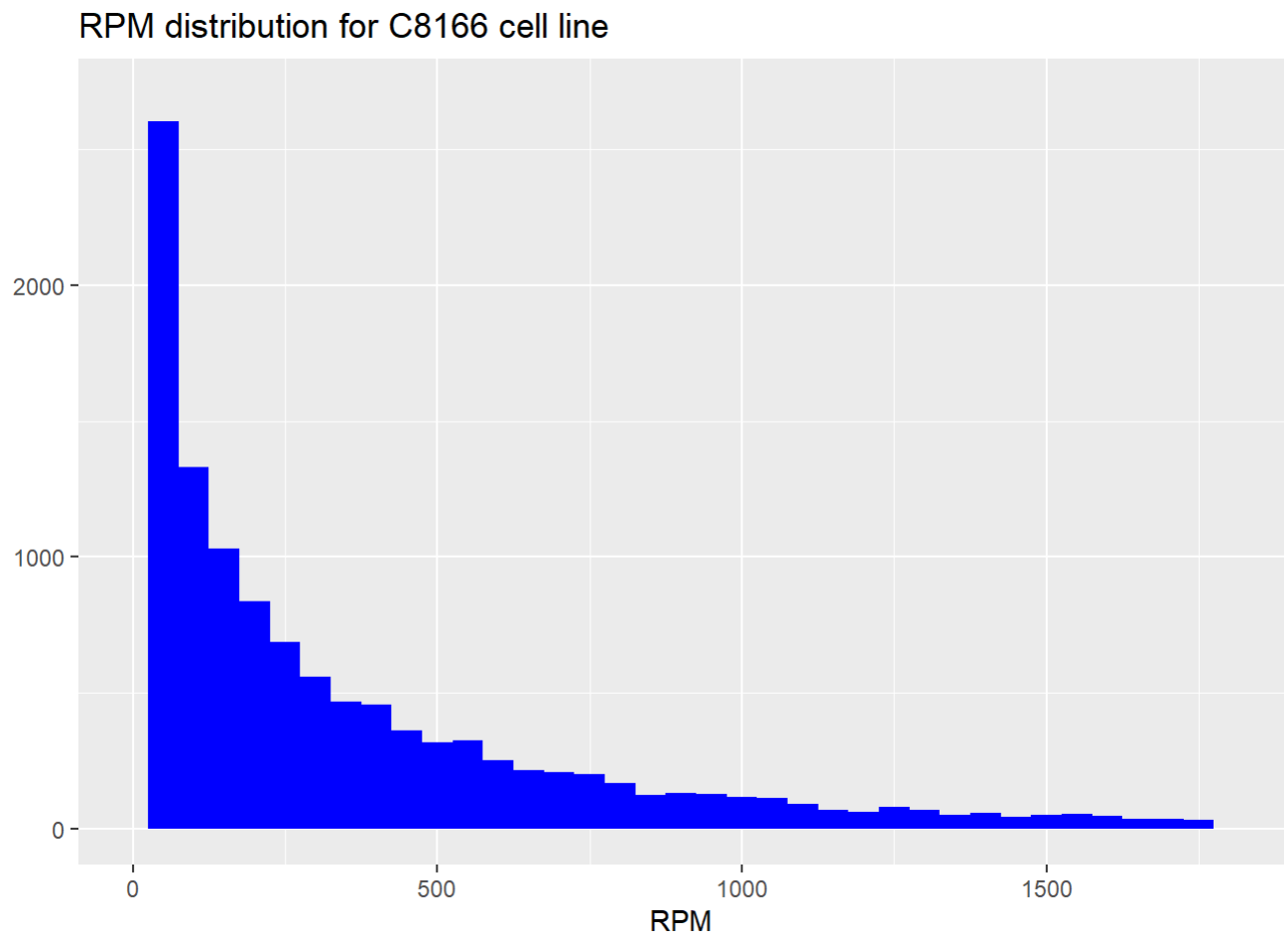



```
cell_line = read.delim("SRR2166630.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for C8166 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,1800),
      ylim=c(0,2700))
```

```
## Warning: Removed 874 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

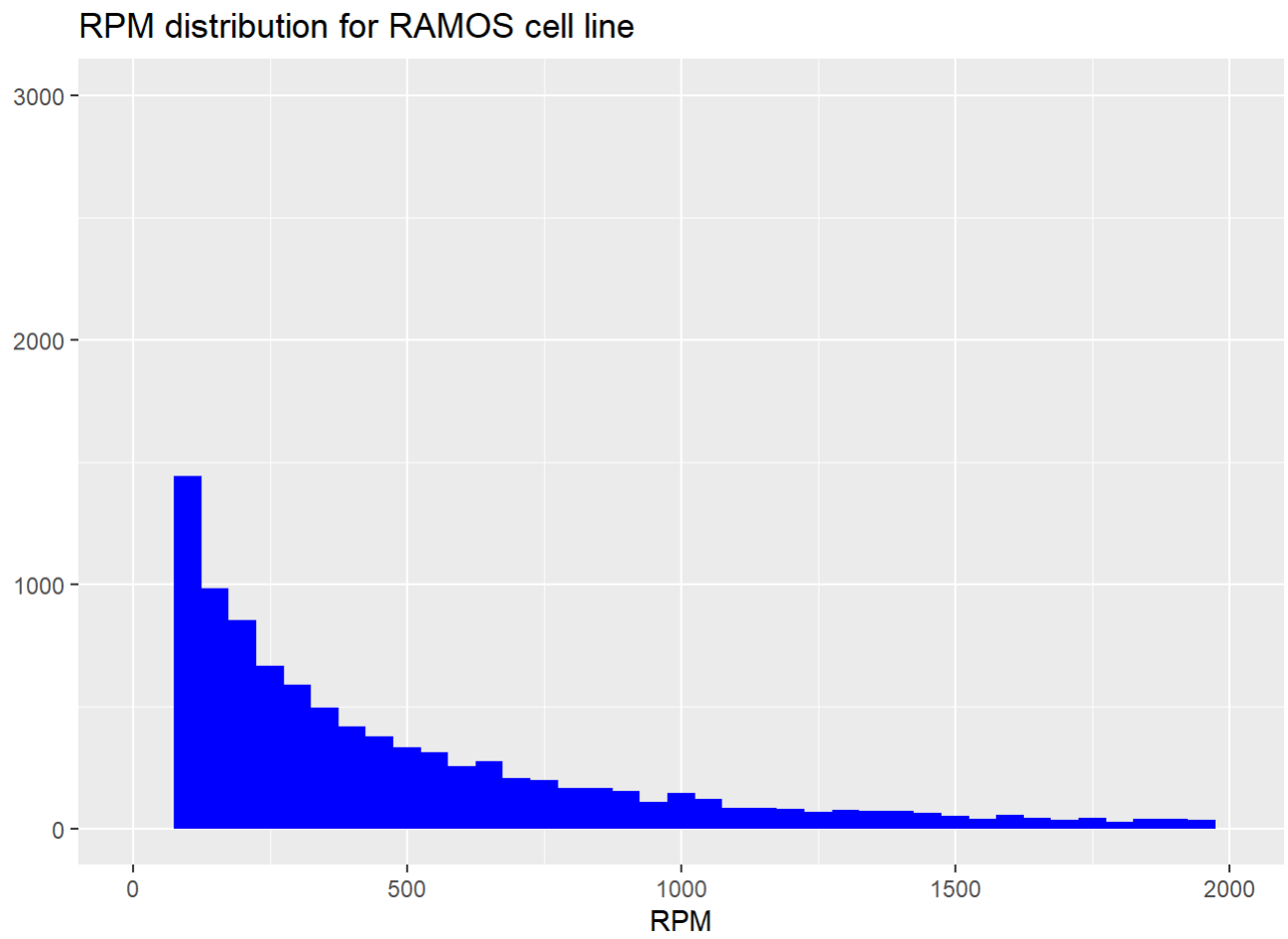



```
cell_line = read.delim("SRR2166631.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for RAMOS cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2000),
      ylim=c(0,3000))
```

```
## Warning: Removed 779 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```

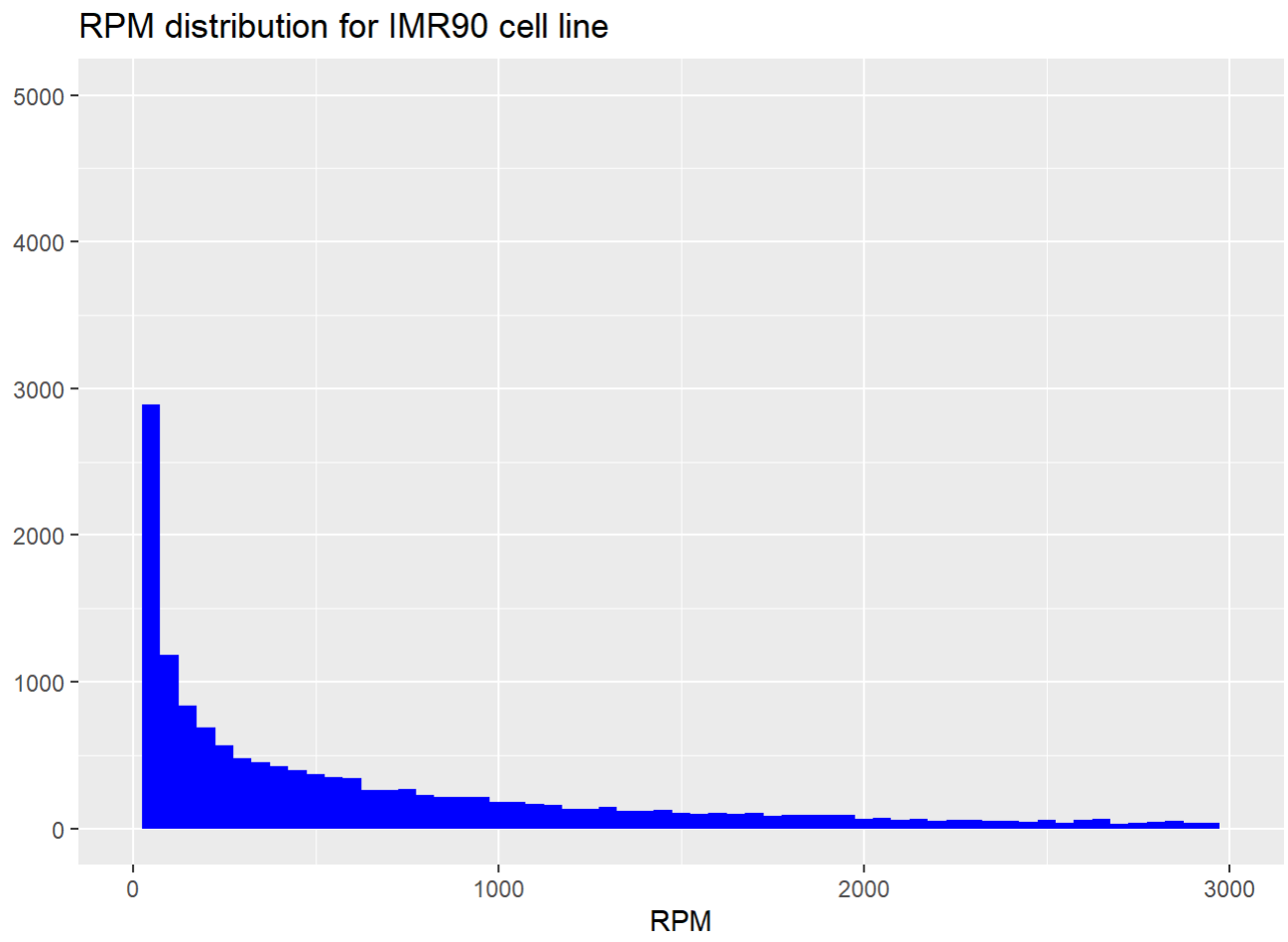



```
cell_line = read.delim("SRR2166632.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for IMR90 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,3000),
      ylim=c(0,5000))
```

```
## Warning: Removed 1715 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

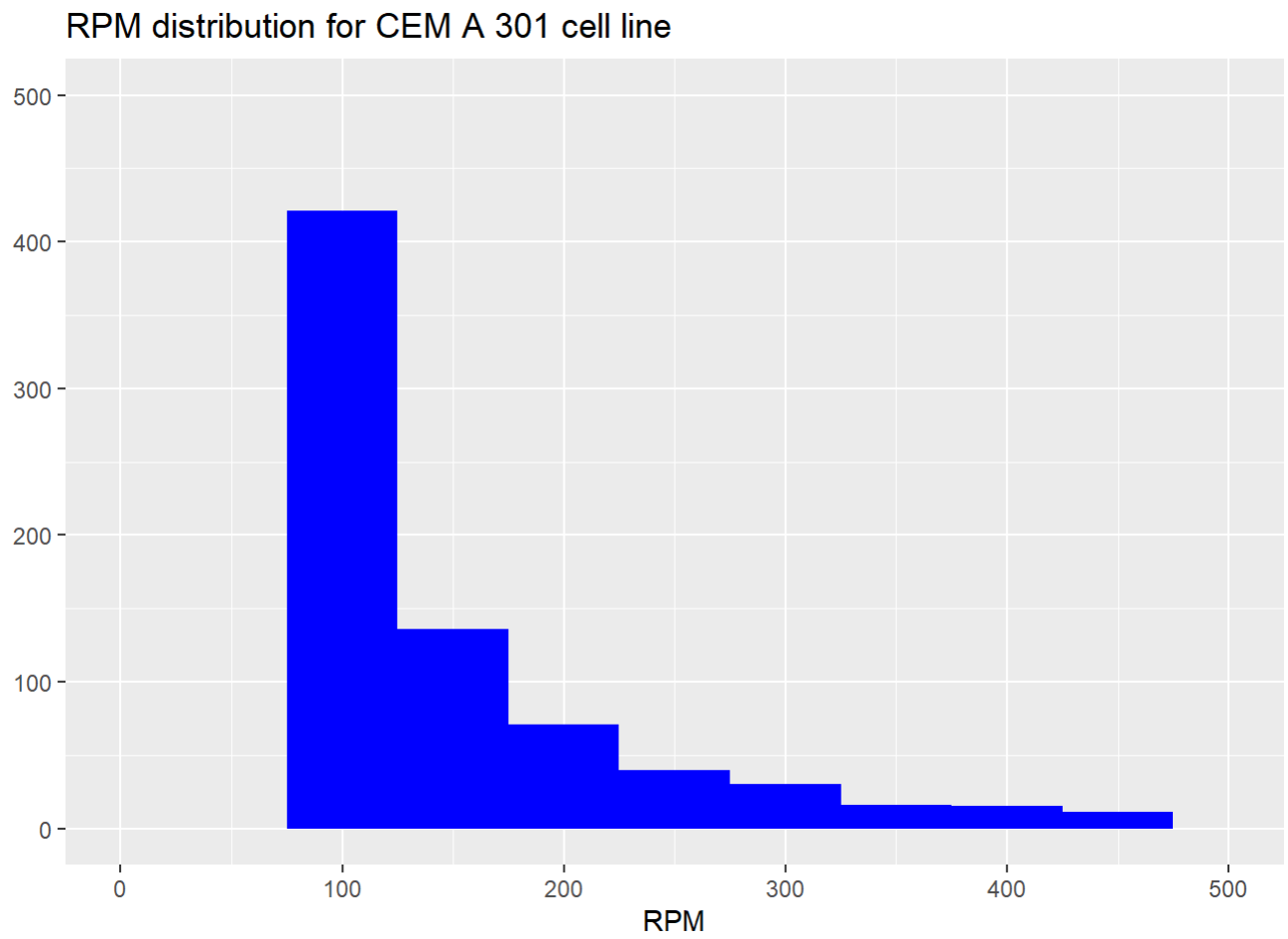



```
cell_line = read.delim("SRR2166633.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for CEM A 301 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,500),
      ylim=c(0,500))
```

```
## Warning: Removed 43 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```

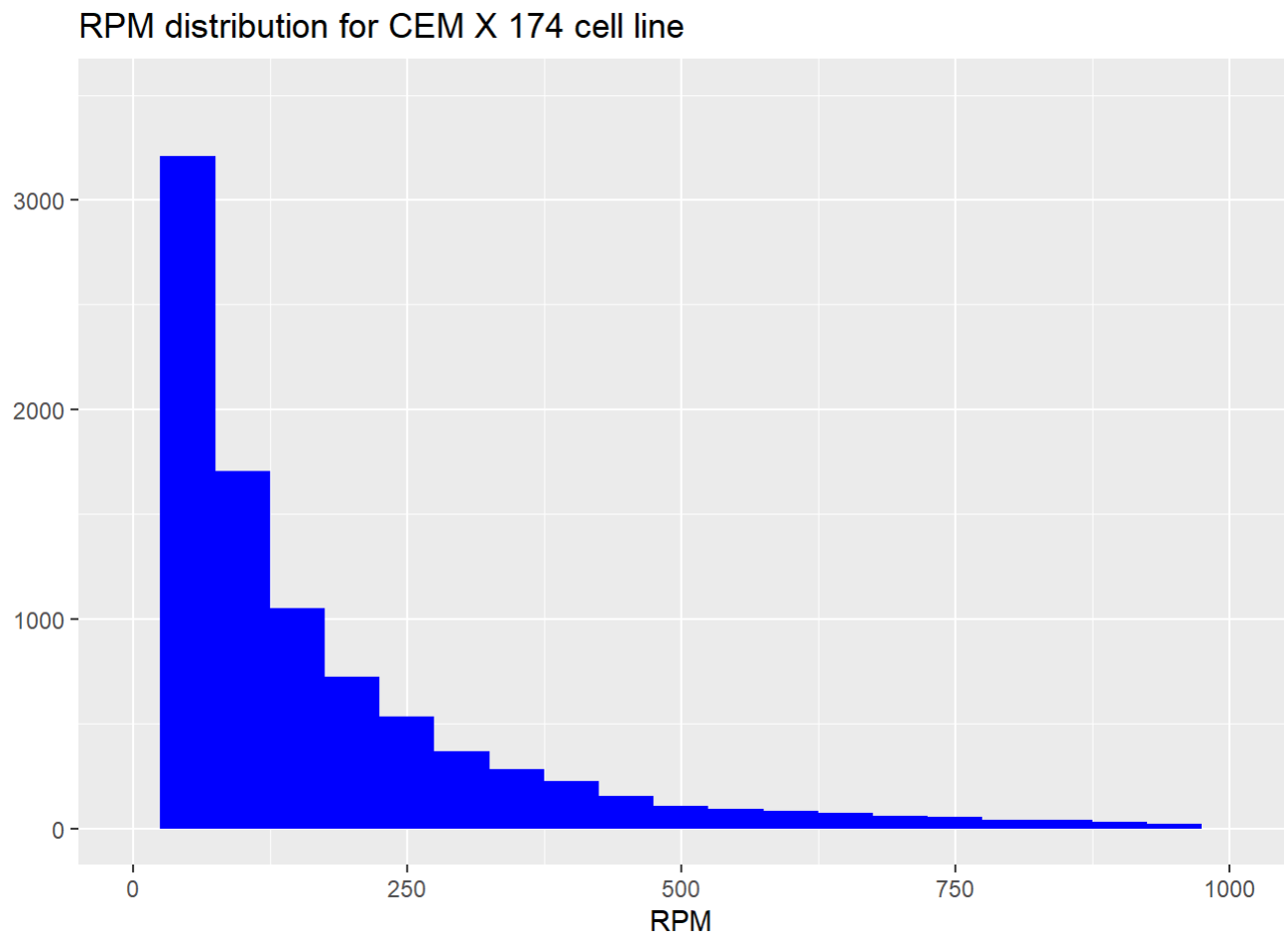



```
cell_line = read.delim("SRR2166634.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for CEM X 174 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,1000),
      ylim=c(0,3500))
```

```
## Warning: Removed 392 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

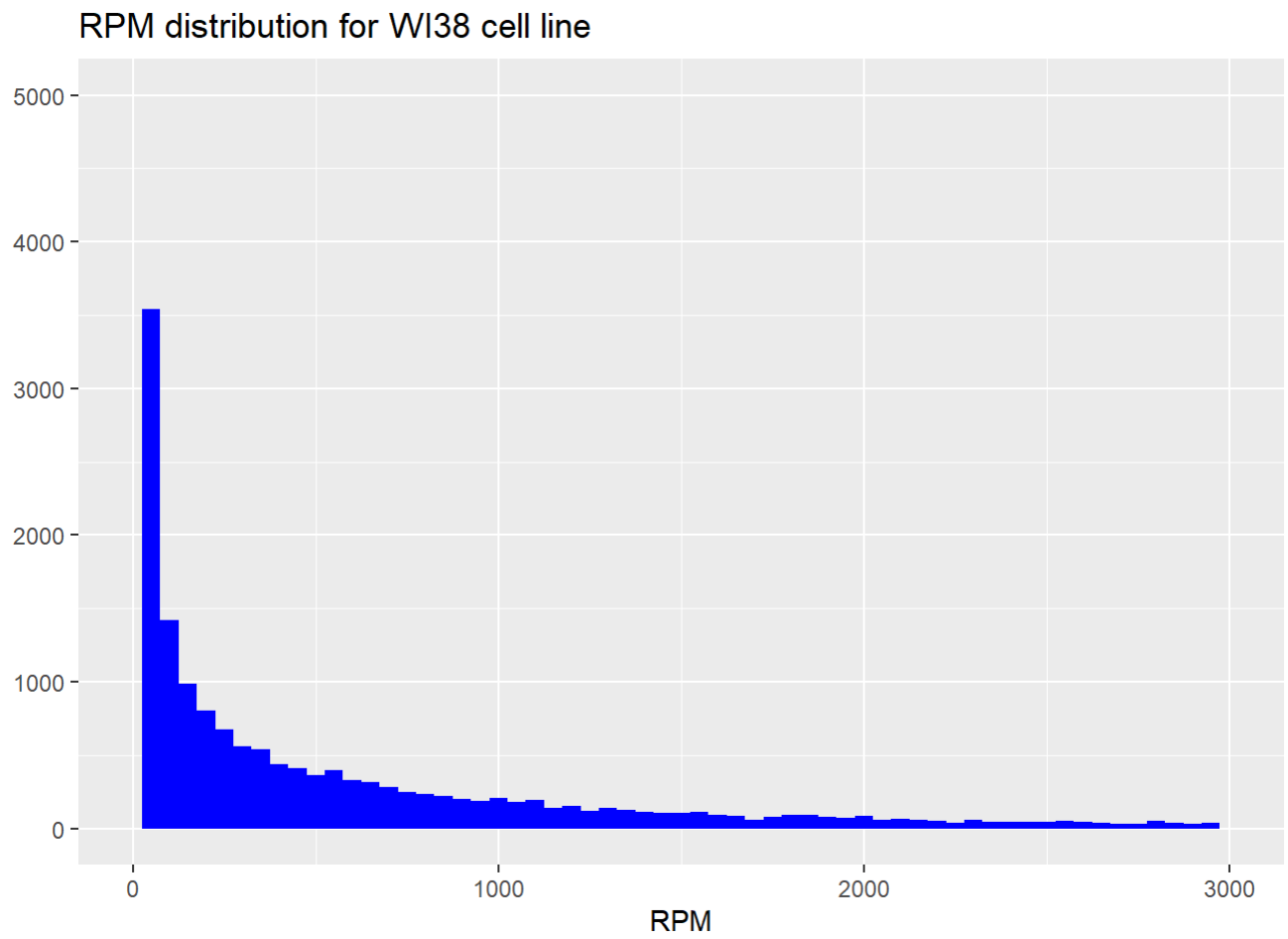



```
cell_line = read.delim("SRR2166635.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for WI38 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,3000),
      ylim=c(0,5000))
```

```
## Warning: Removed 1471 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

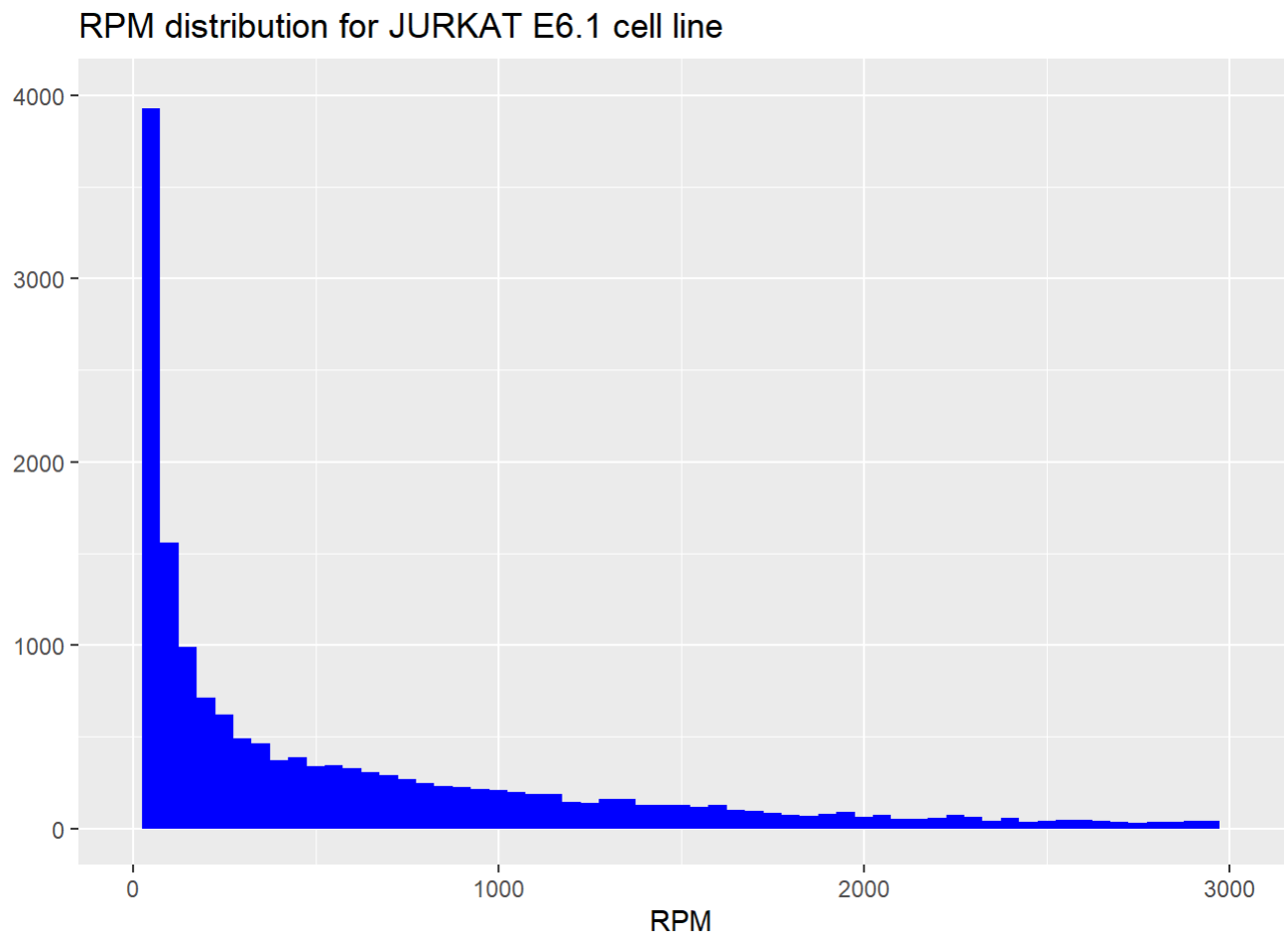



```
cell_line = read.delim("SRR2166636.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for JURKAT E6.1 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,3000),
      ylim=c(0,4000))
```

```
## Warning: Removed 1361 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

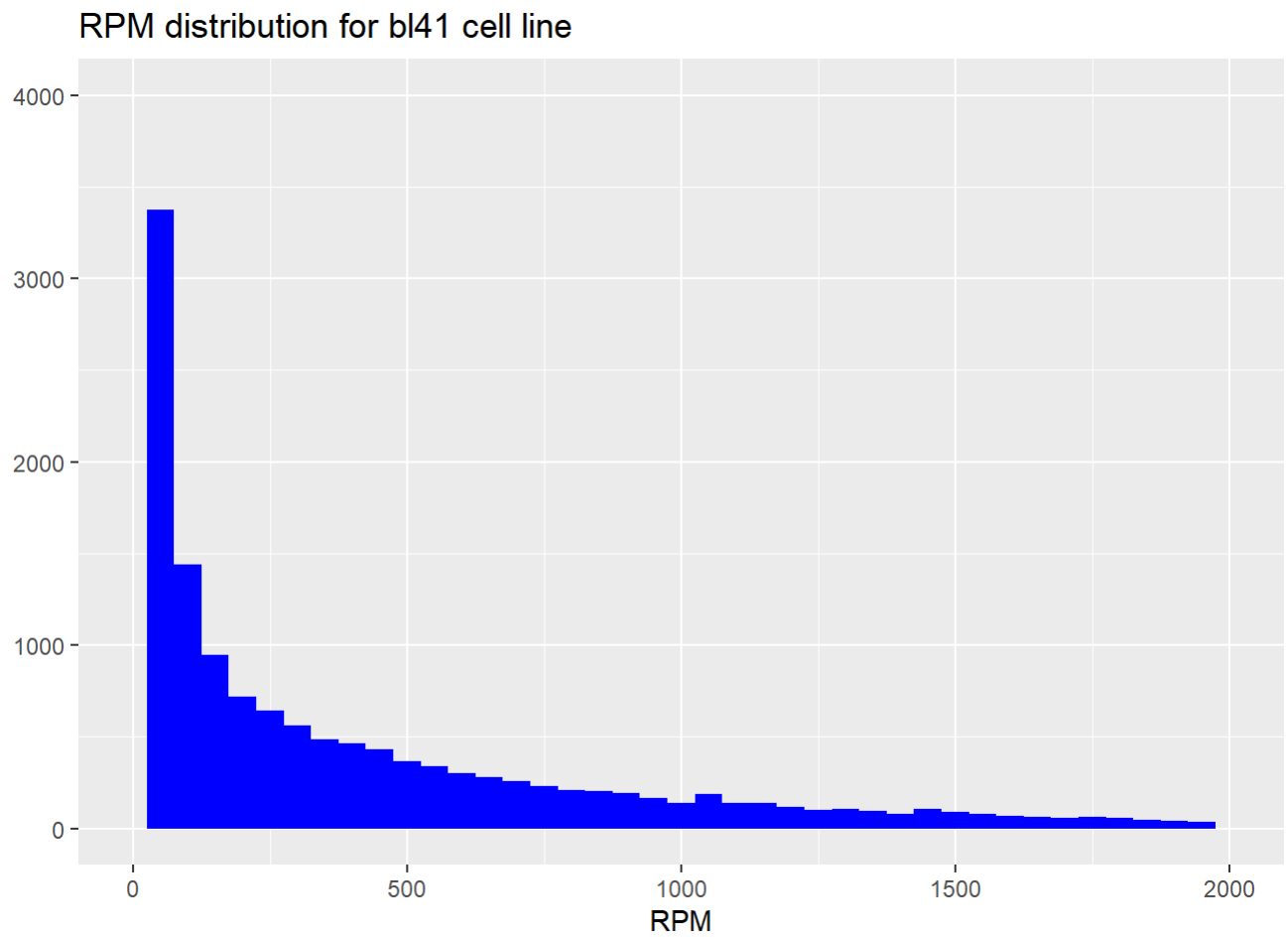



```
cell_line = read.delim("SRR2166637.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for bl41 cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2000),
      ylim=c(0,4000))
```

```
## Warning: Removed 1273 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

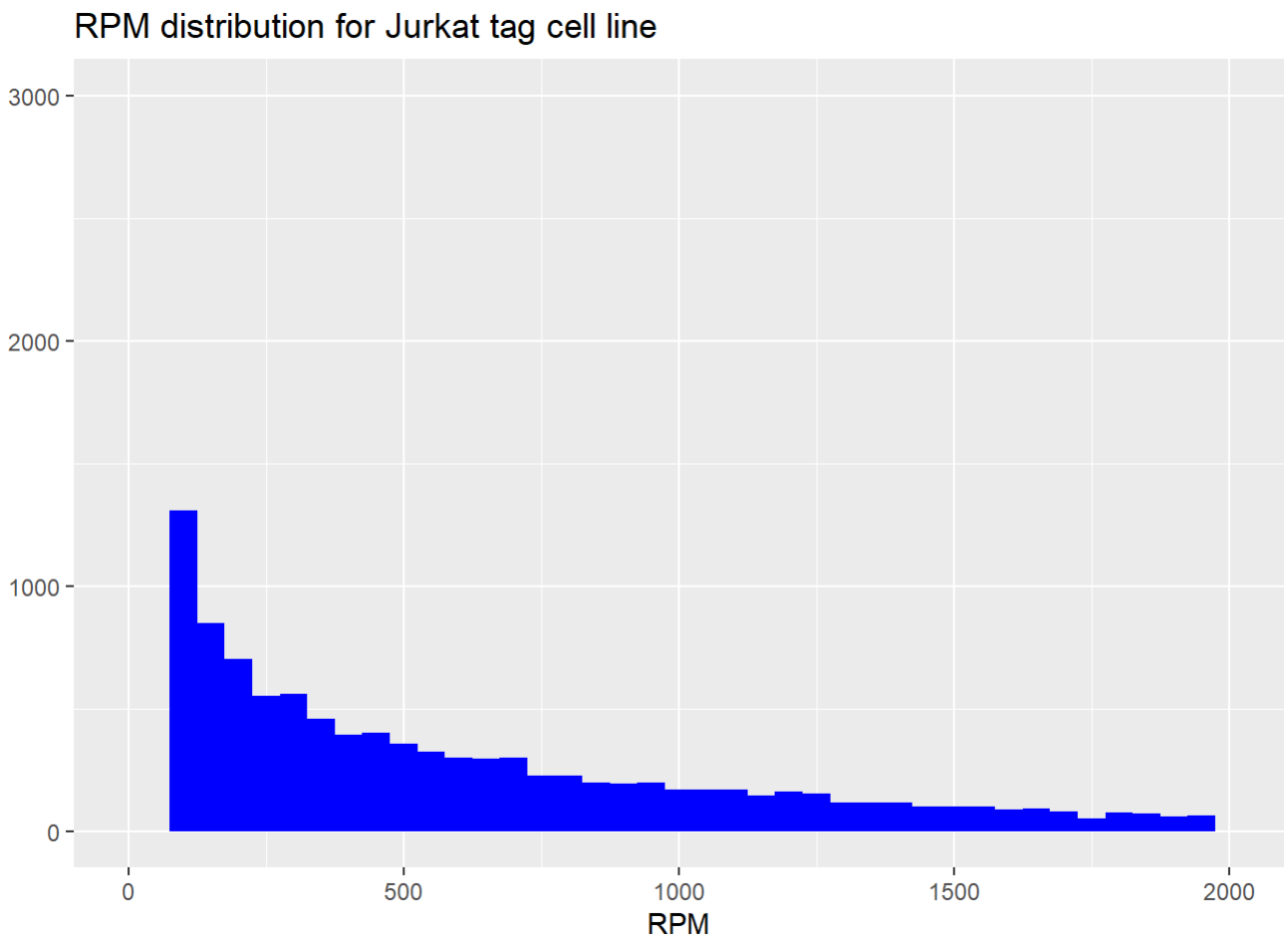



```
cell_line = read.delim("SRR2166638.htseq", header=F)
cell_line <- cell_line[cell_line$V2 != 0,]
```

```
qplot(cell_line$V2,
      geom="histogram",
      binwidth = 50,
      main = "RPM distribution for Jurkat tag cell line",
      xlab = "RPM",
      fill=I("blue"),
      xlim=c(0,2000),
      ylim=c(0,3000))
```

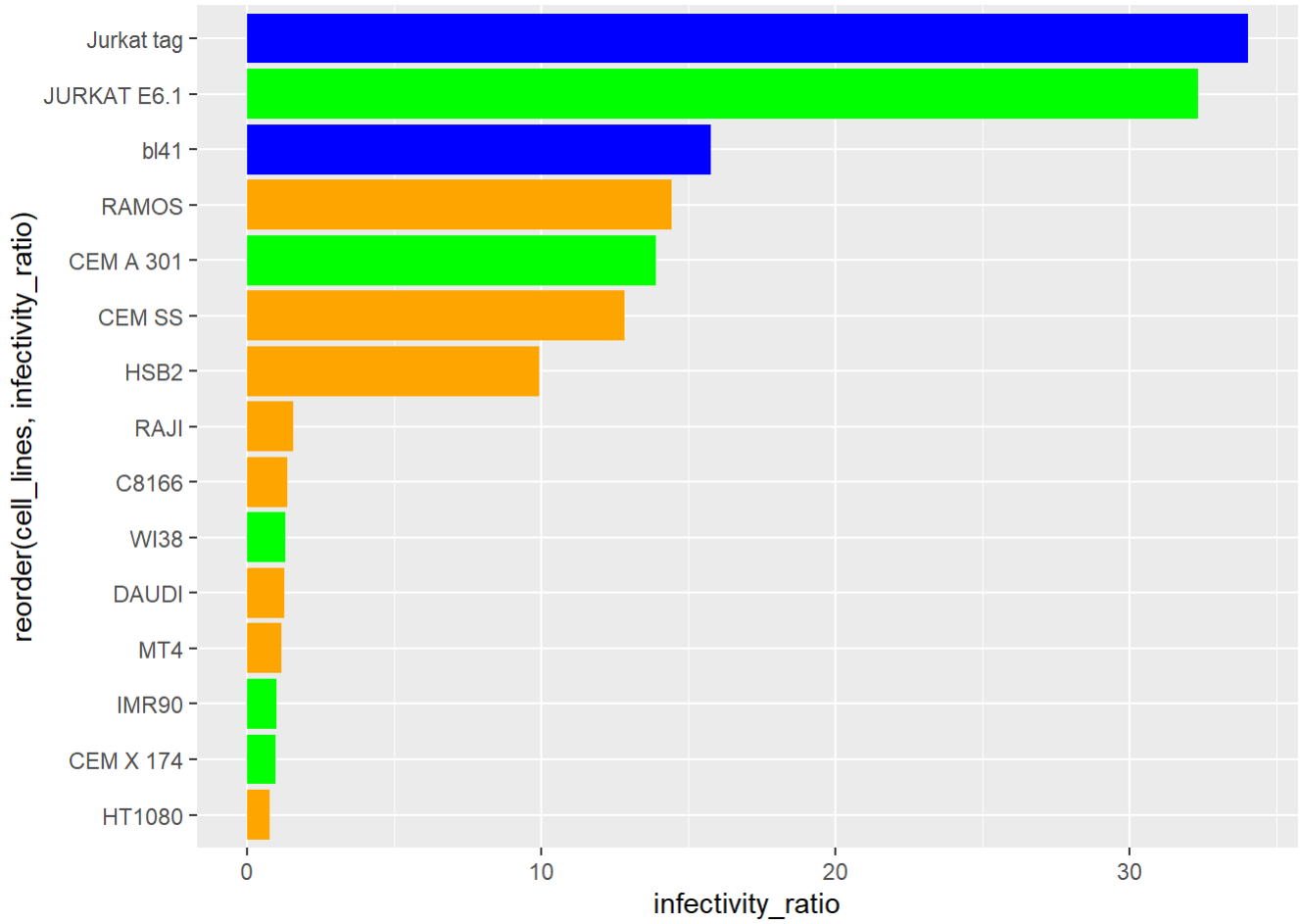
```
## Warning: Removed 1933 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```



Bar Plot

```
ggplot(data = infect, aes(x= reorder(cell_lines, infectivity_ratio), y= infectivity_ratio))+geom
_bar(stat = "identity", fill=c("orange", "orange", "orange", "orange", "orange", "orange", "oran
ge", "orange", "green", "green", "green", "green", "green", "blue", "blue" ))+ coord_flip()
```



Normalizing the datasets

Here we are making function to convert reads into normal reads per million

```
Normaliser <- function(x){
  x/(sum(x)/1000000)
}
```

Using the above function in our samples dataset

```
samples[,2:16] <- lapply(samples[,2:16], Normaliser)
```

Dividing data into 6 subsets for better computation

```
sample_1 <- samples[1:10000,]
sample_2 <- samples[10001:20000,]
sample_3 <- samples[20001: 30000,]
sample_4 <- samples[30001: 40000,]
sample_5 <- samples [40001: 50000,]
sample_6 <- samples [50001: 58307,]
```

re-defing data

transposing the datasets

```
sample_t_1 <- as.data.frame(t(sample_1))
names(sample_t_1) <- sample_t_1[1,]
sample_t_1 <- sample_t_1[-1,]

sample_t_2 <- as.data.frame(t(sample_2))
names(sample_t_2) <- sample_t_2[1,]
sample_t_2 <- sample_t_2[-1,]

sample_t_3 <- as.data.frame(t(sample_3))
names(sample_t_3) <- sample_t_3[1,]
sample_t_3 <- sample_t_3[-1,]

sample_t_4 <- as.data.frame(t(sample_4))
names(sample_t_4) <- sample_t_4[1,]
sample_t_4 <- sample_t_4[-1,]

sample_t_5 <- as.data.frame(t(sample_5))
names(sample_t_5) <- sample_t_5[1,]
sample_t_5 <- sample_t_5[-1,]

sample_t_6 <- as.data.frame(t(sample_6))
names(sample_t_6) <- sample_t_6[1,]
sample_t_6 <- sample_t_6[-1,]
```

```
sample_t_1_infect <- data.frame(infect$infectivity_ratio ,sample_t_1)
sample_t_2_infect <- data.frame(infect$infectivity_ratio ,sample_t_2)
sample_t_3_infect <- data.frame(infect$infectivity_ratio ,sample_t_3)
sample_t_4_infect <- data.frame(infect$infectivity_ratio ,sample_t_4)
sample_t_5_infect <- data.frame(infect$infectivity_ratio ,sample_t_5)
sample_t_6_infect <- data.frame(infect$infectivity_ratio ,sample_t_6)
```

creating data frame which will use in doing correlation test

```
samples_t <- as.data.frame(t(samples))
names(samples_t) <- samples_t[1,]
samples_t <- samples_t[-1,]

samples_infect <- data.frame(infect$infectivity_ratio ,samples_t)
```

correlation test

```
i <- c()
cor_e <- c()
cor_p <- c()

for (i in 2:58308)
{
  m = cor.test(samples_infect[,1],as.numeric(samples_infect[,i]),method = 'spearman', use = 'pai
rwise.complete.obs',p.adjust.methods = "fdr", exact = F)
  cor_e[i-1] = m$estimate
  cor_p[i-1] = m$p.value
}

cor_test <- data.frame(Gene = colnames(samples_infect[,2:58308]),
                           cor_value = cor_e,
                           p_value = cor_p
                           )

cor_test <- na.omit(cor_test)

#cor_test
```

subseting genes of interest

```
gene_of_interest <- na.omit(cor_test[cor_test$cor_value > 0.7 & cor_test$p_value < 0.01 ,])
gene_of_interest
```

```
##                  Gene cor_value      p_value
## 14    ENSG00000001561 0.7035714 3.424337e-03
## 181   ENSG00000007350 0.7453940 1.424820e-03
## 196   ENSG00000007952 0.8357143 1.044469e-04
## 355   ENSG00000015133 0.7035714 3.424337e-03
## 631   ENSG00000047662 0.7035714 3.424337e-03
## 654   ENSG00000049167 0.7321429 1.912964e-03
## 772   ENSG00000057468 0.7107143 2.978995e-03
## 913   ENSG00000065060 0.7035714 3.424337e-03
## 923   ENSG00000065328 0.7464286 1.391411e-03
## 944   ENSG00000065675 0.7214286 2.398868e-03
## 1110  ENSG00000070388 0.7063350 3.246150e-03
## 1120  ENSG00000070601 0.7456984 1.414923e-03
## 1185  ENSG00000072201 0.7035714 3.424337e-03
## 1349  ENSG00000076555 0.7178571 2.581165e-03
## 1359  ENSG00000076864 0.7642857 9.073101e-04
## 1504  ENSG00000080644 0.8321429 1.190282e-04
## 1519  ENSG00000081019 0.7107143 2.978995e-03
## 1695  ENSG00000086475 0.7607143 9.911286e-04
## 1760  ENSG00000088035 0.7142857 2.774384e-03
## 1808  ENSG00000089022 0.7285714 2.065210e-03
## 1996  ENSG00000093072 0.7204347 2.448527e-03
## 2142  ENSG00000099954 0.7077479 3.157952e-03
## 2180  ENSG00000100077 0.7464286 1.391411e-03
## 2191  ENSG00000100104 0.7142857 2.774384e-03
## 2321  ENSG00000100473 0.7178571 2.581165e-03
## 2381  ENSG00000100749 0.7714286 7.568751e-04
## 2388  ENSG00000100811 0.7107143 2.978995e-03
## 2402  ENSG00000100890 0.7250000 2.227032e-03
## 2436  ENSG00000101082 0.7285714 2.065210e-03
## 2546  ENSG00000101489 0.8074062 2.729958e-04
## 2653  ENSG00000102245 0.7418247 1.545087e-03
## 2696  ENSG00000102743 0.7214286 2.398868e-03
## 2760  ENSG00000103089 0.7574736 1.072508e-03
## 2786  ENSG00000103245 0.7285714 2.065210e-03
## 2856  ENSG00000103832 0.7453940 1.424820e-03
## 2903  ENSG00000104327 0.7060977 3.261150e-03
## 3028  ENSG00000105053 0.7321429 1.912964e-03
## 3186  ENSG00000105737 0.8275249 1.403264e-04
## 3322  ENSG00000106537 0.7321429 1.912964e-03
## 3367  ENSG00000106809 0.7357143 1.769869e-03
## 3414  ENSG00000107362 0.7142857 2.774384e-03
## 3420  ENSG00000107447 0.7104413 2.995115e-03
## 3564  ENSG00000108622 0.7107143 2.978995e-03
## 3590  ENSG00000108813 0.7250398 2.225177e-03
## 3687  ENSG00000109667 0.7741985 7.042931e-04
## 3716  ENSG00000109906 0.7611935 9.795316e-04
## 3821  ENSG00000110876 0.7488832 1.314665e-03
## 3882  ENSG00000111325 0.7178571 2.581165e-03
## 3892  ENSG00000111364 0.7107143 2.978995e-03
## 4019  ENSG00000112195 0.7670300 8.468866e-04
## 4030  ENSG00000112242 0.7035714 3.424337e-03
## 4194  ENSG00000113532 0.7071429 3.195481e-03
```

```
## 4235  ENSG00000113838 0.7071429 3.195481e-03
## 4371  ENSG00000115155 0.7506211 1.262425e-03
## 4446  ENSG00000115526 0.7714286 7.568751e-04
## 4719  ENSG00000117481 0.7214286 2.398868e-03
## 4753  ENSG00000117697 0.7464286 1.391411e-03
## 4807  ENSG00000118412 0.7107143 2.978995e-03
## 4822  ENSG00000118513 0.7535714 1.177583e-03
## 4902  ENSG00000119411 0.8074062 2.729958e-04
## 4903  ENSG00000119414 0.7285714 2.065210e-03
## 5066  ENSG00000120555 0.7535714 1.177583e-03
## 5235  ENSG00000122121 0.7419408 1.541049e-03
## 5244  ENSG00000122188 0.8321429 1.190282e-04
## 5248  ENSG00000122223 0.7297593 2.013528e-03
## 5266  ENSG00000122435 0.7142857 2.774384e-03
## 5346  ENSG00000123124 0.7357143 1.769869e-03
## 5361  ENSG00000123201 0.7816483 5.774243e-04
## 5380  ENSG00000123388 0.7608598 9.875965e-04
## 5427  ENSG00000123836 0.7428571 1.509489e-03
## 5461  ENSG00000124143 0.7654379 8.815309e-04
## 5551  ENSG00000124575 0.7250000 2.227032e-03
## 5667  ENSG00000125514 0.7164821 2.654240e-03
## 5691  ENSG00000125686 0.7571429 1.081108e-03
## 5808  ENSG00000126259 0.8649663 3.144484e-05
## 6083  ENSG00000128908 0.7142857 2.774384e-03
## 6084  ENSG00000128915 0.7392857 1.635511e-03
## 6110  ENSG00000129159 0.7178571 2.581165e-03
## 6170  ENSG00000129675 0.7428571 1.509489e-03
## 6197  ENSG00000129993 0.8168010 2.020605e-04
## 6271  ENSG00000130475 0.7178571 2.581165e-03
## 6330  ENSG00000130734 0.7142857 2.774384e-03
## 6364  ENSG00000130921 0.7142857 2.774384e-03
## 6367  ENSG00000130940 0.7928571 4.220772e-04
## 6559  ENSG00000132286 0.7285714 2.065210e-03
## 6607  ENSG00000132518 0.7529233 1.195815e-03
## 6672  ENSG00000132846 0.7642857 9.073101e-04
## 6739  ENSG00000133246 0.7059878 3.268121e-03
## 6764  ENSG00000133477 0.7189481 2.524343e-03
## 6808  ENSG00000133895 0.7928571 4.220772e-04
## 6818  ENSG00000133997 0.7428571 1.509489e-03
## 6998  ENSG00000135002 0.7357143 1.769869e-03
## 7079  ENSG00000135426 0.7095624 3.047497e-03
## 7216  ENSG00000136104 0.7642857 9.073101e-04
## 7371  ENSG00000136895 0.7250000 2.227032e-03
## 7383  ENSG00000136936 0.7107143 2.978995e-03
## 7745  ENSG00000138795 0.8000000 3.422698e-04
## 7822  ENSG00000139372 0.7714286 7.568751e-04
## 7830  ENSG00000139445 0.7080300 3.140569e-03
## 7878  ENSG00000139737 0.7571429 1.081108e-03
## 7973  ENSG00000140470 0.7189481 2.524343e-03
## 7984  ENSG00000140519 0.7845738 5.330002e-04
## 8103  ENSG00000141425 0.7357143 1.769869e-03
## 8142  ENSG00000141568 0.7214286 2.398868e-03
## 8326  ENSG00000143157 0.7178571 2.581165e-03
## 8373  ENSG00000143363 0.7714286 7.568751e-04
```

```
## 8434  ENSG00000143590 0.7392857 1.635511e-03
## 8535  ENSG00000144214 0.7571429 1.081108e-03
## 8611  ENSG00000144736 0.7178571 2.581165e-03
## 8639  ENSG00000144893 0.7964286 3.804677e-04
## 8932  ENSG00000147138 0.7491088 1.307789e-03
## 9314  ENSG00000150627 0.7035714 3.424337e-03
## 9502  ENSG00000152192 0.8146801 2.165748e-04
## 9529  ENSG00000152382 0.7142857 2.774384e-03
## 9557  ENSG00000152580 0.7642857 9.073101e-04
## 9627  ENSG00000153147 0.7214286 2.398868e-03
## 9778  ENSG00000154537 0.7856175 5.178429e-04
## 9808  ENSG00000154783 0.7992883 3.496203e-04
## 9858  ENSG00000155275 0.7214286 2.398868e-03
## 9882  ENSG00000155545 0.7035714 3.424337e-03
## 9981  ENSG00000156413 0.7237560 2.285727e-03
## 10045 ENSG00000156973 0.7678571 8.293296e-04
## 10216 ENSG00000158445 0.7107143 2.978995e-03
## 10368 ENSG00000159592 0.7250000 2.227032e-03
## 10511 ENSG00000160654 0.7153443 2.715944e-03
## 10670 ENSG00000162006 0.7166453 2.645482e-03
## 10730 ENSG00000162419 0.7392857 1.635511e-03
## 10753 ENSG00000162521 0.7607143 9.911286e-04
## 10797 ENSG00000162676 0.7500000 1.280898e-03
## 10930 ENSG00000163141 0.7750000 6.896454e-04
## 10962 ENSG00000163281 0.7035714 3.424337e-03
## 11091 ENSG00000163655 0.7142857 2.774384e-03
## 11231 ENSG00000164076 0.7077479 3.157952e-03
## 11258 ENSG00000164120 0.7893509 4.664670e-04
## 11315 ENSG00000164300 0.7285714 2.065210e-03
## 11721 ENSG00000165934 0.7035714 3.424337e-03
## 12019 ENSG00000167094 0.7524579 1.209046e-03
## 12103 ENSG00000167491 0.7035714 3.424337e-03
## 12150 ENSG00000167642 0.7571429 1.081108e-03
## 12222 ENSG00000167880 0.7710365 7.645676e-04
## 12294 ENSG00000168135 0.7104413 2.995115e-03
## 12414 ENSG00000168675 0.7357143 1.769869e-03
## 12439 ENSG00000168795 0.7178571 2.581165e-03
## 12623 ENSG00000169570 0.7321429 1.912964e-03
## 12633 ENSG00000169609 0.7392857 1.635511e-03
## 12648 ENSG00000169684 0.7428571 1.509489e-03
## 12942 ENSG00000170946 0.7107143 2.978995e-03
## 13012 ENSG00000171219 0.8500000 5.995889e-05
## 13133 ENSG00000171757 0.7428571 1.509489e-03
## 13134 ENSG00000171759 0.7654379 8.815309e-04
## 13424 ENSG00000172977 0.7142857 2.774384e-03
## 13429 ENSG00000172995 0.7204347 2.448527e-03
## 13626 ENSG00000174010 0.7392857 1.635511e-03
## 13788 ENSG00000174944 0.7107143 2.978995e-03
## 13790 ENSG00000174946 0.7327974 1.886064e-03
## 13862 ENSG00000175305 0.7250000 2.227032e-03
## 13890 ENSG00000175463 0.7535714 1.177583e-03
## 13905 ENSG00000175536 0.7250000 2.227032e-03
## 14044 ENSG00000176244 0.7642857 9.073101e-04
## 14217 ENSG00000177082 0.7178571 2.581165e-03
```

```
## 14231 ENSG00000177144 0.7201190 2.464472e-03
## 14238 ENSG00000177181 0.7131370 2.838928e-03
## 14505 ENSG00000178538 0.7152771 2.719627e-03
## 14781 ENSG00000179988 0.7178571 2.581165e-03
## 14801 ENSG00000180096 0.7642857 9.073101e-04
## 14830 ENSG00000180245 0.7333631 1.863065e-03
## 14871 ENSG00000180448 0.7428571 1.509489e-03
## 14988 ENSG00000181038 0.7071429 3.195481e-03
## 15148 ENSG00000182010 0.7250000 2.227032e-03
## 15263 ENSG00000182512 0.7035714 3.424337e-03
## 15274 ENSG00000182557 0.7526930 1.202346e-03
## 15344 ENSG00000182866 0.7928571 4.220772e-04
## 15373 ENSG00000182973 0.7071429 3.195481e-03
## 15379 ENSG00000182993 0.7285714 2.065210e-03
## 15770 ENSG00000184574 0.7082027 3.129971e-03
## 15795 ENSG00000184677 0.7821429 5.697096e-04
## 15912 ENSG00000185155 0.7957040 3.886259e-04
## 16568 ENSG00000187862 0.8214286 1.731477e-04
## 16716 ENSG00000188452 0.7142857 2.774384e-03
## 16817 ENSG00000188822 0.7670300 8.468866e-04
## 16824 ENSG00000188848 0.7428571 1.509489e-03
## 17041 ENSG00000196247 0.7642857 9.073101e-04
## 17133 ENSG00000196502 0.7321429 1.912964e-03
## 17238 ENSG00000196839 0.7178571 2.581165e-03
## 17336 ENSG00000197153 0.7107143 2.978995e-03
## 17658 ENSG00000198134 0.7476353 1.353242e-03
## 18073 ENSG00000199266 0.7167116 2.641927e-03
## 18165 ENSG00000199471 0.7045141 3.362700e-03
## 18413 ENSG00000200051 0.7650672 8.897579e-04
## 19655 ENSG00000203761 0.7321429 1.912964e-03
## 19962 ENSG00000204624 0.7117406 2.919002e-03
## 20103 ENSG00000205018 0.7128451 2.855520e-03
## 20337 ENSG00000205930 0.7534460 1.181094e-03
## 20348 ENSG00000205981 0.7142857 2.774384e-03
## 21211 ENSG00000207625 0.7196083 2.490444e-03
## 21480 ENSG00000211451 0.7428571 1.509489e-03
## 21486 ENSG00000211460 0.7035714 3.424337e-03
## 21500 ENSG00000211580 0.7081368 3.134009e-03
## 21604 ENSG00000211751 0.7477781 1.348783e-03
## 21765 ENSG00000211955 0.7147084 2.750932e-03
## 21766 ENSG00000211956 0.7471093 1.369776e-03
## 22004 ENSG00000212510 0.7655318 8.794551e-04
## 22471 ENSG00000213641 0.7492148 1.304567e-03
## 22645 ENSG00000213976 0.7035714 3.424337e-03
## 22681 ENSG00000214046 0.7071429 3.195481e-03
## 22879 ENSG00000214584 0.7127715 2.859716e-03
## 22915 ENSG00000214700 0.7528517 1.197841e-03
## 23259 ENSG00000215784 0.7178571 2.581165e-03
## 23263 ENSG00000215795 0.7383560 1.669669e-03
## 23309 ENSG00000215915 0.7500000 1.280898e-03
## 23699 ENSG00000219284 0.7016514 3.552662e-03
## 23873 ENSG00000220721 0.7708779 7.676991e-04
## 24253 ENSG00000222222 0.7378648 1.687946e-03
## 24393 ENSG00000222585 0.7092081 3.068815e-03
```

```
## 25214 ENSG00000224177 0.7016514 3.552662e-03
## 25277 ENSG00000224277 0.7237560 2.285727e-03
## 25353 ENSG00000224383 0.7392857 1.635511e-03
## 25388 ENSG00000224429 0.7500000 1.280898e-03
## 25673 ENSG00000224830 0.7562763 1.103908e-03
## 25674 ENSG00000224831 0.7392857 1.635511e-03
## 25902 ENSG00000225171 0.8203757 1.794073e-04
## 25903 ENSG00000225172 0.7419408 1.541049e-03
## 26301 ENSG00000225756 0.7528517 1.197841e-03
## 27128 ENSG00000226937 0.7107143 2.978995e-03
## 27606 ENSG00000227620 0.7117406 2.919002e-03
## 27634 ENSG00000227673 0.7164821 2.654240e-03
## 27736 ENSG00000227834 0.7670041 8.474412e-04
## 27779 ENSG00000227896 0.7821429 5.697096e-04
## 28084 ENSG00000228343 0.7714286 7.568751e-04
## 28182 ENSG00000228487 0.7240190 2.273217e-03
## 28277 ENSG00000228630 0.7379423 1.685052e-03
## 28446 ENSG00000228878 0.7285714 2.065210e-03
## 28480 ENSG00000228941 0.8583325 4.223457e-05
## 29019 ENSG00000229739 0.7095624 3.047497e-03
## 29052 ENSG00000229785 0.7418247 1.545087e-03
## 29195 ENSG00000229989 0.7714286 7.568751e-04
## 29556 ENSG00000230489 0.8212367 1.742746e-04
## 29580 ENSG00000230524 0.8612962 3.708798e-05
## 29753 ENSG00000230769 0.7285714 2.065210e-03
## 29766 ENSG00000230790 0.8231736 1.631688e-04
## 29794 ENSG00000230830 0.8425424 8.065039e-05
## 29805 ENSG00000230841 0.7425258 1.520841e-03
## 30191 ENSG00000231419 0.7211126 2.414567e-03
## 30262 ENSG00000231527 0.7784100 6.300639e-04
## 30599 ENSG00000232021 0.7535714 1.177583e-03
## 31744 ENSG00000233672 0.7464286 1.391411e-03
## 31978 ENSG00000234005 0.7655318 8.794551e-04
## 32704 ENSG00000235066 0.7321429 1.912964e-03
## 32997 ENSG00000235492 0.7088977 3.087592e-03
## 33093 ENSG00000235635 0.7107143 2.978995e-03
## 33196 ENSG00000235795 0.7285714 2.065210e-03
## 33731 ENSG00000236564 0.8432740 7.839102e-05
## 33783 ENSG00000236654 0.7077479 3.157952e-03
## 33819 ENSG00000236703 0.7204347 2.448527e-03
## 34438 ENSG00000237594 0.7243537 2.257372e-03
## 34597 ENSG00000237828 0.8714286 2.323648e-05
## 34677 ENSG00000237943 0.7500000 1.280898e-03
## 35081 ENSG00000238880 0.7171824 2.616817e-03
## 35105 ENSG00000238959 0.7132662 2.831608e-03
## 35267 ENSG00000239415 0.7035714 3.424337e-03
## 35404 ENSG00000239732 0.7118912 2.910279e-03
## 36043 ENSG00000241293 0.7607143 9.911286e-04
## 36137 ENSG00000241537 0.7142857 2.774384e-03
## 36306 ENSG00000241954 0.7399467 1.611572e-03
## 36312 ENSG00000241964 0.7144087 2.767545e-03
## 36621 ENSG00000242692 0.7471093 1.369776e-03
## 36767 ENSG00000243051 0.7378648 1.687946e-03
## 36777 ENSG00000243069 0.7030871 3.456352e-03
```

```
## 37281 ENSG00000244267 0.8093925 2.565160e-04
## 37446 ENSG00000244671 0.7728148 7.301769e-04
## 37564 ENSG00000245954 0.7205190 2.444284e-03
## 37651 ENSG00000247137 0.7250000 2.227032e-03
## 37822 ENSG00000248278 0.7678571 8.293296e-04
## 38385 ENSG00000249142 0.7311875 1.952776e-03
## 39054 ENSG00000250197 0.7132662 2.831608e-03
## 39367 ENSG00000250651 0.7369668 1.721773e-03
## 40044 ENSG00000251639 0.7746102 6.967375e-04
## 40051 ENSG00000251652 0.7144087 2.767545e-03
## 40366 ENSG00000252112 0.7205190 2.444284e-03
## 41761 ENSG00000253943 0.7045141 3.362700e-03
## 41818 ENSG00000254008 0.7534445 1.181136e-03
## 42644 ENSG00000254978 0.8464286 6.923824e-05
## 42699 ENSG00000255040 0.7027312 3.480029e-03
## 42772 ENSG00000255126 0.7492148 1.304567e-03
## 43086 ENSG00000255468 0.7535714 1.177583e-03
## 43187 ENSG00000255581 0.7165349 2.651403e-03
## 43212 ENSG00000255650 0.8243781 1.565607e-04
## 43333 ENSG00000255964 0.7096820 3.040328e-03
## 43403 ENSG00000256098 0.7351650 1.791299e-03
## 44488 ENSG00000258393 0.7104413 2.995115e-03
## 44752 ENSG00000258699 0.7050951 3.325156e-03
## 44910 ENSG00000258875 0.7750000 6.896454e-04
## 45012 ENSG00000258993 0.7025135 3.494579e-03
## 45211 ENSG00000259216 0.7464286 1.391411e-03
## 45243 ENSG00000259254 0.7500000 1.280898e-03
## 45388 ENSG00000259425 0.7867000 5.024935e-04
## 45471 ENSG00000259521 0.7107143 2.978995e-03
## 45758 ENSG00000259881 0.7492148 1.304567e-03
## 45815 ENSG00000259961 0.7196083 2.490444e-03
## 45941 ENSG00000260128 0.7776192 6.434938e-04
## 46552 ENSG00000260966 0.7428571 1.509489e-03
## 46745 ENSG00000261226 0.7459762 1.405940e-03
## 46979 ENSG00000261546 0.7107143 2.978995e-03
## 47144 ENSG00000261762 0.7285714 2.065210e-03
## 47440 ENSG00000262823 0.7679817 8.267111e-04
## 47792 ENSG00000263818 0.7117406 2.919002e-03
## 48053 ENSG00000264354 0.7132662 2.831608e-03
## 48241 ENSG00000264757 0.7792756 6.156261e-04
## 48263 ENSG00000264808 0.7785714 6.273501e-04
## 48655 ENSG00000265692 0.7637626 9.192143e-04
## 49024 ENSG00000266490 0.7214286 2.398868e-03
## 49395 ENSG00000267136 0.7513818 1.240090e-03
## 49585 ENSG00000267366 0.7764409 6.639336e-04
## 49690 ENSG00000267498 0.8004166 3.380266e-04
## 49693 ENSG00000267503 0.7303356 1.988837e-03
## 49715 ENSG00000267529 0.7957040 3.886259e-04
## 49848 ENSG00000267694 0.7099387 3.024980e-03
## 49854 ENSG00000267702 0.7149243 2.739012e-03
## 50045 ENSG00000268170 0.7237560 2.285727e-03
## 50104 ENSG00000268357 0.7214286 2.398868e-03
## 50146 ENSG00000268510 0.7534460 1.181094e-03
## 50232 ENSG00000268745 0.7107143 2.978995e-03
```

```
## 50568 ENSG00000269802 0.7883098 4.803560e-04
## 50573 ENSG00000269814 0.7108346 2.971910e-03
## 50581 ENSG00000269837 0.7042005 3.383109e-03
## 50733 ENSG00000270096 0.7250000 2.227032e-03
## 50748 ENSG00000270117 0.7607143 9.911286e-04
## 50758 ENSG00000270135 0.7464286 1.391411e-03
## 50993 ENSG00000270542 0.7171824 2.616817e-03
## 51130 ENSG00000270767 0.7016514 3.552662e-03
## 51155 ENSG00000270806 0.7941204 4.069550e-04
## 51286 ENSG00000271005 0.7050951 3.325156e-03
## 51703 ENSG00000271656 0.7516928 1.231054e-03
## 51784 ENSG00000271803 0.7035714 3.424337e-03
## 51890 ENSG00000271993 0.7838156 5.442369e-04
## 52077 ENSG00000272343 0.8192999 1.859937e-04
## 52315 ENSG00000272746 0.7604525 9.975112e-04
## 52411 ENSG00000272910 0.7357143 1.769869e-03
## 52425 ENSG00000272931 0.7225518 2.343717e-03
## 52428 ENSG00000272936 0.7357143 1.769869e-03
## 52500 ENSG00000273058 0.7202863 2.456016e-03
## 52898 ENSG00000273783 0.7703310 7.785728e-04
## 53132 ENSG00000274253 0.7164821 2.654240e-03
## 53201 ENSG00000274372 0.7192683 2.507854e-03
## 53345 ENSG00000274641 0.7357143 1.769869e-03
## 53417 ENSG00000274776 0.8285714 1.352448e-04
## 53519 ENSG00000274979 0.7709081 7.671015e-04
## 53662 ENSG00000275239 0.7310299 1.959405e-03
## 53829 ENSG00000275580 0.7107143 2.978995e-03
## 54076 ENSG00000276063 0.7471093 1.369776e-03
## 54821 ENSG00000277527 0.7185821 2.543291e-03
## 54895 ENSG00000277655 0.7010660 3.592539e-03
## 54909 ENSG00000277684 0.7303340 1.988903e-03
## 55426 ENSG00000278665 0.7727524 7.313614e-04
## 55458 ENSG00000278725 0.7844360 5.350278e-04
## 55599 ENSG00000278937 0.7346669 1.810910e-03
## 56194 ENSG00000279667 0.7155798 2.703082e-03
## 56341 ENSG00000279838 0.7607143 9.911286e-04
## 56420 ENSG00000279949 0.7739427 7.090208e-04
## 56622 ENSG00000280194 0.7357143 1.769869e-03
## 56688 ENSG00000280277 0.7566879 1.093031e-03
## 57045 ENSG00000281852 0.7332989 1.865664e-03
## 57082 ENSG00000282080 0.7356708 1.771556e-03
## 57104 ENSG00000282277 0.7058578 3.276374e-03
## 57494 ENSG00000283414 0.7204347 2.448527e-03
## 57886 ENSG00000284070 0.7204347 2.448527e-03
## 58127 ENSG00000284543 0.7598615 1.012041e-03
## 58177 ENSG00000284621 0.8315466 1.216184e-04
## 58294 ENSG00000284740 0.7828421 5.589454e-04
```

correlation test

```
i <- c()
cor_e <- c()
cor_p <- c()

for (i in 2:58308)
{
  m = cor.test(samples_infect[,1],as.numeric(samples_infect[,i]),method = 'pearson', use = 'pair
wise.complete.obs',p.adjust.methods = "fdr", exact = F)
  cor_e[i-1] = m$estimate
  cor_p[i-1] = m$p.value
}

cor_test <- data.frame(Gene = colnames(samples_infect[,2:58308]),
                           cor_value = cor_e,
                           p_value = cor_p
                           )

#cor_test
```

subseting genes of interest

```
gene_of_interest <- na.omit(cor_test[cor_test$cor_value > 0.9 & cor_test$p_value < 0.01 ,])
gene_of_interest
```

```
##                  Gene cor_value      p_value
## 9778  ENSG00000154537 0.9254256 7.705474e-07
## 10269 ENSG00000158813 0.9019888 4.296749e-06
## 11315 ENSG00000164300 0.9194246 1.255658e-06
## 13429 ENSG00000172995 0.9005775 4.698709e-06
## 29794 ENSG00000230830 0.9046848 3.608298e-06
## 36137 ENSG00000241537 0.9190814 1.289741e-06
## 46552 ENSG00000260966 0.9006792 4.668736e-06
## 47792 ENSG00000263818 0.9107026 2.396988e-06
## 50568 ENSG00000269802 0.9172384 1.486291e-06
## 51770 ENSG00000271776 0.9030633 4.010310e-06
```

SERINC5

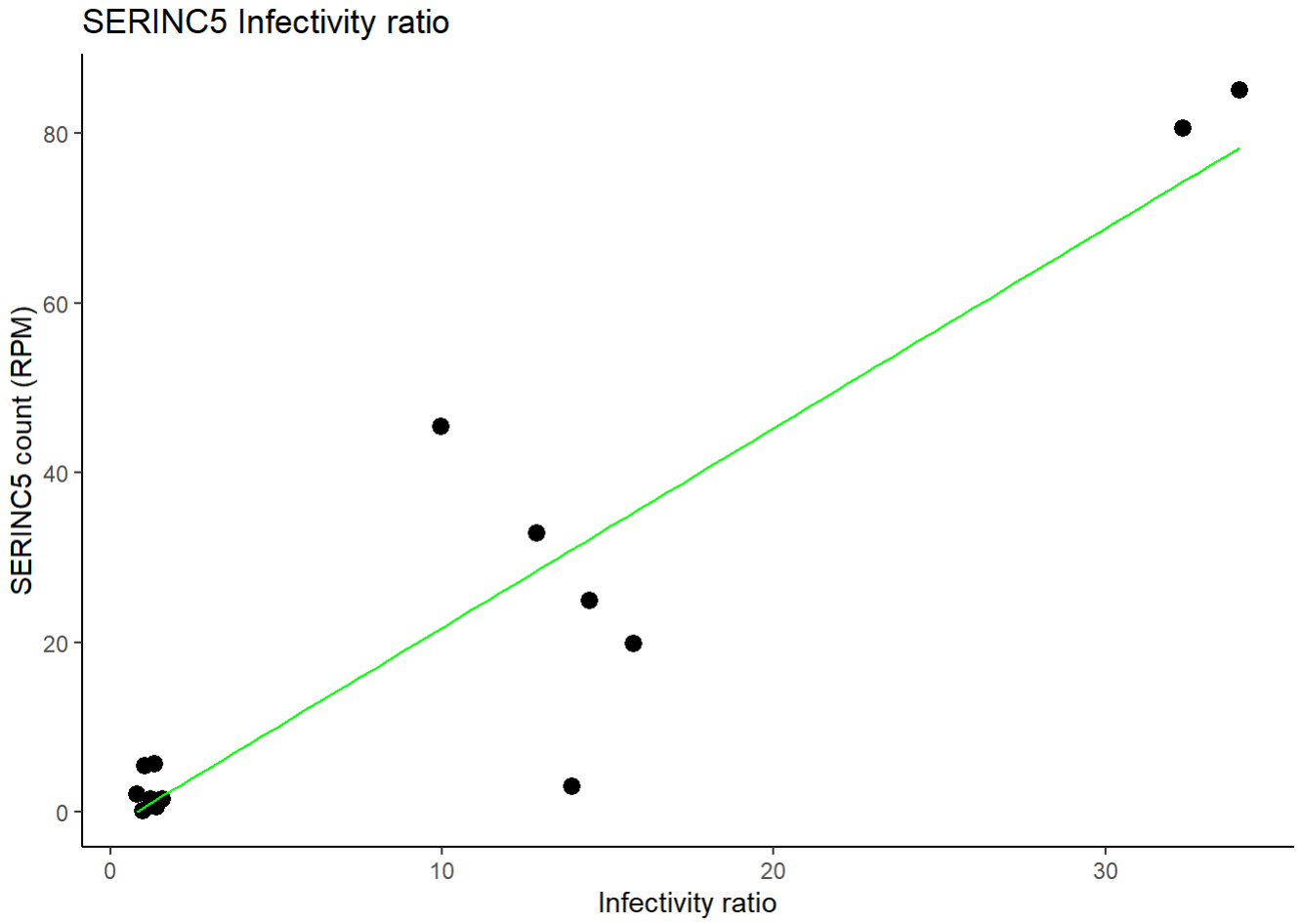
To recreate the results from the paper, we need to look at the correlation value SERINC5
Ensemble ID of **SERINC5** is ENSG00000164300 and from the table above, we can see that it has a correlation value of 0.91.

Now we replicate he Figure 1-d from the paper…

```
Serinc5 <- samples_infect[, colnames(samples_infect) == c("infect.infectivity_ratio", "ENSG00000
164300")]
```

```
ggplot(data = Serinc5, aes(infect.infectivity_ratio, as.numeric(ENSG00000164300), size= 1.0))+
geom_point(pch=20)+
geom_smooth(method = "lm", se=F, col="green", size=0.5)+
xlab("Infectivity ratio")+
ylab("SERINC5 count (RPM)")+
ggtitle("SERINC5 Infectivity ratio")+
geom_smooth(method = "lm", se=F, col="green", size=0.5)+
theme_classic()+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

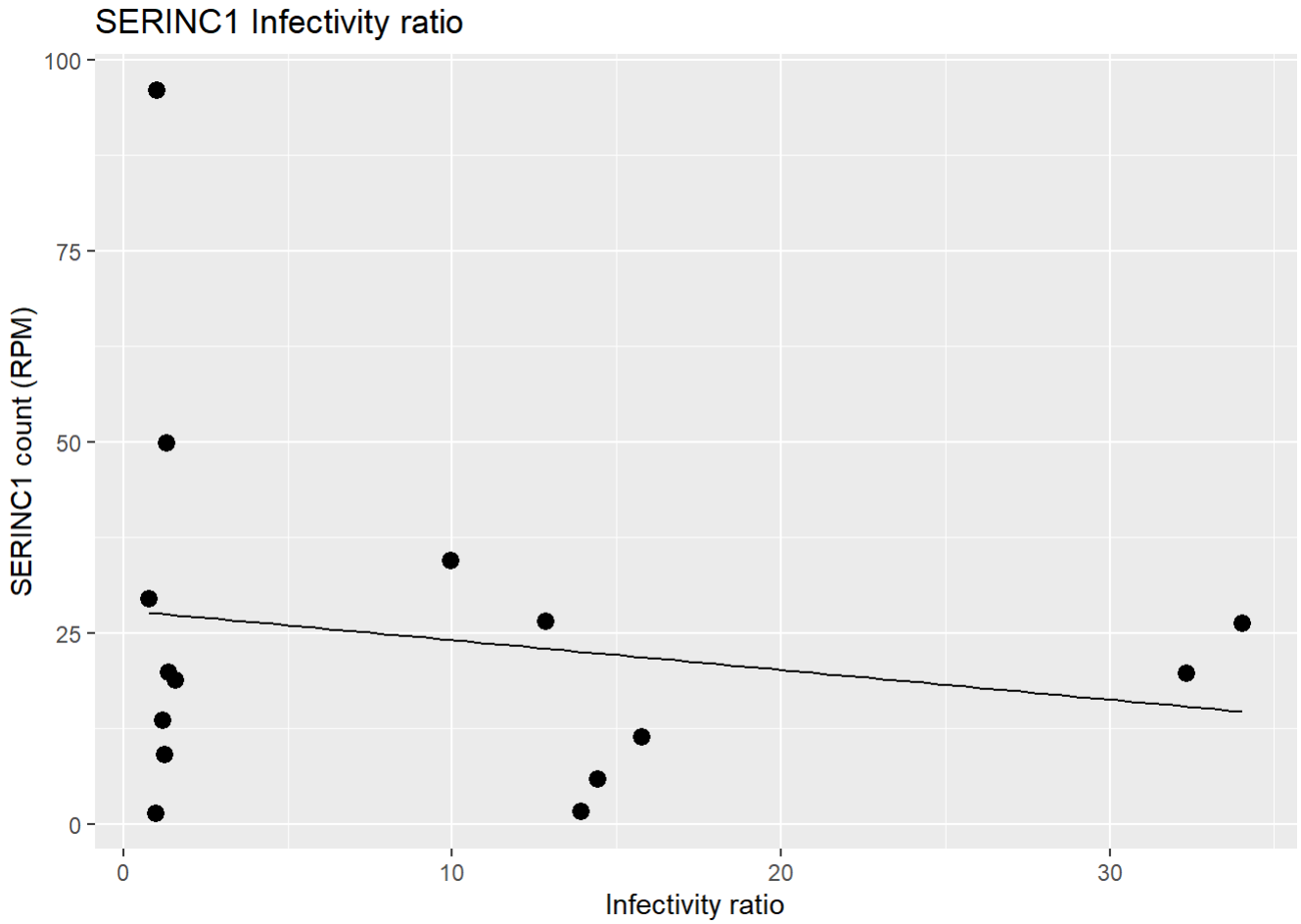


We can also plot the infectivity ratio vs. RPM graph for all the SERNIC genes

SERINC1 ENSG00000111897
SERINC2 ENSG00000168528
SERINC3 ENSG00000132824
SERINC4 ENSG00000184716
SERINC5 ENSG00000164300

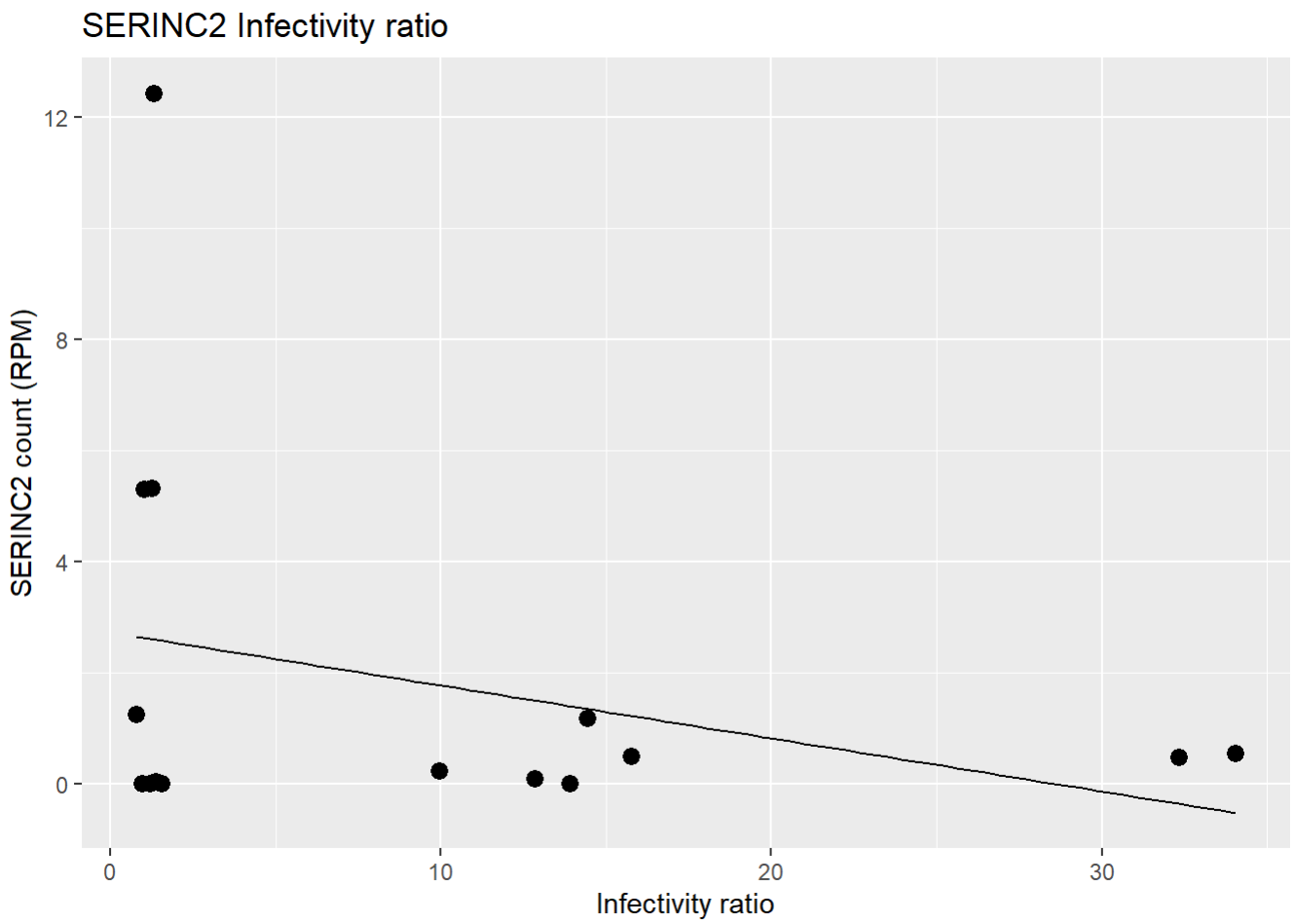
```
ggplot(data = samples_infect, aes(infect.infectivity_ratio, as.numeric(ENSG00000111897), size=
1.0))+
geom_point(pch=20, col="black")+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("SERINC1 count (RPM)")+
ggtitle("SERINC1 Infectivity ratio")+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



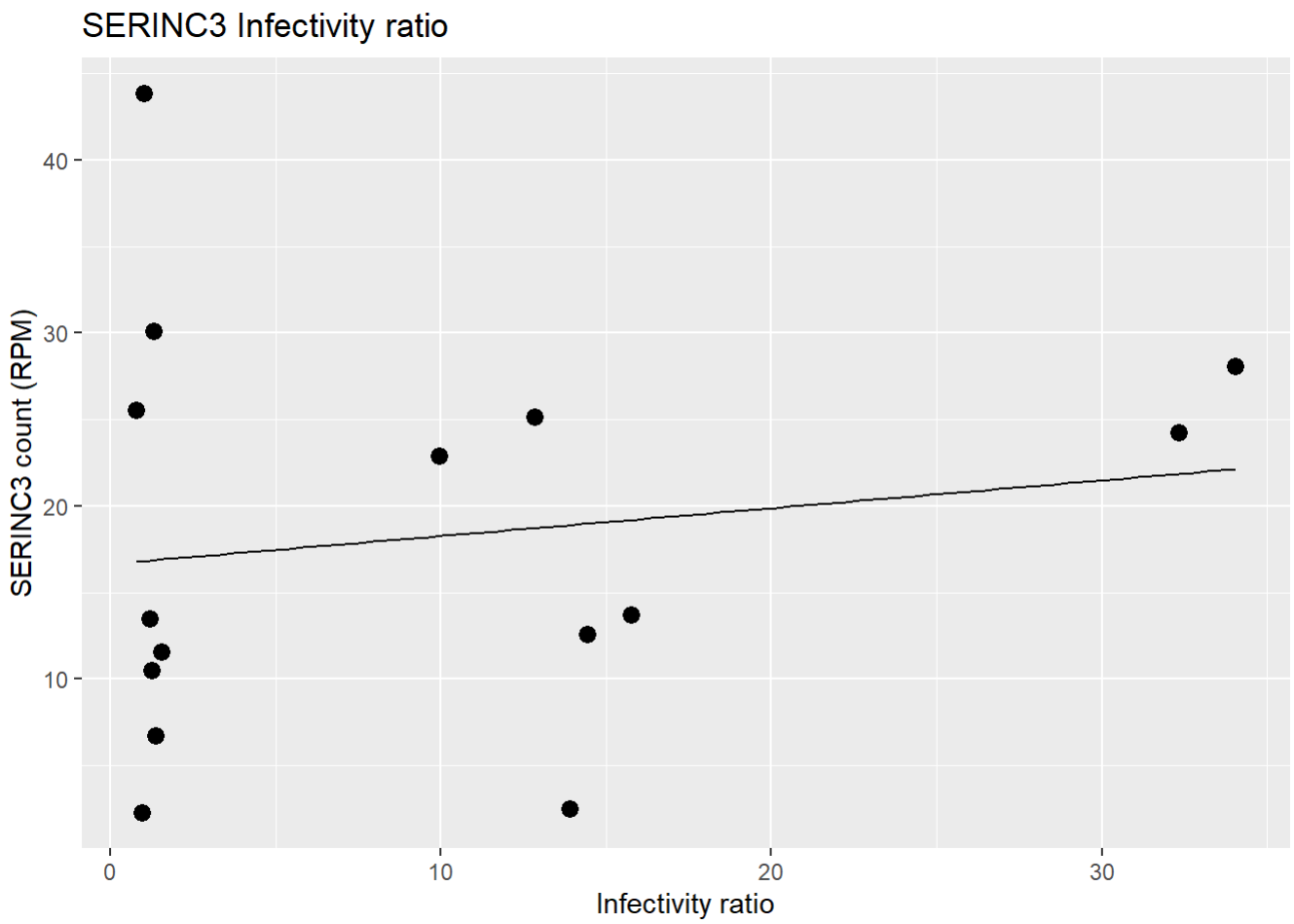

```
ggplot(data = samples_infect, aes(infect.infectivity_ratio, as.numeric(ENSG00000168528), size=
1.0))+
geom_point(pch=20, col="black")+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("SERINC2 count (RPM)")+
ggtitle("SERINC2 Infectivity ratio")+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



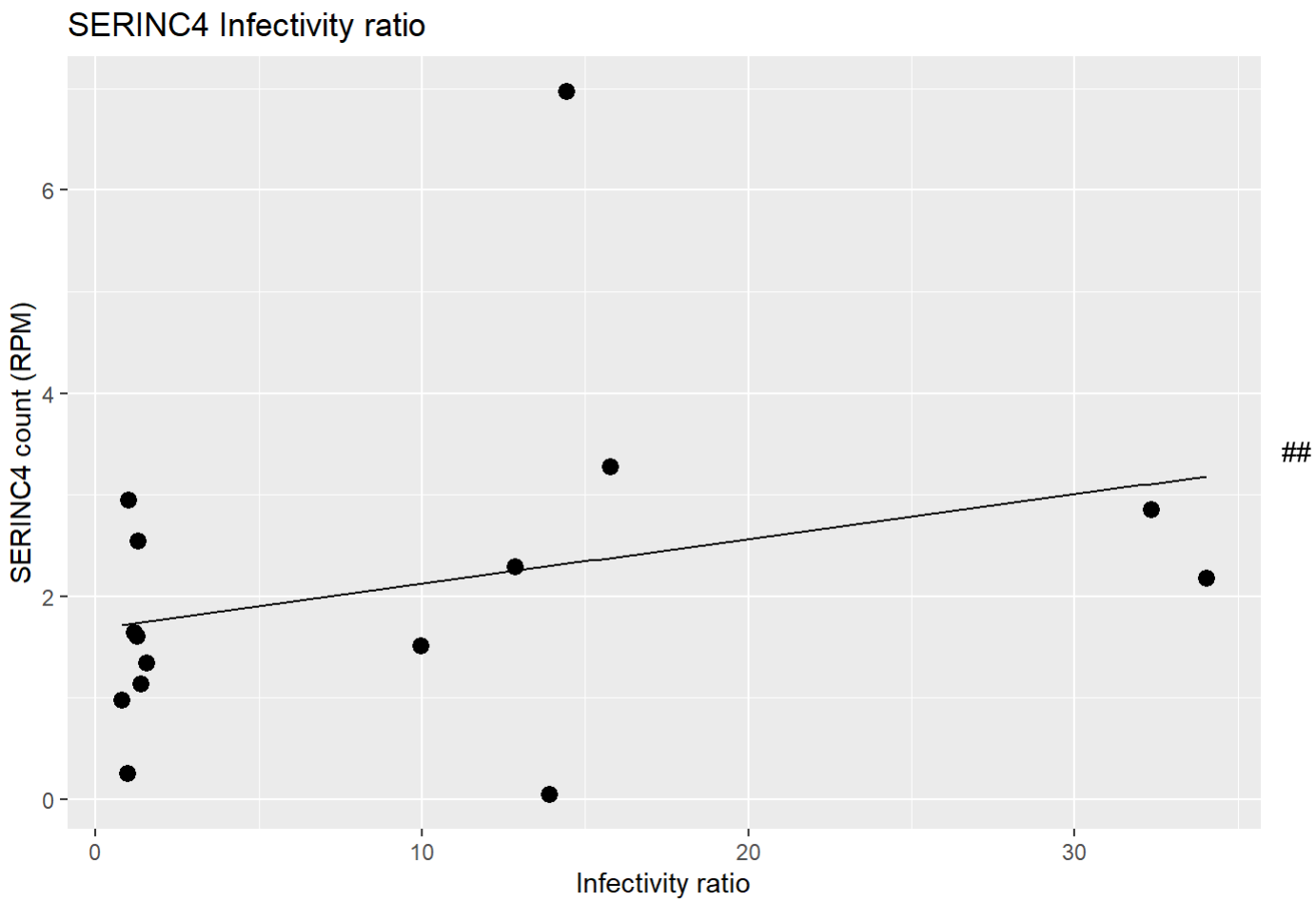

```
ggplot(data = samples_infect, aes(infect.infectivity_ratio, as.numeric(ENSG00000132824), size=
1.0))+
geom_point(pch=20, col="black")+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("SERINC3 count (RPM)")+
ggtitle("SERINC3 Infectivity ratio")+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(data = samples_infect, aes(infect.infectivity_ratio, as.numeric(ENSG00000184716), size=
1.0))+
geom_point(pch=20, col="black")+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("SERINC4 count (RPM)")+
ggtitle("SERINC4 Infectivity ratio")+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



##

Next we find the genes with positive correlation as well as the genes with negative correlation

```
poscor <- cor_test[cor_test$cor_value >= 0.9 & cor_test$p_value <= 0.01,] # For positively corre
lated genes
posgenenames<- poscor$Gene
posgenes <- samples_infect[, posgenenames]
```

```
## Error in `[.data.frame`(samples_infect, , posgenenames): undefined columns selected
```

```
posgenes <- t(posgenes)
```

```
## Error in t(posgenes): object 'posgenes' not found
```

```
negcor <- cor_test[cor_test$cor_value <= -0.6 & cor_test$p_value <= 0.05,] #For negatively corre
lated genes
neggenenames<- negcor$Gene
neggenes <- samples_infect[, neggenenames]
```

```
## Error in `[.data.frame`(samples_infect, , neggenenames): undefined columns selected
```

```
neggenes <- t(neggenes)
```

```
## Error in t(neggenes): object 'neggenes' not found
```

```
head(poscor, n=11L)
```

```
##       Gene cor_value p_value
## NA    <NA>        NA      NA
## NA.1  <NA>        NA      NA
## NA.2  <NA>        NA      NA
## NA.3  <NA>        NA      NA
## NA.4  <NA>        NA      NA
## NA.5  <NA>        NA      NA
## NA.6  <NA>        NA      NA
## NA.7  <NA>        NA      NA
## NA.8  <NA>        NA      NA
## NA.9  <NA>        NA      NA
## NA.10 <NA>        NA      NA
```

```
head(negcor, n=12L)
```

```
##       Gene cor_value p_value
## NA    <NA>        NA      NA
## NA.1  <NA>        NA      NA
## NA.2  <NA>        NA      NA
## NA.3  <NA>        NA      NA
## NA.4  <NA>        NA      NA
## NA.5  <NA>        NA      NA
## NA.6  <NA>        NA      NA
## NA.7  <NA>        NA      NA
## NA.8  <NA>        NA      NA
## NA.9  <NA>        NA      NA
## NA.10 <NA>        NA      NA
## NA.11 <NA>        NA      NA
```

Our Hypothesis

The following genes might be able to block HIV as well… **SAMHD1 (ENSG00000101347): APOBEC3G (ENSG00000239713): MX2 (ENSG00000183486): TRIM5 (ENSG00000132256): Tethering (ENSG00000130303):** Blocs HIV Budding

Generating the data

```
SAMHD1 <- cor_test[cor_test$Gene=="ENSG00000101347",]
SAMHD1$cor_value
```

```
## [1] -0.4216684
```

```
APOBEC3G <- cor_test[cor_test$Gene=="ENSG00000239713",]
APOBEC3G$cor_value
```

```
## [1] -0.1196492
```

```
MX2 <- cor_test[cor_test$Gene=="ENSG00000183486",]
MX2$cor_value
```

```
## [1] 0.5551704
```

```
TRIM5 <- cor_test[cor_test$Gene=="ENSG00000132256",]
TRIM5$cor_value
```

```
## [1] -0.1989562
```

```
TETHERIN <- cor_test[cor_test$Gene=="ENSG00000130303",]
TETHERIN$cor_value
```

```
## [1] -0.2239895
```

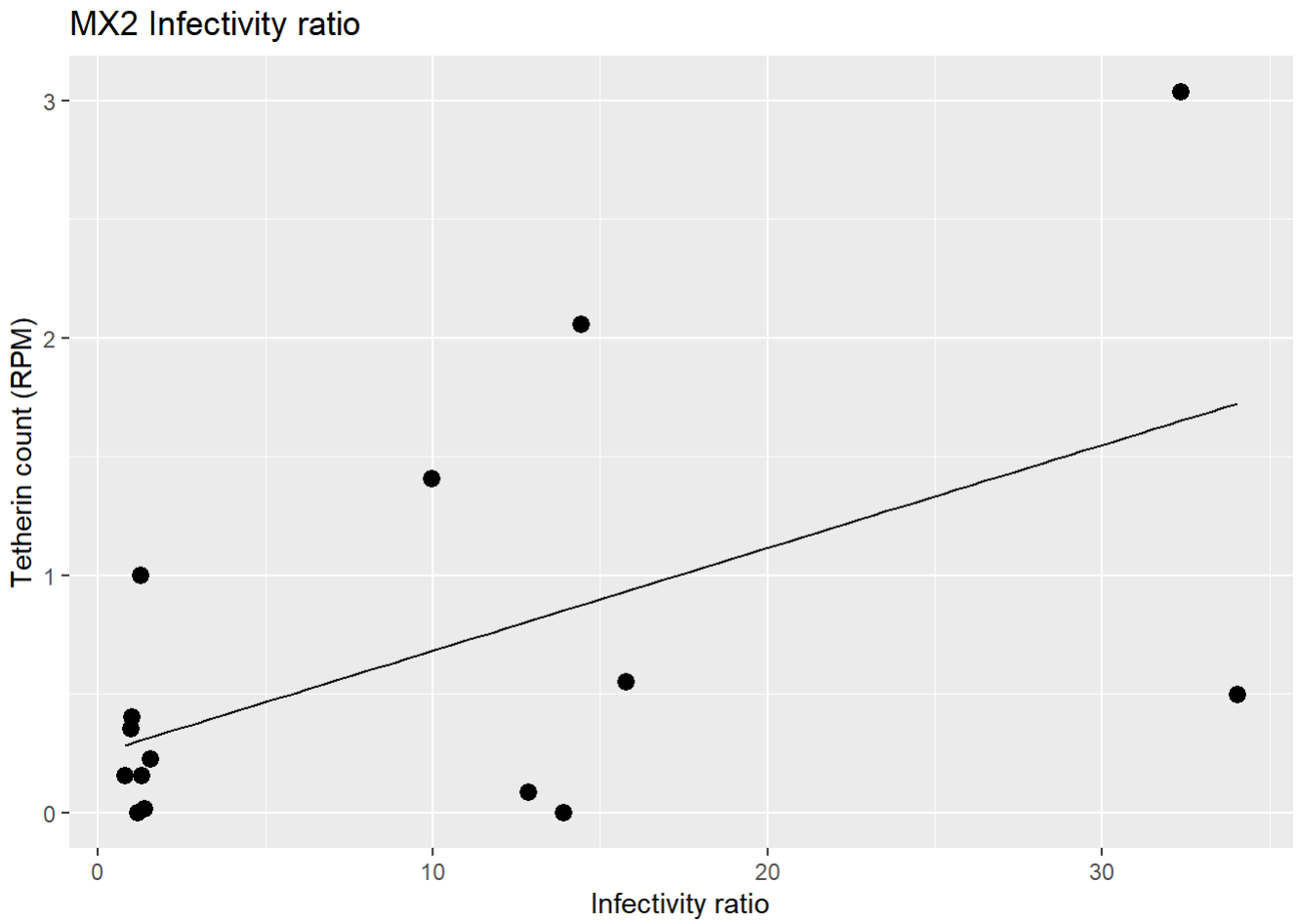
We only find significant correlation with MX2 so we plot its graph…

```
MX2 <- samples_infect[, colnames(samples_infect) == c("infect.infectivity_ratio", "ENSG000001834
86")]
```

Now Plotting

```
ggplot(data = MX2, aes(infect.infectivity_ratio, as.numeric(ENSG00000183486), size= 1.0))+
geom_point(pch=20)+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("Tetherin count (RPM)")+
ggtitle("MX2 Infectivity ratio")+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Upregulatd genes

```
head(poscor)
```

```
##      Gene cor_value p_value
## NA   <NA>        NA      NA
## NA.1 <NA>        NA      NA
## NA.2 <NA>        NA      NA
## NA.3 <NA>        NA      NA
## NA.4 <NA>        NA      NA
## NA.5 <NA>        NA      NA
```

Among the above genes we find the following genes to be of most importance during HIV attack

ENSG00000158813 EDA
ENSG00000269802 AC011491.3

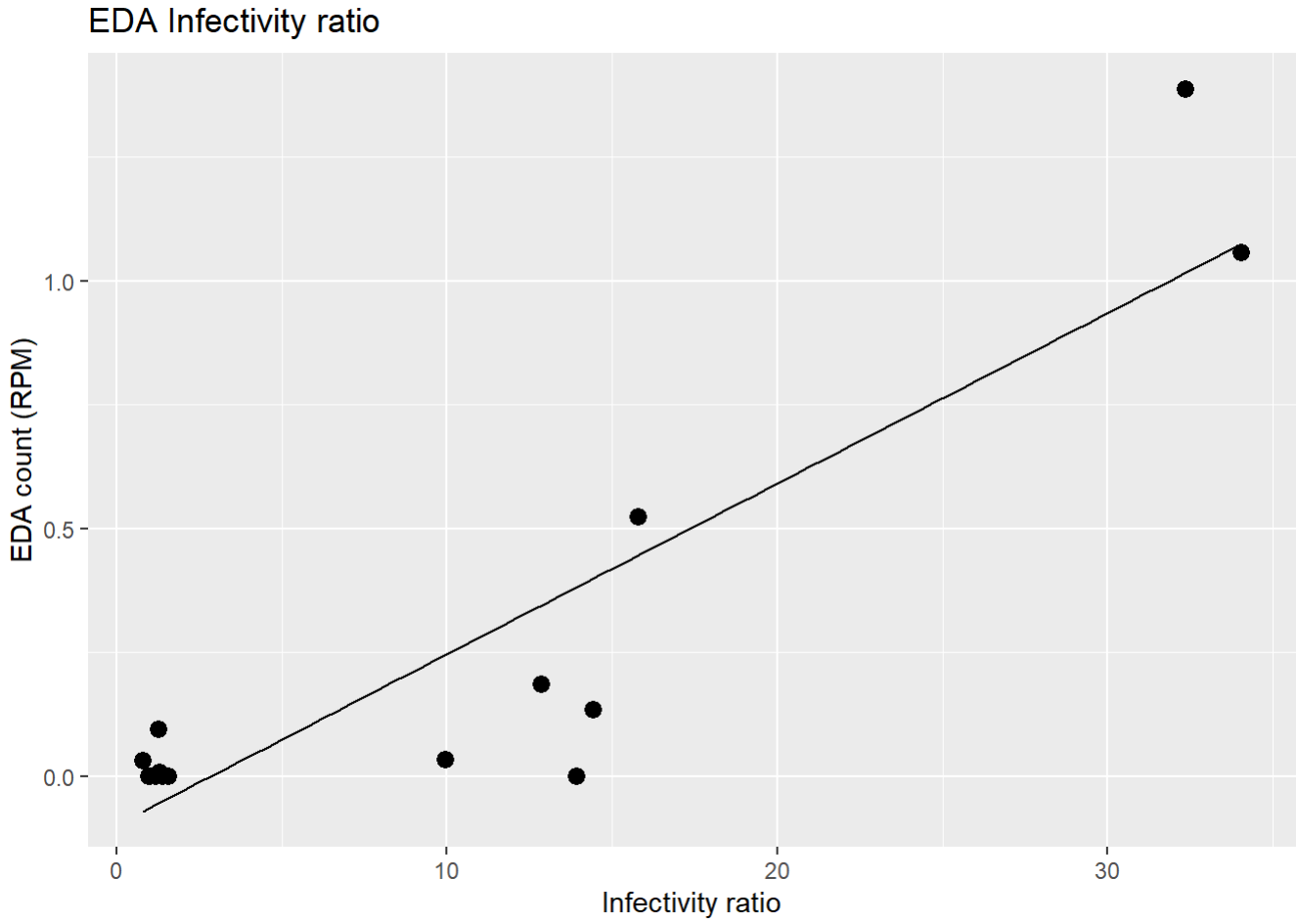
EDA [ENSG00000158813]

About EDA: The protein encoded by this gene is a type II membrane protein that can be cleaved by furin to produce a secreted form.
Hypothesis: Upon infection with HIV, envelope proteins are attached to cell membrane so the fully packaged HIV with capcid can go and attached to cell membrane and leave the cell to infect new cell. So it might be upregulating this gene in order to increase it's infectivity.

```
EDA <- samples_infect[, colnames(samples_infect) == c("infect.infectivity_ratio", "ENSG000001588
13")]
```

```
ggplot(data = EDA, aes(infect.infectivity_ratio, as.numeric(ENSG00000158813), size= 1.0))+
geom_point(pch=20)+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("EDA count (RPM)")+
ggtitle("EDA Infectivity ratio")+
theme(legend.position = "none")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Down regulated genes

```
head(negcor)
```

```
##      Gene cor_value p_value
## NA   <NA>        NA      NA
## NA.1 <NA>        NA      NA
## NA.2 <NA>        NA      NA
## NA.3 <NA>        NA      NA
## NA.4 <NA>        NA      NA
## NA.5 <NA>        NA      NA
```

Among above negatively correlated genes we found PLXNB2 Gene (ENSG00000196576) most interesting.

```
PLXNB2 <- samples_infect[, colnames(samples_infect) == c("infect.infectivity_ratio", "ENSG000001
96576")]
```

Now Plotting

```
ggplot(data = PLXNB2, aes(infect.infectivity_ratio, as.numeric(ENSG00000196576), size= 1.0))+
geom_point(pch=20)+
geom_smooth(method = "lm", se=F, col="black", size=0.5)+
xlab("Infectivity ratio")+
ylab("PLXNB2 count (RPM)")+
ggtitle("PLXNB2 Infectivity ratio")+
theme(legend.position = "none")
```