data_preparation

2022-08-15

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(edgeR)
```

```
## Loading required package: limma
```

```
library(pheatmap)
library(RColorBrewer)
library(EnhancedVolcano)
```

```
## Loading required package: ggrepel
```

```
alldata <- read_csv('data/CellLine_alldata.csv', col_types = cols(
  Gene = col_character(),
  PreferName = col_character(),
  Accession = col_character(),
  IPAS = col_character(),
  SampleDescription = col_character(),
  ExperimentType = col_character(),
  Disease = col_character(),
  SubType = col_character(),
  MW = col_double(),
  TCETotalPep = col_double(),
  MediaTotalPep = col_double(),
  MediaNRatio = col_double(),
  SurfaceTotalPep = col_double(),
  SurfaceNRatio = col_double(),
  NuclearTotalPep = col_double(),
  NuclearNRatio = col_double(),
  TCENumIndisPro = col_double(),
  MediaNumIndisPro = col_double(),
  SurfaceNumIndisPro = col_double(),
  NuclearNumIndisPro = col_double())
  )
```

```
head(alldata)
```

```
## # A tibble: 6 x 20
##   Gene       Prefe~1 Acces~2 IPAS  Sampl~3 Exper~4 Disease SubType    MW TCETo~5
##   <chr>      <chr>   <chr>   <chr> <chr>   <chr>   <chr>   <chr>   <dbl>   <dbl>
## 1 FLJ21687,~ FLJ216~ A0A024~ IP04~ HSAEC1~ CELLLI~ LungAd~ N/A     29420       0
## 2 FLJ21687,~ FLJ216~ A0A024~ IP05~ CAMA1(~ CELLLI~ Breast  Lumina~ 29420       0
## 3 FLJ21687,~ FLJ216~ A0A024~ IP05~ TF-1-#~ CELLLI~ Leukem~ AML     29420       5
## 4 FLJ21687,~ FLJ216~ A0A024~ IP07~ TXOV13~ CELLLI~ Ovarian Xeno    29420       2
## 5 FLJ21687,~ FLJ216~ A0A024~ IP09~ AGS     CELLLI~ Gastric Adeno   29420       5
## 6 FLJ21687,~ FLJ216~ A0A024~ IP17~ H82     CELLLI~ SCLC    NEUROD1 29420       0
## # ... with 10 more variables: MediaTotalPep <dbl>, MediaNRatio <dbl>,
## #   SurfaceTotalPep <dbl>, SurfaceNRatio <dbl>, NuclearTotalPep <dbl>,
## #   NuclearNRatio <dbl>, TCENumIndisPro <dbl>, MediaNumIndisPro <dbl>,
## #   SurfaceNumIndisPro <dbl>, NuclearNumIndisPro <dbl>, and abbreviated
## #   variable names 1: PreferName, 2: Accession, 3: SampleDescription,
## #   4: ExperimentType, 5: TCETotalPep
## # i Use `colnames()` to see all variable names
```

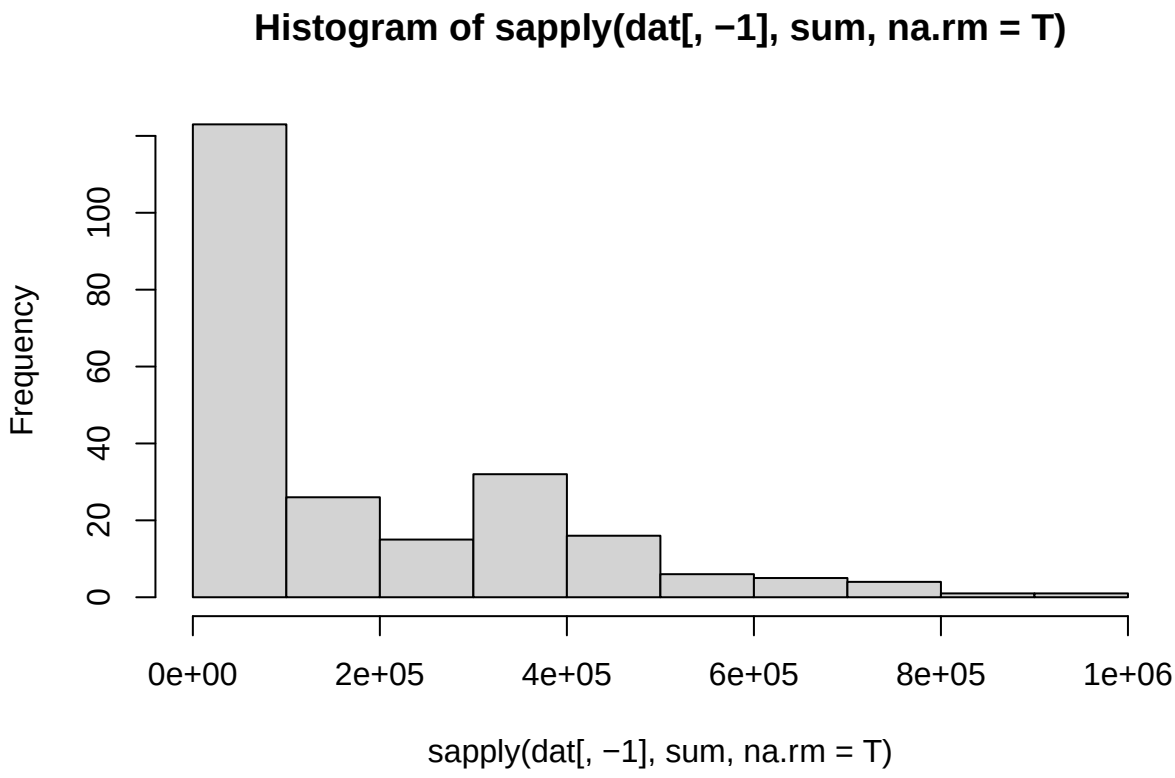
```
dat <- alldata %>%
  select(c('Accession', 'IPAS', 'TCETotalPep')) %>%
   pivot_wider('Accession', names_from='IPAS', values_from = 'TCETotalPep')
  # only use the primary accession (? will have duplicate row)

dat$Accession  <- sapply(strsplit(dat$Accession, ','), '[[', 1)

dat <- dat %>%
  group_by(Accession) %>%
  summarise_all(.funs = sum, na.rm=TRUE)

# rm(alldata)

# save(dat, file = 'SpC.rda')
```

create annotation table

```
IPAS_annotation <- alldata %>% select('ExperimentType', 'Disease', 'SubType', 'IPAS') %>%
  unique()

save(IPAS_annotation, file= 'IPAS_annotation.rda')
```

```
# a small sample data for testing
dat_lite <- dat[sample(1:nrow(dat), 10000), c(1, sample(2:ncol(dat), 10))]
# dat_lite_IPAS_annotation <-
```

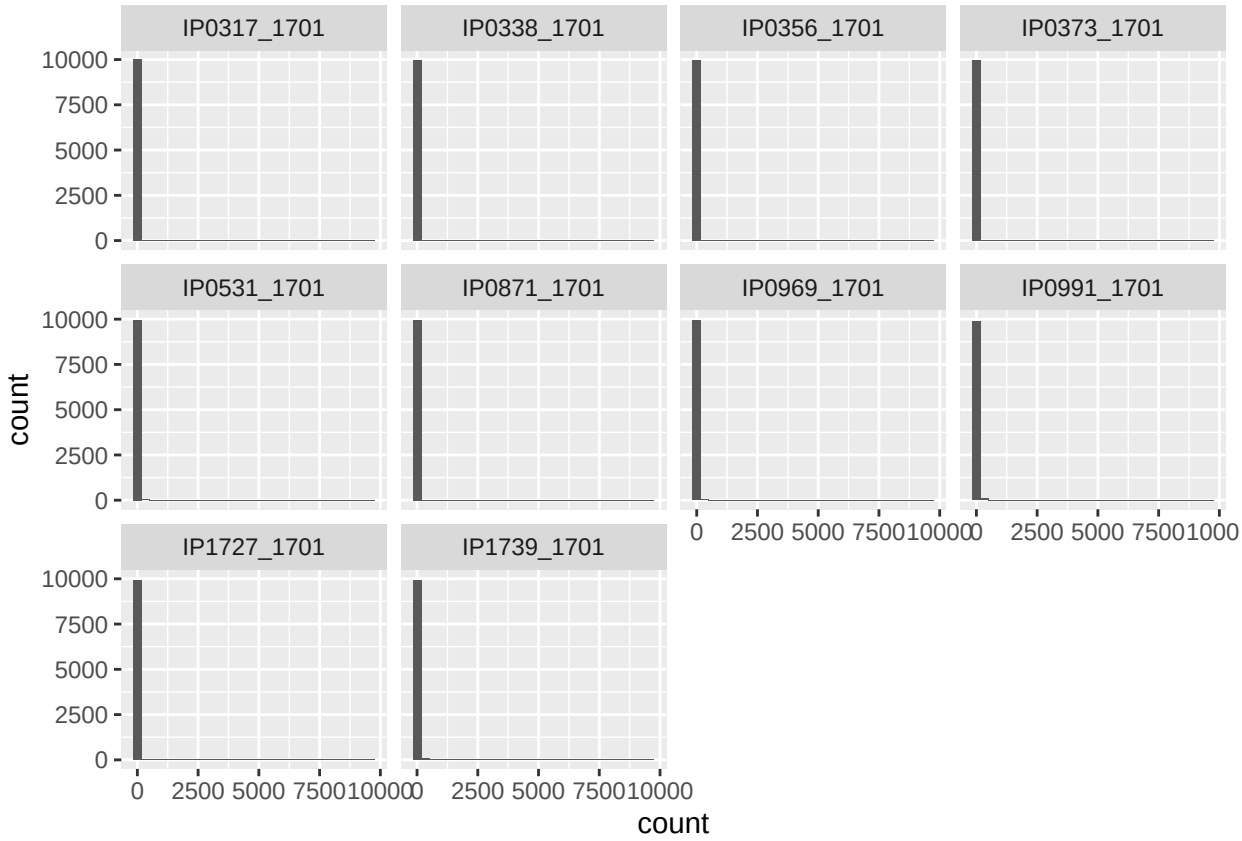
```
hist(sapply(dat[, -1], sum, na.rm=T))
```



check whether the distribution is homogenous

```
dat_lite %>% pivot_longer(cols=!Accession, names_to = 'replicate', values_to = 'count') %>%
  ggplot(aes(x=count)) +
  geom_histogram() +
  facet_wrap(~ replicate)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



save data

```
write_csv(dat, file = 'data/raw_spectral_count_all.csv')
```

```
library(tidyverse)
library(edgeR)
library(pheatmap)
library(RColorBrewer)
library(EnhancedVolcano)
```

Reading data

```
SpC <- read_csv("data/raw_spectral_count_all.csv", show_col_types = FALSE)
```

create corresponding count matrix

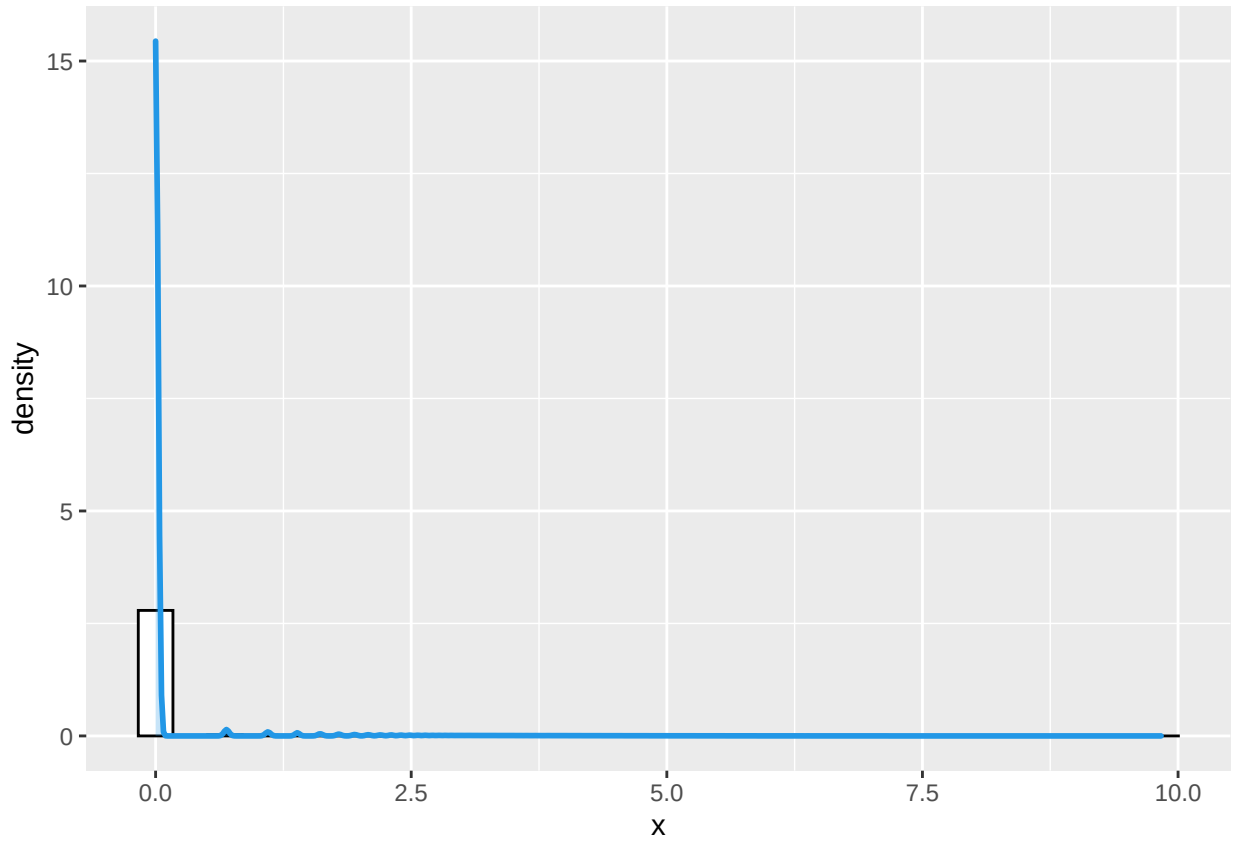
```
SpC_matrix <- as.matrix(SpC[-1])
rownames(SpC_matrix) <-  pull(SpC[1])
# SpC_matrix[is.na(SpC_matrix)] <-  0
# SpC_matrix <- log2(SpC_matrix + 1)
```

Exploration

#check underlying assumption for NB model

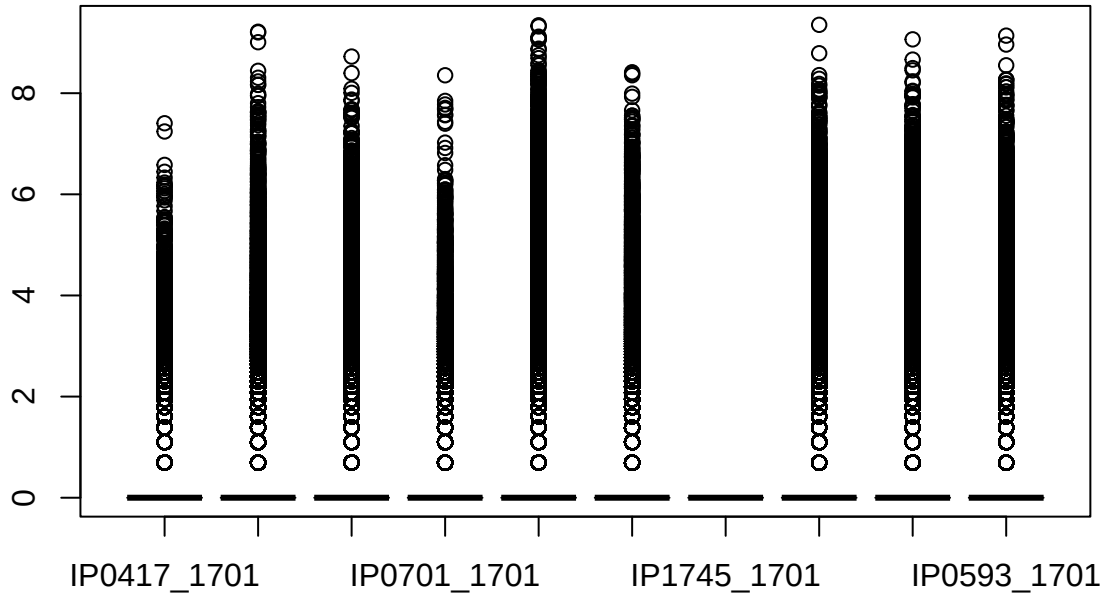
```
pseudo_counts <- log(SpC_matrix + 1)
ggplot(data = data.frame(x=c(as.matrix(pseudo_counts))), aes(x=x)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 4,
               fill = 4, alpha = 0.25)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



use to check normalization status

```
boxplot(pseudo_counts[, 1:10])
```



Missing data handling

missing count

```
# none of proteins has valid value in all replicates
SpC %>% summarise(num_of_proteins = n())
```

```
## # A tibble: 1 x 1
##   num_of_proteins
##             <int>
## 1           70989
```

```
SpC %>% drop_na() %>% summarise(num_of_proteins = n())
```

```
## # A tibble: 1 x 1
##   num_of_proteins
##             <int>
## 1           70989
```

simple imputation: replace NA as 0

```
# SpC <- SpC %>% mutate_all(~replace(., is.na(.), 0))
SpC_matrix[is.na(SpC_matrix)] <- 0
```

remove replicates that library size is 0

```
zero_library_filter <- apply(SpC_matrix, 2, sum) != 0
SpC_matrix <- SpC_matrix[, zero_library_filter]
annotation <- IPAS_annotation[zero_library_filter, ]
```

filter

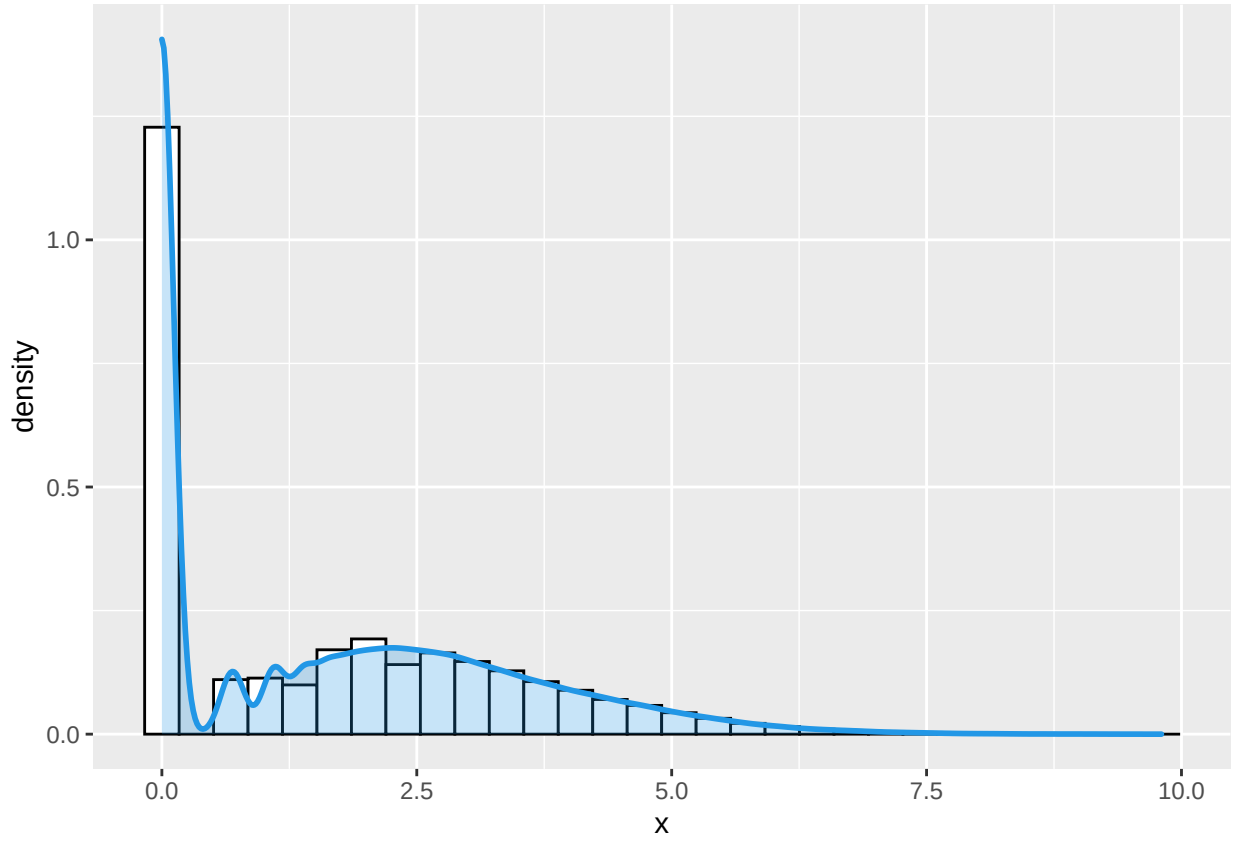
too many low count SpCs, need to find evidence supported filtered that make the tagwise distribution close to NB

```
SpC_matrix_dense <- SpC_matrix[apply(SpC_matrix, 1, function(c) sum(c!=0) >= 100), ]
```

recheck overall distribution

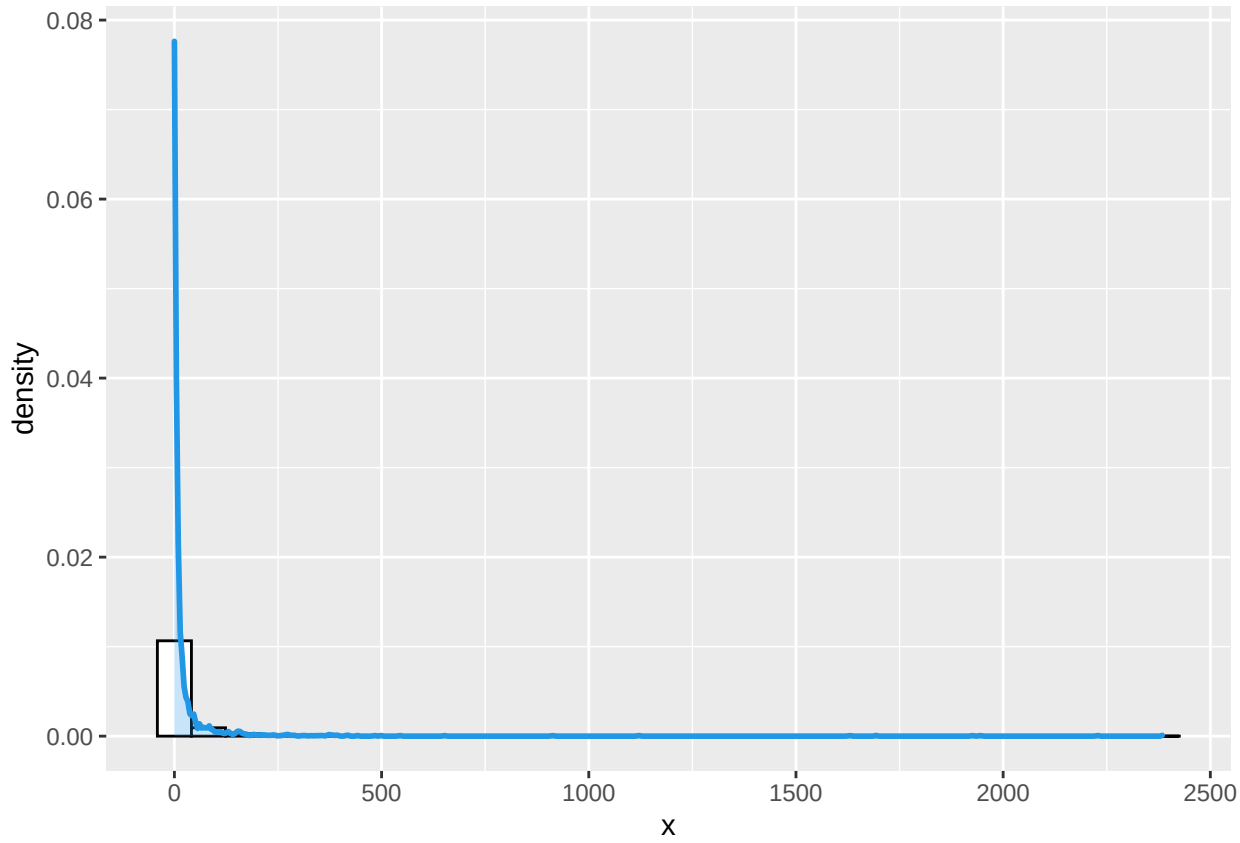
```
ggplot(data = data.frame(x=c(as.matrix(log(SpC_matrix_dense + 1)))), aes(x=x)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 4,
               fill = 4, alpha = 0.25)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(data = data.frame(x=SpC_matrix_dense[, 4]), aes(x=x)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 4,
               fill = 4, alpha = 0.25)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



create DGEList data class

```
group <- factor(annotation$Disease)
y <- DGEList(SpC_matrix_dense, group = group)
```

normalization

If use median normalization, correction factor should be feed back to the model

```
y <- calcNormFactors(y)
```

classic edgeR estimate overdispersion parameters

```
design <- model.matrix(~group)
colnames(design) <- str_replace(colnames(design), 'group', '')
y <- estimateDisp(y, design = design)

fit <- glmQLFit(y, design)
```

ANOVA test

```
ANOVA_test <- glmQLFTest(fit, coef=colnames(design)[-1])
```

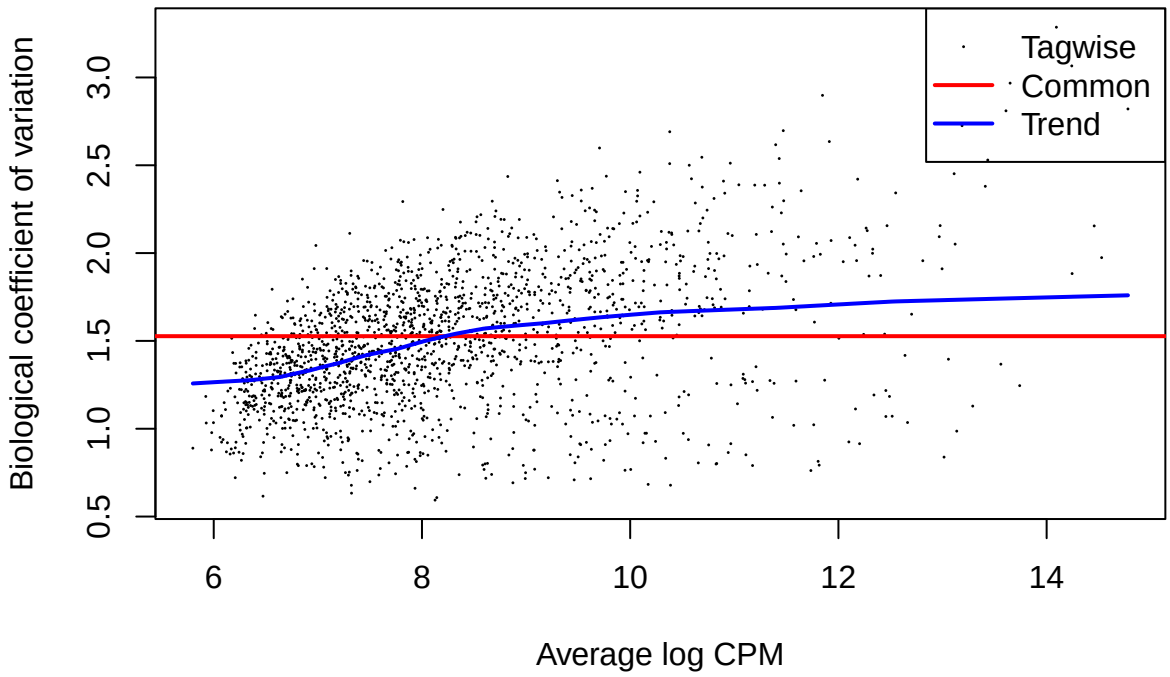
```
Contrast_Colon_breast_test <- glmQLFTest(fit, coef=c('Colon'))
```

```
ANOVA_DEPs <- topTags(ANOVA_test, 1000, p.value = 0.05)
DEPs <- topTags(Contrast_Colon_breast_test, 1000, p.value = 0.05)
```

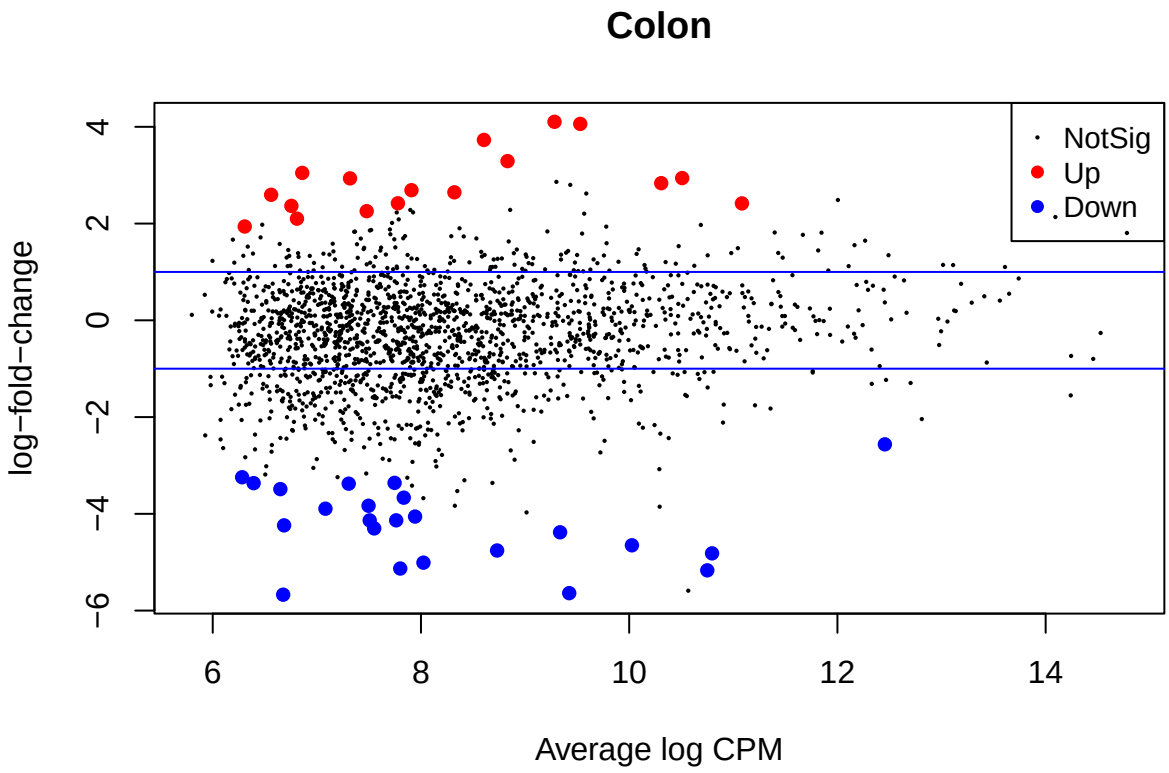
```
# if (!require("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
#
# BiocManager::install("GO.db")

# goana(test)
```

```
# plotMDS(y, labels = group, col=group)
```

```
plotBCV(y)
```



```
plotMD(Contrast_Colon_breast_test)
abline(h=c(-1, 1), col="blue")
```

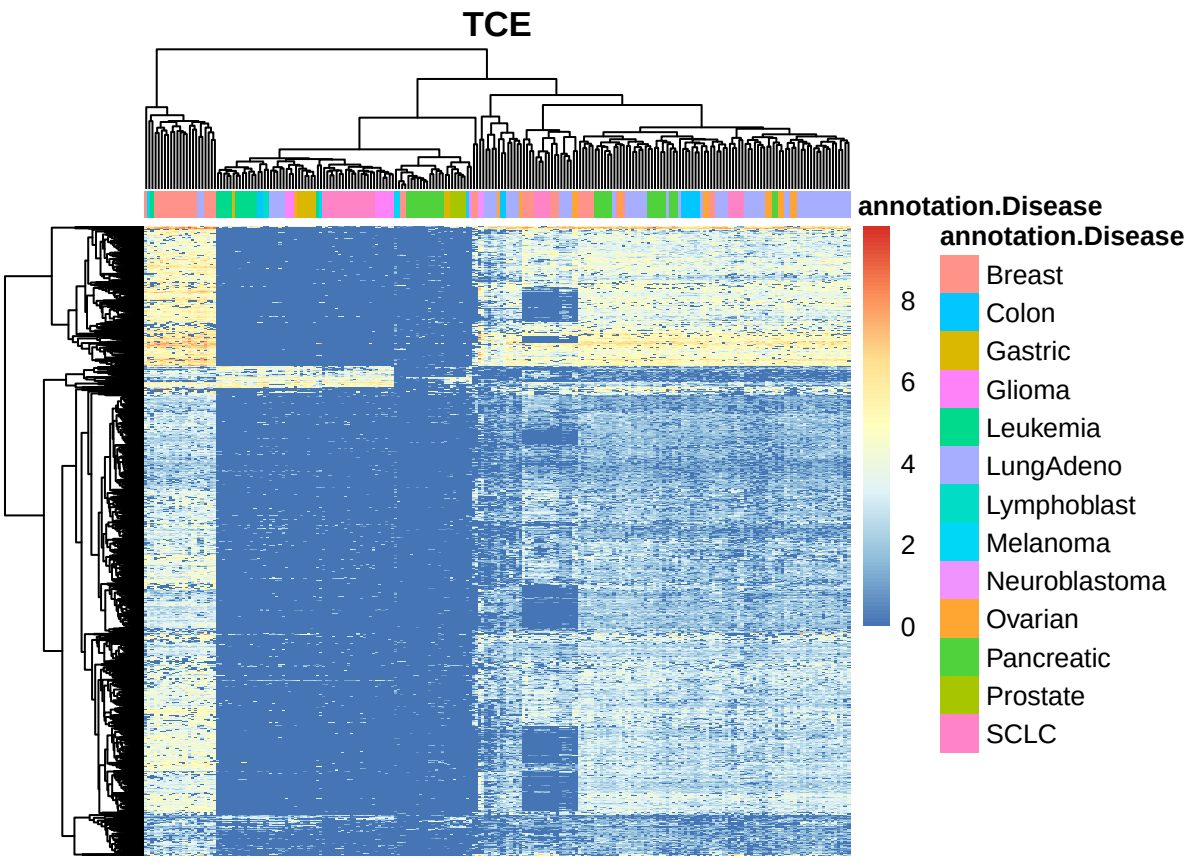


Visulation

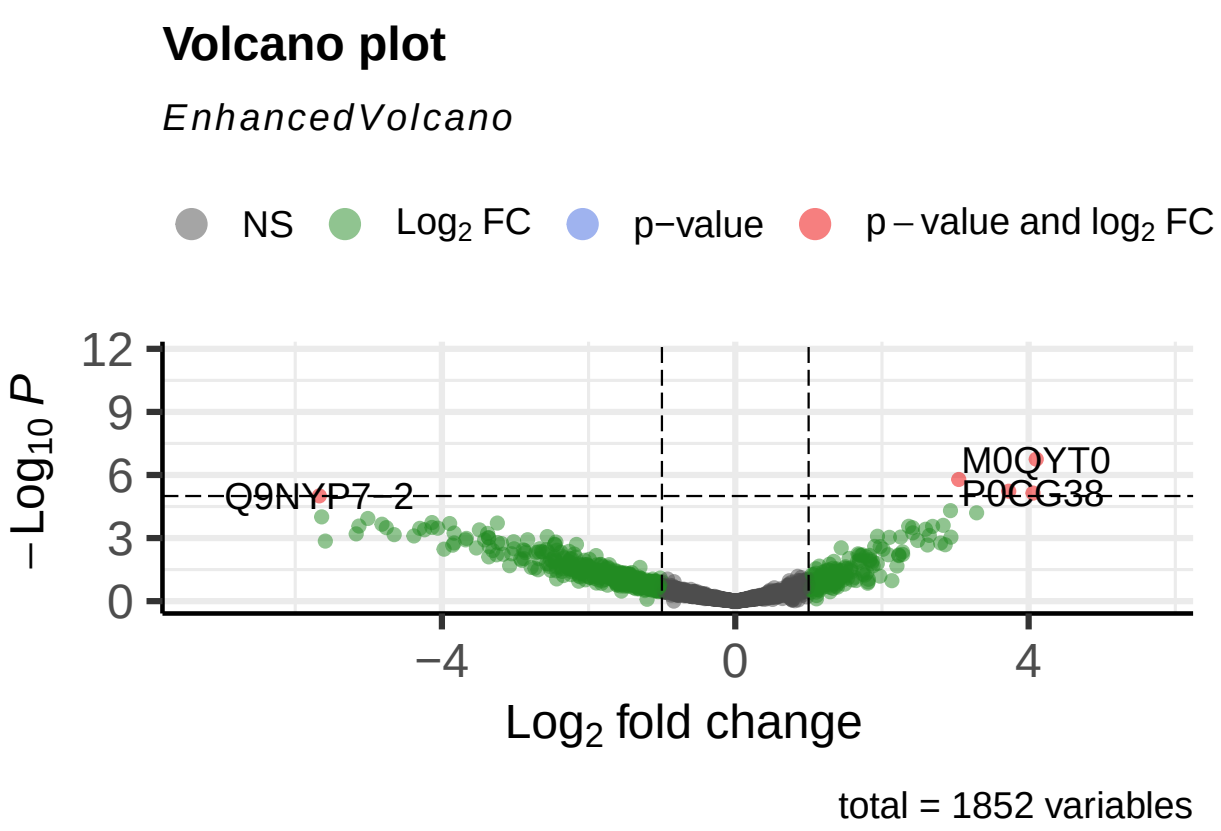
Heatmap for DE proteins in ANOVA test

```
# heatmap_dat <- cpm(y, normalized.lib.sizes = TRUE, log=TRUE, prior.count = 2)
```

```
log_SpC <- log(SpC_matrix_dense[rownames(ANOVA_DEPs$table), ] + 1)
# annotation_col is a dataframe with rownames corresponding to the colnames in the feeded matrix
pheatmap(log_SpC,
         #fontsize_col = 5,
         #fontsize_row = 4,
         show_rownames = F,
         show_colnames = F,
         main = "TCE",
         cluster_cols = T,
         cluster_rows = T,
         annotation= data.frame(annotation$Disease, row.names=annotation$IPAS))
```




```
# if (!requireNamespace('BiocManager', quietly = TRUE))
#   install.packages('BiocManager')
#
# BiocManager::install('EnhancedVolcano')


EnhancedVolcano(Contrast_Colon_breast_test$table,
   lab = rownames(Contrast_Colon_breast_test$table),
   x = 'logFC',
   y = 'PValue')
```



Venn diagram

other processing

1. collapse cancer type category
2. fold-change threshold
3. need corresponding gene name for GO and KEGG analysis

concerns:

imputation missing value with 0 may not be correct