analysis

Chunhui Gu

2024-01-25

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

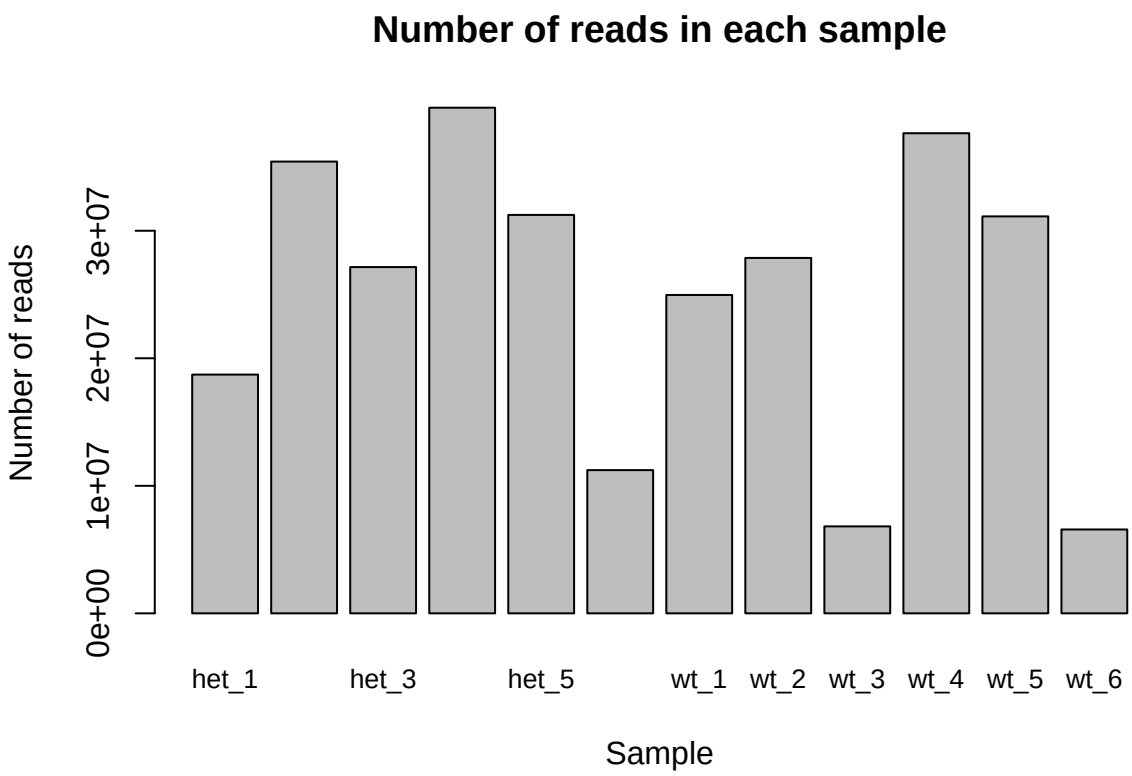
```
library(ggplot2)
library(protools)
```

Load data

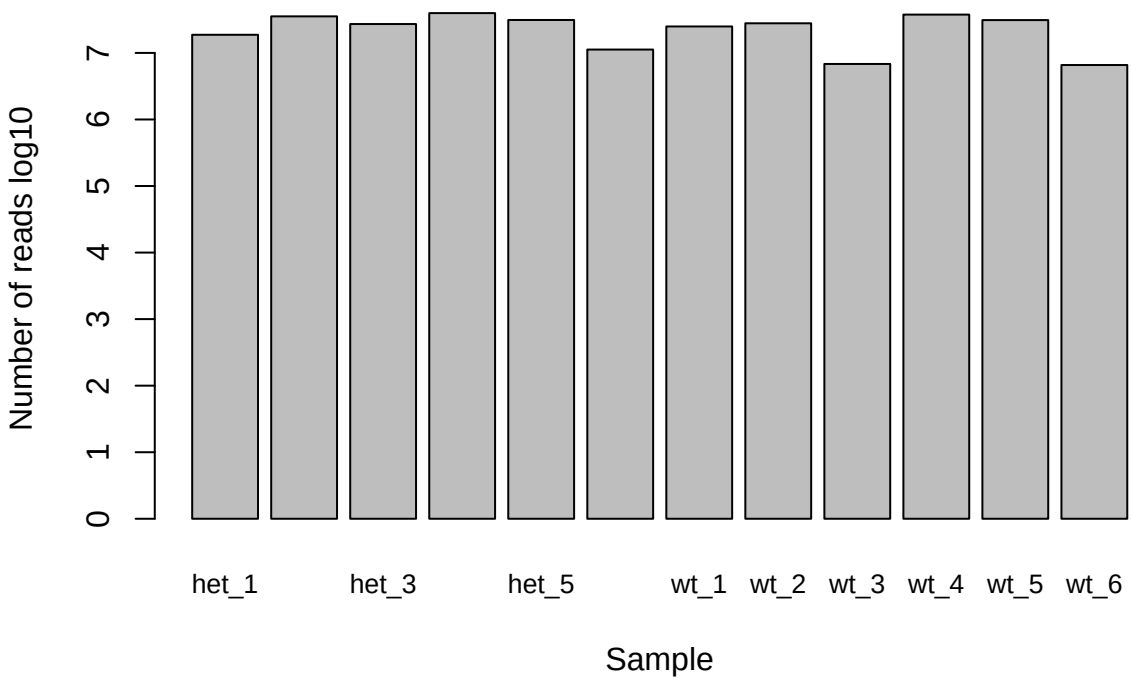
```
df <- read.csv('data/results.csv', row.names =1)
colnames(df) <- gsub('sample_', '', colnames(df))
head(df)
```

```
##                       het_1 het_2 het_3 het_4 het_5 het_6 wt_1 wt_2 wt_3 wt_4
## negative_control_1       25   729   415   454   269     0    0    0    0  459
## negative_control_10       0   401     0   441   385     0  489  445  352  899
## negative_control_100    773   504   738   153   335     0  251  181    0  417
## negative_control_1000   167   908     0   691   133   352  552    0    1  363
## negative_control_1001     0     0     0   339     0     0    0    0    1  161
## negative_control_1002   105   240     0   265     0     0  246    0    0  101
##                       wt_5 wt_6
## negative_control_1       0    0
## negative_control_10   2266    1
## negative_control_100     0    0
## negative_control_1000    0    1
## negative_control_1001    0    0
## negative_control_1002  264    0
```

```
barplot(colSums(df), main = "Number of reads in each sample", xlab = "Sample", ylab = "Number of reads"
```

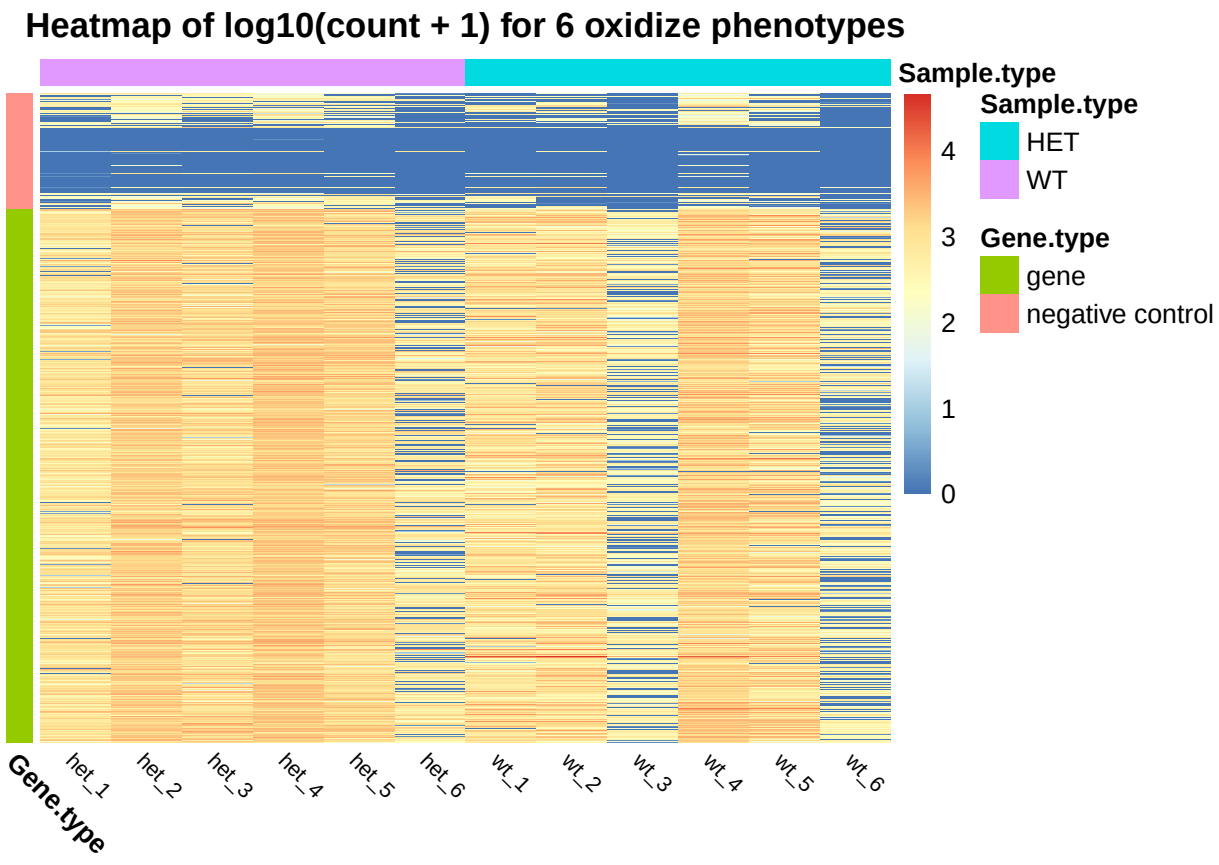



```
barplot(log10(colSums(df)), main = "Number of reads in each sample log10 scale", xlab = "Sample",
        ylab = "Number of reads log10", cex.names=.8)
```



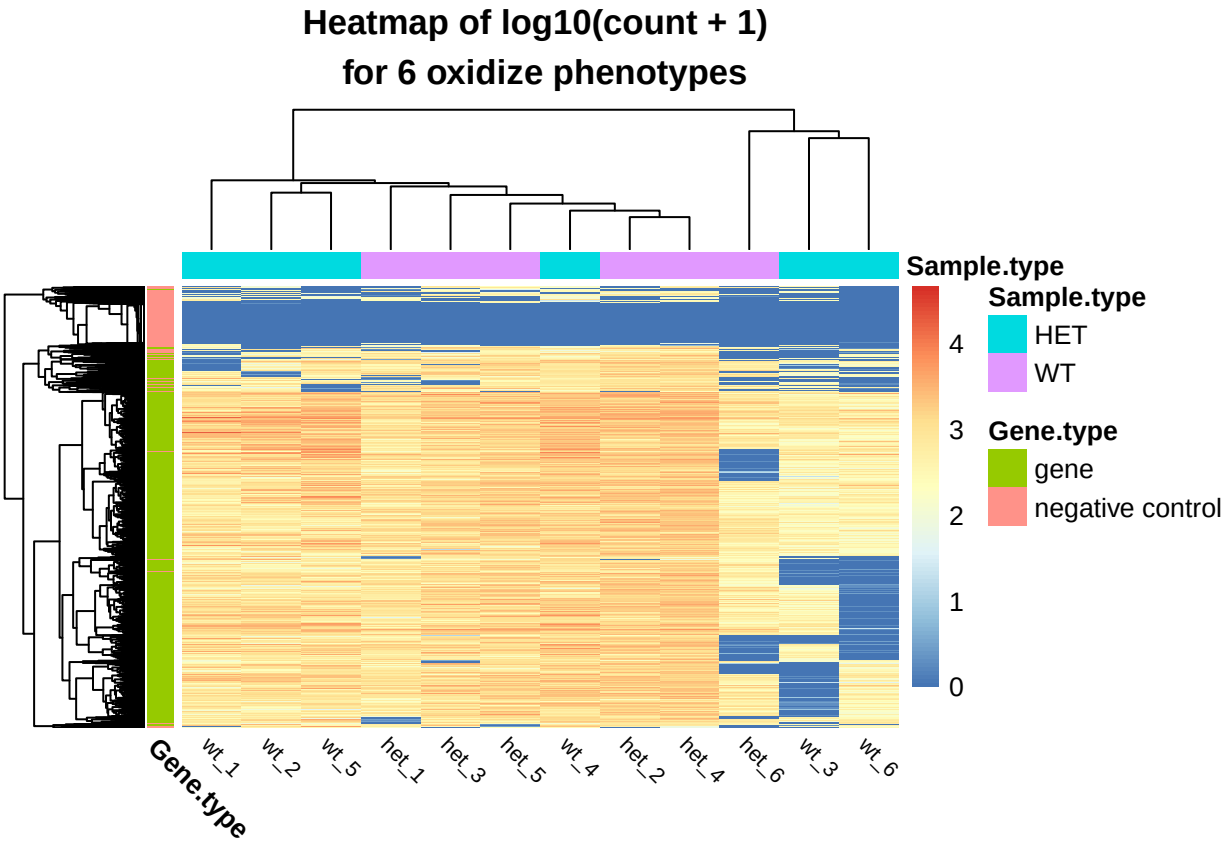
Heatmap without normalization

```
is_negative <- grepl("negative", row.names(df))
# Create a dataframe with the categorization
row_annot_df <- data.frame(
  row.names = rownames(df),
  'Gene type' = ifelse(is_negative, "negative control", "gene")
)

col_annot_df <- data.frame(
  row.names = colnames(df),
  'Sample type' = as.factor(ifelse(grepl("wt", colnames(df)), "HET", "WT"))
)

par(mfrow = c(1, 2))

pheatmap::pheatmap(log10(as.matrix(df) + 1),
                   cluster_rows = FALSE, cluster_cols = FALSE, show_rownames = FALSE, show_colnames = TR
                   angle_col = 315, fontsize_col = 8,
                   annotation_row = row_annot_df,
                   annotation_col = col_annot_df,
                   main = "Heatmap of log10(count + 1) for 6 oxidize phenotypes")
```




```
pheatmap::pheatmap(log10(as.matrix(df) + 1),
                   cluster_rows = TRUE, cluster_cols = TRUE, show_rownames = FALSE, show_colnames = TRU
```

```
                    angle_col = 315, fontsize_col = 8,
                    annotation_row = row_annot_df,
                    annotation_col = col_annot_df,
                    main = "Heatmap of log10(count + 1) \n for 6 oxidize phenotypes")
```



Heatmap with total count normalization

https://support.proteomesoftware.com/hc/en-us/articles/115002739586-Spectrum-Count-Normalization-in-Scaffold

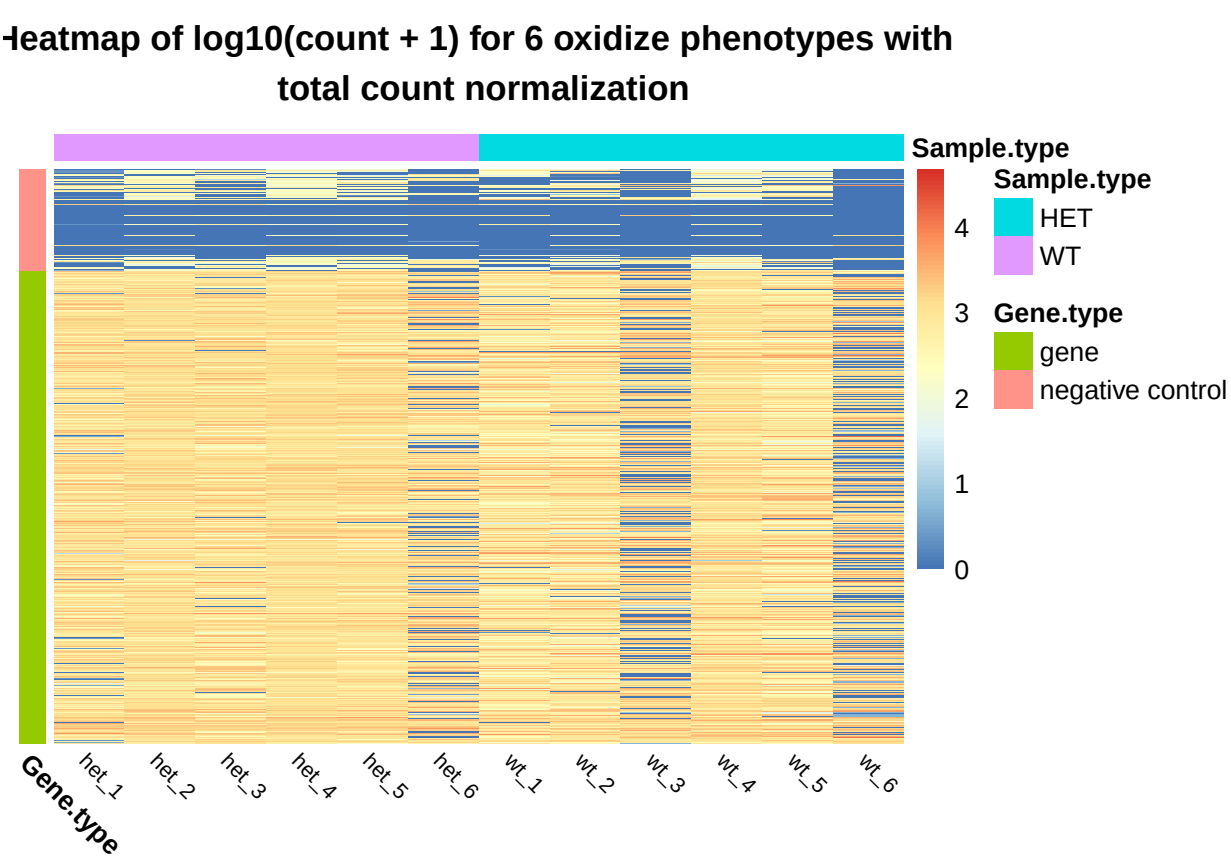
No much difference between normalized and non-normalized. So I will use non-normalized data for further analysis.

```
df_tc_norm <- protools::total_spectral_count_norm(df)
colSums(df_tc_norm) # after normalization, the sum of each column is the same
```

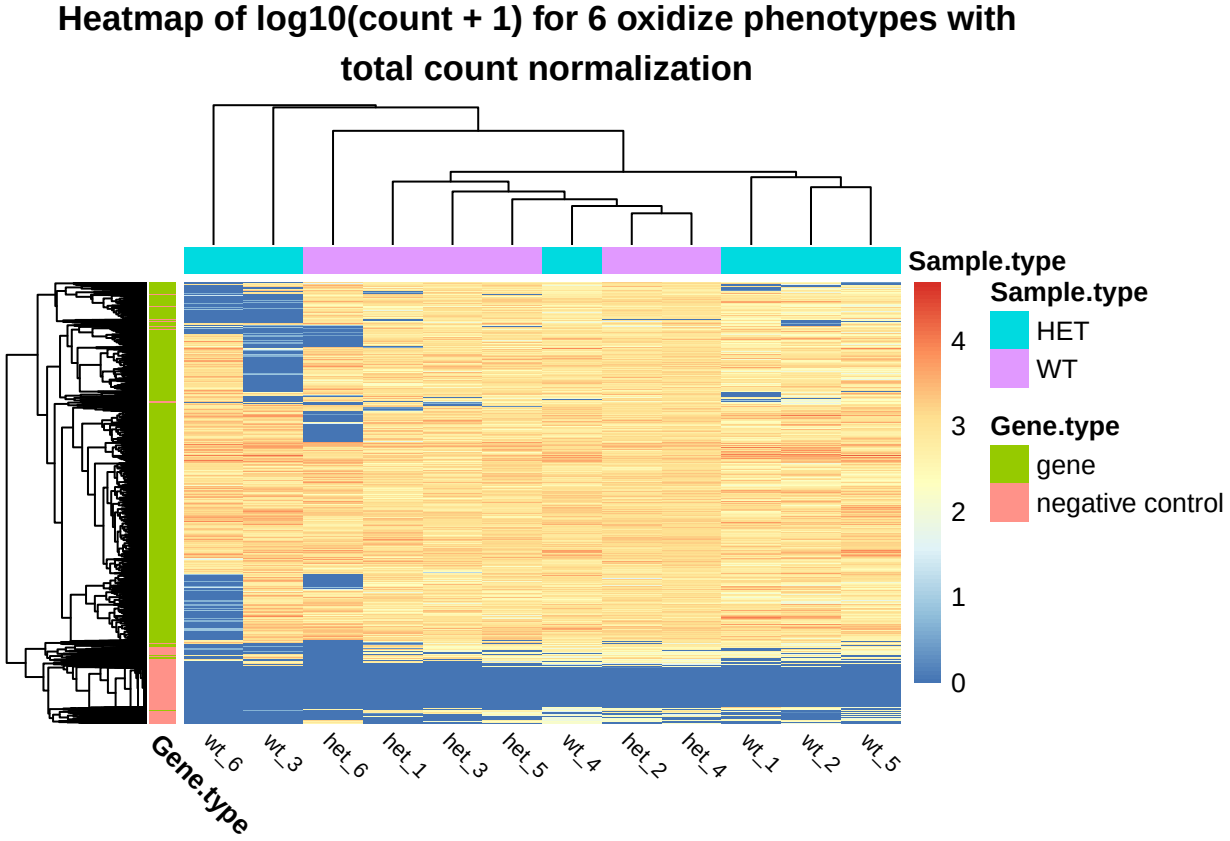
```
##    het_1    het_2    het_3    het_4    het_5    het_6     wt_1     wt_2
## 24867160 24867160 24867160 24867160 24867160 24867160 24867160 24867160
##     wt_3     wt_4     wt_5     wt_6
## 24867160 24867160 24867160 24867160
```

```
pheatmap::pheatmap(log10(as.matrix(df_tc_norm) + 1),
                    cluster_rows = FALSE, cluster_cols = FALSE, show_rownames = FALSE, show_colnames = TR
                    angle_col = 315, fontsize_col = 8,
                    annotation_row = row_annot_df,
```

```
                    annotation_col = col_annot_df,
                    main="Heatmap of log10(count + 1) for 6 oxidize phenotypes with \n total count normal
                    )
```




```
pheatmap::pheatmap(log10(as.matrix(df_tc_norm) + 1),
                    cluster_rows = TRUE, cluster_cols = TRUE, show_rownames = FALSE, show_colnames = TRU
                    angle_col = 315, fontsize_col = 8,
                    annotation_row = row_annot_df,
                    annotation_col = col_annot_df,
                    main = "Heatmap of log10(count + 1) for 6 oxidize phenotypes with \n total count nor
```



Use normalization for downstream analysis

```
df <- df_tc_norm
```

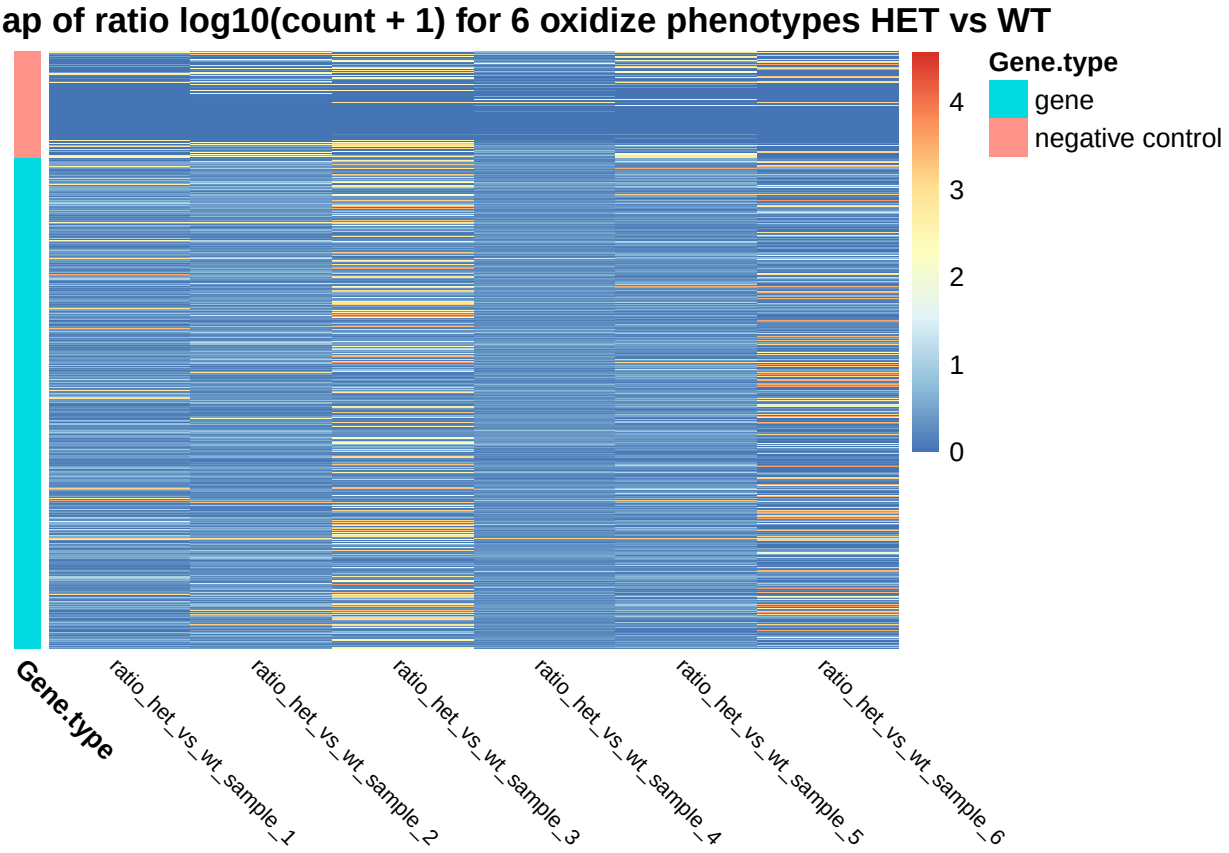
Calculate ratio of het vs wt

```
df_het <- df[paste0('het_', 1:6)]
df_wt <- df[paste0('wt_', 1:6)]

# Subset the dataframe to exclude zeros
df_no_zeros <- df[df != 0]
# Find the minimum value excluding zeros
min_value <- min(df_no_zeros, na.rm = TRUE)

# ratio het/WT. Since some value is 0 in WT, so add a small value (minimum possible count value, 1) to
df_ratio_het_wt <- df_het / (df_wt + min_value)

# max_finite <- max(df_ratio_het_wt[sapply(df_ratio_het_wt, is.finite)])
#
# # Replace Inf with the maximum finite value
# df_ratio_het_wt[sapply(df_ratio_het_wt, is.infinite)] <- max_finite

colnames(df_ratio_het_wt) <- paste0('ratio_het_vs_wt_sample_', 1:6)
```

```
cbind(df, df_ratio_het_wt) %>% write.csv('data/results_with_ratio.csv')
```

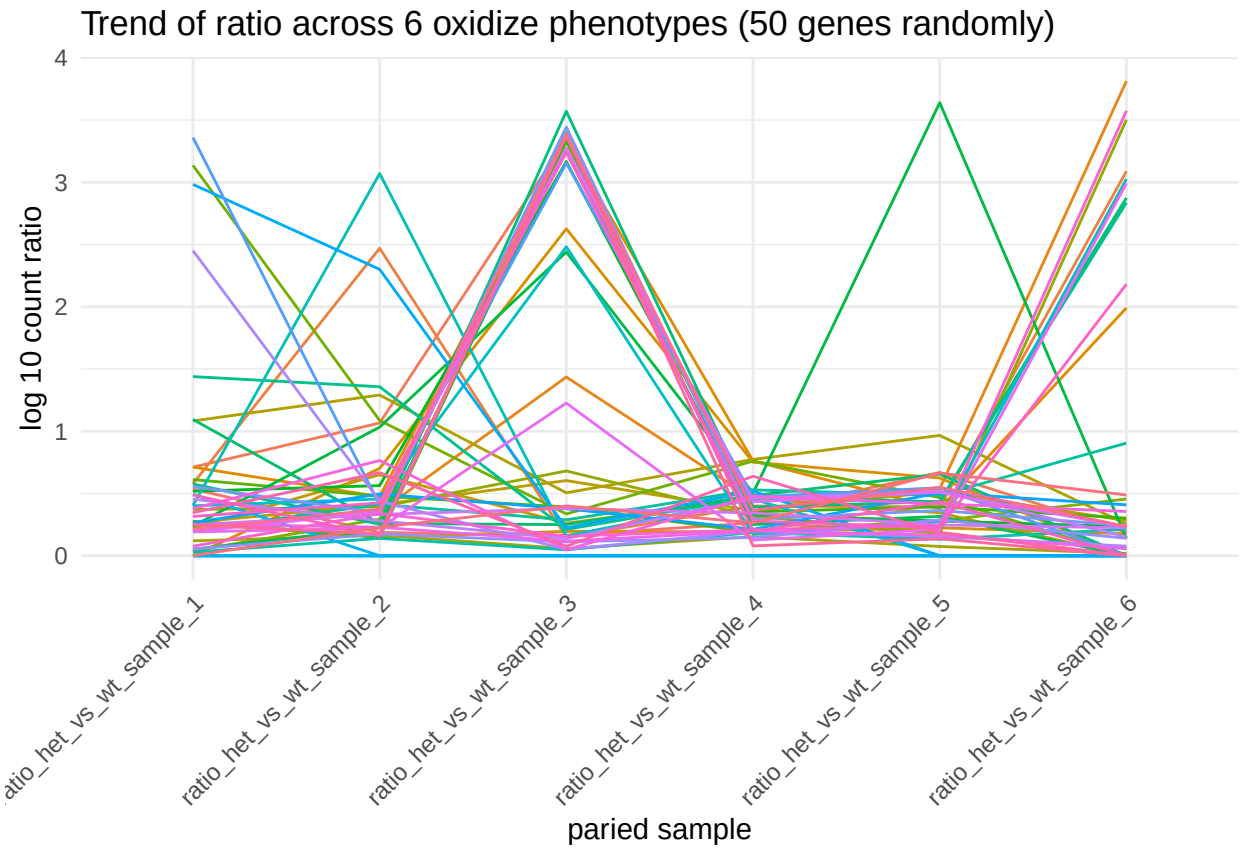
Heatmap for ratio of het vs wt (without normalization)

```
pheatmap::pheatmap(log10(as.matrix(df_ratio_het_wt) + 1), cluster_rows = FALSE, cluster_cols = FALSE, sh
                   angle_col = 315, fontsize_col = 8,
                   annotation_row = row_annot_df,
                   main="Heatmap of ratio log10(count + 1) for 6 oxidize phenotypes HET vs WT")
```



trend of ratio across 6 oxidize phenotypes (without normalization)

```
sample_df <- log10(df_ratio_het_wt[sample(nrow(df_ratio_het_wt), 50), ] + 1)

rownames_to_column(sample_df, var = "gene") %>%
  pivot_longer(cols = -gene, names_to = "sample", values_to = "ratio") %>%
  # ggplot(aes(x = factor(sample, paste0("ratio_het_vs_wt_sample_", c(4, 5, 1, 2, 6, 3))), y = ratio, g
  ggplot(aes(x = sample, y = ratio, group=gene, color=gene)) +
  geom_line() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(title = "Trend of ratio across 6 oxidize phenotypes (50 genes randomly)", x = "paried sample", y
```


Trend of ratio across 6 oxidize phenotypes (50 genes randomly)
log 10 count ratio
4
3
2
1
0
atio_het_vs_wt_sample_1
ratio_het_vs_wt_sample_2
ratio_het_vs_wt_sample_3
ratio_het_vs_wt_sample_4
ratio_het_vs_wt_sample_5
ratio_het_vs_wt_sample_6
paried sample