Maximizing the potential of high-throughput next-generation sequencing through precise normalization based on read-count distribution

2022-12-07

```
library(ggplot2)
library(dplyr)
```

```
##
## 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(ggpubr)
library(svglite)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.8     v purrr   0.3.4
## v tidyr   1.2.1     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x data.table::between() masks dplyr::between()
## x dplyr::filter()       masks stats::filter()
## x data.table::first()   masks dplyr::first()
## x dplyr::lag()          masks stats::lag()
## x data.table::last()    masks dplyr::last()
## x purrr::transpose()    masks data.table::transpose()
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:ggpubr':
##
##     get_legend
```

```
library(stringr)
```

```
iseq_norm_counts_arranged <- read.csv("iseq_norm_counts_arranged.csv")
quant_norm_counts_arranged <-  read.csv("quant_norm_counts_arranged.csv")
```

coefficient of variation (CV) of read-count normalization

```
sd(iseq_norm_counts_arranged$sample_proportion)/mean(iseq_norm_counts_arranged$sample_proportion)
```

```
## [1] 0.3738881
```

coefficient of variation (CV) of fluorescent quantification normalization

```
sd(quant_norm_counts_arranged$sample_proportion)/mean(quant_norm_counts_arranged$sample_proportion)
```

```
## [1] 0.722842
```

```
iseq_median_proportion <- median(iseq_norm_counts_arranged$R1_R2_read_proportion)
quant_median_proportion <- median(quant_norm_counts_arranged$R1_R2_read_proportion)
iseq_median_reads <- median(iseq_norm_counts_arranged$R1_R2_reads_combined)
quant_median_reads <- median(quant_norm_counts_arranged$R1_R2_reads_combined)
```

Saving 5% of the number of samples as a variable

```
percent <- (5*350)/100
percent
```

```
## [1] 17.5
```

Since the samples are already arranged in order of descending read counts, we can take the top and bottom 5% of samples straight from the data table as the samples with the most and least read counts respectively.

```
top_percent_iseq <- head(iseq_norm_counts_arranged, percent)
top_percent_quant <- head(quant_norm_counts_arranged, percent)
bottom_percent_iseq <- tail(iseq_norm_counts_arranged, percent)
bottom_percent_quant <- tail(quant_norm_counts_arranged, percent)
```

Coupling samples with their replicate from each method

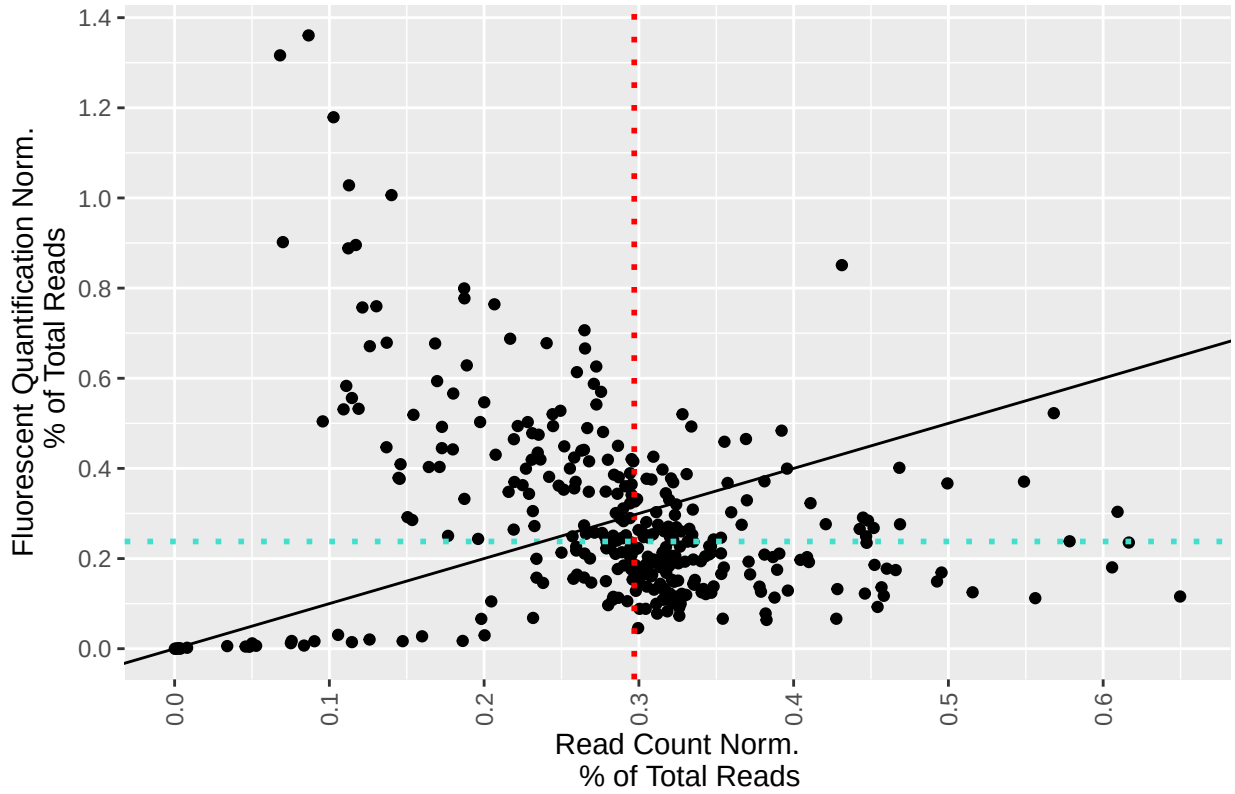
```
matching_quant_samples_iseq_top <- subset(quant_norm_counts_arranged,( Category %in% top_percent_iseq$Ca
matching_quant_samples_iseq_bottom <- subset(quant_norm_counts_arranged,( Category %in% bottom_percent_
matching_iseq_samples_quant_top <- subset(iseq_norm_counts_arranged,( Category %in% top_percent_quant$Ca
matching_iseq_samples_quant_bottom <- subset(iseq_norm_counts_arranged,( Category %in% bottom_percent_qu
```

Merge the data from each normalization method

```
merged_data <- merge(iseq_norm_counts_arranged, quant_norm_counts_arranged, by = "Sample")
#head(merged_data)
```

Create scatter plot displaying correlation of reads counts between each method

```
scatter <- ggplot(merged_data, aes(x = sample_proportion.x, y=sample_proportion.y)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  #guides(color=guide_legend(title="Host Subject ID")) +
  xlab("Read Count Norm.
   % of Total Reads") +
  ylab("Fluorescent Quantification Norm.
 % of Total Reads") +
  ggtitle("") +
  scale_x_continuous(breaks = seq(0, 2, by=0.1)) +
  scale_y_continuous(breaks = seq(0, 2, by=0.2)) +
  geom_vline(xintercept=iseq_median_proportion, linetype="dotted", color="red", size = 1) +
  geom_hline(yintercept=quant_median_proportion, linetype="dotted", color="turquoise", size = 1) +
   geom_abline(slope = 1)
scatter
```



#{r} save_plot('figure_figure_iseq_abundance_scatter.png', scatter, base_width = 10, base_height = 5) #

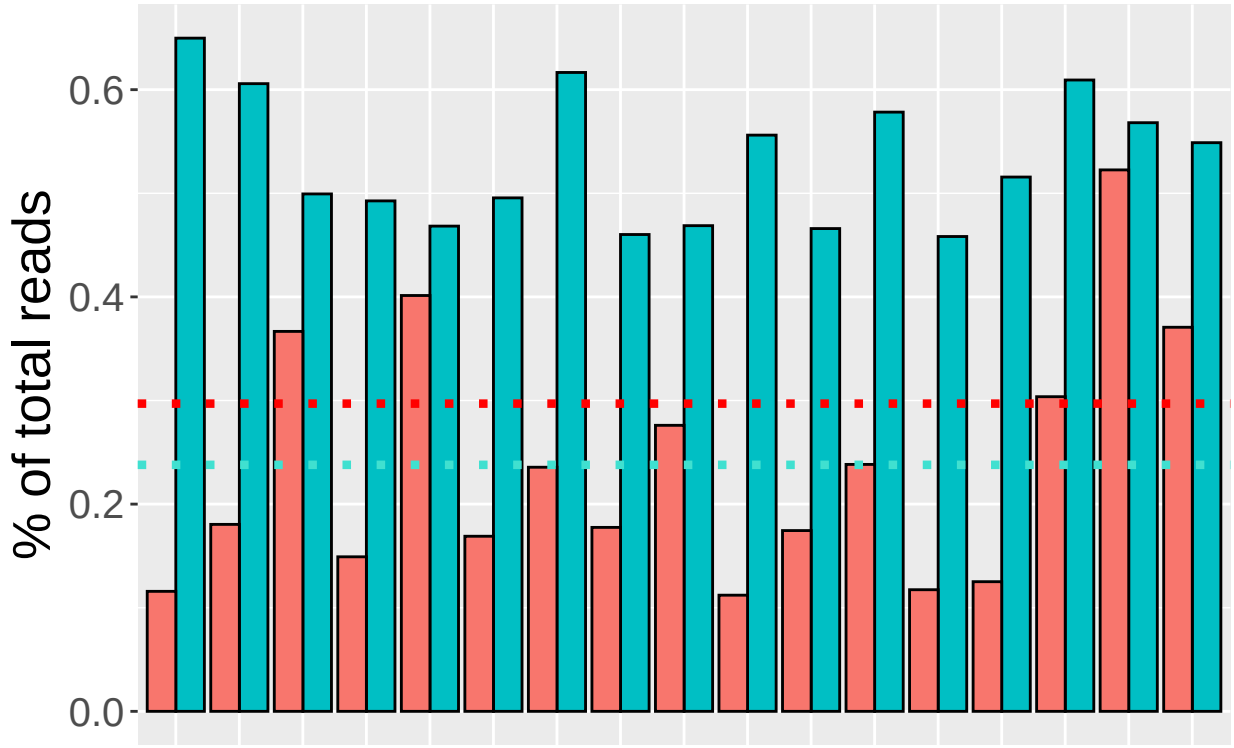
Bind both datasets from each normalization method

```
top_iseq_matching_quant <- rbind(top_percent_iseq, matching_quant_samples_iseq_top)
bottom_iseq_matching_quant <- rbind(bottom_percent_iseq, matching_quant_samples_iseq_bottom)
top_quant_matching_iseq <- rbind(top_percent_quant, matching_iseq_samples_quant_top)
bottom_quant_matching_iseq <- rbind(bottom_percent_quant, matching_iseq_samples_quant_bottom)
```

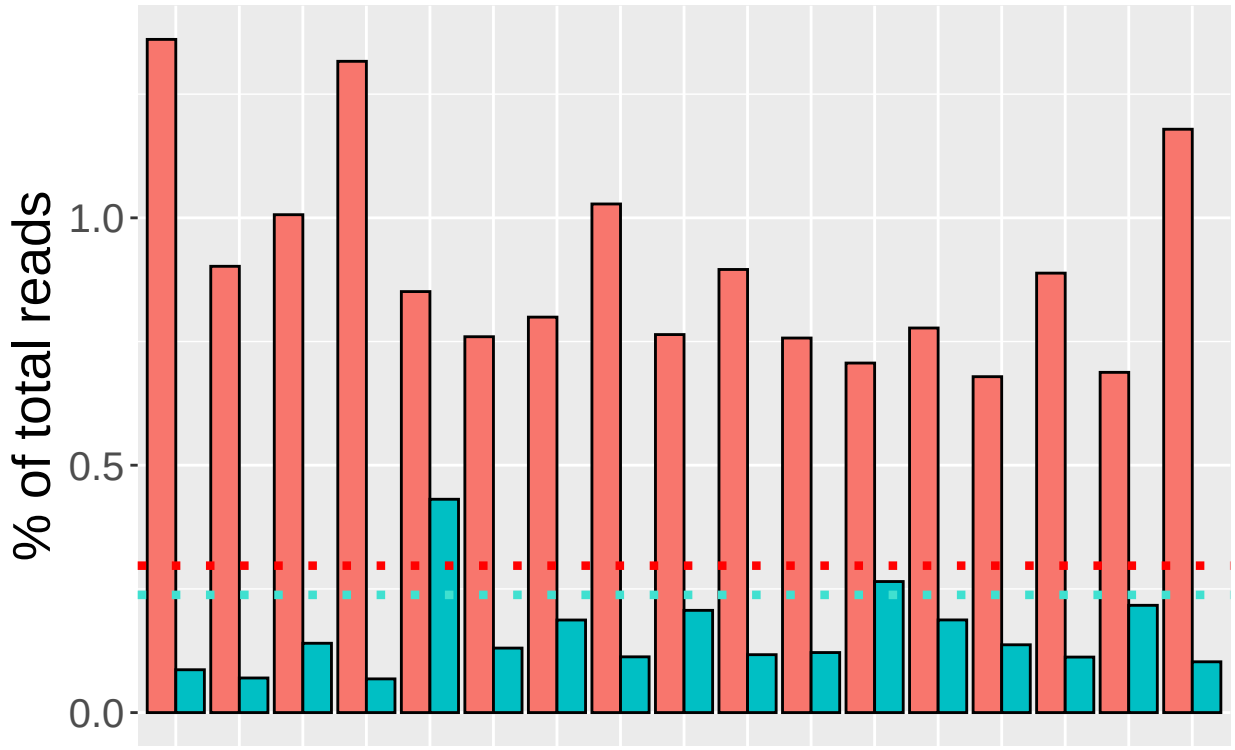
Create bar charts displaying samples from each method with most and least reads coupled with their replicates from the other method

```
#Samples from iseq read count norm with most amount of reads
a <- ggplot(top_iseq_matching_quant, aes(x=Sample, y=sample_proportion, fill=method)) +
geom_bar(stat="identity", color="black", position=position_dodge()) +
  theme(axis.text.x = element_blank(),
        legend.position="none",
        axis.ticks.x = element_blank(),
        axis.title.y = element_text(size = 20),
        axis.text.y = element_text(size = 15),
  plot.title = element_text(size = 15))+
  guides(color=guide_legend(title="Top 2% of Samples with most reads from iSeq Norm")) +
  geom_hline(yintercept=c(iseq_median_proportion,quant_median_proportion), linetype=c("dotted","dotted"
  xlab("") +
  ylab("% of total reads") +
```

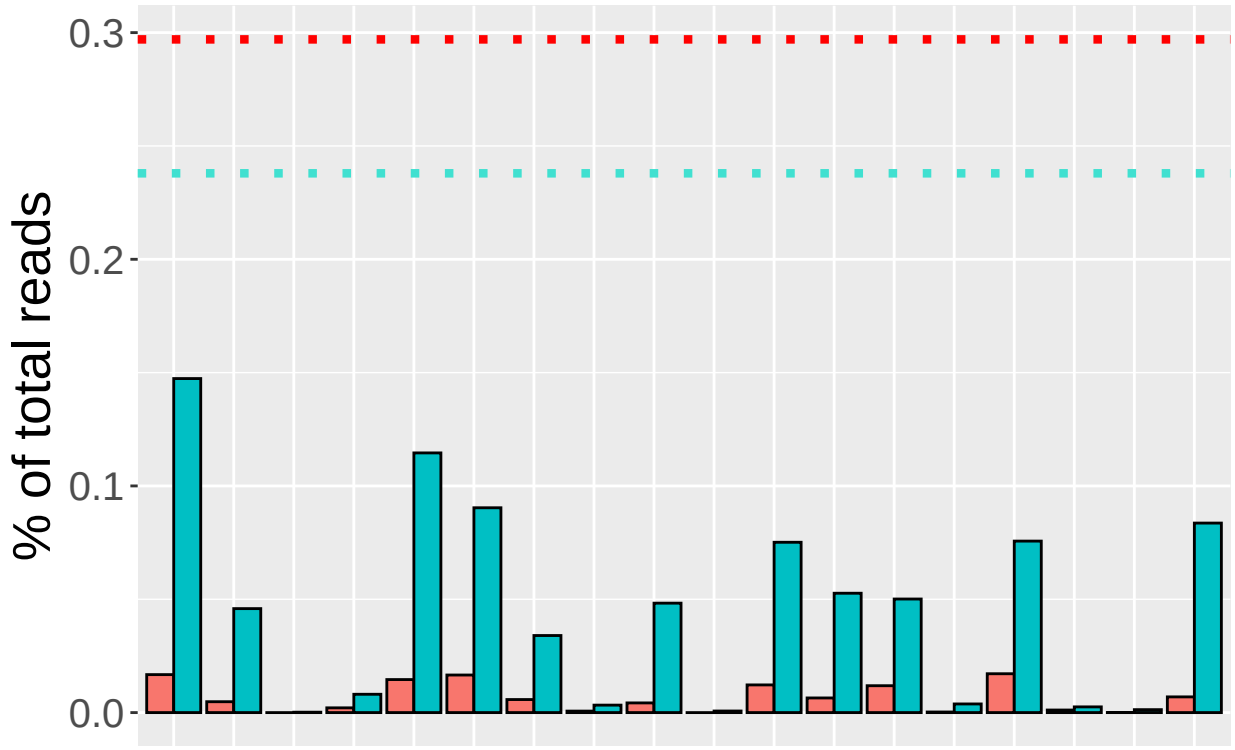
```
  ggtitle("")
a
```



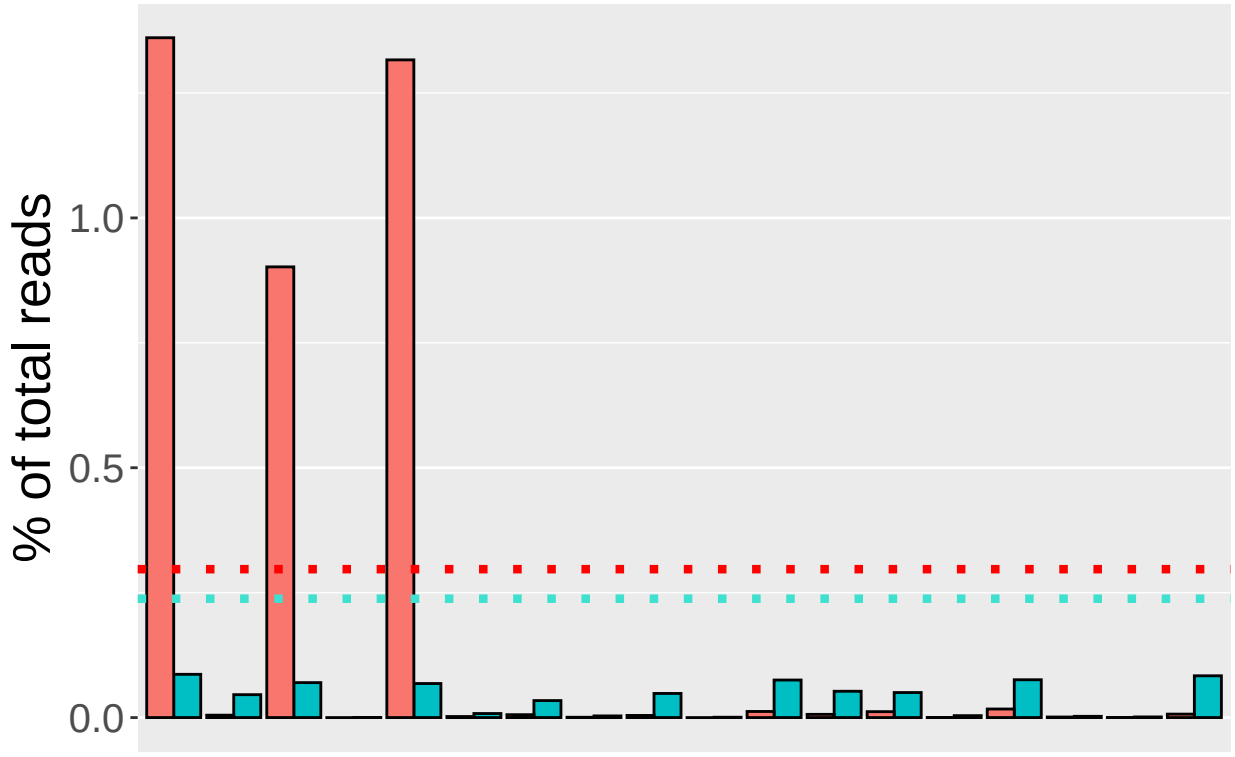

```
#Samples from fluorescent quant norm with most amount of reads
b <- ggplot(top_quant_matching_iseq, aes(x=Sample, y=sample_proportion, fill=method)) +
geom_bar(stat="identity", color="black", position=position_dodge()) +
    theme(axis.text.x = element_blank(),
        legend.position="none",
        axis.ticks.x = element_blank(),
        axis.title.y = element_text(size = 20),
        axis.text.y = element_text(size = 15),
        plot.title = element_text(size = 15)) +
    geom_hline(yintercept=c(iseq_median_proportion,quant_median_proportion), linetype=c("dotted","dotte
  xlab("") +
  ylab("% of total reads") +
  ggtitle("")
b
```




```
#Samples from fluorescent quant norm with least amount of reads
c <- ggplot(bottom_quant_matching_iseq, aes(x=Sample, y=sample_proportion, fill=method)) +
geom_bar(stat="identity", color="black", position=position_dodge()) +
  theme(axis.text.x = element_blank(),
        legend.position="none",
        axis.ticks.x = element_blank(),
        axis.title.y = element_text(size = 20),
        axis.text.y = element_text(size = 15),
        plot.title = element_text(size = 15)) +
    geom_hline(yintercept=c(iseq_median_proportion,quant_median_proportion), linetype=c("dotted","dotte
  xlab("") +
  ylab("% of total reads") +
  ggtitle("")
c
```

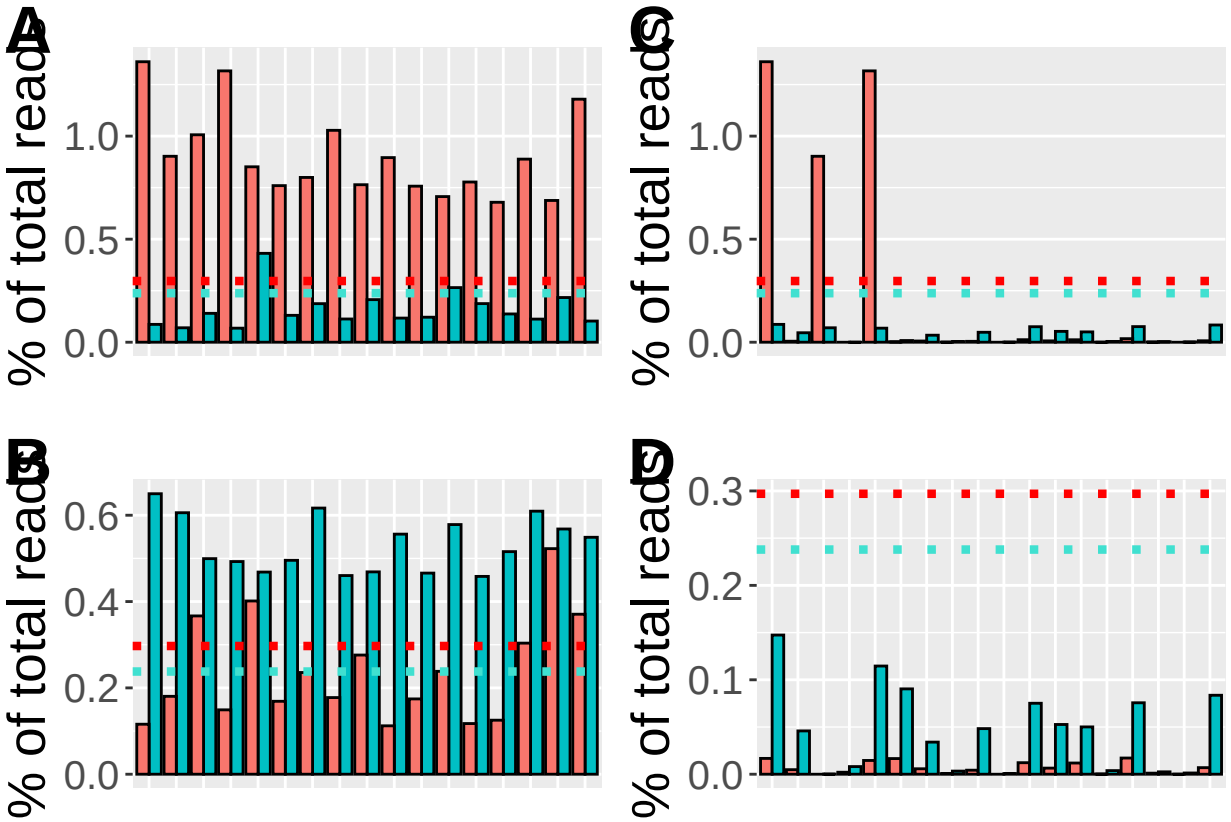



```
#Samples from iseq read count norm with least amount of reads
d <- ggplot(bottom_iseq_matching_quant, aes(x=Sample, y=sample_proportion, fill=method)) +
geom_bar(stat="identity", color="black", position=position_dodge()) +
  #guides(fill=guide_legend(title="Normalization")) +
  theme(axis.text.x = element_blank(),
        legend.position="none",
        axis.ticks.x = element_blank(),
        axis.title.y = element_text(size = 20),
        axis.text.y = element_text(size = 15),
        plot.title = element_text(size = 15)) +
  scale_x_discrete(breaks = seq(0,1.5, by = 0.05)) +
    geom_hline(yintercept=c(iseq_median_proportion,quant_median_proportion), linetype=c("dotted","dotte
  xlab("") +
  ylab("% of total reads") +
  ggtitle("")
d
```



Combine plots

```
figure_iseq_abundance_2 <- plot_grid(#a,
                                     b,
                                   #c,
                                   d,
                                   a,
                                   c,
                                   labels = c("A", "C", "B", "D"),
                                   align = "h",
                                   axis = "l",
                                   label_size = 25,
                                   label_fontfamily = 'sans',
                                   ncol = 2,
                                   hjust = -0.25)
figure_iseq_abundance_2
```



#{r} save_plot('iseq_abundance.png', figure_iseq_abundance_2, base_width = 16, base_height = 9) #

Creating Rarefaction Curve

Arranging the data by decreasing read counts

```
iseq_norm_counts_arranged_samples <- iseq_norm_counts_arranged[order(iseq_norm_counts_arranged$R1_R2_rea

quant_norm_counts_arranged_samples <- quant_norm_counts_arranged[order(quant_norm_counts_arranged$R1_R2_
head(iseq_norm_counts_arranged_samples)
```

```
##                                    Category       Sample     method
## 1  0363146883_S88_L001_R1_001.fastp.fastq.gz 363146883_R1 Read Count
## 2  0363192131_S59_L001_R1_001.fastp.fastq.gz 363192131_R1 Read Count
## 3  0363238220_S68_L001_R1_001.fastp.fastq.gz 363238220_R1 Read Count
## 4 0363146937_S333_L001_R1_001.fastp.fastq.gz 363146937_R1 Read Count
## 5 0363237571_S160_L001_R1_001.fastp.fastq.gz 363237571_R1 Read Count
## 6  0363238229_S75_L001_R1_001.fastp.fastq.gz 363238229_R1 Read Count
##   R1_R2_reads_combined sample_proportion R1_R2_read_proportion
## 1                37112         0.6497290             0.6497290
## 2                35220         0.6166053             0.6166053
## 3                34804         0.6093223             0.6093223
## 4                34602         0.6057858             0.6057858
## 5                33034         0.5783344             0.5783344
## 6                32450         0.5681102             0.5681102
```

Assigning a rank to these reads counts

```
quant_norm_counts_arranged_samples$rank <- c(1:350)
head(quant_norm_counts_arranged_samples)
```

```
##                                     Category       Sample
## 1 0363146697_S317_L001_R1_001.fastp.fastq.gz 363146697_R1
## 2  0363146989_S76_L001_R1_001.fastp.fastq.gz 363146989_R1
## 3  0363238270_S45_L001_R1_001.fastp.fastq.gz 363238270_R1
## 4 0363236818_S356_L001_R1_001.fastp.fastq.gz 363236818_R1
## 5 0363146966_S354_L001_R1_001.fastp.fastq.gz 363146966_R1
## 6  0363146934_S78_L001_R1_001.fastp.fastq.gz 363146934_R1
##                        method R1_R2_reads_combined sample_proportion
## 1 Fluorescent Quantification                62388         1.3606635
## 2 Fluorescent Quantification                60358         1.3163898
## 3 Fluorescent Quantification                54062         1.1790760
## 4 Fluorescent Quantification                47136         1.0280220
## 5 Fluorescent Quantification                46142         1.0063431
## 6 Fluorescent Quantification                41360         0.9020492
##   R1_R2_read_proportion rank
## 1             1.3606635    1
## 2             1.3163898    2
## 3             1.1790760    3
## 4             1.0280220    4
## 5             1.0063431    5
## 6             0.9020492    6
```

```
iseq_norm_counts_arranged_samples$rank <- c(1:350)
head(iseq_norm_counts_arranged_samples)
```

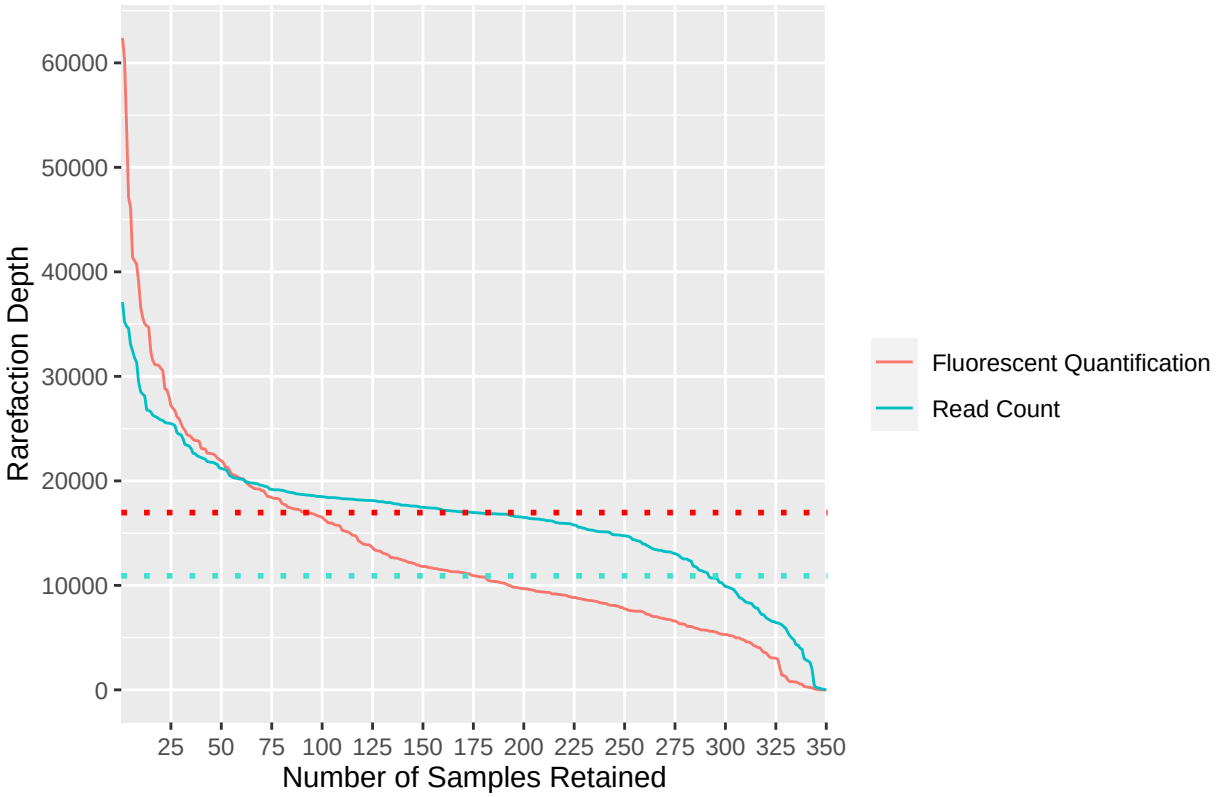
```
##                                     Category       Sample     method
## 1  0363146883_S88_L001_R1_001.fastp.fastq.gz 363146883_R1 Read Count
## 2  0363192131_S59_L001_R1_001.fastp.fastq.gz 363192131_R1 Read Count
## 3  0363238220_S68_L001_R1_001.fastp.fastq.gz 363238220_R1 Read Count
## 4 0363146937_S333_L001_R1_001.fastp.fastq.gz 363146937_R1 Read Count
## 5 0363237571_S160_L001_R1_001.fastp.fastq.gz 363237571_R1 Read Count
## 6  0363238229_S75_L001_R1_001.fastp.fastq.gz 363238229_R1 Read Count
##   R1_R2_reads_combined sample_proportion R1_R2_read_proportion rank
## 1                37112         0.6497290             0.6497290    1
## 2                35220         0.6166053             0.6166053    2
## 3                34804         0.6093223             0.6093223    3
## 4                34602         0.6057858             0.6057858    4
## 5                33034         0.5783344             0.5783344    5
## 6                32450         0.5681102             0.5681102    6
```

Combining both data sets

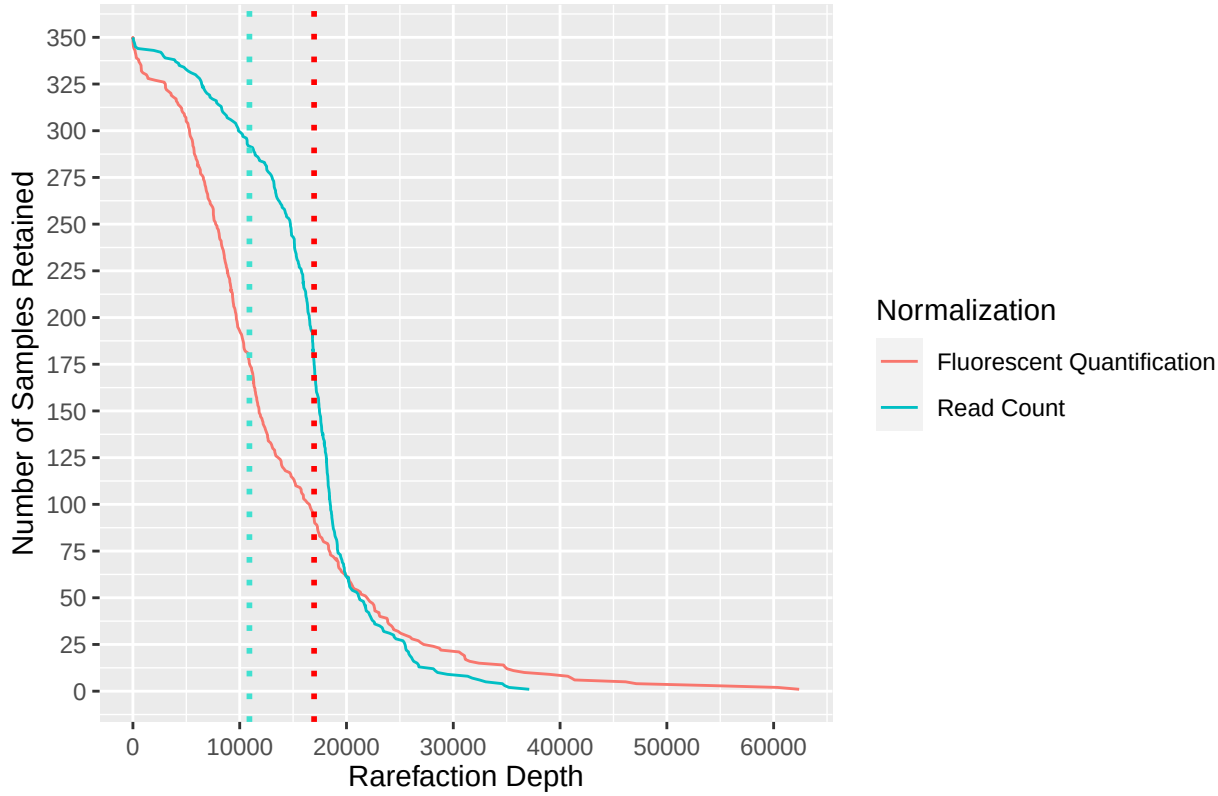
```
total_reads <- rbind(iseq_norm_counts_arranged_samples, quant_norm_counts_arranged_samples)
```

Creating Plot

```
descending_reads_plot <- ggplot(total_reads, aes(x = reorder(rank,-R1_R2_reads_combined), y = R1_R2_reac
  geom_line(aes(color=method)) +
  theme(axis.text.x = element_text(angle = 0, vjust = 0.5)) +
        guides(color=guide_legend(title=""))) +
  scale_x_discrete(breaks = seq(0, 400, by = 25)) +
  scale_y_continuous(breaks = seq(0, 60000, by = 10000)) +
  xlab("Number of Samples Retained") +
  ylab("Rarefaction Depth") +
  ggtitle("") +
   geom_hline(yintercept=c(iseq_median_reads,quant_median_reads), linetype="dotted",color = c("red", "tu
descending_reads_plot
```



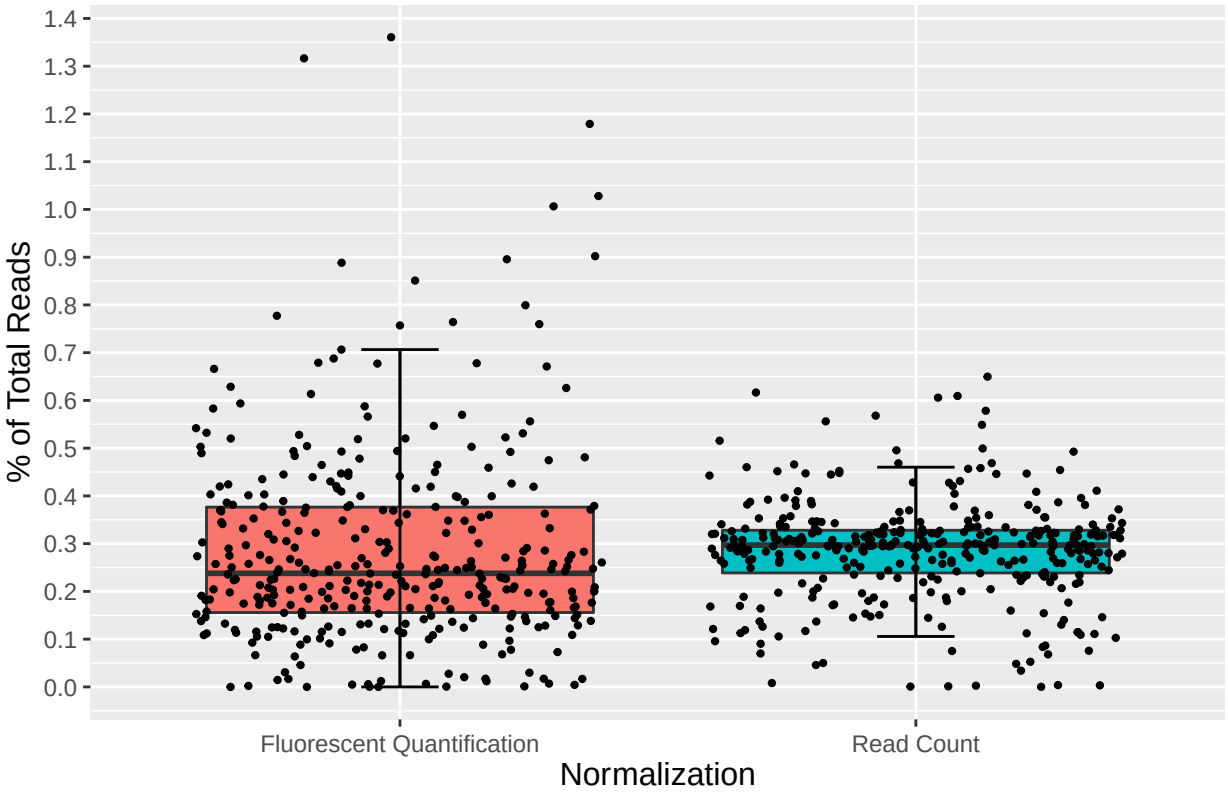

```
rarefaction <- ggplot(total_reads, aes(x = R1_R2_reads_combined, y = rank, group=method)) +
  geom_line(aes(color=method)) +
  theme(axis.text.y = element_text(angle = 0, vjust = 0.5),
        axis.text.x = element_text(angle = 0, vjust = 0.5)) +
        guides(color=guide_legend(title="Normalization")) +
  scale_y_continuous(breaks = seq(0, 400, by = 25)) +
scale_x_continuous(breaks = seq(0, 70000, by = 10000)) +
  ylab("Number of Samples Retained") +
  xlab("Rarefaction Depth") +
  ggtitle("") +
  geom_vline(xintercept=c(iseq_median_reads,quant_median_reads), linetype="dotted",color = c("red", "tu
rarefaction
```



#{r} save_plot('iseq_rarefaction.png', rarefaction, base_width = 8, base_height = 4) #

Creating boxplot displaying read counts per sample per normalization method

```
iseq_boxplot <- ggplot(total_reads, aes(x = method, y =sample_proportion)) +
  geom_boxplot(aes(fill=method), outlier.shape=NA) +
  geom_jitter(size=0.8) +
  theme(axis.text.y = element_text(angle = 0, vjust = 0.5),
        axis.text.x = element_text(angle = 0, vjust = 0.5),
        axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=12),
        legend.position="none")  +
        guides(color=guide_legend(title="")) +
  stat_boxplot(geom = "errorbar",
               width = 0.15) +
  scale_y_continuous(breaks = seq(0, 1.5, by = 0.1)) +
  ylab("% of Total Reads") +
  xlab("Normalization") +
  ggtitle("")
iseq_boxplot
```



#{r} save_plot('iseq_boxplot.png', iseq_boxplot, base_width = 8, base_height = 4) #