Using the Synthetic Ladder to normalize and detect quantitative differences between libraries

Andre Reis

Defining functions

```
# Definition of some custom functions that are used in the script

# Deseq2 normalization
deseq2 <- function(df) {
  library(DESeq2)
  library(reshape2)
  sequence <- df$Sequence
  labels <- paste("kmer",seq(1,length(sequence)),sep="")
  rownames(df) <- labels
  cts <- df[,-1]
  coldata <- data.frame(condition=rep("ladder",length(colnames(cts))))
  rownames(coldata) <- colnames(cts)
  dds <- DESeqDataSetFromMatrix(countData = cts,
                                colData = coldata,
                                design = ~ 1)
  dds <- DESeq(dds)
  norm_counts <- counts(dds,normalized=TRUE)
  norm_counts <- data.table(Sequence = sequence, norm_counts)
  factors <- sizeFactors(dds)
  return(list("norm"=norm_counts,"factors"=factors))
}

# Upper Quartile normalization
uqua <- function(mat) {
  uq1 <-quantile(mat[,1])[4]
  uq2 <-quantile(mat[,2])[4]
  uq <- c(uq1,uq2)
  ratio <- uq/mean(uq)
  return(t(apply(mat, 1, function(x) x/ratio)))
}

# GET EQUATION AND R-SQUARED AS STRING
# SOURCE: https://groups.google.com/forum/#!topic/ggplot2/1TgH-kG5XMA

lm_eqn <- function(df){
    m <- lm(y ~ x, df);
    eq <- substitute(italic(y) == b %.% italic(x),
         list(b = format(unname(coef(m)[2]), digits = 3),
             r2 = format(summary(m)$r.squared, digits = 3)))
    as.character(as.expression(eq));
}
```

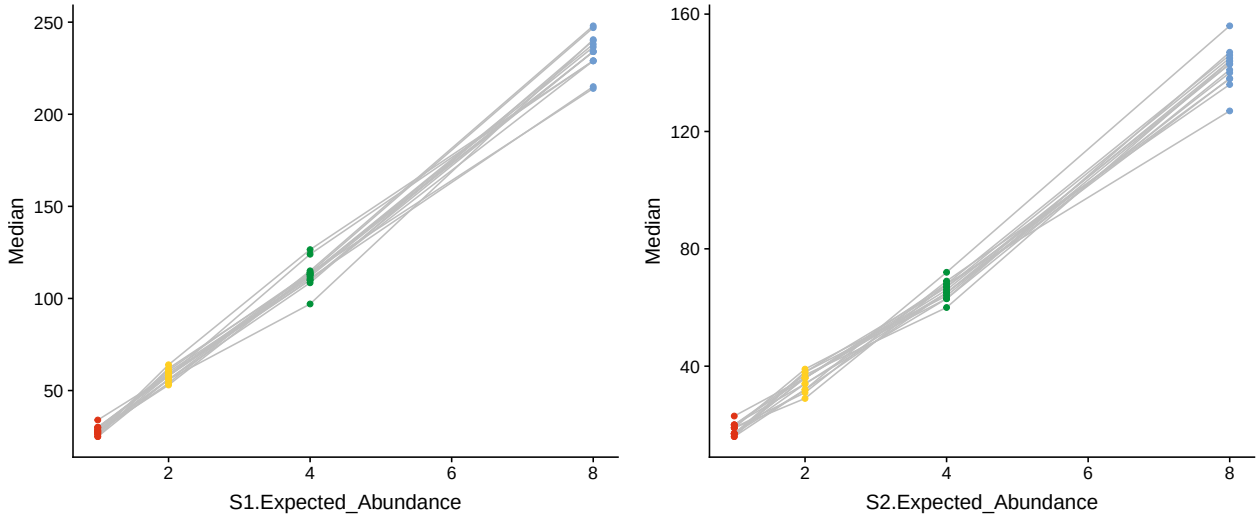
1) Dataset

The dataset contains the following columns:

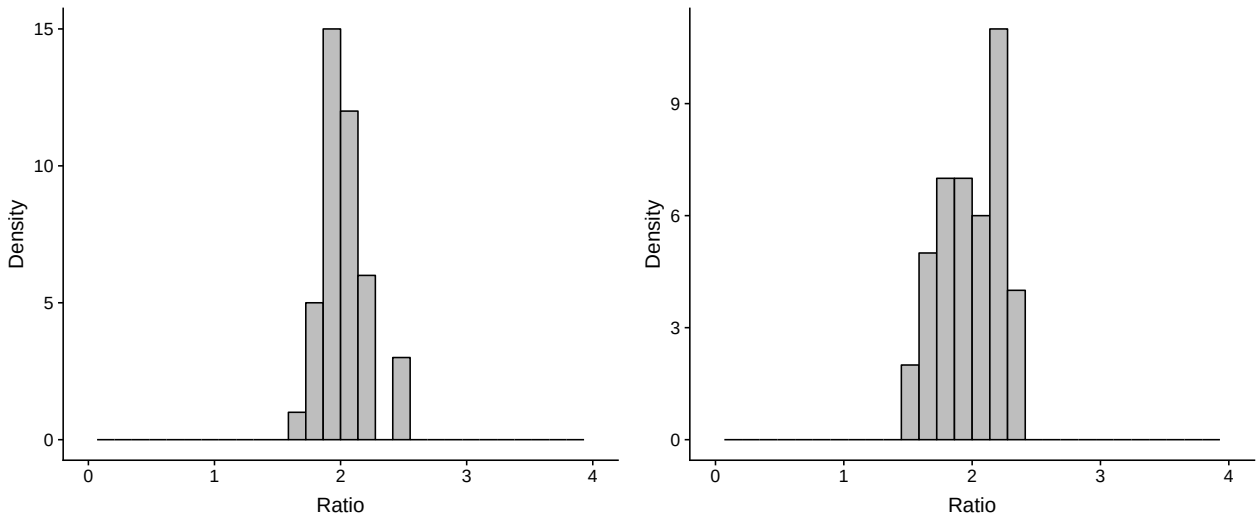
1) Sequence: k-mer sequence
2) S1: observed count for k-mer in sample 1
3) S2: observed count for k-mer in sample 2
4) Name: ID of synthetic ladder or bacterial genome from which the k-mer was originated
5) S1.Expected_Abundance: expected abundance for k-mer in sample 1
6) S1.Expected_Abundance: expected abundance for k-mer in sample 2
7) Type: origin of the k-mer (synthetic ladder or bacterial)

The plots below show the ladder in samples 1 and 2 and how it compares to the accompanying meta k-mers.

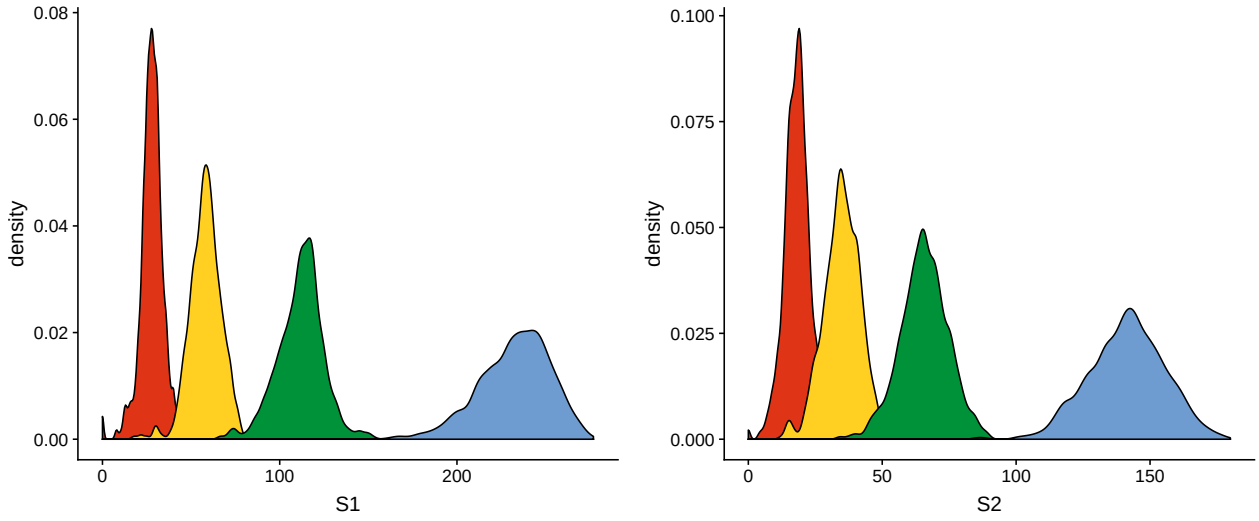
```
# Loading the dataset
dt <- fread('dataset.tab',header=TRUE)

# Partioning ladder k-mers
ladder <- dt[Type=="synthetic_ladder"]
p1 <- ggplot(ladder[,.(Median=median(S1)),by=list(Name,S1.Expected_Abundance)],
             aes(S1.Expected_Abundance,Median,by=Name))+
             geom_line(color="gray")+
             geom_point(aes(color=as.factor(S1.Expected_Abundance)))+
             scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
             guides(color=FALSE)
p2 <- ggplot(ladder[,.(Median=median(S2)),by=list(Name,S2.Expected_Abundance)],
             aes(S2.Expected_Abundance,Median,by=Name))+
             geom_line(color="gray")+
             geom_point(aes(color=as.factor(S2.Expected_Abundance)))+
             scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
             guides(color=FALSE)
# Average k-mer count for different synthetic ladders in S1 and S2
grid.arrange(p1,p2,nrow=1)
```



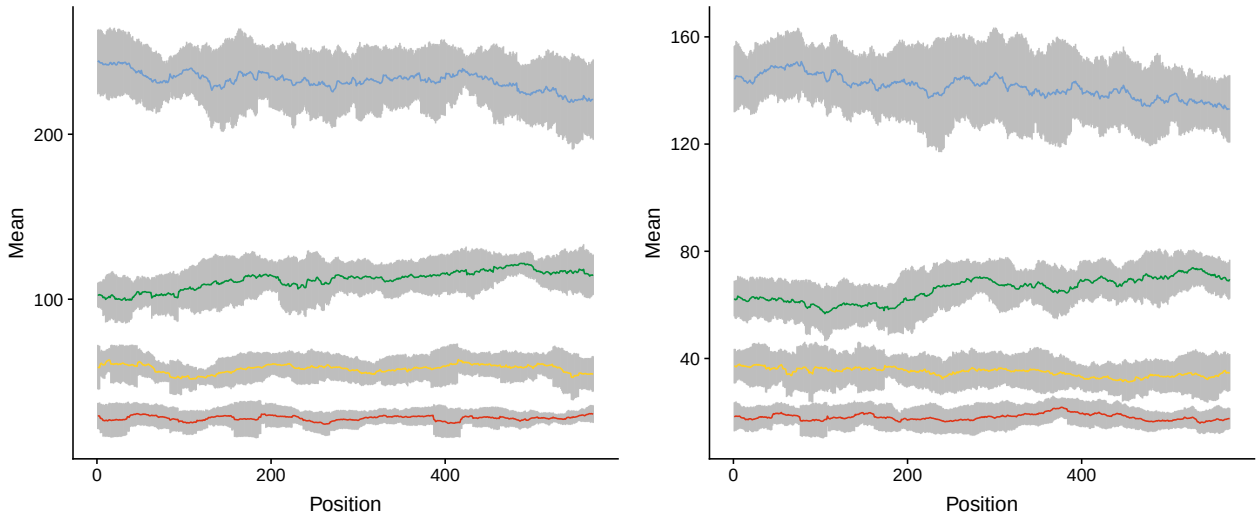
```
ratio.s1 <- ladder[,.(Median=median(S1)),by=list(Name,S1.Expected_Abundance)]
ratio.s1 <- data.table(dcast(ratio.s1,Name~S1.Expected_Abundance,value.var = "Median"))
ratio.s1[,R1:=`2`/`1`]
ratio.s1[,R2:=`4`/`2`]
ratio.s1[,R3:=`8`/`4`]
ratio.s1 <- melt(ratio.s1[,.(Name,R1,R2,R3)],id.vars=c("Name"),variable.name="Variable",value.name="Rati

p1 <- ggplot(ratio.s1,aes(Ratio))+
        geom_histogram(color="black",fill="gray")+
        scale_x_continuous(limits=c(0,4),breaks=0:4)+
        labs(x="Ratio",y="Density")

ratio.s2 <- ladder[,.(Median=median(S2)),by=list(Name,S2.Expected_Abundance)]
ratio.s2 <- data.table(dcast(ratio.s2,Name~S2.Expected_Abundance,value.var = "Median"))
ratio.s2[,R1:=`2`/`1`]
ratio.s2[,R2:=`4`/`2`]
ratio.s2[,R3:=`8`/`4`]
ratio.s2 <- melt(ratio.s2[,.(Name,R1,R2,R3)],id.vars=c("Name"),variable.name="Variable",value.name="Rati

p2 <- ggplot(ratio.s2,aes(Ratio))+
        geom_histogram(color="black",fill="gray")+
        scale_x_continuous(limits=c(0,4),breaks=0:4)+
        labs(x="Ratio",y="Density")
# Average ratio between subsequent copy-numbers in S1 and S2
grid.arrange(p1,p2,nrow=1)
```




```
p1 <- ggplot(ladder,aes(S1,fill=as.factor(S1.Expected_Abundance)))+
        geom_density()+
        scale_fill_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        guides(fill=FALSE)
p2 <- ggplot(ladder,aes(S2,fill=as.factor(S2.Expected_Abundance)))+
        geom_density()+
        scale_fill_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        guides(fill=FALSE)
# Variation at each copy-number in S1 and S2
grid.arrange(p1,p2,nrow=1)
```

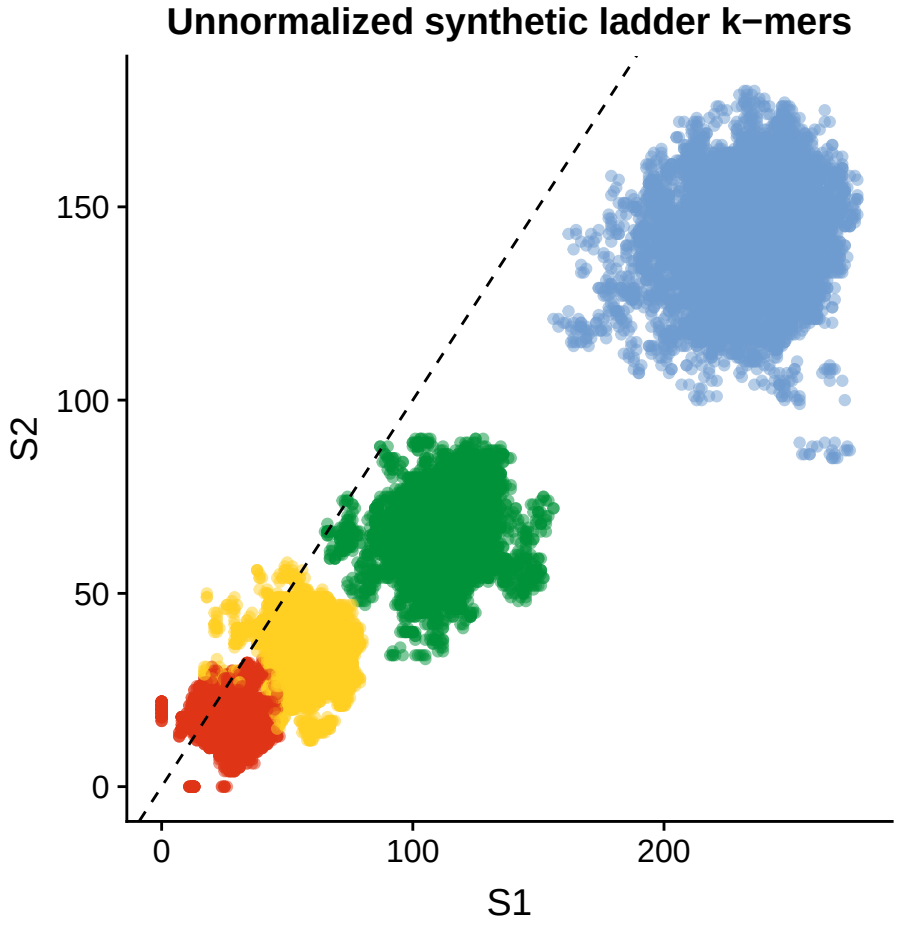



```
position <- fread('ladder_kmer_position.tab')
position <- merge(position,ladder,by=c("Name","Sequence"))
positional_variation.s1 <- position[,.(Mean=mean(S1),SD=sd(S1)),by=list(S1.Expected_Abundance,Position)]
p1 <- ggplot(positional_variation.s1,aes(Position,Mean,color=as.factor(S1.Expected_Abundance)))+
      scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
      geom_errorbar(width=.1, aes(ymin=Mean-SD, ymax=Mean+SD),color="gray")+
      geom_line()+
      guides(color=FALSE)

positional_variation.s2 <- position[,.(Mean=mean(S2),SD=sd(S2)),by=list(S2.Expected_Abundance,Position)]
p2 <- ggplot(positional_variation.s2,aes(Position,Mean,color=as.factor(S2.Expected_Abundance)))+
      scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
      geom_errorbar(width=.1, aes(ymin=Mean-SD, ymax=Mean+SD),color="gray")+
      geom_line()+
      guides(color=FALSE)
# Variation per k-mer position at each copy-number across all ladders in S1 and S2
grid.arrange(p1,p2,nrow=1)
```



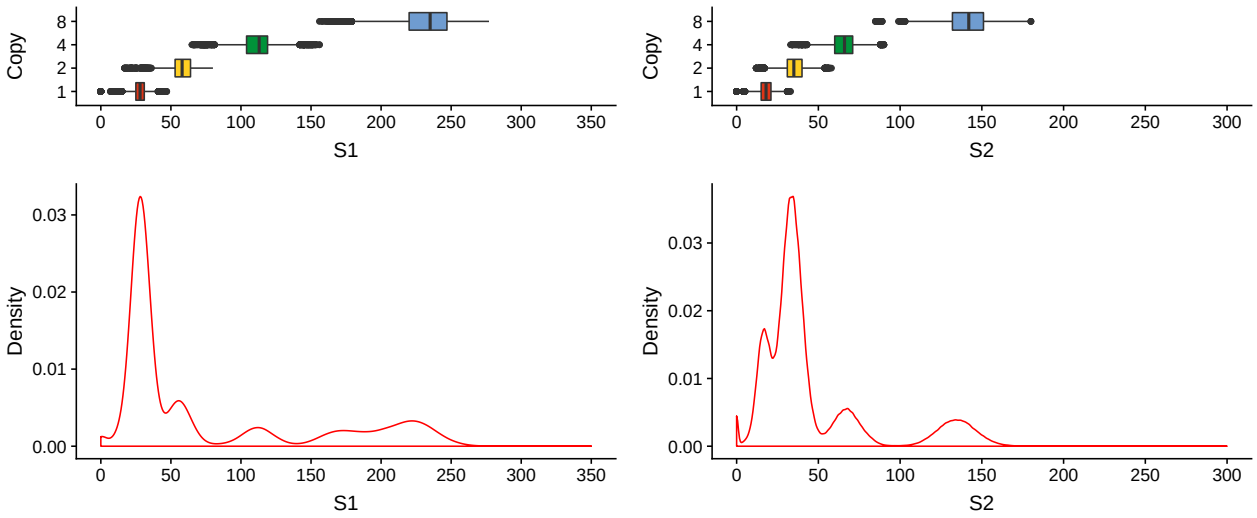

```
#The scatterplot below shows the observed counts for synthetic ladder k-mers in samples 1 and 2. The da
print(ggplot(ladder,aes(S1,S2,color=as.factor(S1.Expected_Abundance)))+
        geom_point(alpha=0.5)+
```

```
        geom_abline(color="black",linetype="dashed")+
        scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        theme(aspect.ratio = 1)+
        labs(title="Unnormalized synthetic ladder k-mers")+
        guides(color=FALSE))
```




```
# Partioning bacterial k-mers
meta <- dt[!(Type=="synthetic_ladder")]
# Getting expected ratio between samples for each k-mer
meta[,Ratio:=round(S2.Expected_Abundance/S1.Expected_Abundance,1)]
p1 <- ggplot(ladder,aes(as.factor(S1.Expected_Abundance),S1,fill=as.factor(S1.Expected_Abundance)))+
        geom_boxplot()+
        scale_fill_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        scale_y_continuous(limits=c(0,350),breaks=seq(0,350,50))+
        coord_flip()+
        labs(x="Copy")+
        guides(fill=FALSE)
p2 <- ggplot(meta,aes(S1))+
        geom_density(color="red")+
        labs(y="Density")+
        scale_x_continuous(limits=c(0,350),breaks=seq(0,350,50))
p3 <- ggplot(ladder,aes(as.factor(S2.Expected_Abundance),S2,fill=as.factor(S2.Expected_Abundance)))+
        geom_boxplot()+
        scale_fill_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
```

```
        scale_y_continuous(limits=c(0,300),breaks=seq(0,300,50))+
        coord_flip()+
        labs(x="Copy")+
        guides(fill=FALSE)
p4 <- ggplot(meta,aes(S2))+
        geom_density(color="red")+
        labs(y="Density")+
        scale_x_continuous(limits=c(0,300),breaks=seq(0,300,50))
# Relative frequency of ladders compared to meta k-mers in S1 and S2.
ggarrange(p1,p3,p2,p4, ncol = 2, nrow = 2,align = "v",heights = c(1/4, 1/2))
```

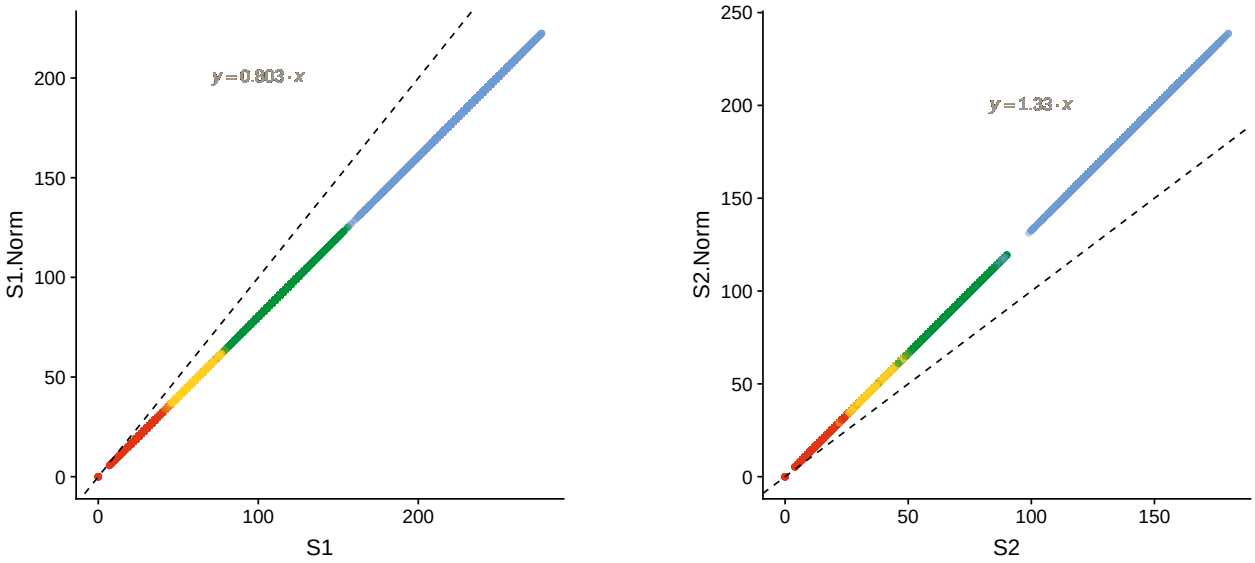


2) Normalization with and without the Synthetic Ladder

To normalize the samples with the synthetic ladder, first, I apply the normalization on synthetic ladder k-mer counts. Then, for each sample, I use a linear regression between unnormalized and normalized counts to determine scaling factors. The scaling factors are then applied to all other k-mers in the samples.

```
# Converting counts to a matrix (expected input for some of the functions)
ladder_matrix <- as.matrix(ladder[,.(S1,S2)])
meta_matrix <- as.matrix(meta[,.(S1,S2)])

# Initializing lists to store normalized ladder and bacterial counts for each normalization method
ladder_norm <- list()
meta_norm <- list()

i <- 1
# Loop through each normalization function
for (norm_name in c('uqua','deseq2','tmm','normalize.quantiles')) {
  norm <- match.fun(norm_name)
  # For DESeq2 I made a custom wrapper function found in the beginning of this document
  if (norm_name == "deseq2") {
    # I suppressed some of the output messages to streamline the output
    # Normalizing the synthetic ladder
    tmp_lad <- suppressMessages(deseq2(ladder[,.(Sequence,S1,S2)]))
    tmp_lad_norm <- tmp_lad$norm[,.(S1,S2)]
    # Normalizing bacterial k-mers directly without the ladder
    tmp_meta <- suppressMessages(deseq2(meta[,.(Sequence,S1,S2)]))
```

```
    tmp_meta_norm <- tmp_meta$norm[,.(S1,S2)]
  } else {
    # Normalizing the synthetic ladder
    tmp_lad_norm <- as.data.table(norm(ladder_matrix))
     # Normalizing bacterial k-mers directly without the ladder
    tmp_meta_norm <- as.data.table(norm(meta_matrix))
  }
  # Create new columns for the normalized counts (synthetic ladder)
  names(tmp_lad_norm) <- c("S1.Norm","S2.Norm")
  tmp_lad_norm <- cbind(tmp_lad_norm,ladder)
  # Rearrange the data.table and add a column to specify the normalization method used (synthetic ladde
  tmp_lad_norm <- tmp_lad_norm[,.(Sequence,S1,S2,S1.Norm,S2.Norm,Name,S1.Expected_Abundance,S2.Expected_
  # Add data.table to the ladder normalized counts list (synthetic ladder)
  ladder_norm[[i]] <- tmp_lad_norm

  # Create new columns for the normalized count (bacterial/without ladder)
  names(tmp_meta_norm) <- c("S1.Norm","S2.Norm")
  tmp_meta_norm <- cbind(tmp_meta_norm,meta)
  # Rearrange the data.table and add a column to specify the normalization method used (bacterial/witho
  tmp_meta_norm <- tmp_meta_norm[,.(Sequence,S1,S2,S1.Norm,S2.Norm,Name,S1.Expected_Abundance,S2.Expecte
  # Add data.table to the ladder normalized counts list (bacterial/without ladder)
  meta_norm[[i]] <- tmp_meta_norm
  i <- i + 1
}
# Collapse the synthetic ladder list of data.tables into a single one
meta_norm <- rbindlist(meta_norm)
# Collapse the bacterial (normalized without the ladder) list of data.tables into a single one
ladder_norm <- rbindlist(ladder_norm)

p1 <- ggplot(ladder_norm[Method=="tmm"],aes(S1,S1.Norm,color=as.factor(S1.Expected_Abundance)))+
        geom_point(alpha=0.5)+
        geom_abline(color="black",linetype="dashed")+
        scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        theme(aspect.ratio = 1)+
        geom_text(x = 100, y = 200, label = lm_eqn(ladder_norm[Method=="tmm",.(x=S1,y=S1.Norm)]), parse
        guides(color=FALSE)
p2 <- ggplot(ladder_norm[Method=="tmm"],aes(S2,S2.Norm,color=as.factor(S1.Expected_Abundance)))+
        geom_point(alpha=0.5)+
        geom_abline(color="black",linetype="dashed")+
        scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        theme(aspect.ratio = 1)+
        geom_text(x = 100, y = 200, label = lm_eqn(ladder_norm[Method=="tmm",.(x=S2,y=S2.Norm)]), parse
        guides(color=FALSE)
# Scatterplots showing unnormalized and normalized counts for samples 1 and 2 with the slope indicated
print(plot_grid(p1,p2,ncol=2))
```



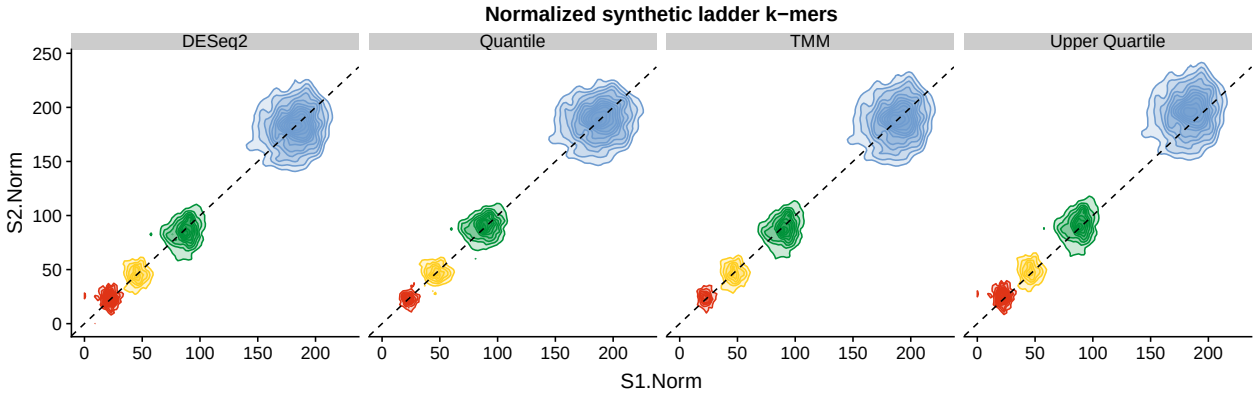
To normalize without the synthetic ladder, I apply the normalization function directly on the bacterial k-mers.

The normalizations I used are:

1) deseq2 (DESeq2 R package)
2) tmm (NOISeq R package)
3) quantile (preprocessCore R package)
4) upper quartile: I divided counts by the ratio of sample upper quartile to mean upper quartile between samples.

The scatterplots below show synthetic ladder k-mers after normalization with the 4 different methods:

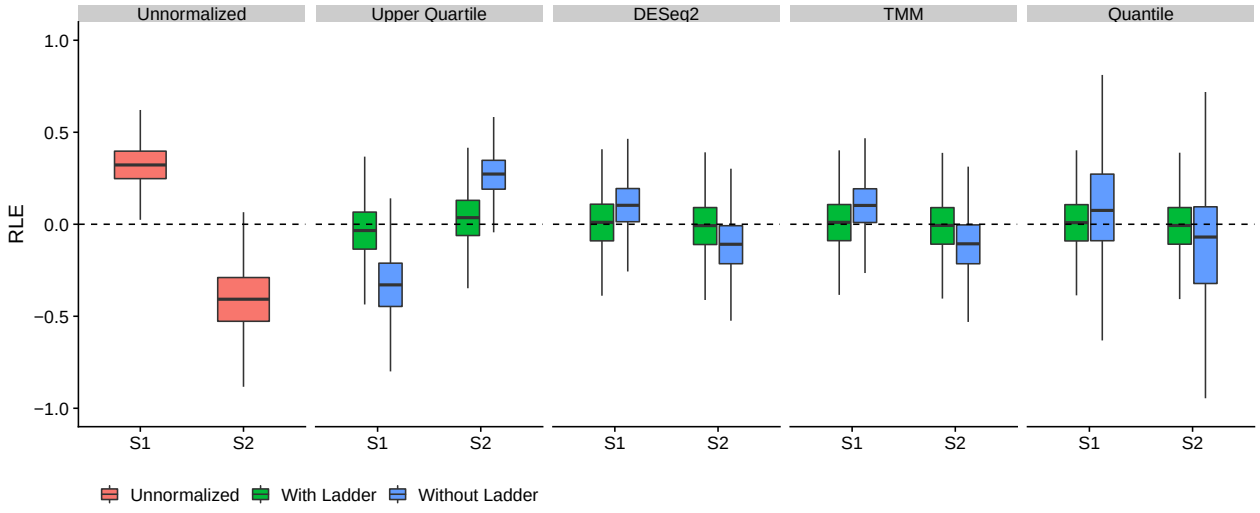
```
ladder_norm_print <- ladder_norm

# Adding proper labels to the different normalizations
ladder_norm_print$Method <- as.factor(ladder_norm_print$Method)
levels(ladder_norm_print$Method) <- c('DESeq2','Quantile','TMM','Upper Quartile')

# Scatterplots of synthetic ladder k-mers counts after normalization
print(ggplot(ladder_norm_print,aes(S1.Norm,S2.Norm,color=as.factor(S1.Expected_Abundance)))+
        stat_density_2d(aes(fill = as.factor(S1.Expected_Abundance)), geom="polygon",alpha=0.2)+
        #scale_color_manual(values=c("1"="#A84399","2"="#6F9BCF","4"="#009239","8"="#6F9CD0"))
        geom_abline(color="black",linetype="dashed")+
        facet_wrap(~Method,ncol=4)+
        scale_color_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        scale_fill_manual(values=c("1"="#DF3416","2"="#FFCF22","4"="#009239","8"="#6F9CD0"))+
        theme(aspect.ratio = 1)+
        labs(title="Normalized synthetic ladder k-mers")+
        guides(color=FALSE,fill=FALSE))
```




```
# Obtain scaling factors based on the normalization of the synthetic ladder
# For each normalization method and sample, I perform a linear regression of unnormalized and normalize
levels(ladder_norm$Method) <- c('deseq2','normalize.quantiles','tmm','uqua')
norm_factor <- ladder_norm[,.(S1.Function=list(lm(S1.Norm~S1-1)),S2.Function=list(lm(S2.Norm~S2-1))),by=

# Initializing a list to store normalized bacterial counts for each normalization method based on scali
meta_lad <- list()
i <- 1
for (norm_name in c('uqua','deseq2','tmm','normalize.quantiles')) {
  # The slope obtained from the linear regression above is used to scale the bacterial k-mers in the ea
  tmp_meta_lad <- data.table(S1.Norm=meta$S1*norm_factor[Method==norm_name,S1.Function][[1]]$coef,
                             S2.Norm=meta$S2*norm_factor[Method==norm_name,S2.Function][[1]]$coef)
  # Rearrange the data.table and add a column to specify the normalization method used (bacterial k-mer
  tmp_meta_lad <- cbind(tmp_meta_lad,meta)
  tmp_meta_lad <- tmp_meta_lad[,.(Sequence,S1,S2,S1.Norm,S2.Norm,Name,S1.Expected_Abundance,S2.Expected_
  meta_lad[[i]] <- tmp_meta_lad
  i <- i + 1
}

# Collapse the bacterial (normalized with the ladder) list of data.tables into a single one
meta_lad <- rbindlist(meta_lad)

# Create a new variable "Norm" to discriminate the counts normalized with and without the synthetic lad
meta_lad[,Norm:="with_ladder"]
meta_norm[,Norm:="without_ladder"]

# Join bacterial k-mers normalized with and wihtout the synthetic ladder
meta_all <- rbind(meta_norm,meta_lad)

# Get expected ratio between samples for each k-mer
meta_all[,Ratio:=round(S2.Expected_Abundance/S1.Expected_Abundance,1)]

# Instead of having different columns for unnormalized (e.g. S1) and normalized (e.g. S1.Norm)
# I put all the counts in the same column and create a new "Norm" factor for unnormalized counts
meta_all <- rbind(meta[,.( Sequence,S1,S2,Name,S1.Expected_Abundance,S2.Expected_Abundance,
                Type,Method="unnormalized",Norm="unnormalized",Ratio)],
      meta_all[,.( Sequence,S1=S1.Norm,S2=S2.Norm,Name,S1.Expected_Abundance,S2.Expected_Abundance,
                  Type,Method,Norm,Ratio)])

# Calculate the observed difference in counts for samples 1 and 2
meta_all[,Diff:=S2-S1]
```

The RLE plots below were calculated based on negative control k-mers, which should have the same expected abundance between the samples.

```
# Making RLE plots to show the impact of normalization on k-mers whose expected abundance was the same

# Calculate average count for each k-mer
meta_all[,AvgCount:=(S1+S2)/2]
# Calculate ratio between normalized counts and average counts
rle <- meta_all[Ratio==1,.(S1=log2(S1)-log2(AvgCount),S2=log2(S2)-log2(AvgCount),Method,Norm)]
# Sampling 30000 points for each group to avoid slowness when plotting
rle <- rle[,.SD[sample(.N,min(.N,30000))],by=list(Method,Norm)]
rle <- melt(rle,measure.vars = c('S1','S2'),variable.name = 'Sample', value.name = 'RLE')
# Renaming some of the categorical variables
rle$Method <-factor(rle$Method,levels=c("unnormalized",'uqua','deseq2','tmm','normalize.quantiles'))
levels(rle$Method) <- c("Unnormalized",'Upper Quartile','DESeq2','TMM','Quantile')
rle$Norm <- as.factor(rle$Norm)
levels(rle$Norm) <- c('Unnormalized','With Ladder','Without Ladder')
# Making RLE plots
suppressWarnings(print(ggplot(rle,aes(Sample,RLE,fill=Norm))+
        geom_boxplot(outlier.shape = NA,width=0.5)+
        facet_grid(~Method)+
        scale_y_continuous(limits = c(-1,1))+
        theme(legend.position = "bottom")+
        geom_hline(yintercept = 0,linetype="dashed",color="black")+
        labs(x=NULL,fill=NULL)))
```



3) Estimating variation from the Syntethic Ladder

To estimate variability in counts at each level of the synthetic ladder, I calculate the obsersed difference in count between the samples after normalization. I then calculate the standard deviation for this variable at each level of the synthetic ladder.

```
# Calculate average normalized count and standard deviation at each level for each sample
# Also calculate the average count and standard deviation for the observed difference in normalized cou
ladder_summary <- ladder_norm[,.(Sequence,S1=S1.Norm,S2=S2.Norm,Diff=S2.Norm-S1.Norm,Expected_Abundance=
ladder_summary <- melt(ladder_summary,measure.vars = c('S1','S2','Diff'),variable.name = 'Variable',valu
ladder_summary <- ladder_summary[,.(Mean=mean(Value),SD=sd(Value)),by=list(Method,Expected_Abundance,Va
```

4) Identifying differential k-mers between libraries

Then, for each bacterial k-mer, in each sample, I find the closest level in the synthetic ladder. If the ladder levels are different for any given k-mer, I attritube the lowest level in the ladder. I calculate the observed difference in counts for bacterial k-mers and use the standard deviation derived from the ladder to estimate a significance associated with that difference (with a t-test).

The table below shows the specificity and sensitivity associated with the different normalizations:

```
# For each bacterial k-mer I need to find the corresponding level of the synthetic ladder and assign th
meta_all_cn <- list()
i <- 1
# I performed this procedure for each normalization method
for (norm_name in c('uqua','deseq2','tmm','normalize.quantiles')) {
  # For each normalization, I subset counts that are unnormalized, normalized with the ladder and norma
  tmp <- meta_all[Method %in% c('unnormalized',norm_name)]
  tmp[,I:=seq(1,.N)]
  # Then I get the average ladder counts for sample 1 after normalizing with that given method
  tmp_lad <- ladder_summary[Variable=="S1" & Method==norm_name,]
  # In the next lines for each k-mer I find the closest level in the ladder (1,2,4, and 8)
  tmp_lad <- rbind(tmp_lad,tmp_lad[,.(Method="unnormalized",Expected_Abundance,Variable,Mean,SD)])
  tmp <- merge(tmp,tmp_lad,by="Method",allow.cartesian=TRUE)
  tmp[,S1.CN_Level:=S1-Mean]
  tmp[,Min:=min(abs(S1.CN_Level)),by=I]
  tmp[,Min:=(Min==abs(S1.CN_Level))]
  # I have the CN level for each bacterial k-mer in sample 1
  tmp1 <- tmp[Min==TRUE,.(Sequence,S1,S2,Name,S1.Expected_Abundance,S2.Expected_Abundance,Type,Method,N
  # I do the same as above for sample 2
  tmp <- meta_all[Method %in% c('unnormalized',norm_name)]
  tmp[,I:=seq(1,.N)]
  tmp_lad <- ladder_summary[Variable=="S2" & Method==norm_name,]
  tmp_lad <- rbind(tmp_lad,tmp_lad[,.(Method="unnormalized",Expected_Abundance,Variable,Mean,SD)])
  tmp <- merge(tmp,tmp_lad,by="Method",allow.cartesian=TRUE)
  tmp[,S2.CN_Level:=S2-Mean]
  tmp[,Min:=min(abs(S2.CN_Level)),by=I]
  tmp[,Min:=(Min==abs(S2.CN_Level))]
  tmp2 <- tmp[Min==TRUE,.(Sequence,S1,S2,Name,S1.Expected_Abundance,S2.Expected_Abundance,Type,Method,N
  # Then for each k-mer I have the CN level in sample 1 and CN level in sample2
  tmp1$S2.CN_Level <- tmp2$S2.CN_Level
  tmp1[,Method:=norm_name]
  # I can then assign the appropriate standard deviation
  # In case where the CN levels in samples 1 and 2 were different, I used the lower CN level as referen
  tmp1[,CN_Level:=min(S1.CN_Level,S2.CN_Level),by=I]
  tmp1[,I:=NULL]
  # I assigned the standard deviation derived from the observed difference in counts for ladder k-mers
  tmp_lad <- ladder_summary[Variable=="Diff" & Method==norm_name,]
  tmp1 <- merge(tmp1,unique(tmp_lad[,.(CN_Level=Expected_Abundance,SD)]),by="CN_Level",allow.cartesian=
  meta_all_cn[[i]] <- tmp1
  i <- i + 1
}
# Collapse the list of data.tables into a single one
meta_all_cn <- rbindlist(meta_all_cn)

# Calculate t-test of the observed difference in counts between samples 1 and 2
# using the standard deviation derived from the ladder
```

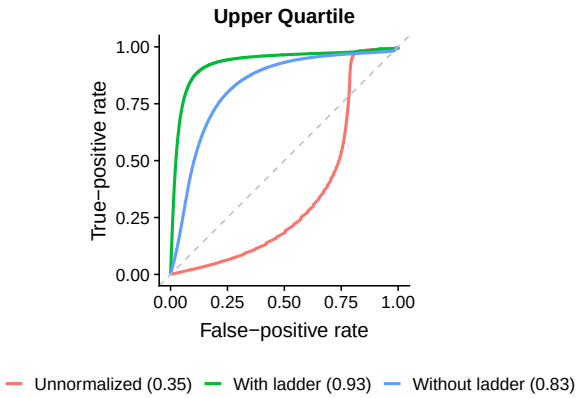
```
# When the difference between samples was greater than 0 I calculated the p-value as P[X > x].
meta_all_cn[Diff>0,P.value:=1-pnorm(Diff, mean =0, sd = SD)]
# if the difference is smaller than or equal to 0 I calculated the p-value as P[X   x]
meta_all_cn[Diff<=0,P.value:=pnorm(Diff, mean =0, sd = SD)]

# To build ROC curves I'm creating a variable that tells whether a k-mer was supposed to be a negative
# 0 = positive control and 1 = negative control
meta_all_cn[,Status:=0]
meta_all_cn[Ratio == 1,Status:=1]

# Calculating the sensitivity and specificity associate with each method
significance <- 0.05
meta_all_cn[,Significance:=FALSE]
meta_all_cn[,P.value.adjust:=p.adjust(P.value,method="fdr")]
meta_all_cn[P.value.adjust < 0.05,Significance:=TRUE]
total <- meta_all_cn[,.(Total=.N),list(Method,Norm,Status)]
stats <- meta_all_cn[,.N,list(Method,Norm,Status,Significance)]
stats <- merge(stats,total,by=c("Method","Norm","Status"))
stats[,N:=N/Total]
stats$Status <- as.factor(stats$Status)
levels(stats$Status) <- c("positive","negative")
stats$Significance <- as.factor(stats$Significance)
levels(stats$Significance) <- c("non-significant","significant")
levels(stats$Significance) <- c("non-significant","significant")
stats <- data.table(dcast(stats,Method+Norm+Status~Significance,value.var="N"))
stats <- stats[,.(Specificity=`non-significant`[2],Sensitivity=significant[1]),by=list(Method,Norm)]
stats$Method <- as.factor(stats$Method)
levels(stats$Method) <- c('DESeq2','Quantile','TMM','Upper Quartile')
stats$Norm <- as.factor(stats$Norm)
levels(stats$Norm) <- c('Unnormalized','With ladder','Without ladder')
names(stats) <- c('Normalization','Type','Specificity','Sensitivity')
print(kable(stats))
```

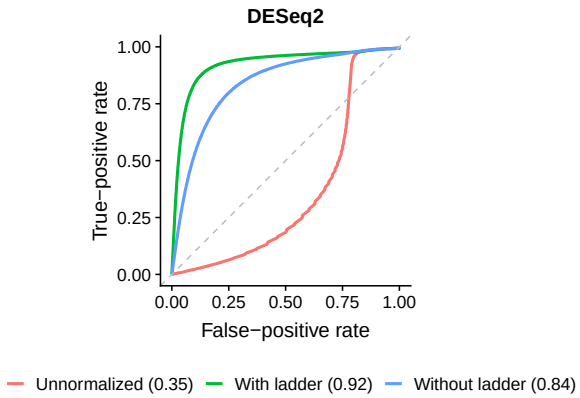
```
##
##
## Normalization    Type              Specificity   Sensitivity
## ---------------  ---------------  ------------  ------------
## DESeq2           Unnormalized        0.3708458     0.3276772
## DESeq2           With ladder         0.9167322     0.8105178
## DESeq2           Without ladder      0.8525436     0.6800519
## Quantile         Unnormalized        0.3744014     0.3279658
## Quantile         With ladder         0.9158399     0.8083935
## Quantile         Without ladder      0.6806090     0.4097115
## TMM              Unnormalized        0.3685642     0.3279636
## TMM              With ladder         0.9168993     0.8120150
## TMM              Without ladder      0.8562619     0.6867308
## Upper Quartile   Unnormalized        0.3659031     0.3275622
## Upper Quartile   With ladder         0.9117647     0.8482758
## Upper Quartile   Without ladder      0.4526362     0.9082512
```

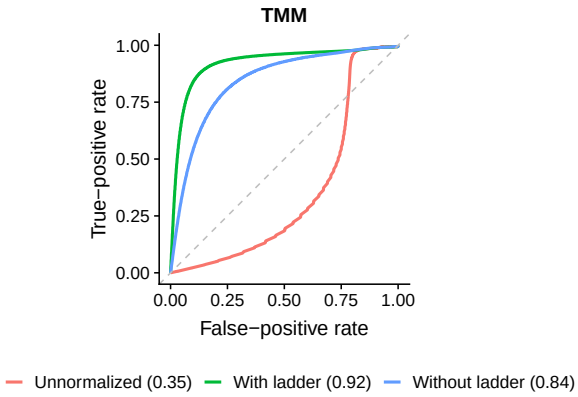
The ROC curves below were calculated based on the p-value derived from the t-test:

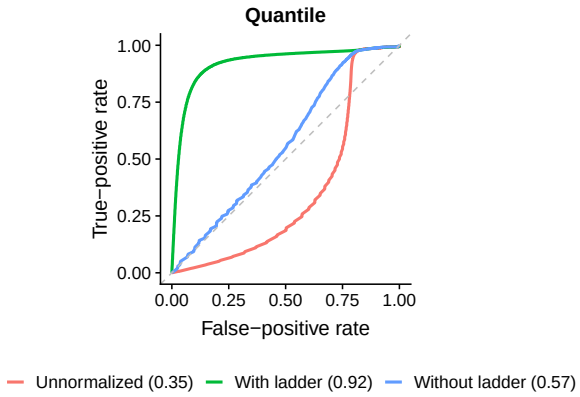
```
# Making ROC-curves of ranking the k-mers based on the p-value calculated from the ladder
plots <- list()
```

```
i <- 1
labels <- c('Upper Quartile','DESeq2','TMM','Quantile')
for (norm_name in c('uqua','deseq2','tmm','normalize.quantiles')) {
  tmp <- meta_all_cn[Method==norm_name]
  tmp$Norm <- as.factor(tmp$Norm)
  roc_plot <- ggplot(tmp,aes(d=Status,m=P.value.adjust,color=Norm))+
    geom_roc(n.cuts = 0)+
    coord_fixed()+
    geom_abline(slope=1,color="gray",linetype="dashed")+
    labs(x="False-positive rate",y="True-positive rate",title=norm_name)
  # Getting area under curve
  auc <- calc_auc(roc_plot)
  auc$AUC <- round(auc$AUC,2)
  auc$Norm <- c("unnormalized","with_ladder","without_ladder")
  tmp <- merge(tmp,auc,by="Norm")
  tmp$Norm <- as.factor(tmp$Norm)
  levels(tmp$Norm) <- c("Unnormalized","With ladder","Without ladder")
  tmp$Label <- paste(tmp$Norm," (",tmp$AUC,")",sep="")
  plots[[i]] <- ggplot(tmp,aes(d=Status,m=P.value.adjust,color=Label))+
    geom_roc(n.cuts = 0)+
    coord_fixed()+
    geom_abline(slope=1,color="gray",linetype="dashed")+
    theme(legend.position = "bottom")+
    labs(x="False-positive rate",y="True-positive rate",color=NULL,title=labels[i])

  i <- i + 1
}
print(plot_grid(plotlist=plots,ncol=2))
```


Upper Quartile
True−positive rate
False−positive rate
0.00
0.25
0.50
0.75
1.00
Unnormalized (0.35)
With ladder (0.93)
Without ladder (0.83)


DESeq2
True−positive rate
False−positive rate
0.00
0.25
0.50
0.75
1.00
Unnormalized (0.35)
With ladder (0.92)
Without ladder (0.84)


TMM
True−positive rate
False−positive rate
0.00
0.25
0.50
0.75
1.00
Unnormalized (0.35)
With ladder (0.92)
Without ladder (0.84)


Quantile
True−positive rate
False−positive rate
0.00
0.25
0.50
0.75
1.00
Unnormalized (0.35)
With ladder (0.92)
Without ladder (0.57)