TMM Normalization

Code

https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-3-r25
(https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-3-r25)

Hide

```
library(edgeR)
library(ggplot2)
library(reshape2)


ndx <- which(colnames(dt_rna_cod) %in% rnaseq_meta_clin$sample_names)
tmp <- as.matrix(dt_rna_cod[, ndx, with = FALSE])
rownames(tmp) <- dt_rna_cod$gene_id

counts <- tmp
colData <- rnaseq_meta_clin

# Step 1: Create DGEList object and calculate TMM normalization factors
dge <- DGEList(counts = counts,group = colData$Cohort)
dge <- calcNormFactors(dge, method = "TMM")
norm_counts <- cpm(dge, normalized.lib.sizes = TRUE)

# Step 2: Estimate dispersion, fit the GLM, and perform LRT
dge <- DGEList(counts = counts, group = colData$Cohort)
dge <- calcNormFactors(dge, method = "TMM")
design <- model.matrix(~ colData$Cohort)
dge <- estimateDisp(dge, design)
fit <- glmFit(dge, design)
lrt <- glmLRT(fit)

# Step 3: Extract and filter DE genes
top_genes <- topTags(lrt, n = 50)$table
filtered_genes <- top_genes[
  abs(top_genes$logFC) > 1 &
  top_genes$PValue < 0.05 &
  top_genes$FDR < 0.1,
]
round(filtered_genes, 6)
```

	logFC <dbl>	logCPM <dbl>	LR <dbl>	PValue <dbl>	FDR <dbl>
ENSG00000119630	-3.083226	1.357485	43.91500	0.000000	0.000001
ENSG00000204020	-1.357607	3.077190	20.48668	0.000006	0.029884
ENSG00000102109	-1.805964	4.318450	18.25757	0.000019	0.058578
ENSG00000205846	-1.414078	0.732552	17.73264	0.000025	0.058578

	logFC <dbl>	logCPM <dbl>	LR <dbl>	PValue <dbl>	FDR <dbl>
ENSG00000134363	3.707579	-2.990114	17.65490	0.000026	0.058578
ENSG00000179058	3.474074	-3.397633	17.17457	0.000034	0.067877
ENSG00000114405	-1.065656	0.299058	16.82590	0.000041	0.068037
ENSG00000171659	-1.384973	1.491211	16.75535	0.000043	0.068037
ENSG00000170819	1.753804	-1.192185	16.53151	0.000048	0.068037
ENSG00000239264	1.653957	2.926370	16.10262	0.000060	0.074652

1-10 of 13 rows Previous 1 2 Next

Hide

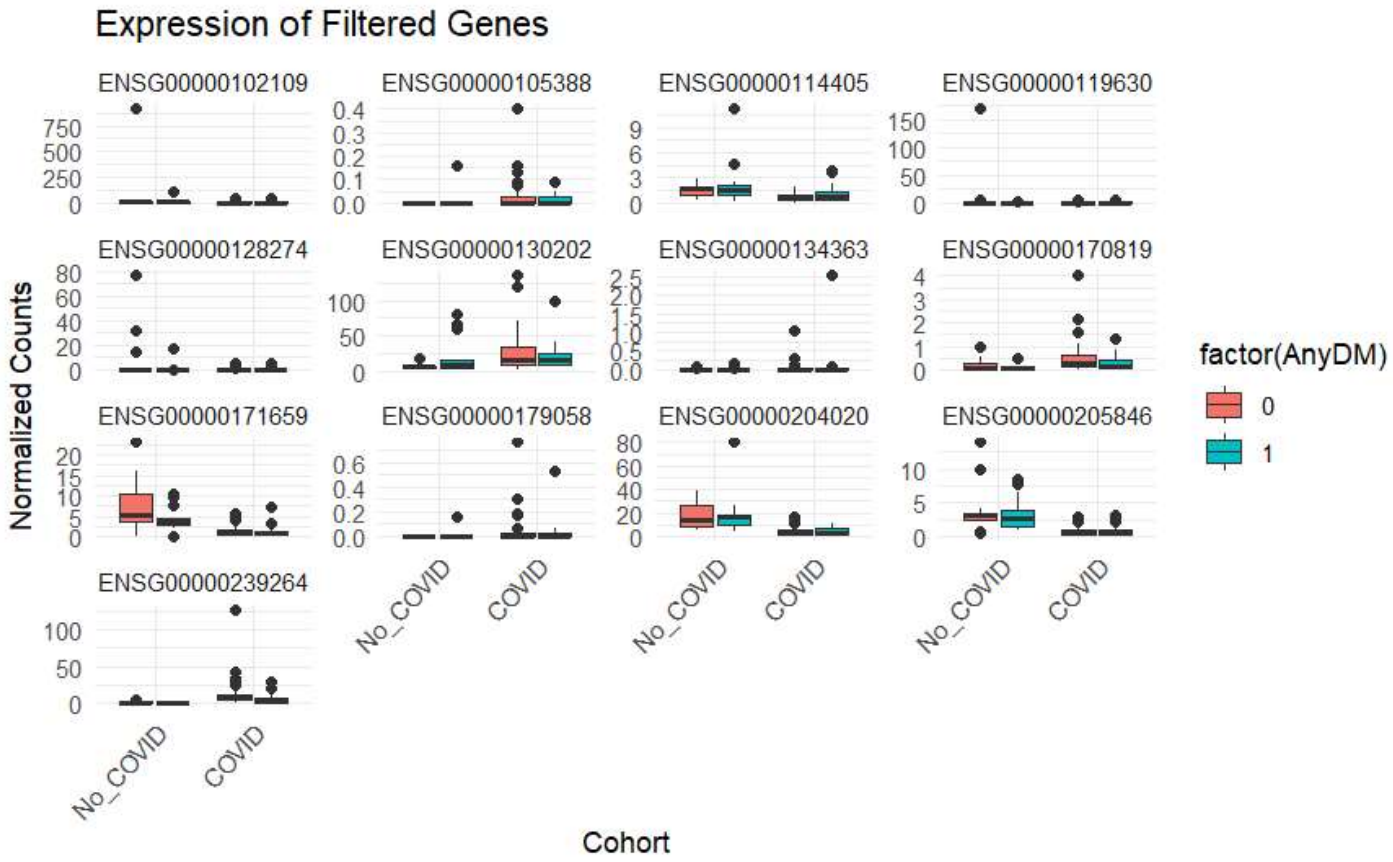
```
filtered_gene_ids <- rownames(filtered_genes)

# Step 4: Extract normalized counts for filtered genes
norm_counts_filtered <- norm_counts[filtered_gene_ids, ]

# Convert to long format for ggplot2
norm_counts_df <- as.data.frame(norm_counts_filtered)
norm_counts_df$gene <- rownames(norm_counts_df)
norm_counts_long <- melt(norm_counts_df, id.vars = "gene")

# Add sample information to the long format data
norm_counts_long <- merge(norm_counts_long, colData, by.x = "variable", by.y = "sample_names")

# Step 5: Create box plots for each gene
ggplot(norm_counts_long, aes(x = factor(Cohort), y = value, fill = factor(AnyDM))) +
  geom_boxplot() +
  facet_wrap(~ gene, scales = "free_y") +
  labs(title = "Expression of Filtered Genes",
       y = "Normalized Counts",
       x = "Cohort") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Expression of Filtered Genes
ENSG00000102109
ENSG00000105388
ENSG00000114405
ENSG00000119630
ENSG00000128274
ENSG00000130202
ENSG00000134363
ENSG00000170819
ENSG00000171659
ENSG00000179058
ENSG00000204020
ENSG00000205846
ENSG00000239264
Normalized Counts
No_COVID
COVID
Cohort
factor(AnyDM)
0
1