Code used data provided by Sun et al. The top 500 differentially expressed genes were plotted and cluster profiling was performed. Data is partially visualized in Figure 1f. Gene ontology was performed using WormCat 2.0 (wormcat.com).

# Load necessary packages

library(edgeR)

library(gplots)

# Import the data

gene\_data <- read.csv("\*\*\*.csv", header = TRUE)

# Extract gene names

gene\_names <- gene\_data[, 1]

# Extract count data

condition\_counts <- gene\_data[, -1]

# Convert count data to matrix

count\_matrix <- as.matrix(condition\_counts)

# Create a DGEList object

dge <- DGEList(counts = count\_matrix, genes = gene\_names)

# Normalize and estimate dispersions

dge <- calcNormFactors(dge)

dge <- estimateCommonDisp(dge)

dge <- estimateTagwiseDisp(dge)

# Design matrix and contrasts (comparing 2 and 400 to 20)

design <- model.matrix(~0 + factor(rep(c("condition20", "condition2", "condition400"), each = 3)))

colnames(design) <- c("condition20", "condition2", "condition400")

# Define contrasts (comparing 2 and 400 to 20)

contrasts <- makeContrasts(condition2 - condition20, condition400 - condition20, levels = design)

# Fitting the generalized linear model

dge\_fit <- glmFit(dge, design)

# Likelihood ratio test

dge\_contrast <- glmLRT(dge\_fit, contrast = contrasts)

# Get the top differentially expressed genes

top\_genes <- topTags(dge\_contrast, n = 500)$table

# Extract expression data for top genes

top\_gene\_names <- top\_genes$genes

top\_gene\_expression <- count\_matrix[which(gene\_names %in% top\_gene\_names), ]

# Normalize expression data (z-score normalization)

normalized\_expression <- scale(top\_gene\_expression)  
#Create distance matrix

dist\_matrix <- dist (log2\_rnaseq\_data, method = “euclidean”)

#Perform hierarchical clustering

Hc\_result <- hclust (dist\_matrix, method = “complete”)

Num\_clusters <- 3

Cluster\_cut <- cutree(hc\_result, k = num\_clusters)