Data Preparation, Heatmaps, and Differential Gene Expression Analysis

Your Name

2023-05-29

Introduction

In this tutorial, we are going to learn how to prepare data, create heatmaps, and perform differential gene expression analysis in R.

Libraries

First, we will load the necessary libraries.

```
library(readr)
library(reshape2)
library(pheatmap)
library(RColorBrewer)
library(viridis)
library(DESeq2)
```

Data Preparation

We define a function `PrepareData` to prepare the data. This function takes in a filename and a path to a gene symbol mapping file, reads the file into a dataset, renames the first column to "Genes", loads the gene symbol mapping information, and then replaces gene IDs in the dataset with corresponding gene symbols.

```
PrepareData <- function(filename, libpath) {
  dataset <- read_csv(filename)
  names(dataset)[1] <- "Genes"
  entrez.cja <- readRDS(libpath)
  idx <- match(dataset$Genes, entrez.cja$gene_id)
  dataset$Genes <- entrez.cja$symbol[idx]
  return(dataset)
}
```

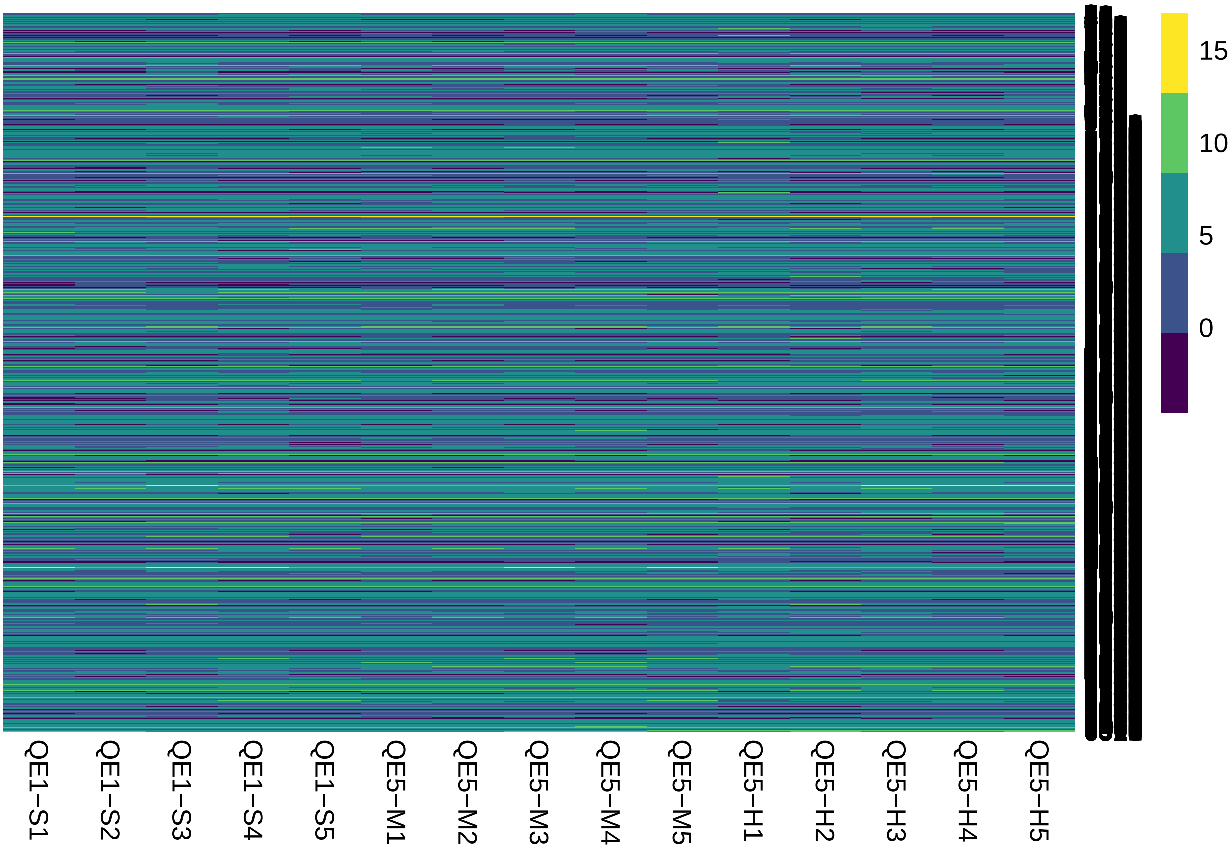
We then specify the path for the gene symbol mapping information and prepare the normalized and non-normalized datasets.

```
libpath <- "data/entrez.rds"
dataset.norm <- PrepareData("data/cjaponica_data_normalized.csv", libpath)
dataset.nonorm <- PrepareData("data/cjaponica_data.csv", libpath)
```

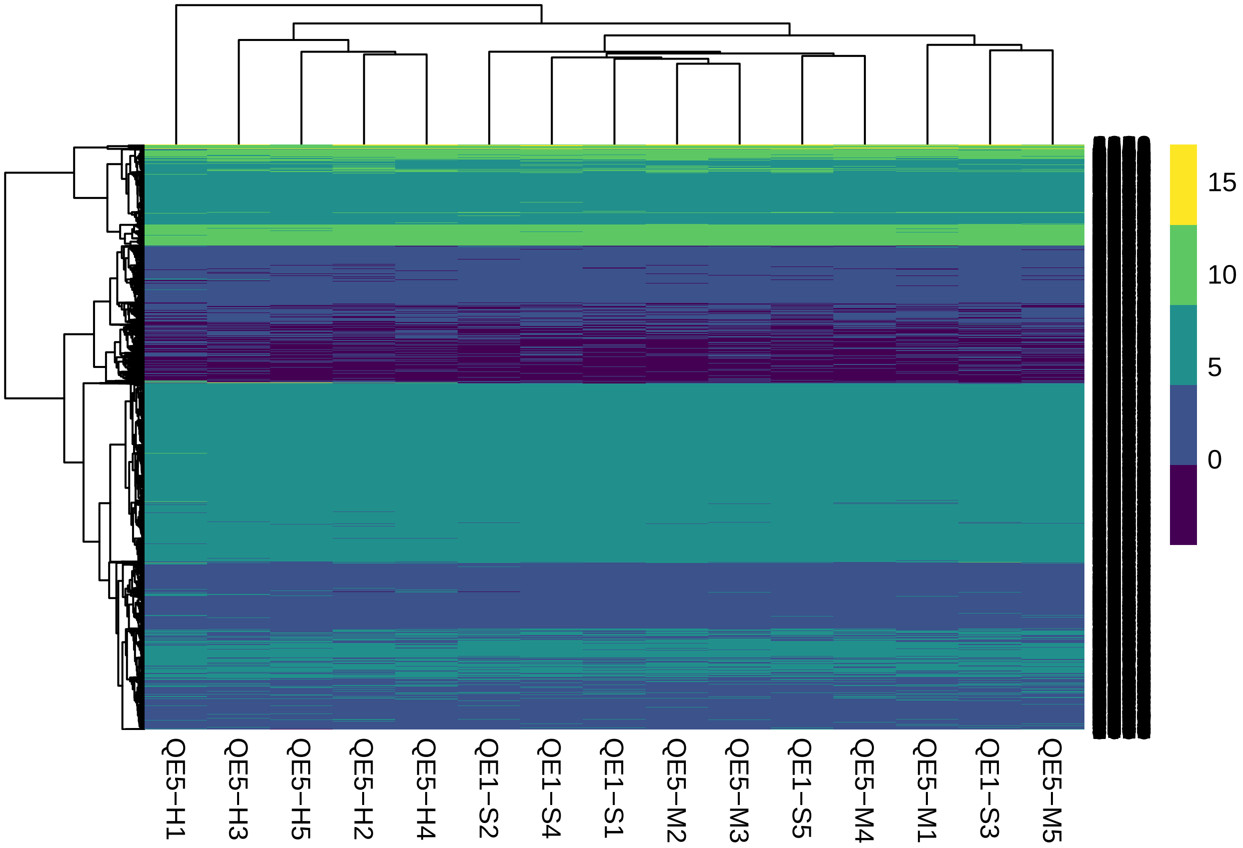
Creating Heatmaps

We set the seed for reproducibility and then create two heatmaps: one without row and column clustering, and another with row and column clustering.

```
set.seed(123)
pheatmap(dataset.norm[,-1], color = viridis(5), cluster_rows = FALSE, cluster_cols = FALSE, fontsize_nu
```



```
pheatmap(dataset.norm[,-1], color = viridis(5), cluster_rows = TRUE, cluster_cols = TRUE, fontsize_numb
```



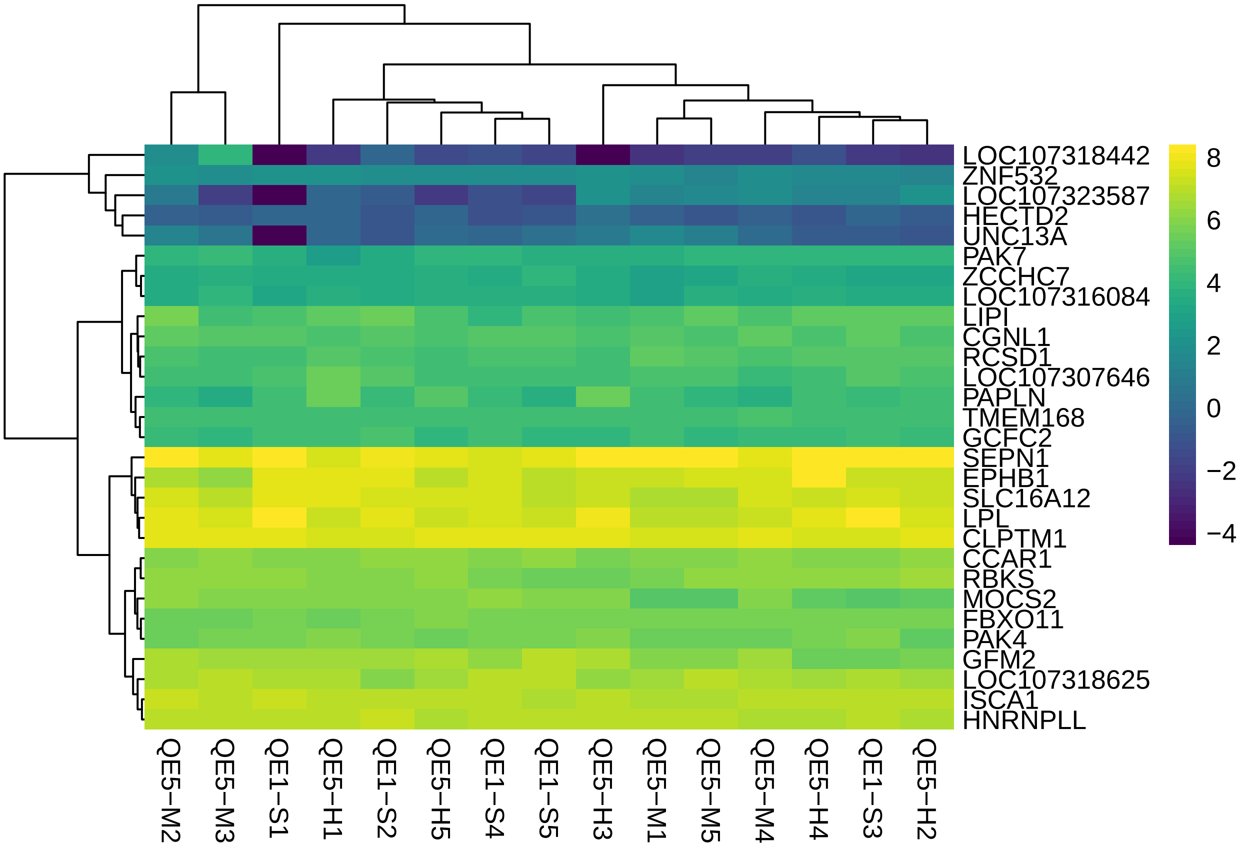
Differential Gene Expression Analysis

We define sample conditions, create a DESeq dataset object, run DESeq2 analysis, get the 30 genes with smallest p-values, and remove genes with missing symbols.

```
colData <- DataFrame(condition = factor(rep(c("Control", "Medium", "High"), each = 5)))
dds <- DESeqDataSetFromMatrix(countData = round(dataset.nonorm[,-1],0), colData = colData, design = ~ c
dds <- DESeq(dds)
res <- results(dds)
top_genes <- head(order(res$pvalue), 30)
gene.names <- dataset.norm[top_genes,1]
top_genes <- top_genes[!is.na(unlist(gene.names))]
```

Finally, we extract the differentially expressed genes (DEGs) from the normalized dataset and create a heatmap with row and column clustering for these DEGs.

```
dataset.norm.deg <- dataset.norm[top_genes,]
dataset.norm.deg <- as.data.frame(dataset.norm.deg)
rownames(dataset.norm.deg) <- unlist(dataset.norm.deg[,1])
pheatmap(dataset.norm.deg[,-1], color = viridis(50), cluster_rows = TRUE, cluster_cols = TRUE, fontsize_
         border_color=NA, labels_row = dataset.norm.deg[,1])
```


LOC107318442
ZNF532
LOC107323587
HECTD2
UNC13A
PAK7
ZCCHC7
LOC107316084
LIPI
CGNL1
RCSD1
LOC107307646
PAPLN
TMEM168
GCFC2
SEPN1
EPHB1
SLC16A12
LPL
CLPTM1
CCAR1
RBKS
MOCS2
FBXO11
PAK4
GFM2
LOC107318625
ISCA1
HNRNPLL
QE5−M2
QE5−M3
QE1−S1
QE5−H1
QE1−S2
QE5−H5
QE1−S4
QE1−S5
QE5−H3
QE5−M1
QE5−M5
QE5−M4
QE5−H4
QE1−S3
QE5−H2
8
6
4
2
0
−2
−4