Gene Expression Heatmap and Boxplot Analysis

Cemile Balkas

2024-12-01

Introduction

This report aims to process expression levels of a given set of genes by extracting information from provided files. The process involves generating two heatmaps and a box-plot to visualise the different expression levels for different treatment groups.

Loading Files

The initial files include three CSV files with information on the expression levels of all genes, gene annotations, and sample annotations. These files are loaded using a custom function, where the row names are set to align with the loaded data for proper referencing during analysis. Additionally, the function does not move forward if the file does not exist in the working directory.

```
cat(
  "Loading files: data_all.csv, gene_annotation.csv, and sample_annotation.csv...\n"
)

# Custom function to load csv files
load_files <- function(file_name) {
  stopifnot(file.exists(file_name))
  return(read.csv(file_name, sep = ",", row.names = 1))
}
data_all <- load_files("./data/data_all.csv")
gene_annotations <- load_files("./data/gene_annotation.csv")
sample_annotation <- load_files("./data/sample_annotation.csv")
cat("All files loaded successfully.\n")
```

Loading Gene List

The gene list is provided in a text format and consists of numeric identifiers of each gene. It is converted to a numeric vector and entries that return NAs are removed. Checks are included to ensure the gene list has more than one gene to allow further data processing.

```
cat("Loading gene list from genelist.txt...\n")

# Ensure the gene list file exists
stopifnot(file.exists("./data/genelist.txt"))

# Read and clean the gene list, code adapted from:
# https://stackoverflow.com/questions/13967063/remove-duplicated-rows
gene_list <- readLines("./data/genelist.txt")
stopifnot(length(gene_list) > 1)

# Convert gene list to numeric and remove NAs, code adapted from:
```

```r
# https://www.statology.org/na-omit-in-r/
gene_list_numeric <- suppressWarnings(as.numeric(gene_list))
if (any(is.na(gene_list_numeric))) {
  cat("Warning: Some entries in the gene list could not be converted to a number and will be removed.\n
}
gene_list_clean <- na.omit(gene_list_numeric)

# Remove any duplicates and sort, and ensure there are enough entires to proceed
gene_list_sorted <- unique(sort(gene_list_clean))
stopifnot(length(gene_list_sorted) > 1) # Makes sure there are still genes left

cat(
  "Gene list loaded and cleared successfully. Number of unique genes:",
  length(gene_list_sorted), "\n"
)
```

Log Scaling of data

To emphasize the doubling of expression rather than subtle changes the data values are transformed to the log2 scale with an offset of +1 in case there are some zero values in the data.

```r
cat("Applying log2 transformation to the expression data...\n")

# Log2 transformation of the expression data, code adapted from lecture code
stopifnot(is.data.frame(data_all))
data_all_log <- log2(data_all + 1)

cat("Log2 transform applied successfully.\n")
```

Data Extraction

The data set is filtered to only include the data for genes from the gene list to focus on the set of genes that are of interest for this analysis.

```r
cat("Filtering expression data to include only genes from the gene list...\n")

# Filter the all gene dataset, code adapted from
# https://csu-r.github.io/Module1/indexing.html and
# https://r4ds.had.co.nz/transform.html
data_gene_list <- data_all_log[rownames(data_all_log) %in% gene_list_sorted, ]
stopifnot(nrow(data_gene_list) > 0)

cat(
  "Completed data extraction. Number of genes in the filtered dataset:",
  nrow(data_gene_list), "\n"
)
```

Gene and Sample Annotation

Gene numbers in the filtered data are replaced with their corresponding "LongName" from the gene_annotations data frame. Additionally, a new vector is created for each gene according to the type of gene it is (XA, XB or XC). Each sample is also annotated for the treatment group that the sample is derived from. These are saved as data frames.

```r
cat("Assigning gene and sample annotations...\n")

rownames(data_gene_list) <- gene_annotations[
  rownames(gene_annotations) %in% gene_list_sorted,
]$LongName

# Annotate each gene type code adapted from:
# https://youtu.be/pTeTH9bz-_s?si=yYxT2MHkHhiu8rN_&t=1774
gene_types <- gene_annotations[
  rownames(gene_annotations) %in% gene_list_sorted,
]$Type

row_annotations <- data.frame(GeneType = gene_types)

# annotate each sample with their treatment groups
col_annotations <- data.frame(TreatmentGroup = sample_annotation$TreatmentGroup)

cat("Gene and sample annotation complete.\n")
```

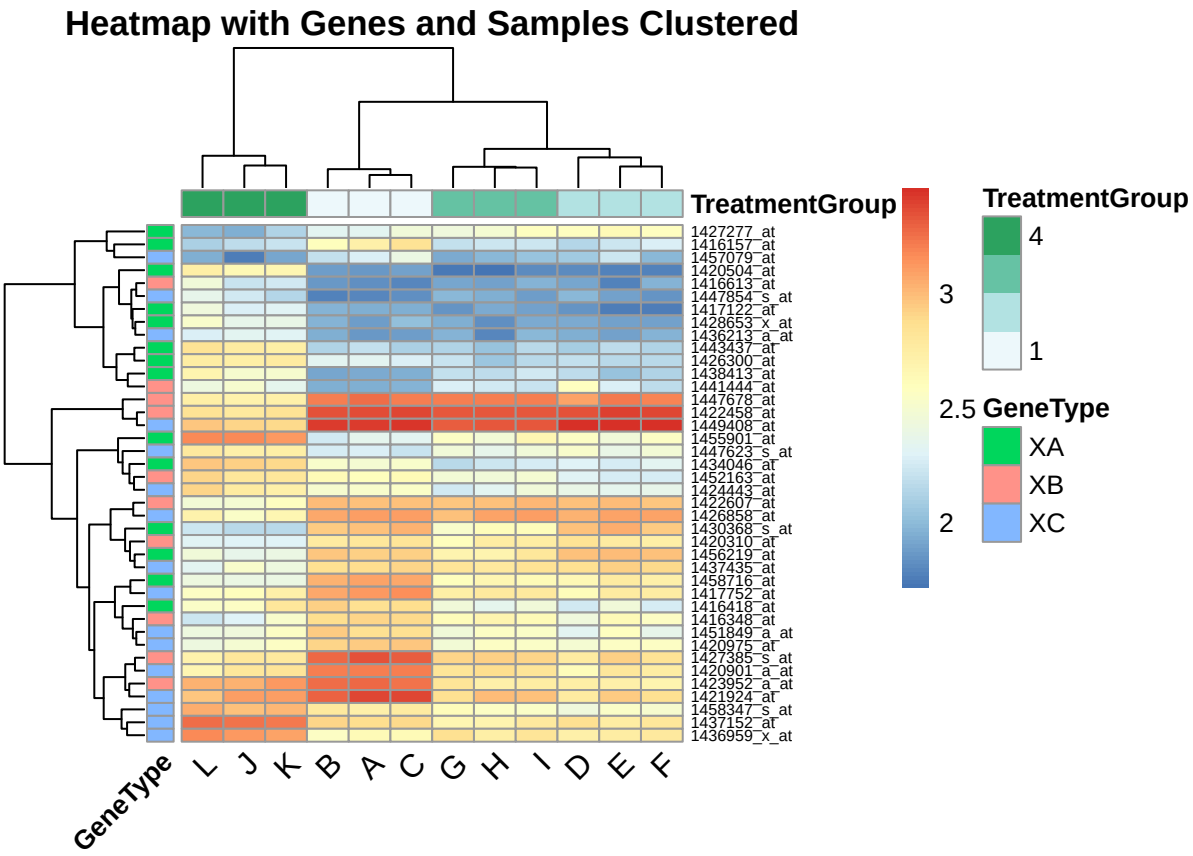
Prepare data for Heatmaps

The gene list data is converted to an expression matrix and row and column annotations for heatmaps are set using the row and column names of the expression matrix.

```r
cat("Converting data to expression matrix for heatmaps...\n")

# Convert data to matrix format
expression_matrix <- as.matrix(data_gene_list)
rownames(row_annotations) <- rownames(expression_matrix)
rownames(col_annotations) <- colnames(expression_matrix)

cat("Expression matrix created. \n")
```

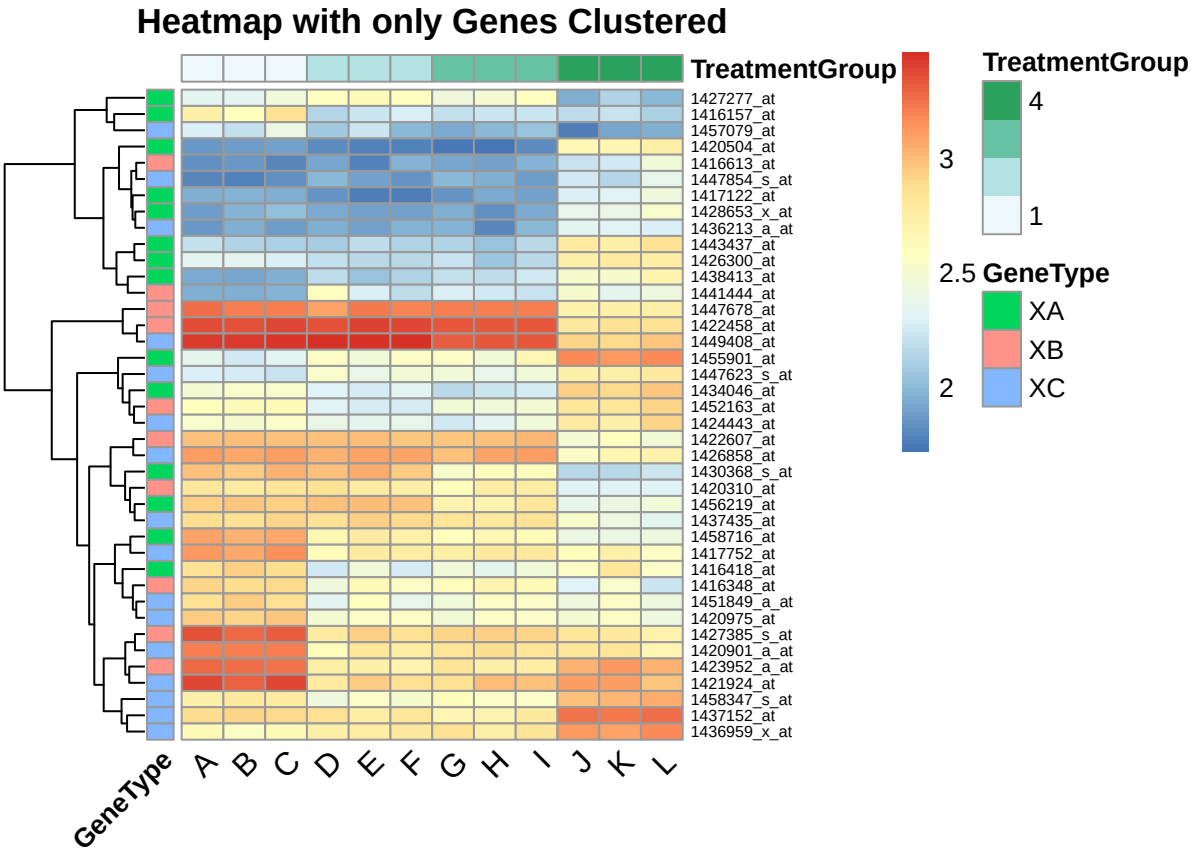
Heatmap with Genes and Samples Clustered

The first heatmap is generated using the pheatmap package, clustering both genes and samples.



Heatmap with only Genes Clustered

The second heatmap is generated where only the genes are clustered, allowing for a clearer comparison of sample-level patterns without reordering the Treatment groups.



Data preparation for Boxplot

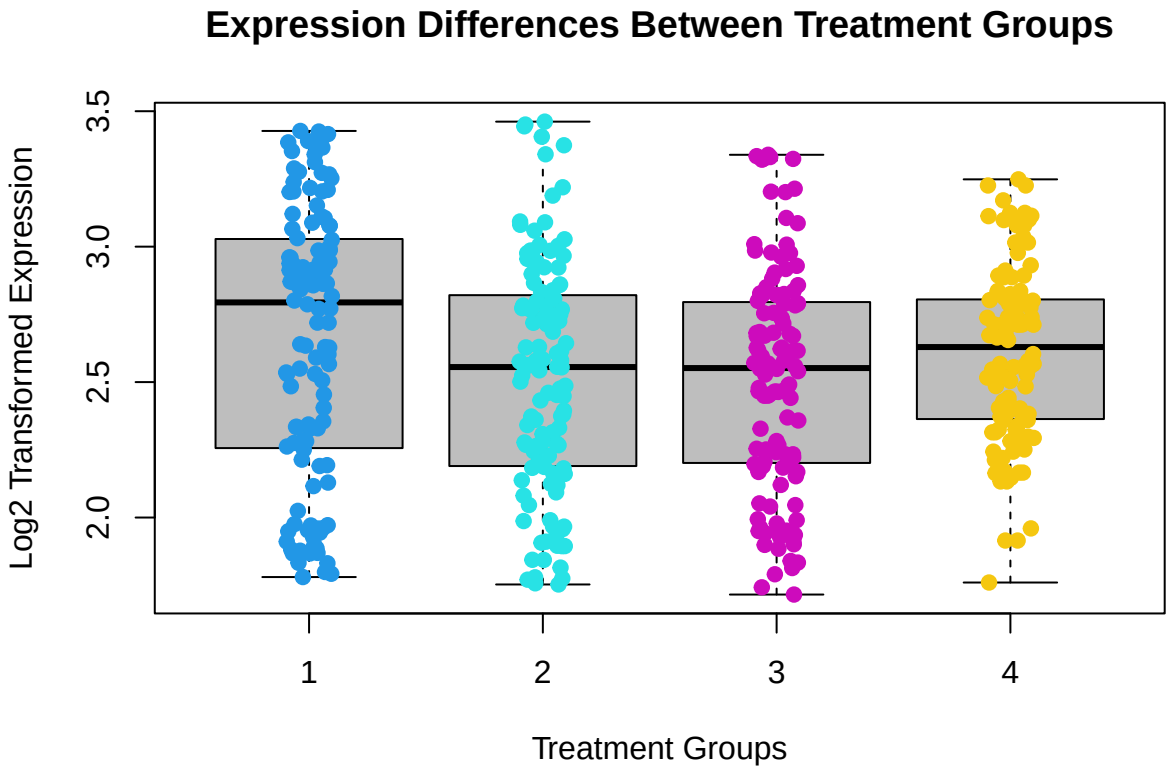
Expression data is grouped by treatment categories. Separate vectors for each group are created, which are then combined into a single data frame.

```
cat("Preparing data for the boxplot...\n")

# Group expression values by treament groups
treatment_group1 <- c(data_gene_list$A, data_gene_list$B, data_gene_list$C)
treatment_group2 <- c(data_gene_list$D, data_gene_list$E, data_gene_list$F)
treatment_group3 <- c(data_gene_list$G, data_gene_list$H, data_gene_list$I)
treatment_group4 <- c(data_gene_list$J, data_gene_list$K, data_gene_list$K)

treatment_counts <- c(
  treatment_group1,
  treatment_group2,
  treatment_group3,
  treatment_group4
)

# Get the length of each treatment
group_sizes <- c(
  length(treatment_group1),
  length(treatment_group2),
  length(treatment_group3),
  length(treatment_group4)
)
```

```r
stopifnot(all(group_sizes > 0))

# Assign group labels for each treatment group, code adapted from assignment instructions
treatment_groups <- c(
  rep("1", group_sizes[1]),
  rep("2", group_sizes[2]),
  rep("3", group_sizes[3]),
  rep("4", group_sizes[4])
)

gene_boxplot_data <- data.frame(
  TreatmentGroup = treatment_groups,
  Expression = treatment_counts
)

cat("Data for boxplot prepared successfully.\n")
```

Expression Differences Between Treatment Groups

The box plot shows the visual differences in gene expression for different treatment groups.



Summary

The heatmaps and the boxplot show patterns in gene expression across treatment groups (1, 2, 3 and 4) and gene types (XA, XB, XC). Treatment Group 1 shows the highest variability in gene expression while Treatment Groups 2 and 3 present more uniform expression levels as can be seen from the boxplot. Treatment group 4 seems to display the lowest gene expression in the heatmaps but its smaller interquartile range in the boxplot suggests more consistent expression levels among most genes.