CHIRAL Bangladesh

**Center for Health Innovation,
Research, Action and Learning**

An Assignment on RNA-Seq Analysis of Peripheral Blood Mononuclear Cells in Multiple Sclerosis Patients and Controls (Data: GSE21942)

Submitted To,

Jubayer Hossain

Founder & Instructor, CHIRAL Bangladesh

Submitted by,

Anisha Tashruba Riya

Research Intern, CHIRAL, Bangladesh

Duration: October 2024 – December 2024

Contents

Introduction

Multiple sclerosis (MS) is a chronic autoimmune disorder that affects the central nervous system, leading to neurological impairment. One of the potential ways to understand the underlying molecular mechanisms of MS is through RNA-Seq analysis, which allows for the identification of gene expression differences in peripheral blood mononuclear cells (PBMCs) from MS patients and healthy controls. In this study, the GEO dataset (GSE21942) was used to perform RNA-Seq analysis, which includes downloading, preprocessing, differential expression analysis, and functional enrichment.

Methods

1. Data Download and Import

The RNA-Seq dataset was first accessed from the Gene Expression Omnibus (GEO) repository using the GEOquery package. The dataset is identified by the GEO accession number **GSE21942**, which contains gene expression profiles from PBMCs of MS patients and healthy controls.

```
gse_data <- getGEO("GSE21942", GSEMatrix = TRUE)
```

The getGEO() function loads the data and stores it as a list. After downloading the dataset, the expression matrix and metadata were extracted from the first element of the list, which corresponds to the expression data.

```
count_data <- exprs(gse_data[[1]])  # Extract expression matrix
metadata <- pData(gse_data[[1]])   # Extract metadata
```

Here, count_data represents the gene expression data, while metadata contains information about the samples, such as their experimental conditions (MS patients or controls).

2. Data Preprocessing

2.1 Data Normalization

RNA-Seq data typically require normalization to adjust for systematic biases introduced by technical factors. Log transformation was applied to the raw count data to stabilize variance across different gene expression levels.

```
normalized_count_data <- log2(count_data + 1)
```

Adding 1 before applying the log transformation avoids issues with zero values in the dataset. This normalization step ensures that the data becomes more comparable across genes and samples, reducing technical variability.

2.2 Filtering Low-Expressed Genes

Genes with insufficient expression with counts < 10 in more than 50% of the samples were filtered out . This helps eliminate genes that are unlikely to provide meaningful biological insights.

```
low_expressed_genes <- normalized_data[rowMeans(normalized_count_data > 10) > 0.5,
```

This step ensures that we retain only the genes with sufficient expression across the majority of the samples, enhancing the power of downstream analyses.

2.3 Mapping Probe IDs to Gene Symbols

The dataset contains probe IDs, which need to be mapped to gene symbols for better interpretability. The **hgu133plus2.db** package was used to map the probe IDs to gene symbols:

```
library(hgu133plus2.db)

gene_symbols <- mapIds(hgu133plus2.db, keys = rownames(count_data), column =
"SYMBOL", keytype = "PROBEID", multiVals = "first")
```

This step facilitates the identification of genes in subsequent analyses. The gene symbols were added as a new column in the dataset:

```
N.count_data <- as.data.frame(normalized_count_data) |>
mutate(GeneSymbol = gene_symbols)
```

2.4 Reshaping Data for Analysis

The next step involved reshaping the data into a long format, which is suitable for further analysis and visualization. The pivot_longer function from the **tidyverse** package was used to reshape the data, creating a new column Expression to store gene expression values:

```
data_long <- N.count_data |>
pivot_longer(cols = -GeneSymbol, names_to = "samples", values_to = "Expression")
```

This transformation allows for easier plotting and further analysis, as each gene expression value is now associated with a corresponding gene symbol and sample.

2.5 Merging Count Data and Metadata

Next, the count data (long format) was merged with the metadata to associate each gene expression value with its corresponding sample's experimental condition:

```
final_data <- data_long |>
left_join(meta.data, by = c('samples' = 'samples'))
```

This merged dataset provides the necessary information for conducting differential expression analysis, where we can investigate how gene expression differs between MS patients and controls.

3. Data Visualization: Before and After Normalization

To evaluate the impact of normalization on the data, the gene expression distributions before and after normalization was visualized using density plots. The following code first transforms the count data into long format for both the pre- and post-normalized data:

```
before_normalized_long <- as.data.frame(count_data) |>
  rownames_to_column("Gene") |>
  pivot_longer(-Gene, names_to = "Sample", values_to = "Expression") |>
  mutate(Status = "Before Normalization")

normalized_count_data <- log2(count_data + 1)

after_normalized_long <- as.data.frame(normalized_count_data) |>
  rownames_to_column("Gene") |>
  pivot_longer(-Gene, names_to = "Sample", values_to = "Expression") |>
  mutate(Status = "After Normalization")
```

Both datasets are then combined into one for visualization:

```
merged_data <- bind_rows(before_normalized_long, after_normalized_long)
```

Finally, density plot was generated to assess the distribution of gene expression values before and after normalization:

```
ggplot(merged_data, aes(x = Expression, y = Status)) +
  geom_density()
```

This plot allows to visually inspect how the data's distribution changes after normalization. Normalization should result in a more consistent expression distribution across samples, which is evident in the density plot.

Results

The dataset was preprocessed, with raw counts transformed using log2 normalization. The resulting dataset contained expression values for various genes, and we successfully mapped probe IDs to gene symbols. The normalization process helped stabilize the variance across genes, and visual inspection via density plots confirmed that the normalization step was effective in reducing skewed distributions.

Discussion

The preprocessing steps undertaken in this study, including normalization, gene filtering, and probe ID mapping, are crucial for preparing the RNA-Seq data for subsequent analysis. The successful transformation of the dataset into a long format and the merging of count data with metadata set the stage for differential expression analysis, which will allow us to identify genes that are differentially expressed between MS patients and healthy controls.

In terms of data quality, the visualization of gene expression distributions before and after normalization indicated that the normalization process had a positive impact on the data, ensuring that biases were minimized and that the gene expression values were more comparable across samples.

Conclusion

This RNA-Seq analysis workflow, from data downloading to preprocessing, laid a strong foundation for further downstream analyses, including differential expression analysis and functional enrichment. By addressing normalization and gene symbol mapping, we ensured that the data was ready for subsequent analyses that could uncover potential biomarkers for multiple sclerosis. Future steps include performing differential gene expression analysis to identify significant genes and conducting pathway enrichment analyses to explore the biological significance of these genes in the context of MS.