Level 1: Module 4 – Differential Gene Expression (DGE) Analysis

Table of Contents

Before starting with the analysis:
Please refer to previous modules for more information.

- Identify outliers (Module 3)

PACKAGES

Install and load the following packages; and explore their documentation for more information on their functions and use cases

- use `BiocManager::install("package_name")`
  - hgu133plus2.db
  - limma
  - EnhancedVolcano
- use `install.packages("package_name")`
  - ggplot2
  - pheatmap

OUTLIERS

Remove samples identified as outliers from both data and metadata.

ANNOTATION

The row names of our data are called probe IDs. They identify specific probes that are used in microarray sequencing. A probe is a DNA sequence and each probe ID corresponds to a specific gene, transcription factor, or other known sequence. Because we need to know these sequences and identify which ones to use in order to conduct microarray sequencing, we need *a priori* knowledge or knowledge "from before."

When analyzing data, we need to know which probe IDs correspond to which known sequences. Because there are many different databases with different probe IDs corresponding to different sequences, this requires an extra step: annotation. In this step we are annotating our probe IDs with gene names (or another identifier) using the appropriate database. In our case, that database is hgu133plus.db. This information can be found on a data set's GEO page under the platform description.

Select the gene symbols

Function: `select()` from AnnotationDbi package (installed with hgu133plus2.db)
Input:
Database (hgu133plus2.db)
Vector of probe IDs
Vector of columns to return (*hint*: we want gene SYMBOLs)

Remove duplicate probe IDs, NA values, and duplicate symbols

Some probes map to the same gene symbol and some gene symbols map to the same probe ID. We also want to remove any rows where the probe ID was not mapped to a symbol (ie. NA values) since we can't accurately analyze them. There are many ways to remove these extraneous rows:

- Keep the first occurrence
- Keep the last occurrence
- Keep the "unique" occurrence

Reason about and pick a rationale for choosing which probe ID/ symbol to use a representative. Here are some functions which may be useful:

- `duplicated()`
- `collapseRows()` from the WCGNA package
- `!` indicates not in most programming languages (ex. `!a` means "not a")

Tip: Remove duplicate probe IDs first. Then remove NA values. If you remove duplicate symbols first, you could lose valuable information.

GENE FILTERING

There are many genes that had a 1:1 mapping with a probe ID. In order to focus on the genes that will provide good quality results, we are going to filter out genes below the 2nd percentile (.02) of the expression distribution of the data set. We can do this by using the `quantile()` function.

Tip: The input of quantile is a numeric vector. You will want to calculate the mean of each row in your matrix. You can do this using the `rowMeans()` function.

LIMMA

Deliverable 1: Write out the differential expression results to a file sorted by adjusted p-value.

Deliverable 2: Report the number of significant genes with an adjusted p-value < 0.05. This will be useful for quickly determining correctness and identifying possible errors.

Limma is an R/Bioconductor package that provides an integrated solution for analyzing data from gene expression experiments. Read the documentation of limma and explore arguments and functions that you would like to implement.

Build a model preserving features of interest - this is the same model from batch correction

Function: `model.matrix()`
Input:
Factor of Feature of Interest (whether the sample is normal or cancer)
- this should be a column in your metadata

Data Frame (metadata)

Limma analysis

Function: `lmFit()`
Input:
Data Frame (Filtered data)
Model from `model.matrix()`

Function: `eBayes()`
Input:
Linear model

Tip: Pay attention to what a 1 represents in your model matrix. If a 1 represents normal tissue, your results will be flipped from someone who has 1 representing cancer tissue. The standard analysis uses 0 for normal tissue (and 1 for diseased or cancer tissue).

Generate a topTable of your DEG

Function: `topTable()`

Input:

Limma model (output of `eBayes()`)

DATA VISUALIZATIONS

Volcano Plot

Deliverable 3: Volcano plot of DEG.

Volcano plots are a visualization used for differential gene expression analysis. The x-axis represents log-fold change and the y-axis represents negative p-value or negative log-p-value. Log-fold change value refers to how exponentially different a gene's average expression value is compared to a baseline or normal expression. Therefore, the greater the absolute value of log-fold change the more differentially expressed a gene. The p-value indicates how statistically significant the log-fold change is based on Bayesian statistics (`eBayes` function). A lower p-value, or higher negative p-value, indicates greater significance. Therefore, points that are in the top right or top left of your volcano plot are the most differentially expressed genes with the greatest significance. Additional resources for understanding volcano plots will be linked on the STEM-Away Level 1: Module 4 post.

You can plot a volcano plot using the `EnhancedVolcano()` function from the package of the same name or using the `volcanoplot()` function from the limma package.

Desired Output: Your plot should look similar to the figure below.

DEG Heatmap

Deliverable 4: Heatmap of top 50 DEG.

Heatmaps use color gradients to represent the intensity of a certain metric. For this heatmap, we'll be using differential expression values to compare the genes based on their samples' expression. **You should use the pheatmap package and function to create this plot.**

The heatmaps made with the pheatmap also annotate the plot based on hierarchical clustering. Hierarchical clustering is an unsupervised learning technique that tries to cluster similar things together with no prior knowledge using dis/similarity metrics. The branching structure can be interpreted much like a family tree. Samples connected by "lower" branches have a "closer relationship" meaning they are more similar than those with further branches. If you want to know more about this clustering method, I recommend you watch some of StatQuest's videos on the topic. You can find them on YouTube. Additional resources will be linked on the STEM-Away Level 1: Module 3 post.

Calculate top 50 DEG

Function: `topTable()`

Arguments:

```
number=50
```

Plot heatmap

Function: `pheatmap()`

Input: Data Frame (`filt[rownames(filt) %in% rownames(top50)),]`)

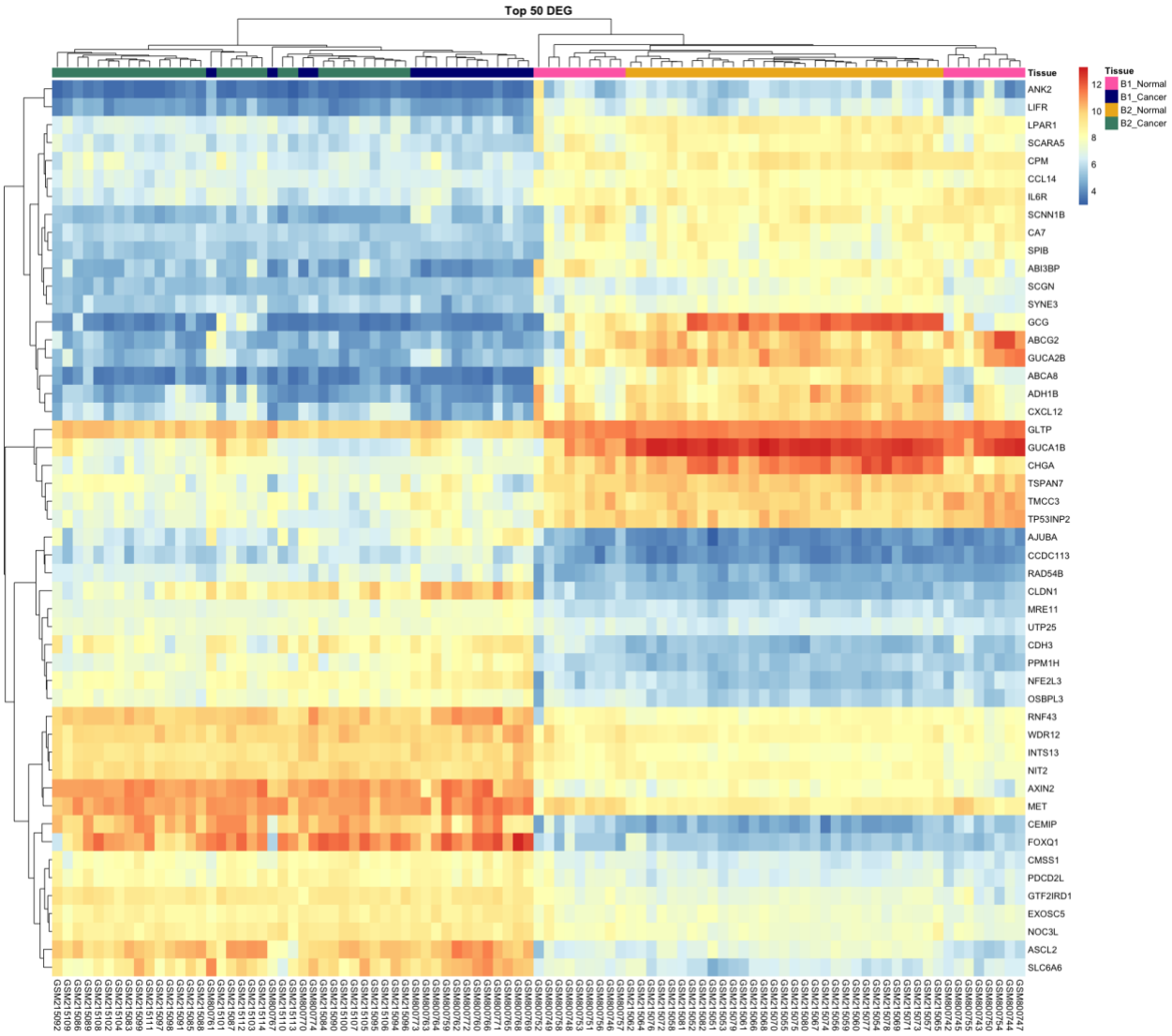
Additional Arguments:

`annotation_col=[T]`	annotate columns
`cluster_rows=T`	cluster rows

`[T]` should be the name of a data frame representation of your 4 groups with sample names as the row names. A portion of an example is shown below:

	TISSUE
GSM215051	B1_Normal
GSM215052	B1_Normal

Desired Output: Your plot should look similar to the figure below. **You should have the different colors representing 4 groups: Batch 1 Cancer, Batch 1 Normal, Batch 2 Cancer, Batch 2 Normal.** To do this, I added a column to my metadata. Colors should represent 2 groups if using 1 data set: Normal, Cancer.



All Deliverables:

1. **Write out the differential expression results to a file sorted by adjusted p-value.**
2. **Report the number of significant genes with an adjusted p-value < 0.05. This will be useful for quickly determining correctness and identifying possible errors.**
3. **Volcano plot of DEG.**
4. **Heatmap of top 50 DEG.**