Gene Expression Analysis with R

Amina Aboulhana

STT-303-001

Dr. Kai Kang

May 8, 2025

Introduction

For this final project, I chose to analyze RNA-seq data from the GTEx (Genotype-Tissue Expression) project, a publicly available database. This dataset includes gene expression levels across a wide range of human tissues and provides rich metadata, including age, sex, and tissue type. My goal for this project was to use statistical and machine learning methods to explore how gene expression varies across biological conditions and whether it could be used to predict factors such as age or tissue type.

While I initially designed a comprehensive R-based project to perform PCA, clustering, regression, and classification, a range of technical issues and data complications interfered with its execution. This reflection will outline the approaches I took, the results I was able to produce, and the key issues that ultimately prevented the project from being fully realized as planned.

Methods

Gathering Data and Variables

To start, I downloaded the GTEx data and converted the file into a compatible .zip file. After introducing this data into R, I focused on a subset of 10,000 genes in order to reduce memory usage and make the analysis more feasible for my resources. The gene expression data came in TPM (Transcripts Per Million), and the metadata included variables such as tissue type, sex, and age. I used a custom function to read in the first 10,000 lines of the gene expression file and then combined it with metadata using “SAMPID”.

The data cleaning process involved several steps. First, I converted the gene expression matrix from character to numeric. I then filtered out genes with low variance using ‘rowMedians’ to remove genes that did not show meaningful statistical significance across samples. Following that, I applied a log transformation (‘log1p’) to normalize the data. Once

cleaned and transformed, the expression matrix was merged with the metadata, and I set up the rest of my code that would conduct the exploratory analysis: PCA and clustering.

The next step was attempting to reduce the dimensionality of the data using PCA ('prcomp') and visualize the results using the 'factoextra' package. My goal by doing this was to see whether samples clustered naturally by tissue or sex. For supervised learning, I prepared to run random forest regression models using the 'caret' package to predict age. I also set up code for heatmap generation using 'heatmaply', intending to explore co-expressed gene clusters.

Results

Some aspects of the code worked as expected. I successfully loaded and merged the expression data with the metadata, applied transformations, and implemented conditions in case PCA or heatmaps could not run. I also validated that filtering and transformation steps preserved numeric integrity and produced a smaller, cleaner dataset.

However, several issues affected the results. The PCA step failed to generate a plot because too few genes passed the variance threshold after filtering. The fallback message I coded was triggered, indicating that there were not enough variable genes to display the PCA plot. In an attempt to correct this, I increased the number of genes to see if it would produce enough variables. However, this is when I ran into a vector memory issue; there was not enough space to process additional genes.

One of the most significant problems I encountered was the issue of memory overload. Despite limiting the dataset to 10,000 genes, I quickly realized that many functions in R, especially those related to matrix transformations and plotting, were not optimized for handling such high-dimensional data on a personal laptop. When I attempted to run PCA or generate heatmaps, the session would crash or return errors due to missing or non-numeric values that

weren't obvious at first. Even after transforming the data and applying filters, some columns contained hidden formatting issues that disrupted modeling functions like 'prcomp()' and 'randomForest'.

To troubleshoot these issues, I worked with a professional who has experience in programming and software development. Together, we went through the code step by step to identify where it was breaking down. After debugging multiple sections, we concluded that the main limitation was the lack of available memory and computational space. Even after subsetting the data and applying filters, the system could not retain enough valid and statistically significant variables for key functions, like PCA and heatmap visualizations, to run properly. This collaboration helped me realize that the structure of the dataset simply exceeded what my local environment could support, and that resolving these issues would likely require either cloud computing resources or a more efficient data handling strategy.

Conclusion

Although this project was not executed as planned, and did not produce polished visualizations or predictive models. It offered a valuable and realistic experience in working with high-dimensional biological data. Through this project, I quickly learned that gene expression analysis is not only data-heavy but also demands a computational environment to handle this kind of analysis. Working within those constraints would require careful planning and continuous adjustment to fit the scope of the data.

If I were to repeat this analysis, I would narrow the scope by focusing on one specific tissue type or a smaller subset of genes, which would allow for more efficient processing and hopefully produce clearer outputs. I would also try to incorporate tools such as 'DelayedArray' and consider using stronger computing environments to avoid space and memory limitations,

like the ones I encountered during this project. Besides technical improvements, this project highlighted the importance of early validation, code structure, and detailed troubleshooting. While the final output did not meet my original goal or follow my detailed outline and expectations, the process itself allowed me to strengthen my programming skills by understanding the complexity of advanced datasets. This project allowed me to face technical issues and forced me to find solutions and appreciate the role of adaptability when facing setbacks in data science.