Identifying Genetic Biomarkers to Differentiate Subtypes of LIHC

Group 17

December 9th, 2024

Jung Kyu Justin Yoon	*61298212*
Zeena Jarallah	*37538964*
Muhammad Al Muhtadin	*15018815*

BMEG 310: Introduction to Bioinformatics
University of British Columbia

Abstract

In this project we utilized gene expression data across 352 patients to identify two subtypes (Subtype A and Subtype B) of Liver Hepatocellular Carcinoma (LIHC), a type of primary liver cancer. Survival analysis revealed significant differences in survival probabilities, with Subtype B having lower survival rates. Mutation analysis identified subtype-specific genetic mutations. While differential expression analysis further identified 18 genes as potential biomarkers for subtype differentiation.

Introduction

Liver Hepatocellular Carcinoma (LIHC), accounts for over 80% of primary liver cancer cases and is the fourth leading cause of cancer-related deaths around the world [1], [2]. Despite advancements in cancer diagnostics and therapeutics, LIHC is often diagnosed at advanced stages, resulting in a low five-year survival rate for LIHC patients [3]. Thus, there is a critical need to identify robust diagnostic biomarkers for early diagnosis of LIHC, and to establish personalized therapeutic strategies.

Recent studies have explored differentially expressed genes (DEGs) to enhance early diagnosis and treatment precision. For example, prior research has identified DEGs associated with pathways for cell cycle regulation, glycan degradation, and immune signalling, that contribute to LIHC progression [4]. These studies have led to the classification of molecular subtypes of LIHC, finding potential targets for individualized therapies. However, most existing studies have focused on identifying DEGs to distinguish biomarkers for general LIHC diagnoses, with less attention given to biomarkers that distinguish subtypes within LIHC. Understanding variations within cancer groups is crucial for developing more nuanced diagnostic, prognostic, and treatment approaches.

In this project, we set out to bridge this gap by identifying genetic biomarkers that differentiate subtypes of LIHC. Through our gene expression analysis, we identified two distinct LIHC subtypes, which we called 'Subtype A' and 'Subtype B'. We also identified the top 18 DEGs that can be used to distinguish these subtypes. Our findings confirm the heterogeneity of LIHC and offer novel insights into subtype-specific gene expression patterns. These results hold potential for advancing diagnosis and personalized treatment for LIHC.

Methods

We began by importing three datasets from the TGCA website: clinical data, which contained patient information, survival statistics, and tumor characteristics; mutational data, which contained genomic differences, chromosome information, and gene/allele information, and RNA-seq expression data containing gene expression profiles. Initially, patient IDs in all datasets (clinical, mutation, and RNA-seq) were standardized to a common format, allowing identification of patients present in all three datasets. The clinical and mutation data were then filtered to contain only rows corresponding to these common patients, while the RNA-seq data was reduced to include only columns (patients) matching this intersection. For the mutation dataset, filtering removed low-impact mutations, intron variants, and synonymous mutations, as these changes are unlikely to have significant biological impact. This reduced noise and narrowed the focus to functional mutations. In the RNA-seq dataset, irrelevant columns were filtered out, while gene names were kept.

To analyze and better understand the clinical data after filtering, bar graphs were made to show age distribution, age vs overall survival, age vs progression free survival, age vs tumor stage, and age vs disease specific survival. Survival analysis was done using Kaplan-Meier Analysis where Kaplan-Meier graphs were made of survival probably vs Gender, survival probably vs Weight, survival probably vs Tumor Stage, survival probably vs Race, and survival probably vs Age-group; where its significance was determined by its p-value.

To create clusters using Expression data, the expression data was normalized using counts per million (CPM) to account for differences in sequencing depth, followed by log transformation to stabilize variance. This prepared the data for identifying the most variably expressed genes. From these, we selected the top 50 genes with the highest variance for visualization and clustering. Visualization was done using a heatmap using the euclidean method.

Clustering was performed using the euclidean method to group patients based on gene expression patterns. A distance matrix was created and Ward's method was applied to construct clusters, which were visualized through dendrograms and principal component analysis (PCA). The dendrogram was cut into two clusters to create two patient groups. Using the list of patients in each cluster, further analysis was done on the mutations, patient, and expression datasets.

The mutation dataset was filtered to create separate mutation data subsets for each cluster, isolating the genomic changes associated with each group. For Cluster 1, the top 20 most frequently mutated genes were sorted by frequency and visualized using a bar chart to highlight key mutations. The same process was repeated for Cluster 2 to allow for comparison. To account for the differences in the size of the clusters, mutation frequencies were normalized by dividing the raw frequency by the number of patients in each cluster. The normalized frequencies for the top 15 genes from each cluster were then combined into a single dataset. A bar chart was created to visualize the normalized mutation frequencies of these genes across clusters. The analysis was repeated to examine mutation chromosomes. Mutation frequencies were calculated and ranked by chromosome for each cluster. Bar charts were created to showcase the top 20 most frequently mutated chromosomes in each cluster, showing any chromosomal patterns unique to a specific group. Similar to the gene analysis, these frequencies were normalized by cluster size, and a combined visualization was generated to compare chromosome-level mutation patterns across clusters. The analysis was repeated once more for the consequences of mutations. The frequency of each mutation consequence was calculated and ranked for each cluster and bar charts were made to visualize the top 20 most common consequences. Normalized frequencies were again compared across clusters.

Using the patient dataset, Kaplan-Meier survival curves were generated to compare overall survival and progression-free survival probabilities, stratified by clusters. Kaplan-Meier Analysis was done to compare clusters by gender vs survival probability, tumor stage vs survival probability, race vs survival probability, age vs survival probability, sex vs survival probability, weight group vs survival probability, and genetic ancestry vs survival probability to determine if there were any differences between the two clusters that are identifiable using the Kaplan-Meier Analysis; where its significance was determined by its p-value.

Using the expression data, differential expression analysis was done using DESeq2 to identify genes with significant expression changes between clusters. This involved creating a DESeq dataset with the normalized RNA-seq data and running the DESeq pipeline. The results were summarized in a volcano plot, highlighting upregulated and downregulated genes based on log fold changes and the p-values. Next, cluster-specific DEGs were identified and visualized. All significant genes were isolated. Normalized expression counts for the cluster-specific genes were found and put in an MA plot to compare the two clusters.

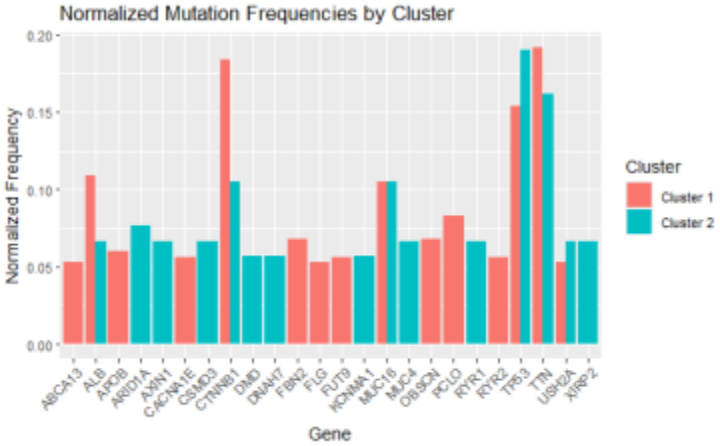
The top 10 DEGs per cluster were identified, and normalized counts were obtained for those genes. The cluster information was added and the average expression per gene per cluster was calculated to retrieve only the top expressed genes by cluster. Boxplots were generated, comparing the distribution of expression levels of these genes in each cluster, with each DEG displayed in a separate panel.

Finally, pathway analysis was conducted to relate differential gene expression to biological pathways. Using the GAGE and Pathview packages, metabolic pathways that differentiate the clusters were identified.

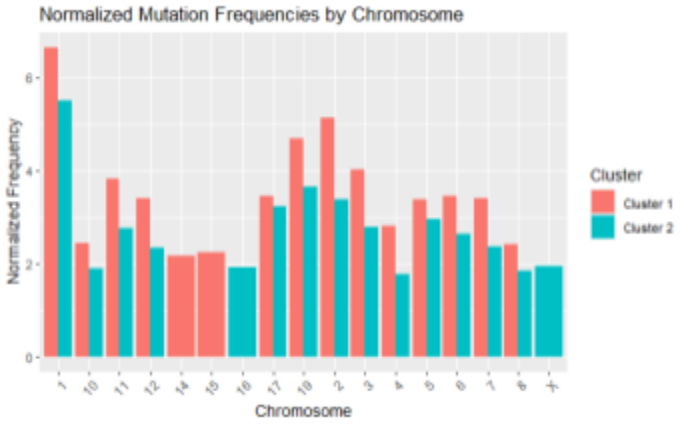
Results

a) Gene Mutations Frequency Profile Comparisons:

Analysis of mutation data showed that the top three most frequently mutated genes were the same for both Subtypes, shown below; Subtype A (Cluster 1) and Subtype B (Cluster 2). Those genes were *TTN*, *CTNNB1*, and *TP53*. A comparison of the top 20 mutated genes between the two clusters, visualized in the figure on the right, shows that *CTNNB1* is far more frequently mutated in Subtype A than in Subtype B. *ALB* is also more frequently mutated in Subtype A than B. While the genes *ABCA13*, *APOB*, *CACNA1E*, *FBN2*, *FLG*, *FUT9*,



OBSCN, *PCLO*, *RYR2*, and *ARID1A*, are the most frequently mutated for Subtype A patients, they aren't as frequently mutated in Subtype B. Conversely, the genes *AXIN1*, *CSMD3*, *DMD*, *DNAH7*, *KCNMA1*, *MUC4*, *RYR1*, and *XIRP2* are the most frequently mutated for Subtype B patients, and are not as frequently mutated in Subtype A. *Supplemental figures can be found in supplemental appendix figures 14-16.*



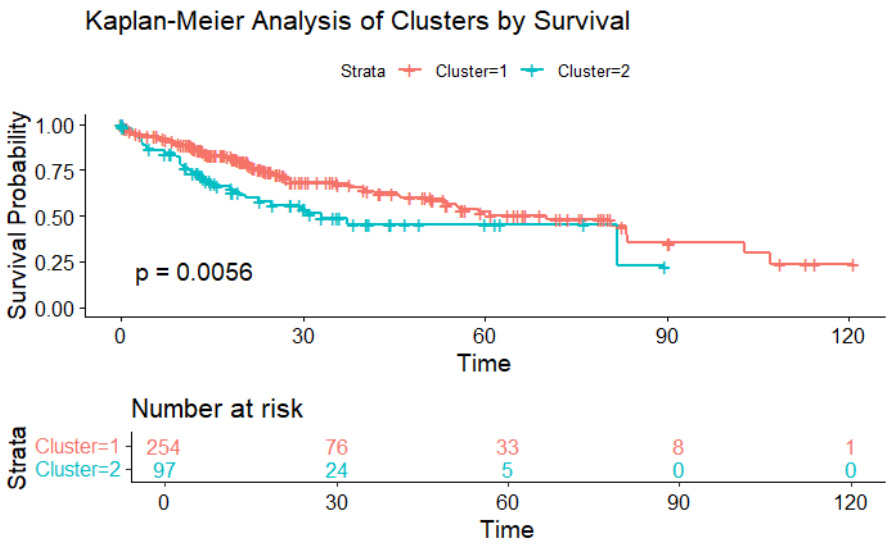
b) Chromosomal mutations Profile Comparisons: A comparison of the normalized top 20 most frequently mutated chromosomes between the two subtypes was conducted, as shown in the figure on the left. The chromosomes 14 and 15 were frequently mutated in Subtype A, but not in Subtype B. The chromosomes 16 and X were frequently mutated in Subtype B, but not in Subtype A. There was no major difference between the subtypes in mutational frequencies across the other chromosomes. *Supplemental figures can be found in supplemental appendix figures 17-19.*

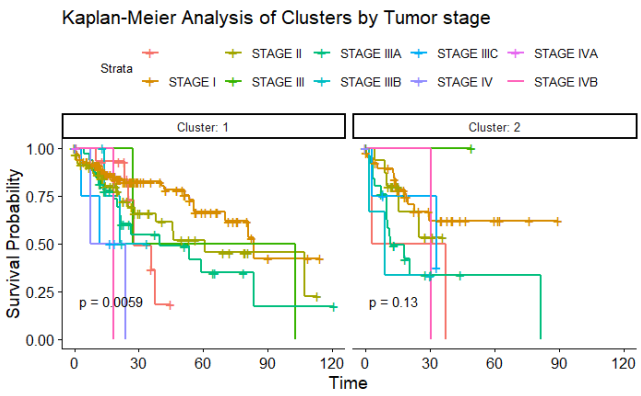
c) Mutation consequences Profile Comparisons:

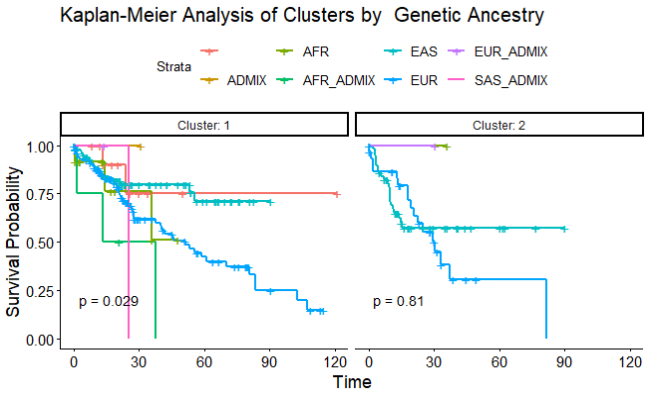
A comparison of the top 20 most frequent consequences of mutations between the two subtypes was conducted and is visualized in the figure on the right. Results showed that the frequency of Downstream Gene Variants displayed in Subtype A, is not shared with Subtype B. Similarly, the frequency of Upstream Gene Variants displayed in Subtype B, is not shared with Subtype A. The most frequent consequential variants for both subtypes were Missense Variants. *Supplemental figures can be found in supplemental appendix figures 20-22.*

d) Survival Analysis

Survival analysis revealed a significant difference in survival probabilities between the two subtypes ($p = 0.0056$) where Subtype B showed a lower survival probability across time compared to Subtype A, shown below. Within Subtype A, tumor stage was found to significantly impact survival ($p = 0.0059$), with advanced tumor stages (IVB and IVA) associated with worse outcomes. In contrast, tumor stage did not significantly affect survival in Subtype B ($p = 0.13$). Additionally, genetic ancestry was significantly associated with survival in Subtype A ($p = 0.029$), with patients of AFR_ADMIX ancestry demonstrating worse outcomes. However, no significant differences in survival based on genetic ancestry were observed in Subtype B ($p = 0.81$).





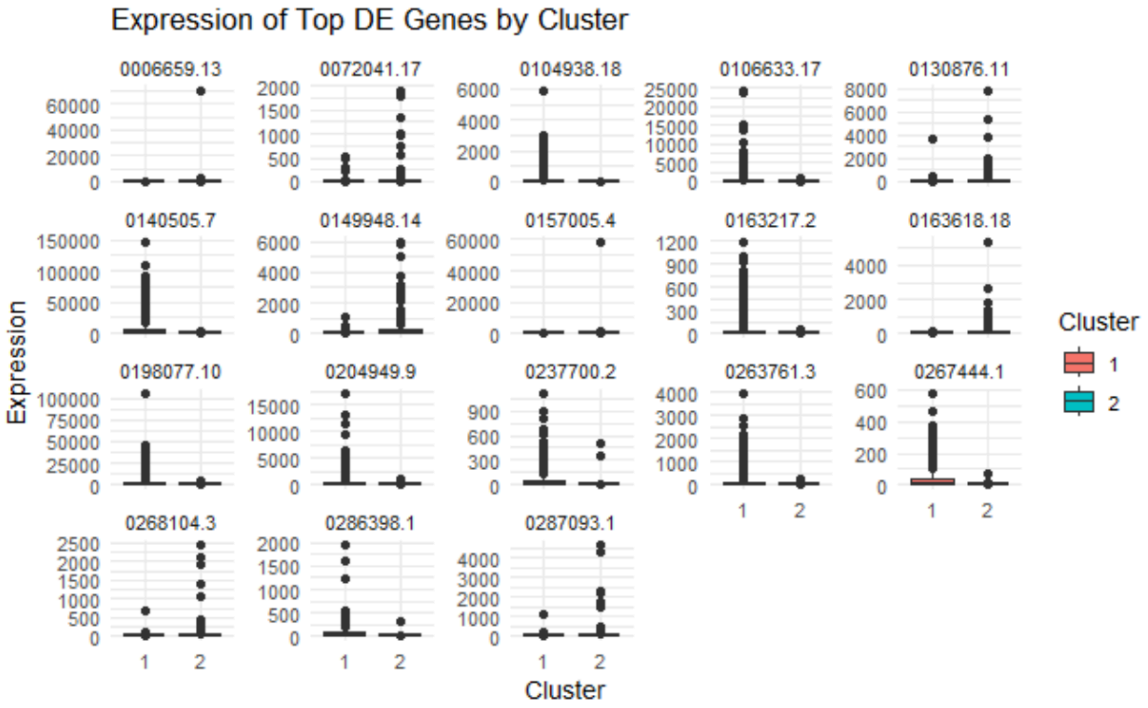


Supplemental figures can be found in supplemental appendix figures 23-30.

e) Differential Expression Analysis

Differential Expression Analysis revealed 18 DEGs that were differentially expressed between the two Subtypes, as showcased in the figure below. Results showed that the genes GCK (ENSG00000106633.17), SMUG1P1 (ENSG00000267444.1), CLEC4M (ENSG00000104938.18), CYP1A2 (ENSG00000140505.7), PRAMEF33 (ENSG00000237700.2), FAM83A-AS1 (ENSG00000204949.9), GDF2 (ENSG00000263761.3), CYP2A7 (ENSG00000198077.10), BMP10 (ENSG00000163217.2), and a novel transcript found on gene GRCh37 (ENSG00000286398.1), all displayed high expression levels in Subtype A, but very low expression levels in Subtype B. Conversely, the genes LGALS14 (ENSG00000006659.13), HMGA2 (ENSG00000149948.14), SLC6A15 (ENSG00000072041.17), SLC6A14 (ENSG00000268104.3), SST (ENSG00000157005.4), CADPS (ENSG00000163618.18), SLC7A10 (ENSG00000130876.11), and a novel gene found on gene GRCh38 (ENSG00000287093.1), all displayed high expression levels in Subtype B, but very low expression levels in Subtype A. These results are summarized in Table 1 in the Appendix.

Expression of Top DE Genes by Cluster
(Note, the Ensemble Gene ID is displayed at the top of each box plot, with the repeating first part "ENSG0000" redacted for figure clarity).



Supplemental figures can be found in supplemental appendix figures 31-33

f) Pathway Analysis

From the pathway analysis, we found that the top 5 upregulated and downregulated pathways were "hsa00232", "hsa00983", "hsa00230", "hsa04514", and "hsa04010". Pathway analysis showed identical results for both clusters, with the same pathway for the biological pathways. This indicates that, despite differences in gene-level and mutational profiles, the functional impact at the pathway level does not affect between Cluster 1 and Cluster 2. The comparison of the top 5 pathways for each cluster can be found in table 2 in the Appendix and individual pathways can be found in supplemental Appendix figure 34-48.

Discussion

In this project, we uncovered heterogeneity in Liver Hepatocellular Carcinoma (LIHC). Using gene expression data, we identified two distinct subtypes of LIHC (Subtype A and Subtype B), and characterized their molecular and clinical profiles based on patient data, mutation data, and gene expression data for a common group of 352 patients.

Survival analysis showed a significant difference in survival probabilities that distinguishes the two subtypes, as Subtype B appeared to showcase significantly lower survival probabilities than Subtype A. This distinction means that correctly diagnosing subtypes can be crucial for developing distinct treatments for patients with each subtype, especially for those with Subtype B, to increase the survival probabilities for patients of all subtypes.

Mutation analysis showed that the genes *ABCA13*, *APOB*, *CACNA1E*, *FBN2*, *FLG*, *FUT9*, *OBSCN*, *PCLO*, *RYR2*, and *ARID1A*, were significantly more often mutated for patients with Subtype A. While the genes *AXIN1*, *CSMD3*, *DMD*, *DNAH7*, *KCNMA1*, *MUC4*, *RYR1*, and *XIRP2* were significantly more often mutated in patients with Subtype B.

Differential Expression Analysis identified 18 DEGs that can be used to diagnose patients with one of the two subtypes. The genes GCK, SMUG1P1, CLEC4M, CYP1A2, PRAMEF33, FAM83A-AS1, GDF2, CYP2A7, BMP10, and a novel transcript found on gene GRCh37, are found to have high expression levels in Subtype A, but very low expression levels in Subtype B. While the genes LGALS14, HMGA2, SLC6A15, SLC6A14, SST, CADPS, SLC7A10, and a novel gene found on gene GRCh38, are found to have high expression levels in Subtype B, but very low levels in Subtype A.

While the results of this project successfully identified genetic biomarkers that differentiate subtypes of LIHC, they are still limited by two main aspects. The first limitation is the use of unsupervised machine learning to create the clusters. Due to the lack of objective evaluation metrics, difficulty in filtering out noise, and the lack of an objectively correct number of clusters, there may be inaccuracies in the cluster groups we chose to work with. The second limitation is the lack of data for a non-cancerous control group. Thus, our results were never compared to a dataset outside of LIHC patients, which would be helpful in placing our findings within a broader context of genetic data outside of LIHC. In the future, the results of this project should be verified using a larger dataset, and the data should be studied with relation to a control group.

References

[1] X. Ouyang, Q. Fan, G. Ling, Y. Shi, and F. Hu, "Identification of Diagnostic Biomarkers and Subtypes of Liver Hepatocellular Carcinoma by Multi-Omics Data Analysis," Genes, vol. 11, no. 9, p. 1051, Sep. 2020, doi: https://doi.org/10.3390/genes11091051.

[2] R. Sun, Y. Gao, and F. Shen, "Identification of subtypes of hepatocellular carcinoma and screening of prognostic molecular diagnostic markers based on cell adhesion molecule related genes," Frontiers in Genetics, vol. 13, no. 13:1042540, Nov. 2022, doi: https://doi.org/10.3389/fgene.2022.1042540.

[3] S. Gao, J. Gang, M. Yu, G. Xin, and H. Tan, "Computational analysis for identification of early diagnostic biomarkers and prognostic biomarkers of liver cancer based on GEO and TCGA databases and studies on pathways and biological functions affecting the survival time of liver cancer," BMC Cancer, vol. 21, no. 1, Jul. 2021, doi: https://doi.org/10.1186/s12885-021-08520-1.

[4] R. Agarwal, J. Narayan, A. Bhattacharyya, M. Saraswat, and A. K. Tomar, "Gene expression profiling, pathway analysis and subtype classification reveal molecular heterogeneity in hepatocellular carcinoma and suggest subtype specific therapeutic targets," Cancer Genetics, vol. 216–217, no. 216-217:3751, pp. 37–51, Oct. 2017, doi: https://doi.org/10.1016/j.cancergen.2017.06.002.

Contributions

For the coding part of the assignment, Justin created the Git Repository, documentation files, meeting minutes, and coordinated group meetings. Justin imported all libraries and datasets needed, then Zeena filtered the datasets. All group members contributed in exploring the data before a specific direction was set. Justin then did the Clinical data analysis on the filtered dataset to get bar graphs to compare age, survival, PFS status, and tumor stage, and did a survival analysis to get kaplan-meier graphs for gender, weight, tumor stage, race, and age group. Zeena then prepared the expression data analysis and performed hierarchical clustering to create the clusters used for further analysis. Using those clusters, Justin did more survival analysis on the clinical data to compare patients in the clusters to get Kaplan-Meier analysis of clusters to compare survival, gender, tumor stage, race, age, sex, weight group, genetic and ancestry ancestry. Zeena used the mutation data to compare differences in frequency of mutations on genes and chromosomes, as well as the consequences of mutations, between the clusters. Zeena then performed a DESeq2 Analysis on the expression data, and created the volcano plot. She then found the top differentially expressed genes between the clusters. Lastly, Justin did the pathway analysis to draw plots for the top 5 pathways for each cluster.

For the report part of the assignment, all group members contributed to literature review. Zeena wrote the abstract and introduction. Justin wrote the methods and results. Muhammad helped in writing results and introduction. Finally, Zeena wrote the discussion. Justin and Zeena both worked on reviewing, formatting, and fine-tuning the report.

For the presentation, Muhammad did the introduction and its corresponding slides, Justin did the methods and its corresponding slides, and Zeena did results and discussion and its corresponding slides.

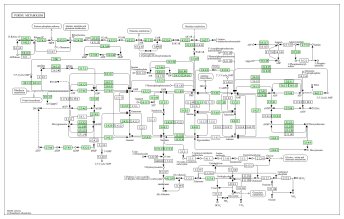
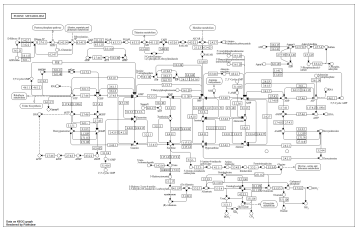
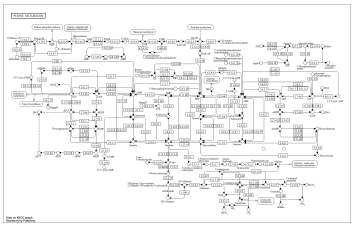
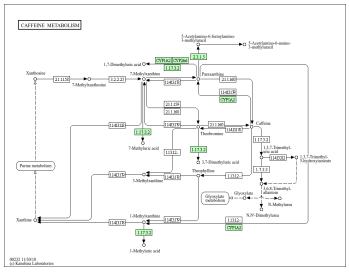
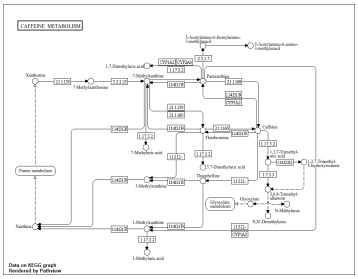
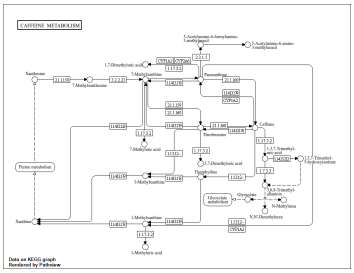
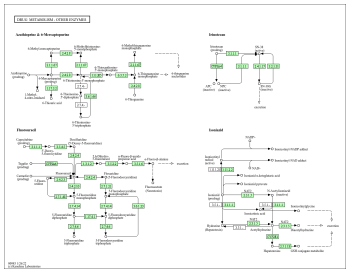
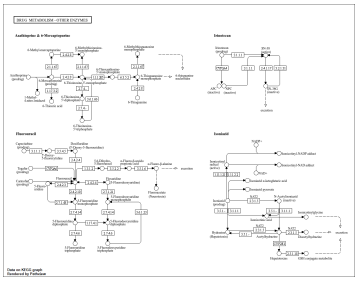
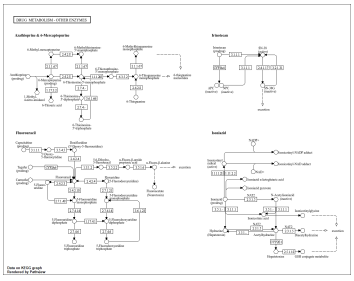
Appendix

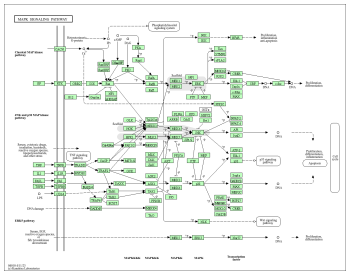
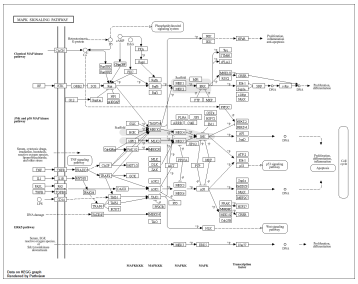
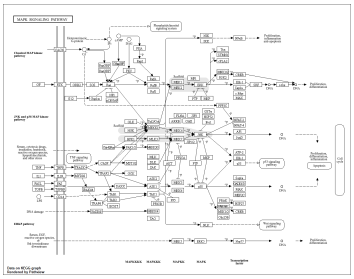
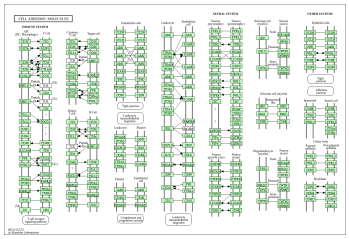
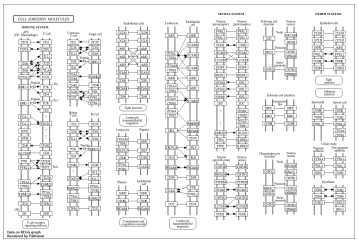
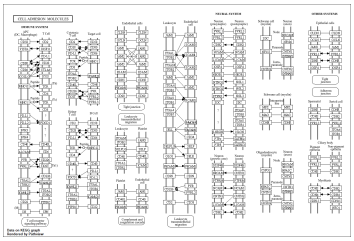
***Table 1**: Summary of Differential Expression Results Table*

Common Gene Name	Symbol	Ensembl Gene ID	Expression in Subtype A	Expression in Subtype B
Glucokinase	GCK	ENSG00000106633.17	High Expression	Low Expression
single-strand-selective monofunctional uracil-DNA glycosylase 1 pseudogene 1	SMUG1P1	ENSG00000267444.1	High Expression	Low Expression
novel transcript	found in GRCh37	ENSG00000286398.1	High Expression	Low Expression
C-type lectin domain family 4 member M	CLEC4M	ENSG00000104938.18	High Expression	Low Expression
cytochrome P450 family 1 subfamily A member 2	CYP1A2	ENSG00000140505.7	High Expression	Low Expression
PRAME family member 33	PRAMEF33	ENSG00000237700.2	HIgh Expression	Moderate to Low Expression
FAM83A antisense RNA 1	FAM83A-AS1	ENSG00000204949.9	High Expression	Low Expression
growth differentiation factor 2	GDF2	ENSG00000263761.3	High Expression	Low Expression
cytochrome P450 family 2 subfamily A member 7	CYP2A7	ENSG00000198077.10	High Expression	Low Expression
bone morphogenetic protein 10	BMP10	ENSG00000163217.2	High Expression	Low Expression
galectin 14	LGALS14	ENSG00000006659.13	Low Expression	One Occurance of High Expression (mostly, low expression)
novel gene	found in GRCh38	ENSG00000287093.1	Low Expression	High Expression

high mobility group AT-hook 2	HMGA2	ENSG00000149948.14	Low Expression	High Expression
solute carrier family 6 member 15	SLC6A15	ENSG00000072041.17	Low Expression	High Expression
solute carrier family 6 member 14	SLC6A14	ENSG00000268104.3	Low Expression	High Expression
somatostatin	SST	ENSG00000157005.4	Low Expression	One Occurance of High Expression (mostly, low expression)
calcium dependent secretion activator	CADPS	ENSG00000163618.18	Low Expression	High Expression
solute carrier family 7 member 10	SLC7A10	ENSG00000130876.11	Low Expression	High Expression

***Table 2**: comparison of the top 5 pathways for Cluster 1 and 2.*

Top 5 pathways	Pathview	Cluster 1	Cluster 2
HSA00230			
HSA00232			
HSA00983			

HSA04010			
HSA04514			

Individual pathways are located in the supplementary appendix figures 34-48

Supplementary Appendix

***Figure 1**: Age distribution of clinical dataset bar graph*

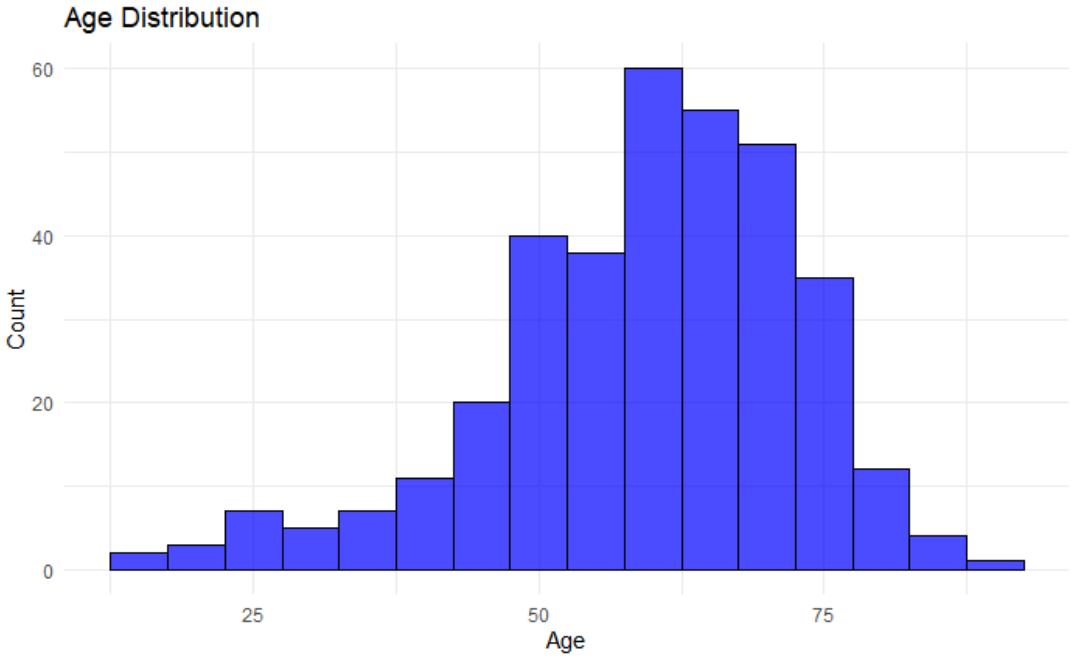


***Figure 2**: Age Distribution by Overall Survival Status bar Graph*

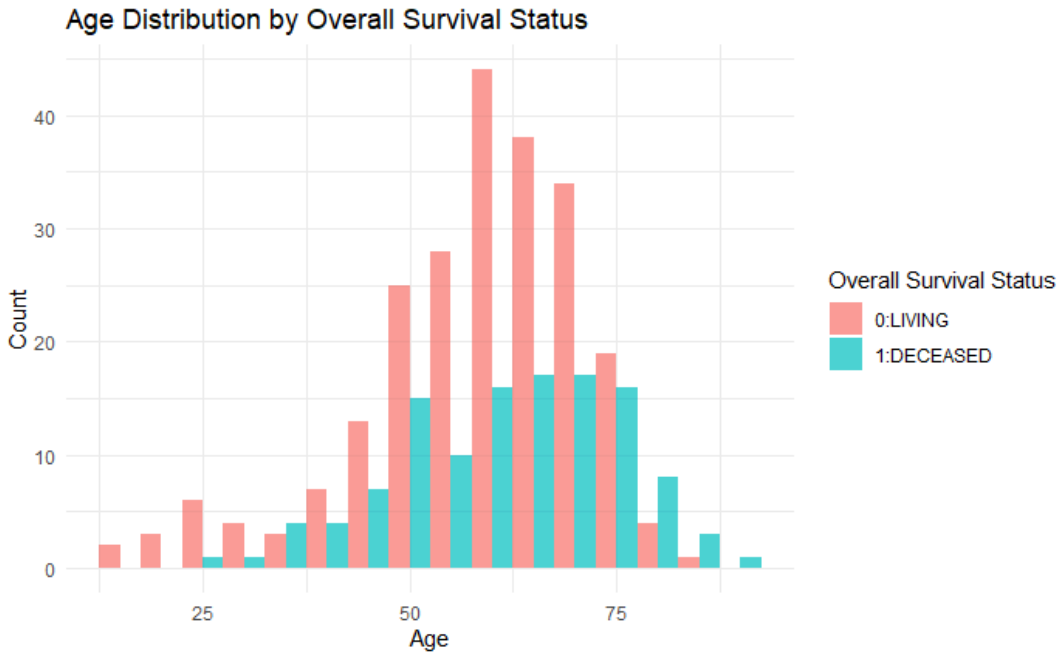


***Figure 3**: Age vs Progression-Free Survival (PFS) Status*

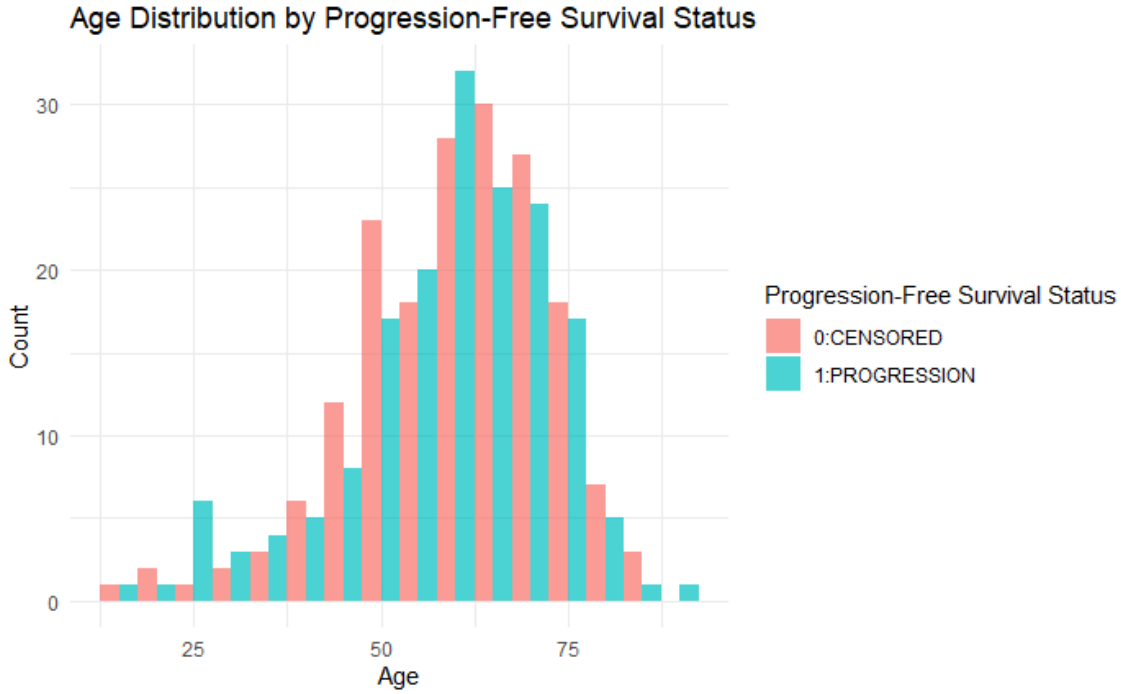


***Figure 4**: Age Distribution with Boxplot Grouped by Tumor Stage*

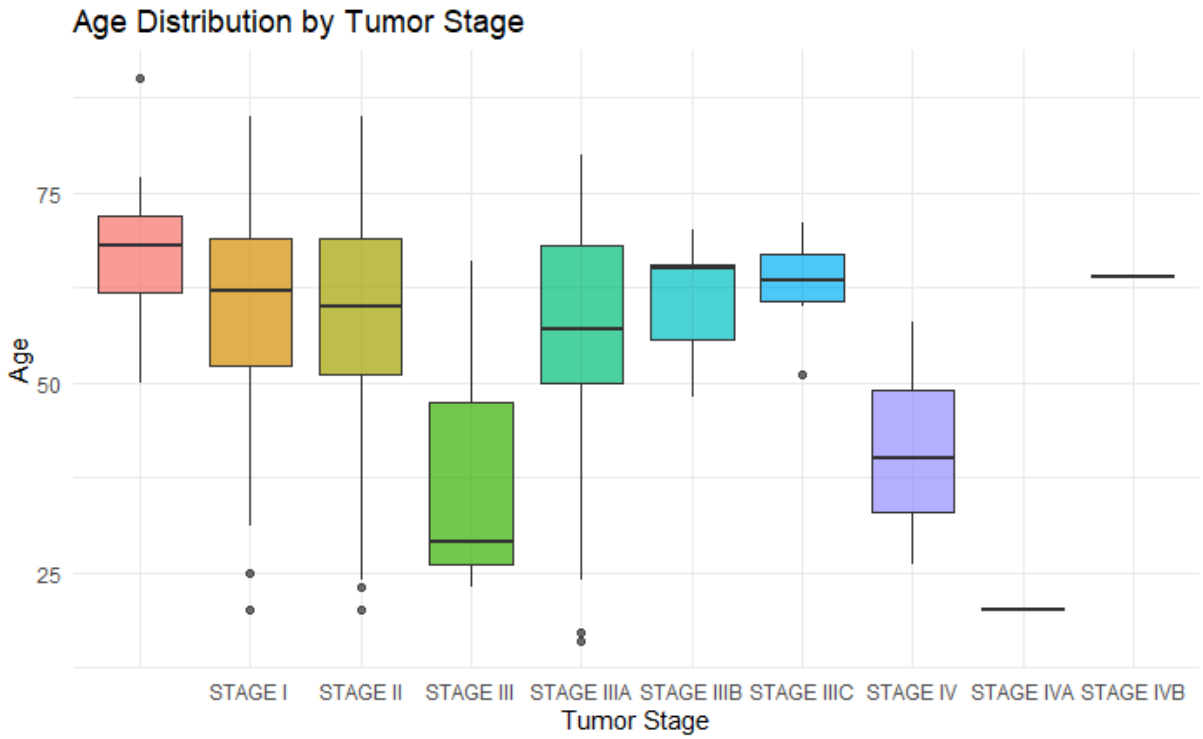
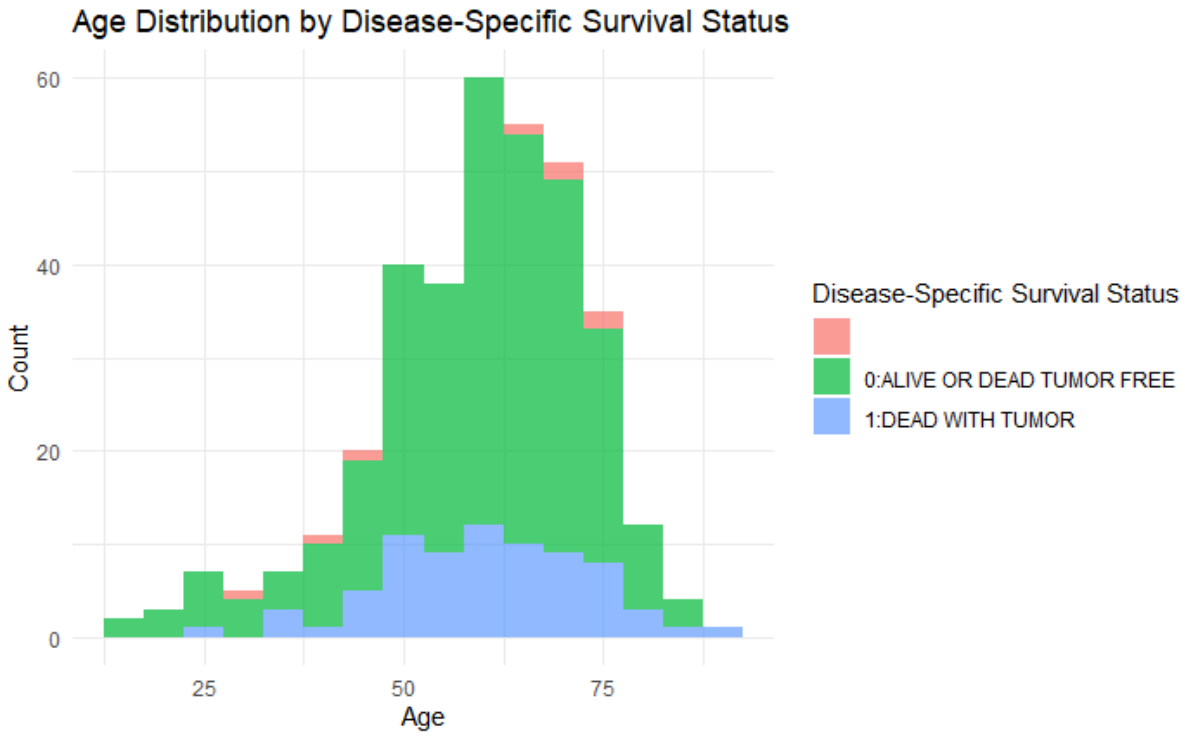


***Figure 5**: Age Distribution by Disease-Specific Survival Sta*



tus

***Figure 6**: Kaplan-Meier Analysis by Gender*

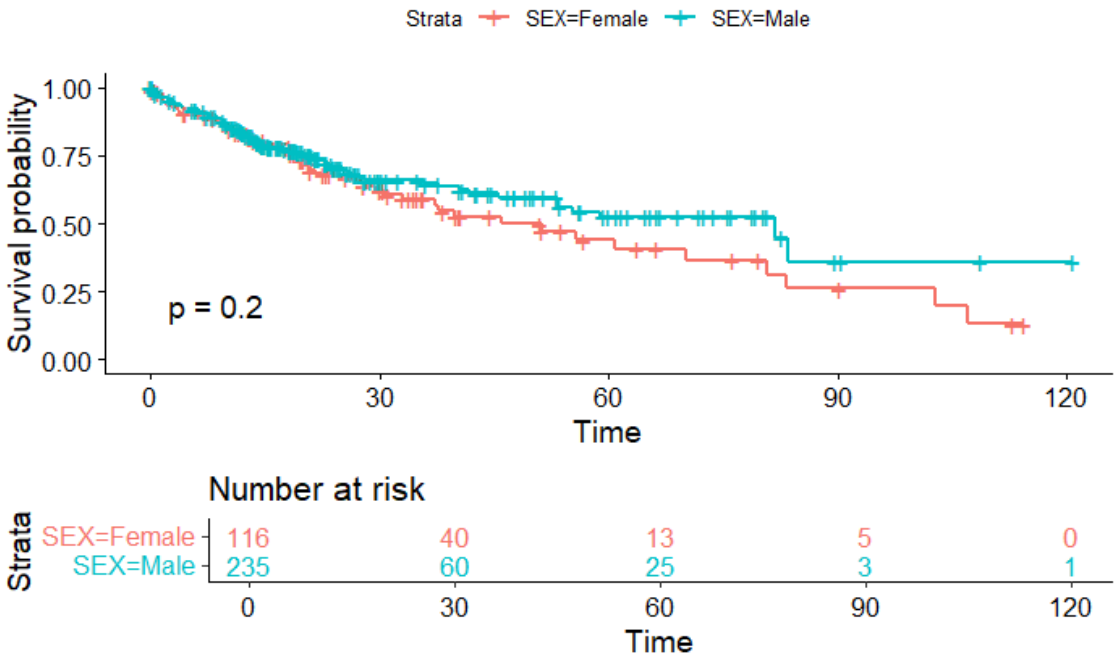


***Figure 7**: Kaplan-Meier Analysis by Weight*

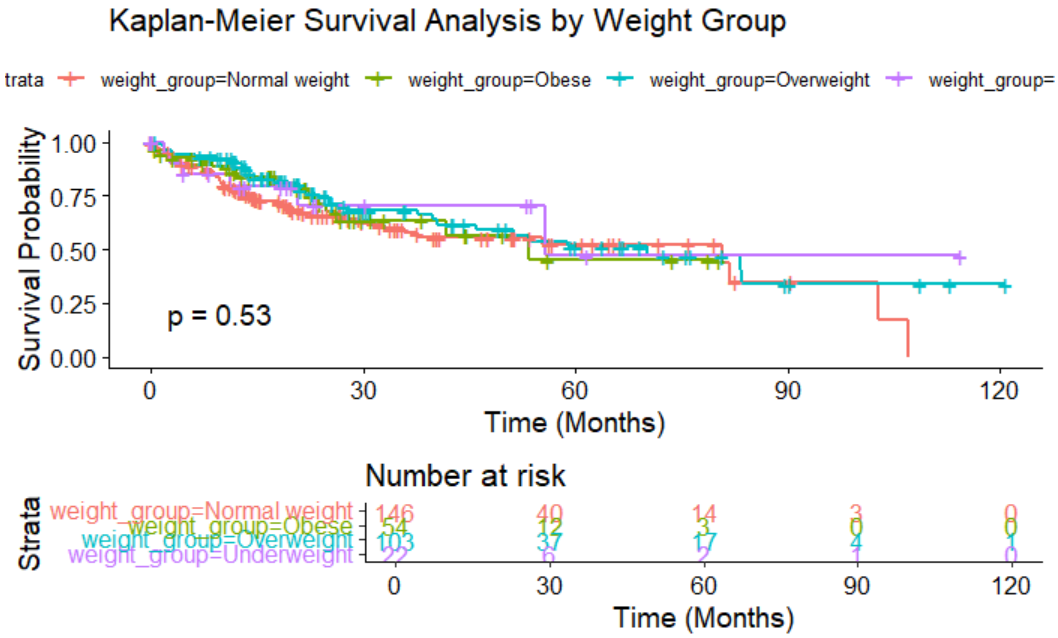


***Figure 8**: Kaplan-Meier Analysis by Tumor Stage*

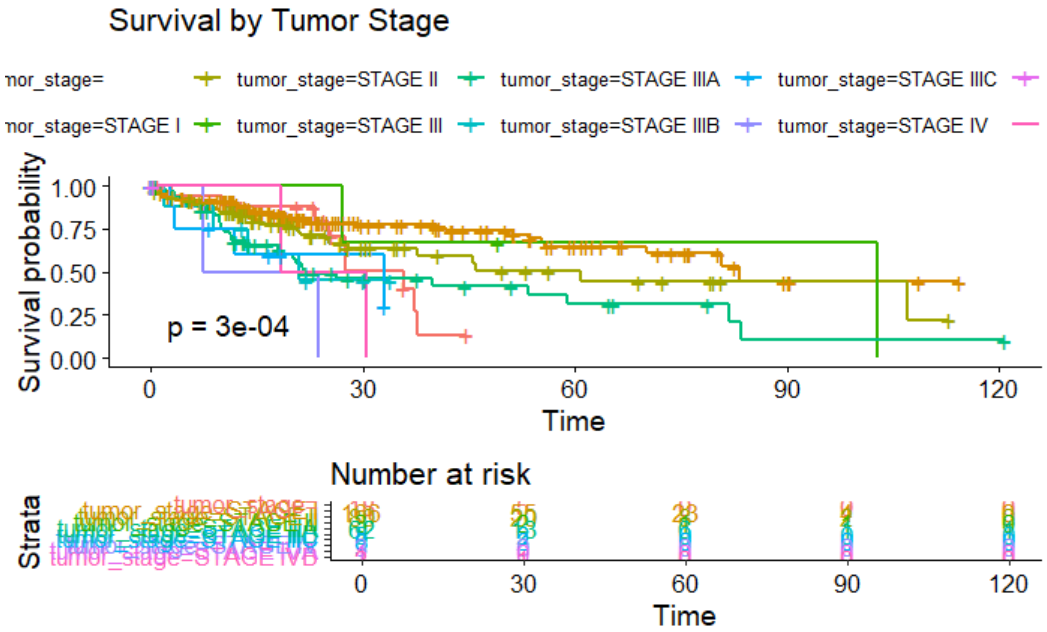


***Figure 9**: Kaplan-Meier Analysis by race*

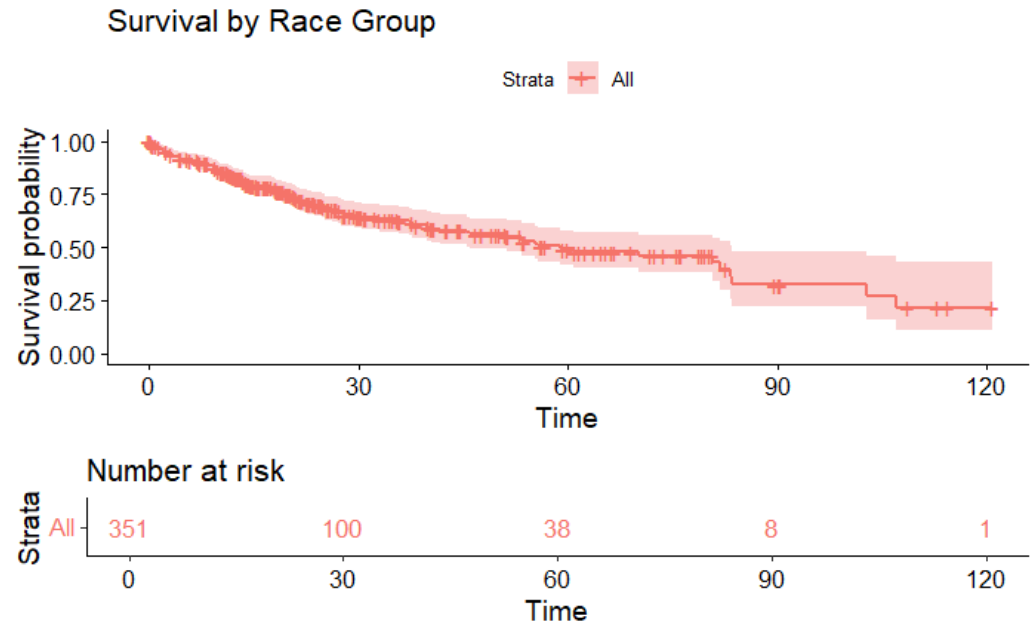


***Figure 10**: Kaplan-Meier Analysis by Age Group*

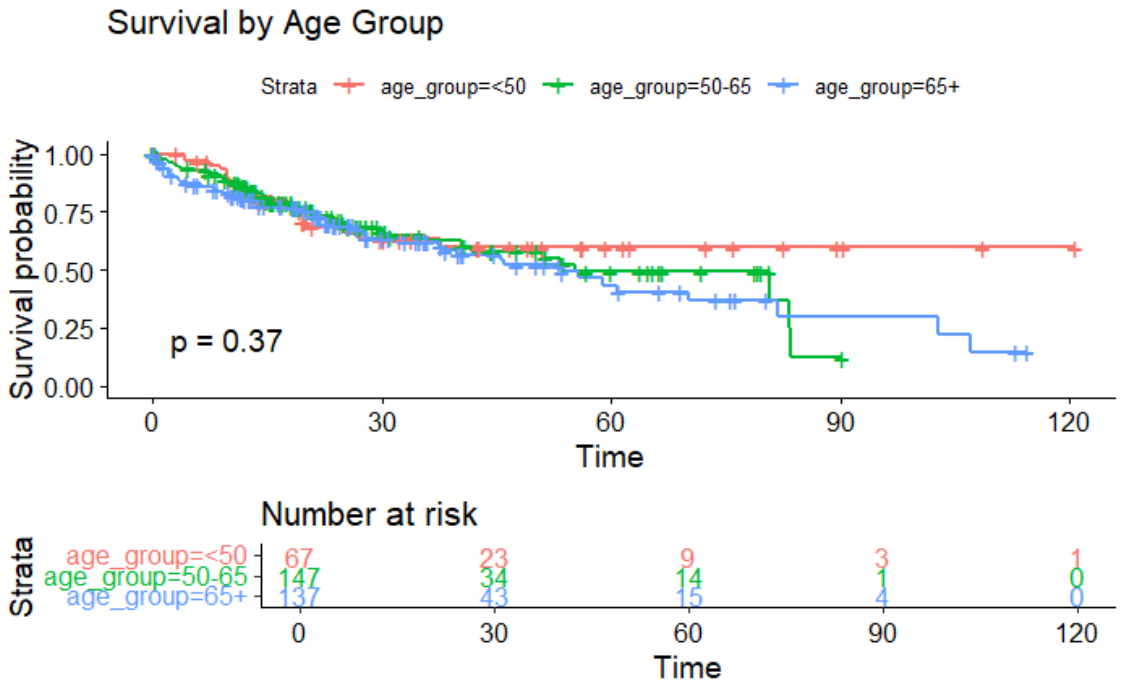


***Figure 11**: heatmap for top 50 Variable Genes*

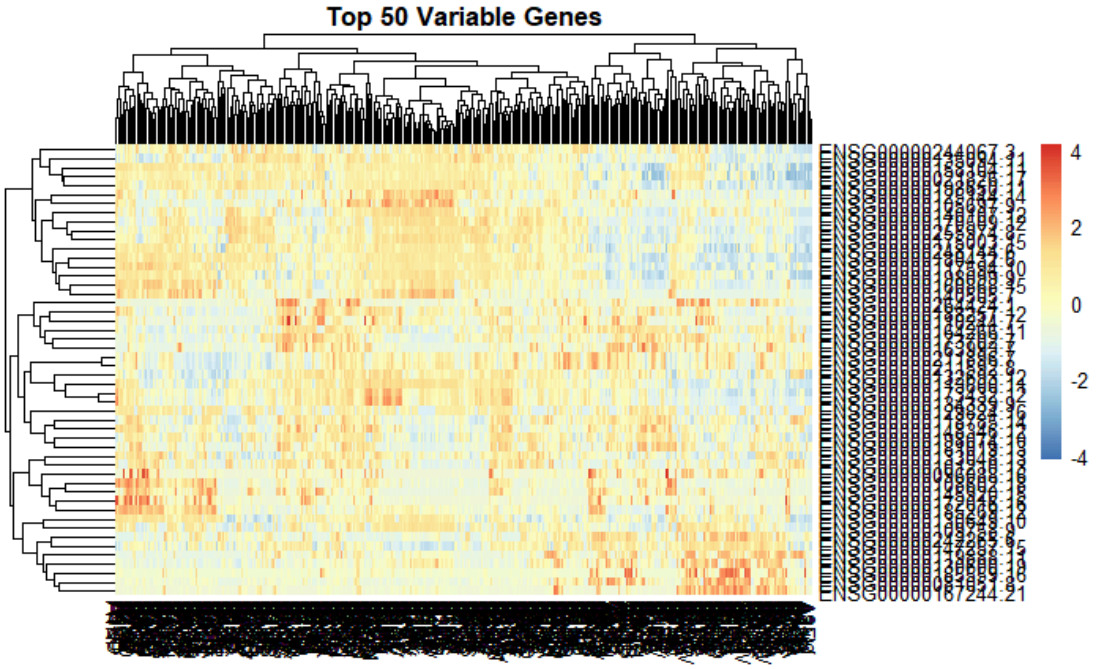


***Figure 12**: Hierarchical Clustering*

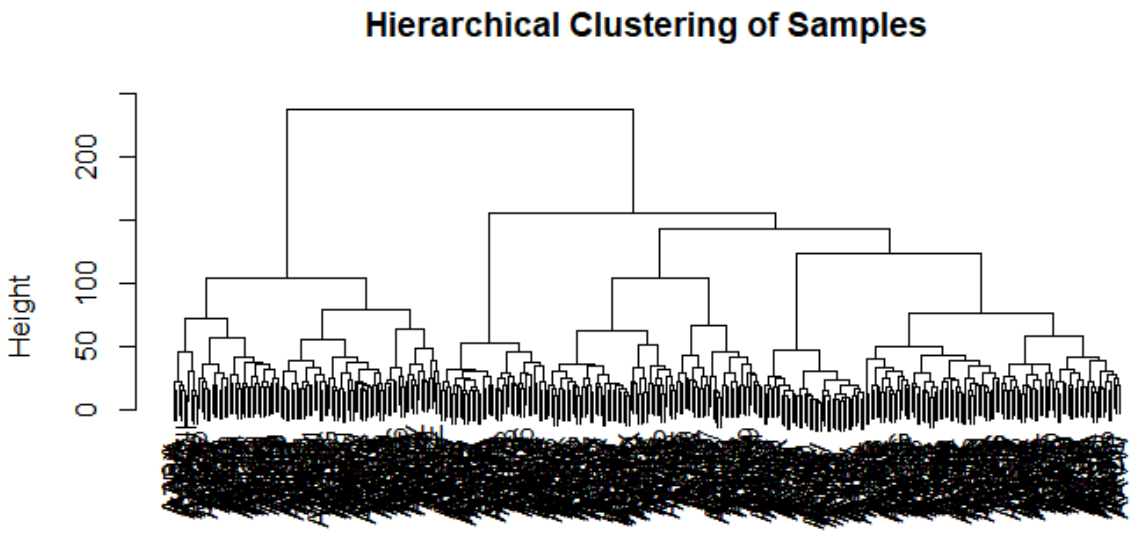


***Figure 13**: PCA Plot*

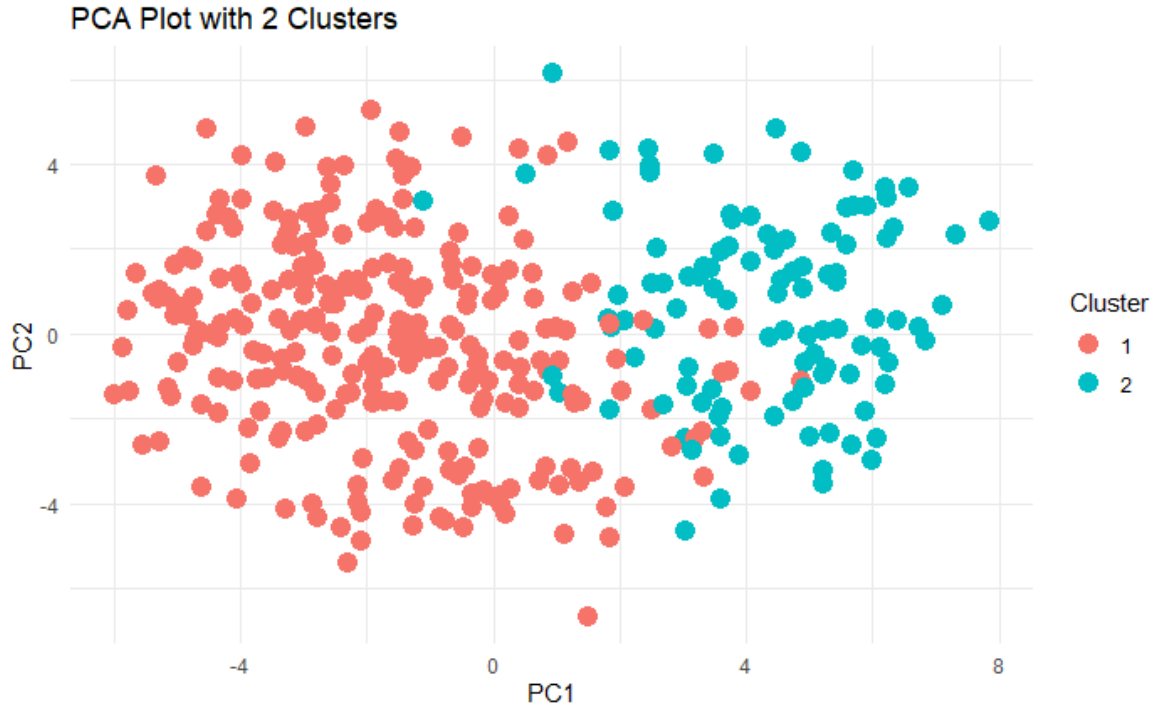


***Figure 14**: top 20 most frequently mutated genes in Cluster 1*

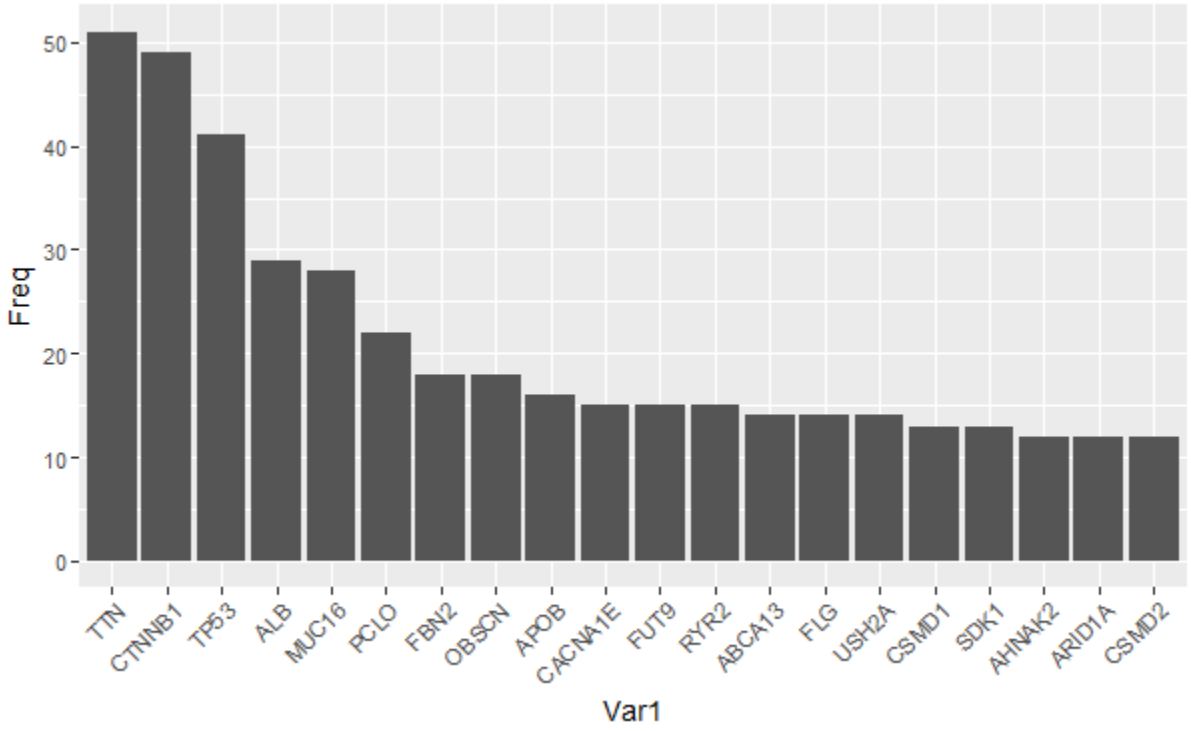


***Figure 15**: top 20 most frequently mutated genes in Cluster 2*

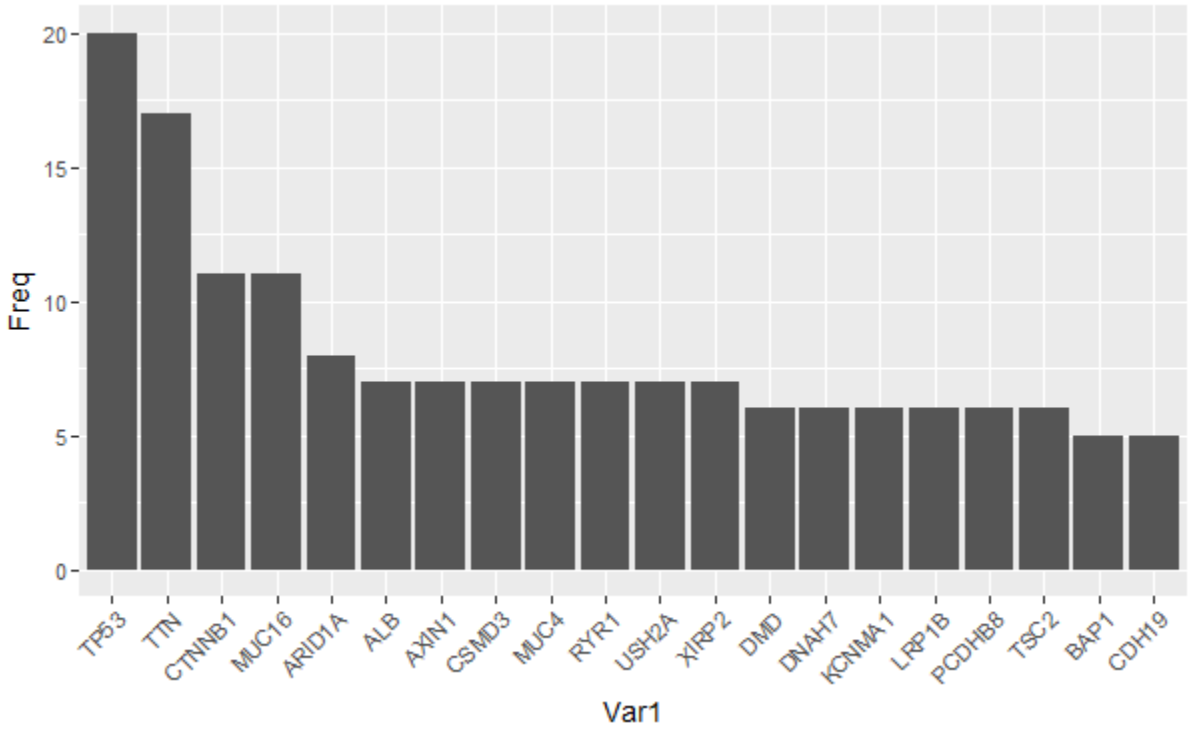


***Figure 16**: top 15 genes from both clusters*

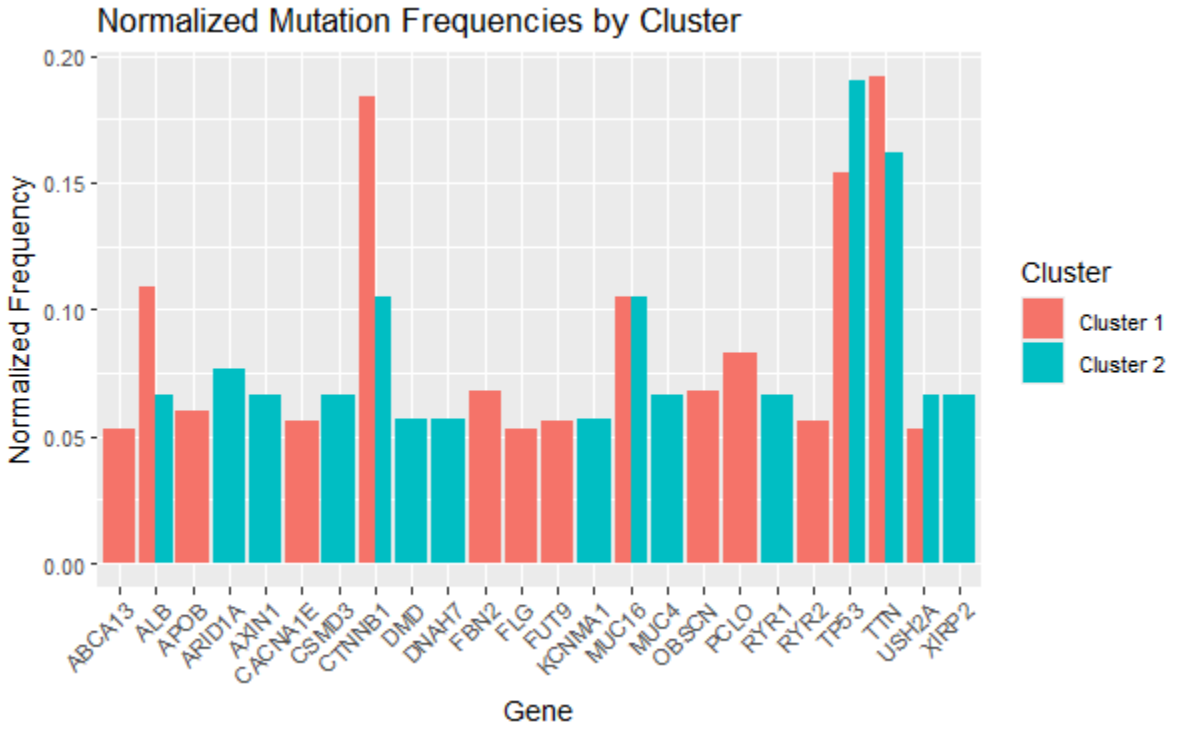


***Figure 17**: top 20 mutated chromosomes in Cluster 1*

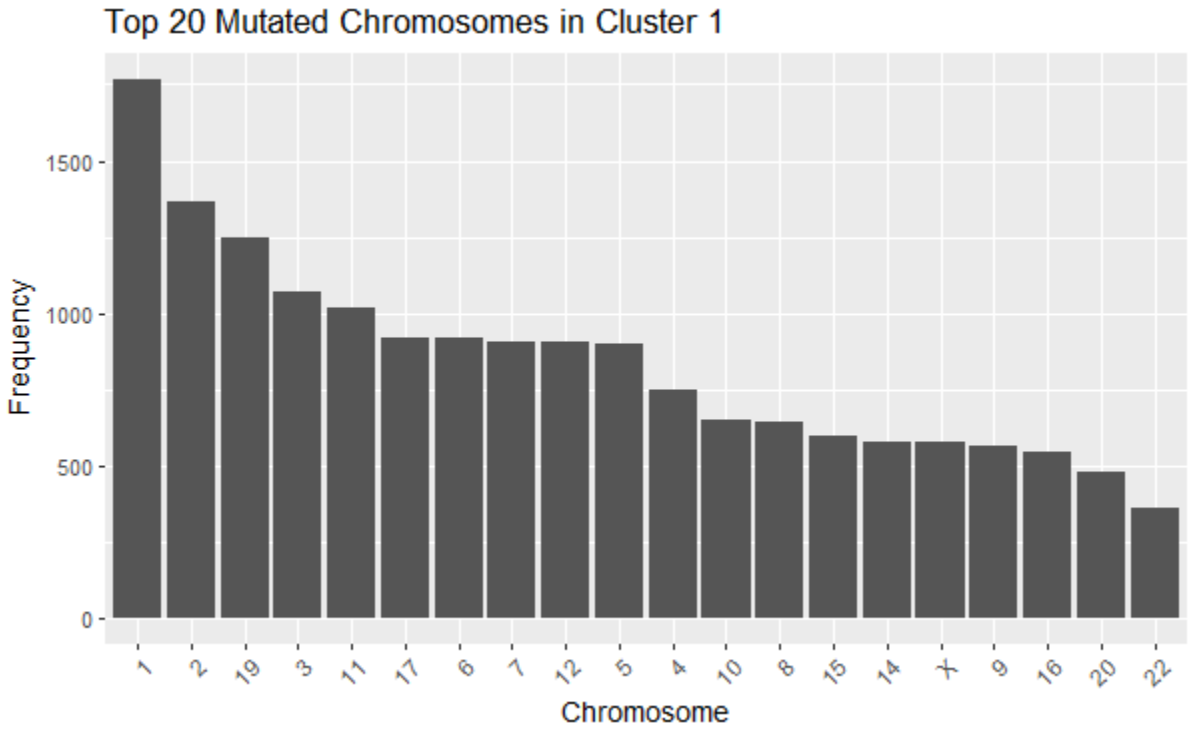


***Figure 18**: top 20 mutated chromosomes in Cluster 2*

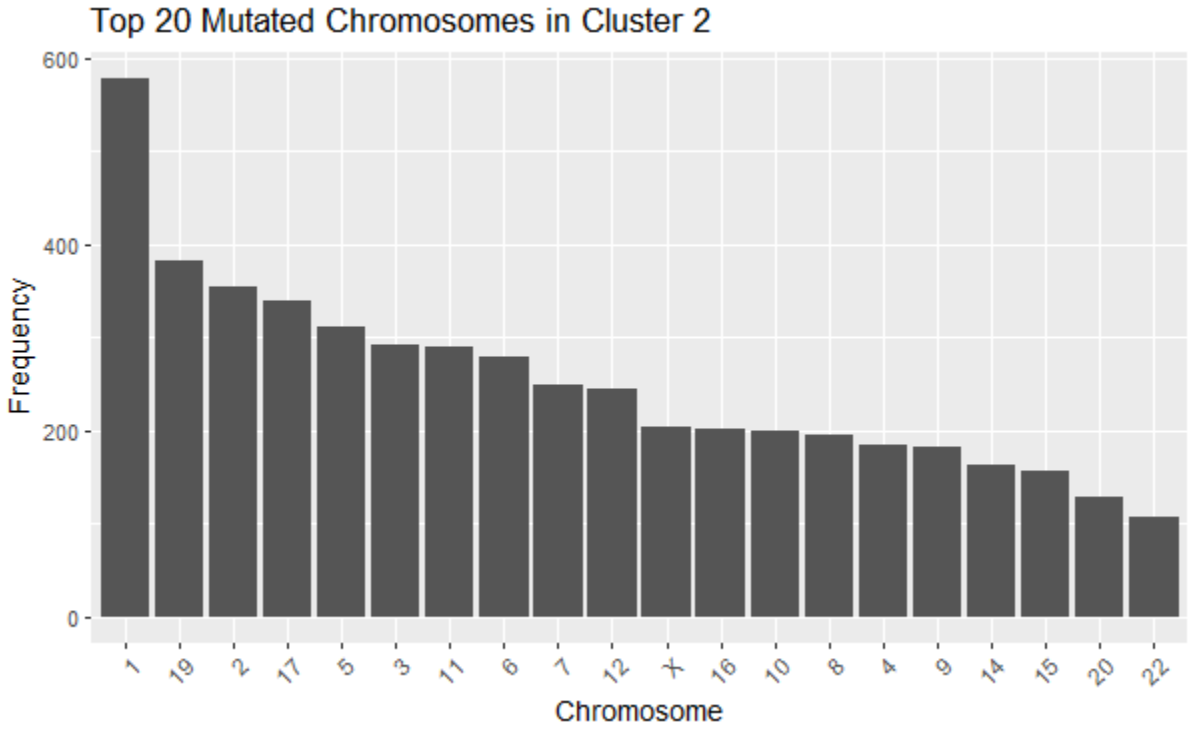


***Figure 19**: normalized mutation frequencies by Chromosome*

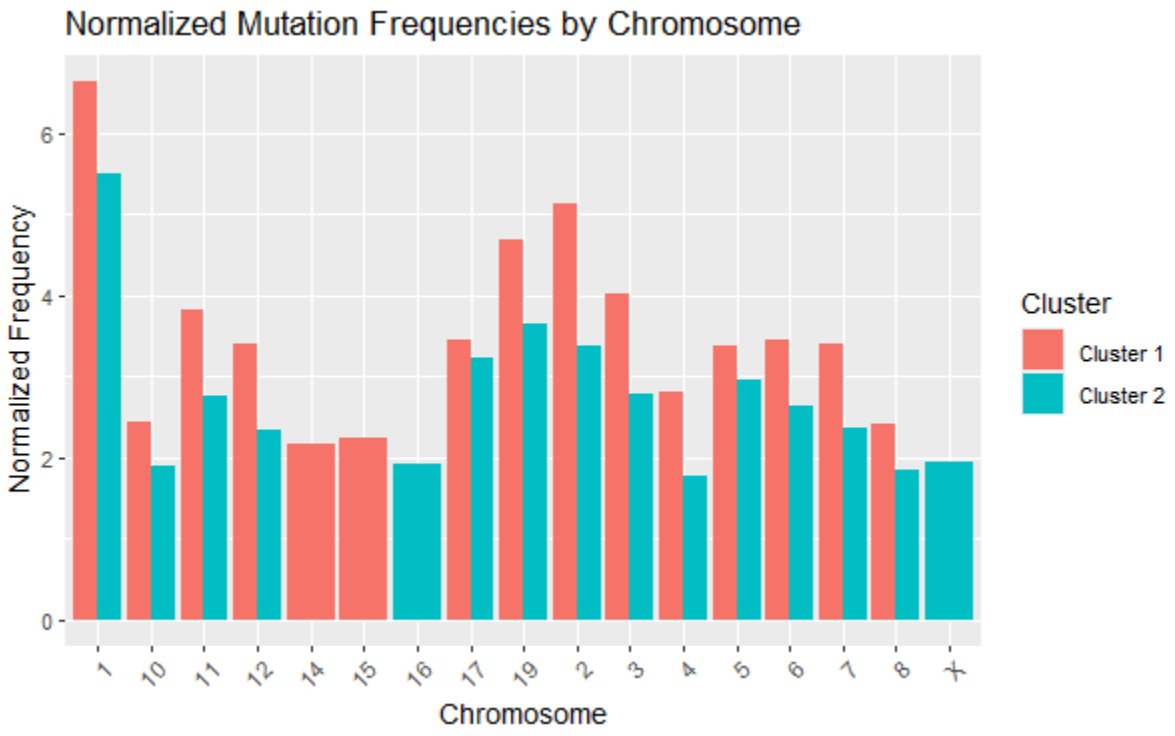


***Figure 20**: top 20 most frequent consequences in Cluster 1*

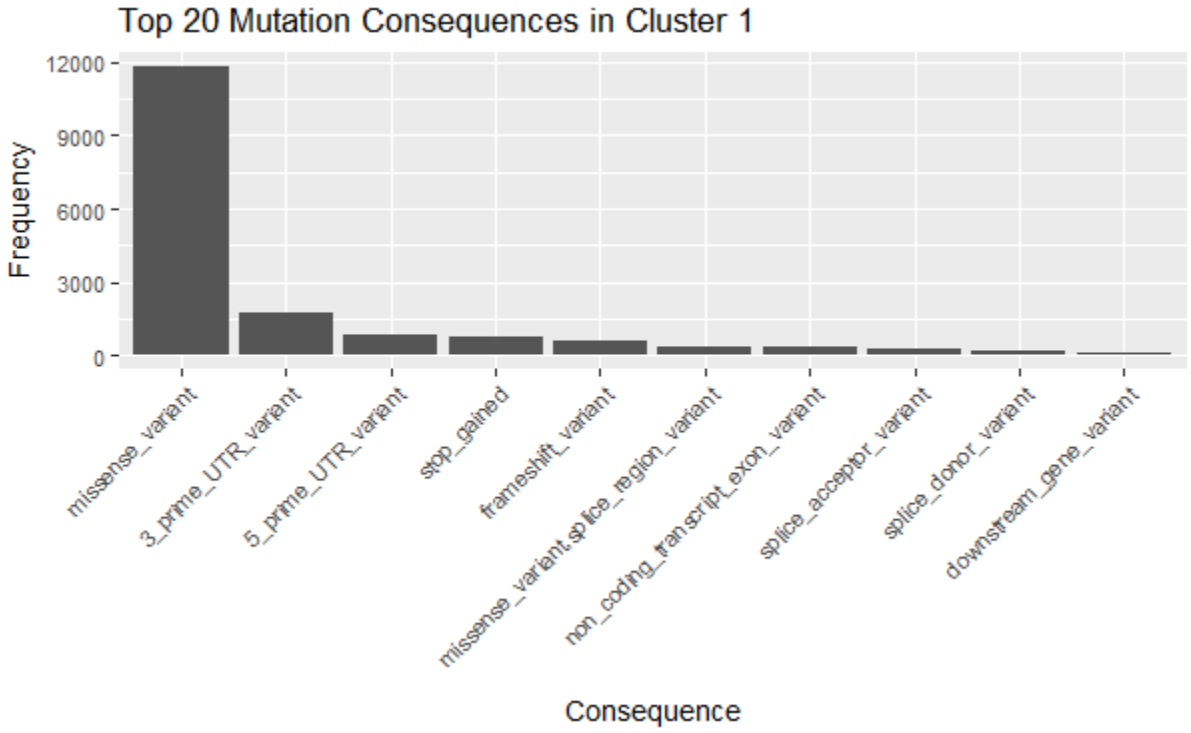


***Figure 21**: top 20 most frequent consequences in Cluster 2*

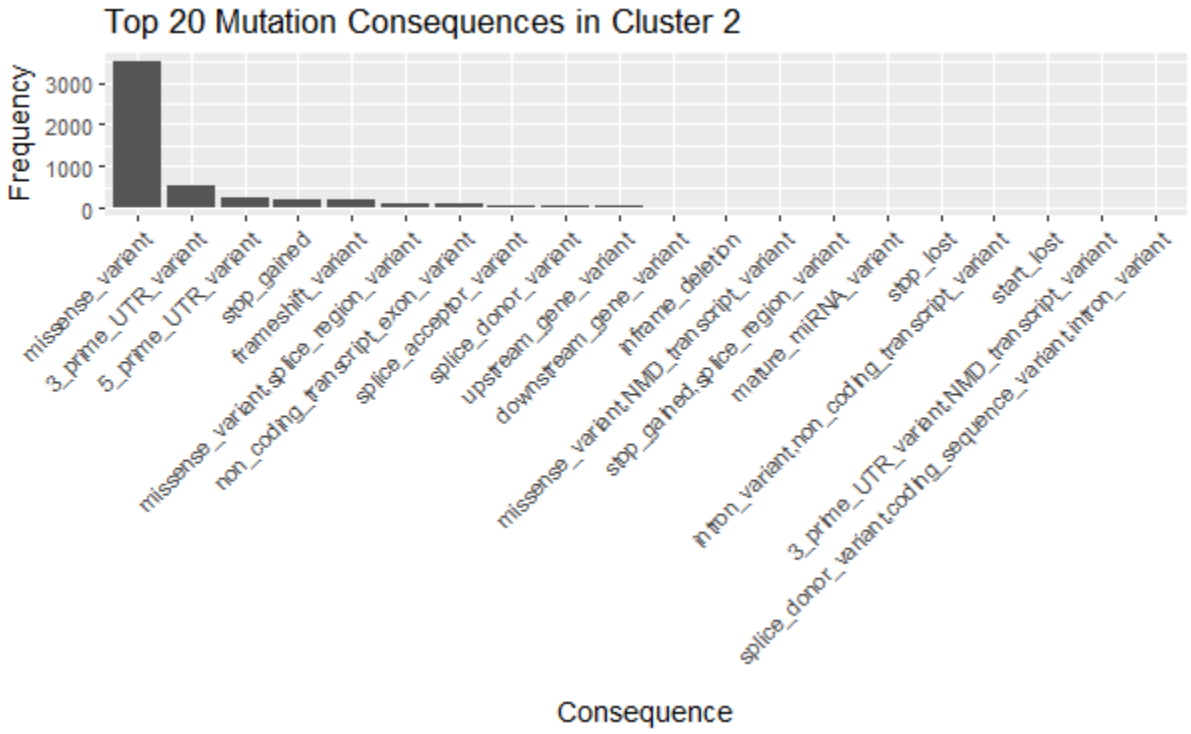


***Figure 22**: normalized mutation consequence frequencies by cluster*

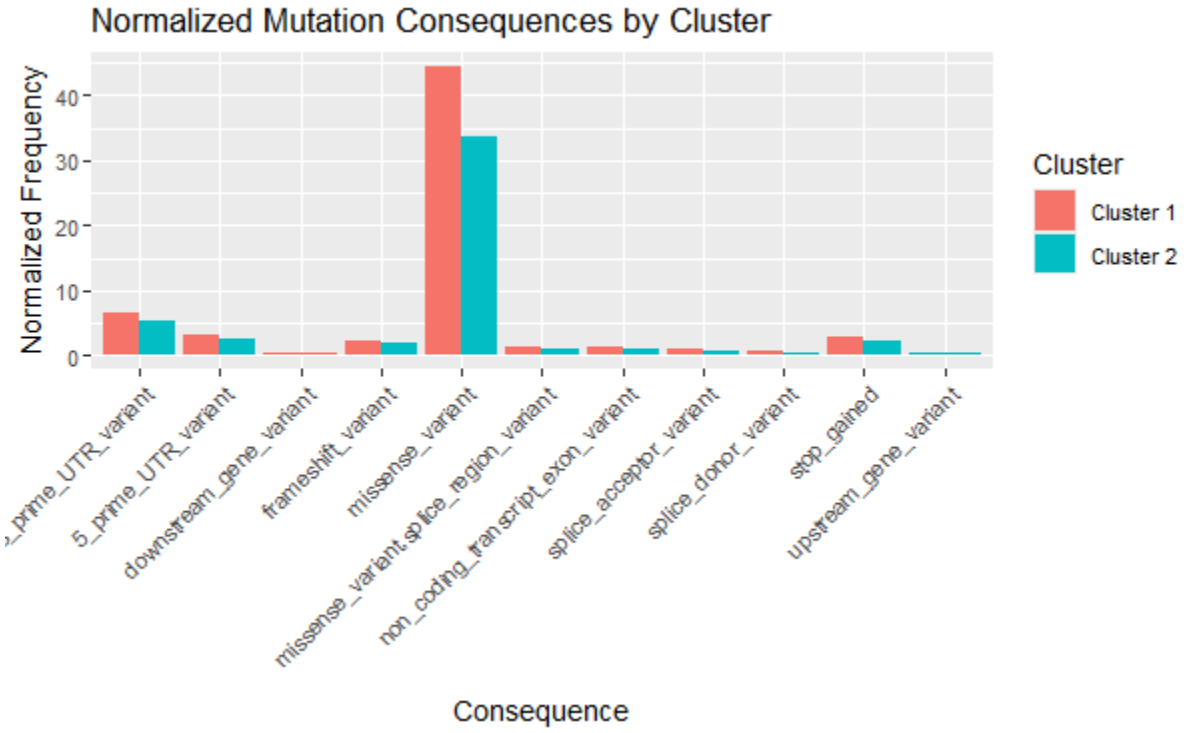


***Figure 23**: Kaplan-Meier Analysis of Clusters by Survival*

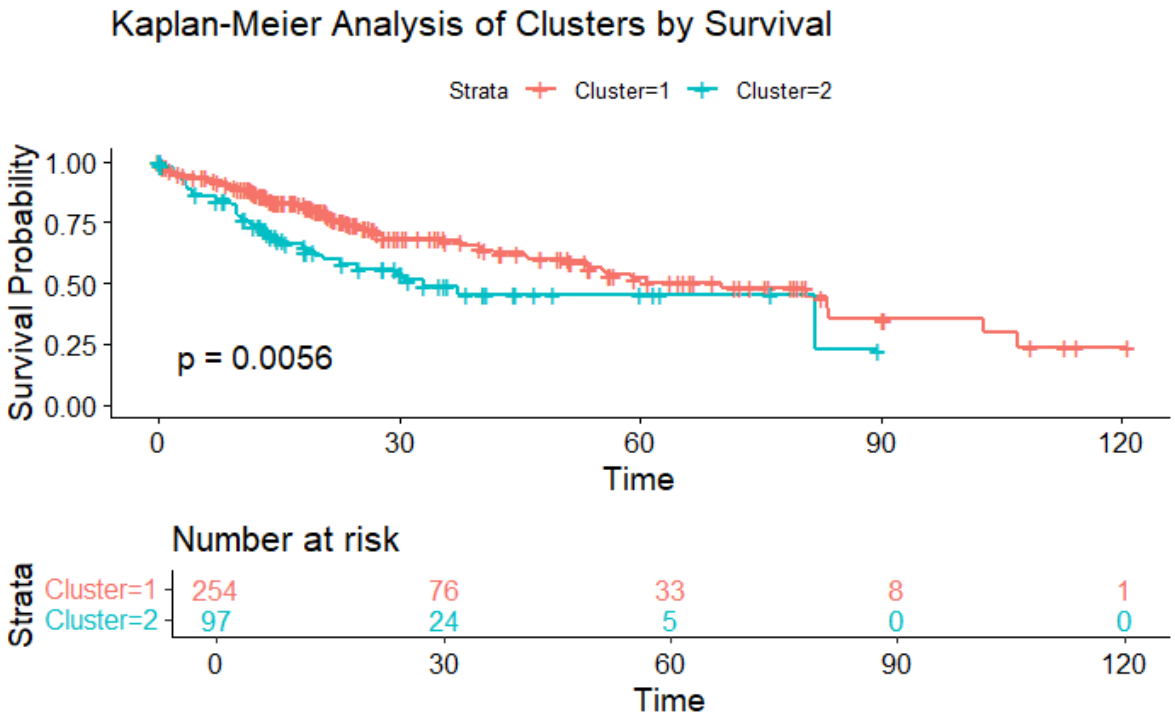


***Figure 24**: Kaplan-Meier Analysis of Clusters by Gender*

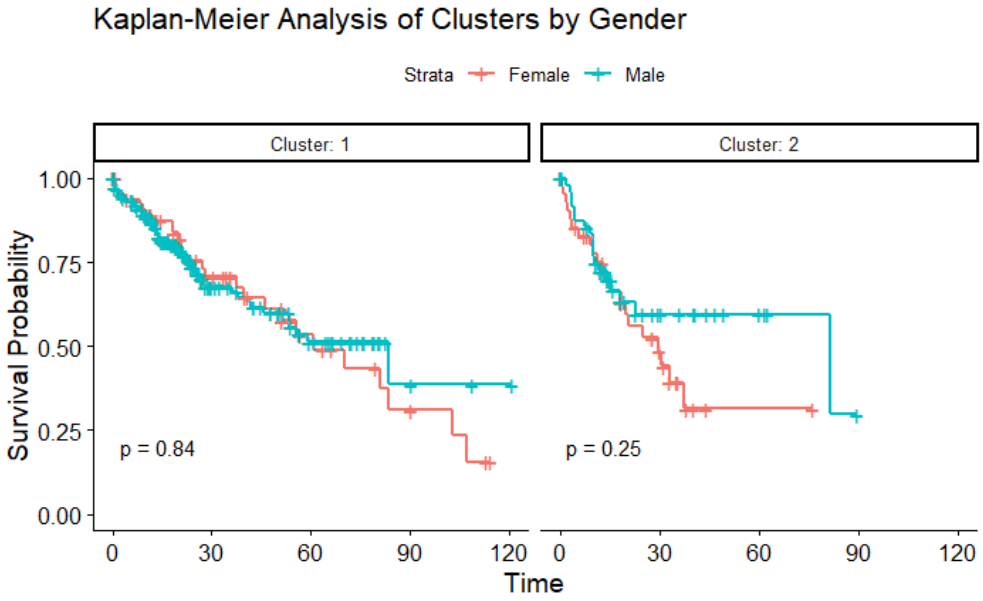


***Figure 25**: Kaplan-Meier Analysis of Clusters by Tumor stage*

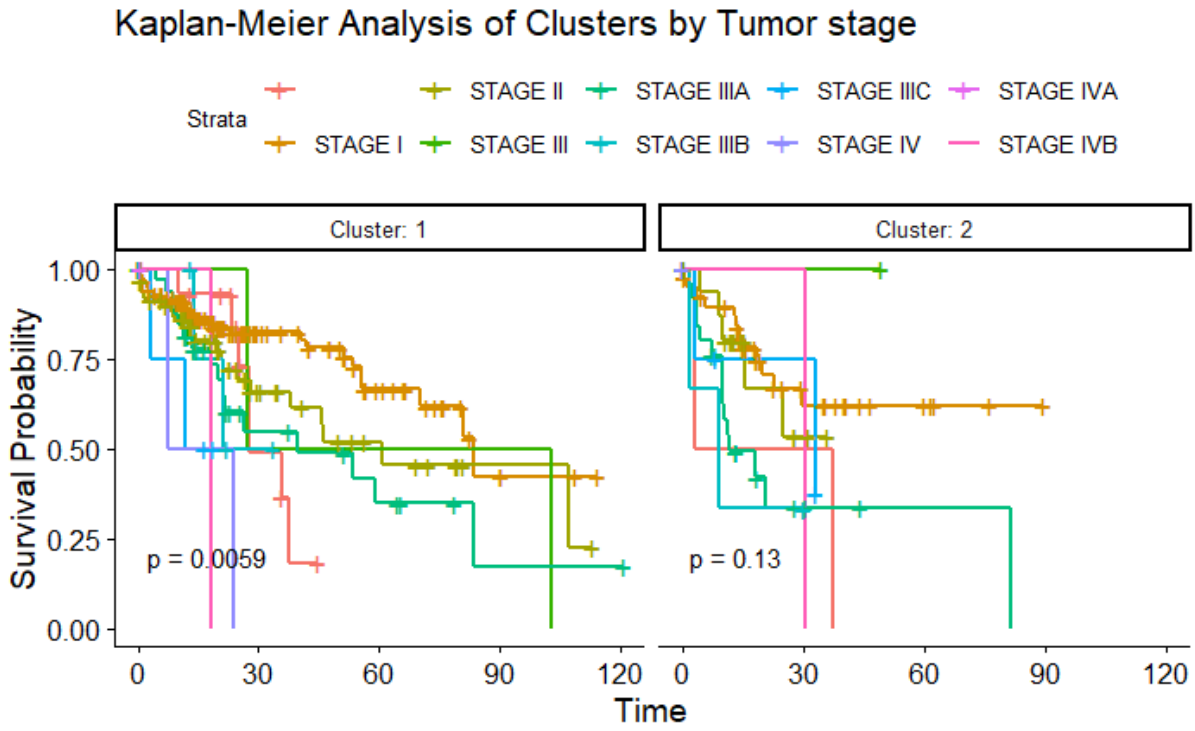


***Figure 26**: Kaplan-Meier Analysis of Clusters by Race*

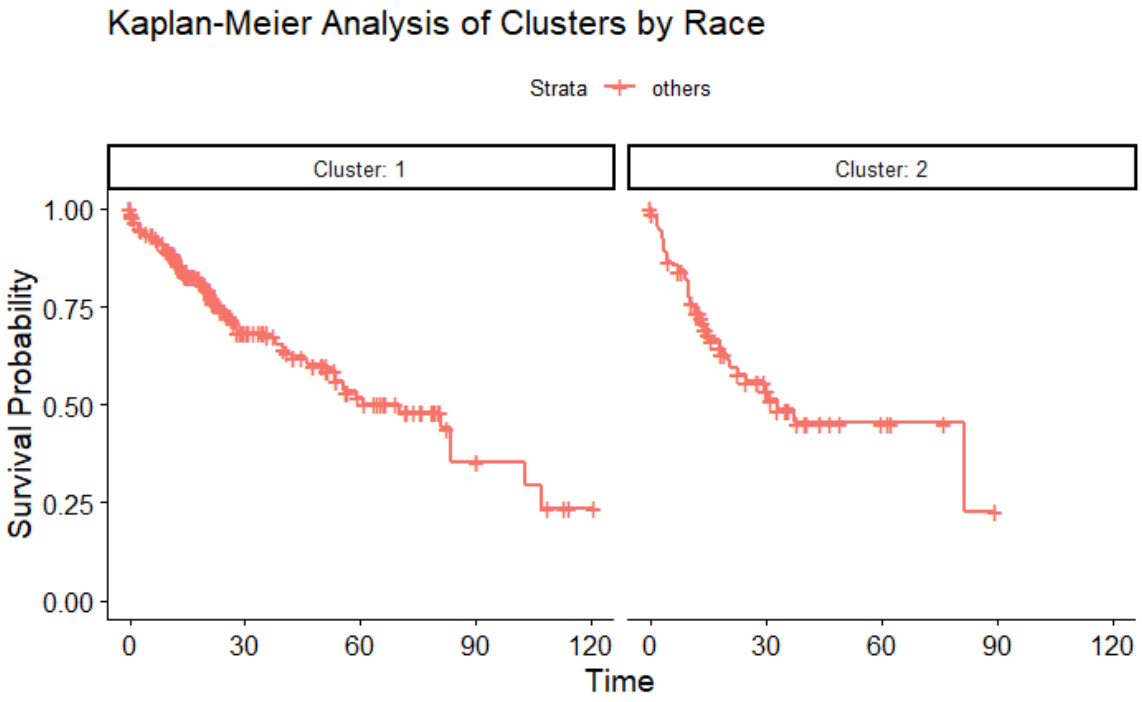


***Figure 27**: Kaplan-Meier Analysis of Clusters by Age*

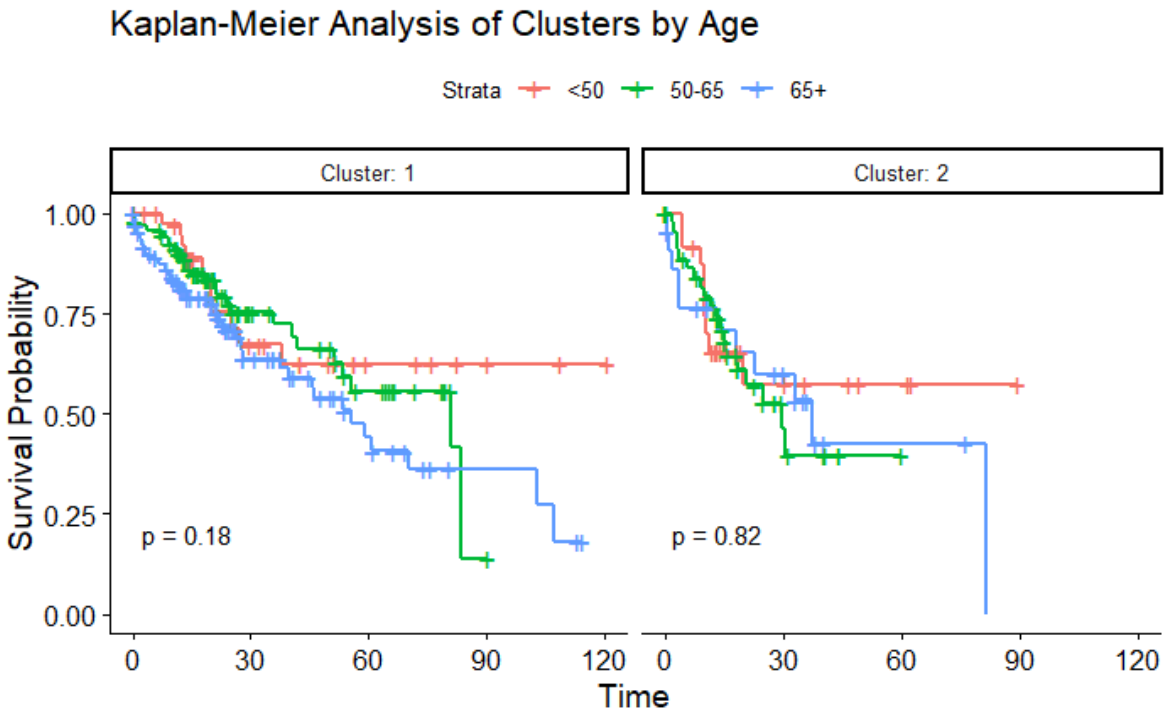


***Figure 28**: Kaplan-Meier Analysis of Clusters by sex*

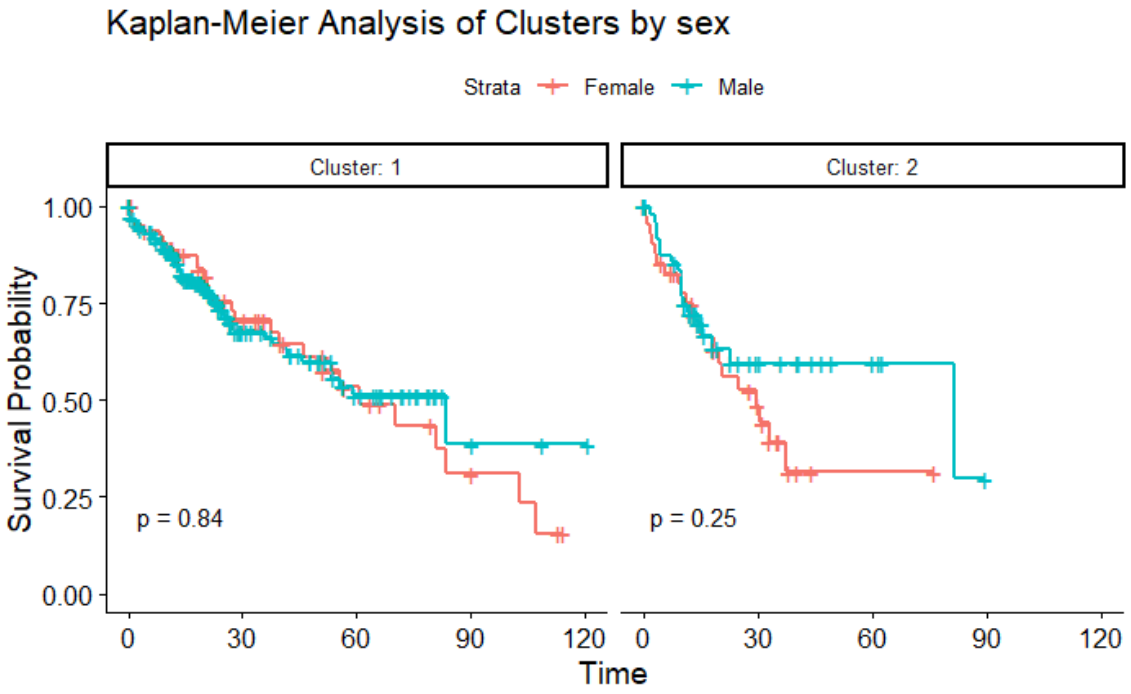


***Figure 29**: Kaplan-Meier Analysis of Clusters by Weight Group*

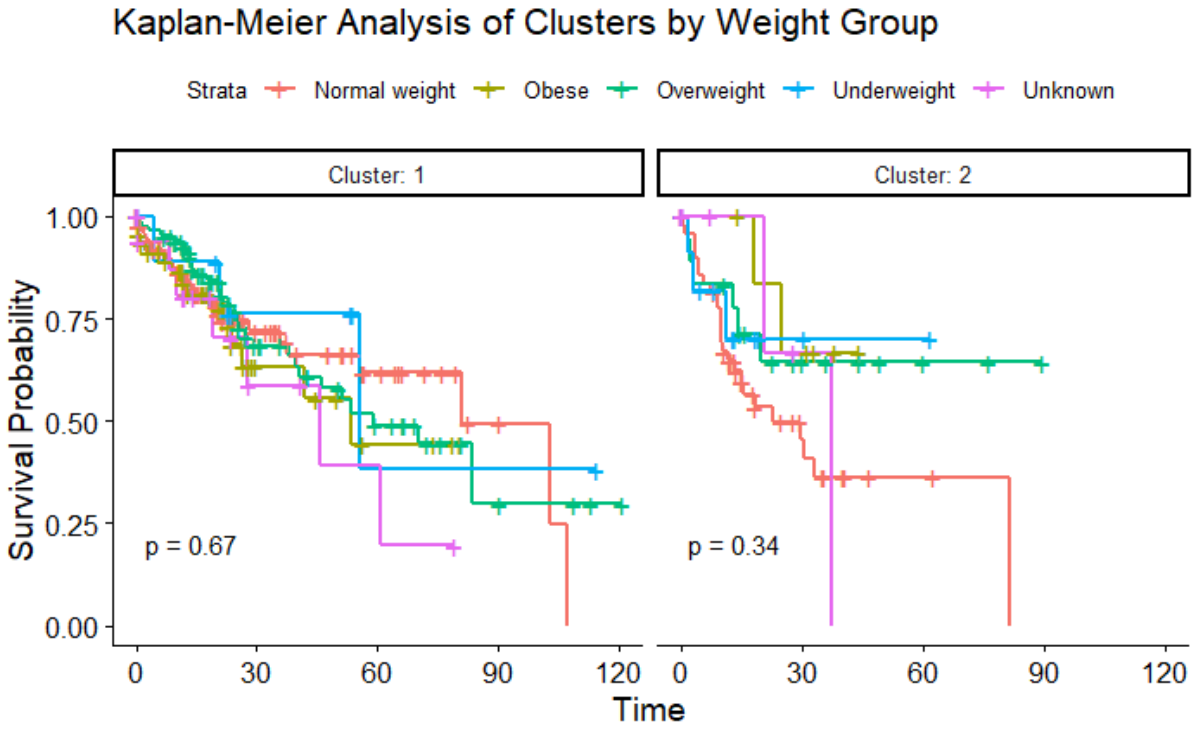


***Figure 30**: Kaplan-Meier Analysis of Clusters by Genetic Ancestry*

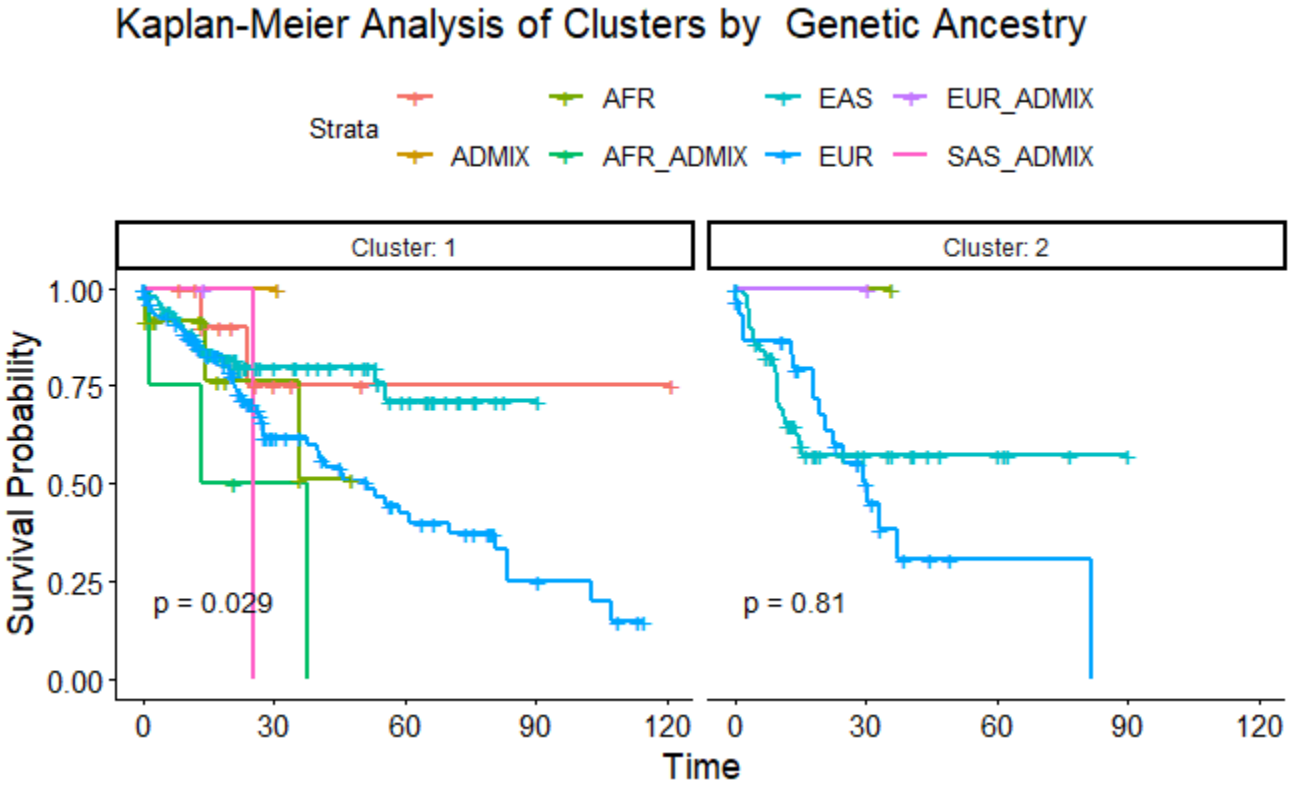


***Figure 31**: Volcano Plot for hoxa1_kd vs control_sirna*

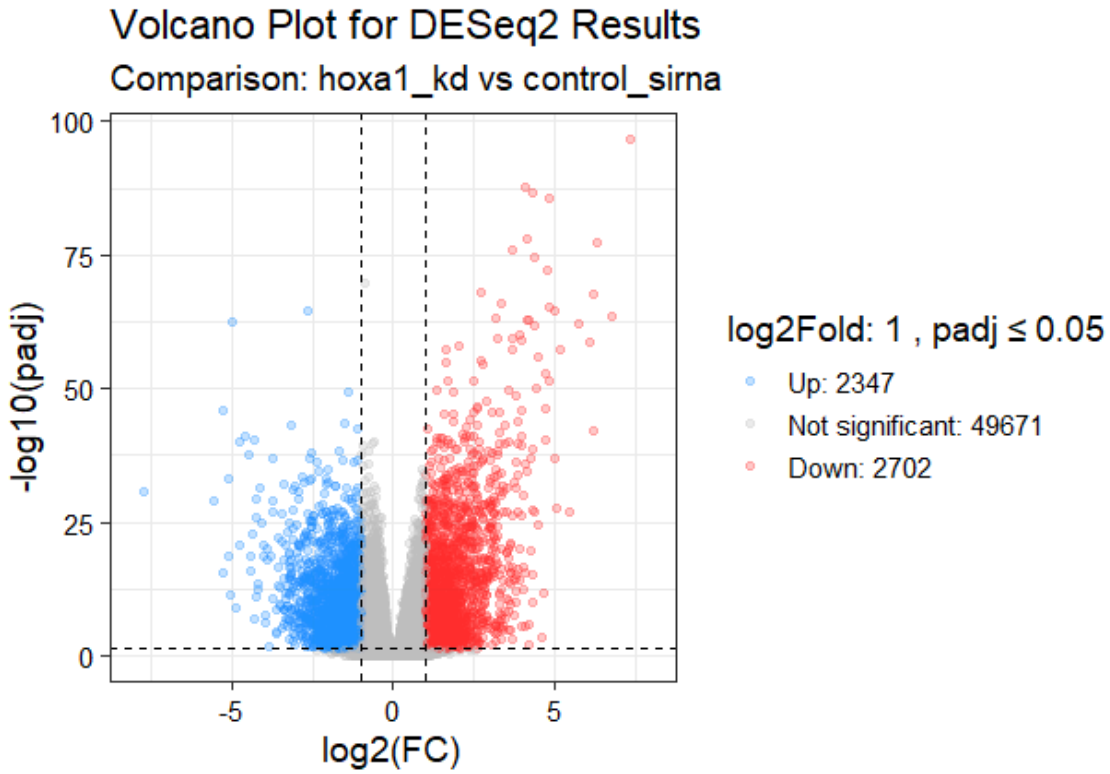


***Figure 32**: MA Plot: Cluster 1 vs Cluster 2*

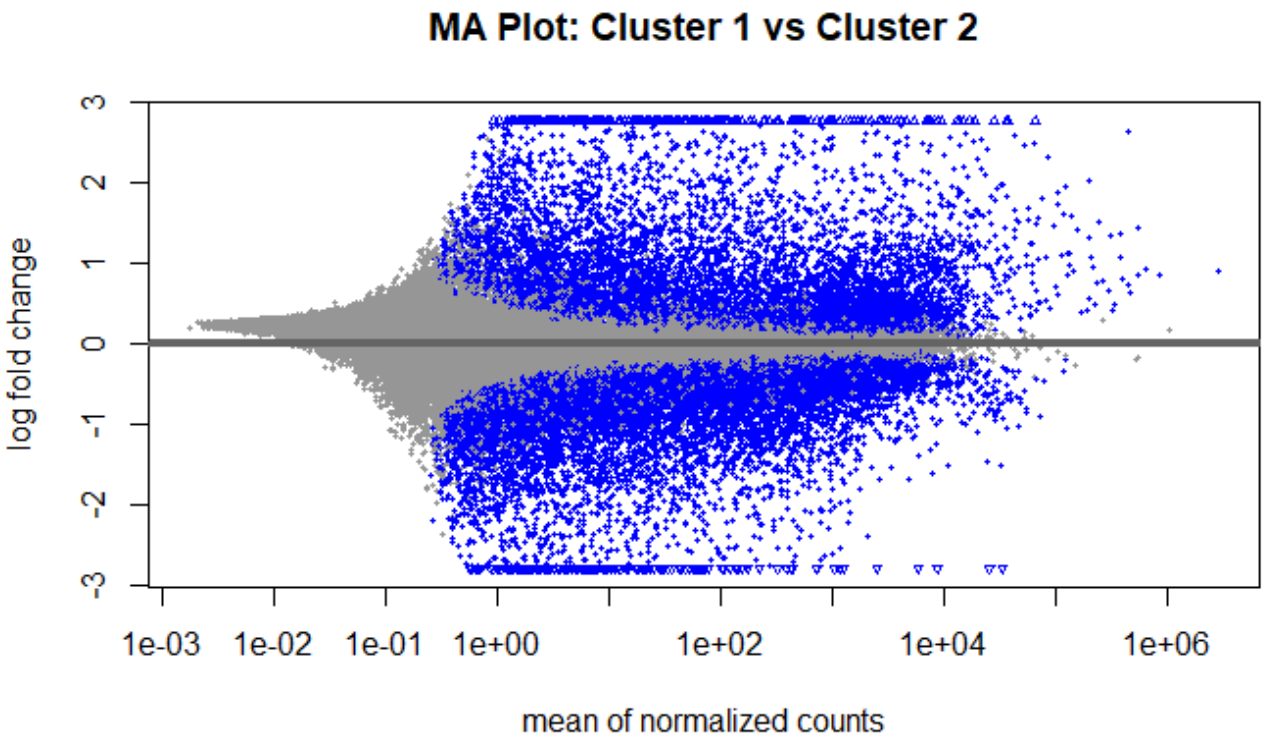


***Figure 33**: Expression of Top DE Genes by Cluster*

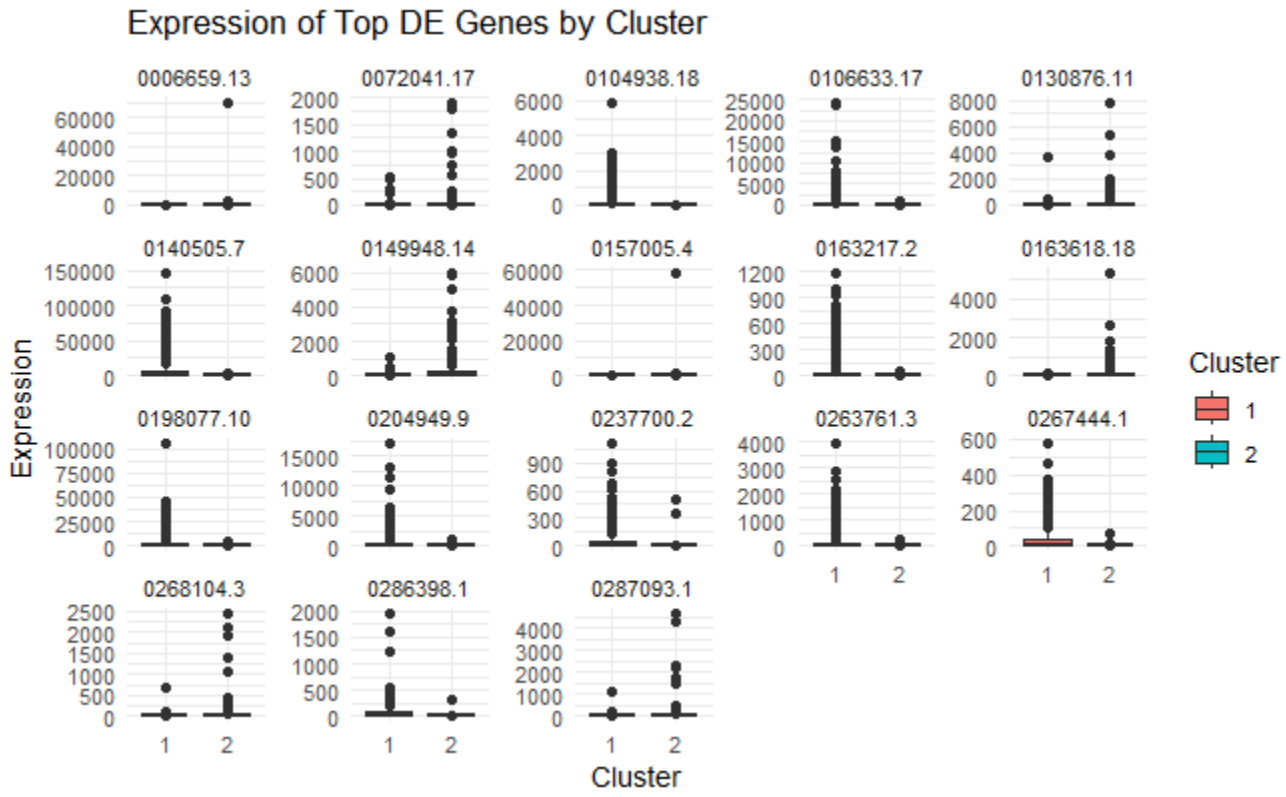


***Figure 34**: hsa00230 pathview*

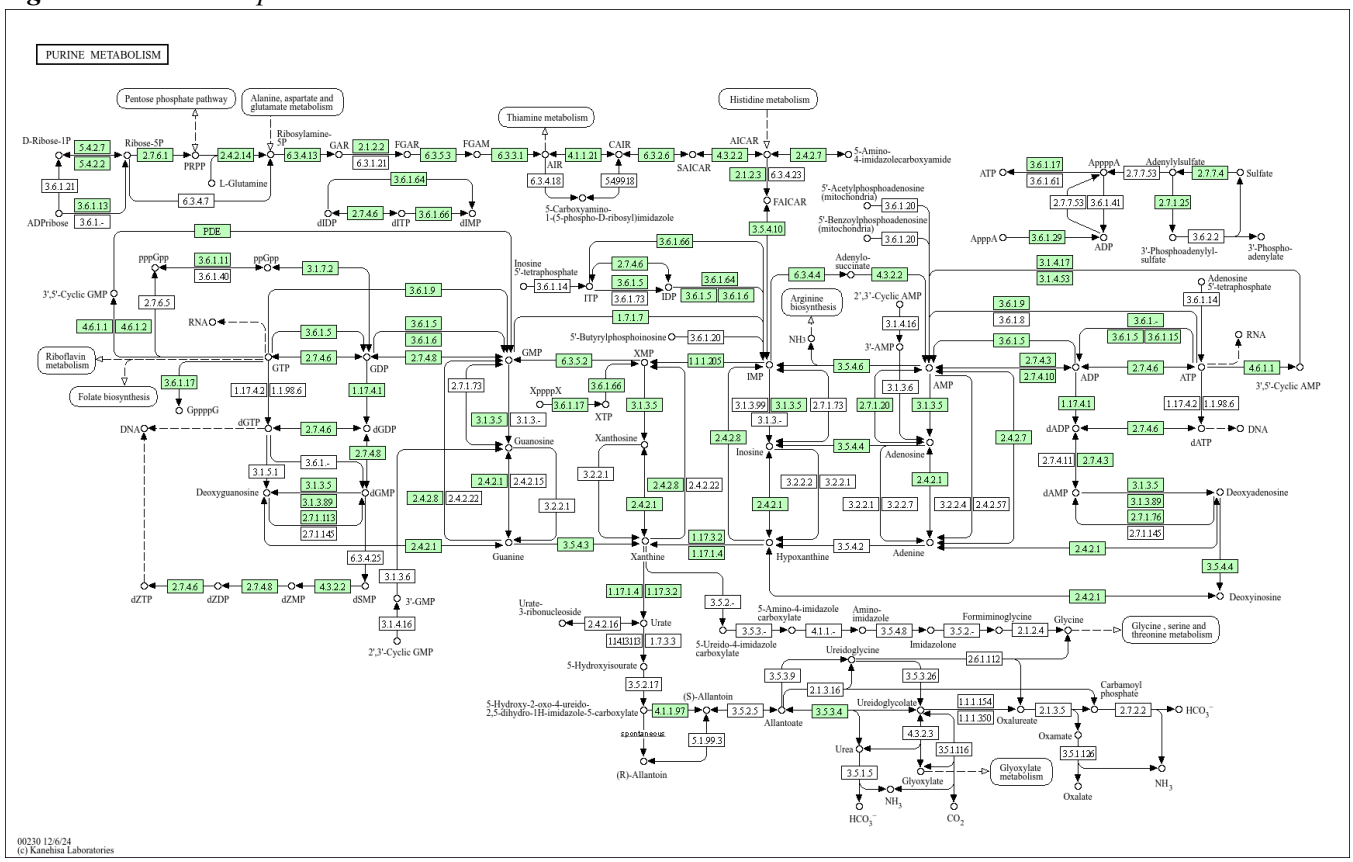


***Figure 35**: hsa00230 Cluster 1*

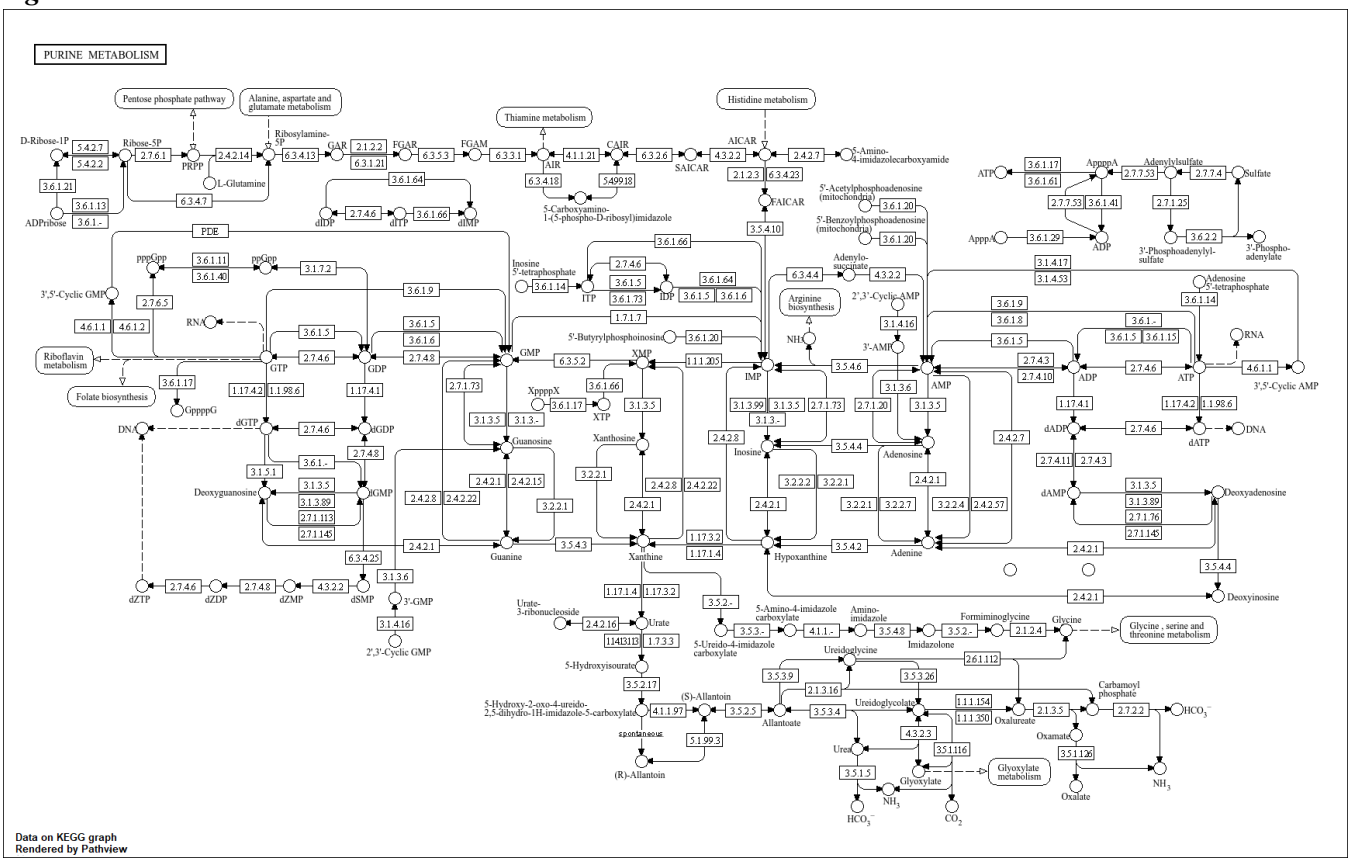


***Figure 36**: hsa00230 Cluster 2*

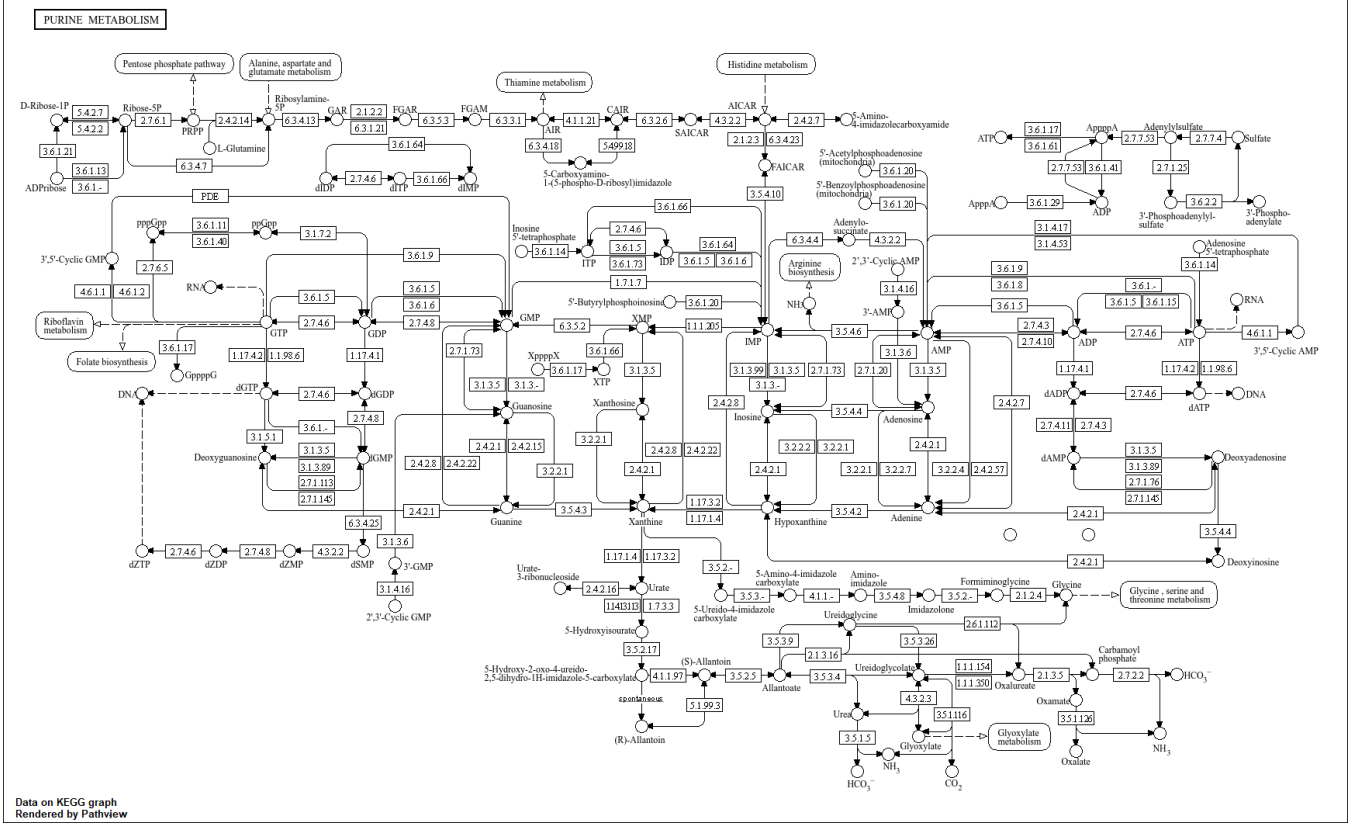


***Figure 37**: hsa00232 pathview*

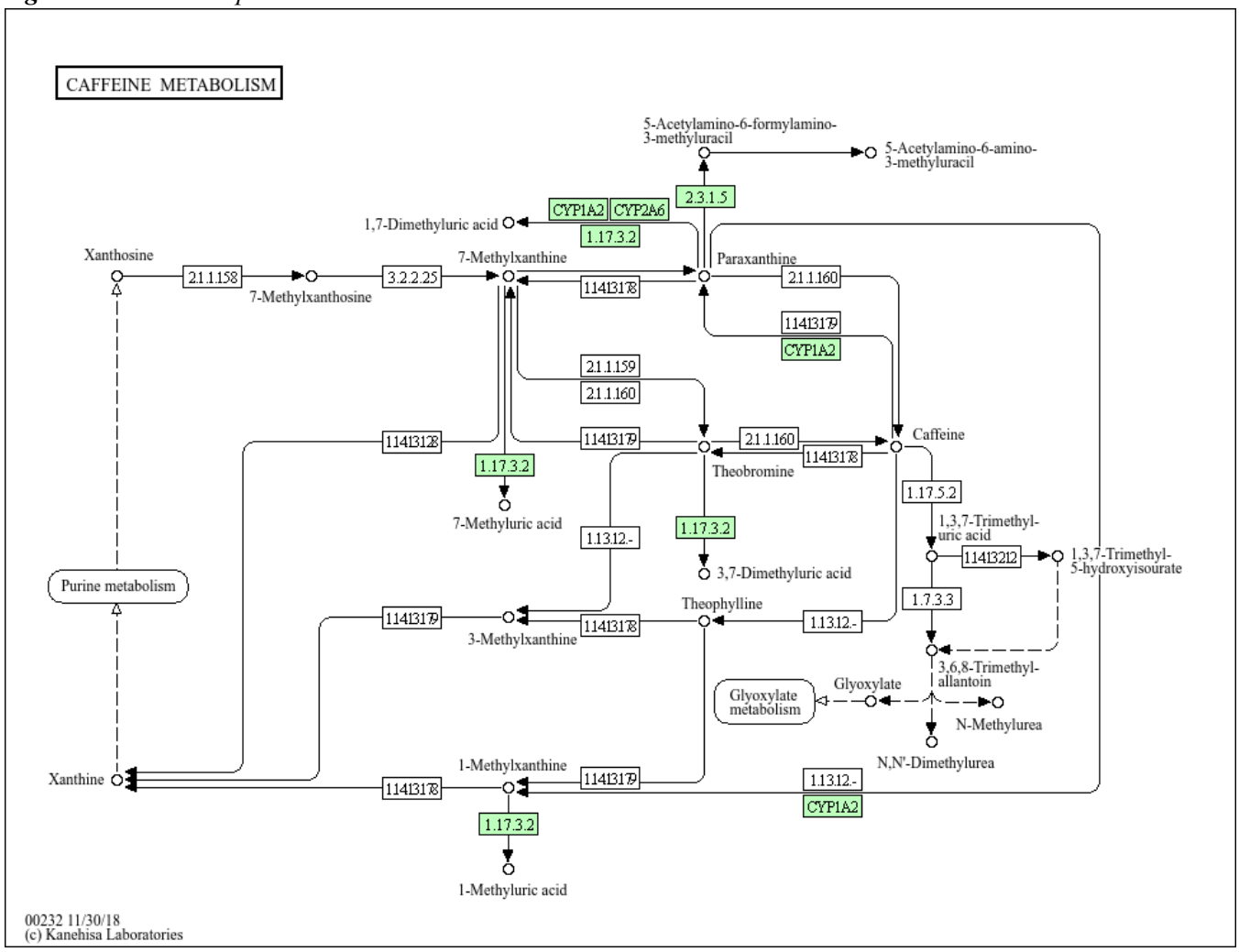


***Figure 38**: hsa00232 Cluster 1*

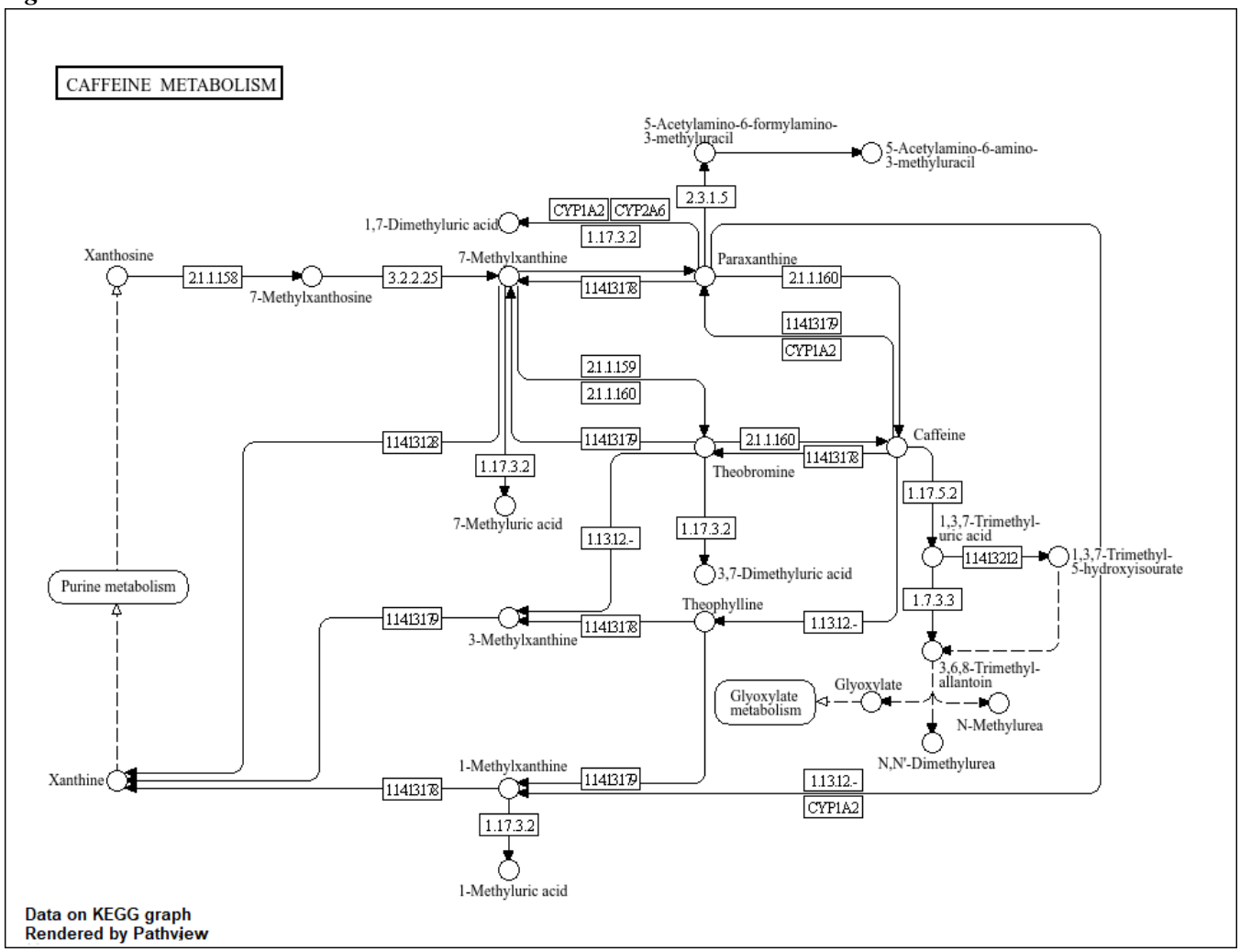


***Figure 39**: hsa00232 Cluster 2*

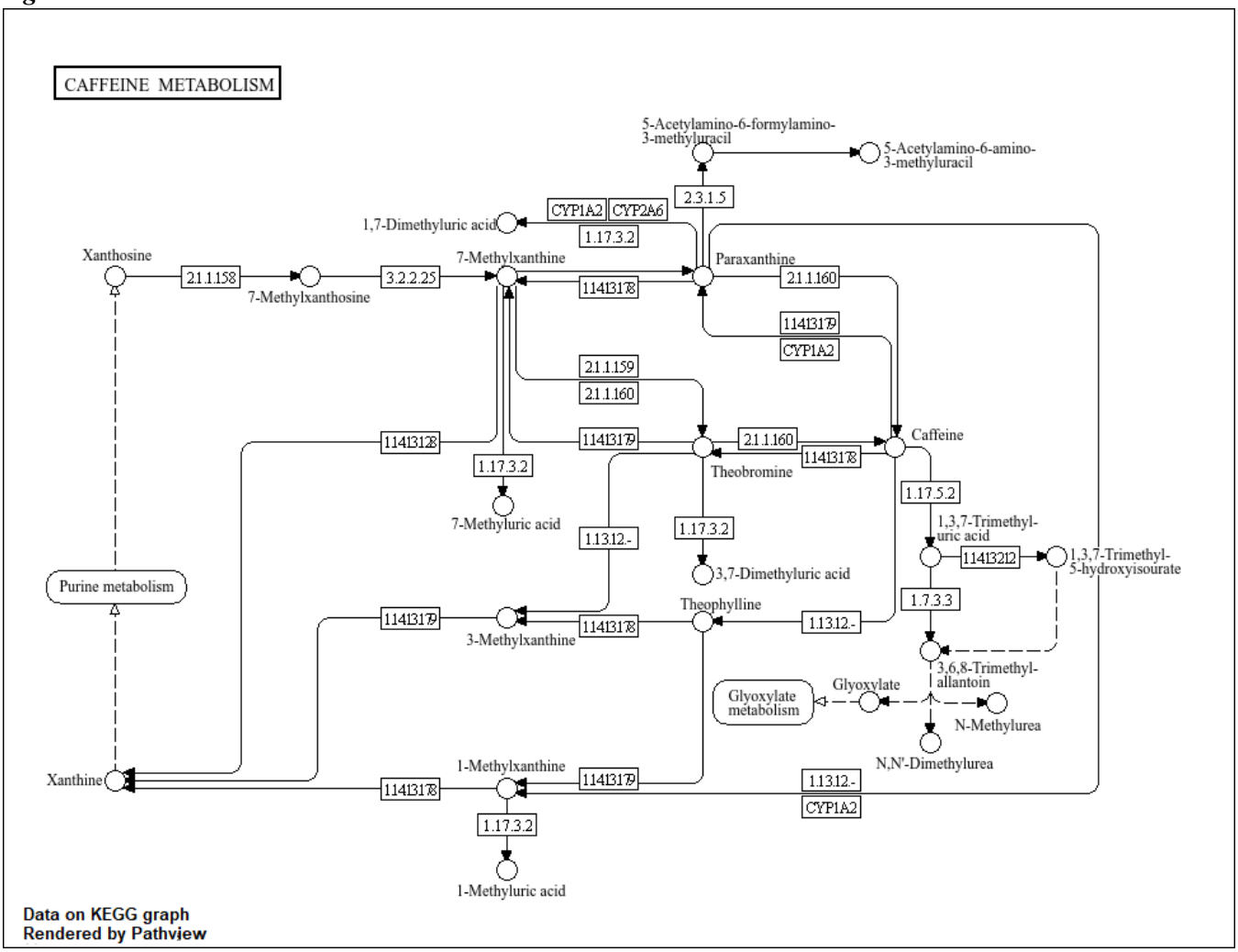


***Figure 40**: hsa00983 pathview*

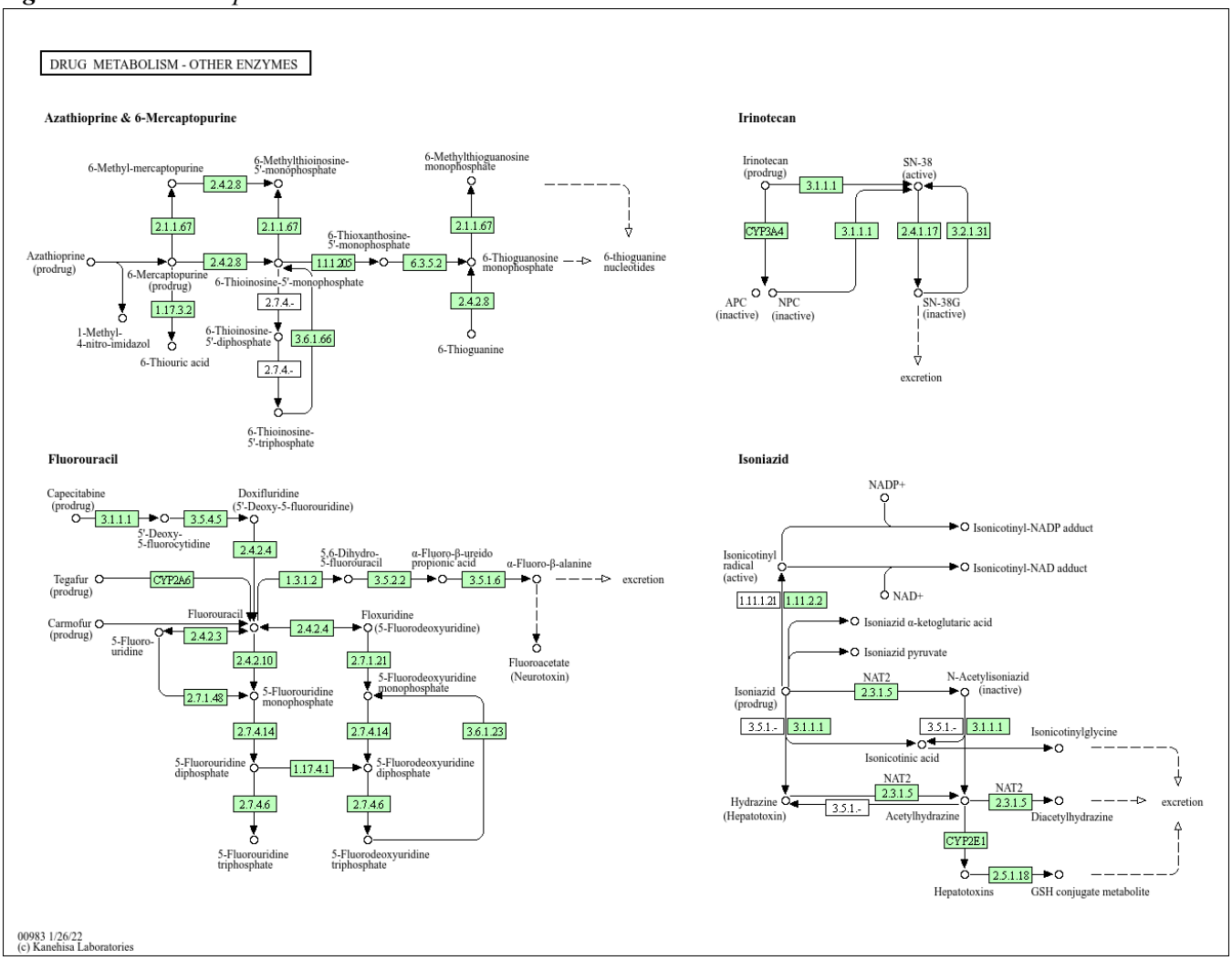


***Figure 41**: hsa00983 Cluster 1*

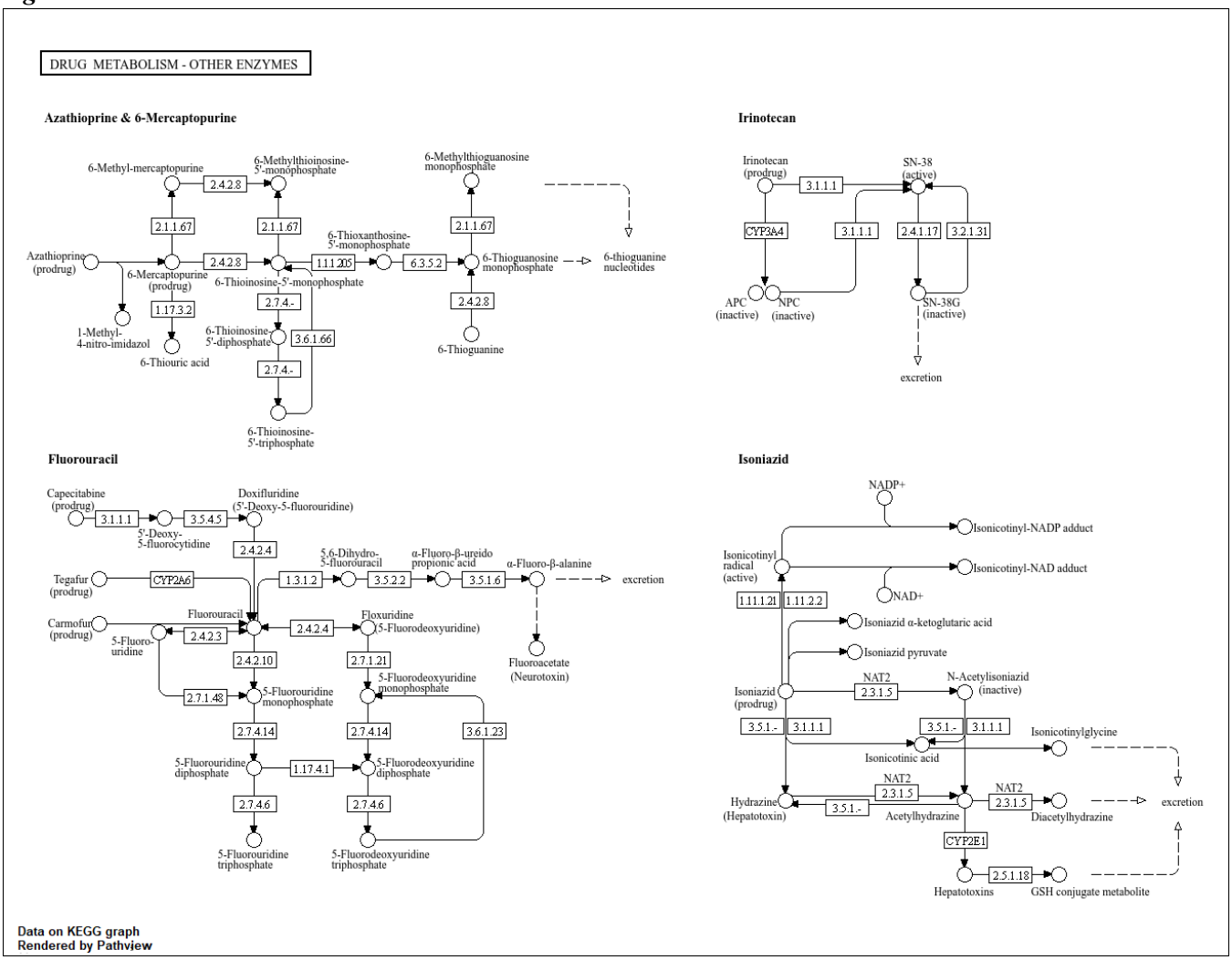


***Figure 42**: hsa00983 Cluster 2*

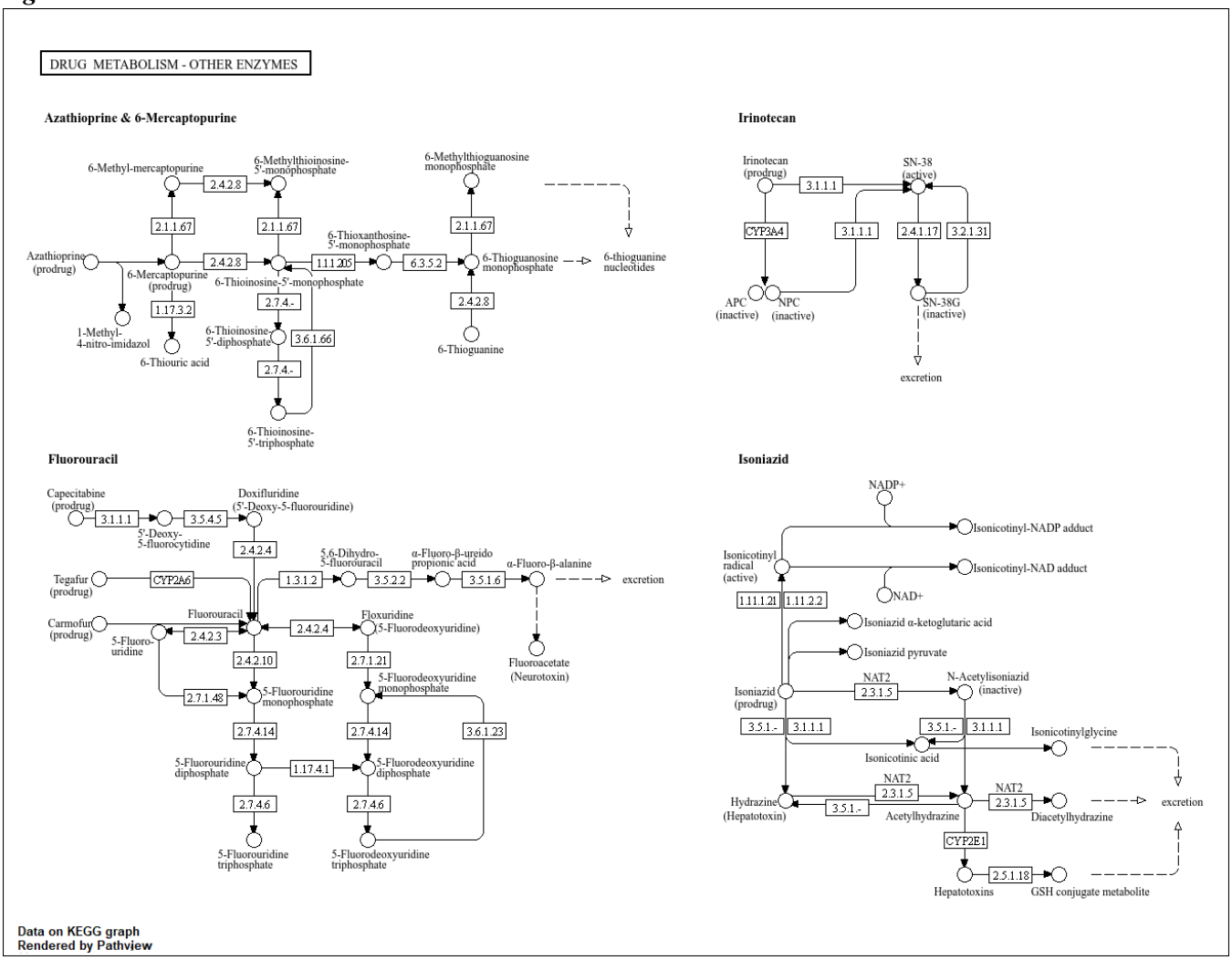


***Figure 43**: hsa04010 pathview*

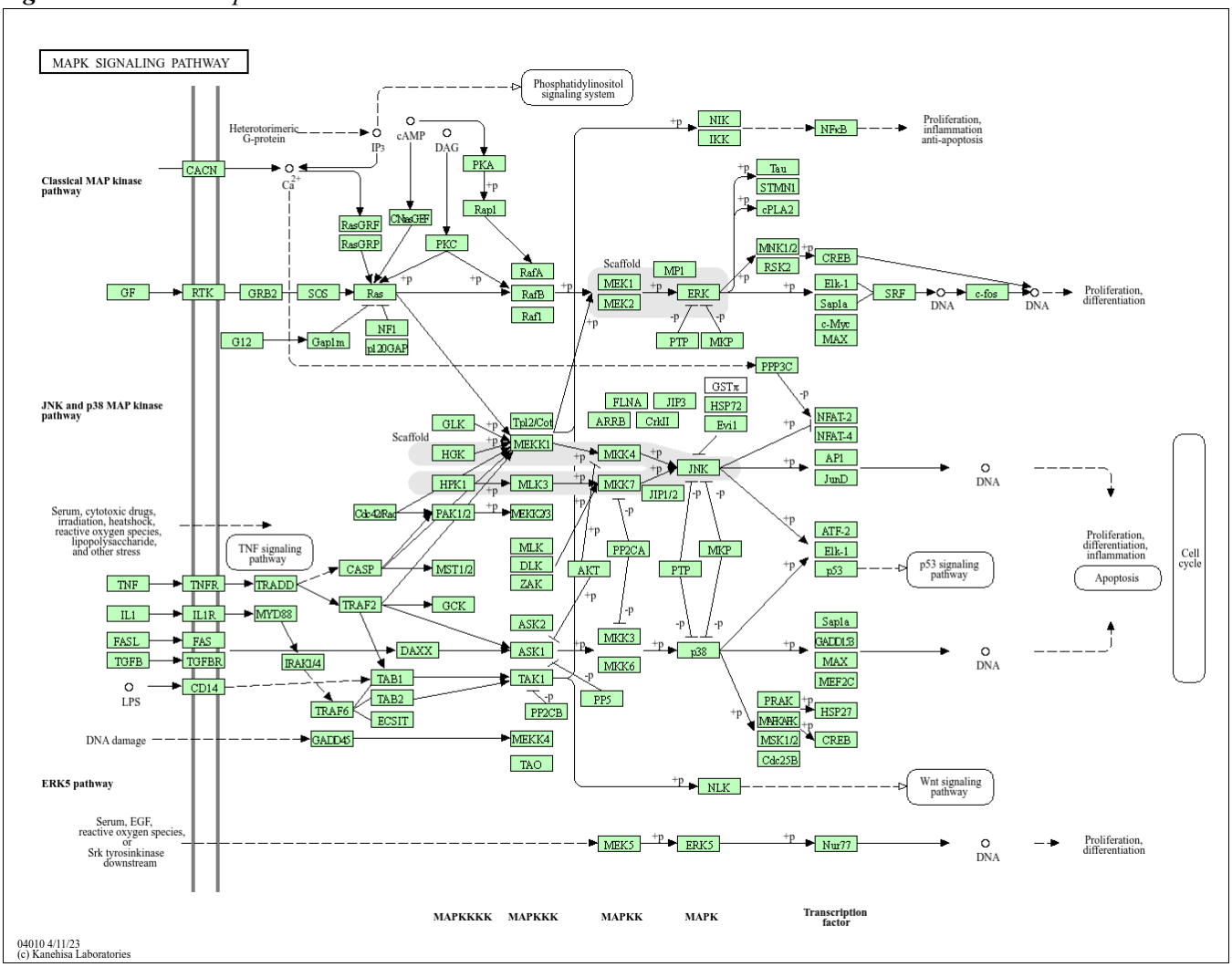


***Figure 44**: hsa04010 Cluster 1*

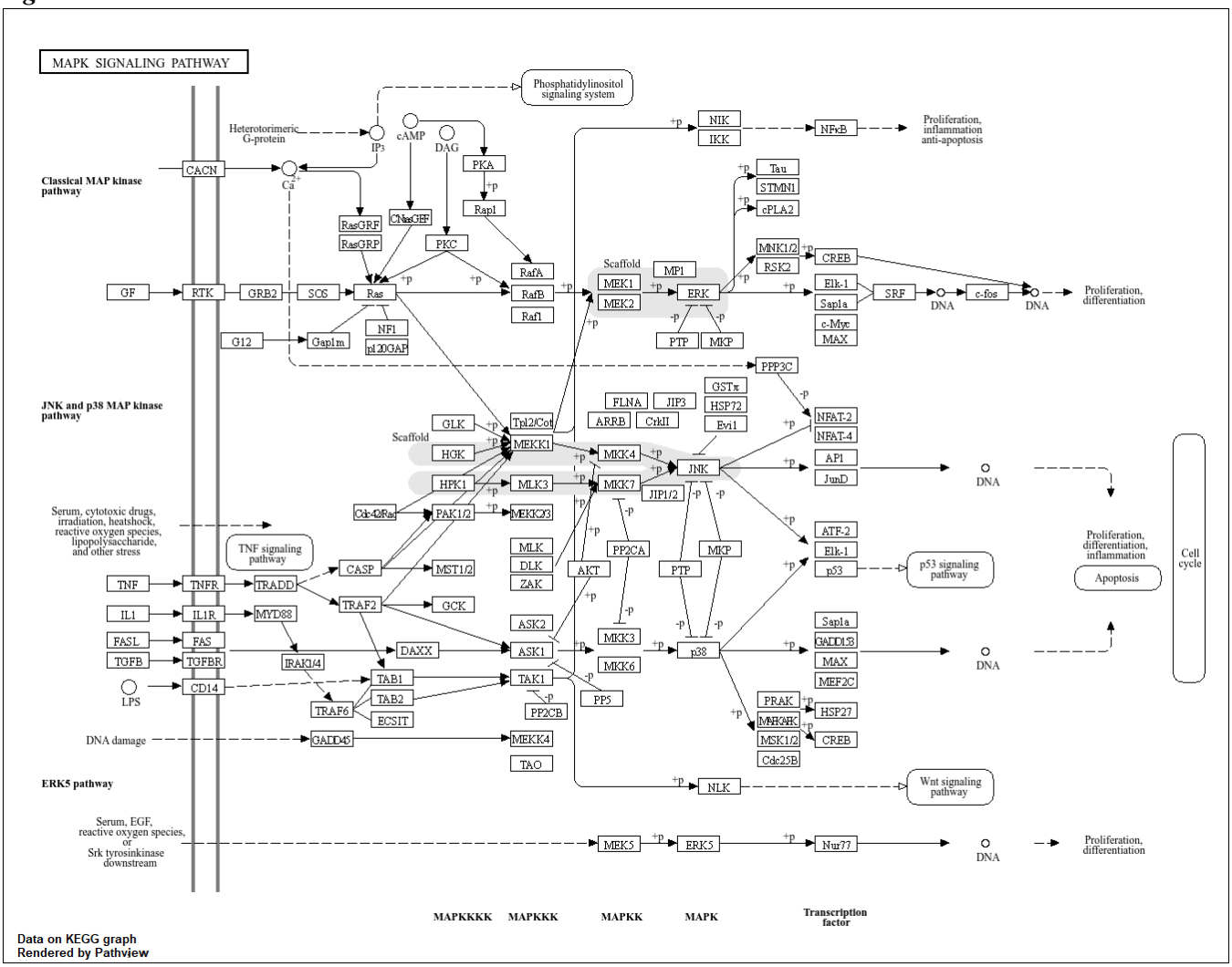


***Figure 45**: hsa04010 Cluster 2*

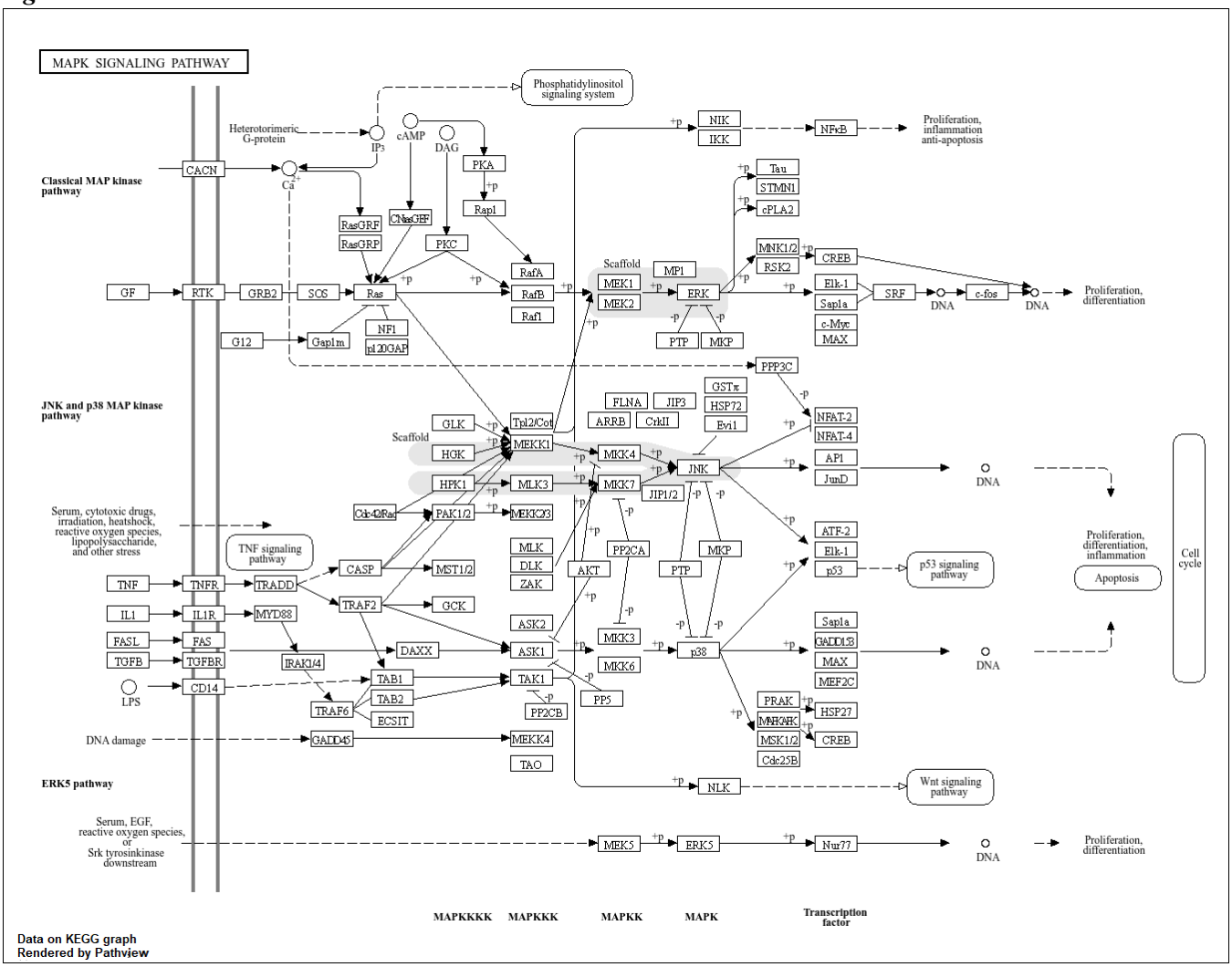


***Figure 46**: hsa04514 pathview*

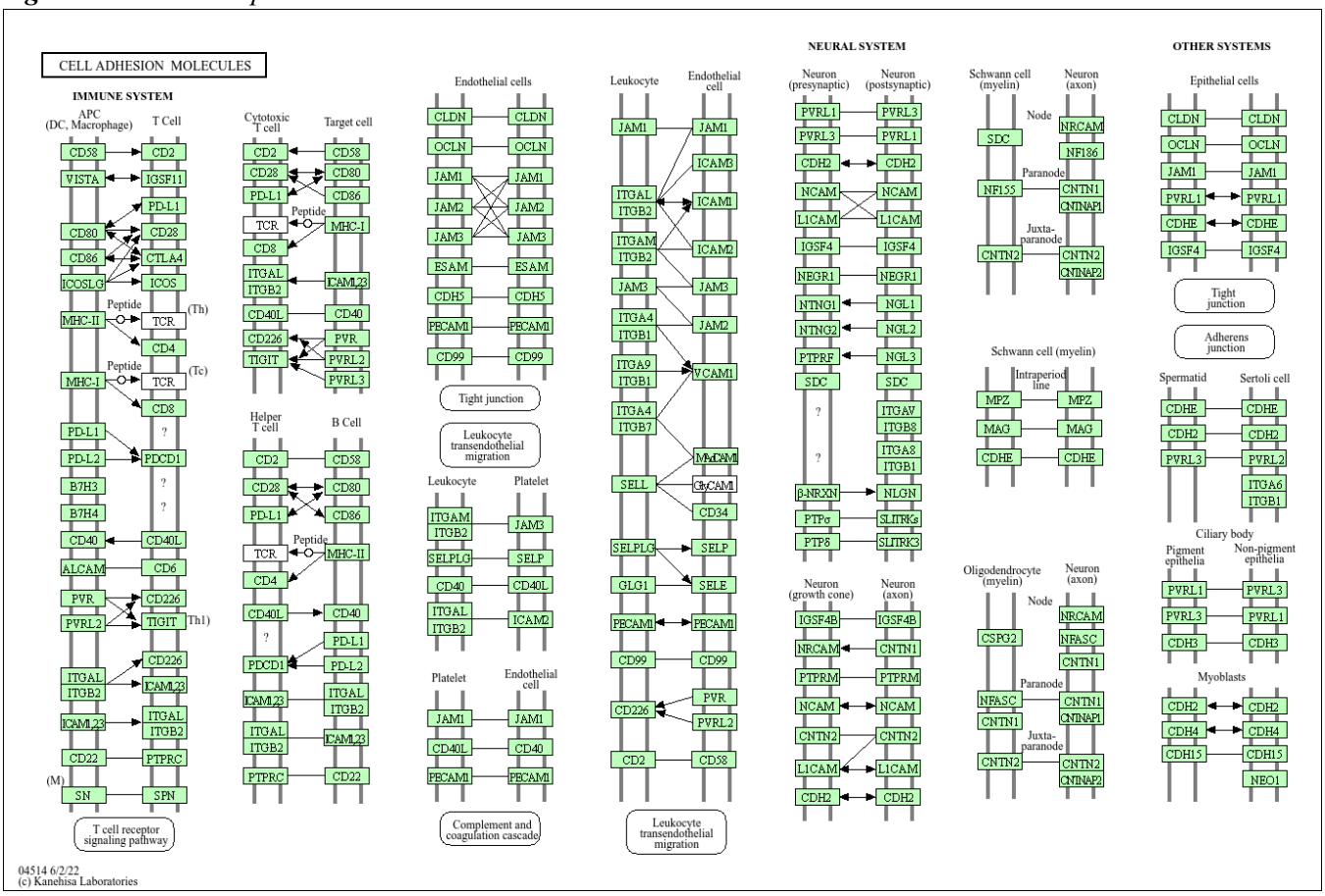


***Figure 47**: hsa04514 Cluster*

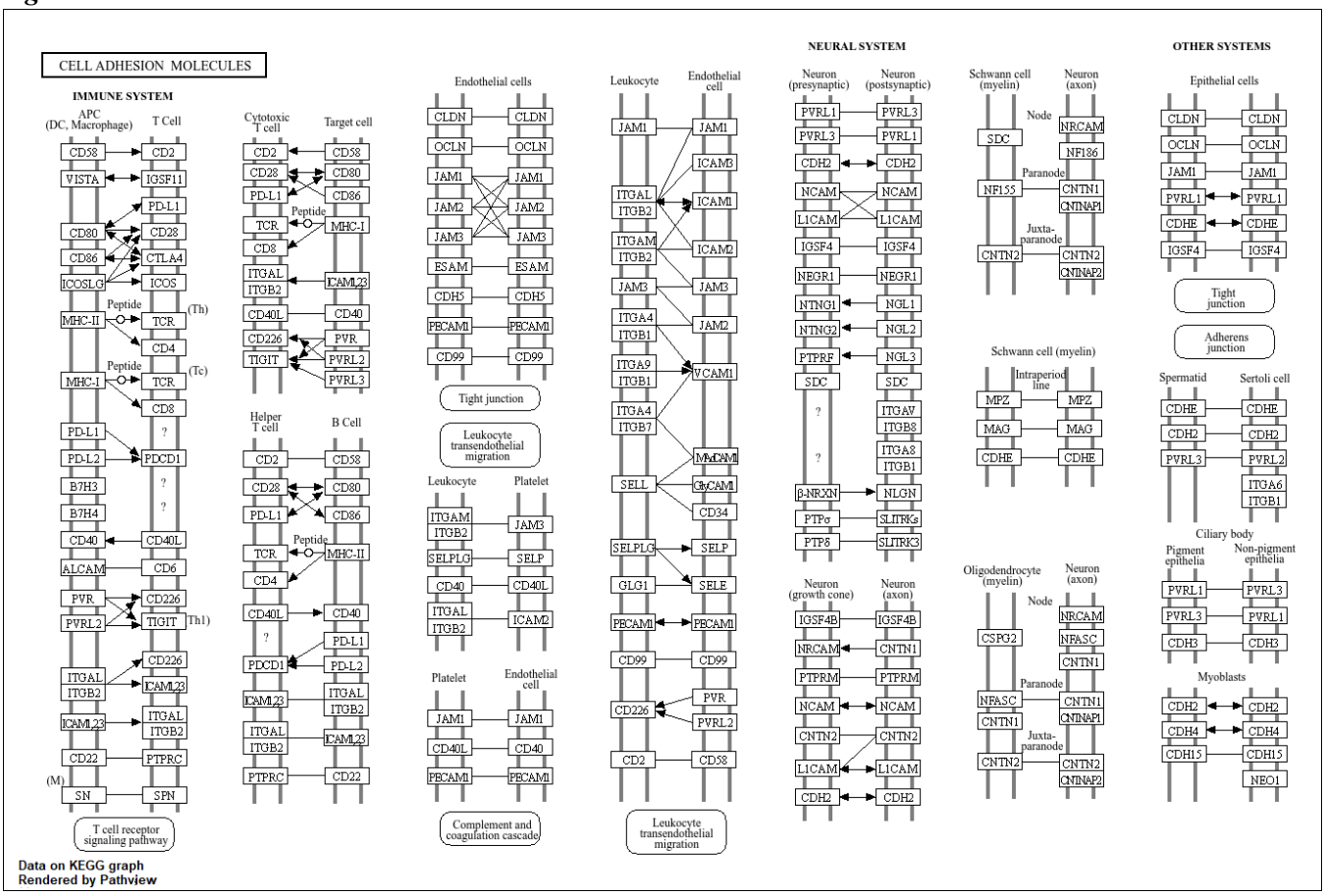
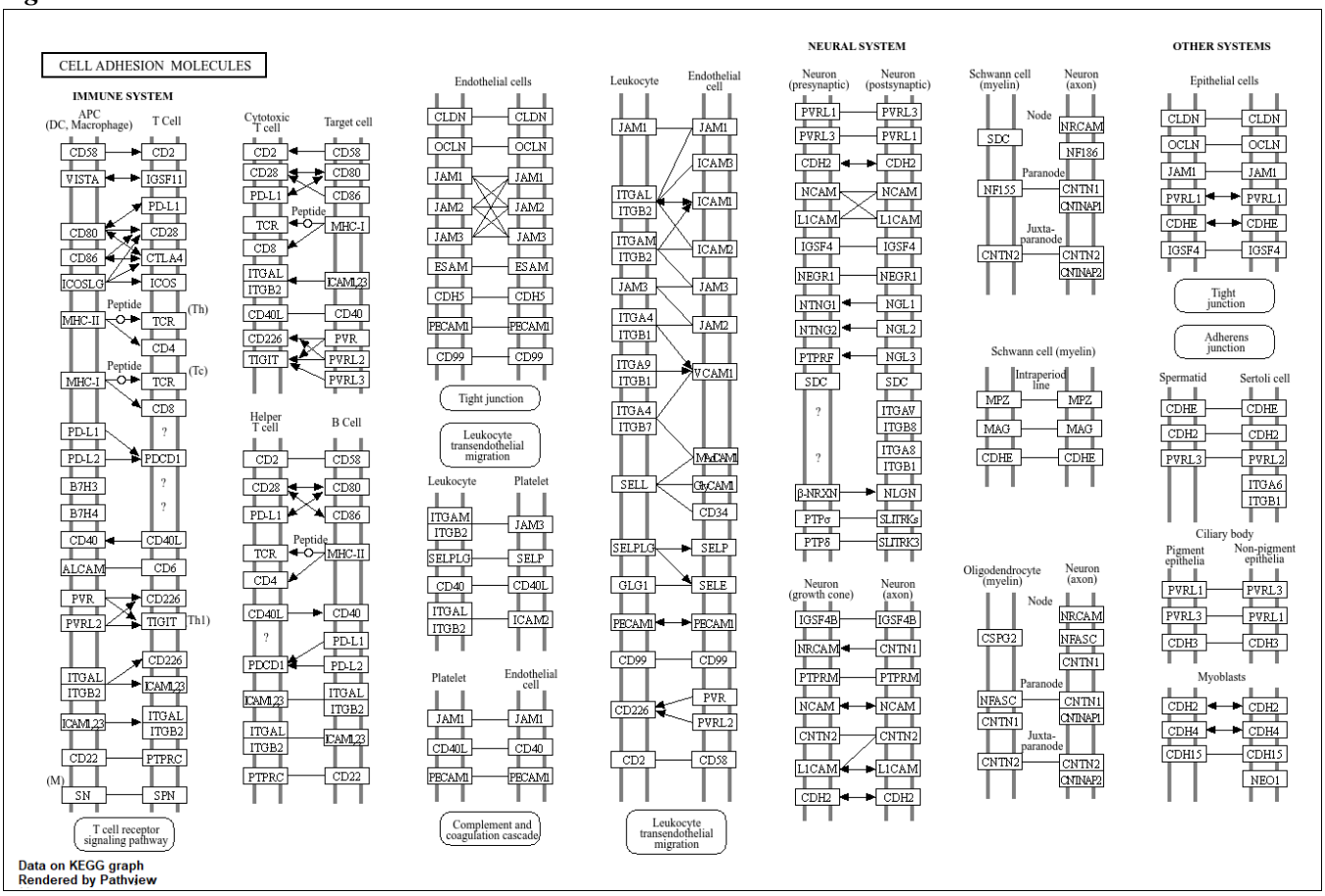


***Figure 48**: hsa04514 Cluster 2*



BMEG 310 Final Project

Jung Kyu Justin Yoon, Zeena Jarallah

2024-11-28

#Import libraries:

```
#all libraries from tutorials:

#Introduction to R Basics
library(graphics)
#install.packages("ggplot2")
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
#install.packages("RColorBrewer")
library(RColorBrewer)
#install.packages("rmarkdown")
library(rmarkdown)
```

```
## Warning: package 'rmarkdown' was built under R version 4.4.2
```

```
#Introduction to Machine Learning
#install.packages("devtools")
library(devtools)
```

```
## Loading required package: usethis
```

```
#install.packages("remotes")
#remotes::install_github("vqv/ggbiplot")
library(ggbiplot)
```

```
## Loading required package: plyr
```

```
## Loading required package: scales
```

```
## Loading required package: grid
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages("readxl")
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.2
```

```
#install.packages("ISLR")
require(ISLR)
```

```
## Loading required package: ISLR
```

```
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
#install.packages("caret")
library(caret)
```

```
## Loading required package: lattice
```

```
#install.packages("randomForest")
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.2

## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
#Introduction to Survival Analysis
#BiocManager::install("TCGAbiolinks")
library(TCGAbiolinks)
#BiocManager::install("survival")
library(survival)
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster
```

```r
#BiocManager::install("survminer")
library(survminer)
```

```
## Warning: package 'survminer' was built under R version 4.4.2

## Loading required package: ggpubr

##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':
##
##     mutate

##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

```r
library(SummarizedExperiment)
```

```
## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count

## The following object is masked from 'package:plyr':
##
##     count
```

```
##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following object is masked from 'package:randomForest':
##
##     combine

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, table,
##     tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:plyr':
##
##     rename

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:plyr':
##
##     desc

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
```

```
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
#Introduction to Mutation Analysis
# install.packages("pheatmap")
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.4.2
```

```
#Introduction to Differential Analysis
# BiocManager::install("AnnotationDbi")
# BiocManager::install("org.Hs.eg.db")
# BiocManager::install("pathview")
# BiocManager::install("gage")
# BiocManager::install("gageData")
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following object is masked from 'package:BiocGenerics':
##
##     combine

## The following object is masked from 'package:randomForest':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggrepel)
library(AnnotationDbi)
```

```
##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(org.Hs.eg.db)
```

```
##
```

```
library(pathview)
```

```
## ##############################################################################
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications
## or products. For details, do citation("pathview") within R.
##
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## ##############################################################################
```

```
#install.packages("gage")
library(gage)
```

```
##
```

```
#install.packages("gageData")
library(gageData)


#Additional useful libraries:
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
library(readr)
```

```
##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:SummarizedExperiment':
##
##     shift
```

```
## The following object is masked from 'package:GenomicRanges':
##
##     shift

## The following object is masked from 'package:IRanges':
##
##     shift

## The following objects are masked from 'package:S4Vectors':
##
##     first, second

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(stats)
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(org.Hs.eg.db)
library(enrichplot)
```

```
##
## Attaching package: 'enrichplot'

## The following object is masked from 'package:ggpubr':
##
##     color_palette

## The following object is masked from 'package:lattice':
##
##     dotplot
```

```
library(DESeq2)
#install.packages(magrittr)
library (dplyr)
library(TCGAbiolinks)
#BiocManager::install("edgeR")
library(edgeR)
```

```
## Loading required package: limma

##
## Attaching package: 'limma'

## The following object is masked from 'package:DESeq2':
##
##     plotMA
```

```
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
```

```
#BiocManager::install("ComplexHeatmap")
library(ComplexHeatmap)
```

```
## ========================================
## ComplexHeatmap version 2.20.0
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##     genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
## ! pheatmap() has been masked by ComplexHeatmap::pheatmap(). Most of the arguments
##    in the original pheatmap() are identically supported in the new function. You
##    can still use the original function by explicitly calling pheatmap::pheatmap().

##
## Attaching package: 'ComplexHeatmap'

## The following object is masked from 'package:pheatmap':
##
##     pheatmap
```

```
library(circlize)
```

```
## Warning: package 'circlize' was built under R version 4.4.2

## ========================================
## circlize version 0.4.16
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```r
#install.packages("reshape2")
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.4.2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

#Import data:

```r
RNA.data <- read.csv("RNAseq_LIHC.csv")
patient.data <- read.table("data_clinical_patient.txt", header = TRUE, sep = "\t")
mutations.data <- read.table("data_mutations.txt", header = TRUE, sep = "\t")
```

##summaries of data

```r
#summary(RNA.data)
#summary(patient.data)
#summary(mutations.data)
#colnames(RNA.data)
#colnames(patient.data)
#colnames(mutations.data)
```

#filter data

```r
patient_df<-patient.data
patient_df$PATIENT_ID<-substr(patient_df$PATIENT_ID,9,12)

mutations_df<-mutations.data
mutations_df$Tumor_Sample_Barcode<-substr(mutations_df$Tumor_Sample_Barcode,9,12)

#make copy of RNA data frame without the first column
RNA_df <- RNA.data[,-1]

#rename columns to match patient id naming convention of other data sets
colnames(RNA_df)<-substr(colnames(RNA_df),9,12)

#find unique patients
unique_patients<-unique(patient_df$PATIENT_ID)
unique_mutations<-unique(mutations_df$Tumor_Sample_Barcode)
unique_RNA<-unique(colnames(RNA_df))
#find common patients
common_patients<-Reduce(intersect, list(unique_patients,unique_mutations,unique_RNA))
```

```
#patient data with patients in all three data sets
unq_patient_df<-patient_df[patient_df$PATIENT_ID %in% common_patients,]

#mutation data with patients in all three data sets
unq_mutations_df<-mutations_df[mutations_df$Tumor_Sample_Barcode %in% common_patients,]

#RNA data with patients in all three data sets
unq_RNA_ID<-RNA_df[, colnames(RNA_df)%in% common_patients]

#add back the first column (X) from the RNA data frame
RNA_df$X<-RNA.data$X[match(rownames(RNA_df), rownames(RNA.data))]


#Take out Low impact mutations, intron and synonymous variants
filtered_mutations <- subset(unq_mutations_df,
                             ((unq_mutations_df$IMPACT != "LOW") & (unq_mutations_df$Consequence != "in
                              & (unq_mutations_df$Consequence != "synonymous_variant")))
```

##Summaries of filtered data

```
#summary(unq_patient_df)
#summary(filtered_mutations)
#summary(RNA_df)
```

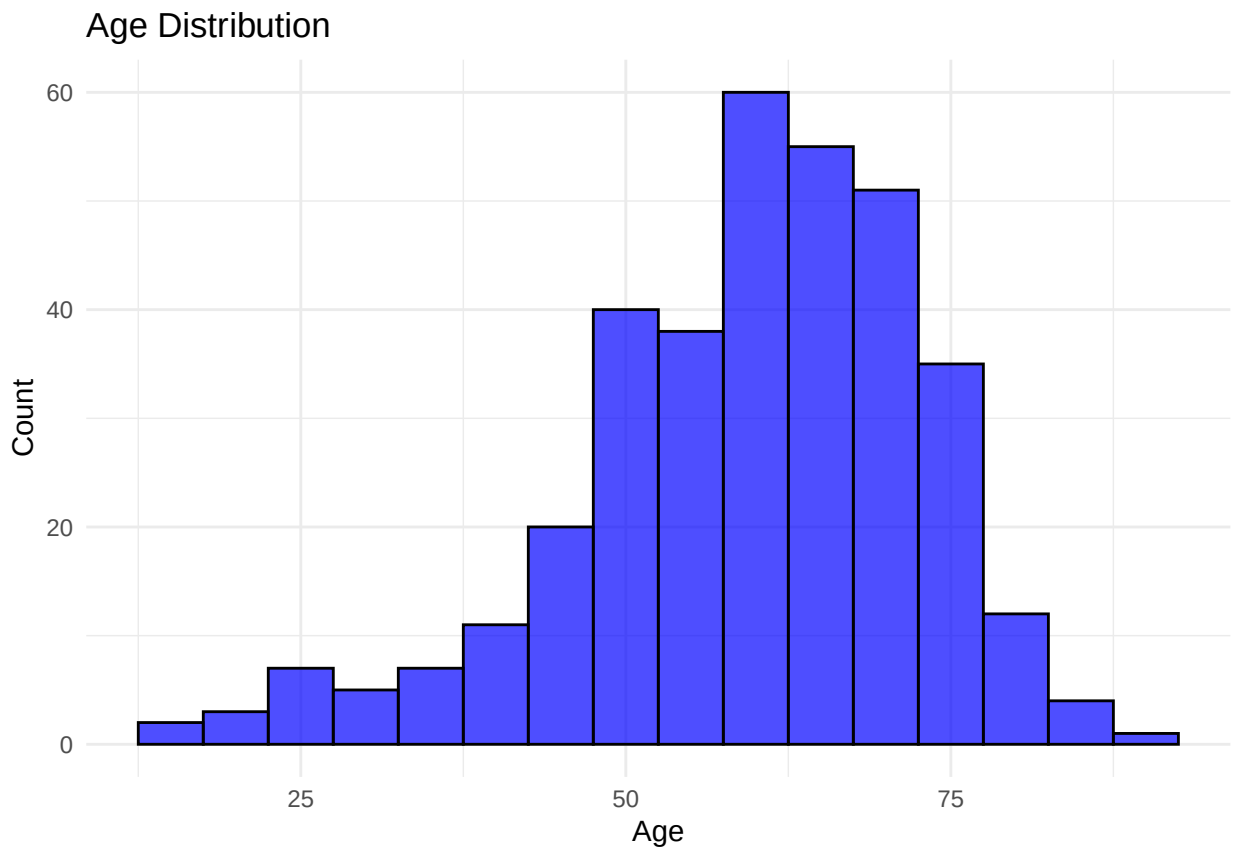
#Clinical Data Analysis:

##Clinical Data: ##1. Explore clinical data wrt the distribution based on various variable/attributes such as age, stage, survival, etc

###Age Distribution

```
ggplot(unq_patient_df, aes(x = AGE)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Age Distribution", x = "Age", y = "Count") +
  theme_minimal()
```

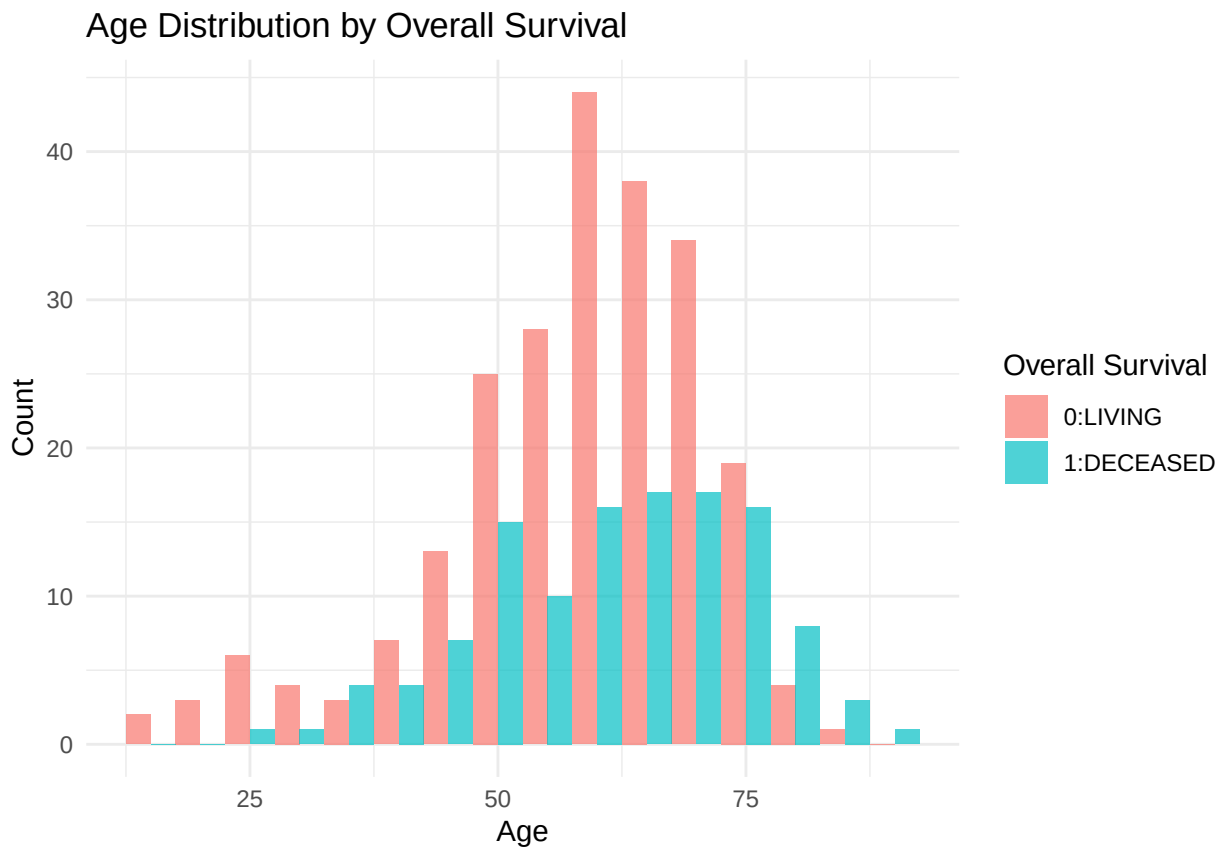
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```



###Age vs Overall Survival

```
ggplot(unq_patient_df, aes(x = AGE, fill = OS_STATUS)) +
  geom_histogram(binwidth = 5, position = "dodge", alpha = 0.7) +
  labs(title = "Age Distribution by Overall Survival", x = "Age", y = "Count", fill = "Overall Survival
  theme_minimal()
```

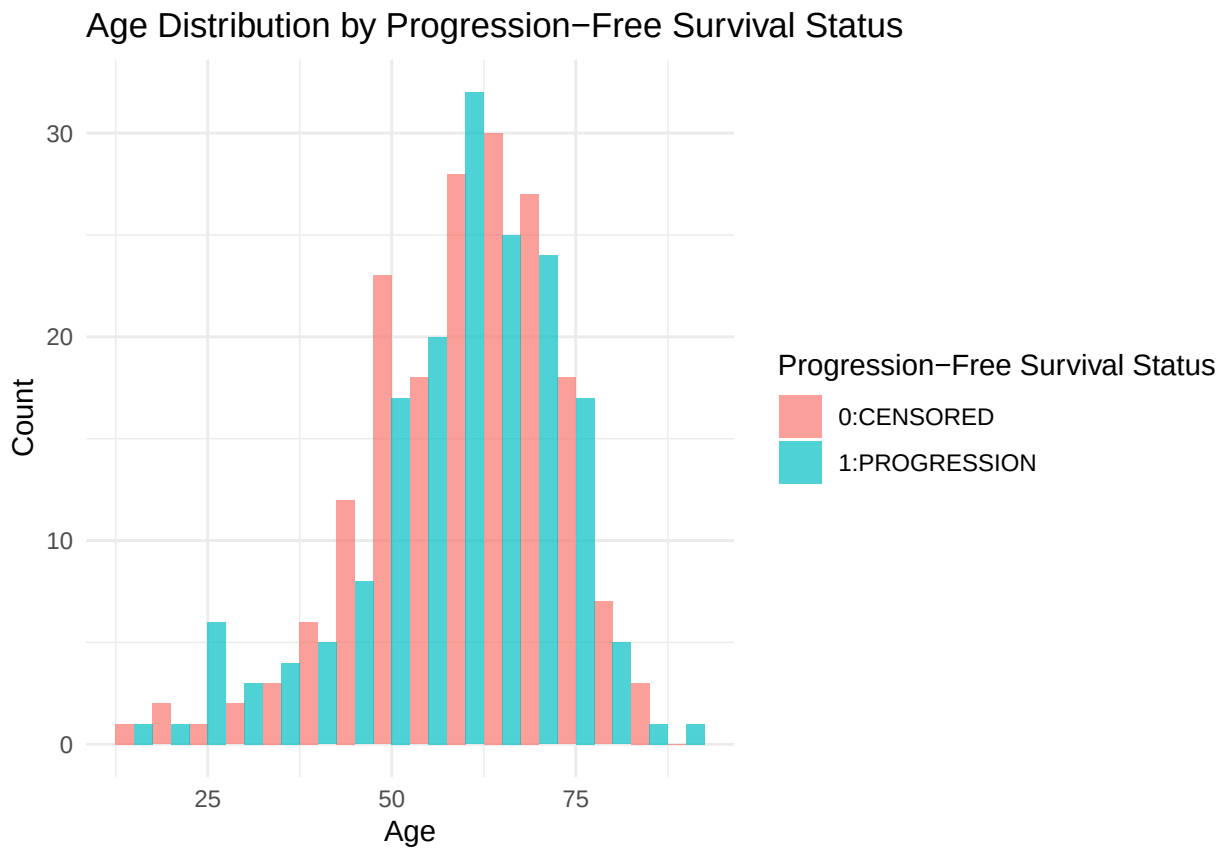
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```



###Age vs Progression-Free Survival (PFS) Status

```
ggplot(unq_patient_df, aes(x = AGE, fill = PFS_STATUS)) +
  geom_histogram(binwidth = 5, position = "dodge", alpha = 0.7) +
  labs(title = "Age Distribution by Progression-Free Survival Status", x = "Age", y = "Count", fill = "
  theme_minimal()
```

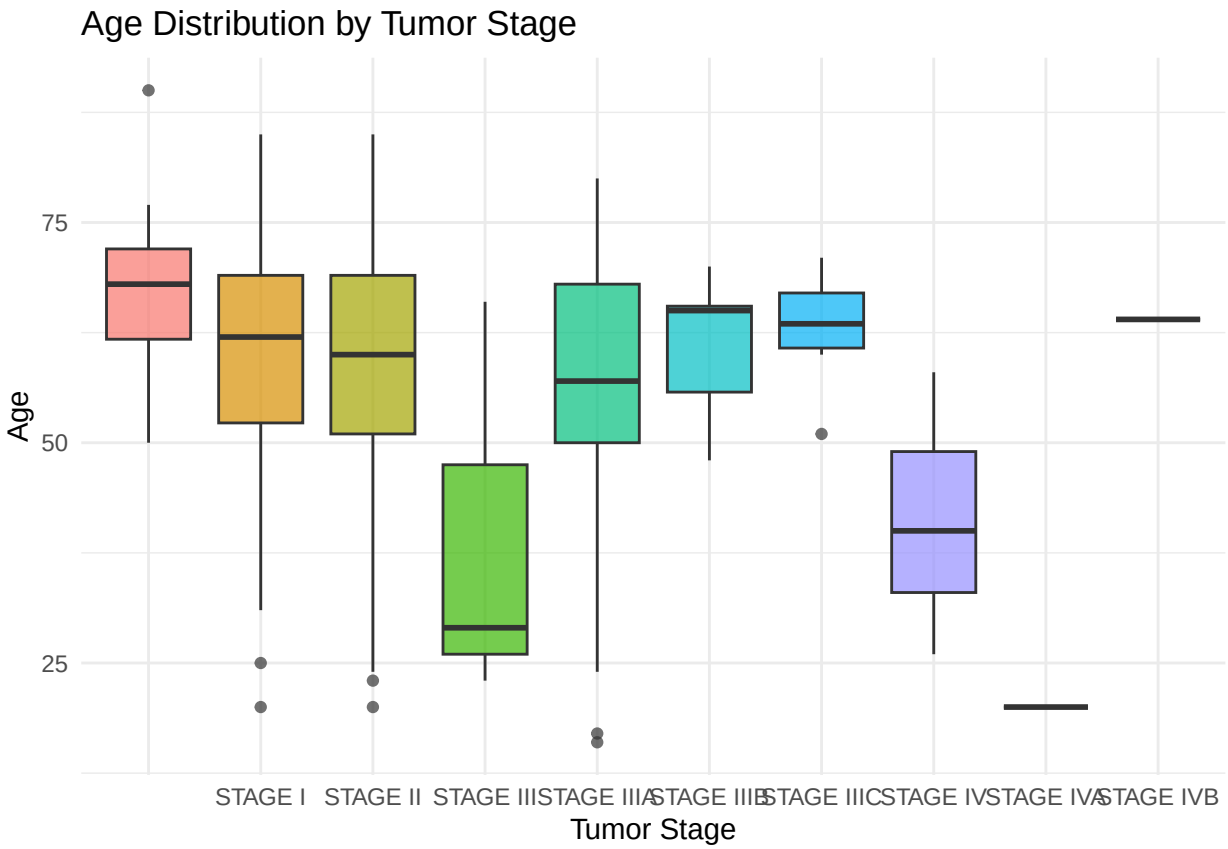
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```



###Age Distribution vs Tumor Stagewith (Boxplot)

```
ggplot(unq_patient_df, aes(x = AJCC_PATHOLOGIC_TUMOR_STAGE, y = AGE, fill = AJCC_PATHOLOGIC_TUMOR_STAGE
  geom_boxplot(alpha = 0.7) +
  labs(title = "Age Distribution by Tumor Stage", x = "Tumor Stage", y = "Age") +
  theme_minimal() +
  theme(legend.position = "none")
```

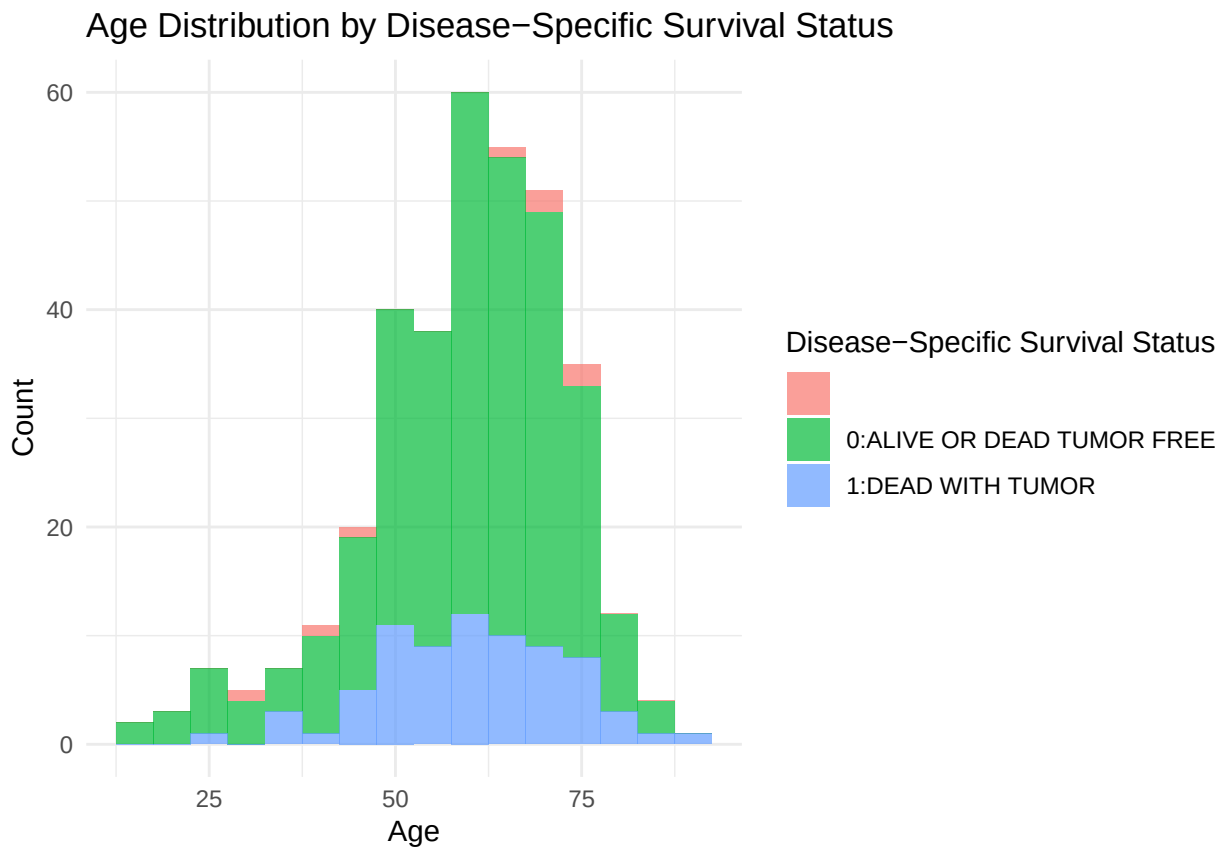
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



###Age Distribution vs Disease-Specific Survival Status

```
ggplot(unq_patient_df, aes(x = AGE, fill = DSS_STATUS)) +
  geom_histogram(binwidth = 5, position = "stack", alpha = 0.7) +
  labs(title = "Age Distribution by Disease-Specific Survival Status", x = "Age", y = "Count", fill = "
  theme_minimal()
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```



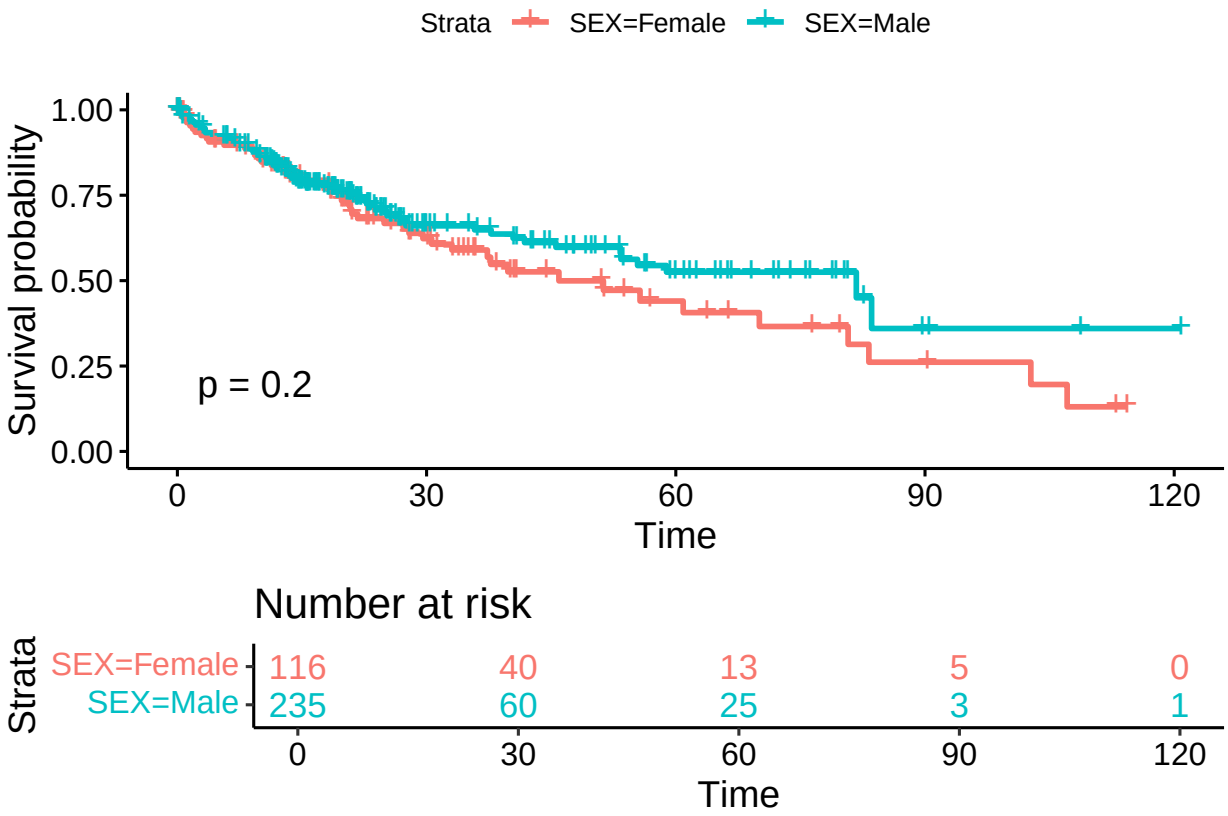
#Survival analysis

##Prepare Data

```
#survival data
unq_patient_df <- unq_patient_df %>%
  mutate(
    #convert OS_STATUS to a logical deceased indicator
    deceased = ifelse(OS_STATUS == "1:DECEASED", TRUE, FALSE),
    #convert OS_MONTHS to numeric
    overall_survival = as.numeric(OS_MONTHS)
  )
```

##Kaplan-Meier Analysis by Gender

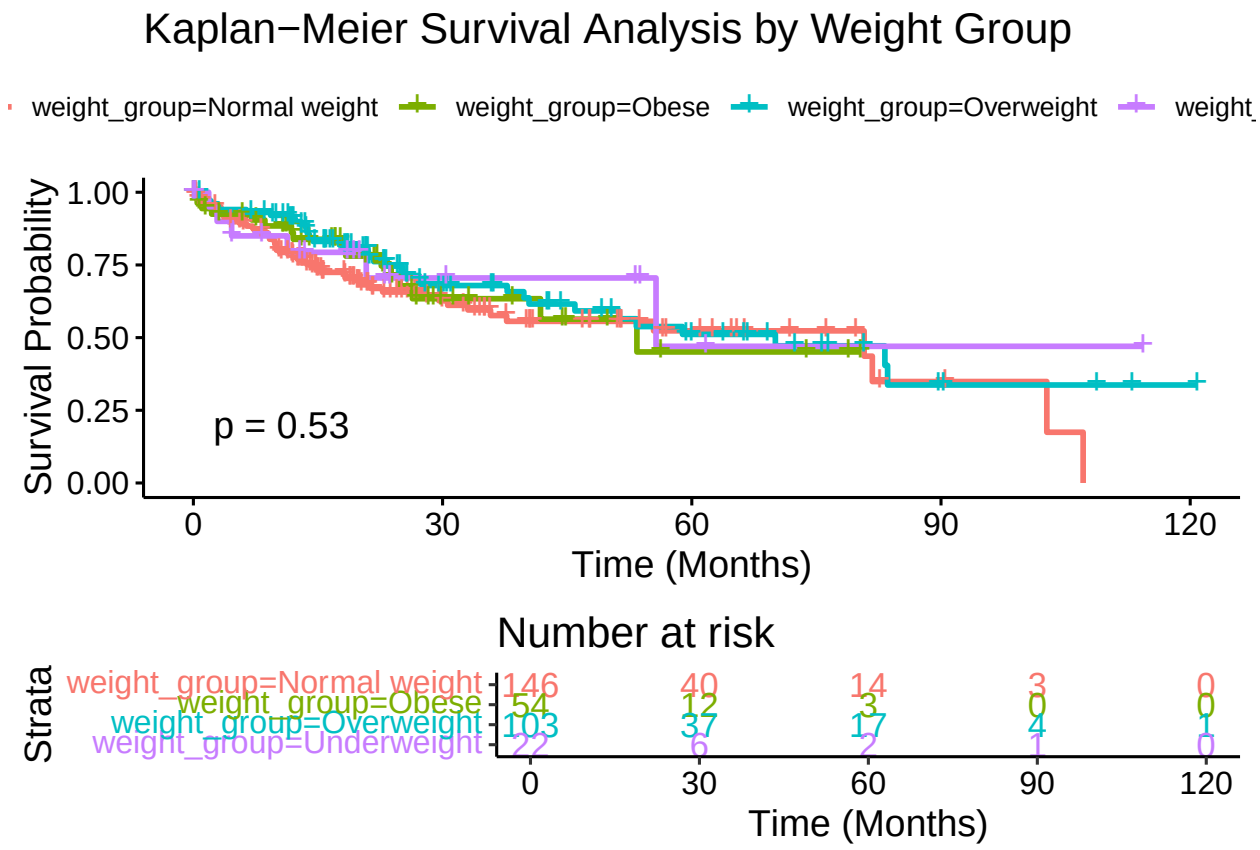
```
#fit survival model for gender
fit_gender <- survfit(Surv(overall_survival, deceased) ~ SEX, data = unq_patient_df)

#Kaplan-Meier plot for gender
ggsurvplot(fit_gender, data = unq_patient_df, pval = TRUE, risk.table = TRUE, risk.table.col = "strata"
```



##Kaplan-Meier Analysis by Weight

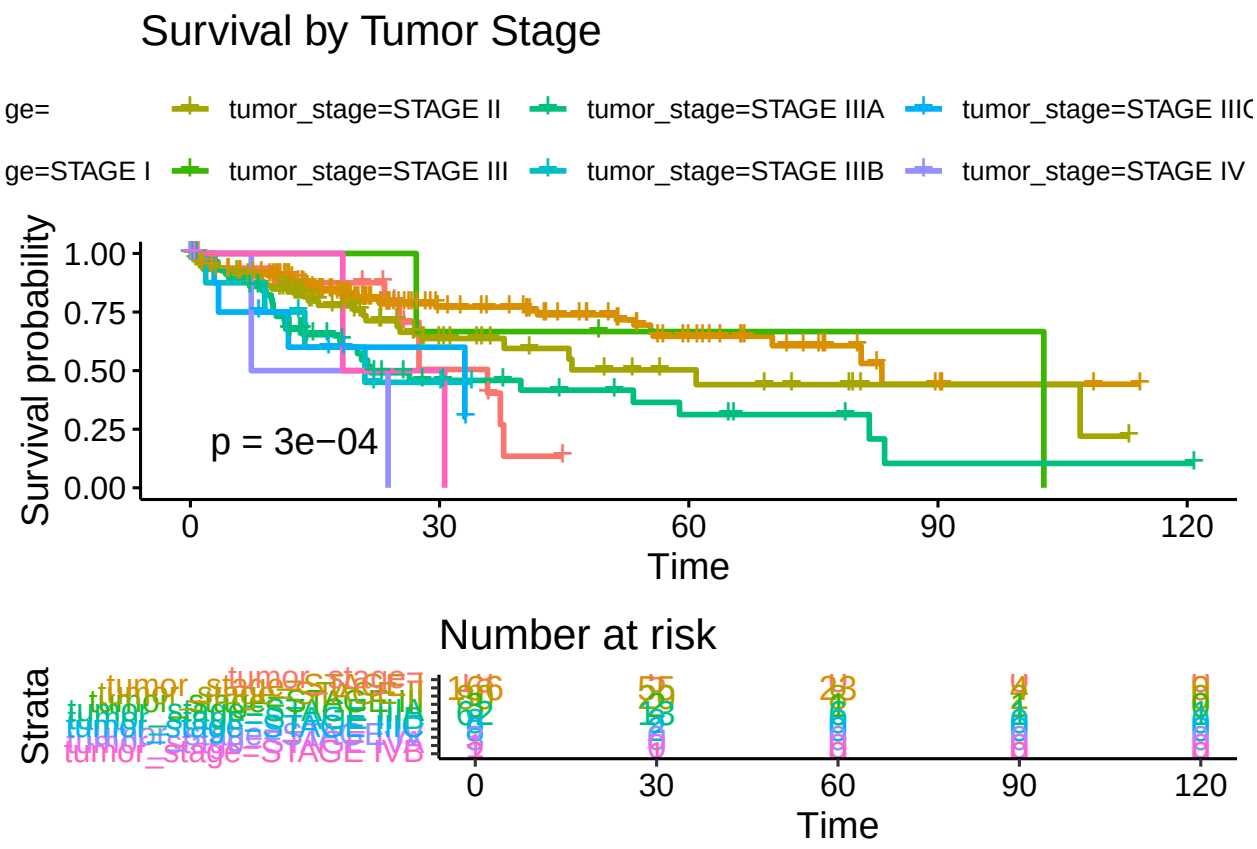
```
# categorize patients by weight
unq_patient_df <- unq_patient_df %>%
  mutate(weight_group = case_when(
    WEIGHT < 50 ~ "Underweight",
    WEIGHT >= 50 & WEIGHT < 70 ~ "Normal weight",
    WEIGHT >= 70 & WEIGHT < 90 ~ "Overweight",
    WEIGHT >= 90 ~ "Obese",
    is.na(WEIGHT) ~ "Unknown"
  ))

#exclude patients with unknown weight
unq_patient_df_filtered <- unq_patient_df %>%
  filter(weight_group != "Unknown")

#fit survival model for weight groups
fit_weight <- survfit(Surv(overall_survival, deceased) ~ weight_group, data = unq_patient_df_filtered)

#Kaplan-Meier for weight groups
ggsurvplot(fit_weight,
           data = unq_patient_df_filtered,
           pval = TRUE,                        # Display p-value for log-rank test
           risk.table = TRUE,                  # Add risk table below the plot
           risk.table.col = "strata",          # Color risk table by strata
           risk.table.height = 0.3,            # Adjust risk table height
```

```
          title = "Kaplan-Meier Survival Analysis by Weight Group",
          xlab = "Time (Months)",
          ylab = "Survival Probability")
```



##Kaplan-Meier Analysis by Tumor Stage

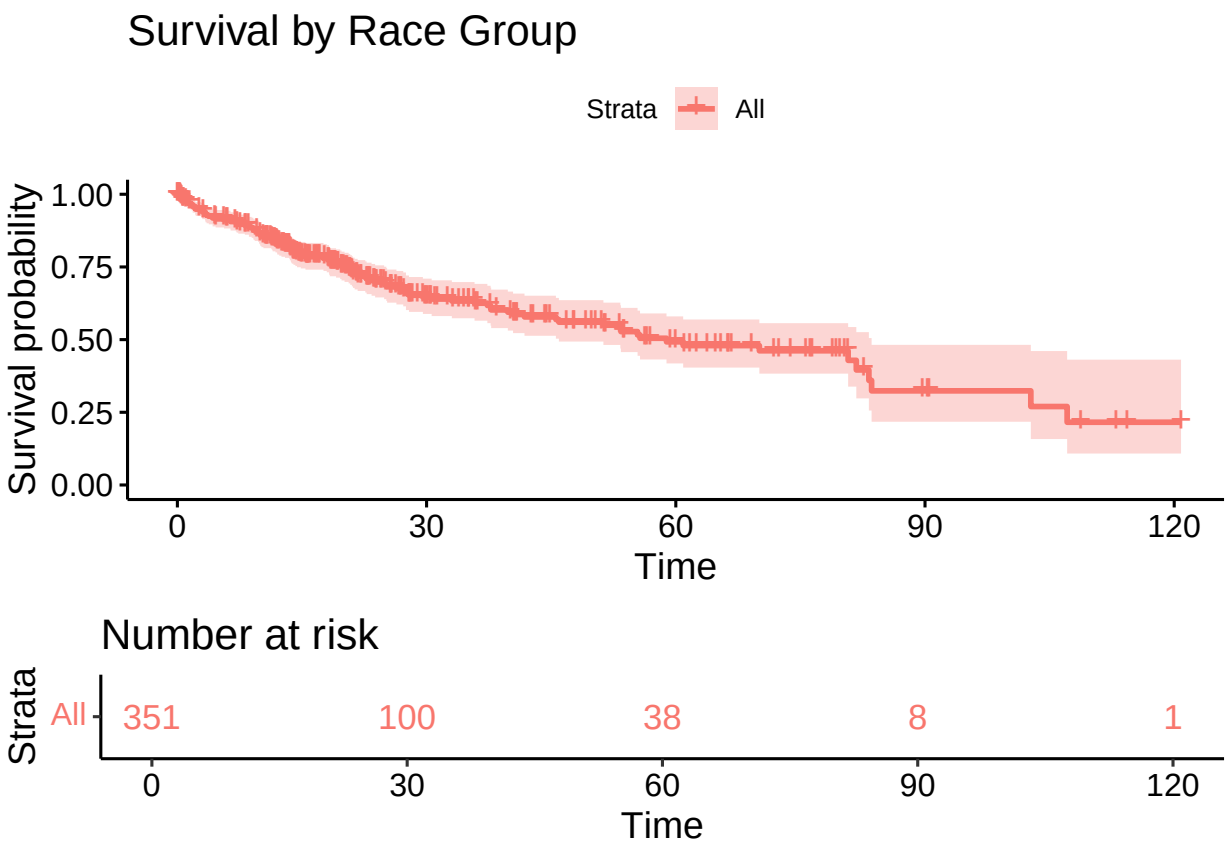
```
# remove sub-stages of AJCC_PATHOLOGIC_TUMOR_STAGE
unq_patient_df <- unq_patient_df %>%
  mutate(tumor_stage = gsub("[abc]$", "", AJCC_PATHOLOGIC_TUMOR_STAGE))

#fit survival model by tumor stage
fit_stage <- survfit(Surv(overall_survival, deceased) ~ tumor_stage, data = unq_patient_df)

#Kaplan-Meier for tumor stage
ggsurvplot(fit_stage,
           data = unq_patient_df,
           pval = TRUE,
           risk.table = TRUE,
           risk.table.col = "strata",
           risk.table.height = 0.3,
           title = "Survival by Tumor Stage")
```



##Kaplan-Meier Analysis by race

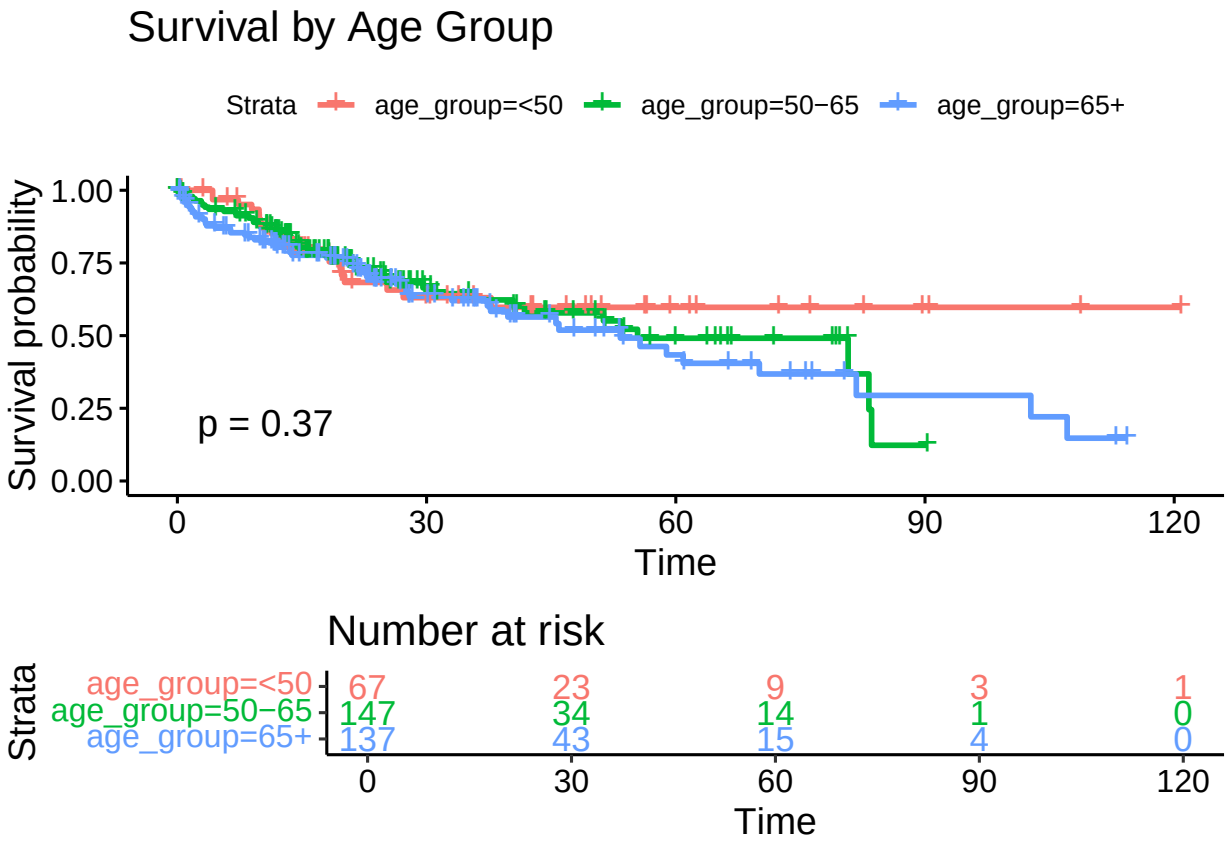
```
#Group races
  #"white", "asian", "black", and "others"...?
unq_patient_df <- unq_patient_df %>%
  mutate(race_group = ifelse(RACE %in% c("white", "asian", "black or african american"), RACE, "others")

#fit survival model by race group
fit_race <- survfit(Surv(overall_survival, deceased) ~ race_group, data = unq_patient_df)

#Kaplan-Meier plot for race
ggsurvplot(fit_race,
           data = unq_patient_df,
           pval = TRUE,
           risk.table = TRUE,
           risk.table.col = "strata",
           risk.table.height = 0.3,
           title = "Survival by Race Group")
```

```
## Warning in .pvalue(fit, data = data, method = method, pval = pval, pval.coord = pval.coord, : There a
##  This is a null model.
```



##Kaplan-Meier Analysis by Age Group

```
# age groups
unq_patient_df <- unq_patient_df %>%
  mutate(age_group = case_when(
    AGE < 50 ~ "<50",
    AGE >= 50 & AGE < 65 ~ "50-65",
    AGE >= 65 ~ "65+"
  ))

#fit survival model by age group
fit_age <- survfit(Surv(overall_survival, deceased) ~ age_group, data = unq_patient_df)

#Kaplan-Meier
ggsurvplot(fit_age,
           data = unq_patient_df,
           pval = TRUE,
           risk.table = TRUE,
           risk.table.col = "strata",
           risk.table.height = 0.3,
           title = "Survival by Age Group")
```



#Create Clusters Using Expression Data

Pre-Processing

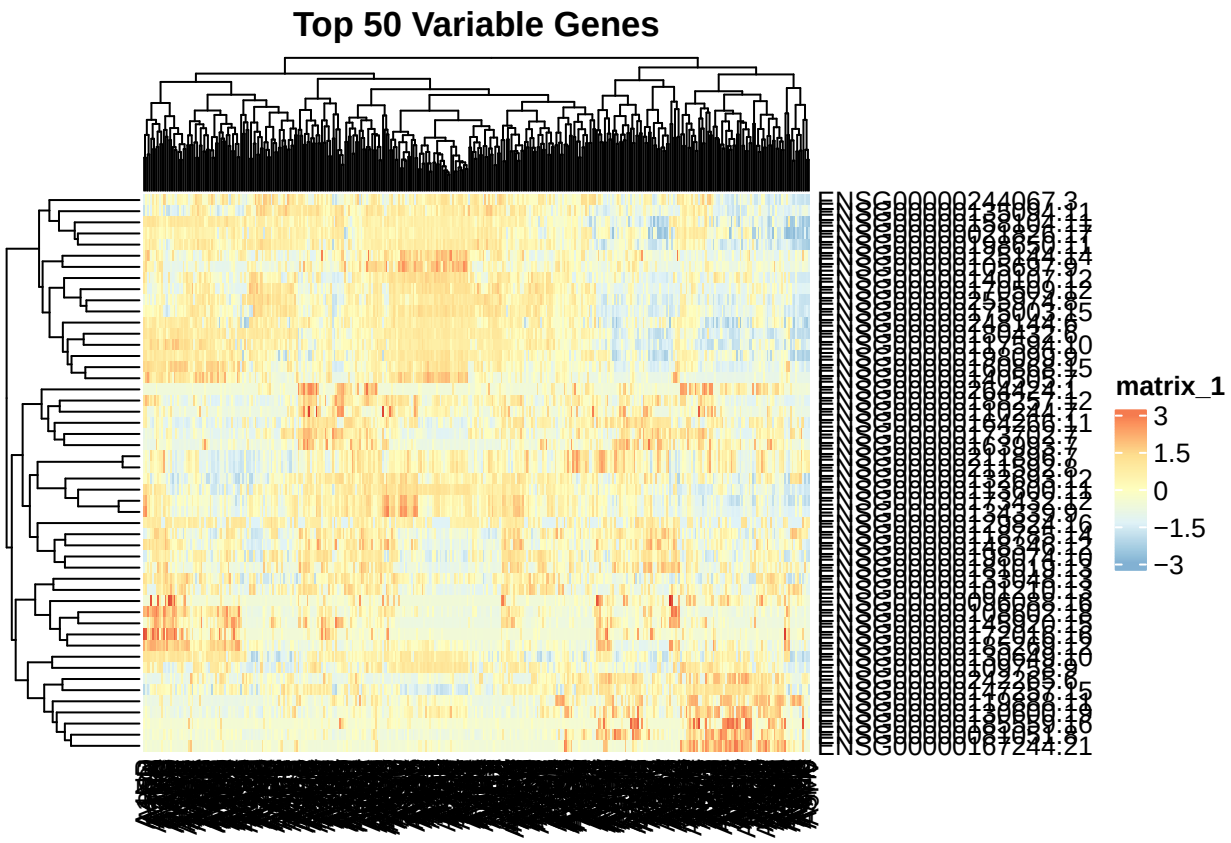
```
#Move Column "X" to the first position

RNA_df <- RNA_df[, c("X", setdiff(names(RNA_df), "X"))]

#remove column "X", as it is not a sample
preprocessed_RNA = RNA_df[,-1]

#convert to matrix
preprocessed_RNA = as.matrix(preprocessed_RNA)

gene_names <- RNA_df$X  # Extract gene names
rownames(preprocessed_RNA) <- gene_names  # Set as row names

#filter data that has 0 or 1 read count across all samples

preprocessed_RNA <- preprocessed_RNA[rowSums(preprocessed_RNA)>1,]
```

```
# CPM normalize data
cpm_RNA <- cpm(preprocessed_RNA)
```

```
#log normalize the counts
log_RNA <- log2(cpm_RNA + 1)
```

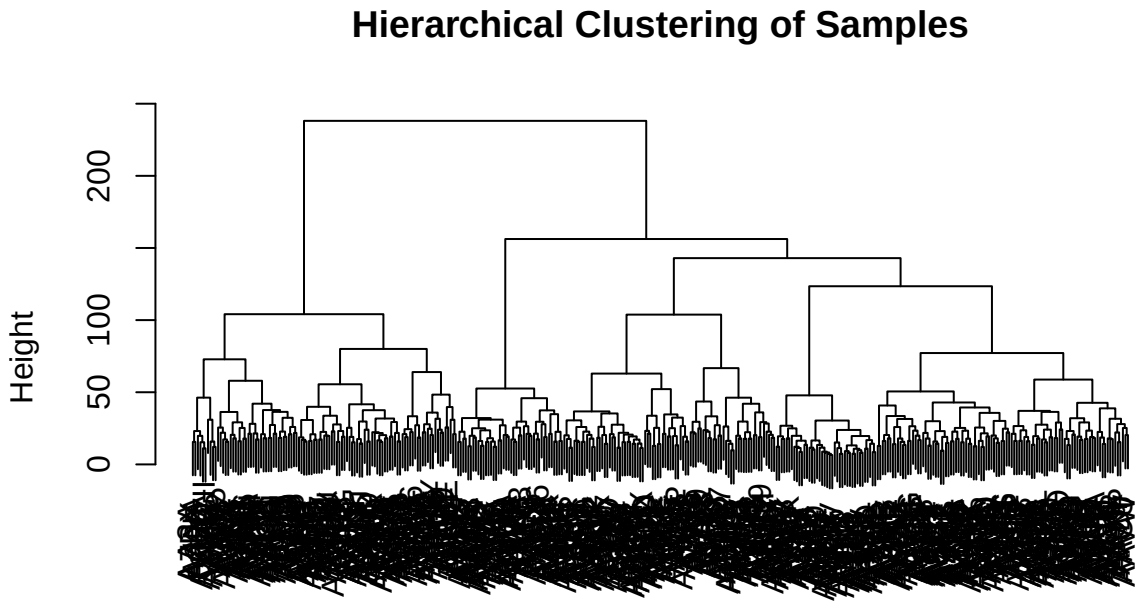
```
#Find the 50 most variably expressed genes

#Find Variance for each gene
expression_variance <- apply(log_RNA, 1, var)

#Sort based on variance
sorted_expression <- sort(expression_variance, decreasing = TRUE)

#Get top 50 genes
top_genes <- names(sorted_expression)[1:50]
```

```
#subset the expression data to only include the top 50 genes
top_RNA <- log_RNA[top_genes, ]

#Visualize using a heatmap
pheatmap(top_RNA, scale = "row", clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean", main = "Top 50 Variable Genes")
```



Clustering

```
# Create a distance matrix
# Transpose data to cluster
dist_matrix <- dist(t(top_RNA), method = "euclidean")

#Hierarchical clustering
hc <- hclust(dist_matrix, method = "ward.D2") #ward's method

#plot
plot(hc, main = "Hierarchical Clustering of Samples", xlab = "", sub = "")
```




```
#Cut the Dendrogram into Clusters
k <- 2
clusters <- as.data.frame(cutree(hc, k=k))
colnames(clusters) <- "Cluster"

table(clusters)
```

```
## Cluster
##   1   2
## 266 105
```

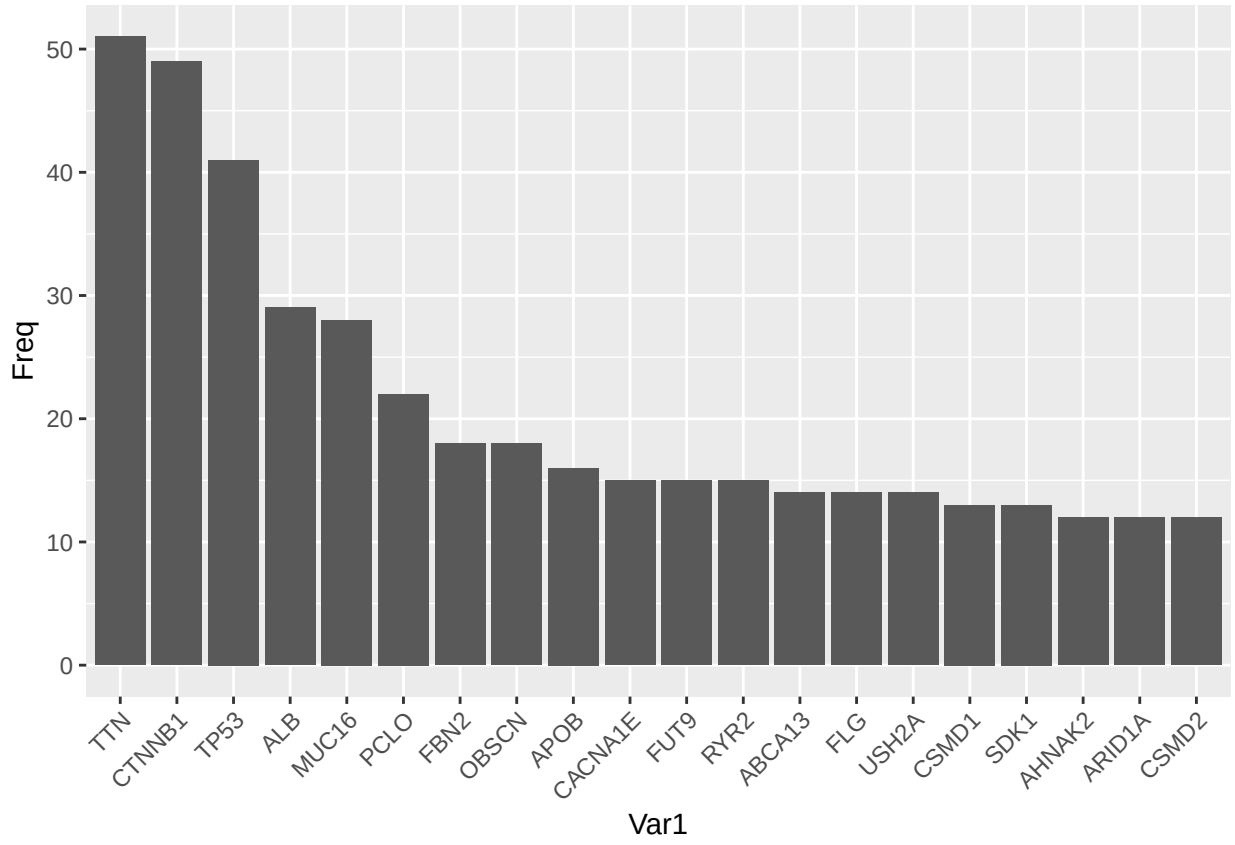
```r
#PCA
transposed_RNA <- t(top_RNA)  #transpose
pca <- prcomp(transposed_RNA, scale = TRUE)

pca_df <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2], Cluster = factor(clusters$Cluster), PATIENT_ID

#put in labels
pca_df$PATIENT_ID <- colnames(top_RNA)

#visualize with ggplot
ggplot(pca_df, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(size =4) +
  labs(title = paste("PCA Plot with", k, "Clusters"), x = "PC1", y = "PC2") +
  theme_minimal()
```

Frequency of Gene Muation Comparison Between Clusters

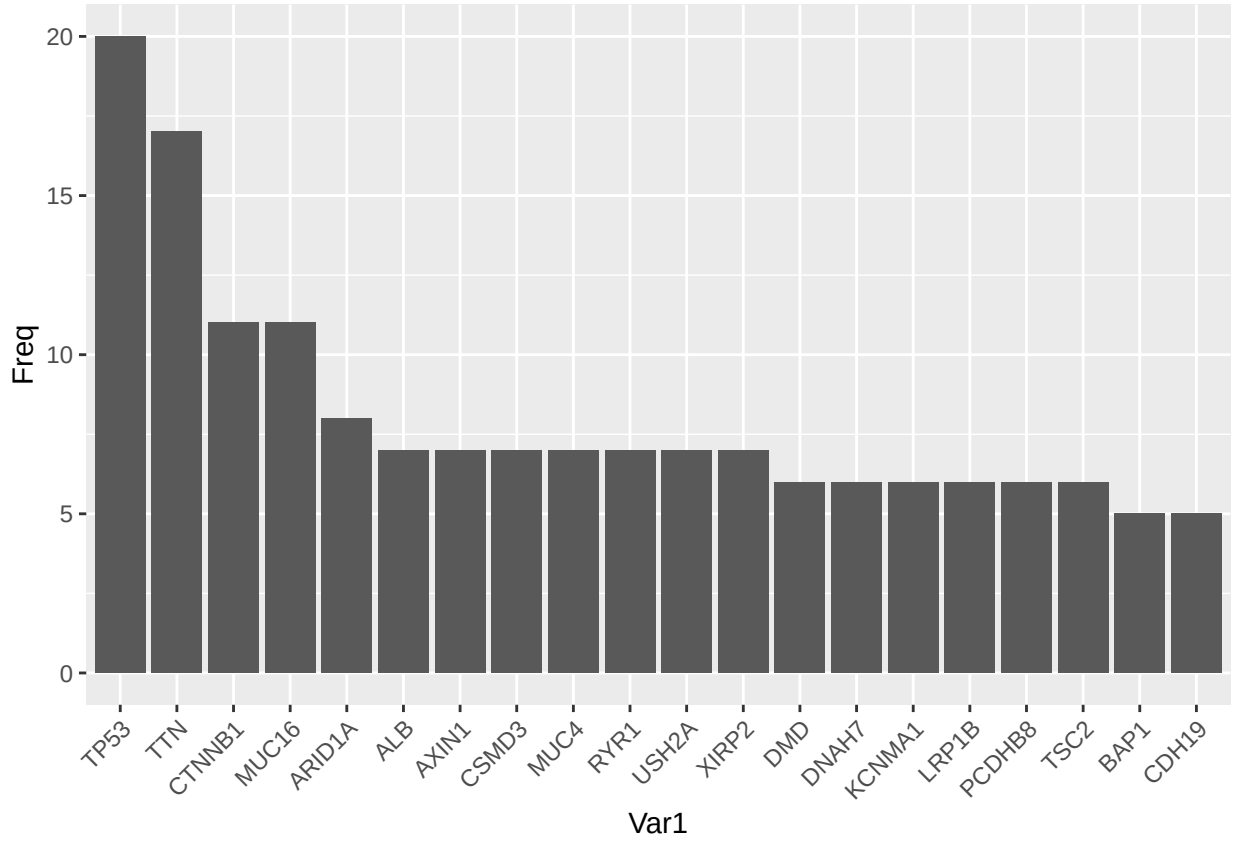
```r
#split clustered patients into lists
cluster_pat <- split(rownames(clusters), clusters$Cluster)

clust1_pat <- cluster_pat[[1]] #cluster 1 patients
clust2_pat <- cluster_pat[[2]] #cluster 2 patients
```

```r
#split mutation data into sets based on patients in each cluster
clust1_mut <- filtered_mutations[filtered_mutations$Tumor_Sample_Barcode %in% clust1_pat,]
clust2_mut <- filtered_mutations[filtered_mutations$Tumor_Sample_Barcode %in% clust2_pat,]
```

```r
#Finding the top 20 most frequently mutated genes in Cluster 1

clust1_genes_freq <- as.data.frame(table(clust1_mut$Hugo_Symbol))

clust1_genes_ordered <- clust1_genes_freq[order(-clust1_genes_freq$Freq),]

ggplot(data = clust1_genes_ordered[1:20,], aes(x = Var1, y = Freq)) +
  geom_col()+ #bar chart
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ #tilt text to make legible
  scale_x_discrete( limits = clust1_genes_ordered[1:20,]$Var1) #to retain new order
```



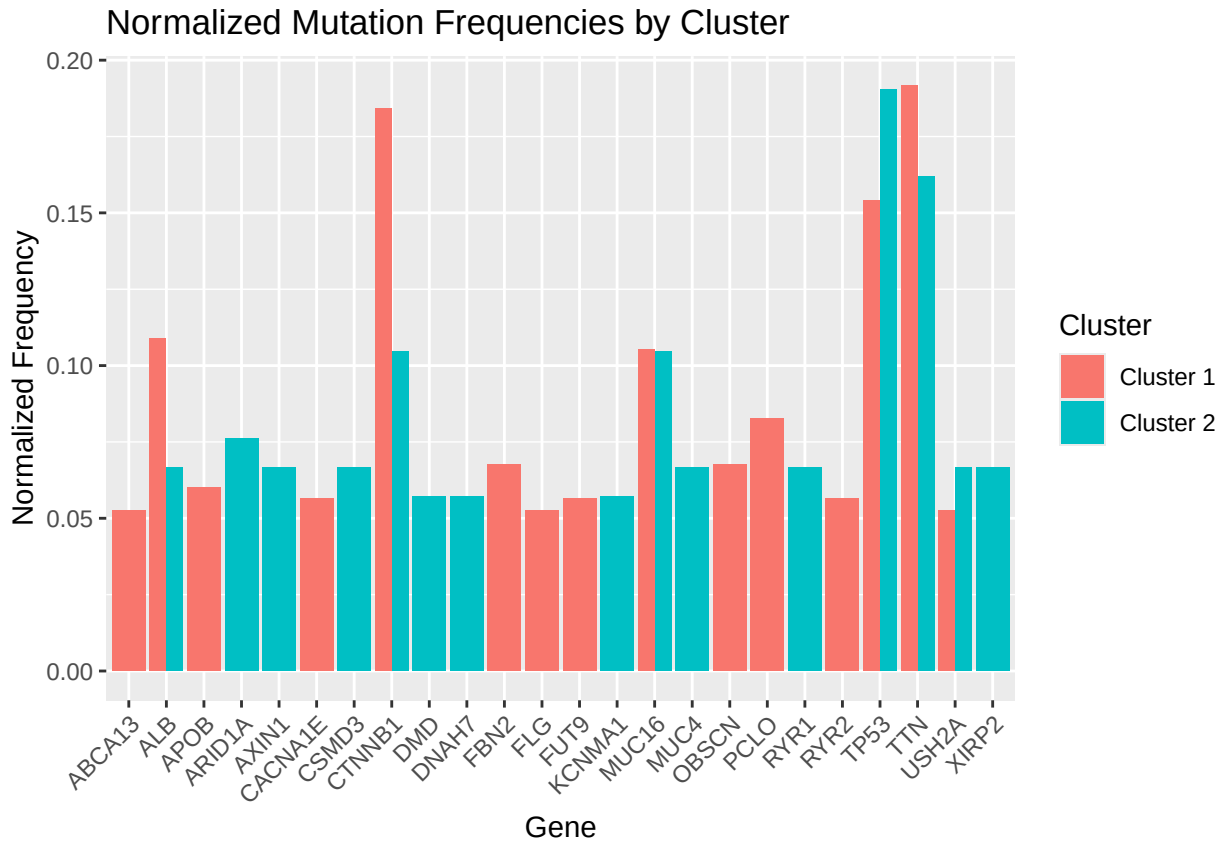

```r
#Finding the top 20 most frequently mutated genes in Cluster 2

clust2_genes_freq <- as.data.frame(table(clust2_mut$Hugo_Symbol))

clust2_genes_ordered <- clust2_genes_freq[order(-clust2_genes_freq$Freq),]

ggplot(data = clust2_genes_ordered[1:20,], aes(x = Var1, y = Freq)) +
  geom_col()+ #bar chart
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ #tilt text to make legible
  scale_x_discrete( limits = clust2_genes_ordered[1:20,]$Var1) #to retain new order
```



Find whether there is a meaningful difference in genes mutated between the clusters.

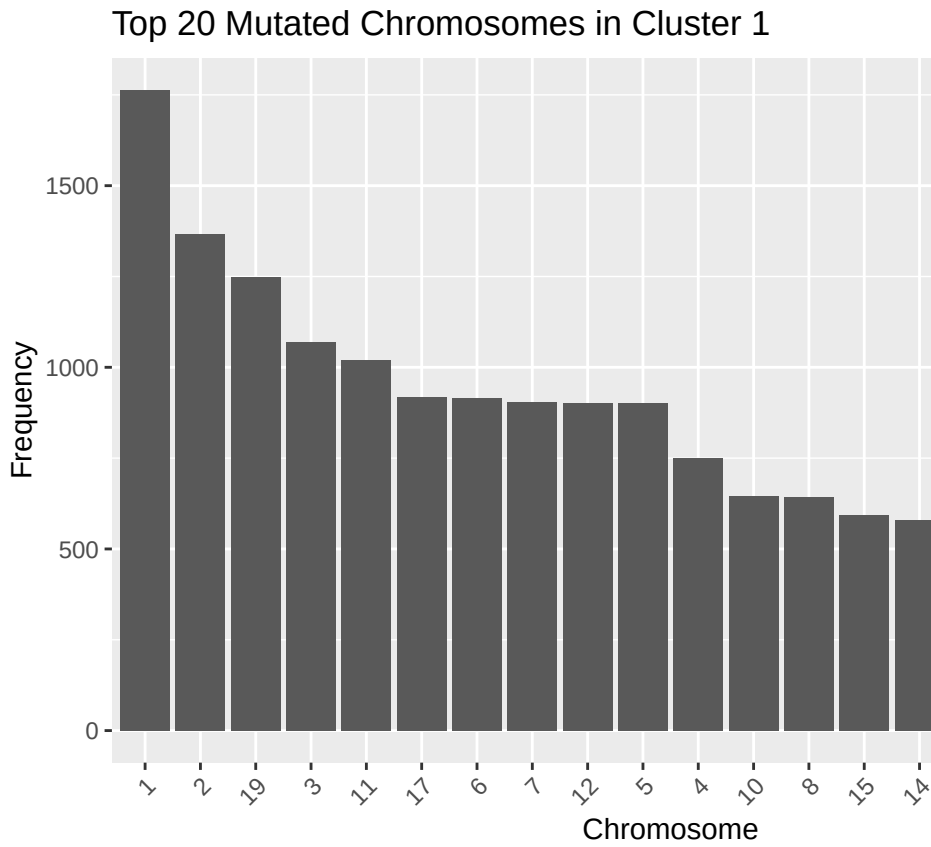
```
# Normalize Mutation Frequencies (to account for difference in cluster sizes)

# Normalize mutation frequencies for cluster 1
clust1_genes_ordered$Normalized_Freq <-
  clust1_genes_ordered$Freq / 266

# Normalize mutation frequencies for cluster 2
clust2_genes_ordered$Normalized_Freq <-
  clust2_genes_ordered$Freq / 105
```

```
#combine top genes from each cluster for comparison

# Add a cluster identifier to each data frame
clust1_genes_ordered$Cluster <- "Cluster 1"
clust2_genes_ordered$Cluster <- "Cluster 2"

# Combine the top 15 genes from both clusters
top_genes_combined <- rbind(clust1_genes_ordered[1:15,], clust2_genes_ordered[1:15,])

#plot
ggplot(top_genes_combined, aes(x = Var1, y = Normalized_Freq, fill = Cluster)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Normalized Mutation Frequencies by Cluster",
       x = "Gene", y = "Normalized Frequency")
```



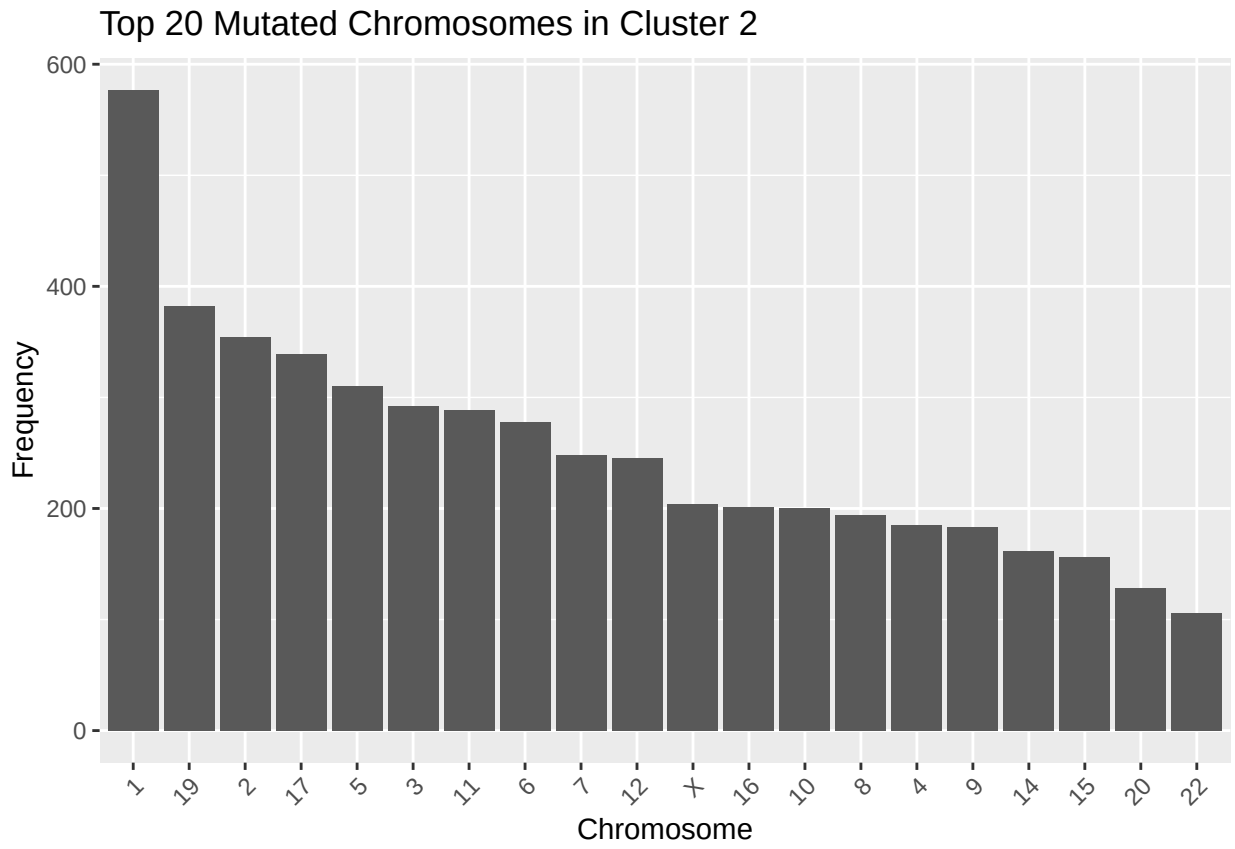

```
#we are interested in the genes with a significant mutational frequency difference between the two clus
```

```
# Frequency of mutations by chromosome in Cluster 1
clust1_chrom_freq <- as.data.frame(table(clust1_mut$Chromosome))

# Order chromosomes by frequency
clust1_chrom_ordered <- clust1_chrom_freq[order(-clust1_chrom_freq$Freq),]

# Plot the top 20 mutated chromosomes in Cluster 1
ggplot(data = clust1_chrom_ordered[1:20,], aes(x = Var1, y = Freq)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(limits = clust1_chrom_ordered[1:20,]$Var1) +
  labs(title = "Top 20 Mutated Chromosomes in Cluster 1", x = "Chromosome", y = "Frequency")
```

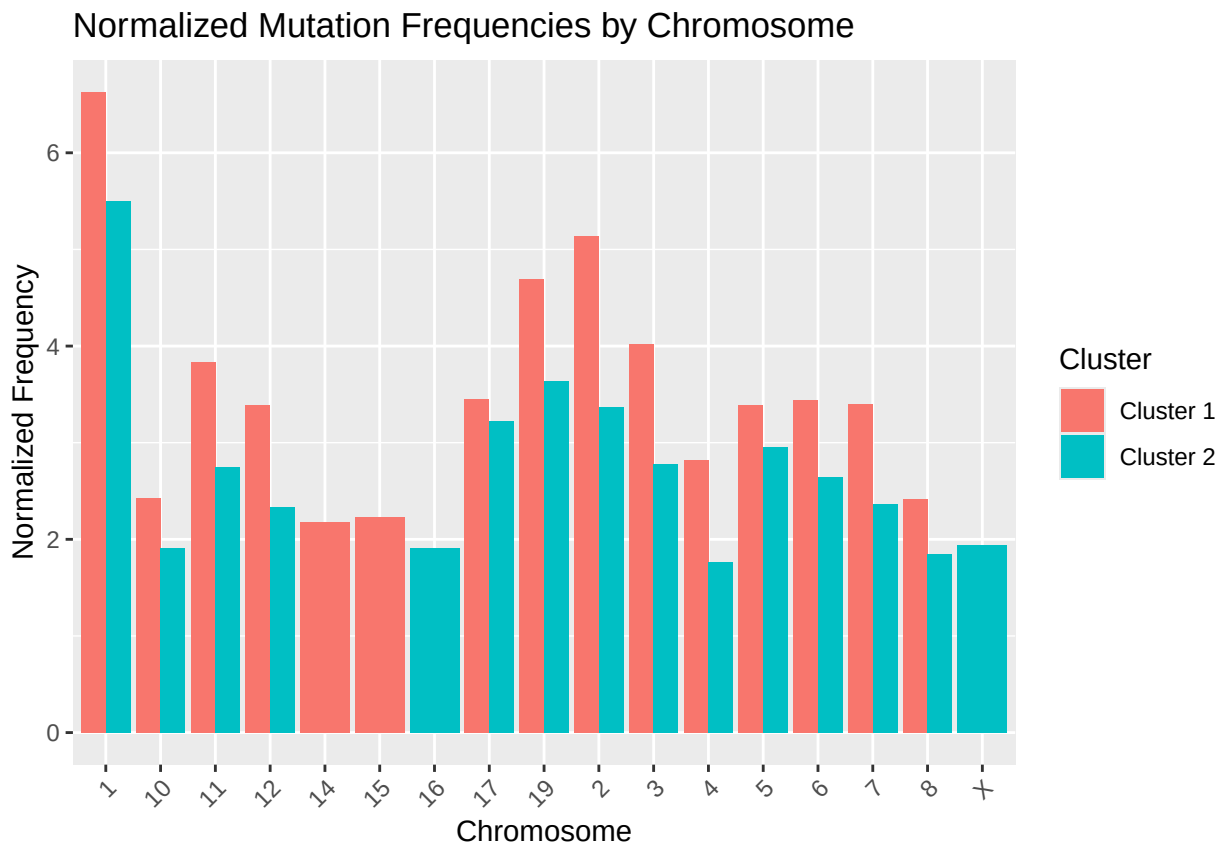


Repeat the Process for Chromosome

```
# Frequency of mutations by chromosome in Cluster 2
clust2_chrom_freq <- as.data.frame(table(clust2_mut$Chromosome))

# Order chromosomes by frequency
clust2_chrom_ordered <- clust2_chrom_freq[order(-clust2_chrom_freq$Freq),]

# Plot the top 20 mutated chromosomes in Cluster 2
ggplot(data = clust2_chrom_ordered[1:20,], aes(x = Var1, y = Freq)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(limits = clust2_chrom_ordered[1:20,]$Var1) +
  labs(title = "Top 20 Mutated Chromosomes in Cluster 2", x = "Chromosome", y = "Frequency")
```

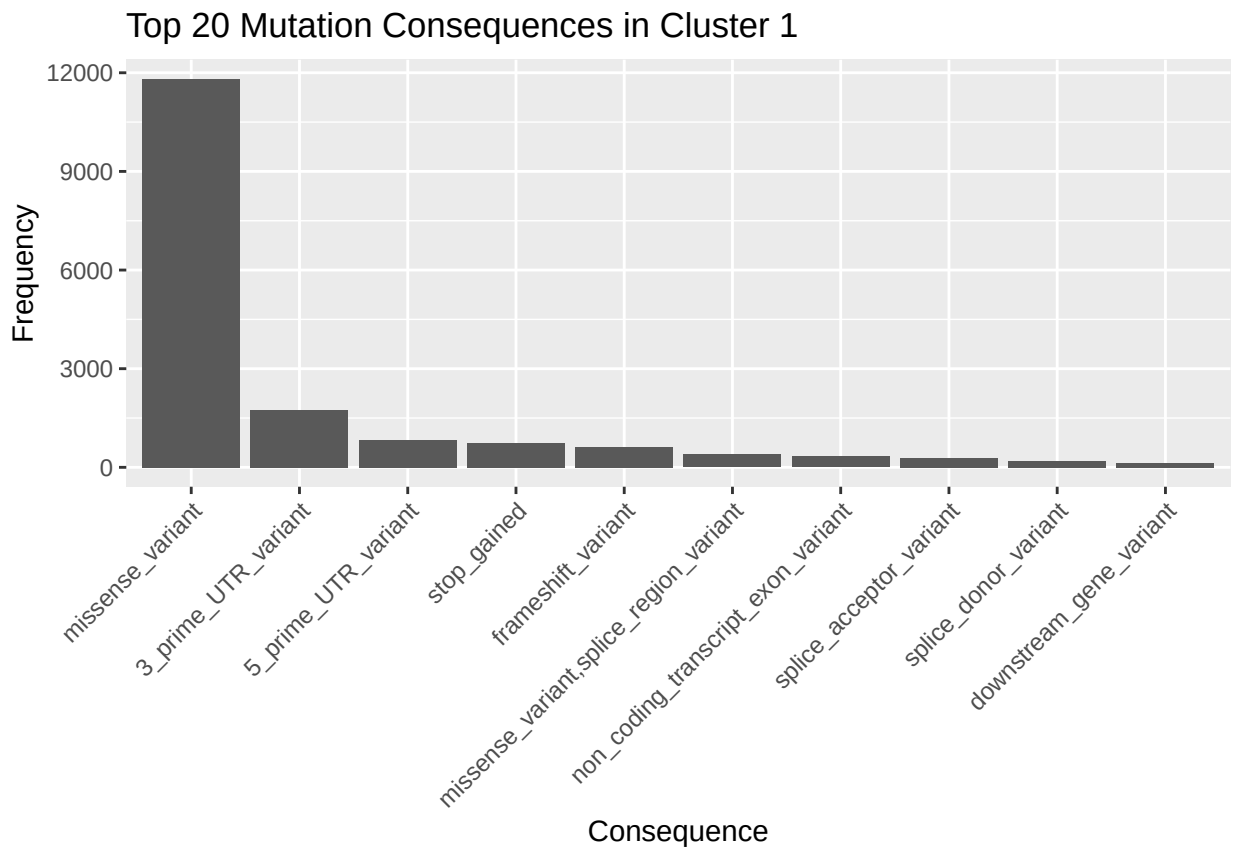



```
# Normalize mutation frequencies for Cluster 1
clust1_chrom_ordered$Normalized_Freq <- clust1_chrom_ordered$Freq / 266

# Normalize mutation frequencies for Cluster 2
clust2_chrom_ordered$Normalized_Freq <- clust2_chrom_ordered$Freq / 105

# Add a cluster identifier to each dataframe
clust1_chrom_ordered$Cluster <- "Cluster 1"
clust2_chrom_ordered$Cluster <- "Cluster 2"

# Combine the top 15 chromosomes from both clusters
top_chrom_combined <- rbind(clust1_chrom_ordered[1:15,], clust2_chrom_ordered[1:15,])

# Plot the normalized mutation frequencies by cluster
ggplot(top_chrom_combined, aes(x = Var1, y = Normalized_Freq, fill = Cluster)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Normalized Mutation Frequencies by Chromosome",
       x = "Chromosome", y = "Normalized Frequency")
```

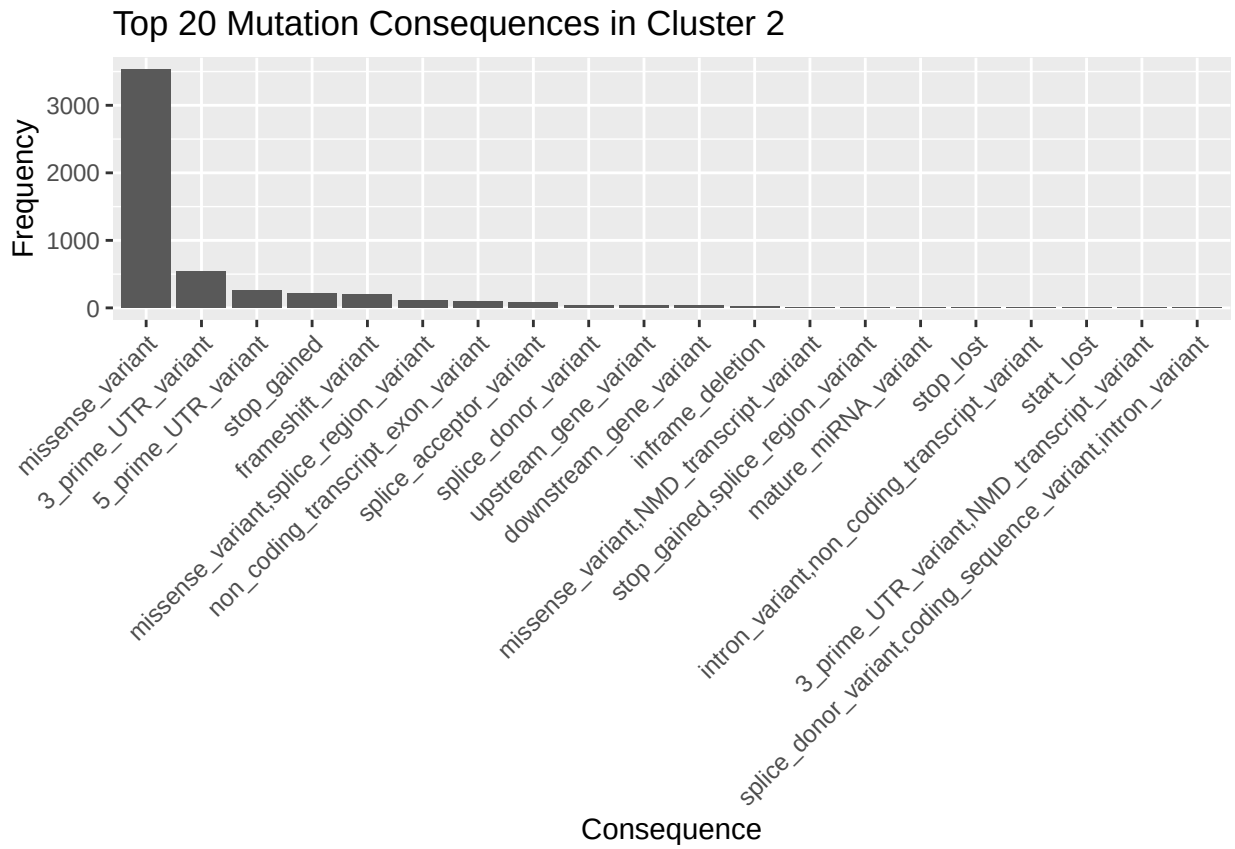


Repeat the Process for Consequence (to compare the resultant types of variants)

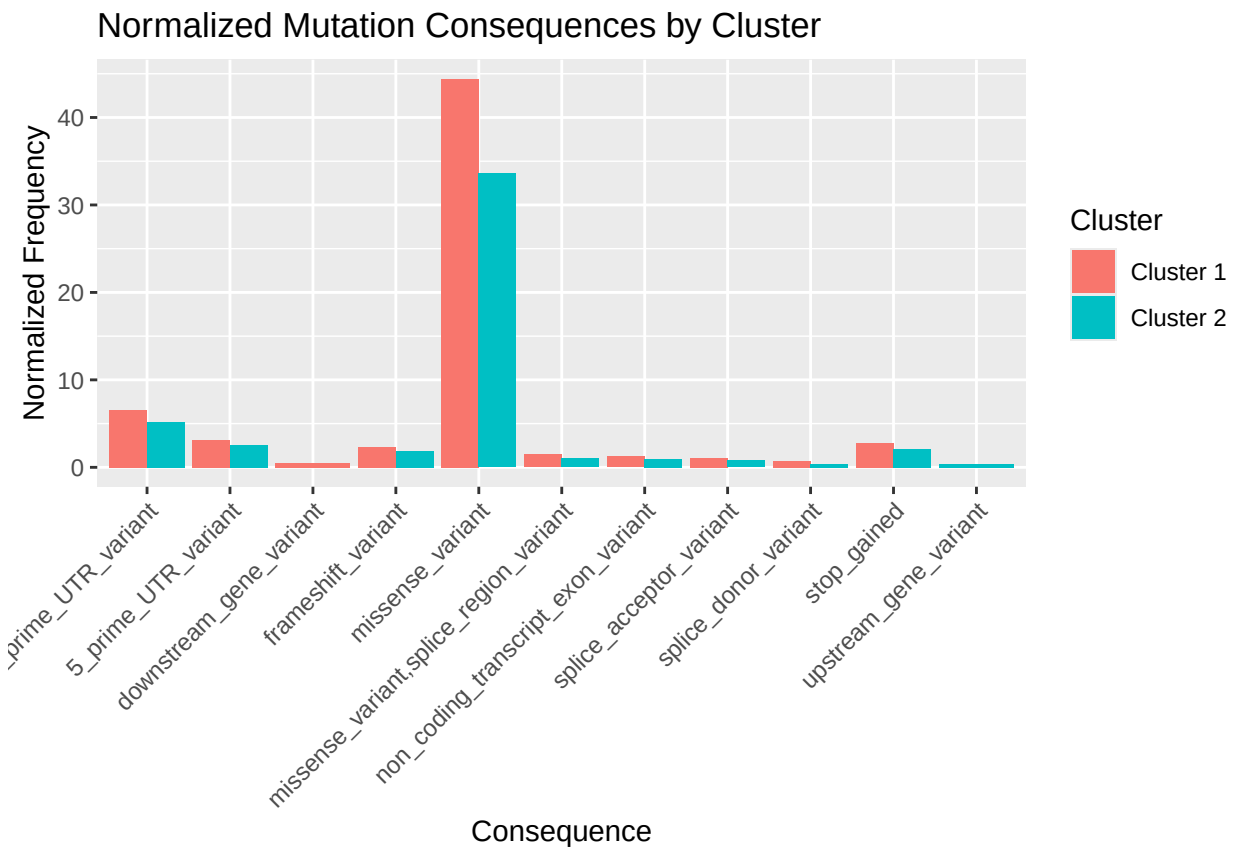
```
# Frequency of mutation consequences in Cluster 1
clust1_conseq_freq <- as.data.frame(table(clust1_mut$Consequence))

# Order consequences by frequency
clust1_conseq_ordered <- clust1_conseq_freq[order(-clust1_conseq_freq$Freq),]

# Plot the top 20 most frequent consequences in Cluster 1
ggplot(data = clust1_conseq_ordered[1:10,], aes(x = Var1, y = Freq)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(limits = clust1_conseq_ordered[1:10,]$Var1) +
  labs(title = "Top 20 Mutation Consequences in Cluster 1", x = "Consequence", y = "Frequency")
```




```r
# Frequency of mutation consequences in Cluster 2
clust2_conseq_freq <- as.data.frame(table(clust2_mut$Consequence))

# Order consequences by frequency
clust2_conseq_ordered <- clust2_conseq_freq[order(-clust2_conseq_freq$Freq),]

# Plot the top 20 most frequent consequences in Cluster 2
ggplot(data = clust2_conseq_ordered[1:20,], aes(x = Var1, y = Freq)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(limits = clust2_conseq_ordered[1:20,]$Var1) +
  labs(title = "Top 20 Mutation Consequences in Cluster 2", x = "Consequence", y = "Frequency")
```




```r
# Normalize mutation frequencies for Cluster 1
clust1_conseq_ordered$Normalized_Freq <- clust1_conseq_ordered$Freq / 266

# Normalize mutation frequencies for Cluster 2
clust2_conseq_ordered$Normalized_Freq <- clust2_conseq_ordered$Freq / 105

# Add a cluster identifier to each dataframe
clust1_conseq_ordered$Cluster <- "Cluster 1"
clust2_conseq_ordered$Cluster <- "Cluster 2"

# Combine the top 15 consequences from both clusters
top_conseq_combined <- rbind(clust1_conseq_ordered[1:10,], clust2_conseq_ordered[1:10,])

# Plot the normalized mutation frequencies by cluster
ggplot(top_conseq_combined, aes(x = Var1, y = Normalized_Freq, fill = Cluster)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Normalized Mutation Consequences by Cluster",
       x = "Consequence", y = "Normalized Frequency")
```



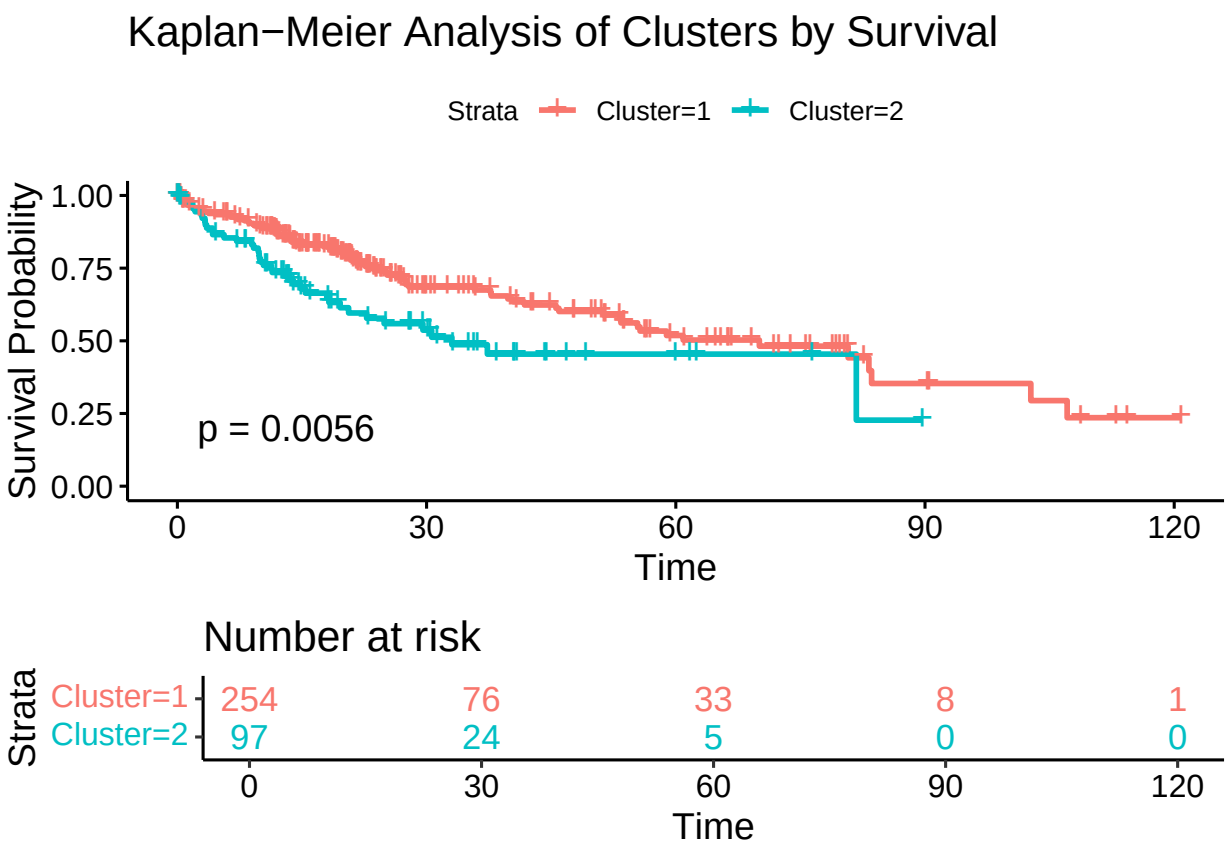
Observation: Cluster one had more missense variants. Cluster 1 had more downstream gene variants, while Cluster 2 had more upstream gene variants.

```
# Extract List of patients in each cluster

clust1_IDs <- names(clusters[clusters == 1])
clust2_IDs <- names(clusters[clusters == 2])

# Save patient names to lists
list_of_patients <- list(
  Cluster1 = clust1_IDs,
  Cluster2 = clust2_IDs
)
```

#Survival Analysis on Clinical Data to compare Patients in the Clusters ####Kaplan-Meier Analysis of Clusters by Survival

```
#add cluster information to the clinical data
unq_patient_df <- unq_patient_df %>%
  mutate(Cluster = clusters$Cluster[match(PATIENT_ID, rownames(clusters))])

#missing clusters?
unq_patient_df <- unq_patient_df %>%
  filter(!is.na(Cluster))

#survival model based on clusters
fit_clusters <- survfit(Surv(overall_survival, deceased) ~ Cluster, data = unq_patient_df)
```

```r
#Kaplan-Meier Analysis of Clusters by Survival
ggsurvplot(
  fit_clusters,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  risk.table.height = 0.3,
  title = "Kaplan-Meier Analysis of Clusters by Survival",
  xlab = "Time",
  ylab = "Survival Probability",
)
```

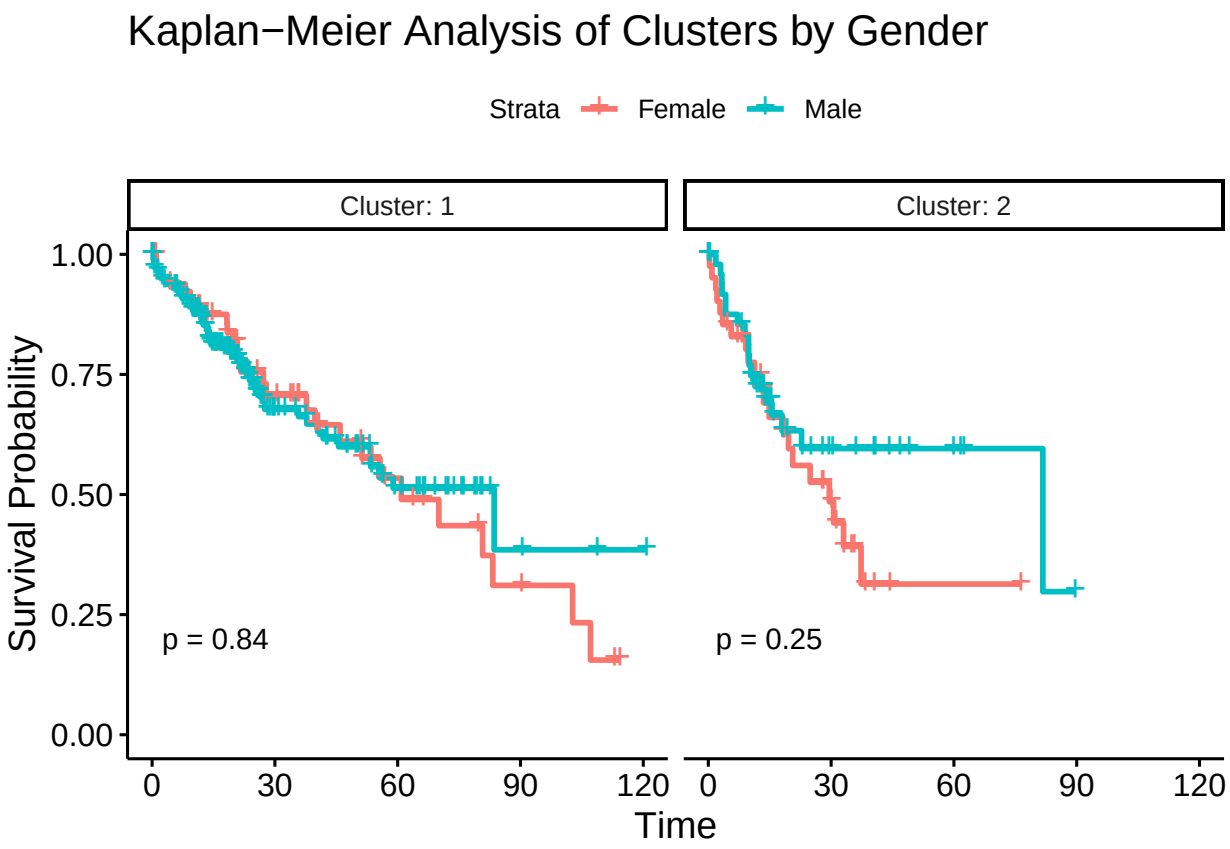


####Kaplan-Meier Analysis of Clusters by Gender

```r
#survival model by gender within clusters
fit_gender_cluster <- survfit(Surv(overall_survival, deceased) ~ SEX + Cluster, data = unq_patient_df)

#Kaplan-Meier plot
ggsurvplot(
  fit_gender_cluster,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
```

```
  risk.table.height = 0.3,
  facet.by = "Cluster",                #separate plots for each cluster (hehe looks pretty)
  title = "Kaplan-Meier Analysis of Clusters by Gender",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 4
```

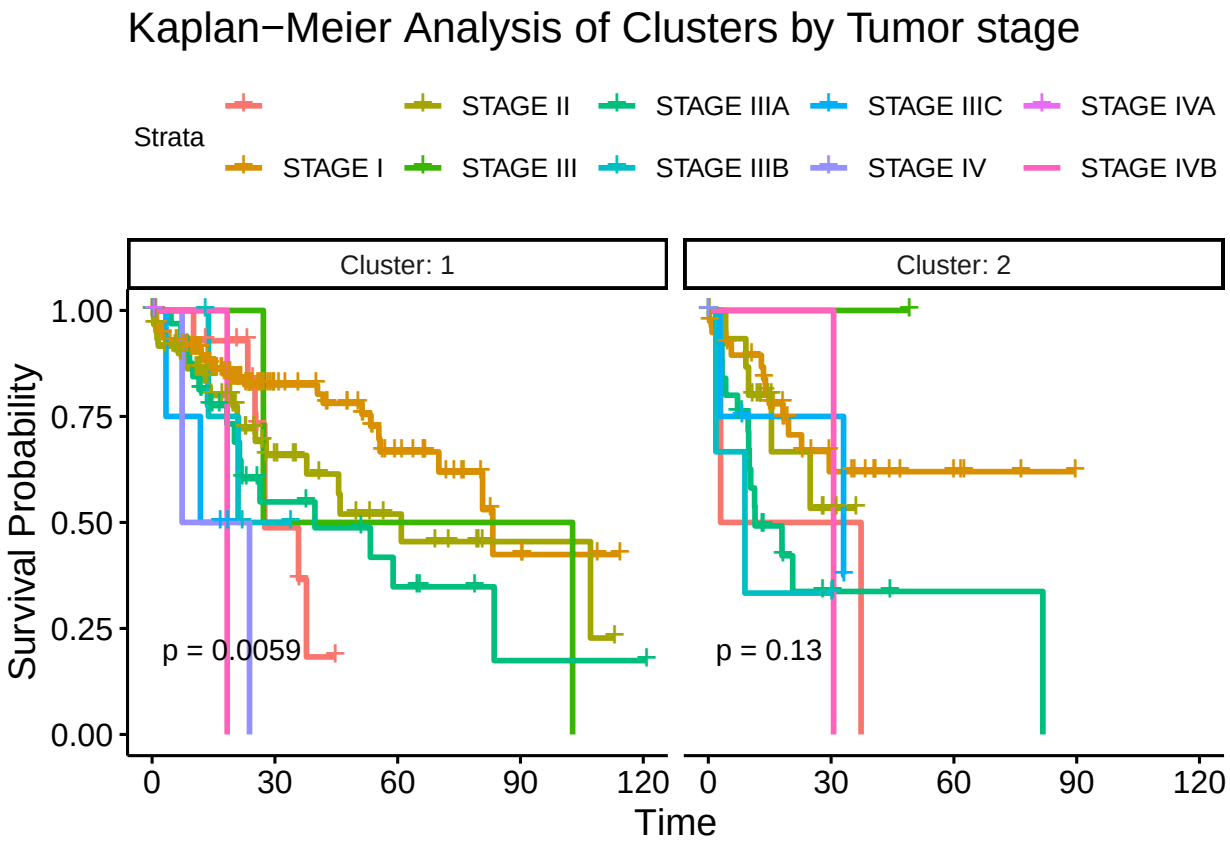


####Kaplan-Meier Analysis of Clusters by Tumor stage

```
#Simplify tumor stage for analysis
unq_patient_df <- unq_patient_df %>%
  mutate(tumor_stage = gsub("[abc]$", "", AJCC_PATHOLOGIC_TUMOR_STAGE))

#survival model by tumor stage within clusters
fit_tumor_stage_cluster <- survfit(Surv(overall_survival, deceased) ~ tumor_stage + Cluster, data = unq

#kaplan-Meier plot
ggsurvplot(
  fit_tumor_stage_cluster,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
```

```
  risk.table.col = "strata",
  risk.table.height = 0.3,
  facet.by = "Cluster",
  title = "Kaplan-Meier Analysis of Clusters by Tumor stage",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 19
```



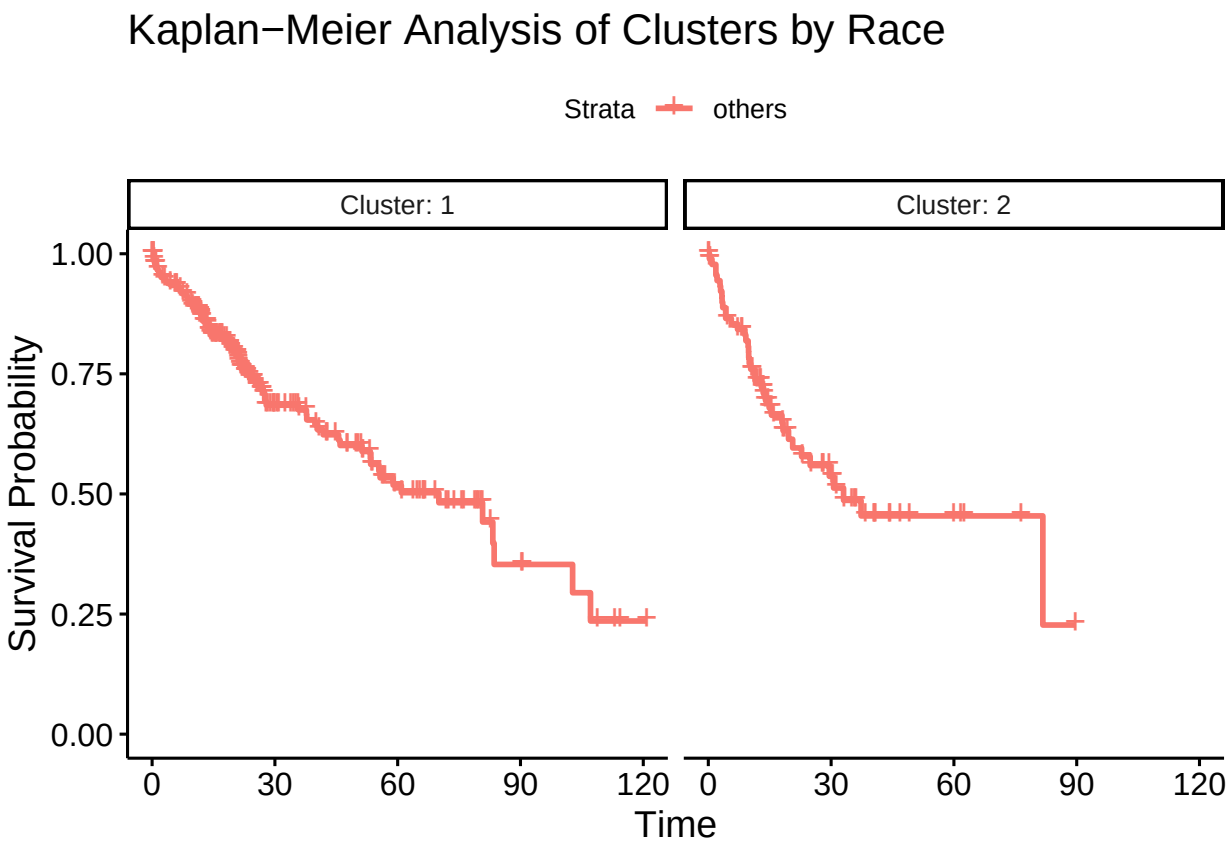
####Kaplan-Meier Analysis of Clusters by Race

```
#Simplify Group races
unq_patient_df <- unq_patient_df %>%
  mutate(race_group = ifelse(RACE %in% c("white", "asian", "black or african american"), RACE, "others")

#survival model by race within clusters
fit_race_cluster <- survfit(Surv(overall_survival, deceased) ~ race_group + Cluster, data = unq_patient_

#Kaplan-Meier plot
ggsurvplot(
  fit_race_cluster,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
```

```
  risk.table.col = "strata",
  risk.table.height = 0.3,
  facet.by = "Cluster",
  title = "Kaplan-Meier Analysis of Clusters by Race",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 2
```

```
## Warning in .f(.x[[i]], .y[[i]], ...): There are no survival curves to be compared.
##  This is a null model.
## Warning in .f(.x[[i]], .y[[i]], ...): There are no survival curves to be compared.
##  This is a null model.
```

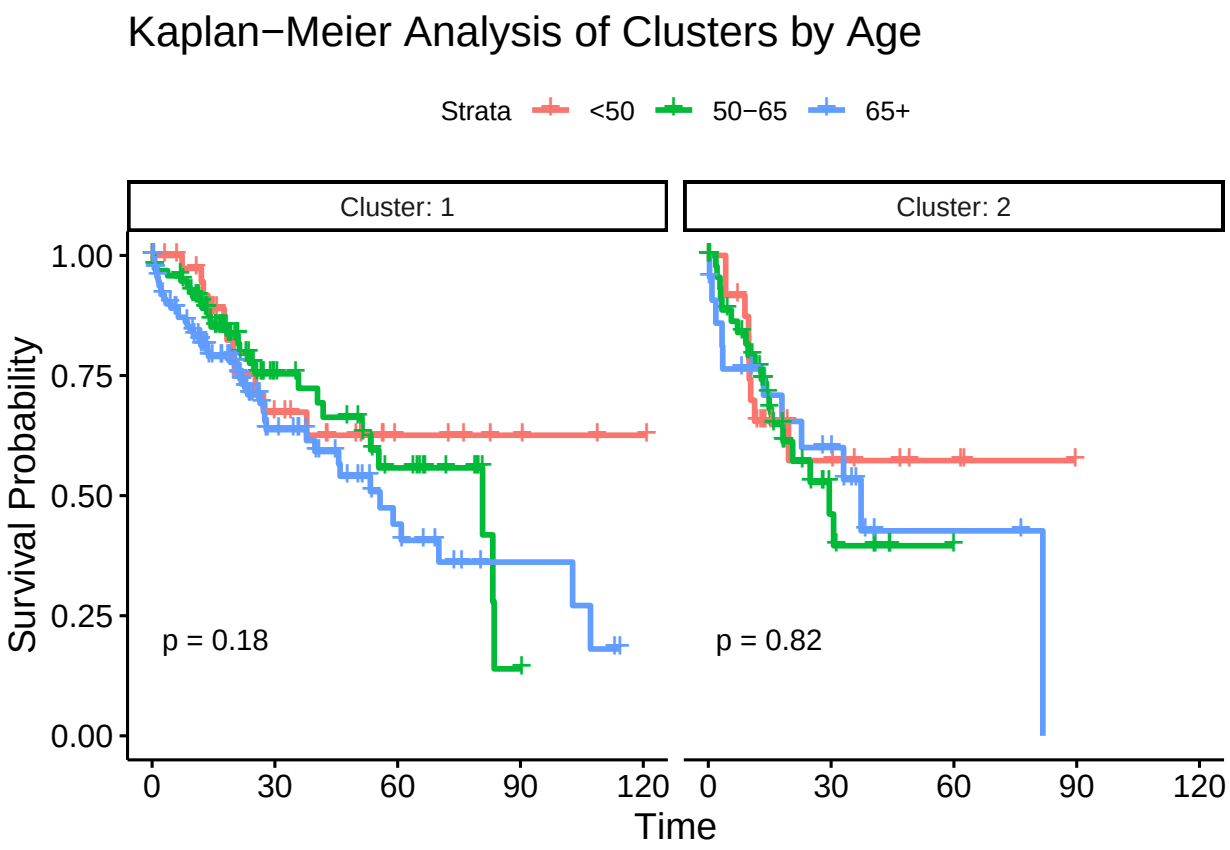


####Kaplan-Meier Analysis of Clusters by Age

```
#categorize age groups
unq_patient_df <- unq_patient_df %>%
  mutate(age_group = case_when(
    AGE < 50 ~ "<50",
    AGE >= 50 & AGE < 65 ~ "50-65",
    AGE >= 65 ~ "65+"
  ))
```

```
#survival model by age group within clusters
fit_age_group_cluster <- survfit(Surv(overall_survival, deceased) ~ age_group + Cluster, data = unq_pati

#Kaplan-Meier plot
ggsurvplot(
  fit_age_group_cluster,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  risk.table.height = 0.3,
  facet.by = "Cluster",
  title = "Kaplan-Meier Analysis of Clusters by Age",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 6
```

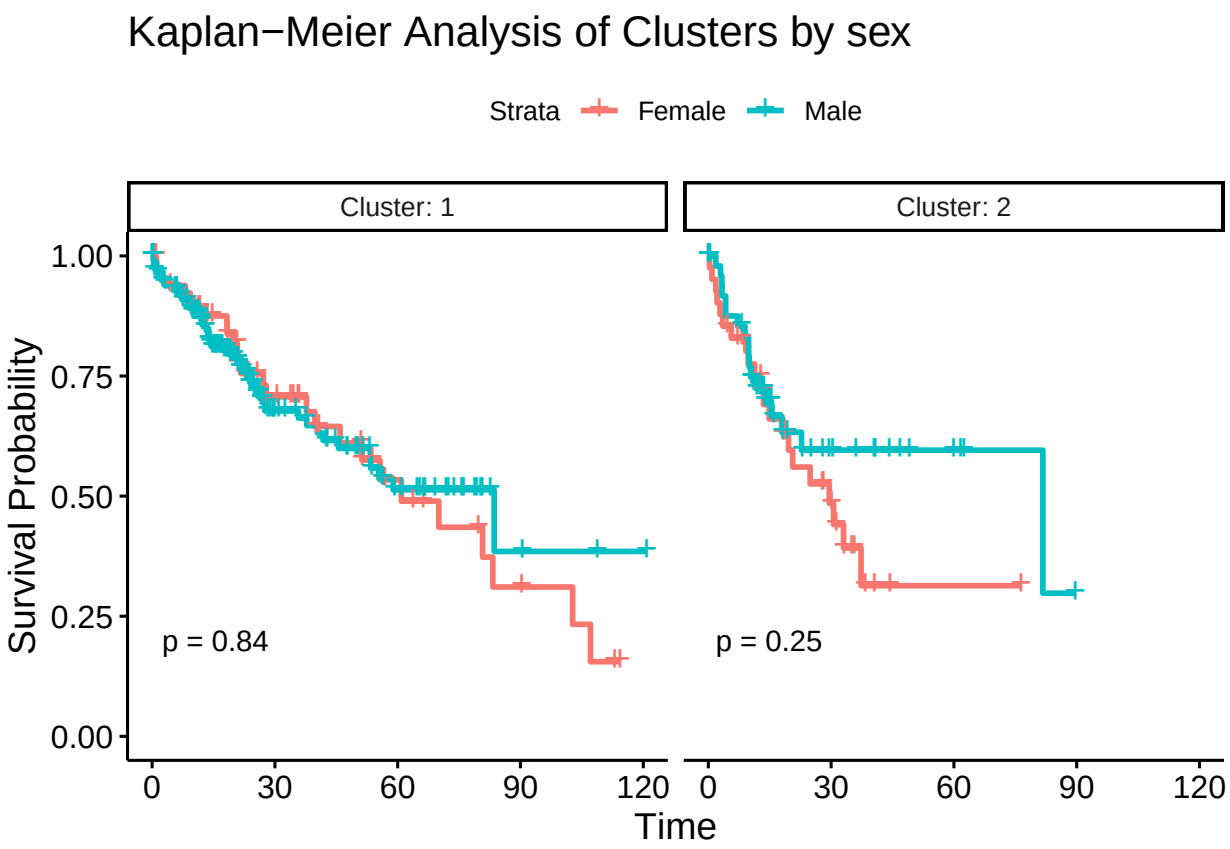


####Kaplan-Meier Analysis of Clusters by sex

```
#model by SEX within clusters
fit_sex_cluster <- survfit(Surv(overall_survival, deceased) ~ SEX + Cluster, data = unq_patient_df)
```

```
# Kaplan-Meier plot
ggsurvplot(
  fit_sex_cluster,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  risk.table.height = 0.3,
  facet.by = "Cluster",
  title = "Kaplan-Meier Analysis of Clusters by sex",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 4
```

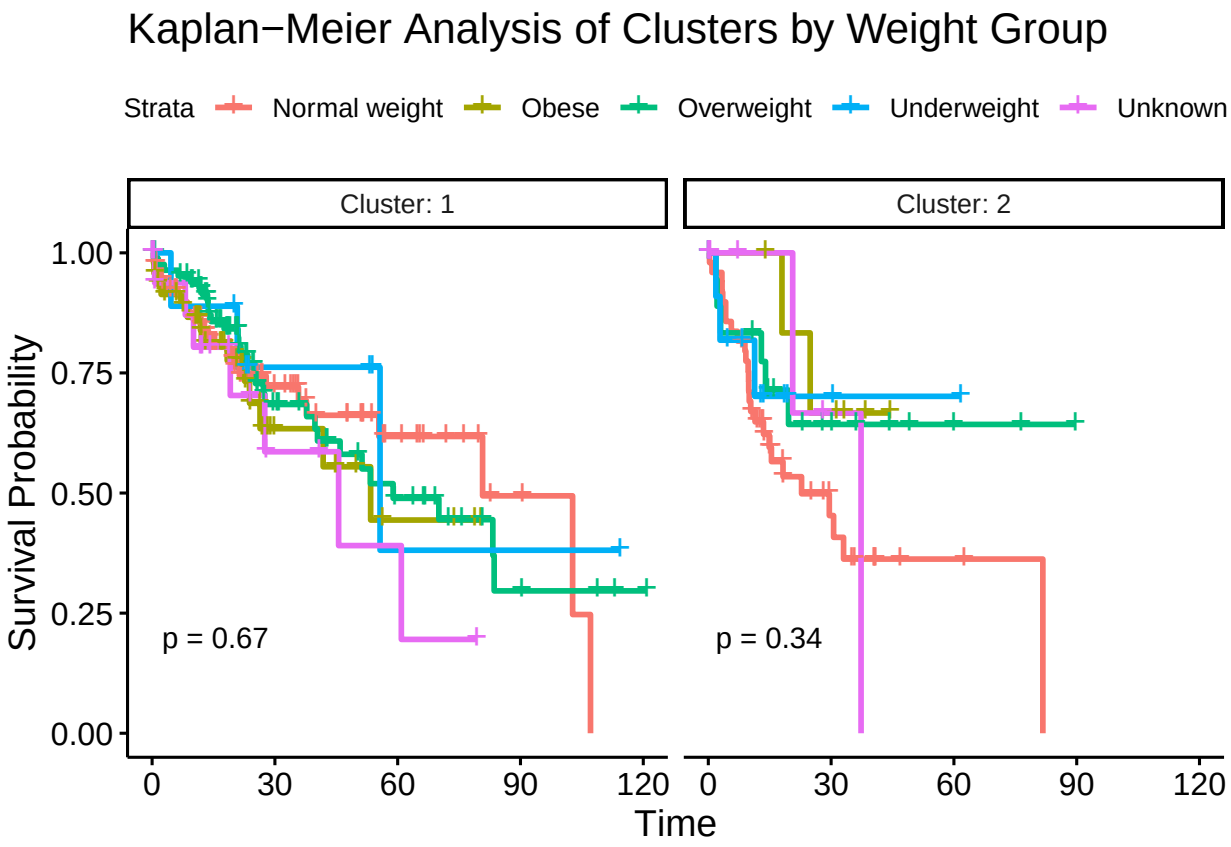


####Kaplan-Meier Analysis of Clusters by Weight Group

```
#model by weight group within clusters
fit_weight_group_cluster <- survfit(Surv(overall_survival, deceased) ~ weight_group + Cluster, data = u

#Kaplan-Meier plot
ggsurvplot(
  fit_weight_group_cluster,
```

```
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  risk.table.height = 0.3,
  facet.by = "Cluster",
  title = "Kaplan-Meier Analysis of Clusters by Weight Group",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 10
```

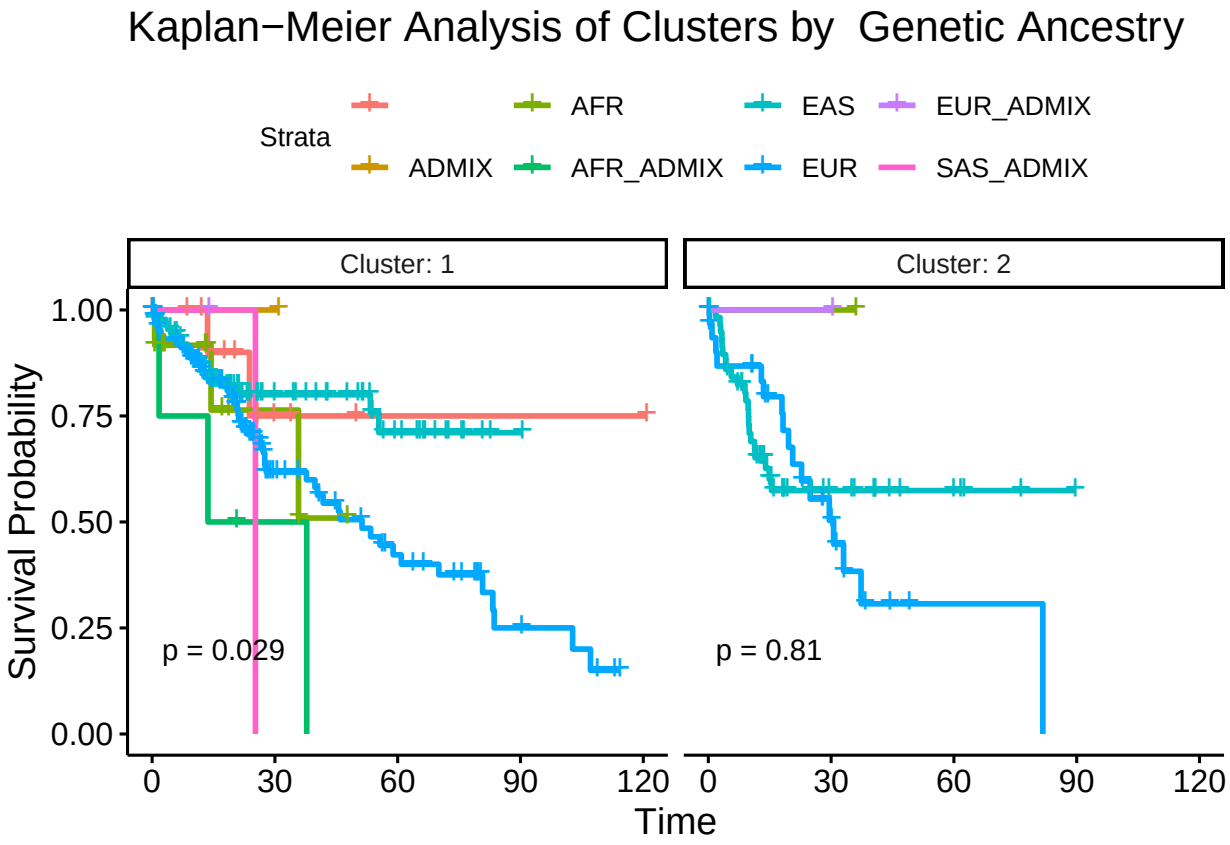


####Kaplan-Meier Analysis of Clusters by Genetic Ancestry

```
#model by genetic ancestry within clusters
fit_genetic_ancestry_cluster <- survfit(Surv(overall_survival, deceased) ~ GENETIC_ANCESTRY_LABEL + Clus

#Kaplan-Meier plot
ggsurvplot(
  fit_genetic_ancestry_cluster,
  data = unq_patient_df,
  pval = TRUE,
  risk.table = TRUE,
```

```
  risk.table.col = "strata",
  risk.table.height = 0.3,
  facet.by = "Cluster",
  title = "Kaplan-Meier Analysis of Clusters by  Genetic Ancestry",
  xlab = "Time",
  ylab = "Survival Probability"
)
```

```
## Warning in (function (survsummary, times, survtable = c("cumevents",
## "risk.table", : The length of legend.labs should be 13
```



#DESeq2 Analysis on Expression Data to compare the clusters

```
# prepare expression data for DESeq

de_RNA <- preprocessed_RNA
# Setup DESeq Object and run DESeq Pipeline
colData <- clusters
countData <- de_RNA

# Set up Cluster numbers as factor for DESeq
colData$Cluster <-  as.factor(colData$Cluster)

# Create DESeq data set
dds <- DESeqDataSetFromMatrix(countData =  countData,
```

```
                                colData = colData,
                                design = ~ Cluster)
```

```
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 6140 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```
dds
```

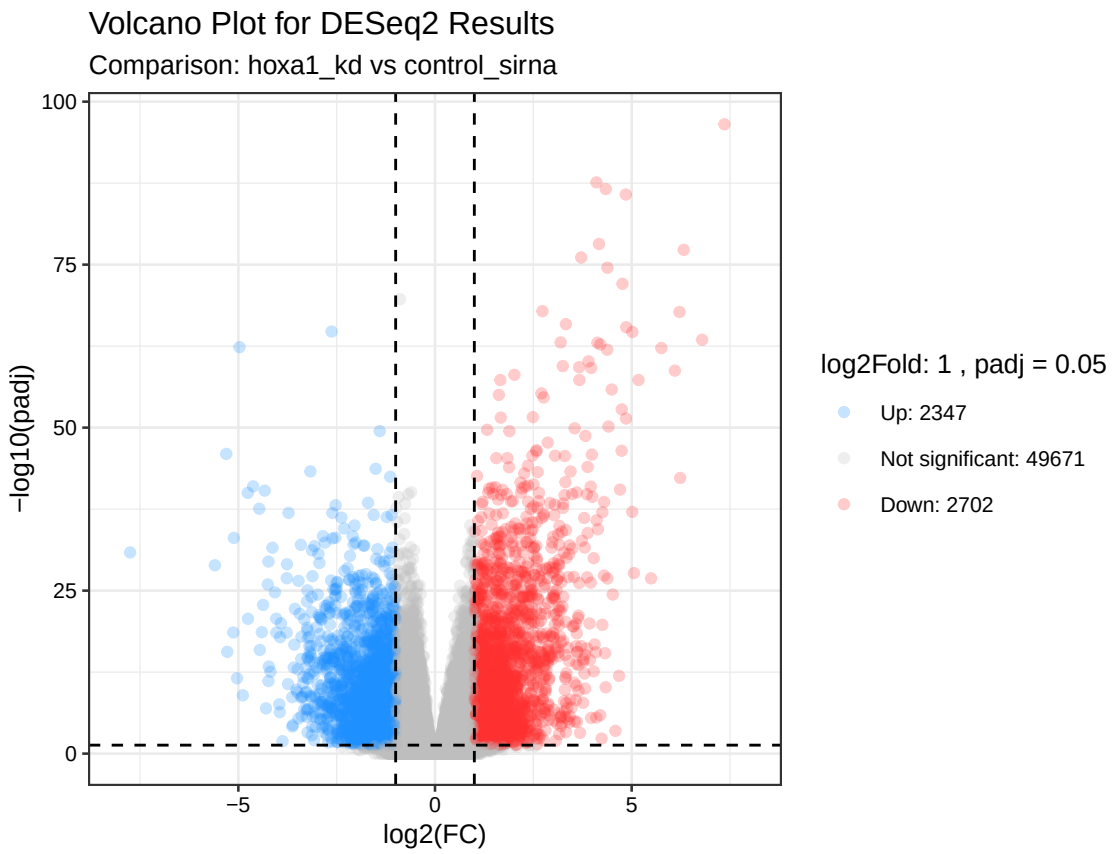
```
## class: DESeqDataSet
## dim: 54720 371
## metadata(1): version
## assays(6): counts mu ... replaceCounts replaceCooks
## rownames(54720): ENSG00000000003.15 ENSG00000000005.6 ...
##   ENSG00000288674.1 ENSG00000288675.1
## rowData names(23): baseMean baseVar ... maxCooks replace
## colnames(371): A1EG A7SF ... AACC AA1D
## colData names(3): Cluster sizeFactor replaceable
```

```
#building DESeq results table
res <- results(dds, contrast = c("Cluster", "1", "2"))
res_df <- as.data.frame(res)

#summary
summary(res)
```

```
##
## out of 54714 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 7964, 15%
## LFC < 0 (down)     : 12262, 22%
## outliers [1]       : 0, 0%
## low counts [2]     : 18039, 33%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```r
#Create Volcano Plot

# Function 1: Prepare DESeq2 results for plotting
# Convert DESeqResults object to data.frame
prepare_res_for_plot <- function(res) {
  res_df
  # rownames as a gene column
  res_df$hgnc_symbol <- rownames(res_df)

  return(res_df)
}

# Function 2: Volcano plot function
volcplot <- function(data, padj_threshold = 0.05, log2Fold_threshold = 1, plot_title = 'Volcano Plot', p
  # Set the fold-change thresholds
  neg_log2fc <- -log2Fold_threshold
  pos_log2fc <- log2Fold_threshold

  # Replace NA values
  data$padj[is.na(data$padj)] <- 1
  data$log2FoldChange[is.na(data$log2FoldChange)] <- 0

  # Add log2fc_threshold column
  data$log2fc_threshold <- ifelse(
    data$log2FoldChange >= pos_log2fc & data$padj <= padj_threshold, 'up',
    ifelse(data$log2FoldChange <= neg_log2fc & data$padj <= padj_threshold, 'down', 'ns')
  )


  # Count up, down, and unchanged genes
  up_genes <- sum(data$log2fc_threshold == 'up')
  down_genes <- sum(data$log2fc_threshold == 'down')
  unchanged_genes <- sum(data$log2fc_threshold == 'ns')

  # Generate legend labels
  legend_labels <- c(
    paste0('Up: ', up_genes),
    paste0('Not significant: ', unchanged_genes),
    paste0('Down: ', down_genes)
  )

  # Calculate x-axis limits
  x_axis_limits <- ceiling(max(abs(data$log2FoldChange)))

  # Define plot colors
  plot_colors <- c(
    'up' = 'firebrick1',
    'ns' = 'gray',
    'down' = 'dodgerblue1'
  )

  # Create the plot
  plot <- ggplot(data) +
```

```
    geom_point(
      aes(x = log2FoldChange, y = -log10(padj), color = log2fc_threshold),
      alpha = 0.25,
      size = 1.5
    ) +
    geom_vline(xintercept = c(neg_log2fc, pos_log2fc), linetype = 'dashed') +
    geom_hline(yintercept = -log10(padj_threshold), linetype = 'dashed') +
    scale_x_continuous(
      'log2(FC)',
      limits = c(-x_axis_limits, x_axis_limits)
    ) +
    scale_color_manual(
      values = plot_colors,
      labels = legend_labels
    ) +
    labs(
      color = paste('log2Fold:', log2Fold_threshold, ', padj', expression("\u2264"), padj_threshold),
      title = plot_title,
      subtitle = plot_subtitle
    ) +
    theme_bw(base_size = 10) +
    theme(
      aspect.ratio = 1,
      axis.text = element_text(color = 'black'),
    )

  return(plot)
}

# Generate the volcano plot
volcplot(
  data = res_df,
  padj_threshold = 0.05,
  log2Fold_threshold = 1,
  plot_title = "Volcano Plot for DESeq2 Results",
  plot_subtitle = "Comparison: hoxa1_kd vs control_sirna"
)
```




```
# retrieve all significant genes

genes.significant <- rownames(res)[!is.na(res$padj &
                  res$padj < 0.05 &
                  abs(res$log2FoldChange) > 1)]

resSig <- res[!is.na(res$padj) & res$padj < 0.05, ]
# Get the names of top 20 upregulated and downregulated genes
genes.top.upreg <- rownames(resSig)[order(resSig$log2FoldChange, decreasing = TRUE)[1:20]]
genes.top.downreg <- rownames(resSig)[order(resSig$log2FoldChange, decreasing = FALSE)[1:20]]


# Bind the two lists of genes into a single vector
genes.top <- c(genes.top.upreg, genes.top.downreg)
genes.top
```

```
##  [1] "ENSG00000106633.17" "ENSG00000267444.1"  "ENSG00000286398.1"
##  [4] "ENSG00000104938.18" "ENSG00000140505.7"  "ENSG00000237700.2"
##  [7] "ENSG00000204949.9"  "ENSG00000263761.3"  "ENSG00000198077.10"
## [10] "ENSG00000163217.2"  "ENSG00000173432.12" "ENSG00000165682.14"
## [13] "ENSG00000151365.2"  "ENSG00000160339.16" "ENSG00000172425.12"
## [16] "ENSG00000249173.6"  "ENSG00000134339.9"  "ENSG00000197888.2"
## [19] "ENSG00000230435.1"  "ENSG00000251049.2"  "ENSG00000006659.13"
## [22] "ENSG00000287093.1"  "ENSG00000149948.14" "ENSG00000256081.2"
## [25] "ENSG00000072041.17" "ENSG00000268104.3"  "ENSG00000157005.4"
## [28] "ENSG00000163618.18" "ENSG00000269586.7"  "ENSG00000130876.11"
```

```
## [31] "ENSG00000286101.1"  "ENSG00000081051.8"  "ENSG00000174498.14"
## [34] "ENSG00000185559.16" "ENSG00000287550.1"  "ENSG00000188620.11"
## [37] "ENSG00000100604.13" "ENSG00000177468.7"  "ENSG00000180066.10"
## [40] "ENSG00000118017.4"
```

```
#Find Cluster-specific genes

#Cluster 1 DE genes
clust1_DE_genes <- rownames(res)[!is.na(res$padj) &
                                 res$padj < 0.05 &
                                 res$log2FoldChange > 1]
#Cluster 2 DE genes
cluster2_DE_genes <- rownames(res)[!is.na(res$padj) &
                                 res$padj < 0.05 &
                                 res$log2FoldChange < -1]

#Compare with Global DEGs

cluster1_global <- intersect(clust1_DE_genes, genes.significant)
cluster2_global <- intersect(cluster2_DE_genes, genes.significant)

# Extract normalized counts
normalized_counts <- counts(dds, normalized = TRUE)

# Subset for cluster1-specific genes
cluster1_counts <- normalized_counts[clust1_DE_genes, ]

# Subset for cluster2-specific genes
cluster2_counts <- normalized_counts[cluster2_DE_genes, ]
```

```
# MA plot
#plotMA(res, main = "MA Plot: Cluster 1 vs Cluster 2", alpha = 0.05)
#worked on one laptop but not in the overall. graph is included in the report.
#supplimental appendix figure 32
```

```
#Extract Cluster-Specific top genes


# Top 10 upregulated genes for Cluster 1
top_genes_cluster1 <- head(rownames(res[order(res$log2FoldChange, decreasing = TRUE), ]), 10)

# Top 10 upregulated genes for Cluster 2
top_genes_cluster2 <- head(rownames(res[order(res$log2FoldChange, decreasing = FALSE), ]), 10)

# Combine them into a single list of top genes
top_genes <- c(top_genes_cluster1, top_genes_cluster2)

# Get normalized counts for the top genes
normalized_counts <- counts(dds, normalized = TRUE)
heatmap_data <- normalized_counts[top_genes, ]


# Add cluster information as an annotation
```

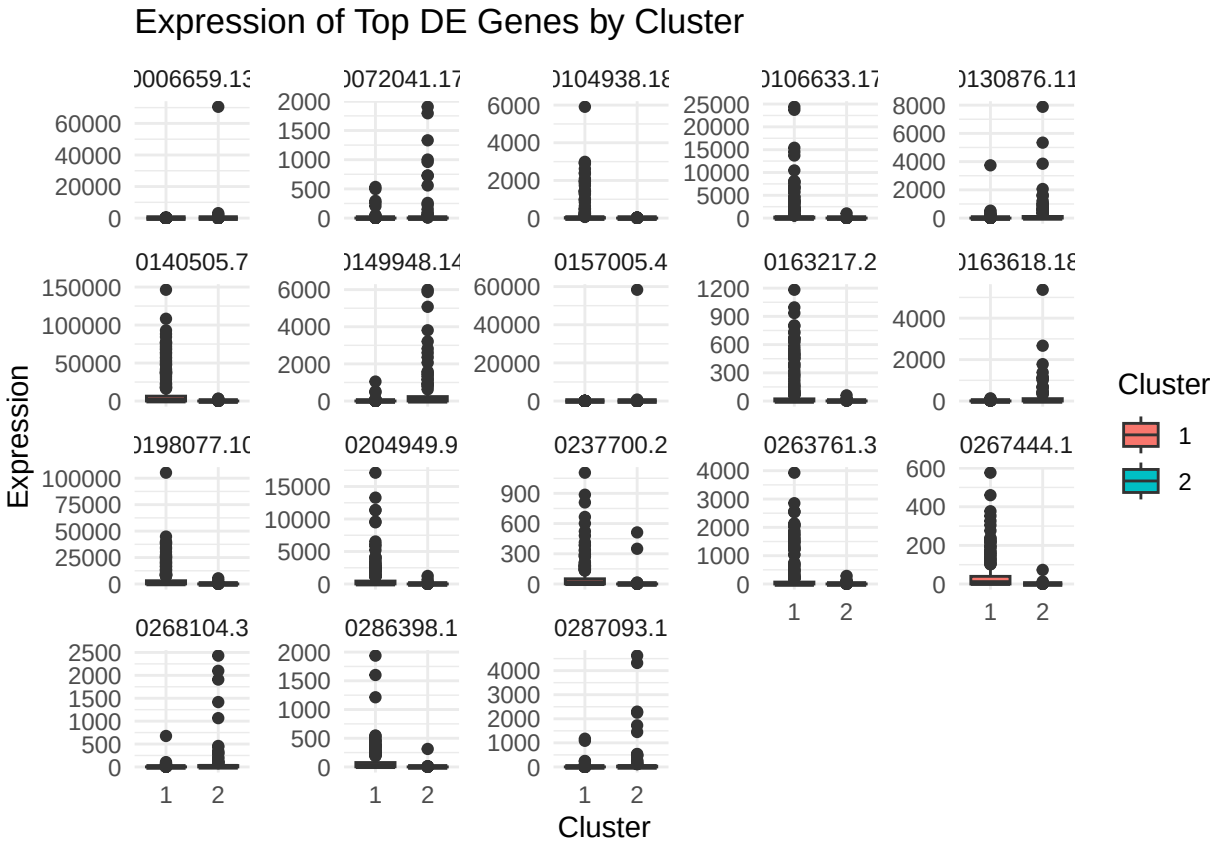
```r
annotation <- data.frame(Cluster = colData(dds)$Cluster)
rownames(annotation) <- colnames(heatmap_data)


# Convert normalized counts to a data frame with genes as rownames
heatmap_data_df <- as.data.frame(heatmap_data)
heatmap_data_df$Gene <- rownames(heatmap_data)

# Melt the data frame to long format
heatmap_data_long <- melt(heatmap_data_df, id.vars = "Gene", variable.name = "Sample", value.name = "Ex

# Add cluster information
heatmap_data_long$Cluster <- colData(dds)$Cluster[match(heatmap_data_long$Sample, colnames(normalized_c

# Convert Cluster to a factor
heatmap_data_long$Cluster <- as.factor(heatmap_data_long$Cluster)
```

```r
#retrieve only top expressed genes by cluster

# Calculate average expression per gene per cluster
top_genes <- heatmap_data_long %>%
  group_by(Gene, Cluster) %>%
  summarise(avg_expression = mean(Expression), .groups = "drop")

# Get top 10 expressed genes for each cluster
top_10_genes_per_cluster <- top_genes %>%
  group_by(Cluster) %>%
  top_n(10, avg_expression) %>%  # Select the top 10 genes per cluster
  ungroup()

# Filter the original data to keep only top expressed genes for each cluster
top_expression_data <- heatmap_data_long %>%
  filter(Gene %in% top_10_genes_per_cluster$Gene)


unique(top_expression_data$Gene)
```

```
##  [1] "ENSG00000106633.17" "ENSG00000267444.1"  "ENSG00000286398.1"
##  [4] "ENSG00000104938.18" "ENSG00000140505.7"  "ENSG00000237700.2"
##  [7] "ENSG00000204949.9"  "ENSG00000263761.3"  "ENSG00000198077.10"
## [10] "ENSG00000163217.2"  "ENSG00000006659.13" "ENSG00000287093.1"
## [13] "ENSG00000149948.14" "ENSG00000072041.17" "ENSG00000268104.3"
## [16] "ENSG00000157005.4"  "ENSG00000163618.18" "ENSG00000130876.11"
```

```r
#plot DE genes to compare between clusters

#So that the gene codes are clear in the plots,
# Modify the Gene column to keep only the relevant part by removing "ENSG0000"

# Create a new df with the modified gene names
top_expression_data_modified <- top_expression_data %>%
  mutate(Gene = gsub("^ENSG0000", "", Gene))
```

```
# Plot expression for each gene
ggplot(top_expression_data_modified, aes(x = Cluster, y = Expression, fill = Cluster)) +
  geom_boxplot() +
  facet_wrap(~ Gene, scales = "free_y") +
  theme_minimal() +
  ggtitle("Expression of Top DE Genes by Cluster")
```



#Pathway Analysis

```
library(gageData)

data(kegg.sets.hs)
data(sigmet.idx.hs)

# Focus on signaling and metabolic pathways only
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]


foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)
```

```
## [1]  0.27540494  1.73553991 -0.01283373 -0.02713267 -0.05811631 -0.09142681
```

```r
# Get the results
keggres = gage(foldchanges, gsets=kegg.sets.hs)


## Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways <- rownames(keggres$greater)[1:5]

# Extract the 8 character long IDs part of each string
keggresids = substr(keggrespathways, start=1, stop=8)
keggresids
```

```
## [1] "hsa00232" "hsa00983" "hsa00230" "hsa04514" "hsa04010"
```

```r
pathview(gene.data=foldchanges, pathway.id=keggresids, species="hsa")
```

```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00232.pathview.png

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00983.pathview.png

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00230.pathview.png

## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
```

```
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa04514.pathview.png

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa04010.pathview.png
```

```r
#KEGG data
data(kegg.sets.hs)
data(sigmet.idx.hs)
kegg.sets.hs <- kegg.sets.hs[sigmet.idx.hs]  # For signaling and metabolic pathways

#function to run pathway analysis for a cluster
run_pathway_analysis_top5 <- function(cluster_name, log2_fold_changes) {
  #GAGE pathway analysis
  keggres <- gage(log2_fold_changes, gsets = kegg.sets.hs)

  #top pathways (upregulated and downregulated)
  top_up_pathways <- head(keggres$greater[order(keggres$greater[, "q.val"]), ], 5)
  top_down_pathways <- head(keggres$less[order(keggres$less[, "q.val"]), ], 5)
```

```r
  #top 5 upregulated pathways
  if (!is.null(top_up_pathways) && nrow(top_up_pathways) > 0) {
    for (i in 1:min(5, nrow(top_up_pathways))) {
      top_pathway_id <- substr(rownames(top_up_pathways)[i], 1, 8)  #KEGG pathway ID
      message("Visualizing Upregulated Pathway ", i, ": ", rownames(top_up_pathways)[i])
      pathview(
        gene.data = log2_fold_changes,
        pathway.id = top_pathway_id,
        species = "hsa",
        out.suffix = paste0("Cluster_", cluster_name, "_Upregulated_", i)
      )
    }
  }

  #top 5 downregulated pathways
  if (!is.null(top_down_pathways) && nrow(top_down_pathways) > 0) {
    for (i in 1:min(5, nrow(top_down_pathways))) {
      top_pathway_id <- substr(rownames(top_down_pathways)[i], 1, 8)  #KEGG pathway ID
      message("Visualizing Downregulated Pathway ", i, ": ", rownames(top_down_pathways)[i])
      pathview(
        gene.data = log2_fold_changes,
        pathway.id = top_pathway_id,
        species = "hsa",
        out.suffix = paste0("Cluster_", cluster_name, "_Downregulated_", i)
      )
    }
  }
}

#subset the expression data for each cluster
cluster_1_data <- preprocessed_RNA[, colnames(preprocessed_RNA) %in% clust1_IDs]
cluster_2_data <- preprocessed_RNA[, colnames(preprocessed_RNA) %in% clust2_IDs]

#calculate log2 fold changes for each cluster
cluster_1_log2fc <- rowMeans(cluster_1_data) - rowMeans(preprocessed_RNA)
cluster_2_log2fc <- rowMeans(cluster_2_data) - rowMeans(preprocessed_RNA)

#perform pathway analysis
run_pathway_analysis_top5("Cluster_1", cluster_1_log2fc)
```

```
## Visualizing Upregulated Pathway 1: hsa00232 Caffeine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00232.Cluster_Cluster_1_Upregulated_1.png

## Visualizing Upregulated Pathway 2: hsa00983 Drug metabolism - other enzymes
```

```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00983.Cluster_Cluster_1_Upregulated_2.png

## Visualizing Upregulated Pathway 3: hsa00230 Purine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00230.Cluster_Cluster_1_Upregulated_3.png

## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)

## Visualizing Upregulated Pathway 4: hsa04514 Cell adhesion molecules (CAMs)

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa04514.Cluster_Cluster_1_Upregulated_4.png

## Visualizing Upregulated Pathway 5: hsa04010 MAPK signaling pathway

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa04010.Cluster_Cluster_1_Upregulated_5.png

## Visualizing Downregulated Pathway 1: hsa00232 Caffeine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00232.Cluster_Cluster_1_Downregulated_1.png

## Visualizing Downregulated Pathway 2: hsa00983 Drug metabolism - other enzymes

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00983.Cluster_Cluster_1_Downregulated_2.png

## Visualizing Downregulated Pathway 3: hsa00230 Purine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00230.Cluster_Cluster_1_Downregulated_3.png
```

```
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
```

```
## Visualizing Downregulated Pathway 4: hsa04514 Cell adhesion molecules (CAMs)
```

```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3
```

```
## Info: Writing image file hsa04514.Cluster_Cluster_1_Downregulated_4.png
```

```
## Visualizing Downregulated Pathway 5: hsa04010 MAPK signaling pathway
```

```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3
```

```
## Info: Writing image file hsa04010.Cluster_Cluster_1_Downregulated_5.png
```

```
run_pathway_analysis_top5("Cluster_2", cluster_2_log2fc)
```

```
## Visualizing Upregulated Pathway 1: hsa00232 Caffeine metabolism
```

```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00232.Cluster_Cluster_2_Upregulated_1.png

## Visualizing Upregulated Pathway 2: hsa00983 Drug metabolism - other enzymes

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00983.Cluster_Cluster_2_Upregulated_2.png

## Visualizing Upregulated Pathway 3: hsa00230 Purine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00230.Cluster_Cluster_2_Upregulated_3.png

## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
```

```
## Visualizing Upregulated Pathway 4: hsa04514 Cell adhesion molecules (CAMs)

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa04514.Cluster_Cluster_2_Upregulated_4.png

## Visualizing Upregulated Pathway 5: hsa04010 MAPK signaling pathway

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa04010.Cluster_Cluster_2_Upregulated_5.png

## Visualizing Downregulated Pathway 1: hsa00232 Caffeine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00232.Cluster_Cluster_2_Downregulated_1.png

## Visualizing Downregulated Pathway 2: hsa00983 Drug metabolism - other enzymes

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG3

## Info: Writing image file hsa00983.Cluster_Cluster_2_Downregulated_2.png

## Visualizing Downregulated Pathway 3: hsa00230 Purine metabolism

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.
```

```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG[illegible]

## Info: Writing image file hsa00230.Cluster_Cluster_2_Downregulated_3.png

## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)

## Visualizing Downregulated Pathway 4: hsa04514 Cell adhesion molecules (CAMs)

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG[illegible]

## Info: Writing image file hsa04514.Cluster_Cluster_2_Downregulated_4.png

## Visualizing Downregulated Pathway 5: hsa04010 MAPK signaling pathway

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/jyoon/OneDrive - UBC/UBC Engineering/Year 3/Term 1/BMEG 310/BMEG[illegible]

## Info: Writing image file hsa04010.Cluster_Cluster_2_Downregulated_5.png
```