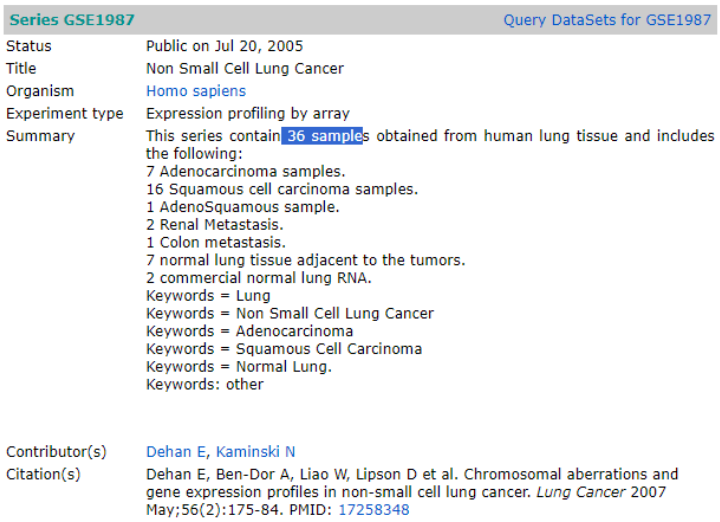
# Significance Tests and Clustering for GEO-based Omics Data

**(a)** We searched for the name Dehan on GEO, and as a result, we obtained information that the relevant data is in the GSE1987 dataset. (Figure 1).



| Series GSE1987 | Query DataSets for GSE1987 |
|---|---|
| Status | Public on Jul 20, 2005 |
| Title | Non Small Cell Lung Cancer |
| Organism | Homo sapiens |
| Experiment type | Expression profiling by array |
| Summary | This series contain 36 samples obtained from human lung tissue and includes the following:<br>7 Adenocarcinoma samples.<br>16 Squamous cell carcinoma samples.<br>1 AdenoSquamous sample.<br>2 Renal Metastasis.<br>1 Colon metastasis.<br>7 normal lung tissue adjacent to the tumors.<br>2 commercial normal lung RNA.<br>Keywords = Lung<br>Keywords = Non Small Cell Lung Cancer<br>Keywords = Adenocarcinoma<br>Keywords = Squamous Cell Carcinoma<br>Keywords = Normal Lung.<br>Keywords: other |
| Contributor(s) | Dehan E, Kaminski N |
| Citation(s) | Dehan E, Ben-Dor A, Liao W, Lipson D et al. Chromosomal aberrations and gene expression profiles in non-small cell lung cancer. *Lung Cancer* 2007 May;56(2):175-84. PMID: 17258348 |

**Figure 1. Visualize GSE Number (GSE1987).**

In continuation, after installing BiocManager, we specified the package we wanted to download. We called the GEOquery package to be able to use the getGEO function. Then, by entering the code GSE1987, we downloaded the relevant file. For the question A, since we were asked to remove the files related to Renal Metastasis and Colon Metastasis samples, we applied this in the final step and formatted the data as requested (Figure 2).

```r
#Under the codes we runned the command line in R.

#if (!require("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#BiocManager::install()


BiocManager::install("GEOquery")  #GSE1987
library(GEOquery)

lung_c <- getGEO("GSE1987", AnnotGPL = TRUE)

lung_c1 <- lung_c[[1]]

ExpInfo = lung_c1@phenoData@data
ExpInfo$description
#We would learn the renal and colon metastasis
#remove the 2 Renal Metastasis 1 Colon metastasis
#[16] "Renal Metastasis. Male."
#[18] "Colon Metastasis. Female."
#[20] "Renal Metastasis. Male."

gse_lung <- lung_c1@assayData$exprs

columns_to_remove <- c(2, 4, 6)

gse_lung <- gse_lung[, -columns_to_remove]
```

**Figure 2. Codes of question "a"**

**(b)** At this stage, we removed genes that satisfy the condition of having zero standard deviation because their expression values are all the same. We applied the same modification to the featureData named feature_lung since we visualize gene names here, and we will need gene name information. Subsequently, we used the log2 function to take the logarithm of our data, which we will use in the t-test. Using the t.test function, we calculated the p-values for our tumor and normal cells. As a result, we displayed the number of genes with p-values less than 0.01 as 1534 (Figure 3).

```r
#sd=0 genes, we removed the featureData
feature_lung <- lung_c1@featureData
feature_lung <- feature_lung[-zero_sd_genes, ]

log_gse_lung_filtered <- log2(gse_lung_filtered)
head(log_gse_lung_filtered)

#t-test
p_value = NULL
for (i in 1:nrow(log_gse_lung_filtered)) {
  p_value[i] <- t.test(log_gse_lung_filtered[i, c(1:24, 34)], log_gse_lung_filtered[i, 25:33])$p.value
}

p.val1 <- length(which(p_value<0.01))
p.val1 #output 1534
```

**Figure 3. Codes of question "b".**

**(c)** At this stage, we applied the Benjamini-Hochberg correction using the p-values obtained in the previous question. After this correction, we identified the number of genes with p-values

less than 0.05 as our reference, considering them to be significant. Consequently, we learned that 1063 genes are significant based on the 0.05 cut-off value (Figure 4).

```
#question c

#Benjamini-Hochberg correction
corrected_p_values <- p.adjust(p_value, method = "BH")

# Find significantly changed genes using a corrected p-value cutoff of 0.05
significant_genes <- which(corrected_p_values < 0.05)

sig_genes <- length(significant_genes)
sig_genes #1063
```

**Figure 3. Codes of question "c".**

**(d)** No, I checked, and the results are not exactly the same, but they are close. I think that the reason for this may be that during this study we removed genes with a standard deviation of 0 from the data set.

**(e)** (Hint:use cor.testfunction). Repeat the same analysis with Spearman correlation. Do you see any difference in the pattern? Discuss the results.

We first identify the top 3 genes based on their p-values, and then apply Pearson correlation tests for these genes in both normal and tumor conditions. We chose to implement the tests within a loop instead of printing each of the 3 genes separately.

```r
#question e
#3 genes-most significant p-values, Find the indices of the smallest three values
most_3 <- order(corrected_p_values)[1:3]
print(most_3) #4346-1183-1922

a <- feature_lung@data$`Gene symbol`
a[most_3] #most common 3 genes; "SPP1"  "SPP1"  "DDX11"

#Select columns for tumor (1:24 and 34) and normal tissue (25:33)

#CORRELATION TEST
top3_genes <- log_gse_lung_filtered[most_3, ]

c_top3_genes <- data.frame(t(top3_genes[, c(1:24, 34)]))
colnames(c_top3_genes) <- rownames(top3_genes)
n_top3_genes <- data.frame(t(top3_genes[, 25:33]))
colnames(n_top3_genes) <- rownames(top3_genes)

# Cancer samples correlation tests
p_values_cancer <- matrix(NA, nrow = 3, ncol = 3)

for (i in 1:3) {
  for (j in 1:3) {
    p_values_cancer[i, j] <- cor.test(c_top3_genes[, i], c_top3_genes[, j], method = 'pearson')$p.value
  }
}

# Normal samples correlation tests
p_values_normal <- matrix(NA, nrow = 3, ncol = 3)

for (i in 1:3) {
  for (j in 1:3) {
    p_values_normal[i, j] <- cor.test(n_top3_genes[, i], n_top3_genes[, j], method = 'pearson')$p.value
  }
}

# Print the results asthe pvalues-cancer and pvalues-normal
print("Cancer Samples Correlation P-Values:")
print(p_values_cancer)
print("Normal Samples Correlation P-Values:")
print(p_values_normal)
```

**Figure 4. Codes of question “e”.**

In the context of the results obtained for cancerous cells, given that the p-values are very small, we can state that these relationships are statistically significant. Similarly, when looking at the p-values for normal cells, the results are also highly significant (Figure 5).

```
[1] "Cancer Samples Correlation P-Values:"
> print(p_values_cancer)
              [,1]         [,2]          [,3]
[1,] 1.586425e-181 0.0002387822  7.951017e-02
[2,]  2.387822e-04 0.0000000000  8.309627e-02
[3,]  7.951017e-02 0.0830962666 1.586425e-181
> print("Normal Samples Correlation P-Values:")
[1] "Normal Samples Correlation P-Values:"
> print(p_values_normal)
             [,1]         [,2]         [,3]
[1,] 0.000000e+00 6.603003e-05 1.837054e-01
[2,] 6.603003e-05 0.000000e+00 1.107678e-01
[3,] 1.837054e-01 1.107678e-01 2.220226e-54
```

**Figure 5. Results of Pearson Correlation Most Significat 3 Genes.**

At this stage, we applied a Spearman correlation similar to what we did in Pearson correlation. Similarly, we have used loops here as well (Figure 6).

```r
#Cancer samples Spearman correlation tests
p_values_cancer_spearman <- matrix(NA, nrow = 3, ncol = 3)

for (i in 1:3) {
  for (j in 1:3) {
    p_values_cancer_spearman[i, j] <- cor.test(c_top3_genes[, i], c_top3_genes[, j], method = 'spearman')$p.value
  }
}

# Normal samples Spearman correlation tests
p_values_normal_spearman <- matrix(NA, nrow = 3, ncol = 3)

for (i in 1:3) {
  for (j in 1:3) {
    p_values_normal_spearman[i, j] <- cor.test(n_top3_genes[, i], n_top3_genes[, j], method = 'spearman')$p.value
  }
}

# Print the results for Spearman correlation
print("Cancer Samples Spearman Correlation P-Values:")
print(p_values_cancer_spearman)
print("Normal Samples Spearman Correlation P-Values:")
print(p_values_normal_spearman)
```

**Figure 6. Spearman Correlation for Normal and Tumor Cells.**

In the context of the results as the spearman correlation, obtained for cancerous cells, given that the p-values are very small, we can state that these relationships are statistically significant. Similarly, when looking at the p-values for normal cells, the results are also highly significant (Figure 7).

```
[1] "Cancer Samples Spearman Correlation P-Values:"
> print(p_values_cancer_spearman)
              [,1]         [,2]         [,3]
[1,] 0.0000000000 0.0003271861 1.582656e-02
[2,] 0.0003271861 0.0000000000 3.286291e-02
[3,] 0.0158265591 0.0328629064 3.195672e-07
> print("Normal Samples Spearman Correlation P-Values:")
[1] "Normal Samples Spearman Correlation P-Values:"
> print(p_values_normal_spearman)
              [,1]         [,2]         [,3]
[1,] 5.511464e-06 3.112324e-02 7.435406e-01
[2,] 3.112324e-02 5.511464e-06 5.888999e-02
[3,] 7.435406e-01 5.888999e-02 5.511464e-06
```

**Figure 7. Result of Spearman Correlation for Normal and Tumor Cells.**

Spearman and Pearson correlation tests are two distinct statistical methods that assess the relationship between two variables, measuring different types of associations. According to the results, the Spearman correlation test has produced lower p-values than the Pearson test. This indicates the presence of a non-linear relationship between the variables. In this case, the relationship among the ordered versions of the data appears to be better captured by Spearman.

**(f)** In this context, we set the cut-off value to 0.01 and applied the Benjamini-Hochberg correction. Then we did the rerun query as the ensembl IDs.

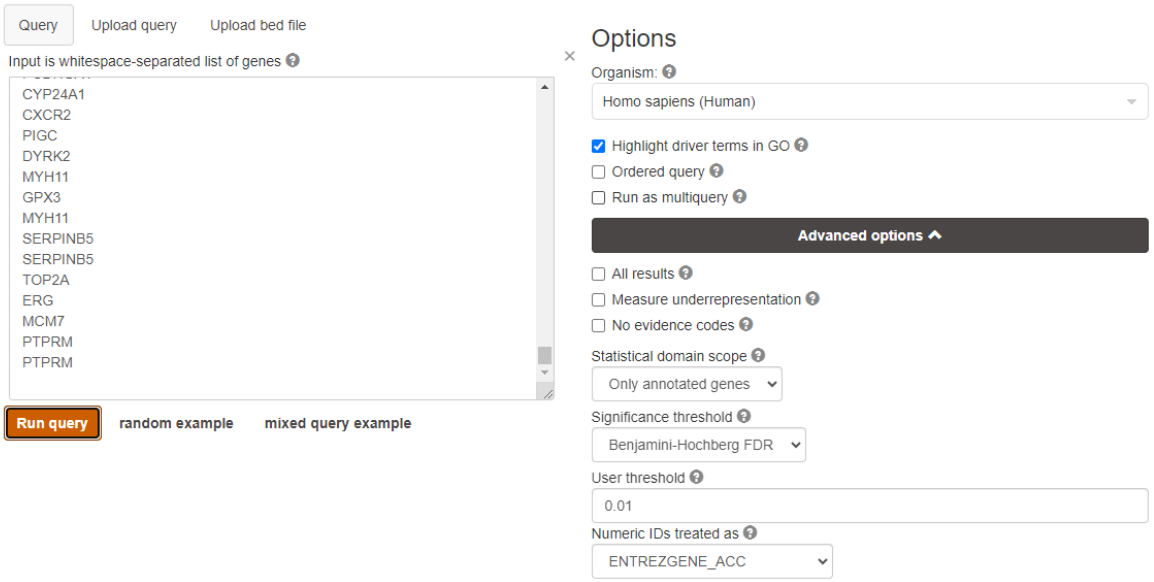



**Figure 8. Perform gProfiler for KEGG.**

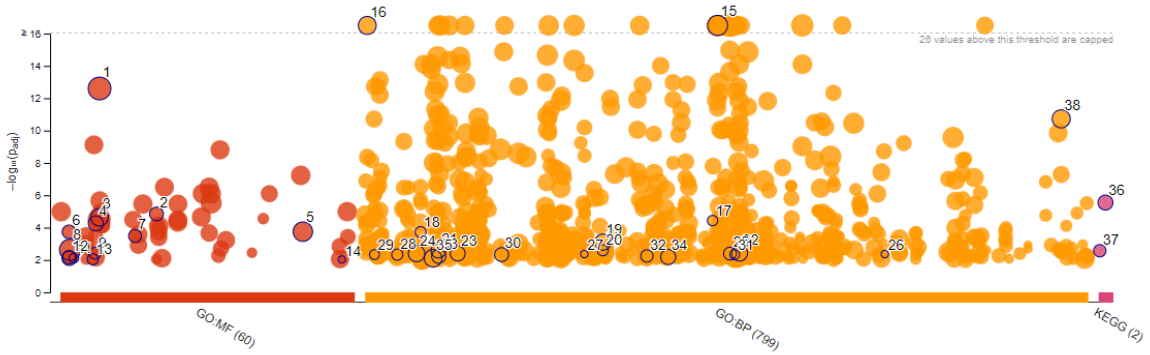



**Figure 9. Displayed GO-Molecular Function, GO-Biological Process, and KEGG(point of 36 and 37).** We interested in the point of 36 and 37.

| ID | Source | Term ID | Term Name | $p_{adj}$ (query_1) |
|---|---|---|---|---|
| 15 | GO:BP | GO:0048513 | animal organ development | $1.230\times10^{-26}$ |
| 16 | GO:BP | GO:0000278 | mitotic cell cycle | $4.843\times10^{-19}$ |
| 1 | GO:MF | GO:0005515 | protein binding | $2.495\times10^{-13}$ |
| 38 | GO:BP | GO:2000145 | regulation of cell motility | $1.878\times10^{-11}$ |
| 36 | KEGG | KEGG:04110 | Cell cycle | $2.823\times10^{-6}$ |
| 2 | GO:MF | GO:0019199 | transmembrane receptor protein kinase activity | $1.434\times10^{-5}$ |
| 3 | GO:MF | GO:0005509 | calcium ion binding | $2.279\times10^{-5}$ |
| 17 | GO:BP | GO:0048251 | elastic fiber assembly | $3.672\times10^{-5}$ |
| 4 | GO:MF | GO:0005201 | extracellular matrix structural constituent | $5.466\times10^{-5}$ |
| 5 | GO:MF | GO:0098772 | molecular function regulator activity | $1.775\times10^{-4}$ |
| 6 | GO:MF | GO:0003684 | damaged DNA binding | $1.829\times10^{-4}$ |
| 18 | GO:BP | GO:0006271 | DNA strand elongation involved in DNA replication | $1.873\times10^{-4}$ |
| 7 | GO:MF | GO:0016538 | cyclin-dependent protein serine/threonine kinase ... | $3.264\times10^{-4}$ |
| 19 | GO:BP | GO:0034504 | protein localization to nucleus | $7.910\times10^{-4}$ |
| 8 | GO:MF | GO:0003824 | catalytic activity | $2.362\times10^{-3}$ |
| 20 | GO:BP | GO:0034501 | protein localization to kinetochore | $2.529\times10^{-3}$ |
| 37 | KEGG | KEGG:00350 | Tyrosine metabolism | $2.681\times10^{-3}$ |
| 22 | GO:BP | GO:0051258 | protein polymerization | $3.760\times10^{-3}$ |
| 9 | GO:MF | GO:0005160 | transforming growth factor beta receptor binding | $3.819\times10^{-3}$ |
| 23 | GO:BP | GO:0009411 | response to UV | $3.976\times10^{-3}$ |
| 24 | GO:BP | GO:0006066 | alcohol metabolic process | $3.994\times10^{-3}$ |
| 25 | GO:BP | GO:0050798 | activated T cell proliferation | $4.031\times10^{-3}$ |
| 26 | GO:BP | GO:0098886 | modification of dendritic spine | $4.312\times10^{-3}$ |
| 27 | GO:BP | GO:0033030 | negative regulation of neutrophil apoptotic process | $4.312\times10^{-3}$ |
| 28 | GO:BP | GO:0002693 | positive regulation of cellular extravasation | $4.654\times10^{-3}$ |
| 29 | GO:BP | GO:0001554 | luteolysis | $4.667\times10^{-3}$ |
| 30 | GO:BP | GO:0016101 | diterpenoid metabolic process | $4.667\times10^{-3}$ |

**Figure 10. Displayed GO-Molecular Function, GO-Biological Process, and KEGG Details.** Term ID, Term Name, and adjective p-value.

The question asks us to define KEGG pathways. According to the information obtained from gProfiler, genes with high expression levels and correlation, using a cut-off value of 0.01 as a reference, are significant in the Cell Cycle and Tyrosine Metabolism pathways.

It is associated with KEGG:00350 Tyrosine Metabolism. The genes identified here are: **AOC3, ADH1A, ADH1B, ADH1C, TYRP1, AOX1, MAOB,** and ***ALDH3B2***.

It is associated with KEGG04110 Cell Cycle. The genes identified here are: ***TGFB2, CDC6, CDKN1C, CCND3, CDK1, PCNA, CCNB1, DDX11, CCNB2, MCM2, BUB1B, ATR, MAD2L1, ORC5, ESPL1, MCM6, PTTG1, CCNE1, BUB1, BUB3, TTK,*** and ***MCM7***.

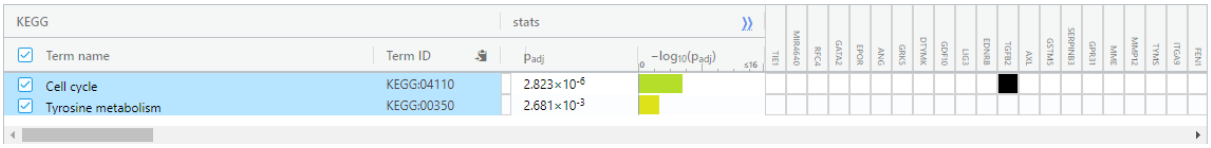



**Figure 11. KEGG Pathways, p-values and Gene Names.**

**(g)** At this stage, to obtain the requested graph, it is necessary to install and call the gplots package. Afterwards, to simplify the view, we defined the log_gse_lung_filtered dataset as 'data'. Subsequently, the as.dist function is used. The as.dist function is crucial for converting correlation matrices into a format suitable for hierarchical clustering algorithms. The resulting condensed distance vectors are later used in constructing dendrograms, and these dendrograms represent the hierarchical relationships between the rows and columns of the original data matrix (Figure 12).

```
#question g
# Load required libraries
install.packages(c("gplots", "RColorBrewer"))
library(gplots)
# Read the data
data <- log_gse_lung_filtered[significant_genes,]

# Perform hierarchical clustering using Spearman correlation
sg <- as.dist(1 - cor(data, method = "spearman"))
ss <- as.dist(1 - cor(t(data), method = "spearman"))

hc_sg <- hclust(sg, method = "complete")
hc_ss <- hclust(ss, method = "complete")

dend_sg <- as.dendrogram(hc_sg)
dend_ss <- as.dendrogram(hc_ss)

heatmap(data, Colv = dend_sg, Rowv = dend_ss, scale = "row")
```

**Figure 12. Perform Hierarchical Clustering Using Spearman Correlation on R.**

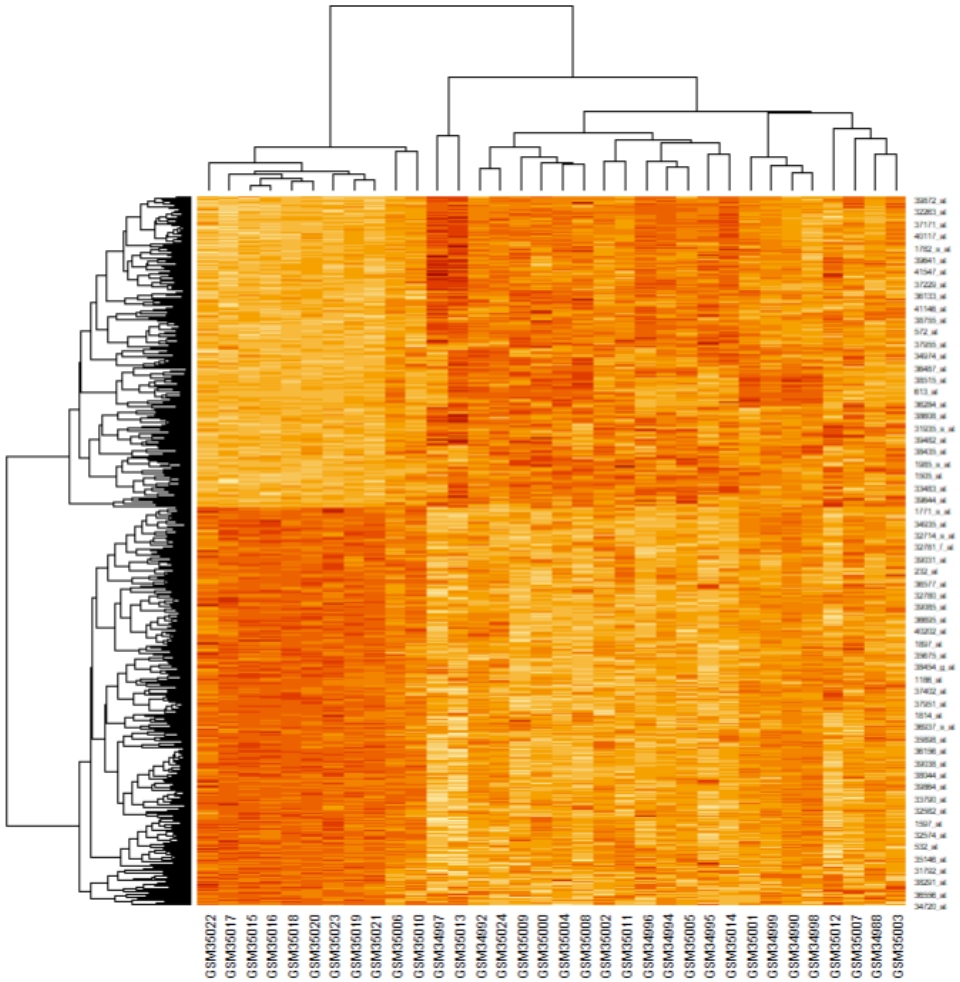



**Figure 13. Spearman Correlation on Heatmap.**

Bir önceki aşamada uygulama sırasında kullanılan scriptler method olarak pearson tercih edilerek uygulanmıştır. Pearson dışında kullanılan tüm parametreler spearman correleation için kullanılanlar ile aynıdır (Figure 14).

```
166 # Perform hierarchical clustering using pearson correlation
167 pg <- as.dist(1 - cor(data, method = "pearson"))
168 ps <- as.dist(1 - cor(t(data), method = "pearson"))
169
170 hc_pg <- hclust(pg, method = "complete")
171 hc_ps <- hclust(ps, method = "complete")
172
173 dend_pg <- as.dendrogram(hc_pg)
174 dend_ps <- as.dendrogram(hc_ps)
175
176 heatmap(data, Colv = dend_pg, Rowv = dend_ps, scale = "row")
```

**Figure 14. Perform Hierarchical Clustering Using Pearson Correlation On R.**

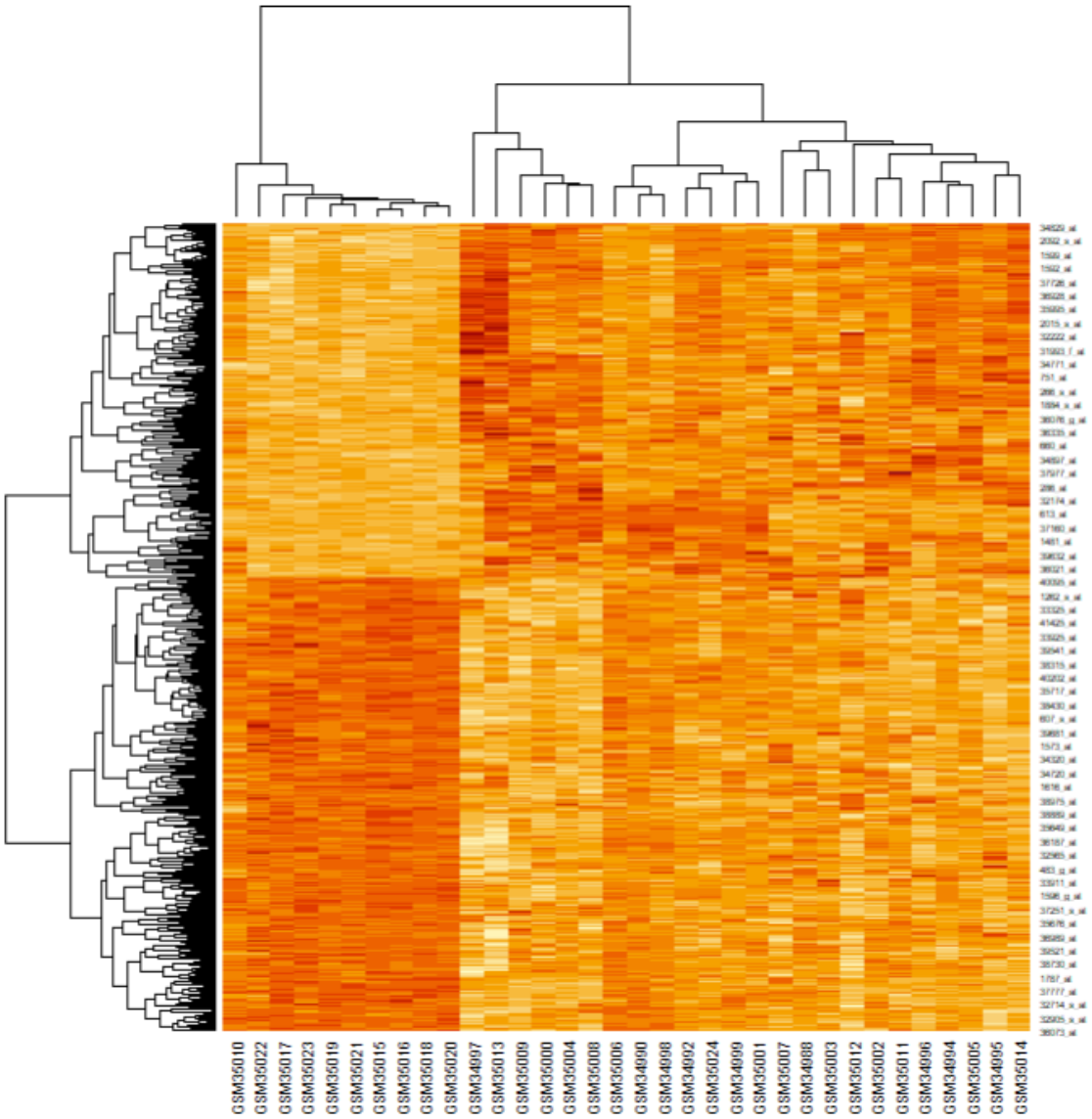



**Figure 15. Pearson Correlation on Heatmap.**

According to the Pearson and Spearman correlation heatmaps, while some genes show high correlation in Pearson, the same genes exhibit even higher correlation in Spearman. However, when comparing these two heatmaps overall, we observe that the clusters are focused on similar regions. We think that in the Spearman result, there are 8 clusters in the heatmap. On the other hand, in the Pearson result, we estimate that there are 7 clusters. We note that the high correlation regions observed in Pearson are more distinctly delineated on the color scale. This characteristic can be considered important for clearer separation of the groups, but it may be specific to our dataset. In our effort to determine the number of clusters, in addition to examining the color distribution on the heatmap, we considered how the upper dendrogram was constructed and how dendroid lengths varied(Figure 13 and Figure 15).