

# LAGH\_LAB36

Araceli Guerrero

9/3/2022

Modificado en práctica de Laboratorio 36 por Araceli Guerrero Herrera (ZuRy) :3

Laboratorio 36 - MAPA DE CALOR -TÉRMICO- with pheatmap

DATOS GENÉTICOS TOMADOS DE Sahir Bhatnagar

PRÁCTICA DE CODERS

Objetivo: Realizar un heatmap con datos genéticos

En este ejercicio vamos a:

1. Cargar nuestra matriz hipotética de datos y dataframes adicionales
2. Realizar varios heatmaps

Un mapa de calor es una representación gráfica de datos que utiliza un sistema de codificación de colores para representar diferentes valores

Heatmaps with pheatmap

Simulated data created by Sahir Bhatnagar.

possible data pre-processing - normalization - quantile, median, etc., log transform not necessary here - we have log fold change data that has already been normalized

Calculating your distance matrix (see dist objects): compute how similar or different you values are  
parametric - distance measures based on Pearson correlation non parametric - spearman rank - replace by ranks and calculate correlation, Kendall's - relative ordering euclidean - shortest distance between values (has to be normalized), takes magnitude into account city block/Manhattan - sum of distances along each dimension distance 1-correlation - of all pairs of items to be clustered

Cluster your samples (see hclust objects): hierarchical, organizes into a tree structure based on similarity - short branches if similar and longer branches as similarity decreases repeated cycles where the 2 closest remaining items (smallest distance) get joined by a branch with the length of the branch reflecting the distance between them, the distance between this item and all other remaining items are computed until only one object remains single linkage clustering - distance between 2 items is the minimum of all pairwise distances between items contained in x and y - fast b/c no other calculations need to be performed once you have your distance matrix complete linkage is the maximum of all pairwise distances between x and y average linkage - mean of all pairwise distances between items contained in x and y k-means organize into clusters (self-chosen number) - items are randomly assigned to a cluster - the mean vector for all items in each hcluster is computed, items are reassigned to the cluster whose center is closest to them - random starting points so will not always get the same answer, number of trial done to deal with the randomness self organizing maps

Paso 1. Instalar y ejecutar la paquetería de pheatmap.

install.packages("pheatmap")

```
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 3.5.3
```

## Paso 2. Importar datos (ubicar la ruta de los 3 archivos que se utilizarán).

```
file.choose()
```

```
## [1] "C:\\Users\\Araceli Guerrero\\Documents\\Laboratorios de R\\LAB36\\heatmap_data.csv"
```

```
file.choose()
```

```
## [1] "C:\\Users\\Araceli Guerrero\\Documents\\Laboratorios de R\\LAB36\\annotation_col.csv"
```

```
file.choose()
```

```
## [1] "C:\\Users\\Araceli Guerrero\\Documents\\Laboratorios de R\\LAB36\\annotation_row.csv"
```

## Paso 3. Copiar y pegar la ubicación del primer archivo para abrirlo como matriz.

```
genes <- as.matrix(  
  read.csv("C:\\Users\\Araceli Guerrero\\Documents\\Laboratorios de R\\LAB36\\heatmap_data.csv",  
    sep = ",",  
    header = T,  
    row.names = 1))
```

## Paso 4. Visualizar los primeros datos de la matriz genes.

```
head(genes[,1:5])
```

```
##      Patient1 Patient2 Patient3 Patient4 Patient5  
## Gene1 -0.1914190 -1.23415247 -0.04346178 0.14572108 -0.4017205  
## Gene2 -0.8978632 0.39549549 0.56691534 -0.59602679 -0.5403446  
## Gene3 1.0639177 2.79564490 -0.49031509 -0.64460585 0.8419138  
## Gene4 0.3524772 0.07475244 0.15824275 0.06590853 1.0301932  
## Gene5 -0.5016748 0.21616881 -0.86702910 2.05925069 -0.2852306  
## Gene6 0.8912122 0.19006278 0.45290320 0.22524625 -1.3451477
```

## Paso 5. Copiar y pegar la ubicación del archivo de las anotaciones de columnas para leerlo como dataframe.

```
annotation_col <- read.csv("C:\\Users\\Araceli Guerrero\\Documents\\Laboratorios de R\\LAB36\\annotation_col.csv",  
  header = T,  
  row.names = 1)
```

## Paso 6. Visualizar los primeros datos del dataframe annotation\_col.

```
head(annotation_col[,1:2])
```

```
##           Exposure   Type
## Patient1      X=0 T-cell
## Patient2      X=0 T-cell
## Patient3      X=0 B-cell
## Patient4      X=0 B-cell
## Patient5      X=0 B-cell
## Patient6      X=0 T-cell
```

Paso 7. Copiar y pegar la ubicación del archivo de las anotaciones de renglones para leerlo como dataframe.

```
annotation_row <- read.csv("C:\\Users\\Araceli Guerrero\\Documents\\Laboratorios d
e R\\LAB36\\annotation_row.csv",
                           header = T,
                           row.names = 1)
```

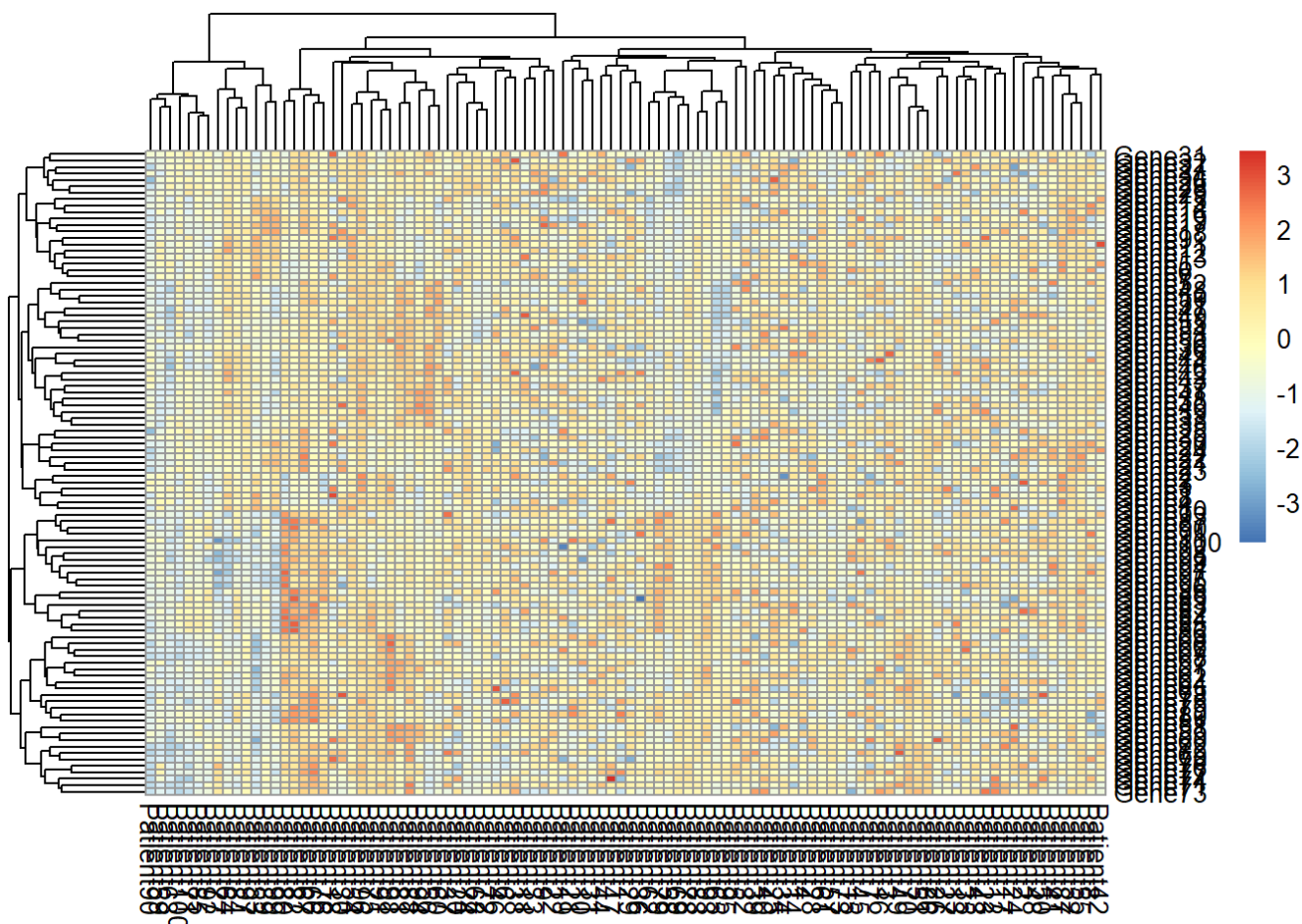
Paso 8. Visualizar los primeros datos del dataframe annotation\_row.

```
head(annotation_row[,1:1])
```

```
## [1] 1 1 1 1 1 1
```

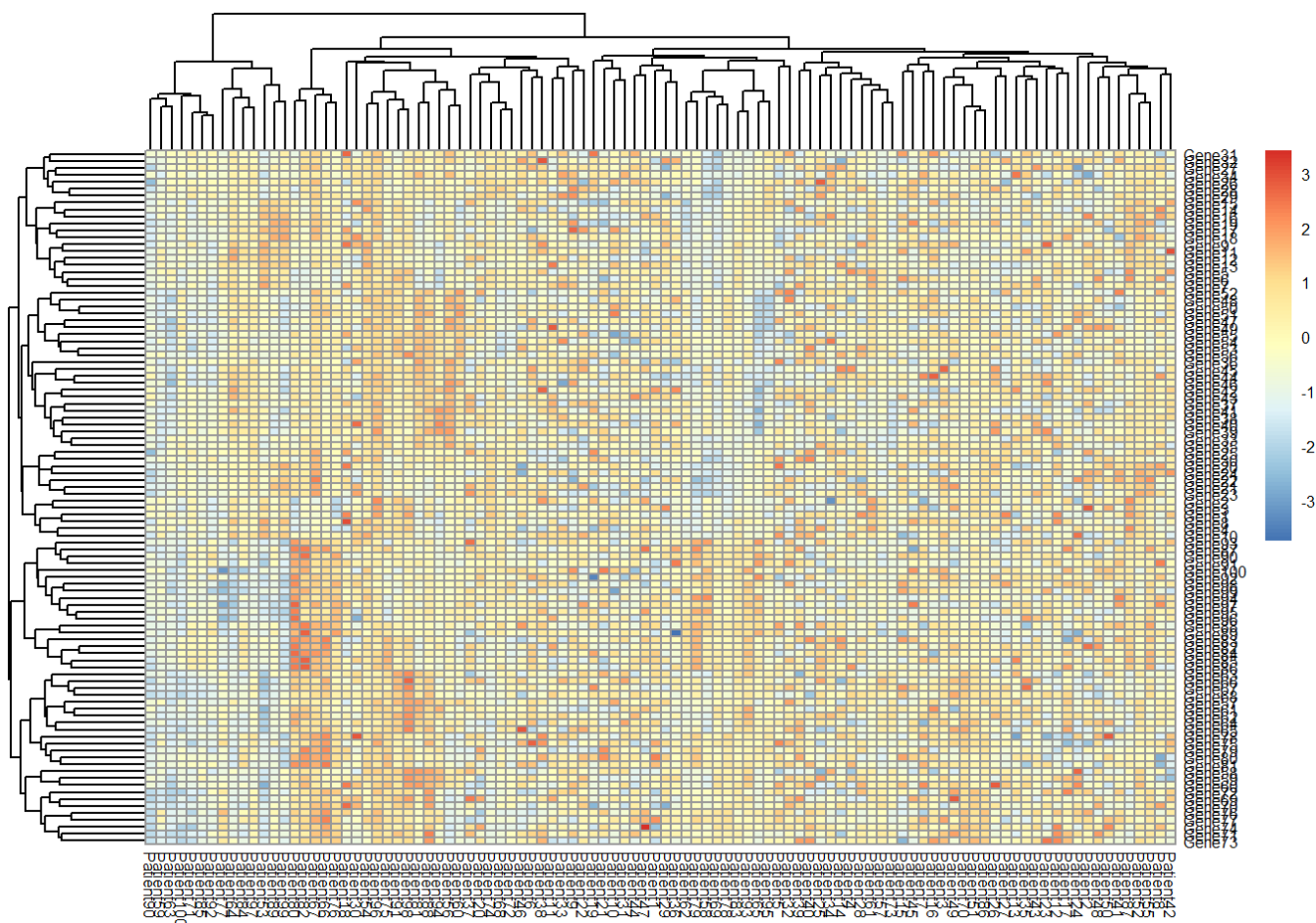
Paso 9. Dibujar el heatmap.

```
pheatmap(genes)
```



Paso 10. Cambiar el tamaño de la fuente (font).

```
pheatmap(genes, fontsize = 6)
```



Por default se dibujan los dendogramas (clustering) tanto en los renglones como en las columnas.

Clúster por genes (grupos de genes similares) se encuentra en los renglones.

Paso 11. Eliminar dendograma de las columnas (pacientes).

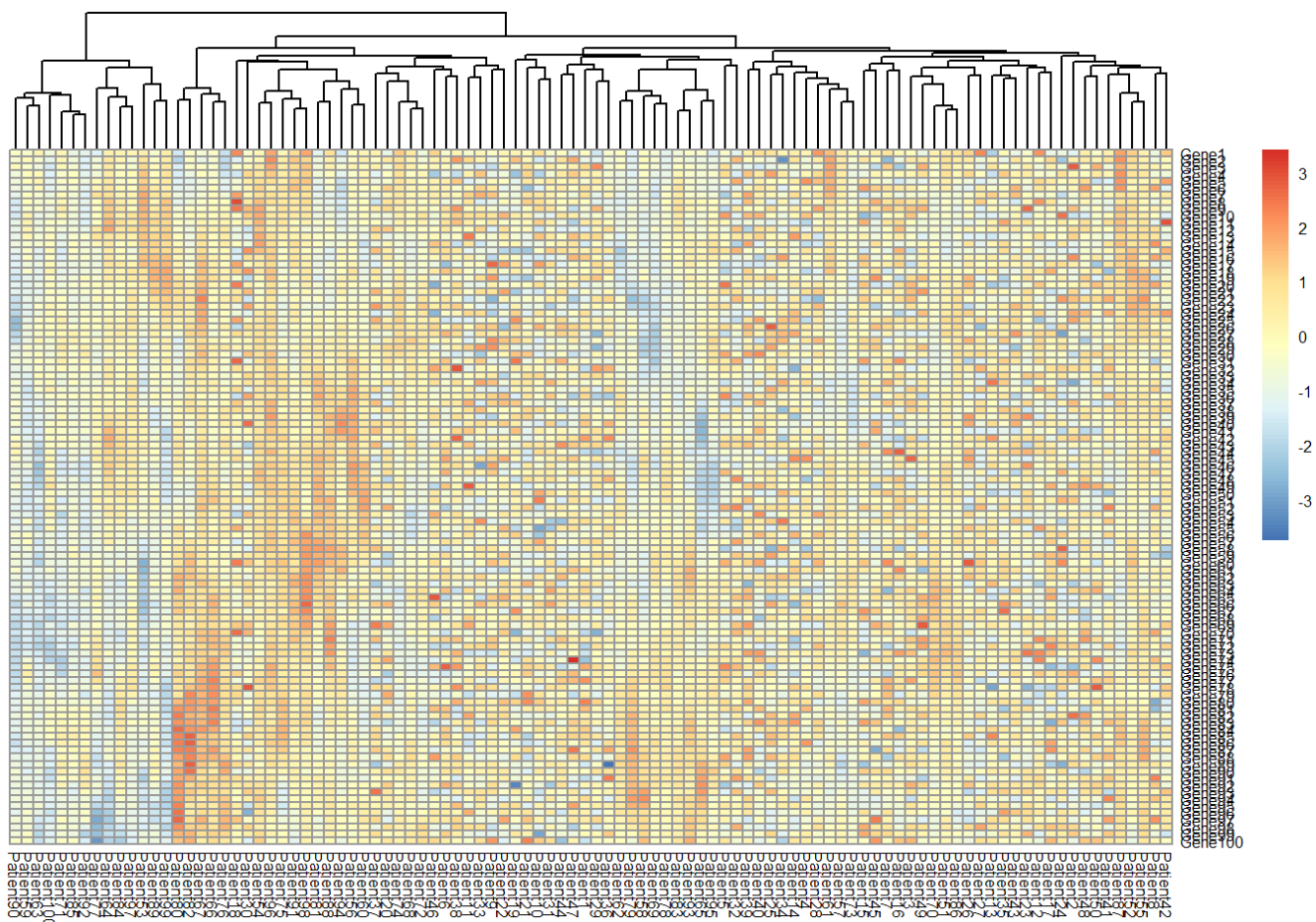
```
pheatmap(genes, fontsize = 6, cluster_rows = T, cluster_cols = F)
```



Clúster por pacientes (grupos de pacientes similares) se encuentra en las columnas.

Paso 12. Eliminar dendograma de los renglones (genes).

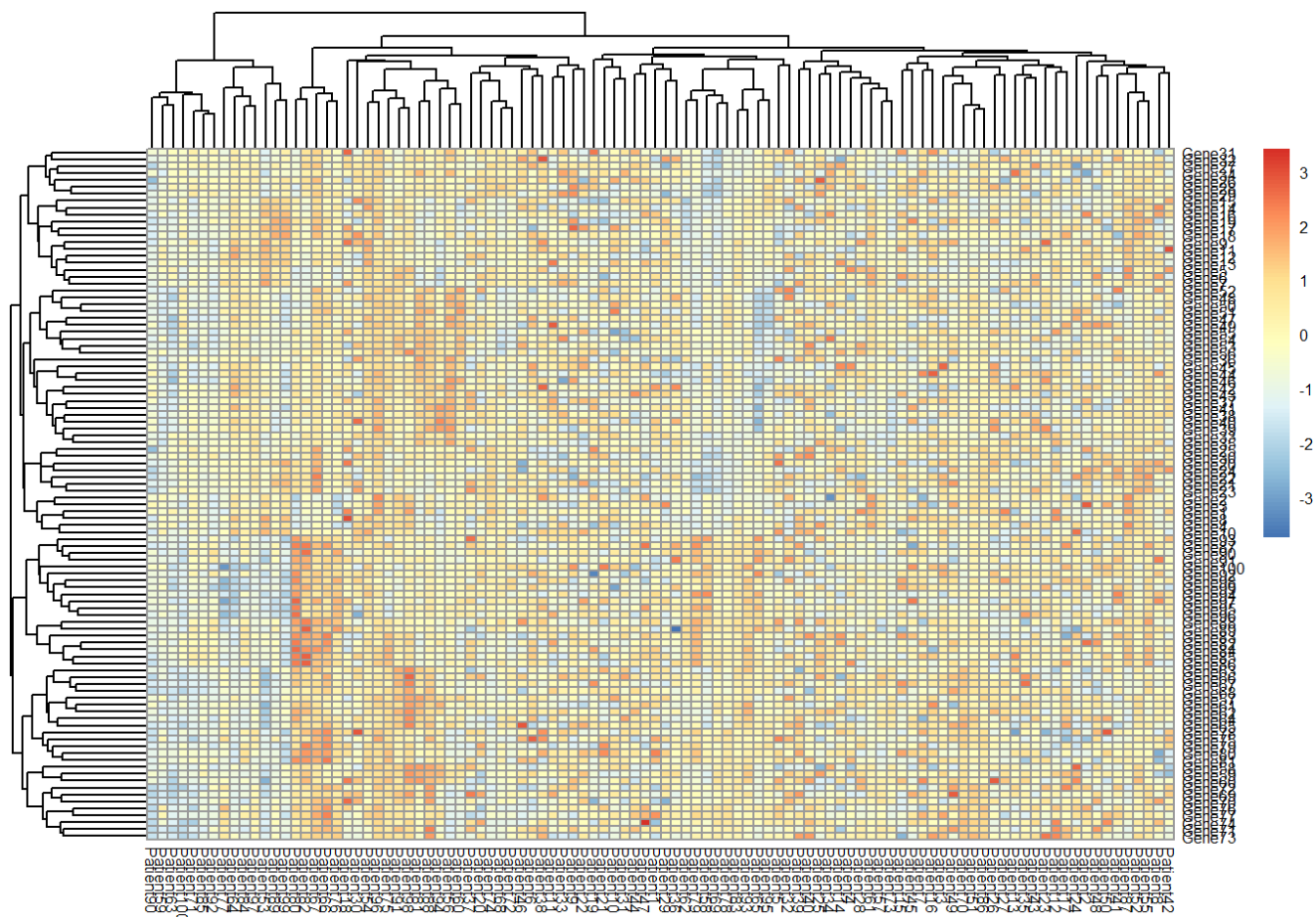
```
pheatmap(genes, fontsize = 6, cluster_rows = F, cluster_cols = T)
```



Paso 13. Visualizar ambos dendogramas.

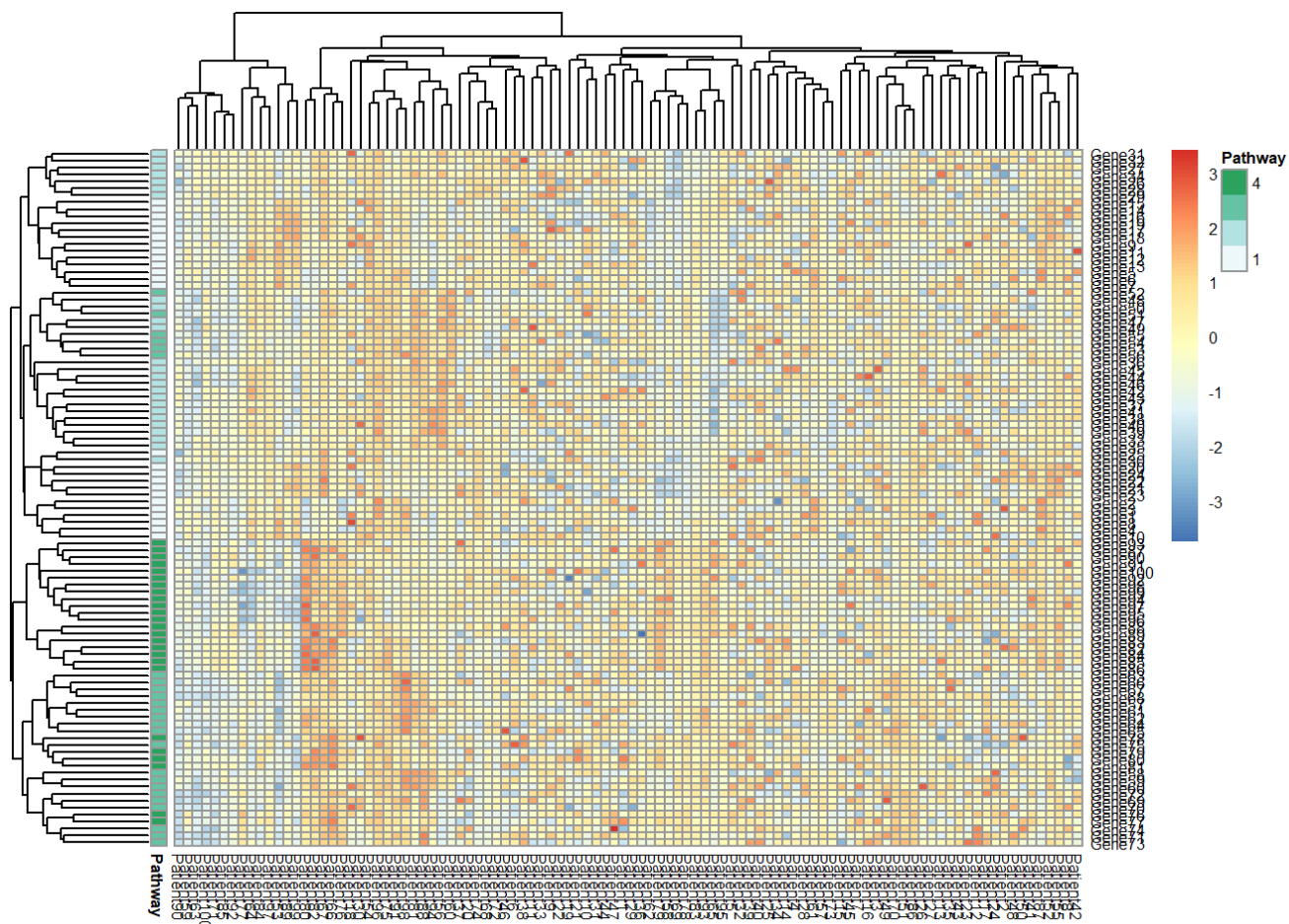
```
pheatmap(genes, fontsize = 6, cluster_rows = T, cluster_cols = T)
```





Paso 14. Identificar patrones subyacentes a las anotaciones de los renglones.

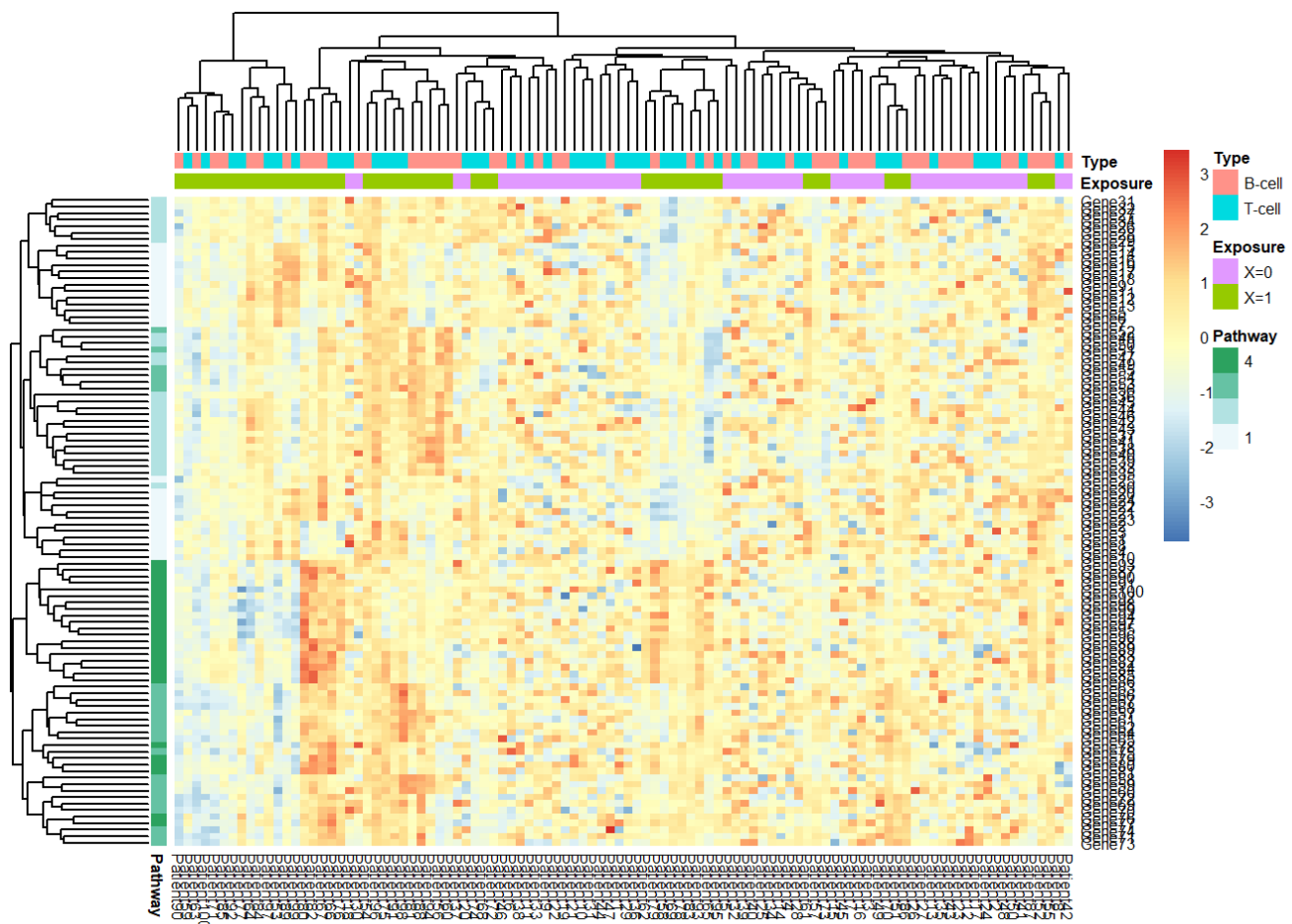
```
pheatmap(genes, fontsize = 6, cluster_rows = T, cluster_cols = T, annotation_row =
annotation_row)
```



Paso 15. Agregar las anotaciones de las columnas.

```
pheatmap(genes, fontsize = 6, cluster_rows = T, cluster_cols = T, annotation_row =
  annotation_row,
  annotation_col = annotation_col)
```



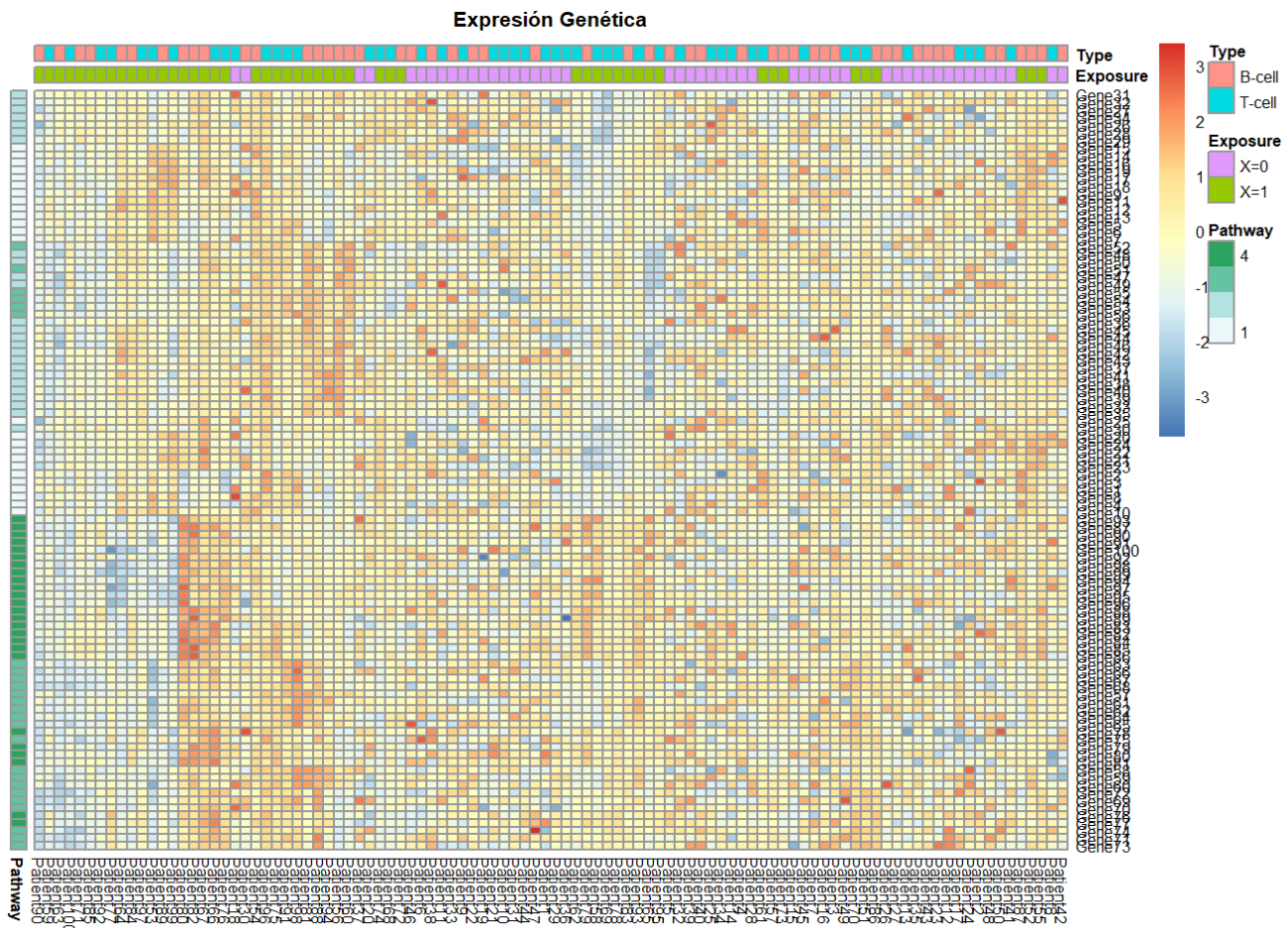


Los genes están en clúster de acuerdo a los patrones subyacentes a los que pertenecen.

Ahora se tiene información sobre el medicamento y la condición.

Paso 16. Generar el gráfico completo quitando clústers (árboles de agrupación o dendogramas).

```
pheatmap(genes, fontsize = 6, cluster_rows = T, cluster_cols = T, annotation_row =
annotation_row,
        annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0,
        main = "Expresión Genética")
```



Paso 17. Tomar una muestra de la matriz original para crear un subset.

```
sub <- genes [c(1:5, 55:60), c(1:5, 20:35, 55:60)]
```

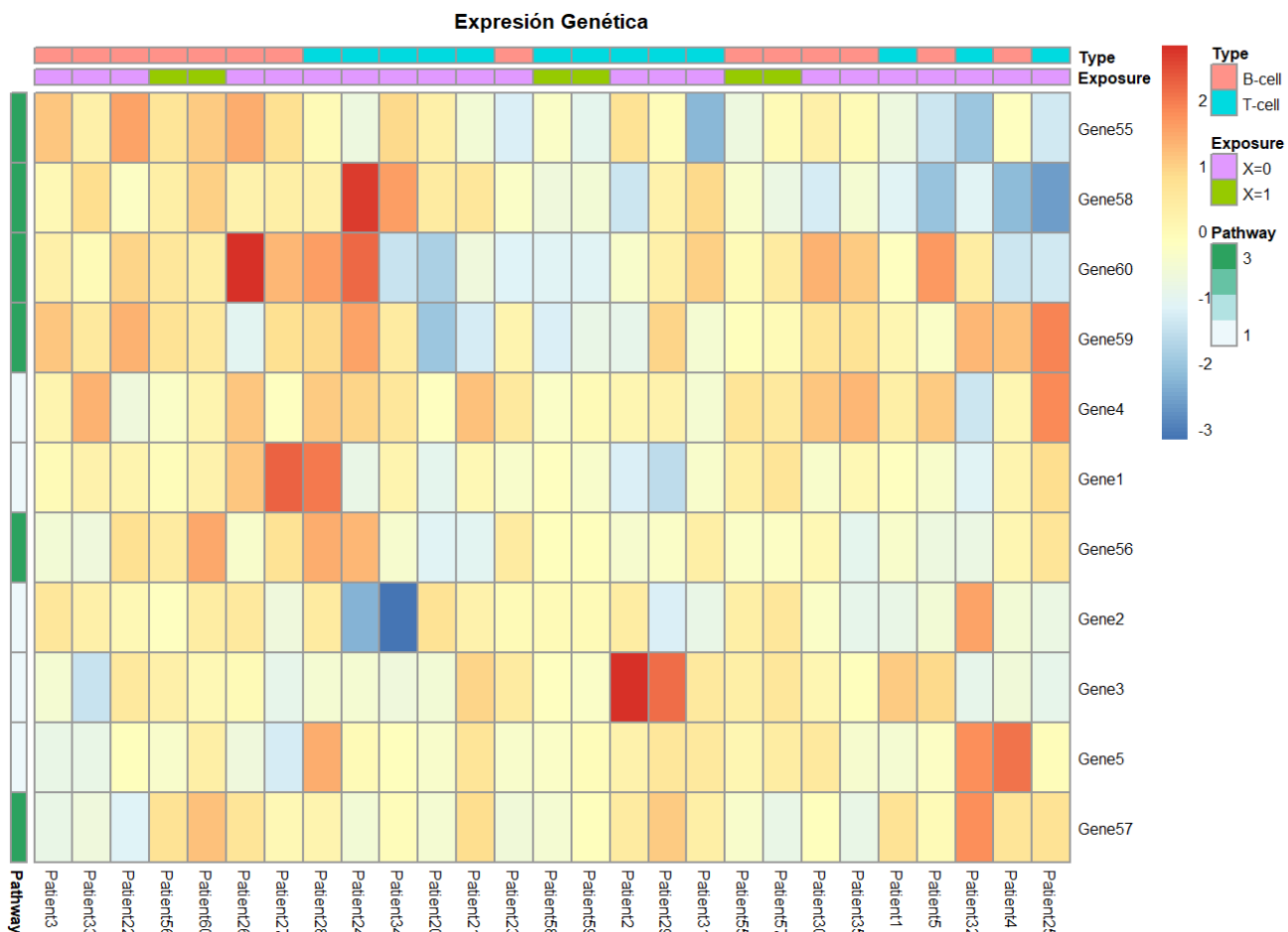
Paso 18. Visualizar los primeros datos del subset.

```
head(sub[, 1:5])
```

##	Patient1	Patient2	Patient3	Patient4	Patient5
## Gene1	-0.1914190	-1.23415247	-0.04346178	0.14572108	-0.4017205
## Gene2	-0.8978632	0.39549549	0.56691534	-0.59602679	-0.5403446
## Gene3	1.0639177	2.79564490	-0.49031509	-0.64460585	0.8419138
## Gene4	0.3524772	0.07475244	0.15824275	0.06590853	1.0301932
## Gene5	-0.5016748	0.21616881	-0.86702910	2.05925069	-0.2852306
## Gene55	-0.7221485	0.67533146	1.09262138	-0.21338367	-1.4104579

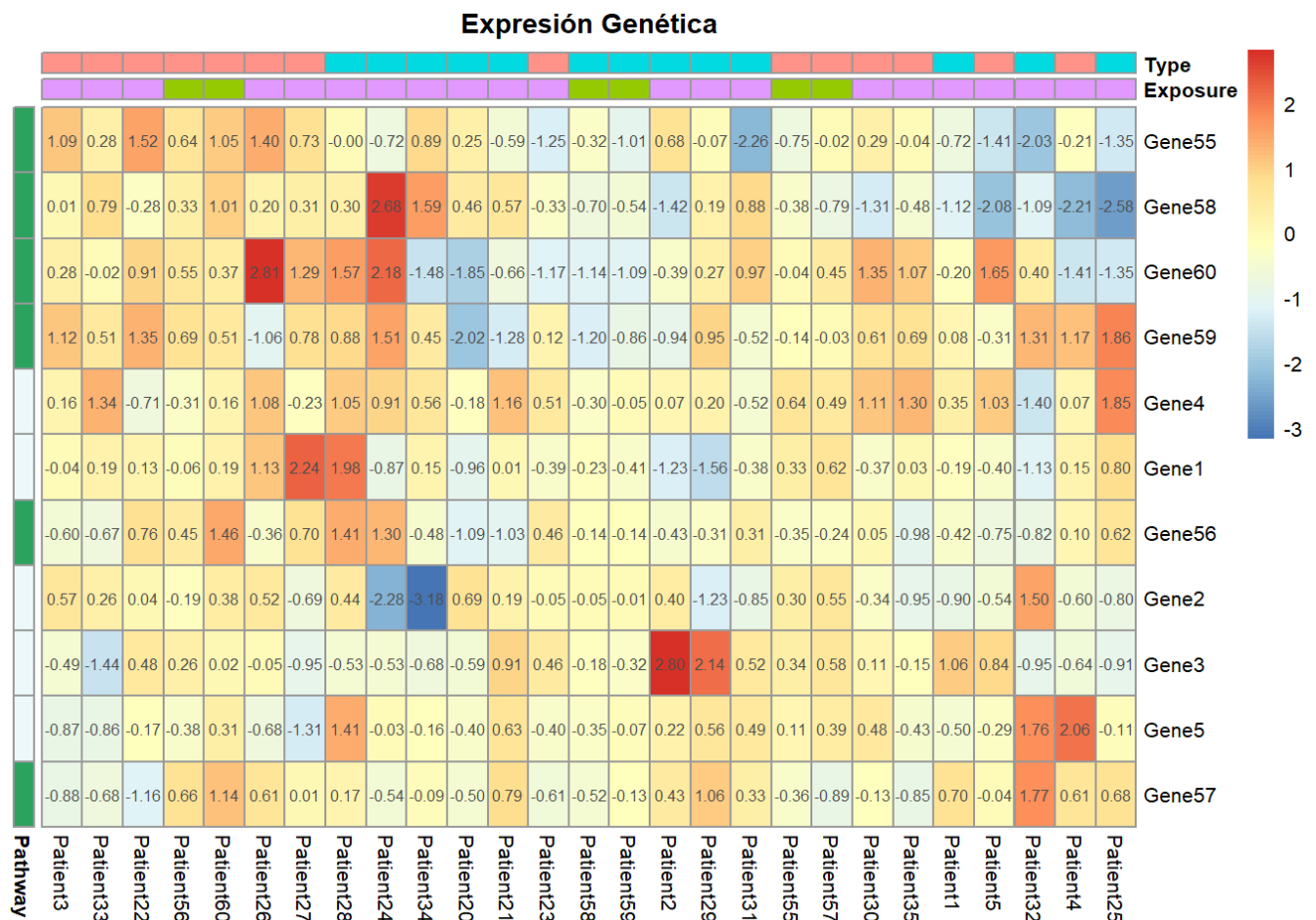
Paso 19. Generar el gráfico del subset.

```
pheatmap(sub, fontsize = 6, cluster_rows = T, cluster_cols = T, annotation_row = a
nnotation_row,
  annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0,
  main = "Expresión Genética")
```



Paso 20. Desplegar los valores del gráfico.

```
pheatmap(sub, cluster_rows = T, cluster_cols = T, annotation_row = annotation_row,
  annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0,
  main = "Expresión Genética", fontsize = 8, annotation_legend = FALSE, display_numbers = TRUE,
  fontsize_number = 6)
```



Paso 21. Instalar y ejecutar la paquetería de viridis para tener disponibles las paletas de colores magma, plasma, cividis e inferno.

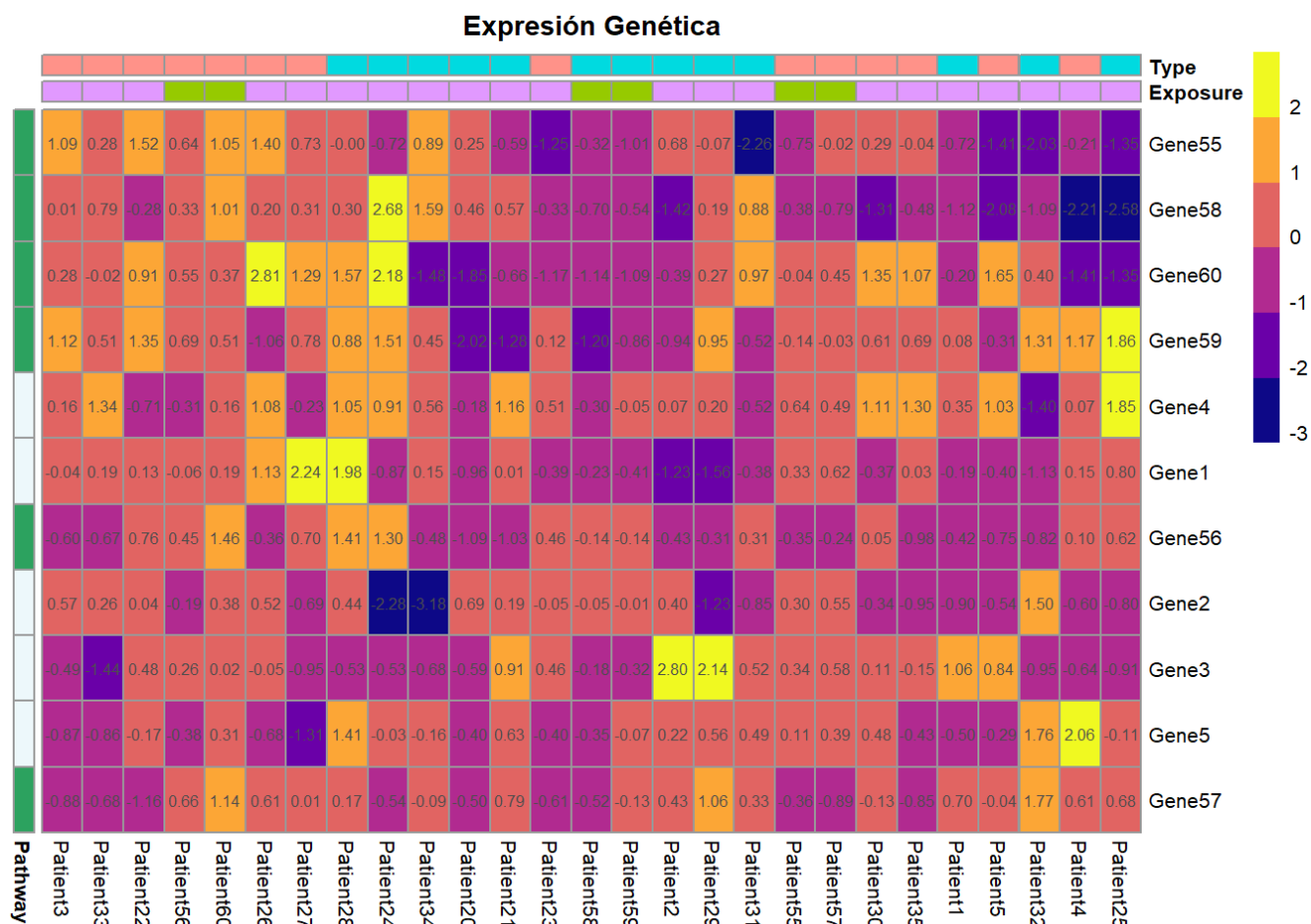
```
install.packages("viridis")
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

Paso 22. Añadir colores con las paletas cargadas.

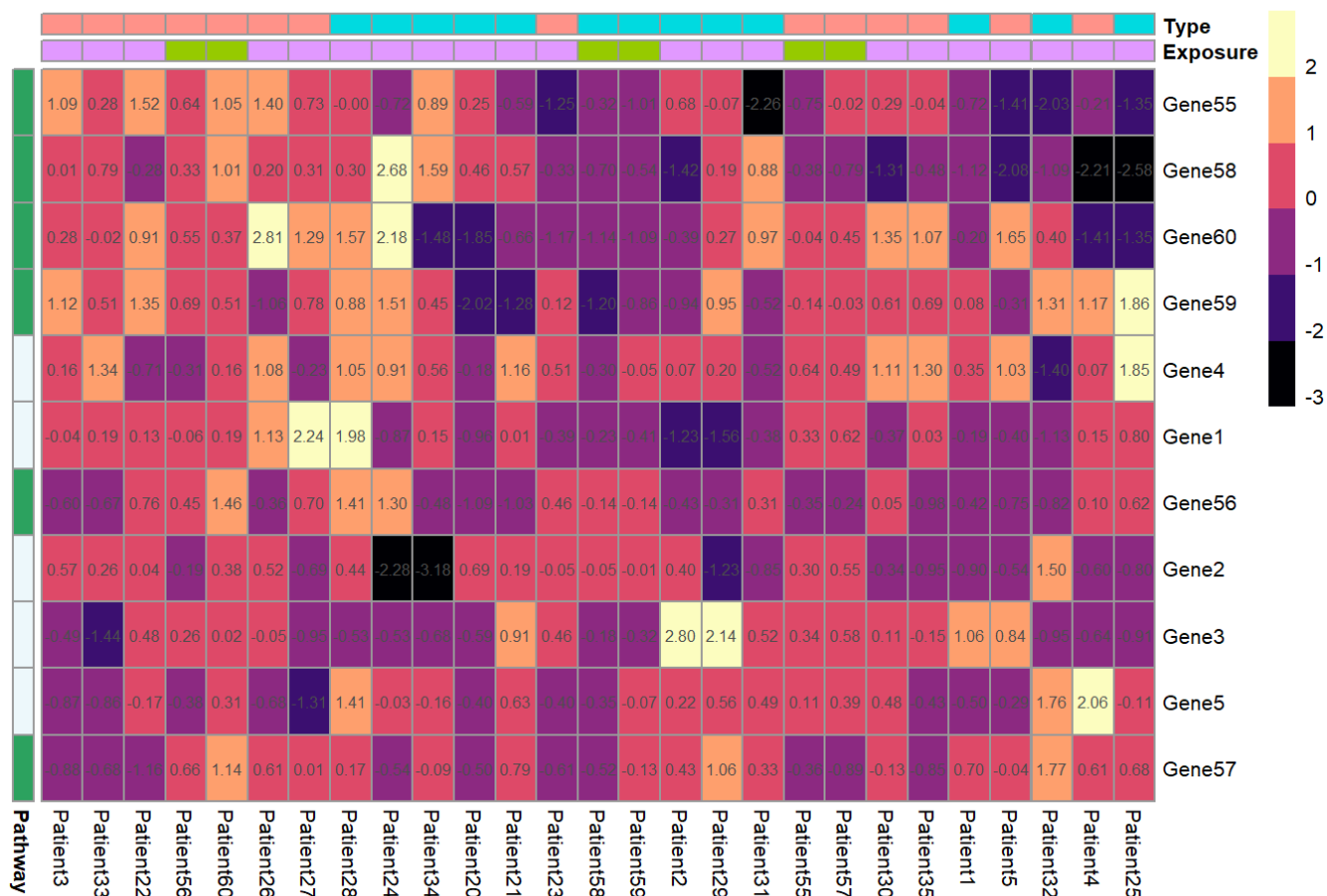
```
pheatmap(sub, cluster_rows = T, cluster_cols = T, annotation_row = annotation_row,
          annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0,
          main = "Expresión Genética", fontsize = 8, annotation_legend = FALSE, display_numbers = TRUE,
          fontsize_number = 6, col = viridis_pal(option = "plasma") (6))
```



Paso 23. Cambiar colores de la paleta.

```
pheatmap(sub, cluster_rows = T, cluster_cols = T, annotation_row = annotation_row,
          annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0,
          main = "Expresión Genética", fontsize = 8, annotation_legend = FALSE, display_numbers = TRUE,
          fontsize_number = 6, col = viridis_pal(option = "magma") (6))
```

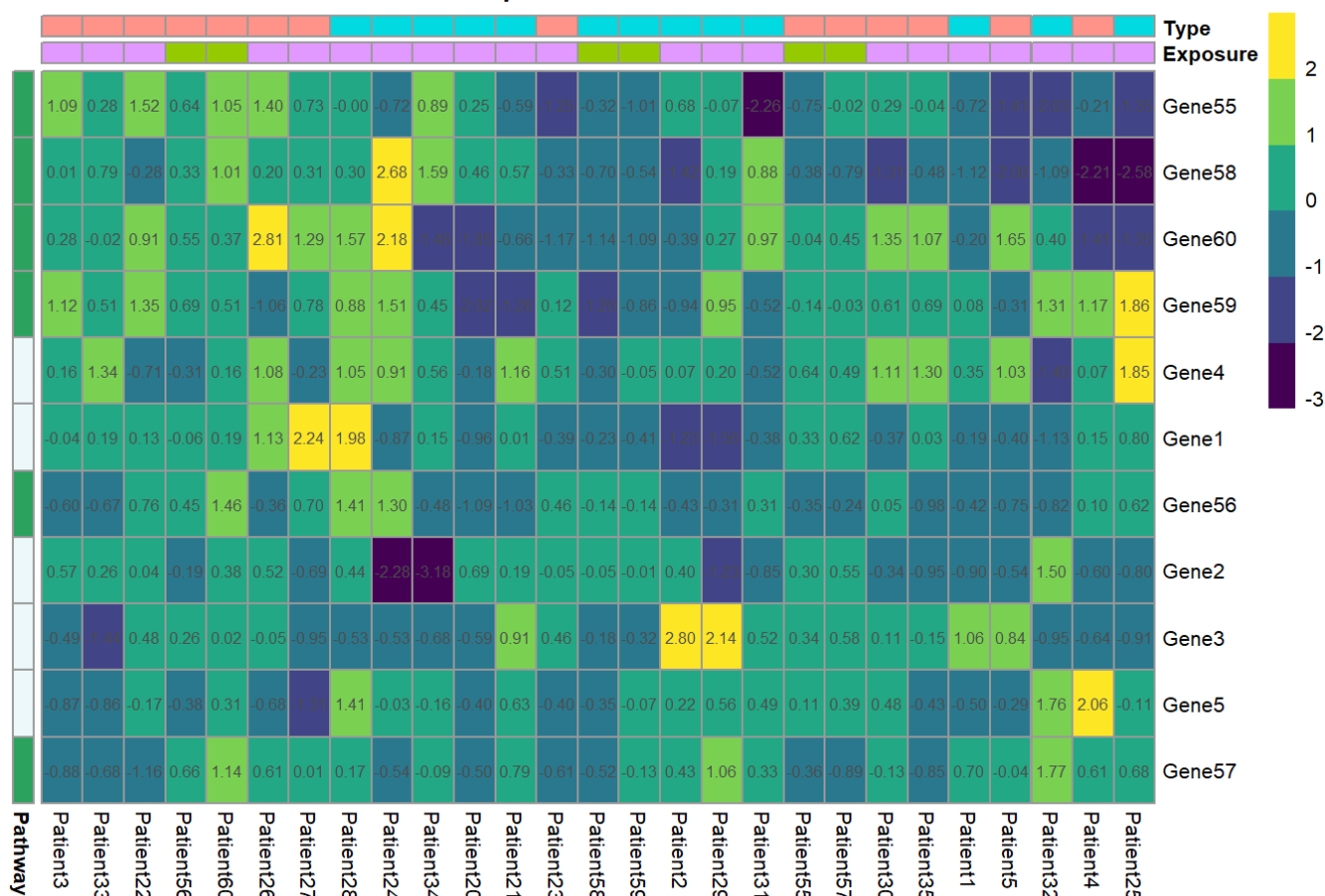
## Expresión Genética



```
pheatmap(sub, cluster_rows = T, cluster_cols = T, annotation_row = annotation_row,
  annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0,
  main = "Expresión Genética", fontsize = 8, annotation_legend = FALSE, display_numbers = TRUE,
  fontsize_number = 6, col = viridis_pal(option = "viridis") (6))
```



## Expresión Genética



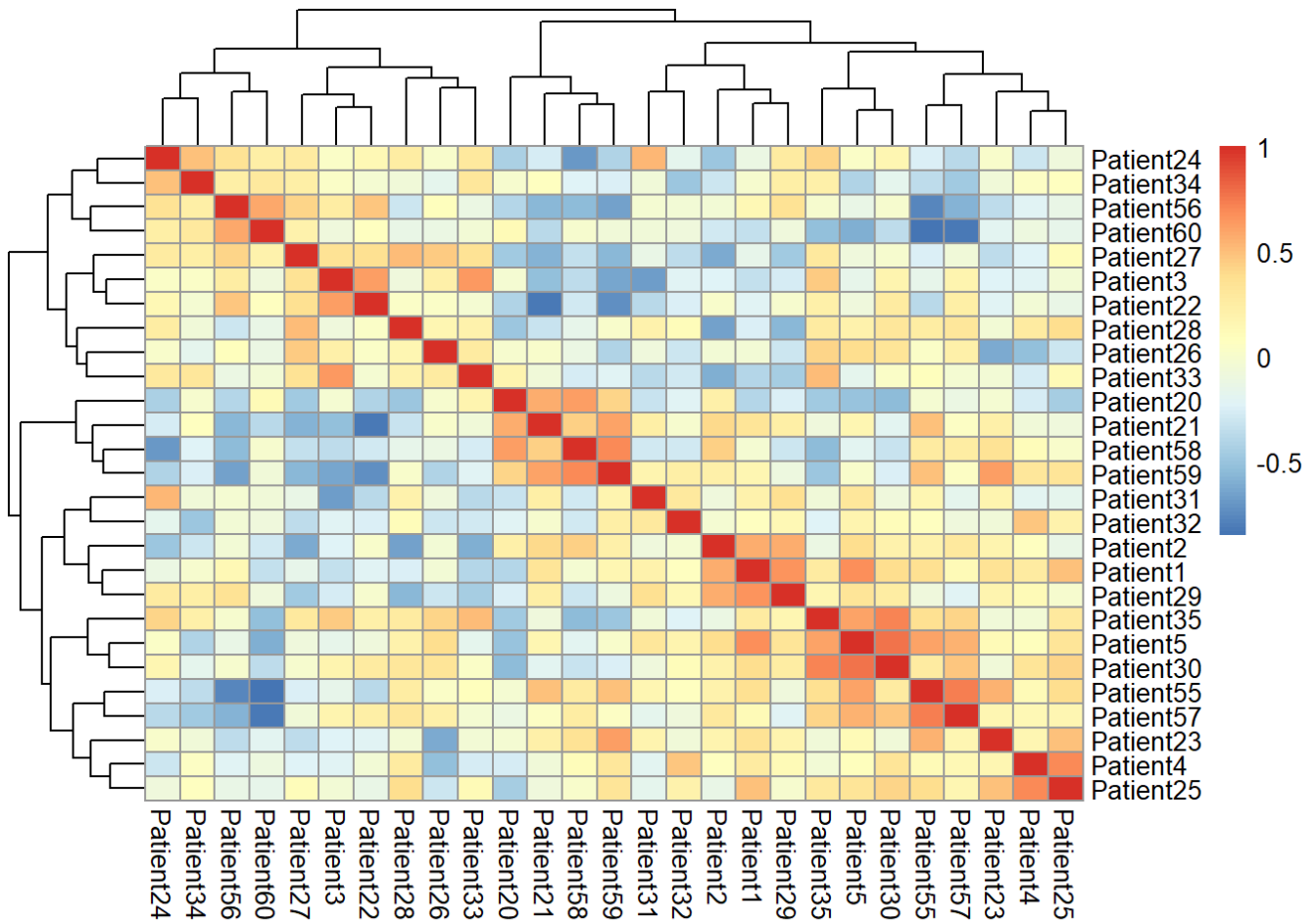
Paso 24. Identificar las distancias entre los genes.

```
dist(sub)
```

```
##          Gene1      Gene2      Gene3      Gene4      Gene5      Gene55      Gene56      Gene57
## Gene2    6.506125
## Gene3    7.823569  7.021725
## Gene4    5.253565  7.649124  6.516104
## Gene5    6.411847  5.977640  5.967513  6.184570
## Gene55   5.703940  6.969997  7.096321  6.837653  7.534618
## Gene56   4.544832  6.723925  6.542745  5.805165  5.150859  6.028094
## Gene57   6.124657  6.069362  5.550487  6.004035  3.881691  7.122986  5.209746
## Gene58   7.417422  8.796956  8.462521  7.874145  8.030439  6.777444  6.292359  7.669524
## Gene59   6.189649  8.293720  7.977707  6.115718  5.821355  7.317126  4.835770  6.104449
## Gene60   6.623226  8.133474  7.665999  6.837342  7.659167  7.569942  6.373711  7.296198
##          Gene58      Gene59
## Gene2
## Gene3
## Gene4
## Gene5
## Gene55
## Gene56
## Gene57
## Gene58
## Gene59  8.312043
## Gene60  7.813793  6.992657
```

Paso 25. Identificar el heatmap de la correlación entre pacientes.

```
pheatmap(cor(sub))
```



Paso 26. Identificar el heatmap de la correlación entre genes con la matriz traspuesta.

```
trasp <- t(sub)
pheatmap(cor(trasp))
```

