

Heatmaps en R

Midhely García

3/9/2022

Laboratorio 36 - MAPA DE CALOR -TÉRMICO- with pheatmap DATOS GENETICOS 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 your 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 trials done to deal with the randomness self organizing maps

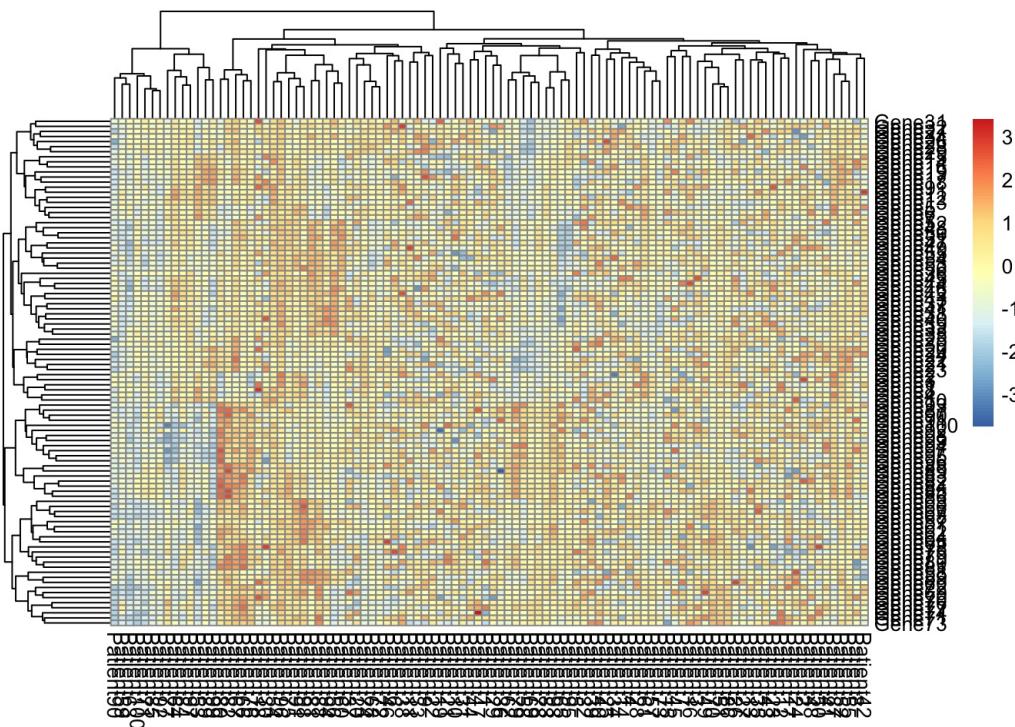
Instalar paquetería y llamar a librería install.packages("pheatmap") library(pheatmap)

Importar datos utilizando el comando file.choose() para saber su ubicación

```
genes <- as.matrix(  
  read.csv("/Users/midhely/Desktop/BMGO_LAB36/INPUT/heatmap_data.csv",  
    sep = ",",  
    header = T,  
    row.names = 1))  
  
annotation_col <- read.csv("/Users/midhely/Desktop/BMGO_LAB36/INPUT/annotation_col.csv", header = T,  
  row.names = 1)  
  
annotation_row <- read.csv("/Users/midhely/Desktop/BMGO_LAB36/INPUT/annotation_row.csv", header = T,  
  row.names = 1)
```

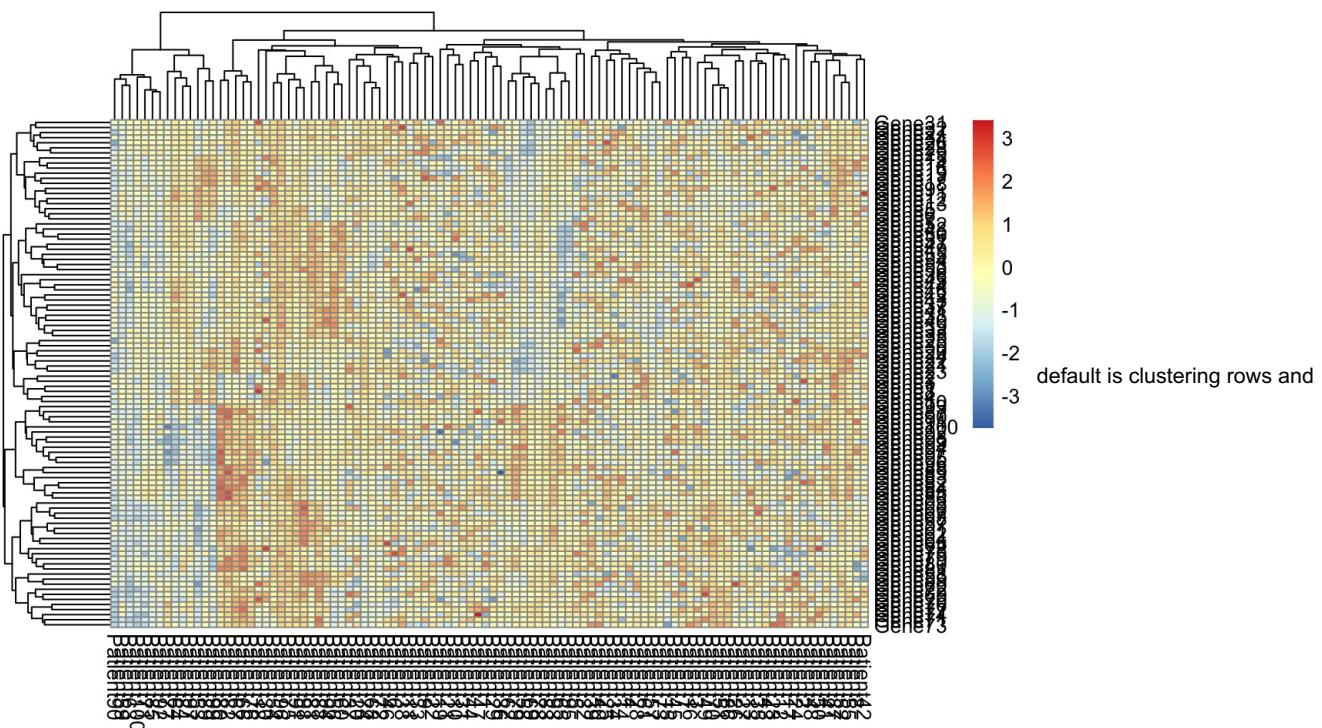
Plotting with pheatmap

```
library(pheatmap)  
pheatmap(genes)
```



change font

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



columns

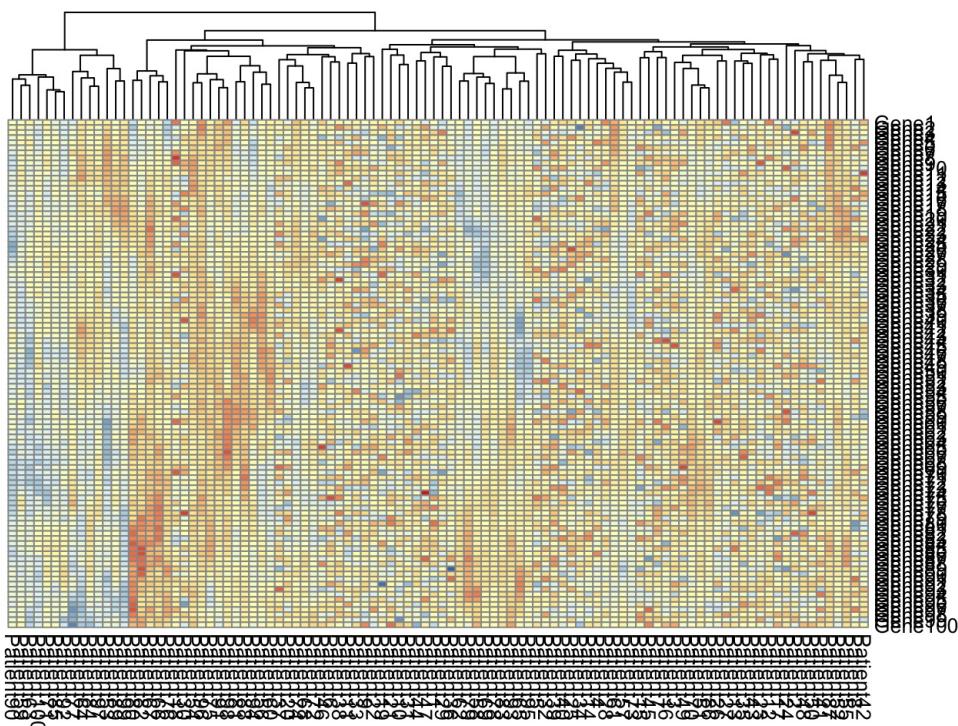
cluster by gene - groups of similar genes—LOS GENES ESTAN EN LOS RENGLONES POR DEFAULT CLUSTEA LOS RENGLONES

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



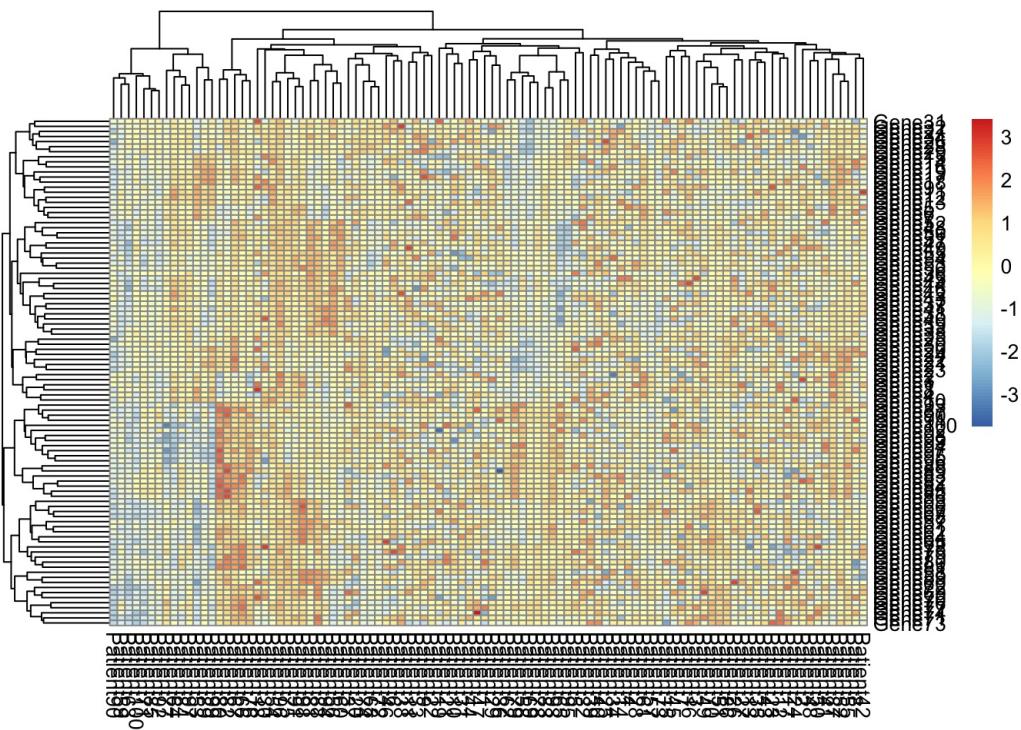
cluster by patient - groups of similar patients DEBES HACER QUE LAS COLUMNAS SE TRANFORMEN A RENGLONES

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



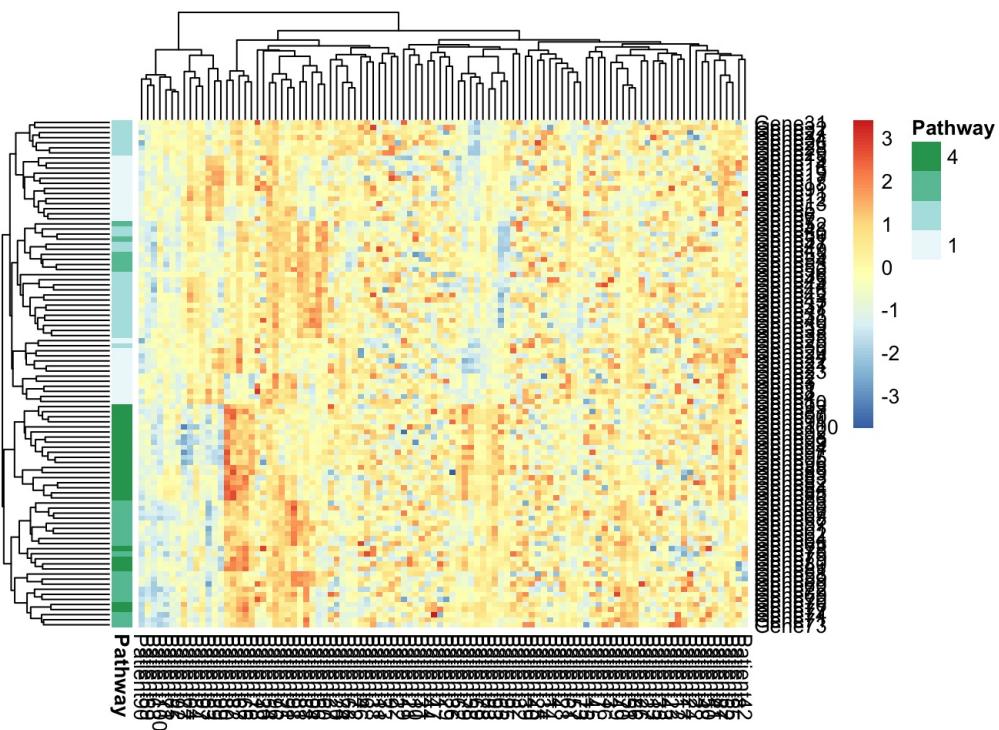
usually order by both

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



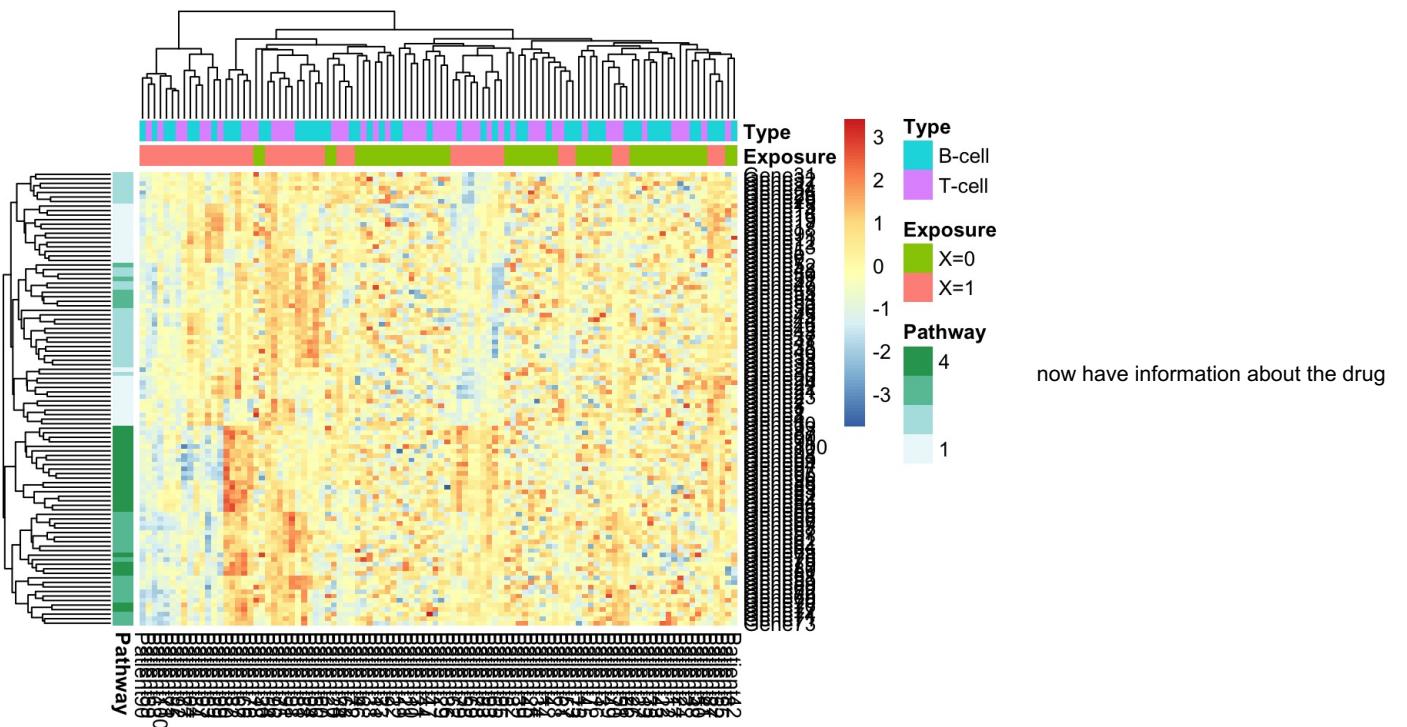
seeing some patterns emerge - but what do they mean? Great time to add annotation to our plot

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



add to row first, see that genes are clustering according to the pathways they belong to

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



and condition

GRAFICO COMPLETO G1

```
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")
```

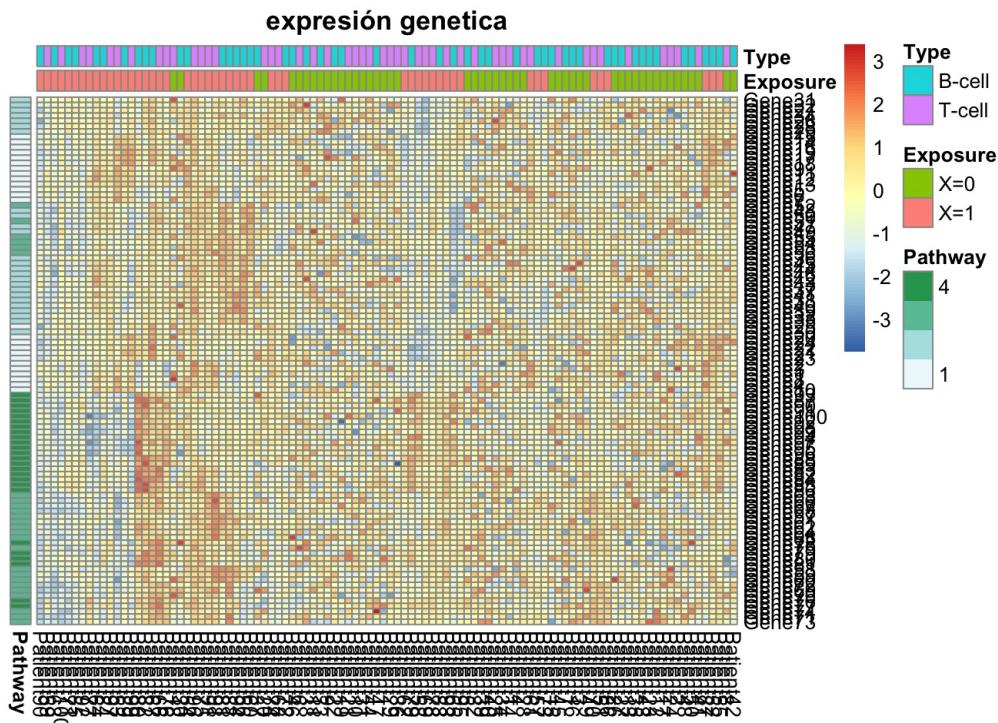


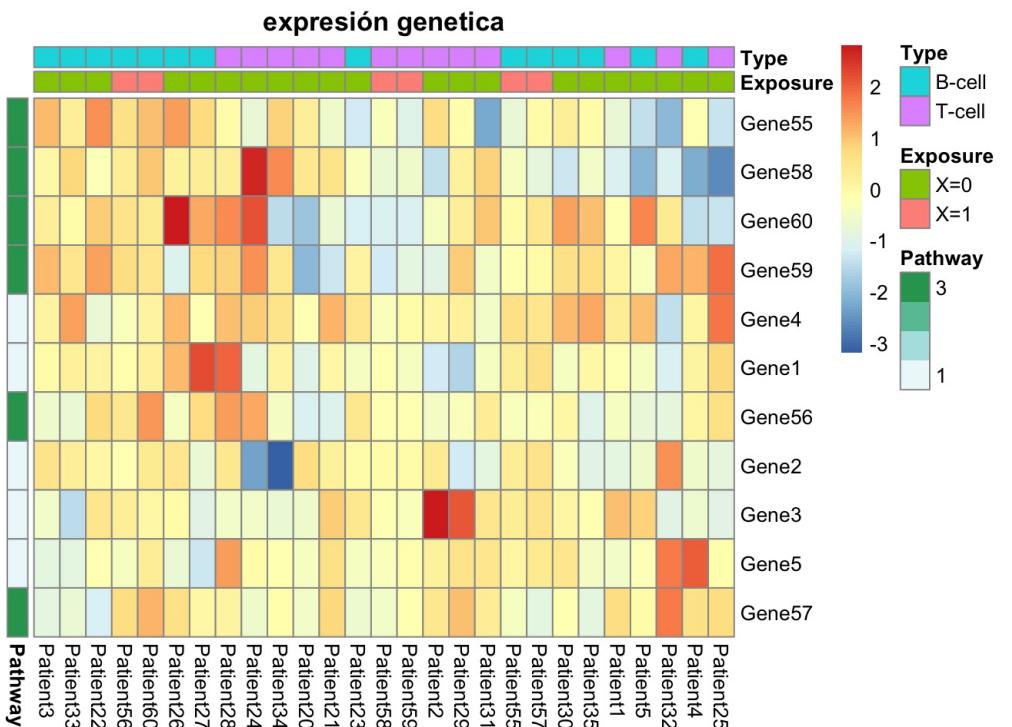
GRAFICO QUITANDO CLUSTERS (ARBOLES DE AGRUPACIÓN-DENDROGRAMAS)

take a smaller subset

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

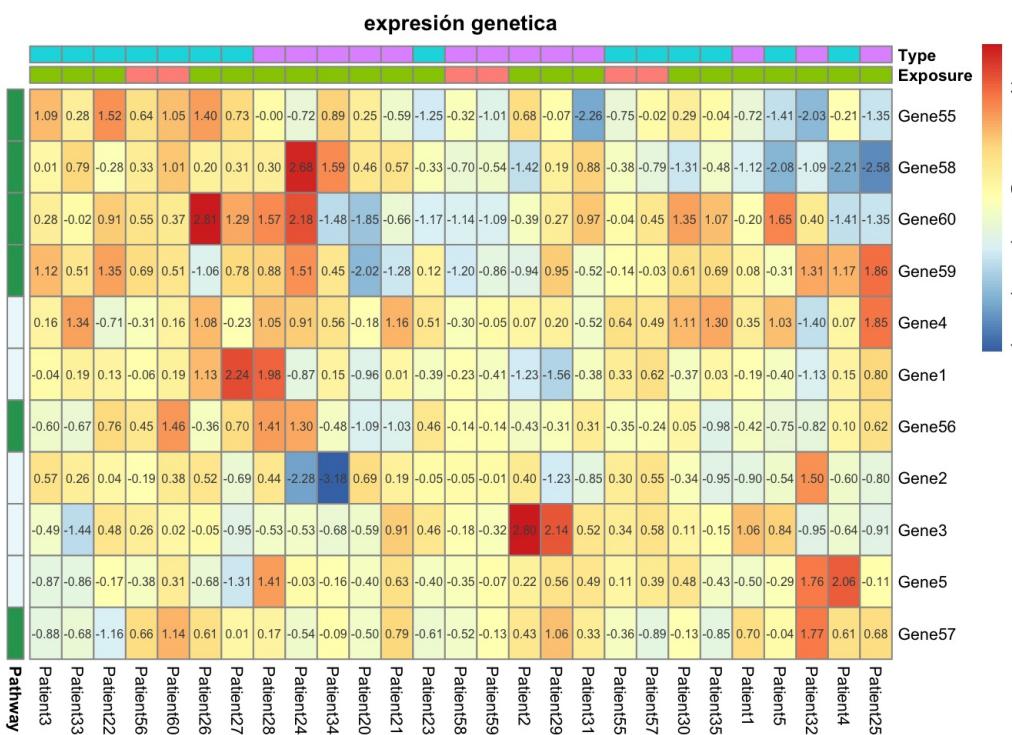
con subset 1 (COPIAR G1)

```
pheatmap(sub, 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")
```



con subset 2 – DESPLEGAR VALORES

```
pheatmap(sub, frontsize = 6, 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_l egend = FALSE, display_numbers = TRUE, fontsize_number = 6)
```



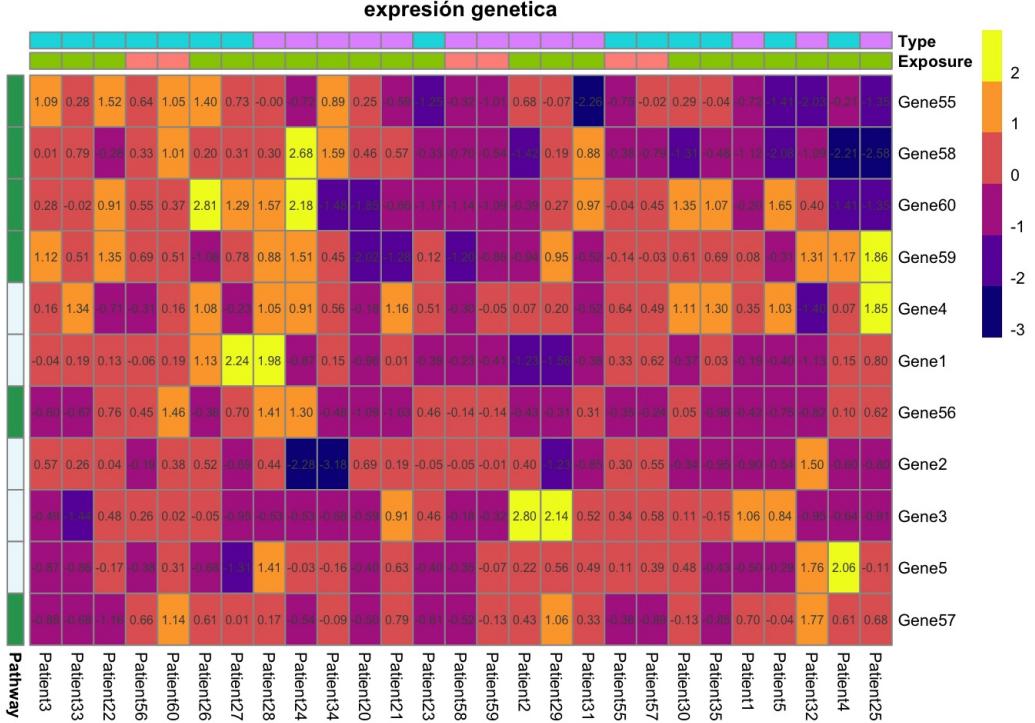
Utilizar viridis, magma, plasma, cividis, inferno para diseño, asegurarse de instalar paquetería y llamar a librería install.packages("viridis") library(viridis) install.packages("viridisLite") library(viridisLite)

Gráficos con color

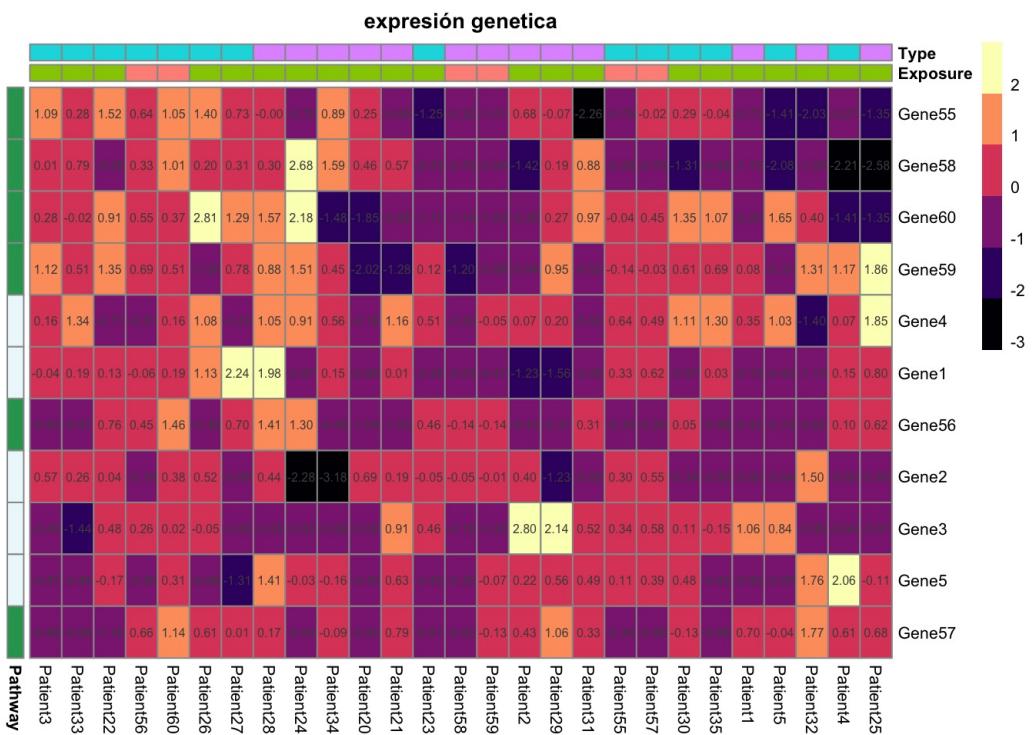
```
library(viridis)
```

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

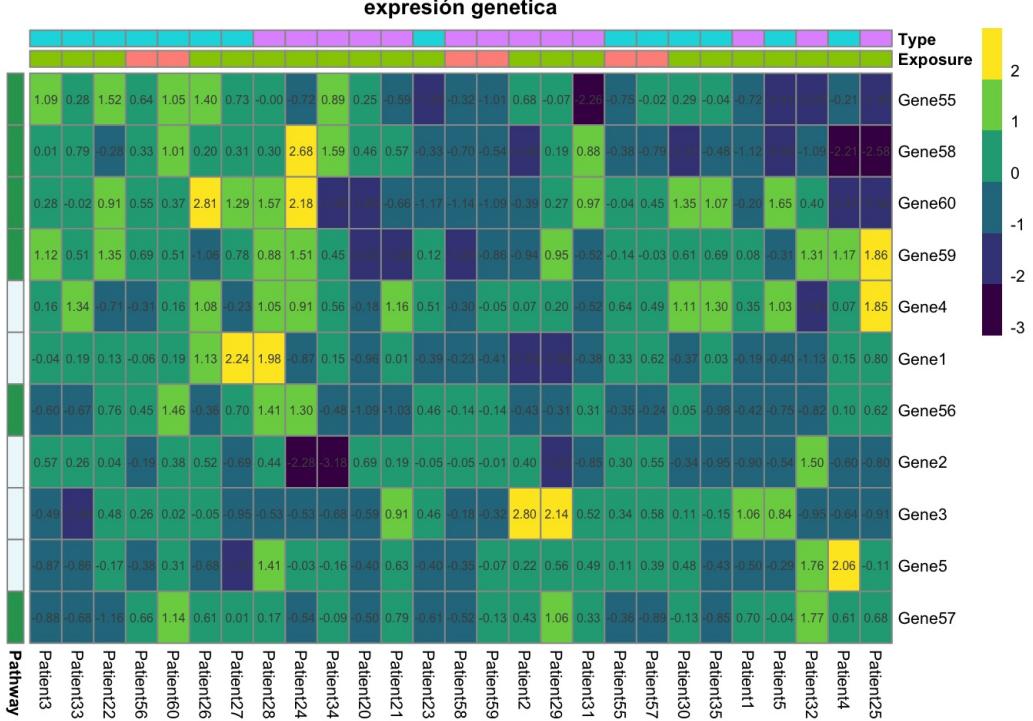
```
pheatmap(sub, frontsize = 6, cluster_rows = T, cluster_cols = T, annotation_row = annotation_row, annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0, main = "expresión genética", fontsize = 8, annotation_l egend = FALSE, display_numbers = TRUE, fontsize_number = 6, col = viridis_pal(option = "plasma") (6))
```



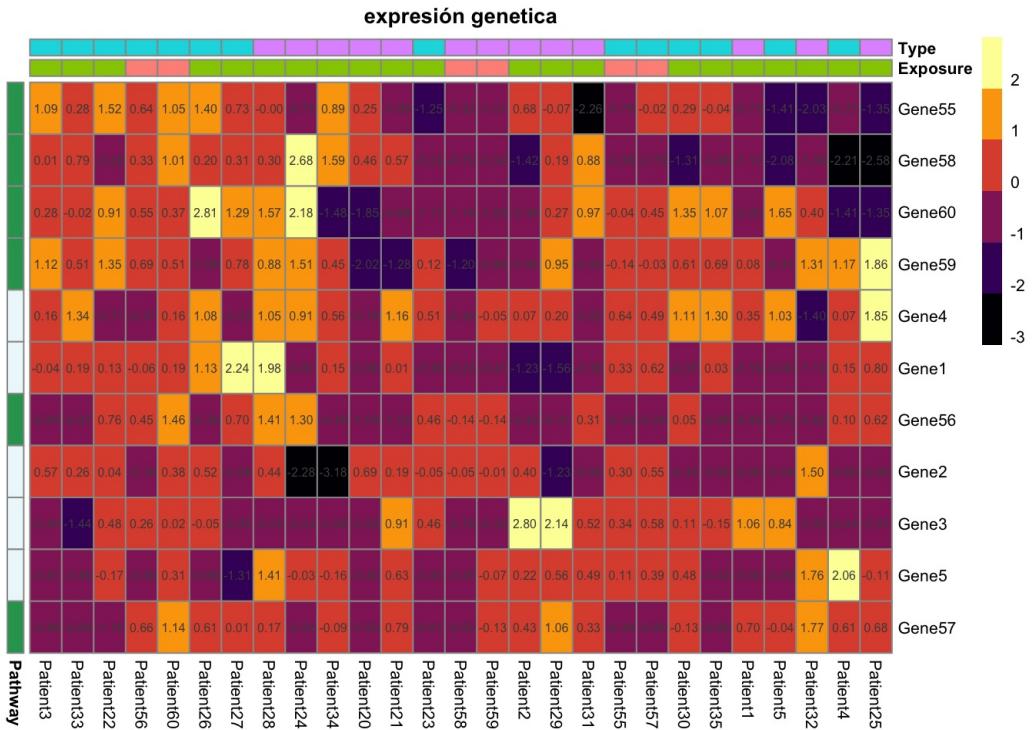
```
heatmap(sub, frontsize = 6, cluster_rows = T, cluster_cols = T, annotation_row = annotation_row, annotation_col = annotation_col, treeheight_row = 0, treeheight_col = 0, main = "expresión genética", fontsize = 8, annotation_legend = FALSE, display_numbers = TRUE, fontsize_number = 6, col = viridis_pal(option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



```
heatmap(sub, frontsize = 6, 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))
```



```
heatmap(sub, frontsize = 6, 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 = "inferno") (6))
```



Añadir elementos adicionales (identificar la distancia entre datos/genes)

dist(sub)

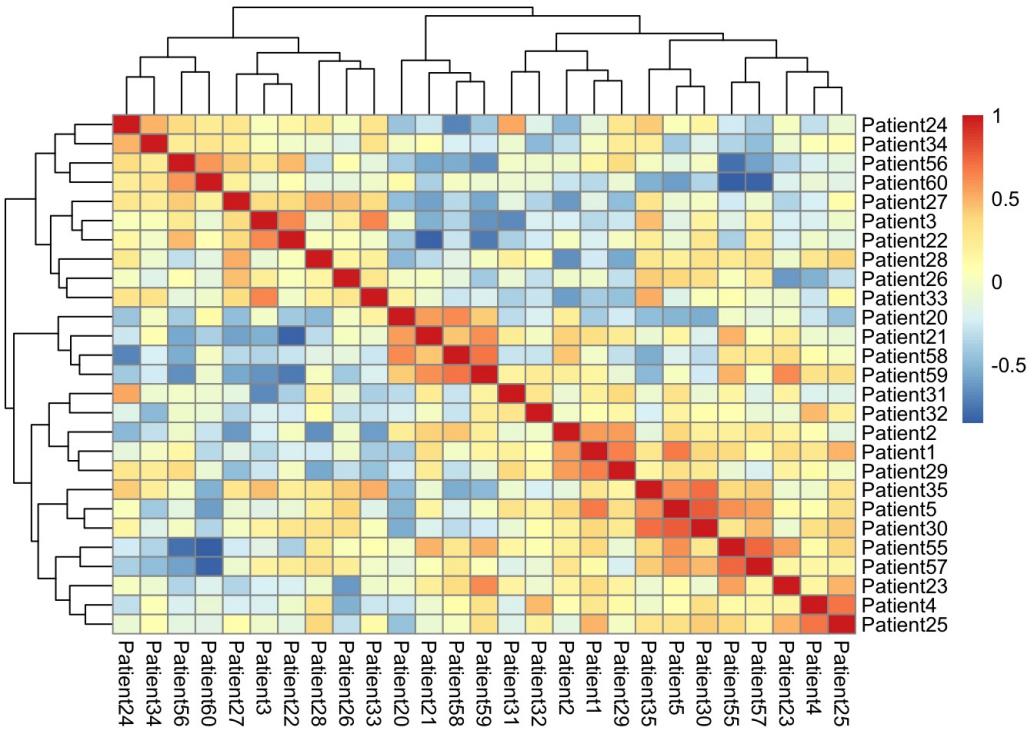
```

##          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

```

Identificar el mapa de calor de la correlación de los datos (entre pacientes)

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

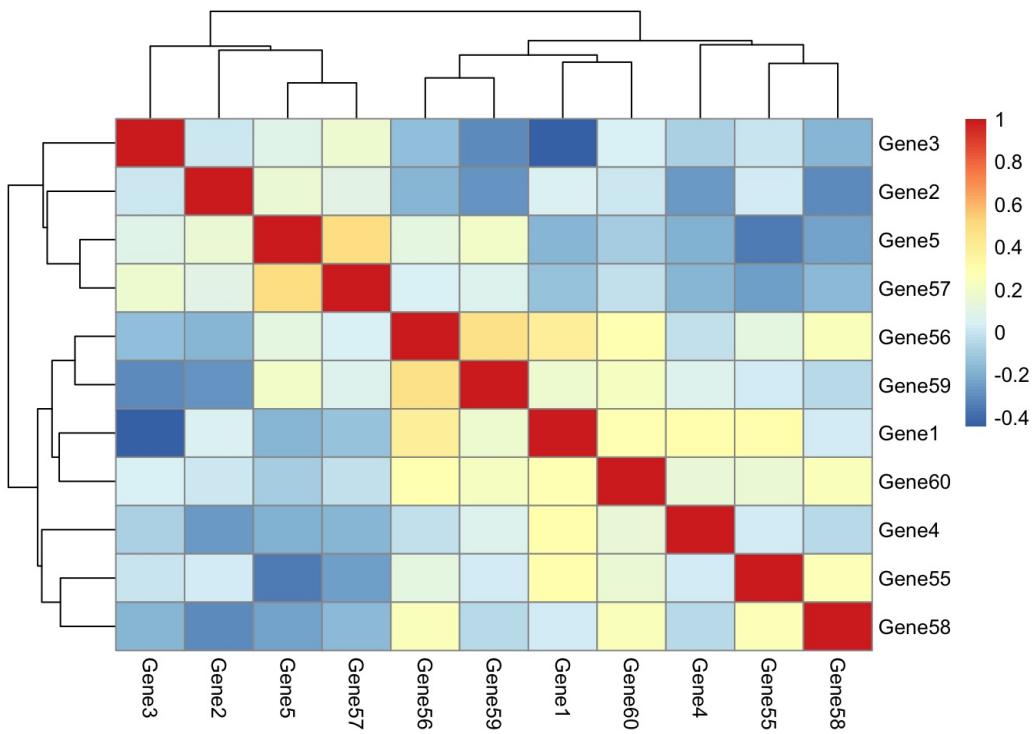


Identificar la correlación entre genes, utilizando matriz transpuesta

```

trans <- t(sub)
pheatmap(cor(trans))

```



slightly artificial with color bar, without and with ordering: cc <- rainbow(nrow(Ca)) heatmap(Ca, Rowv = FALSE, symm = TRUE, RowSideColors = cc, ColSideColors = cc, margins = c(6,6)) heatmap(Ca, symm = TRUE, RowSideColors = cc, ColSideColors = cc, margins = c(6,6))