

Neuronal and glial DNA methylation and gene expression changes in early epileptogenesis

Erika Nathalia Ordoñez Guzmán

2025-02-07

Introducción:

Según los datos la Organización Mundial de la Salud proporcionados en 2024: “50 millones de personas padecen epilepsia” además “se estima que el 70% de las personas con epilepsia podrían vivir sin convulsiones si se diagnosticaran y trataran adecuadamente”. Este es un trastorno que afecta principalmente a las células que conforman el cerebro. El artículo: “Neuronal and glial DNA methylation and gene expression changes in early epileptogenesis” trabaja con células neuronales y de glía como lo es la microglía. Este padecimiento se caracteriza por descargas eléctricas en el cerebro y afectan negativamente a las personas que la padecen. Estudios recientes, resaltan que la metilación tiene un papel importante en este proceso, por lo que el equipo de trabajo dirigido por Toni C Berger, son pioneros en este campo de estudio.

Por la naturaleza del proceso de metilación, es un proceso heredable, modificable y estable, pero temporal, además de voluble y susceptible a cambios por influencia ambiental. Por eso, en el artículo se experimenta con células neuronales de ratón, en la inducción a la metilación con ácido kainico, que son el grupo experimental, mientras que el control se trató con solución salina. Se buscó encontrar si existe una diferencia significativa de expresión en ambas condiciones. Con este artículo, se busca encontrar el efecto que tiene la metilación en la patología y así, estar un paso más cerca de encontrar agentes terapéuticos para contrarrestar su efecto y apoyar a la salud mundial.

El set de datos obtenido fue obtenido desde recount3, cuenta con 28 muestras. Los datos demuestran las lecturas del mDNA que se dieron como consecuencia de la metilación y secuenciación de mRNA (mRNAseq).

Objetivos del proyecto:

- Encontrar blancos terapéuticos en células neuronales y glía que juegan un papel crucial en la epilepsia.
- Analizar el resultado del mRNAseq para comprobar la diferencia de expresión y correlacionarlo con la epilepsia.

Resultados:

```
# Explorar objeto de rse
rse_gene_SRP223512$sra.sample_attributes[1:5]

## [1] "neun;;negative|source_name;;Hippocampus|tissue;;Hippocampus|treatment;;Kainate"
## [2] "neun;;positive|source_name;;Hippocampus|tissue;;Hippocampus|treatment;;Kainate"
## [3] "neun;;negative|source_name;;Hippocampus|tissue;;Hippocampus|treatment;;Sham"
## [4] "neun;;positive|source_name;;Hippocampus|tissue;;Hippocampus|treatment;;Sham"
## [5] "neun;;negative|source_name;;Hippocampus|tissue;;Hippocampus|treatment;;Kainate"
```

```
hist(rse_gene_SRP223512$assigned_gene_prop)
```

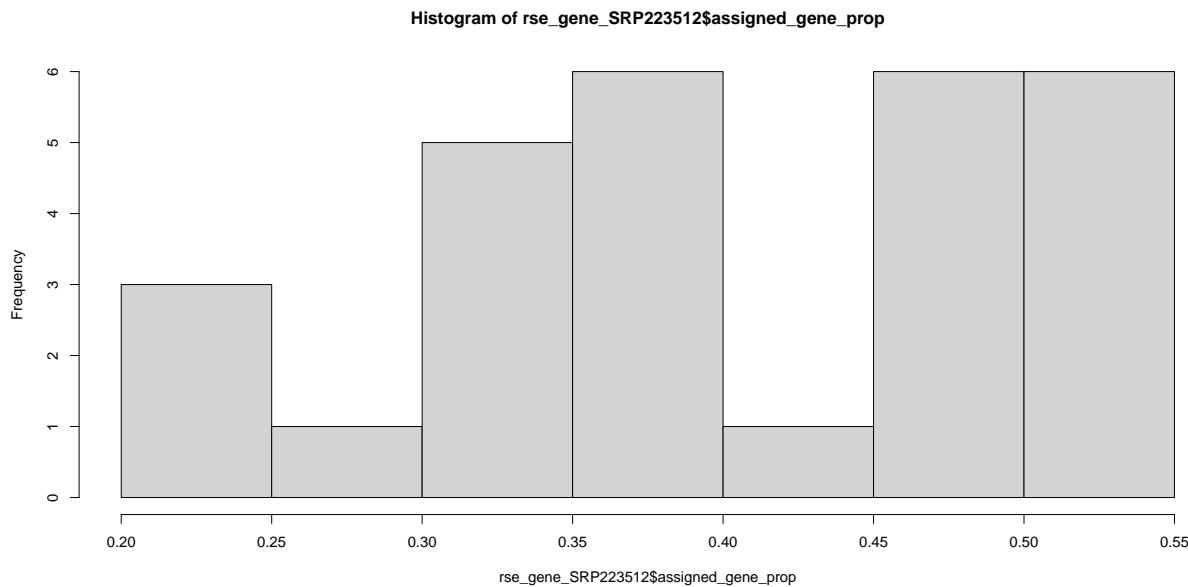


Figure 1: Frecuencias de la proporción de genes

En la figura 1 se muestra la frecuencia de las muestras según la proporción de los genes. Sirve para visualizar las muestras que se pueden conservar para el análisis.

```
# Se muestra de manera más numérica el resultado del histograma  
table(rse_gene_SRP223512$assigned_gene_prop < 0.3)
```

```
##  
## FALSE TRUE  
##    24     4
```

```
table(rse_gene_SRP223512$assigned_gene_prop < 0.4)
```

```
##  
## FALSE TRUE  
##    13     15
```

```
table(rse_gene_SRP223512$assigned_gene_prop < 0.5)
```

```
##  
## FALSE TRUE  
##     6    22
```

Como podemos ver con los resultados anteriores, el mejor umbral para obtener la cantidad de muestras para trabajar es la proporción de 0.3. Debido a que no buscamos un umbral muy estricto y quedarnos sin una buena cantidad de muestras para analizar.

```
rse_gene_SRP223512 <- rse_gene_SRP223512[, rse_gene_SRP223512$assigned_gene_prop > 0.3]
```

Resumen de los niveles medios de expresión de genes:

```
summary(gene_means)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	0.00	0.50	2.88	96.98	33.42	132204.88

Se eligió un umbral de 0.1 para la elegir los genes con los que se trabajará en el análisis. El resto de genes se eliminan. De decir, que los genes que tienen una media menor a 0.1 se eliminan.

```
rse_gene_SRP223512 <- rse_gene_SRP223512[gene_means > 0.1, ]  
dim(rse_gene_SRP223512)
```

```
## [1] 49447    24
```

Porcentaje de genes conservados:

```
round(nrow(rse_gene_SRP223512) / nrow(rse_gene_SRP223512_unfiltered) * 100, 2)
```

```
## [1] 89.22
```

```
ggplot(as.data.frame(colData(rse_gene_SRP223512)), aes(y=assigned_gene_prop, x=treatment)) +  
  geom_boxplot() +  
  theme_bw(base_size = 20) +  
  ylab("Assigned Gene Prop") +  
  xlab("Treatment")
```

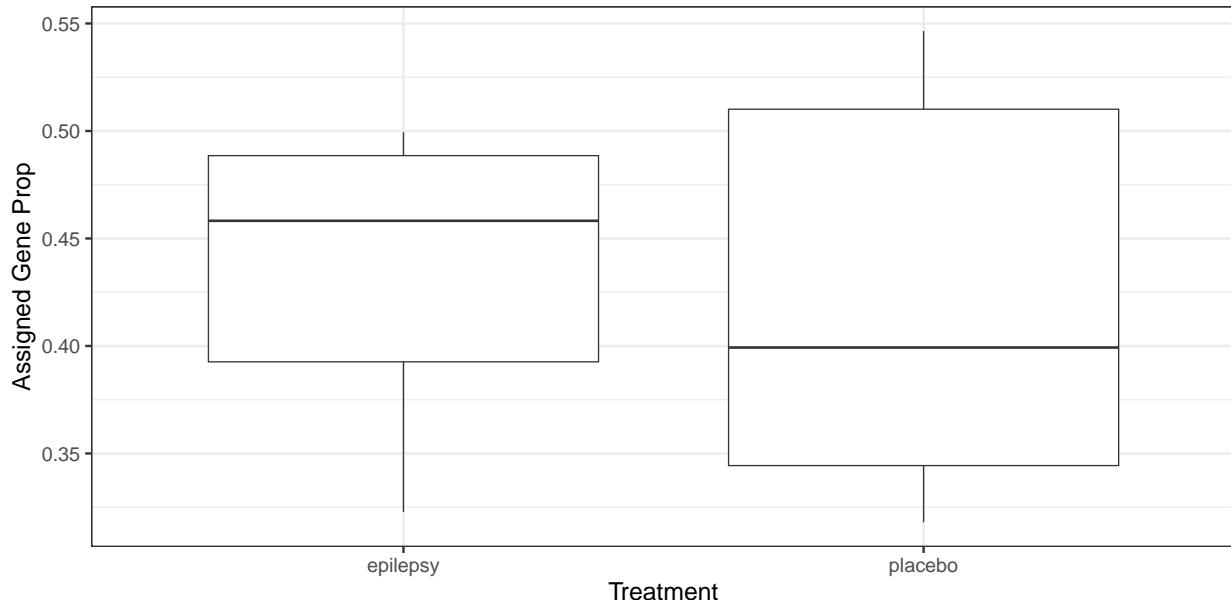


Figure 2: Boxplot con información sobre la proporción de genes con respecto a cada tratamiento

En la figura 2, se evidencia que la media de la proporción de genes es mayor en el tratamiento con ácido kainico.

```

# Modelo estadístico:
mod <- model.matrix(
  ~ treatment + sra_attribute.neun + assigned_gene_prop,
  data = colData(rse_gene_SRP223512)
)

# Se muestran los nombres de las columnas para evitar errores.
colnames(mod)

## [1] "(Intercept)"           "treatmentplacebo"
## [3] "sra_attribute.neunpositive" "assigned_gene_prop"

# Comprobación del modelo estadístico:
# Para visualizar el modelo:
# Data frame para la visualización.
sampleData <- data.frame(
  treatment = rse_gene_SRP223512$treatment,
  sra_attribute.neun = rse_gene_SRP223512$sra_attribute.neun
)
# Creación de las imágenes:
vd <- ExploreModelMatrix::VisualizeDesign(
  sampleData = sampleData,
  designFormula = ~ 0 + treatment + sra_attribute.neun,
  textSizeFitted = 3
)
cowplot::plot_grid(plotlist=vd$plot, nrow = 1)

```

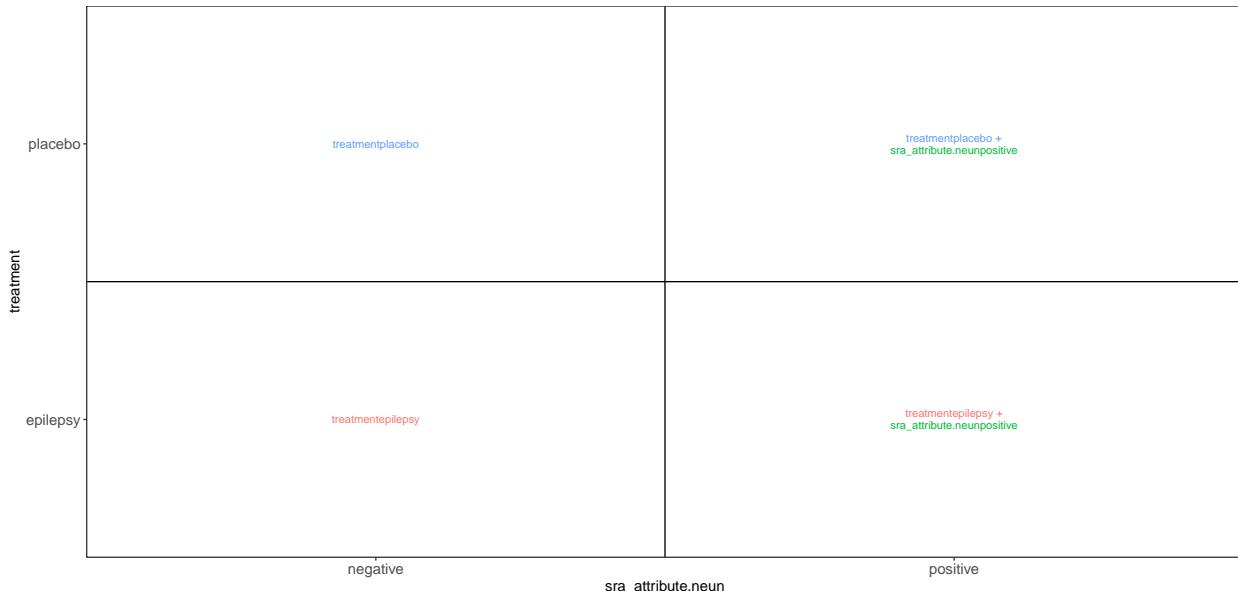


Figure 3: Cuadro con la comprobación del modelo de regresión lineal aplicado anteriormente

En el cuadro anterior, podemos darnos cuenta que el modelo fue correctamente planteado, esto se puede notar porque, si restamos la columna titulada negative a la titulada positive el resultado es

sra_attribute.neunpositive, lo que quiere decir que tuvo un efecto el receptor NeuN en las neuronas. Si se resta la fila de placebo a la de epilepsia en la columna positive, nos afirma que efectivamente, la fila de placebo es el control experimental mientras que la segunda fila de epilepsia afirma que ese es el grupo al que se aplicó el ácido kainico y por tanto el grupo experimental.

NOTA: Para recrear la imagen, no se aplicó el intercepto y tampoco se contempló la variable continua, que en este caso es la proporción de genes. Pues se sabe cual es esta variable y para visualizar de mejor manera se obvió. Por otro lado, esta variable si fue implementada en la construcción del modelo, junto con la aplicación del intercepto, para tener un modelo más confiable.

```
# Usar limma para el análisis de expresión diferencial:  
library("limma")  
# Crear el objeto DGEList  
# Se produce la gráfica para visualizar los datos.
```

```
vGene <- voom(dge, mod, plot=T)
```

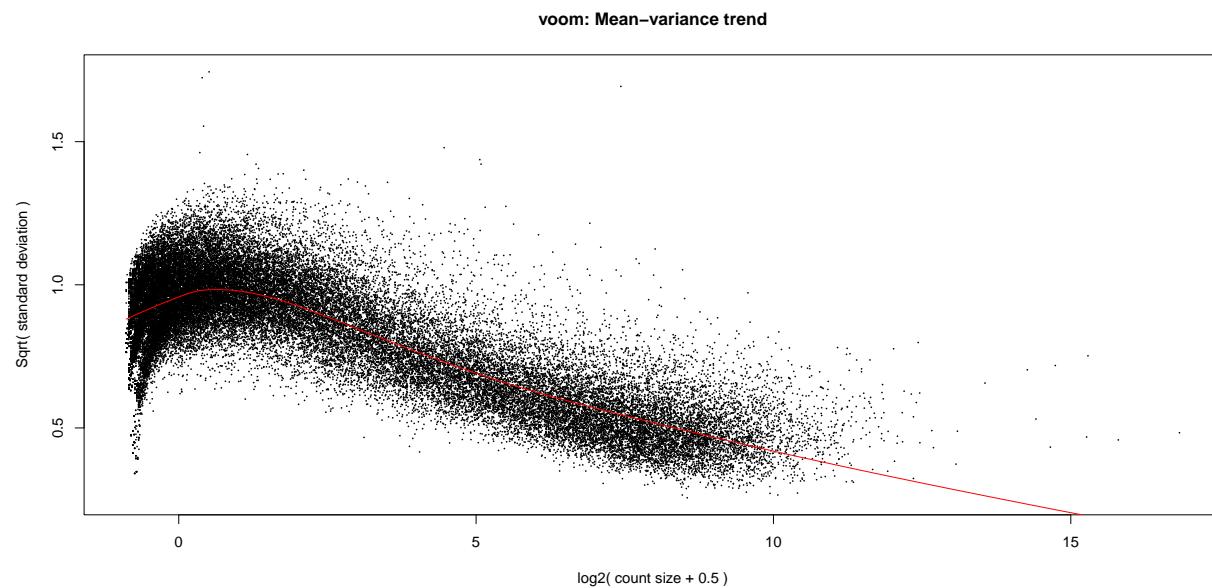


Figure 4: Gráfica para ver la desviación estándar

La gráfica anterior muestra los genes, es decir, cada punto es un gen. Muestra la buena calidad de los datos.

```
# Utiliza el modelo bayesiano para el análisis de expresión diferencial  
# Además, ajusta el modelo lineal  
eb_results <- eBayes(lmFit(vGene))  
  
# Selecciona los genes con expresión diferencial significativa.  
de_results <- topTable(  
  eb_results,  
  coef=2,  
  number=nrow(rse_gene_SRP223512),  
  sort.by="none"  
)
```

```

# Pequeña exploración y visualización de los datos.
dim(de_results)

## [1] 49447    17

head(de_results)

##           source type bp_length phase          gene_id
## ENSMUSG00000079800.2 ENSEMBL gene      1271      NA ENSMUSG00000079800.2
## ENSMUSG00000095092.1 ENSEMBL gene      366       NA ENSMUSG00000095092.1
## ENSMUSG00000079192.2 ENSEMBL gene      255       NA ENSMUSG00000079192.2
## ENSMUSG00000079794.2 ENSEMBL gene      255       NA ENSMUSG00000079794.2
## ENSMUSG00000094799.1 ENSEMBL gene      366       NA ENSMUSG00000094799.1
## ENSMUSG00000095250.1 ENSEMBL gene      366       NA ENSMUSG00000095250.1
##             gene_type gene_name level mgi_id havana_gene tag
## ENSMUSG00000079800.2 protein_coding AC125149.3     3 <NA> <NA> <NA>
## ENSMUSG00000095092.1 protein_coding AC125149.5     3 <NA> <NA> <NA>
## ENSMUSG00000079192.2 protein_coding AC125149.1     3 <NA> <NA> <NA>
## ENSMUSG00000079794.2 protein_coding AC125149.2     3 <NA> <NA> <NA>
## ENSMUSG00000094799.1 protein_coding AC125149.4     3 <NA> <NA> <NA>
## ENSMUSG00000095250.1 protein_coding AC133103.4     3 <NA> <NA> <NA>
##            logFC AveExpr          t   P.Value adj.P.Val
## ENSMUSG00000079800.2 -0.79211380 -1.339359 -1.83477639 0.07752263 0.3033380
## ENSMUSG00000095092.1  0.35531783 -2.827025  1.01138413 0.32076462 0.6599825
## ENSMUSG00000079192.2 -0.01990891 -1.833560 -0.05463589 0.95682874 0.9886429
## ENSMUSG00000079794.2 -0.60064052 -2.628905 -1.64792109 0.11089901 0.3774520
## ENSMUSG00000094799.1  0.41875215 -2.449011  1.52330036 0.13925145 0.4300254
## ENSMUSG00000095250.1  0.19371546 -1.691030  0.56942454 0.57375106 0.8360616
##           B
## ENSMUSG00000079800.2 -4.497655
## ENSMUSG00000095092.1 -5.630542
## ENSMUSG00000079192.2 -6.028328
## ENSMUSG00000079794.2 -4.808607
## ENSMUSG00000094799.1 -5.015600
## ENSMUSG00000095250.1 -5.907050

# Selección de genes con una expresión diferencial significativa
table(de_results$adj.P.Val < 0.05)

## 
## FALSE  TRUE
## 43793 5654

# Visualización de datos - plotMA
plotMA(eb_results, coef=2)

```

Cada punto en la gráfica anterior muestra un gen, se perciben más genes que se mencionen cerca del cero, es decir, que no tienen una expresión significativa. Hay más genes que se tienen un log fold-change con expresión negativa.

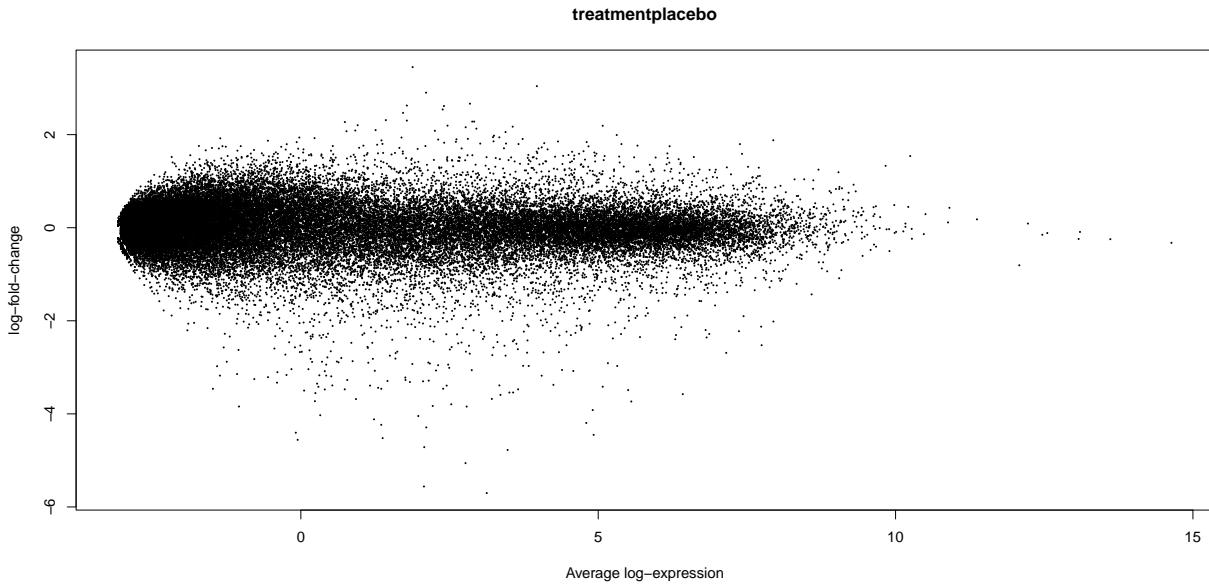


Figure 5: plotMA para evaluar la relación entre la expresión media y el cambio en la expresión

```
# Visualización de datos - Volcano:
volcanoplot(eb_results, coef=2, highlight=4, names=de_results$gene_name)
```

La imagen muestra 4 genes particulares que se encuentran diferencialmente expresados en color azul.

```
# Extraer los 50 genes con expresión más significativa
exprs_heatmap <- vGene$E[rank(de_results$P.Value) <= 50, ]

# Data frame
df <- as.data.frame(colData(rse_gene_SRP223512)[, c("treatment", "sra_attribute.neun")])

# Nombre de las columnas del data frame
colnames(df) <- c("treatment", "Neun/marcador")

# Heatmap
# Si no se encuentra instalado el paquete, correr la siguiente línea:
# install.packages("pheatmap")
library("pheatmap")
pheatmap(
  exprs_heatmap,
  cluster_rows = T,
  cluster_cols = T,
  show_rownames = F,
  show_colnames = F,
  annotation_col = df,
)

library("RColorBrewer")
```

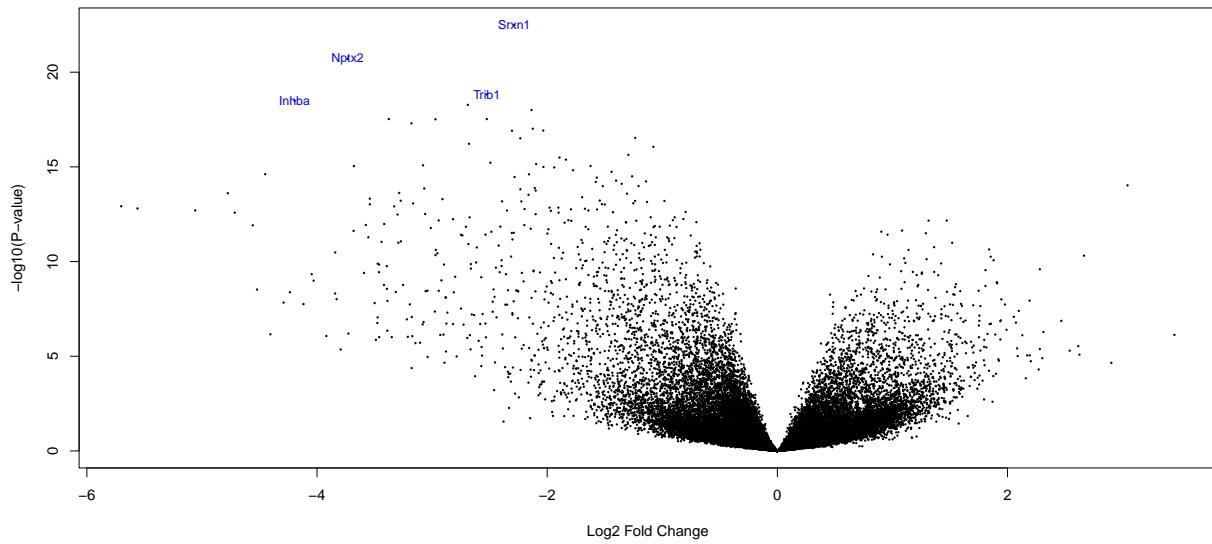


Figure 6: Los genes con expresión diferencial significativa resaltados en color azul

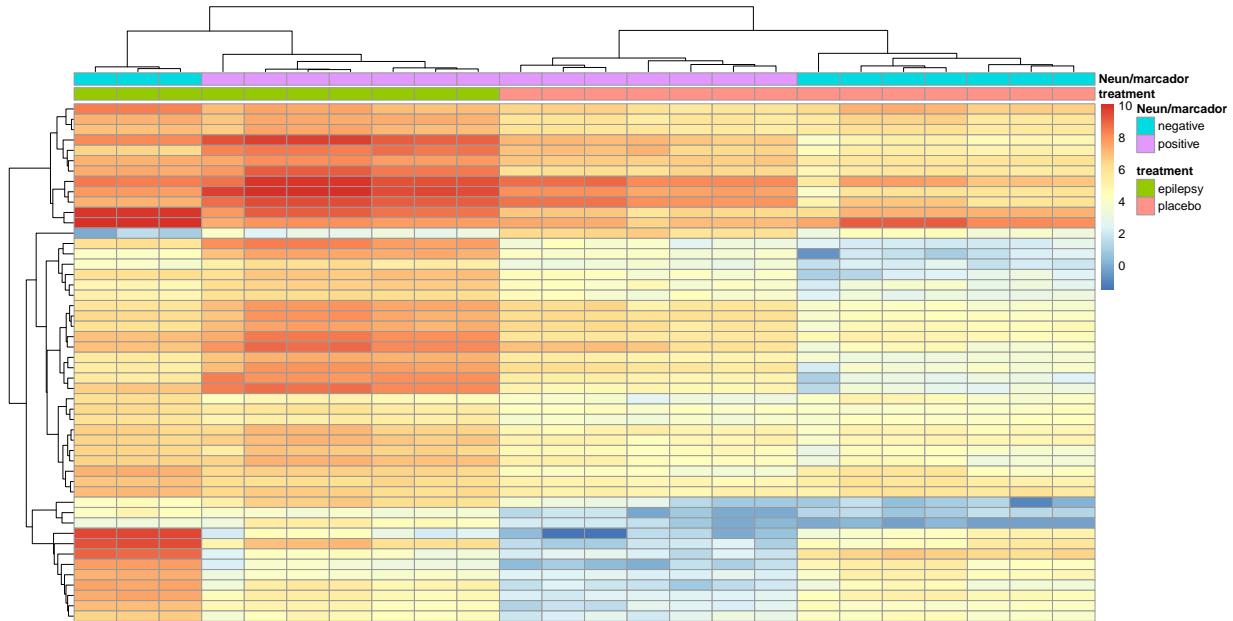


Figure 7: Heatmap con los 50 genes con expresión más significativa

```

# PCA - con el tratamiento: Kainate/Sham
col.group <- as.factor(df$treatment)
levels(col.group) <- brewer.pal(nlevels(col.group), "Set1")

## Warning in brewer.pal(nlevels(col.group), "Set1"): minimal value for n is 3, returning requested pal

col.group <- as.character(col.group)

plotMDS(vGene$E, labels = df$treatment, col=col.group)

```

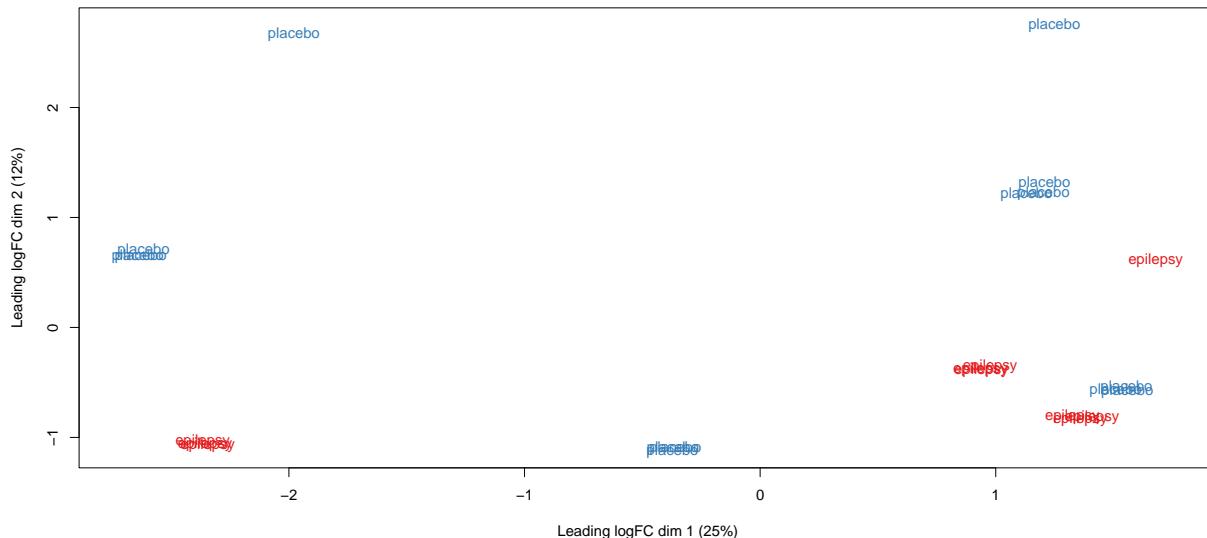


Figure 8: Heatmap con los 50 genes con expresión más significativa

```

# PCA - con el Neun: positivo/negativo
col.neun <- df`Neun/marcador`
levels(col.neun) <- brewer.pal(nlevels(col.neun), "Dark2")

## Warning in brewer.pal(nlevels(col.neun), "Dark2"): minimal value for n is 3, returning requested pal

col.neun <- as.character(col.neun)
plotMDS(vGene$E, labels = df`Neun/marcador`, col=col.neun)

```

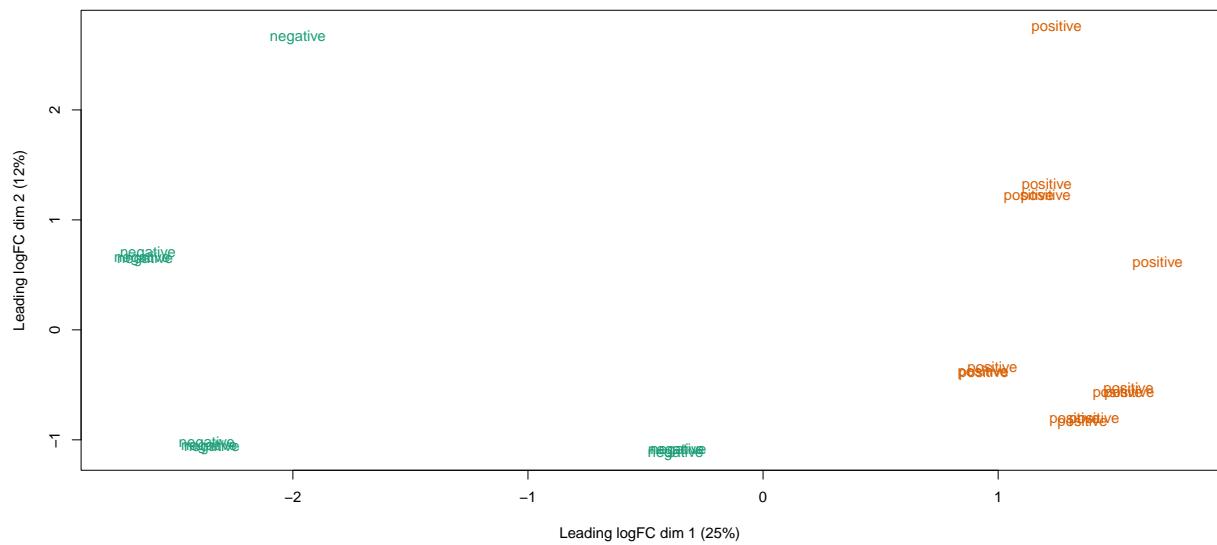


Figure 9: Heatmap con los 50 genes con expresión más significativa