

Proyecto Final

Angel Adrian De la Cruz Castillo

28/2/2021

Título de set de datos a utilizar: Single-nucleus RNA sequencing of human cortex affected by multiple sclerosis

Abstract:

Multiple sclerosis (MS) is characterized by cell proliferation, migration and damage in various cell types in different CNS regions and causes disabilities related to distinct neurological pathways, such as walking, vision and cognition. Here, region-specific transcriptomic approach was used to determine changes in gene expression in five different CNS regions (hippocampus, frontal cortex, internal capsule, corpus callosum, and parietal cortex) in MS. Overall design: Five MS patients (average age = 57.6 years) and five age-matched healthy controls (average age = 56.2 years) fresh frozen autopsy samples were obtained from Human Brain and Spinal Fluid Research Center in Los Angeles.

- Obtención de datos y modificaciones iniciales

```
library("recount3")

## Warning: package 'recount3' was built under R version 4.0.3
## Warning: package 'SummarizedExperiment' was built under R version 4.0.3
## Warning: package 'MatrixGenerics' was built under R version 4.0.3
## Warning: package 'matrixStats' was built under R version 4.0.3
## Warning: package 'GenomicRanges' was built under R version 4.0.3
## Warning: package 'BiocGenerics' was built under R version 4.0.3
## Warning: package 'S4Vectors' was built under R version 4.0.3
## Warning: package 'IRanges' was built under R version 4.0.3
## Warning: package 'GenomeInfoDb' was built under R version 4.0.3
## Warning: package 'Biobase' was built under R version 4.0.3
## Obtaining projects
human_projects <- available_projects()

## Title:
## Single-nucleus RNA sequencing of human cortex affected by multiple sclerosis

## Creating RSE object with specific info
```

```

rse_gene_SRP173190 <- create_rse(
  subset(
    human_projects,
    project == "SRP173190" & project_type == "data_sources"
  )
)

## Extracting read counts
assay(rse_gene_SRP173190, "counts") <- compute_read_counts(rse_gene_SRP173190)

## Making info easier to handle
rse_gene_SRP173190 <- expand_sra_attributes(rse_gene_SRP173190)

```

- Exploración y filtrado de datos

```

## Exploring interest columns

colData(rse_gene_SRP173190) [
  ,
  grepl("^sra_attribute", colnames(colData(rse_gene_SRP173190)))
]

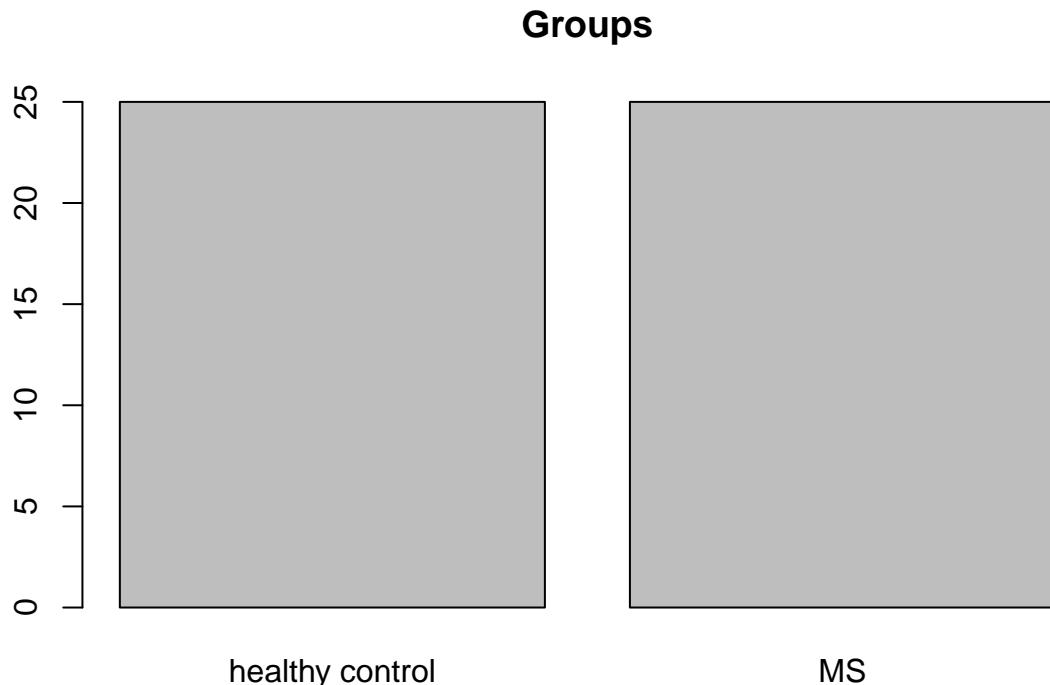
## DataFrame with 50 rows and 3 columns
##           sra_attribute.disease_state sra_attribute.source_name
##           <character>                  <character>
## SRR8307929          MS            corpus callosum
## SRR8307930          MS            frontal cortex
## SRR8307931          MS            parietal cortex
## SRR8307932          MS            hippocampus
## SRR8307933          MS            internal capsule
## ...                   ...
## SRR8307975          healthy control      frontal cortex
## SRR8307976          healthy control      parietal cortex
## SRR8307977          healthy control      hippocampus
## SRR8307978          healthy control      internal capsule
## SRR8307973          healthy control      internal capsule
##           sra_attribute.tissue
##           <character>
## SRR8307929          corpus callosum
## SRR8307930          frontal cortex
## SRR8307931          parietal cortex
## SRR8307932          hippocampus
## SRR8307933          internal capsule
## ...                   ...
## SRR8307975          frontal cortex
## SRR8307976          parietal cortex
## SRR8307977          hippocampus
## SRR8307978          internal capsule
## SRR8307973          internal capsule

```

```
## Saving SE object just in case
rse_gene_SRP173190_unfiltered <- rse_gene_SRP173190
```

Se observa que solamente hay 3 columnas con atributos de interés con estado de la enfermedad y tejidos de los cuales provienen las muestras, a las cuales no hay que realizarles ningún cambio ya que tienen el tipo de dato correcto.

```
## Exploring differences between groups
barplot(table(rse_gene_SRP173190$sra_attribute.disease_state), main = "Groups")
```



```
table(rse_gene_SRP173190$sra_attribute.disease_state)
```

```
##
## healthy control      MS
##                 25      25
```

En la gráfica de barras se observa que en ambos grupos hay el mismo número de muestras, por lo que no habrá ningún sesgo relacionado a alguna diferencia en el número de muestras. Esto se confirma justamente con la tabla mostrada.

```
## Exploring expression means
summary(rowMeans(assay(rse_gene_SRP173190, "counts")))
```

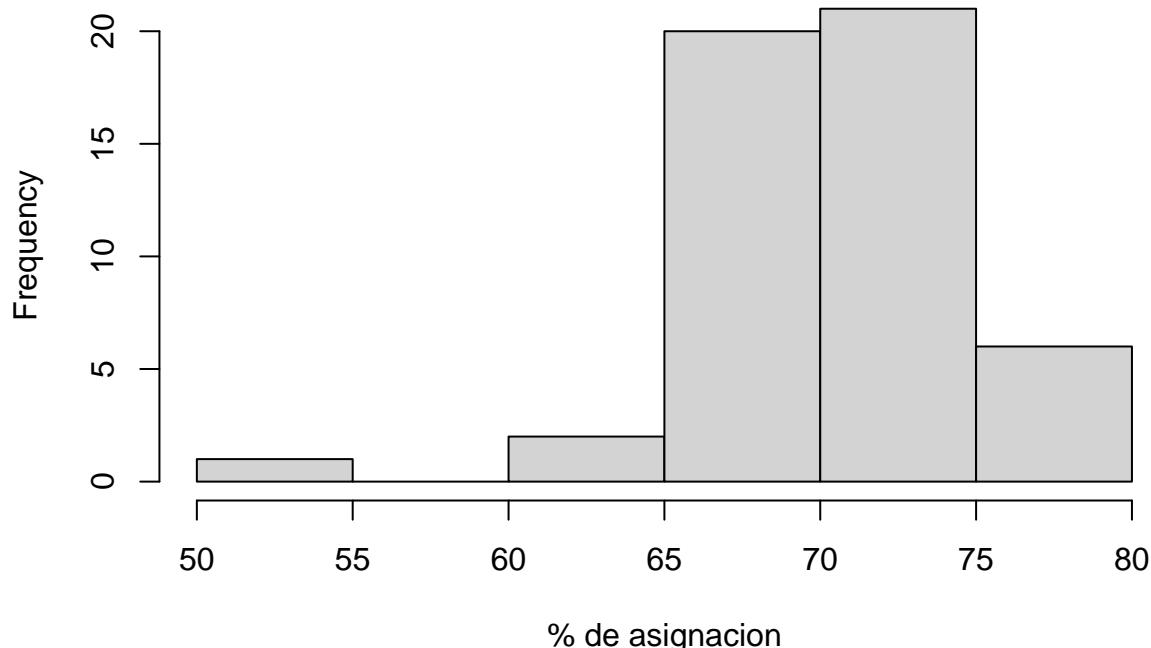
```
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
##      0.0      0.1      4.5    830.1   184.1 3099733.3
```

Se ve que el primer cuartil es de muy baja expresión, por lo que convendría filtrar esos genes de baja expresión.

```
## Filtering of low expression genes
expr_means <- rowMeans(assay(rse_gene_SRP173190, "counts"))
rse_gene_SRP173190 <- rse_gene_SRP173190[expr_means > 0.1, ]
```

Un posible problema que podría haber es que el porcentaje de secuencias asignadas a un gen sea demasiado bajo, por lo cual se requiere observar cuáles son las tendencias.

```
## Exploring assign percentage
hist(rse_gene_SRP173190$`recount_qc.gene_fc.all_%`, xlab = "% de asignacion", main = "")
```



```
## No filtering because assignment percentage is quite good
```

En el histograma del porcentaje de asignación se ve que el porcentaje es generalmente bueno, el porcentaje de asignación más pequeño es 50%, por lo que realmente no conviene filtrar esos datos.

```
## How much did we keep
round(nrow(rse_gene_SRP173190) / nrow(rse_gene_SRP173190_unfiltered) * 100, 2)
## [1] 75.89
```

Después de filtrar los genes de baja expresión, nos quedamos con el 75% de los genes que había originalmente.

- Normalización de datos

```
library("edgeR")  
  
## Warning: package 'edgeR' was built under R version 4.0.3  
## Warning: package 'limma' was built under R version 4.0.3  
dge <- DGEList(  
  counts = assay(rse_gene_SRP173190, "counts"),  
  genes = rowData(rse_gene_SRP173190)  
)  
dge <- calcNormFactors(dge)
```

- Exploración de modelo estadístico propuesto

Se propone un modelo en el que se va a comparar la expresión de genes en varios tejidos de cerebro de pacientes sanos y pacientes con esclerosis múltiple. El modelo propuesto se formularía como

$$grupoDeEnfermedad + tejido$$

Se debe verificar que este modelo sea full rank, es decir, que las variables a utilizar sean linealmente independientes.

```
## Exploring our proposed statistic model  
  
library("ExploreModelMatrix")  
  
## Warning: package 'ExploreModelMatrix' was built under R version 4.0.3  
data <- data.frame(disease_state = colData(rse_gene_SRP173190)$sra_attribute.disease_state,  
                     source = colData(rse_gene_SRP173190)$sra_attribute.source_name)  
vd <- ExploreModelMatrix::VisualizeDesign(  
  sampleData = data,  
  designFormula = ~ disease_state + source,  
  textSizeFitted = 2  
)  
  
cowplot::plot_grid(plotlist = vd$plotlist)
```

	corpus callosum	frontal cortex	hippocampus	internal capsule	parietal cortex
MS	(Intercept) + disease_stateMS	(Intercept) + disease_stateMS + sourcefrontal cortex	(Intercept) + disease_stateMS + sourcehippocampus	(Intercept) + disease_stateMS + sourceinternal capsule	(Intercept) + disease_stateMS + sourceparietal cortex
healthy control	(Intercept)	(Intercept) + sourcefrontal cortex	(Intercept) + sourcehippocampus	(Intercept) + sourceinternal capsule	(Intercept) + sourceparietal cortex

```
## Proposed model (~disease_state + source) is full rank, checked with ExploreModelMatrix()

## Assigning our already defined model to variable

mod <- model.matrix(~ sra_attribute.disease_state + sra_attribute.source_name,
                     data = colData(rse_gene_SRP173190)
)
```

Con ayuda de la función `ExploreModelMatrix()` se verificó que efectivamente era full rank y se rotó para facilitar su visualización:

- Explorando un posible sesgo de asignación de secuencias en los dos grupos a considerar

En este punto, se pensó que un posible sesgo que podría ocurrir es que debido a que no se filtraron los datos de las secuencias por su porcentaje de asignación, hubiera más secuencias no asignadas en un grupo que en otro, por lo que se visualizó esto con ayuda de una gráfica.

```
library("ggplot2")

## Warning: package 'ggplot2' was built under R version 4.0.3
library("ggsignif")

## Warning: package 'ggsignif' was built under R version 4.0.3
```

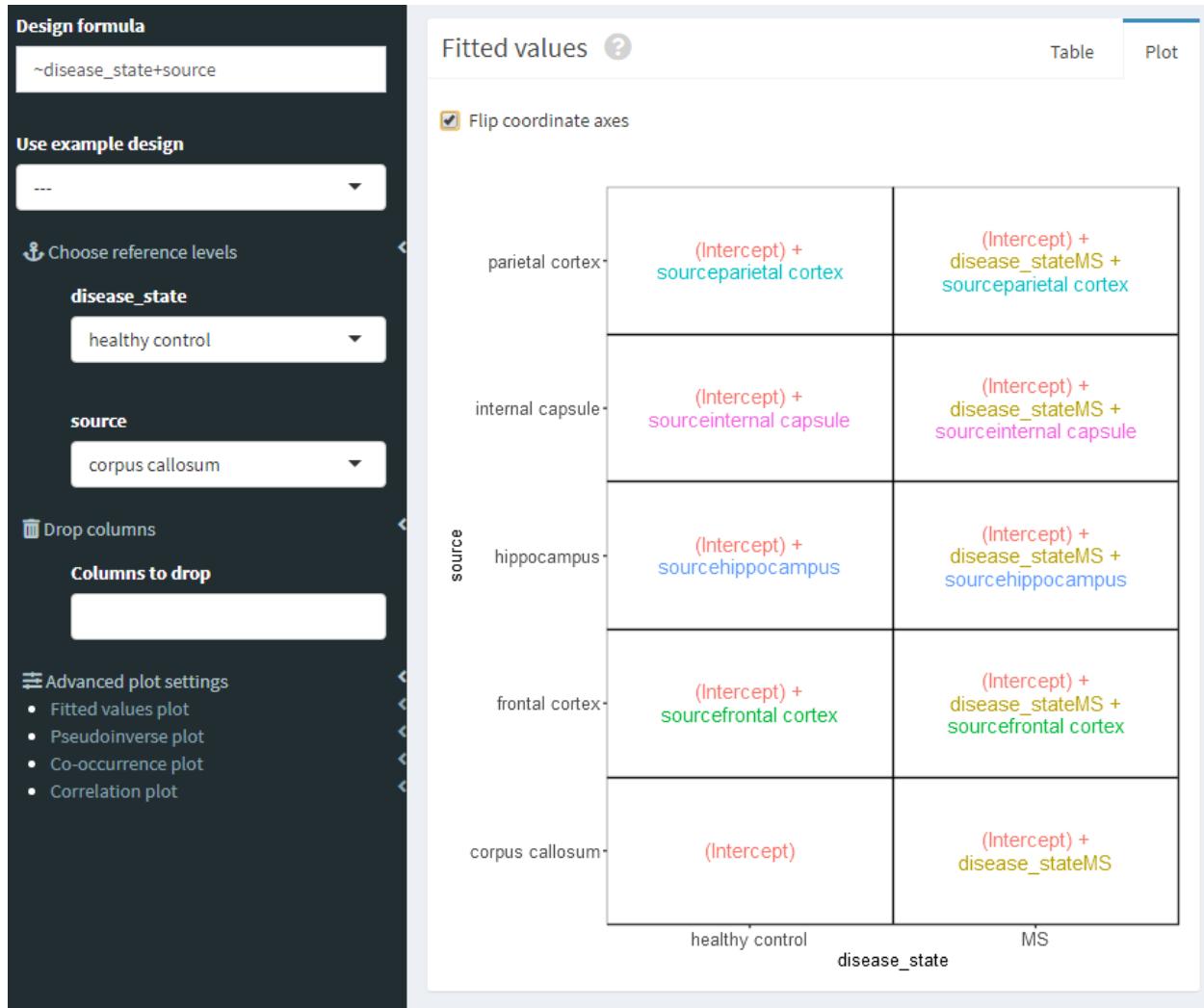


Figure 1: Modelo estadístico propuesto

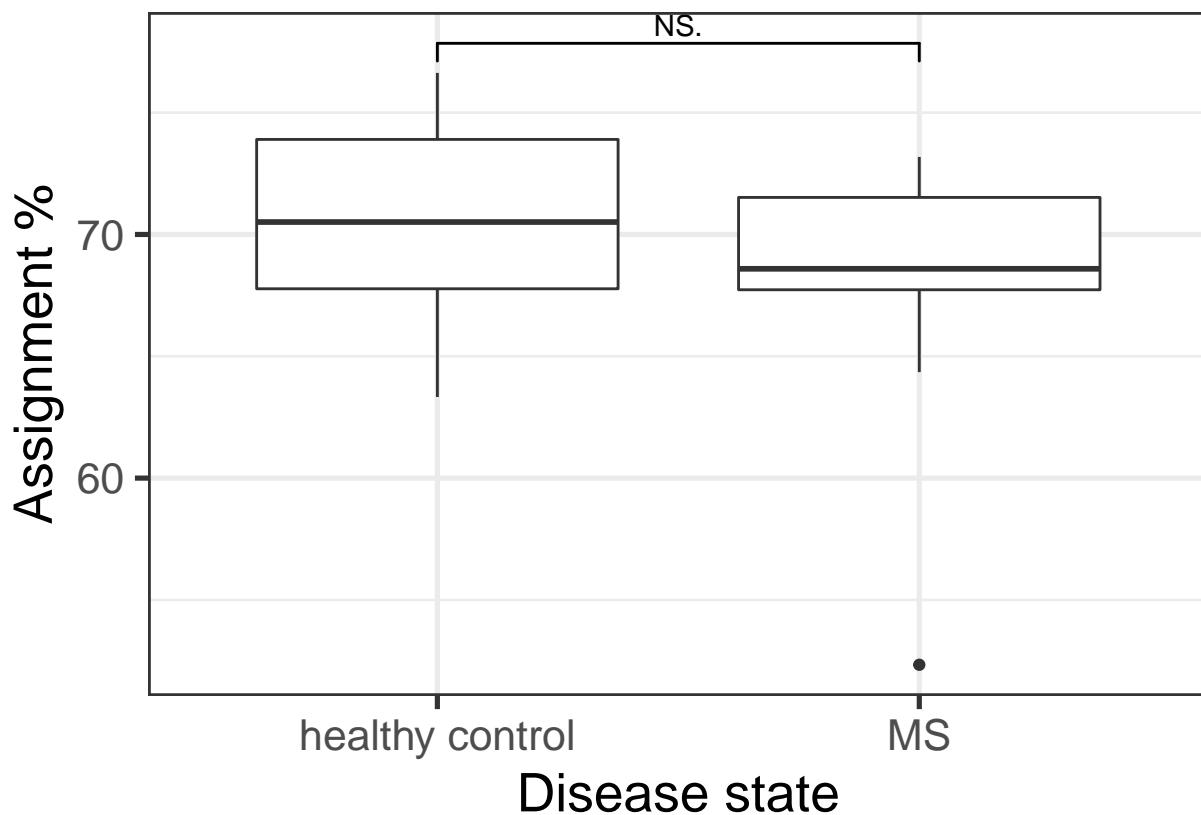
```

## Checking if differences are significant in total % of assignment
df <- data.frame(disease_state = rse_gene_SRP173190$sra_attribute.disease_state,
                  gene_assign =
                    rse_gene_SRP173190$`recount_qc.gene_fc.all_%`)

ggplot(df, aes(x = disease_state, y = gene_assign)) +
  geom_boxplot() +
  theme_bw(base_size = 20) +
  ylab("Assignment %") +
  xlab("Disease state") +
  geom_signif(comparisons=list(c("MS", "healthy control")),map_signif_level=TRUE)

## Warning in wilcox.test.default(c(66.06, 70.41, 68.52, 68.59, 72.24, 65.9, :
## cannot compute exact p-value with ties

```



A primera vista podría parecer que sí hay una diferencia, pero al usar la librería ggsignif que calcula el p-value, se comprueba que estas diferencias no son significativas.

- Análisis DE

```

library("limma")
vGene <- voom(dge, mod, plot = FALSE)

```

```

## Creates linear regression model and calculates p-values
eb_results <- eBayes(lmFit(vGene))

## Summary of expression results
de_results <- topTable(
  eb_results,
  coef = 2,
  number = nrow(rse_gene_SRP173190),
  sort.by = "none"
)

```

A partir del objeto que se crea con `topTable()`, se pueden filtrar los genes cuya expresión sí es estadísticamente diferente en ambos grupos.

```

## how many genes are actually expressing differentially
table(de_results$adj.P.Val < 0.05)

##
## FALSE TRUE
## 43082 5376

```

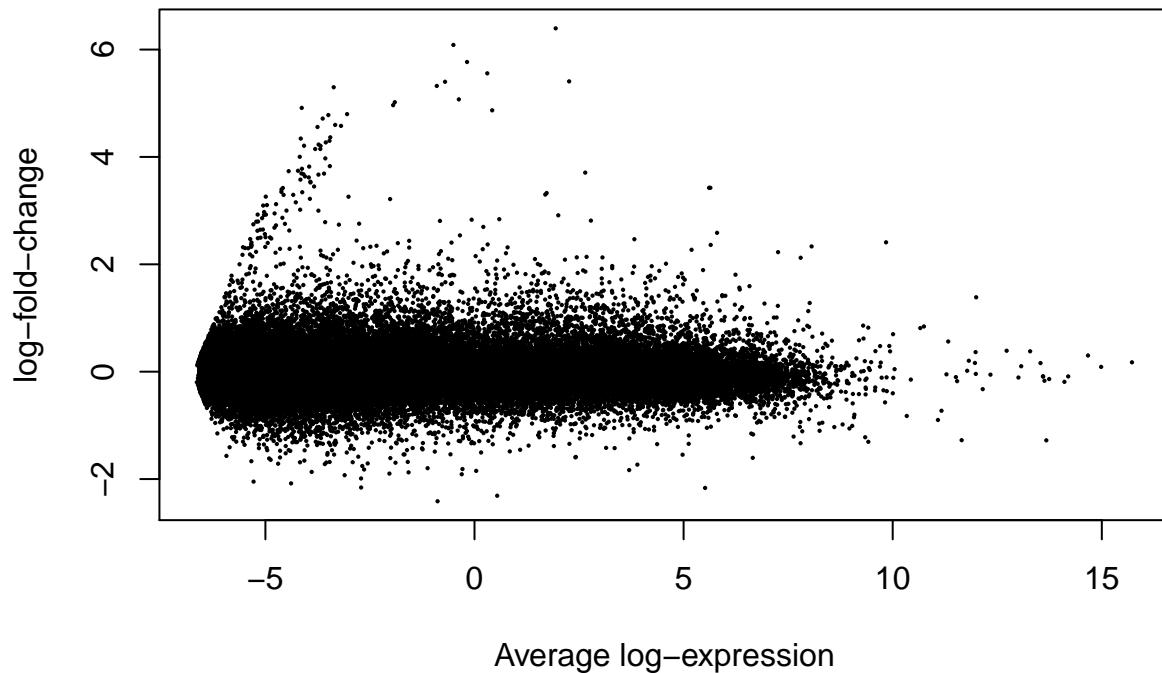
Se observa que 5376 genes de los que se tenían son los que tienen un p-value que indican que sí se expresan significativamente diferente.

```

## Visualizing expression differences in both groups
plotMA(eb_results, coef = 2)

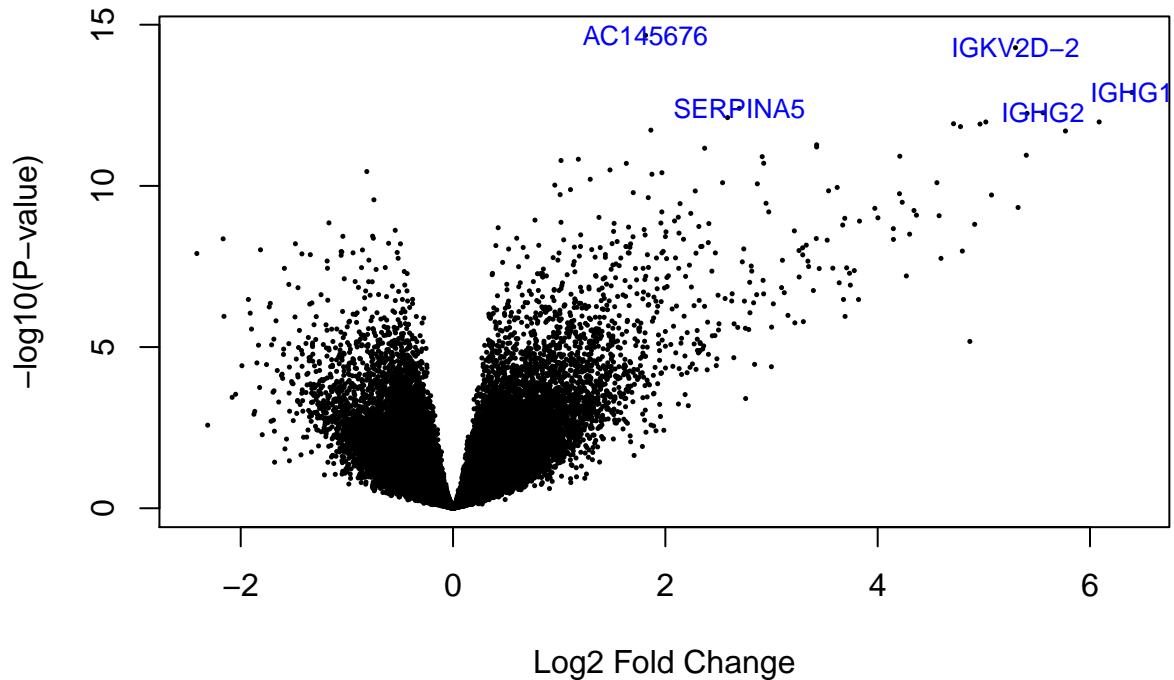
```

sra_attribute.disease_stateMS



En la gráfica anterior, se muestran las diferencias de expresión en grupos sanos y con esclerosis múltiple, valores positivos en el eje de las Y indican que hay mayor expresión de un gen en el grupo sano, mientras que valores negativos en el eje Y, indican mayor expresión en el grupo con esclerosis múltiple. En el eje X, el promedio de expresión de los genes. Básicamente la gráfica muestra que, a simple vista, hay un poco mas de genes expresados en el grupo sano que en el grupo con esclerosis múltiple, ya que ahí es donde hay más valores extremos (alejados del cero).

```
## Highlighting 5 genes with lower p-values
volcanoplot(eb_results, coef = 2, highlight = 5, names = de_results$gene_name)
```



En la gráfica de volcán, valores más alejados del cero en el eje Y indican menor p-value y, por lo tanto, mayor significancia estadística. En esta gráfica en específico, se muestran los 5 genes con p-value más pequeño. En el eje X simplemente el log fold change de los valores de expresión de cada gen.

```

##Extracting most important genes
exprs_heatmap <- vGene$E[rank(de_results$adj.P.Val) <= 50, ]

## creating data.frame
df <- as.data.frame(colData(rse_gene_SRP173190)[, c("sra_attribute.disease_state",
                                                       "sra_attribute.source_name")])
colnames(df) <- c("DiseaseState", "Tissue")
our_match <- which(rowRanges(rse_gene_SRP173190)$gene_id %in% rownames(exprs_heatmap))
rownames(exprs_heatmap) <- rowRanges(rse_gene_SRP173190)$gene_name[our_match]

## Creating pheatmap
library("pheatmap")

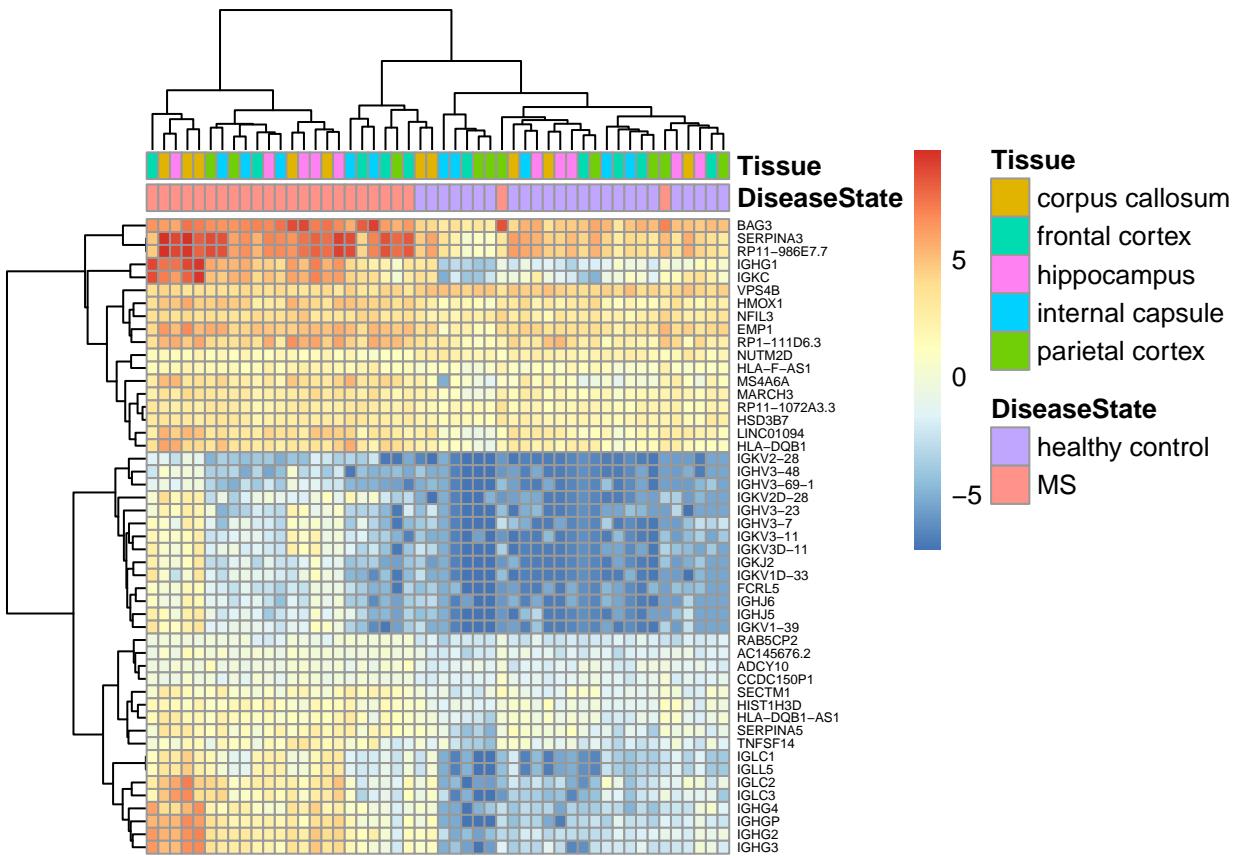
## Warning: package 'pheatmap' was built under R version 4.0.3
pheatmap(
  exprs_heatmap,
  cluster_rows = TRUE,
  cluster_cols = TRUE,
  show_rownames = TRUE,
  show_colnames = FALSE,

```

```

annotation_col = df,
fontsize_row = 5
)

```



En el pheatmap anterior se pueden observar los 50 genes con expresión significativamente diferente. Se observa, por ejemplo, que BAG3, SERPINA3 y RP11 son los genes más expresados tanto en pacientes sanos como en pacientes con esclerosis, aunque en los 3 casos, la expresión es mayor en pacientes con esclerosis (concordante con <https://nn.neurology.org/content/8/2/e941>). Y también, dos genes que resaltan a la vista, son IGHG1 y IGKC, ya que se observa que la expresión en pacientes con esclerosis múltiple es bastante mayor que en pacientes sanos.

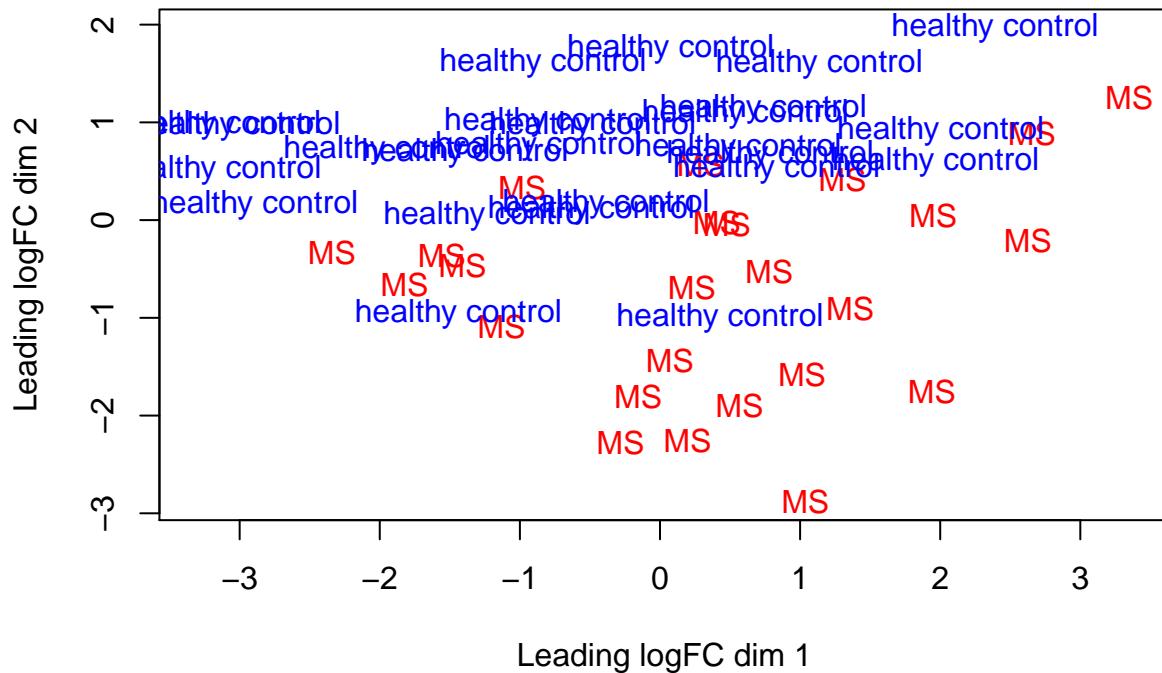
Ahora, para observar mejor si sí hay una diferencia de expresiones en nuestros dos grupos, se hace un Multidimensional Scaling.

```

library("RColorBrewer")

## Warning: package 'RColorBrewer' was built under R version 4.0.3
df <- as.data.frame(colData(rse_gene_SRP173190)[, c("sra_attribute.disease_state", "sra_attribute.source")]
## Disease state MDS
plotMDS(vGene$E, labels = df$sra_attribute.disease_state, col = rep(c("red", "blue"), each = 25))

```



En la gráfica anterior se puede observar como hay una separación de ambos grupos, dicha separación es un poco difusa en el centro, donde hay algunos puntos sobreapantes. Esto indica que si se buscan genes expresados diferencialmente entre pacientes con sanos y pacientes con esclerosis múltiple, es probable que algunos lo sean.

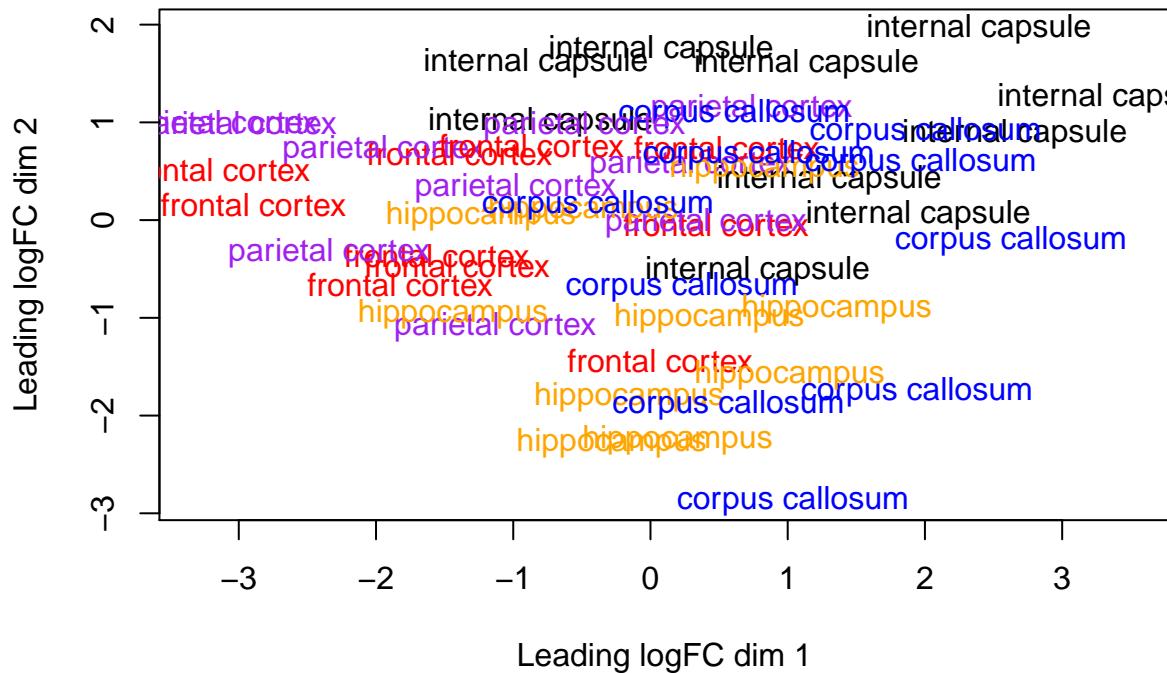
```
## Tissue MDS

col.tissue <- df$sra_attribute.source_name

col.tissue[col.tissue==unique(df$sra_attribute.source_name)] <- c("blue", "red",
                                                               "purple",
                                                               "orange", "black")

col.tissue[45:49] <- c("blue", "red", "purple", "orange", "black")

plotMDS(vGene$E, labels = df$sra_attribute.source_name, col = col.tissue)
```



En la gráfica anterior se muestra que los tejidos no forman ningún tipo de clúster específico. Es probable que al relacionar los tejidos con alguna otra variable desconocida se formen cúmulos que expliquen la forma de la distribución de los tejidos y por lo tanto, haya relación entre más variables desconocidas.

- Pequeña conclusión personal

En el trabajo presentado, me di cuenta de que la ciencia de datos es realmente compleja pero interesante. Hay muchísimas herramientas disponibles para facilitar el trabajo lo más posible de extracción de datos, cada una con muchísimas funcionalidades por explorar, por lo que estoy consciente de que esto no fue mas que una pequeña muestra de todo lo que se puede hacer con una muestra de datos y todas las posibles interpretaciones que puede haber. Realmente me gustaría conocer más a fondo las herramientas que hay para este tipo de análisis y aprender a usarlas profesionalmente.