Actividad 2- *From gene counts to DEG and pathways*

Objetivo

El propósito de esta actividad es que el estudiante demuestre la adquisición de habilidades y competencias necesarias para llevar a cabo un tratamiento estadístico adecuado sobre los datos de conteo de un experimento de RNA-seq. Esto implica determinar los genes expresados de manera diferencial entre diferentes escenarios y extraer los principales términos ontológicos enriquecidos.

Obtención de los datos

La actividad consiste en el análisis de una muestra real de RNA-seq, que forma parte de un estudio más amplio que examina los perfiles de expresión entre poblaciones de células basales y luminales en glándulas mamarias de ratones en diferentes estadios (vírgenes, gestantes y lactantes).

La publicación de referencia es la siguiente:

Fu, Nai Yang, et al. "EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival." Nature Cell Biology 17.4 (2015): 365-375.

Todo lo necesario para poder realizar esta actividad ha sido previamente abordado en clase y, por tanto, todo el material (incluyendo el entorno de trabajo conda 05MBIF y los archivos necesarios) puede encontrarse en la sección de **"Recursos y Materiales/Materiales del profesor/Proyecto_JULIO_2024".**

Formato de la entrega

- La entrega se realizará utilizando este documento como plantilla, que se convertirá a PDF y se adjuntará a la actividad correspondiente dentro de Campus VIU.
- Las respuestas se presentarán de forma clara y concisa, justificando su contenido.
- Se deberá adicionar SIEMPRE los comandos empleados, las capturas de pantalla que muestren su ejecución (eliminar todo aquello que no sea informativo) y los gráficos generados que apoyen sus repuestas.
- No es necesario explicar con detalle los comandos, opciones y/o argumentos empleados.
- Los gráficos o capturas de pantalla deberán tener una calidad y tamaño suficiente para su lectura.

Preguntas

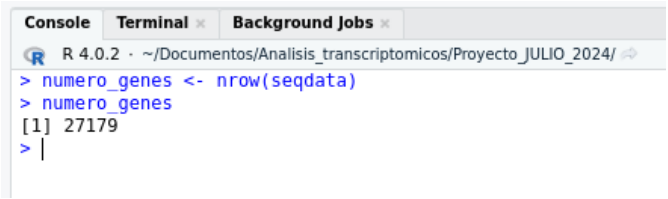
P1. Lee la matriz de recuentos llamada GSE60450_Lactation-GenewiseCounts.txt, selecciona únicamente las columnas que representan las muestras y reduce el nombre de las muestras tal y como hicimos en clase. Contesta a las siguientes preguntas que te permitirán conocer un poco mejor su contenido y estructura. Recuerda añadir siempre los comandos empleados y el resultado de su ejecución (2 pts).

- ¿Cuántos genes figuran en la matriz de recuentos?

  Comandos:

  ```
  numero_genes <- nrow(seqdata)
  numero_genes
  ```

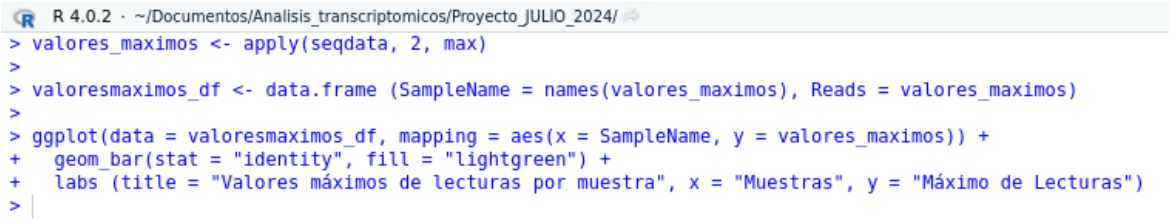
  Hay 27.179 genes en la matriz de recuentos.



- Genera un gráfico de barras con los valores máximos de lecturas para cada una de las muestras de estudio y comenta los resultados. ¿A qué gen se asocian en cada caso?

  Comandos:

  ```
  valores_maximos <- apply(seqdata, 2, max)
  valoresmaximos_df <- data.frame (SampleName = names(valores_maximos), Reads = valores_maximos)

  ggplot(data = valoresmaximos_df, mapping = aes(x = SampleName, y = valores_maximos)) + geom_bar(stat = "identity", fill = "lightgreen") + labs (title = "Valores máximos de lecturas por muestra", x = "Muestras", y = "Máximo de Lecturas")
  ```



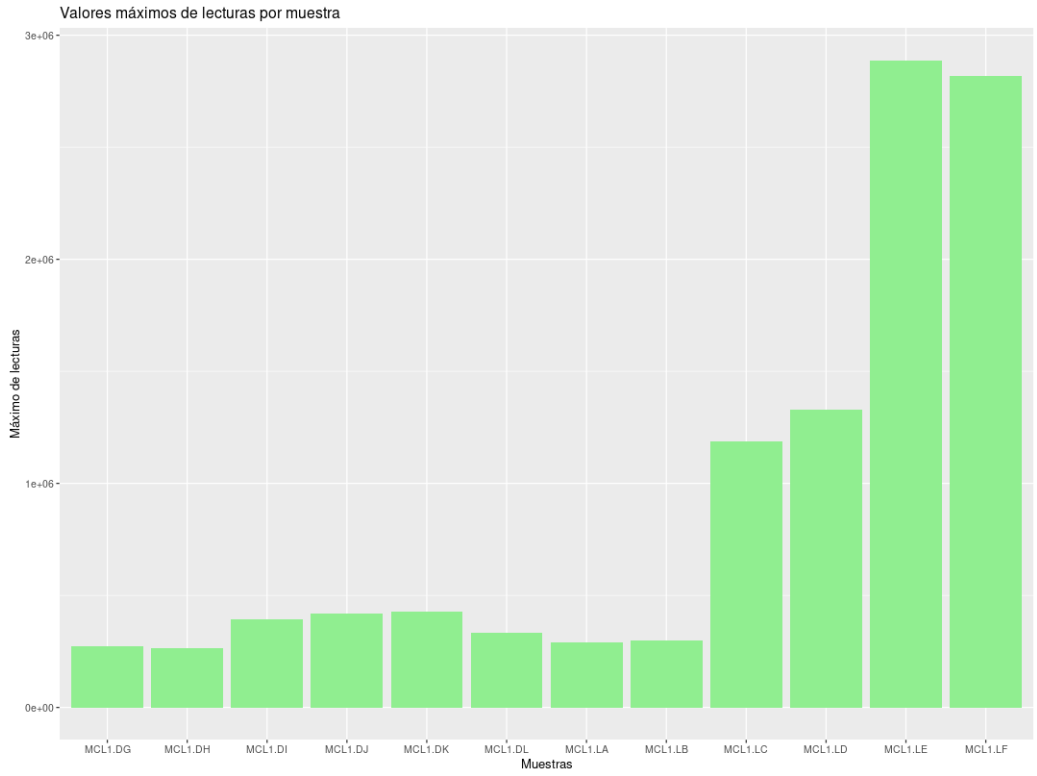


Figura 1: Gráfico de barras con los valores máximos de lecturas para cada muestra.

```
genes_vmax <- apply(seqdata, 2, function(x) rownames(seqdata)[which.max(x)])
```

```
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> genes_vmax <- apply(seqdata, 2, function(x) rownames(seqdata)[which.max(x)])
> genes_vmax
MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB MCL1.LC MCL1.LD MCL1.LE MCL1.LF
"17708" "17708" "11475" "11475" "11475" "11475" "17708" "17708" "12993" "12993" "12993" "12993"
>
```

El mayor número de lecturas por gen lo tienen las muestras MCL1.LF, MCL1.LE, MCL1.LD y MCL1.LC, que se trata del gen *Csn1s2a*, casein alpha s2-like A (*ENTREZID*: 12993), con entre 1 y 3 millones de lecturas.

En las muestras MCL1.LB, MCL1.LA, MCL1.DG y MCL1.DH, el gen que más lecturas tiene es *COX1*, cytochrome c oxidase subunit I (*ENTREZID*: 17708), y tiene entre 200.000 y 300.000 lecturas por muestra.

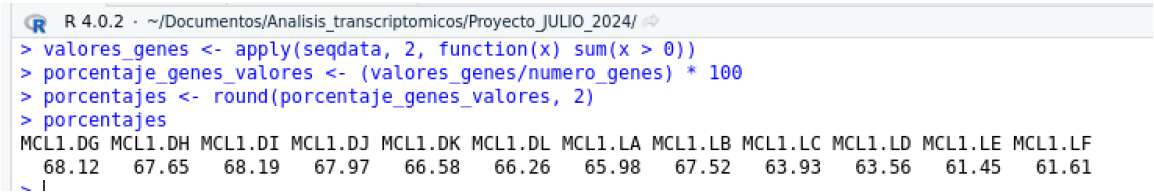
Por último, el gen que más lecturas tiene en las muestras MCL1.DI, MCL1.DJ, MCL1.DK y MCL1.DL es el gen *Acta2*, actin, alpha 2, smooth muscle, aorta (*ENTREZID:* 11475), con un número de lecturas por muestra entre 300.000 y 400.000.

Lo que se puede observar es que las muestras en las que coincide que el mismo gen tiene el mayor número de lecturas, tiene un número de lecturas similar en esas muestras, lo que podría estar relacionado o bien con el tipo de células o bien con el estadío en el que se encuentren.

- ¿Cuál es el porcentaje de genes que presentan un valor de conteos > 0 en cada una de las muestras? Comenta brevemente los resultados.

  Comandos:

  ```
  valores_genes <- apply(seqdata, 2, function(x) sum(x > 0))
  porcentaje_genes_valores <- (valores_genes/numero_genes) * 100
  porcentajes <- round(porcentaje_genes_valores, 2)
  ```



  Porcentaje de genes que presentan valores de conteos > 0 en cada muestra:

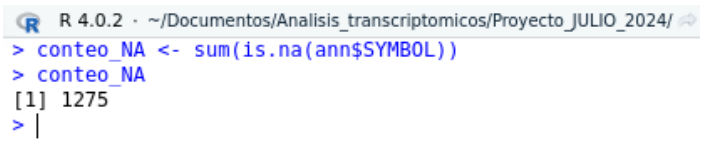
  MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB MCL1.LC MCL1.LD
  68.12 67.65 68.19 67.97 66.58 66.26 65.98 67.52 63.93 63.56
  MCL1.LE MCL1.LF
  61.45 61.61

  En general, se observa que todas las muestras muestran un porcentaje similar de genes con lecturas mayores que 0, entre el 60 – 70%. Además, se puede apreciar que los grupos de muestras del experimento presentan porcentajes más cercanos entre sí. Por ejemplo, el grupo de muestras de células luminales y en fase de lactancia, muestras MCL1.LE y MCL1.LF, tiene un 61 % de genes con lecturas mayores a 0, lo que podría estar relacionado.

- Al realizar la transformación de los identificadores *ENTREZID* de los genes a *SYMBOL* verás que algunos de ellos no se han podido convertir correctamente y la herramienta los ha sustituido por valores NA. ¿Cuántos puedes contabilizar? ¿Qué decisión tomarías respecto a los mismos?

  Comandos:

  ```
  conteo_NA <- sum(is.na(ann$SYMBOL))
  ```



  Hay 1.275 genes en los que no se ha podido realizar correctamente la transformación de *ENTREZID* a *SYMBOL*.

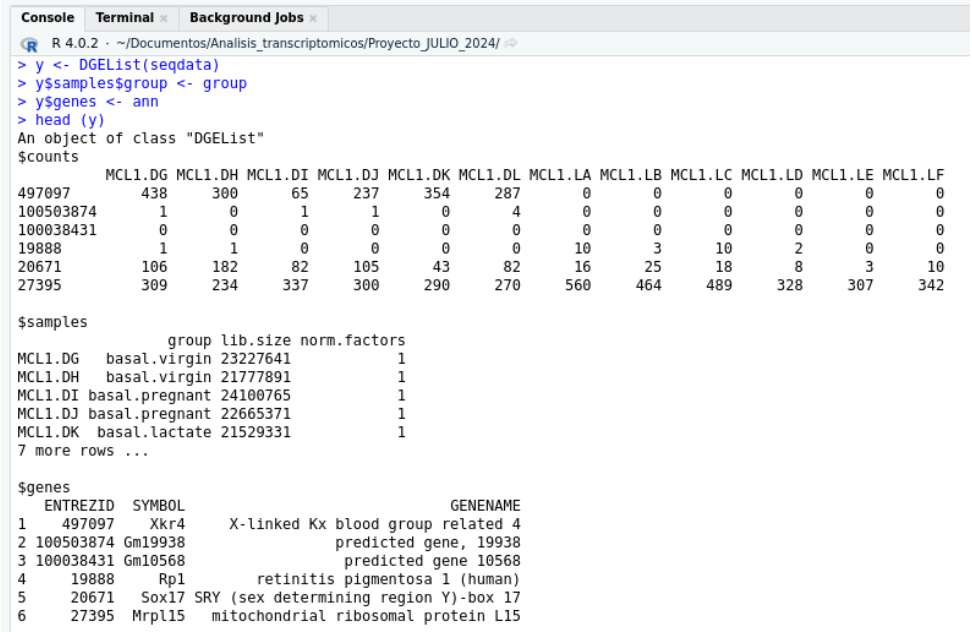
Habría que revisar los conteos en la matriz de recuento de esos genes que no se han identificado y si presentan niveles bajos de conteo se podrían eliminar, ya que su nivel de expresión es bajo. Si los niveles de expresión fueran altos podría valorarse mantenerlos en el análisis.

P2. Convierte la matriz de recuentos en un objeto DGEList y elimina todos los genes que presenten un conteo limitado empleado la función filterByExpr (1 pts).

- ¿Qué porcentaje de genes se mantienen en el *dataset* tras el filtrado?

  Comandos para convertir la matriz de recuentos en un objeto DGElist y eliminar los genes con un conteo limitado:

```
y <- DGEList(seqdata)
y$samples$group <- group
y$genes <- ann
```



```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> y <- DGEList(seqdata)
> y$samples$group <- group
> y$genes <- ann
> head (y)
An object of class "DGEList"
$counts
        MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB MCL1.LC MCL1.LD MCL1.LE MCL1.LF
497097      438     300      65     237     354     287       0       0       0       0       0       0
100503874     1       0       1       1       0       4       0       0       0       0       0       0
100038431     0       0       0       0       0       0       0       0       0       0       0       0
19888         1       1       0       0       0       0      10       3      10       2       0       0
20671       106     182      82     105      43      82      16      25      18       8       3      10
27395       309     234     337     300     290     270     560     464     489     328     307     342

$samples
               group lib.size norm.factors
MCL1.DG   basal.virgin 23227641            1
MCL1.DH   basal.virgin 21777891            1
MCL1.DI basal.pregnant 24100765            1
MCL1.DJ basal.pregnant 22665371            1
MCL1.DK  basal.lactate 21529331            1
7 more rows ...

$genes
   ENTREZID  SYMBOL                              GENENAME
1    497097    Xkr4     X-linked Kx blood group related 4
2 100503874 Gm19938                predicted gene, 19938
3 100038431 Gm10568                 predicted gene 10568
4     19888     Rp1        retinitis pigmentosa 1 (human)
5     20671   Sox17 SRY (sex determining region Y)-box 17
6     27395  Mrpl15   mitochondrial ribosomal protein L15
```

```
keep <- filterByExpr(y)
head(keep)
y <- y[keep, keep.lib.sizes=FALSE]
```

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> keep <- filterByExpr(y)
> y <- y[keep, keep.lib.sizes=FALSE]
> head(y)
An object of class "DGEList"
$counts
       MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB MCL1.LC MCL1.LD MCL1.LE MCL1.LF
497097     438     300      65     237     354     287       0       0       0       0       0       0
20671      106     182      82     105      43      82      16      25      18       8       3      10
27395      309     234     337     300     290     270     560     464     489     328     307     342
18777      652     515     948     935     928     791     826     862     668     646     544     581
21399     1604    1495    1721    1317    1159    1066    1334    1258    1068     926     508     500
58175        4       2      14       4       2       2     170     165     138      60      27      15

$samples
                group lib.size norm.factors
MCL1.DG   basal.virgin 23219195            1
MCL1.DH   basal.virgin 21769326            1
MCL1.DI basal.pregnant 24092719            1
MCL1.DJ basal.pregnant 22657703            1
MCL1.DK  basal.lactate 21522881            1
7 more rows ...

$genes
   ENTREZID SYMBOL                                GENENAME
1    497097   Xkr4         X-linked Kx blood group related 4
5     20671  Sox17     SRY (sex determining region Y)-box 17
6     27395 Mrpl15       mitochondrial ribosomal protein L15
7     18777 Lypla1                      lysophospholipase 1
9     21399  Tcea1 transcription elongation factor A (SII) 1
10    58175  Rgs20       regulator of G-protein signaling 20
```

Comandos para obtener el porcentaje de genes que se mantienen tras el filtrado:

```
numero_genesmantenidos <- nrow(y$counts)
porcentaje_genesmantenidos <- (numero_genesmantenidos/numero_genes) * 100
porcentaje_genesactual <- round(porcentaje_genesmantenidos, 2)
porcentaje_genesactual
```

```
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> numero_genesmantenidos <- nrow(y$counts)
> porcentaje_genesmantenidos <- (numero_genesmantenidos/numero_genes) * 100
> porcentaje_genesactual <- round(porcentaje_genesmantenidos, 2)
> porcentaje_genesactual
[1] 58.75
```

Se mantienen el 58,75% de los genes tras la realización del filtrado.

- Realiza un gráfico de cajas (*boxplot*) con los conteos transformados a logCPM antes de computar los factores de normalización y comenta su resultado.

Comandos:

```
transformacion_logCPM <- cpm(y, log = TRUE)
boxplot(transformacion_logCPM, col = "salmon",
        main = "Número de lecturas transformado a logCPM",
        ylab = "logCPM")
```

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> transformacion_logCPM <- cpm(y, log = TRUE)
> boxplot(transformacion_logCPM, col = "salmon",
+         main = "Número de lecturas transformado a logCPM",
+         ylab = "logCPM")
```

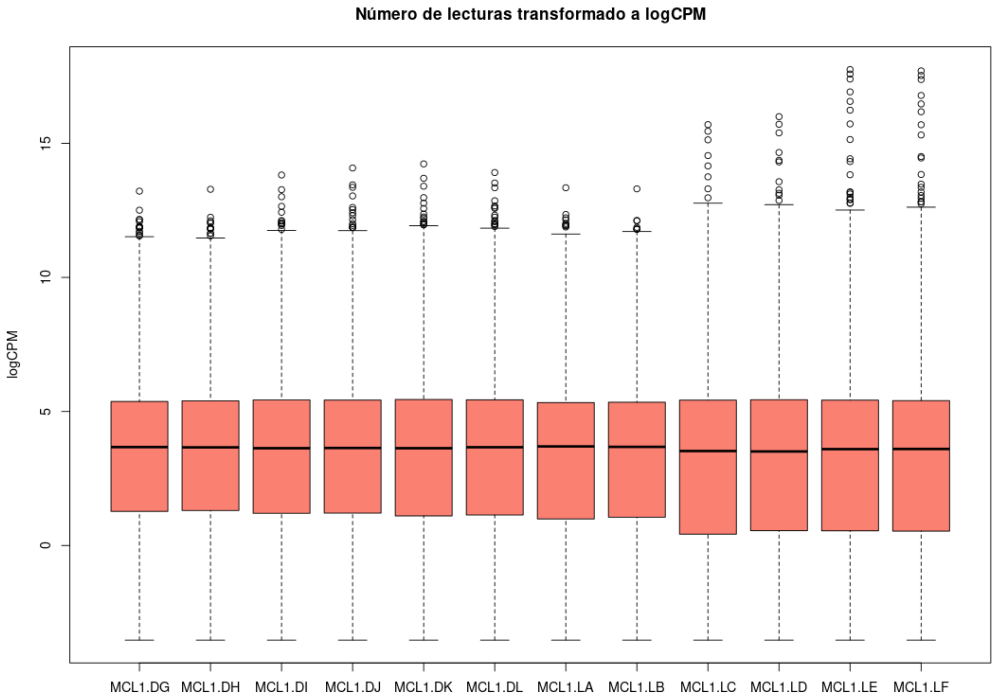


Figura 2: Boxplot del número de lecturas transformado a logCPM.

Se observa un valor de la mediana similar en casi todas las muestras, por lo que parece que el tamaño de la biblioteca es bastante similar entre las diferentes muestras. El rango intercuartílico y los bigotes de las cajas también es bastante similar, variando ligeramente en las últimas muestras, lo que podría deberse a diferencias en la dispersión de los datos o una mayor variabilidad en la expresión génica.

P3. Siguiendo el análisis con la muestra filtrada, computa sus factores de normalización mediante el método TMM. Recuerda añadir siempre los comandos empleados y el resultado de su ejecución y contesta las siguientes preguntas (1,5 pt).

- Realiza un gráfico de cajas (*boxplot*) con los conteos transformados a logCPM tras computar los factores de normalización y comenta su resultado.

  Comandos para la normalización de los datos mediante TMM:

  ```
  y <- calcNormFactors(y)
  ```

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> y <- calcNormFactors(y)
> head(y)
An object of class "DGEList"
$counts
       MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB MCL1.LC MCL1.LD MCL1.LE MCL1.LF
497097     438     300      65     237     354     287       0       0       0       0       0       0
20671      106     182      82     105      43      82      16      25      18       8       3      10
27395      309     234     337     300     290     270     560     464     489     328     307     342
18777      652     515     948     935     928     791     826     862     668     646     544     581
21399     1604    1495    1721    1317    1159    1066    1334    1258    1068     926     508     500
58175        4       2      14       4       2       2     170     165     138      60      27      15

$samples
                group lib.size norm.factors
MCL1.DG   basal.virgin 23219195     1.238429
MCL1.DH   basal.virgin 21769326     1.213792
MCL1.DI basal.pregnant 24092719     1.124800
MCL1.DJ basal.pregnant 22657703     1.071183
MCL1.DK  basal.lactate 21522881     1.036279
7 more rows ...

$genes
   ENTREZID SYMBOL                                GENENAME
1    497097   Xkr4         X-linked Kx blood group related 4
5     20671  Sox17     SRY (sex determining region Y)-box 17
6     27395 Mrpl15       mitochondrial ribosomal protein L15
7     18777 Lypla1                      lysophospholipase 1
9     21399  Tcea1 transcription elongation factor A (SII) 1
10    58175  Rgs20       regulator of G-protein signaling 20
```

Comandos para la realización del boxplot:

```
logCPM_normalizado <- cpm(y, log=TRUE)
boxplot(logCPM_normalizado, col = "lightblue",
        main = "Número de lecturas normalizado",
        ylab = "datos logCPM normalizados")
```

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> logCPM_normalizado <- cpm(y, log=TRUE)
> boxplot(logCPM_normalizado, col = "lightblue",
+         main = "Número de lecturas normalizado",
+         ylab = "datos logCPM normalizados")
```

- ¿Qué muestras presentan un factor de normalización (norm.factors) menor de 1 y qué puede significar esto?

Comandos:

```
factor_norm <- y$samples[y$samples$norm.factors < 1,]
factor_norm
```

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> factor_norm <- y$samples[y$samples$norm.factors < 1,]
> factor_norm
                group lib.size norm.factors
MCL1.LD luminal.pregnant 21983364    0.9226086
MCL1.LE  luminal.lactate 24720123    0.5292854
MCL1.LF  luminal.lactate 24653390    0.5353884
```

Las muestras MCL1.LD, MCL1.LE, MCL1.LF que corresponden todas al tipo celular luminal, son las que presentan un factor de normalización menor de 1. Esto quiere decir que tienen un nivel de expresión global más alto que el resto de muestras porque habrá una pequeña cantidad de genes que tengan un número alto de conteos, que puede hacer que el conteo de otros genes sea más bajo de lo normal debido al tamaño de la librería.

Que coincida que todas las muestras que presentan el factor de normalización menor de 1 sean las del tipo celular luminal puede reflejar una clara diferencia en la expresión de ciertos genes entre las células de tipo luminal y las células de tipo basal. Aunque también podría deberse a errores en la técnica o sesgos por lo que hay que hacer más análisis para ver si son realmente diferencias biológicas.

- Genera un gráfico de MDS en el que cada tipo celular este coloreado de un color distintivo (*basal* vs *luminal*) y las muestras *lactate* se presenten con un círculo lleno, las muestras *pregnant* con un triángulo lleno y las muestras *virgin* con un cuadrado lleno. Comenta brevemente cómo se asocian o diferencian dichas muestras.

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> tipocelular <- ifelse(grepl("basal", group), "basal", "luminal")
> colors <- ifelse(tipocelular == "basal", "red", "blue")
> formas <- ifelse(grepl("virgin", group),15, ifelse(grepl("pregnant", group), 17, 16))
> plotMDS(y, col=colors, pch=formas, cex=1)
> legend("top", legend=levels(group), pch=c(16,17,15,16,17,15), col=c("red","red","red","blue","blue","blue"), title="Leyenda", ncol=1, cex=0.
8)
>
```

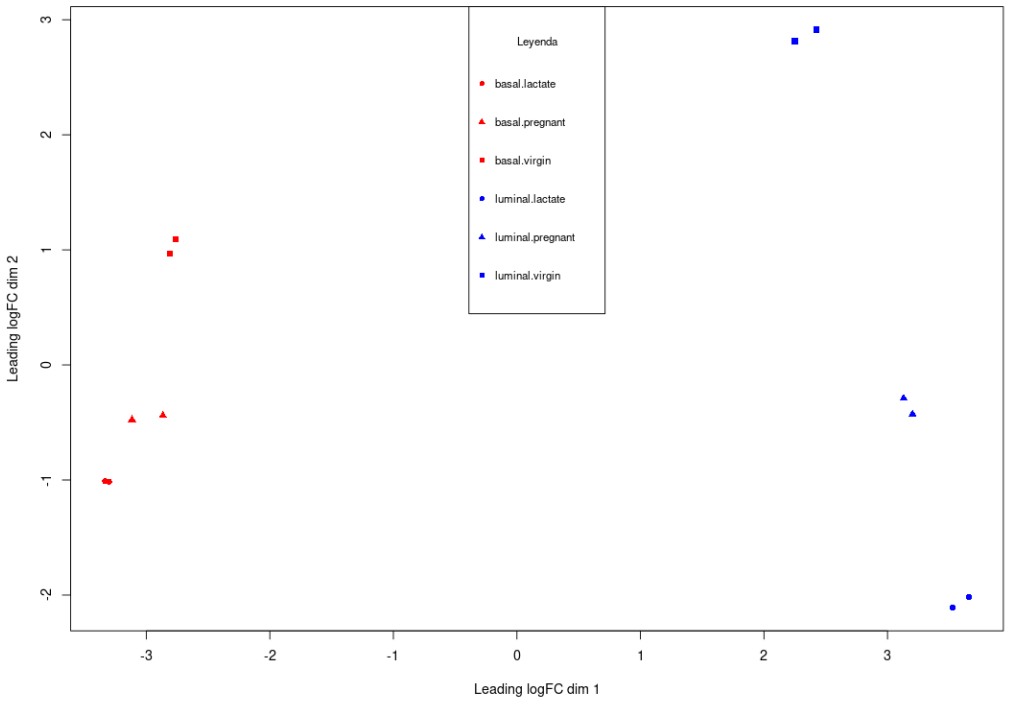


Figura 3: Gráfico MDS con cada grupo de muestras diferenciado por color y forma. Grupo basal en rojo, grupo luminal en azul. Estadío *lactate* representado por círculo, estadío *pregnant* representado por triangulo y estadío *virgin* representado por cuadrado.

Coincidiendo con los resultados del apartado anterior se observa que hay diferencias entre los grupos de células basales, situadas en el lado izquierdo, y el grupo de células luminales, situadas en el lado derecho del gráfico, como cabría esperar, reflejando diferencias biológicas.
Como también cabía esperar, las réplicas biológicas se agrupan muy juntas, ya que en teoría tienen características biológicas similares.
También se observan las asociaciones de los diferentes grupos de muestra en función de los estadíos, habiendo una marcada diferencia entre los grupos en estadío virgen situados en la parte superior por encima del 0, y el resto de grupo situados en la parte inferior por debajo del 0, por lo que los grupos en estadío de lactancia y embarazo tienen más similitud entre sí.
La clara mayor variabilidad entre los datos está asociada muestras según el tipo celular y después según el estadío, como cabría esperar, por lo que da una idea de que las diferencias biológicas entre los grupos principales están bien representadas en los datos y eran las esperadas a analizar.
Las muestras que se observan más dispares al resto son las del tipo celular luminal y estadío virgen.

P4. Calcula los tres tipos de dispersión en el objeto DGEList creado. Recuerda añadir siempre los comandos empleados y el resultado de su ejecución y contesta las siguientes preguntas (0,5 pt).

Comandos empleados para calcular los tres tipos dispersión en el objeto DGEList:

```
y <- estimateDisp(y, design, robust = TRUE)

y$common.dispersion

head(y$trended.dispersion)

head(y$tagwise.disperision)
```

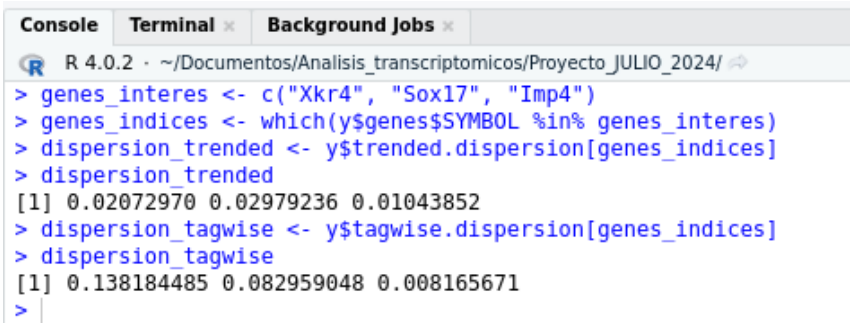
```
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> y <- estimateDisp(y, design, robust = TRUE)
> y$common.dispersion
[1] 0.01344131
> head(y$trended.dispersion)
[1] 0.020729697 0.029792363 0.012993147 0.010059824 0.009572663 0.031347386
> head(y$tagwise.dispersion)
[1] 0.138184485 0.082959048 0.013816341 0.006749923 0.006357810 0.069904043
```

- ¿Cuáles son los valores de “trended dispersion”, y “tagwise dispersion” para los genes ***Imp4***, ***Xkr4*** y ***Sox17***?

Comandos empleados para obtener los valores de trended dispersión y tagwise dispersión de los genes Imp4, Xkr4 y Sox17:

```
genes_interes <- c (“Xkr4”, Sox17”, “Imp4”)
```

```
genes_indices <- which(y$genes$SYMBOL %in% genes_interes)

dispersion_trended <- y$trended.dispersion[genes_indices]

dispersion_trended

dispersion_tagwise <- y$tagwise.dispersion[genes_indices]

dispersion_tagwise
```



Valores de "trended dispersion" para:

Xkr4: 0.02072970

Sox17:0.02979236

Imp4: 0.01043852

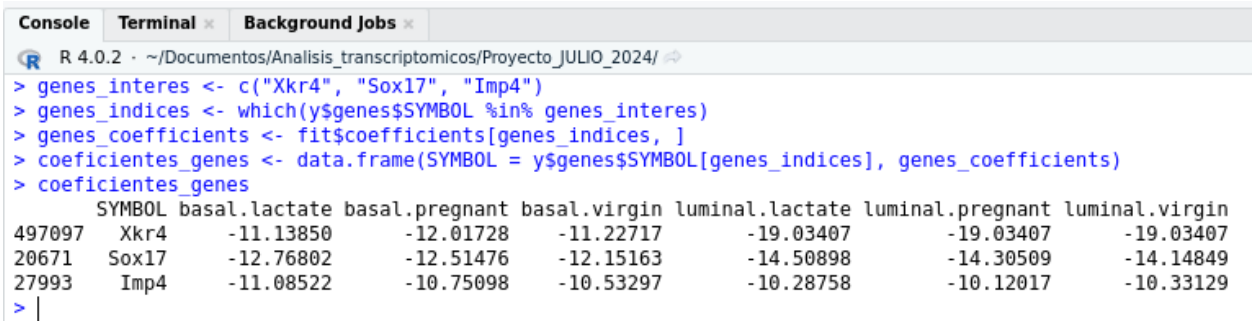
Valores de "tagwise dispersion" para:

Xkr4: 0.138184485

Sox17: 0.082959048

Imp4: 0.008165671

- Después del ajuste de GLM con la función *glmQLFit*, ¿qué valores tienen los *coefficients* de los seis grupos experimentales a estudio para los genes ***Imp4***, ***Xkr4*** y ***Sox17***?



Comandos para obtener los valores que tienen el coefficients de los 6 grupos experimentales de los genes Imp4, Xkr4, y Sox17:

```
genes_coefficients <- fit$coefficients[genes_indices, ]

coeficientes_genes <- data.frame(SYMBOL = y$genes$SYMBOL[genes_indices], genes_coefficients)

coeficientes_genes
```

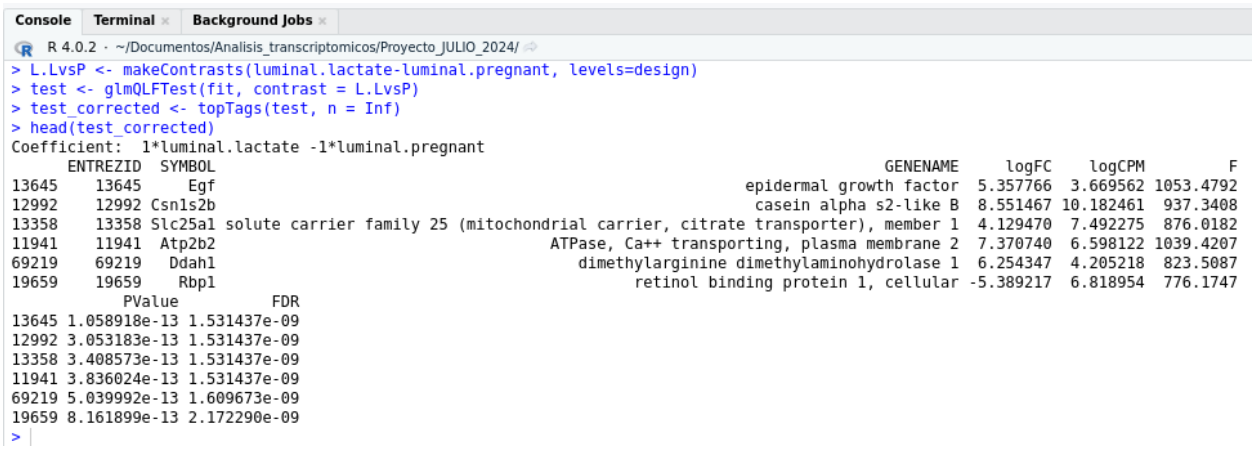
Valores obtenidos de los coefficients para los 6 grupos experimentales de los genes:

SYMBOL	basal.lactate	basal.pregnant	basal.virgin	luminal.lactate	luminal.pregnant	luminal.virgin
Xkr4	-11.13850	-12.01728	-11.22717	-19.03407	-19.03407	-19.03407
Sox17	-12.76802	-12.51476	- 12.15163	-14.50898	-14.30509	-14.14849
Imp4	-11.08522	-10.75098	-10.53297	-10.28758	-10.12017	-10.33129

P5. Utiliza la función *glmQLFTest* para realizar la prueba de hipótesis nula en la siguiente comparación:

- ***Luminal.lactate vs luminal.pregnant***

Utiliza la función *topTags* sobre la prueba realizada anteriormente y observa cómo se genera una nueva columna llamada FDR. Recuerda añadir siempre los comandos empleados y el resultado de su ejecución y contesta a las siguientes preguntas (2,5 pts).



```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> L.LvsP <- makeContrasts(luminal.lactate-luminal.pregnant, levels=design)
> test <- glmQLFTest(fit, contrast = L.LvsP)
> test_corrected <- topTags(test, n = Inf)
> head(test_corrected)
Coefficient:  1*luminal.lactate -1*luminal.pregnant
      ENTREZID  SYMBOL                                                                  GENENAME     logFC    logCPM         F
13645    13645     Egf                                                      epidermal growth factor  5.357766  3.669562 1053.4792
12992    12992 Csn1s2b                                                       casein alpha s2-like B  8.551467 10.182461  937.3408
13358    13358 Slc25a1 solute carrier family 25 (mitochondrial carrier, citrate transporter), member 1  4.129470  7.492275  876.0182
11941    11941  Atp2b2                                        ATPase, Ca++ transporting, plasma membrane 2  7.370740  6.598122 1039.4207
69219    69219   Ddah1                                           dimethylarginine dimethylaminohydrolase 1  6.254347  4.205218  823.5087
19659    19659    Rbp1                                                 retinol binding protein 1, cellular -5.389217  6.818954  776.1747
            PValue          FDR
13645 1.058918e-13 1.531437e-09
12992 3.053183e-13 1.531437e-09
13358 3.408573e-13 1.531437e-09
11941 3.836024e-13 1.531437e-09
69219 5.039992e-13 1.609673e-09
19659 8.161899e-13 2.172290e-09
>
```

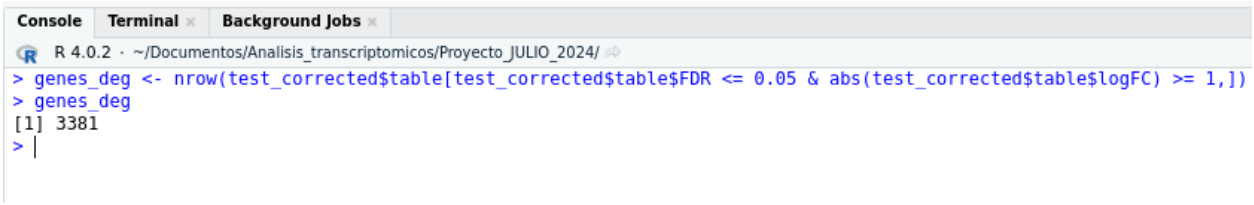
Comandos empleados para realizar la prueba de hipótesis:

```
L.LvsP <- makeContrasts(luminal.lactate-luminal.pregnant, levels=design)

test <- glmQLFTest(fit, contrast = L.LvsP)

test_corrected <- topTags(test, n = Inf)
```

```
head(test_corrected)
```

- ¿Cuántos genes diferencialmente expresados (DEG), aplicando un punto de corte de FDR <= 0.05 y un valor absoluto de logFC >= 1, encuentras?
  Comandos utilizados:

```
genes_deg <- nrow(test_corrected$table[test_corrected$table$FDR <= 0.05 & abs(test_corrected$table$logFC) >= 1,])
genes_deg
```



El número de genes diferencialmente expresados aplicando los criterios de punto de corte de FDR <= 0.05 y un valor absoluto de logFC >= 1 es de 3.381 genes.

- Dibuja un gráfico de Volcán con el resultado obtenido y etiqueta los 3 genes con mayor valor absoluto de LogFC. ¿Cuáles son?
  Comandos para la realización del gráfico Volcán y el etiquetado de los 3 genes con mayor valor absoluto:

```
data <- test_corrected$table
data$DE <- "NO"
data$DE[data$logFC > 1 & data$FDR < 0.05] <- "UP"
data$DE[data$logFC < 1 & data$FDR < 0.05] <- "DOWN"

data$logFC_abs <- abs(data$logFC)
genes_top <- data[order(-data$logFC_abs), ][1:3, ]

ggplot(data, aes(x=logFC, y= -log10(FDR), col=DE)) + geom_point(size=0.5) + theme_minimal() +
 geom_vline(xintercept = c(-1,1), col ="red", linetype = "dashed") +
 geom_hline(yintercept = -log10(0.05), col ="green", linetype = "dashed") +
 geom_text(data=genes_top, aes(x=logFC, y=-log10(FDR), label=SYMBOL),
          vjust=-1, hjust=0.5, size= 5, col="black")
```

```
Console  Terminal  Background Jobs
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> data <- test_corrected$table
> data$DE <- "NO"
> data$DE[data$logFC > 1 & data$FDR < 0.05] <- "UP"
> data$DE[data$logFC < 1 & data$FDR < 0.05] <- "DOWN"
> data$logFC_abs <- abs(data$logFC)
> genes_top <- data[order(-data$logFC_abs), ][1:3, ]
> ggplot(data, aes(x=logFC, y= -log10(FDR), col=DE)) + geom_point(size=0.5) + theme_minimal() +
+   geom_vline(xintercept = c(-1,1), col ="red", linetype = "dashed") +
+   geom_hline(yintercept = -log10(0.05), col ="green", linetype = "dashed") +
+   geom_text(data=genes_top, aes(x=logFC, y=-log10(FDR), label=SYMBOL),
+             vjust=-1, hjust=0.5, size= 5, col="black")
> genes_top
      ENTREZID  SYMBOL                                                    GENENAME      logFC     logCPM         F       PValue
20519    20519 Slc22a3 solute carrier family 22 (organic cation transporter), member 3 -10.050695  0.7977744  79.34347 1.543470e-06
18546    18546    Pcp4                                      Purkinje cell protein 4   9.351746  3.9705583 211.81439 2.341577e-08
20608    20608   Sstr4                                      somatostatin receptor 4   9.189792 -1.2579215  96.18317 2.162050e-06
              FDR   DE logFC_abs
20519 2.035315e-05 DOWN 10.050695
18546 9.701885e-07   UP  9.351746
20608 2.597876e-05   UP  9.189792
>
```

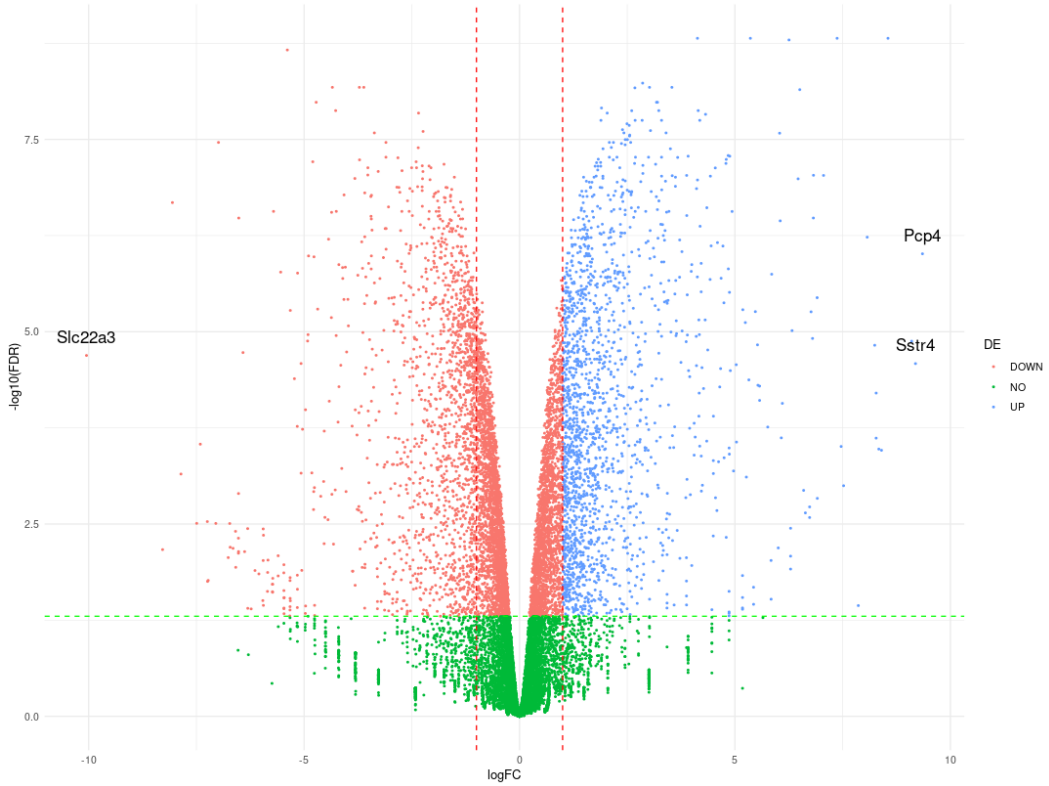


Figura 4: Gráfico de Volcán con los genes diferencialmente expresados, con los 3 genes de mayor valor absoluto etiquetados con el nombre correspondiente.

Los 3 genes con mayor valor absoluto de LogFC son: Slc22a3 (miembro 3 de la familia 22 de transportadores de solutos (transportador de cationes orgánicos)), Pcp4 (proteína celular de Purkinje 4), y Sstr4 (receptor de somatostatina 4).

- ¿Cuántos genes son únicos y compartidos (diferencia entre sobreexpresados e infraexpresados) puedes encontrar entre esta comparación y aquella que realizamos en clase entre *basal.lactate vs basal.pregnant*?. Para ello realiza un diagrama de Venn o un Upset plot donde muestres los resultados obtenidos. Recuerda emplear los datos obtenidos tras aplicar los puntos de corte especificados en el punto 1 de esta pregunta.

Comandos empleados para la realización del Upset plot:

```
genes_luminal_up <- rownames(test_corrected$table[test_corrected$table$logFC > 1 & test_corrected$table$FDR <= 0.05, ])

genes_luminal_down <- rownames(test_corrected$table[test_corrected$table$logFC < -1 & test_corrected$table$FDR <= 0.05, ])


genes_basal_up <- rownames(res_corrected$table[res_corrected$table$logFC > 1 & res_corrected$table$FDR <= 0.05, ])

genes_basal_down <- rownames(res_corrected$table[res_corrected$table$logFC < -1 & res_corrected$table$FDR <= 0.05, ])


lista_genes <- list(
  Luminal_Up = genes_luminal_up,
  Luminal_Down = genes_luminal_down,
  Basal_Up = genes_basal_up,
  Basal_Down = genes_basal_down
)

upset_data <- fromList(lista_genes)

upset(upset_data,
      order.by = "freq",
      main.bar.color = c("darkred", "darkblue", "darkblue","darkred", "blue", "black", "red", "black"),
      sets.bar.color = c("darkred", "darkblue", "darkblue", "darkred"),
      text.scale = 1.5)
```

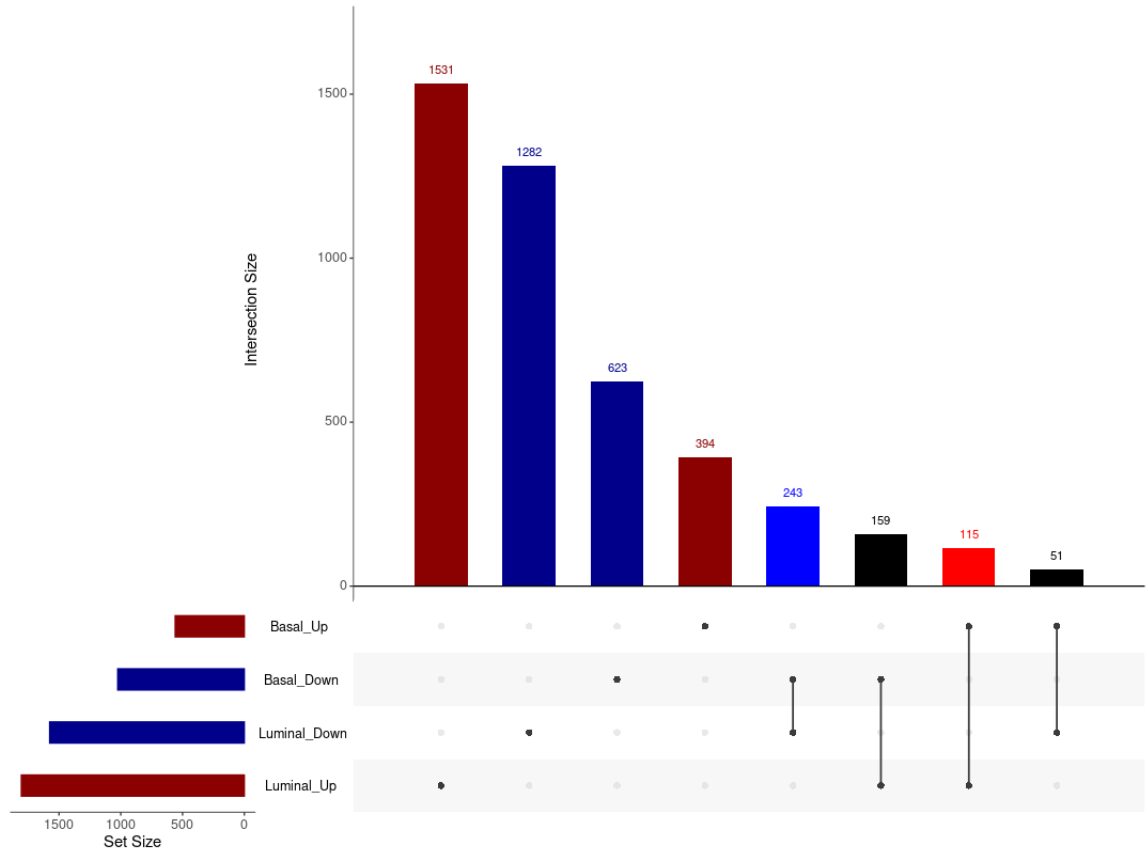


Figura 5: Upset plot que representa los genes únicos y compartidos entre los distintos grupos basal y luminal, en función de si están sobreexpresados o infraexpresados.

El grupo luminal tiene el mayor número de genes únicos sobreexpresados, 1531 genes (luminal_up), y el mayor número de genes únicos infraexpresados, 1282 genes (luminal_down), en el estadio *lactate*, respecto a *pregnant*. Para el grupo basal hay 394 genes que solo se sobreexpresan en este grupo (basal_up) en el estadio *lactate, respecto a pregnant*, mientras que 623 genes se encuentran infraexpresados (luminal_up). Es decir, que las células luminales en estadío *lactate* hay un mayor número de genes únicos sobreexpresados respecto al estadío *basal*, mientras que en las basales hay más genes únicos infraexpresados que sobreexpresados en el estadío lactate respecto al basal.
Los números de genes compartidos entre los grupos son más bajos. Hay 243 genes infraexpresados que son compartidos entre ambos grupos basal y luminal (basal_down y luminal_down). Y 115 genes compartidos en todos los grupos, que puede reflejar la variabilidad de expresión de estos genes según el estadio celular.

P6. Transforma la matriz de conteos completa a datos normalizados (en logcpm). Recuerda añadir siempre los comandos empleados y el resultado de su ejecución (1,5 pts).

- Filtra la matriz completa tomando únicamente los genes compartidos entre los tipos celulares *basal* y *luminate* que se encuentren sobreexpresados en ambas líneas celulares.

Comandos empleados:

logCPM <- cpm(y, log=TRUE)

rownames(logCPM) <- y$genes$SYMBOL

colnames(logCPM) <- paste(y$samples$group, 1:2, sep = "-")

DEG_luminal <-test_corrected$table[test_corrected$table$logFC > 1 & test_corrected$table$FDR <= 0.05,]

DEG_basal <-res_corrected$table[res_corrected$table$logFC > 1 & res_corrected$table$FDR <= 0.05,]

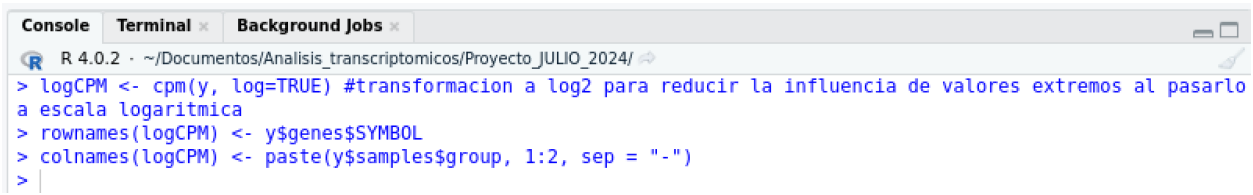
genes_luminal_up <- DEG_luminal$SYMBOL

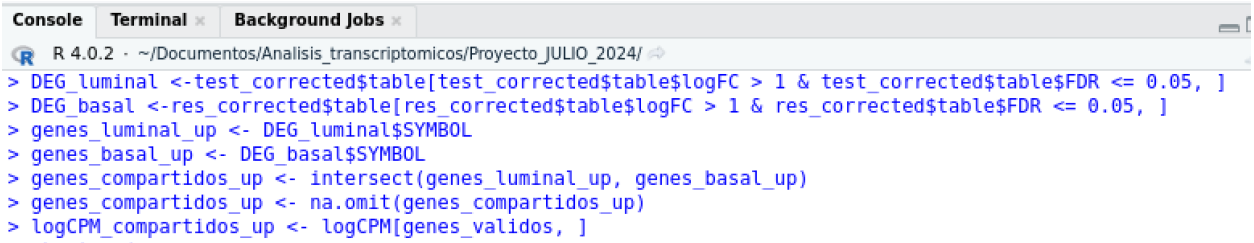
genes_basal_up <- DEG_basal$SYMBOL

genes_compartidos_up <- intersect(genes_luminal_up, genes_basal_up)

genes_compartidos_up <- na.omit(genes_compartidos_up)

logCPM_compartidos_up <- logCPM[genes_validos,]





- Genera un mapa de calor empleando la función *pheatmap* y comenta tu resultado. ¿Cómo se organizan las muestras? ¿El resultado (a nivel muestral) es el mismo si aplicas otro método de aglomeramiento o *linkage*?

Comandos empleados empleados para la realización del pheatmap con el método “ward.D2”:

pheatmap(logCPM_compartidos_up, scale = "row", cluster_rows = TRUE, cluster_cols = TRUE, clustering_method = "ward.D2", cutree_cols = 2, cutree_rows = 2, fontsize_number = 6,

fontsize_row = 6, fontsize_col = 8, angle_col = 45, color = colorRampPalette(c("blue", "white", "red"))(100), main = "Mapa de calor: Genes sobreexpresados y compartidos"

)

```
> pheatmap(
+   logCPM_compartidos_up,
+   scale = "row",
+   cluster_rows = TRUE,        # Permite agrupar las filas (genes)
+   cluster_cols = TRUE,        # Agrupa las columnas (muestras)
+   clustering_method = "ward.D2",
+   cutree_cols = 2,
+   cutree_rows = 2,
+   fontsize_number = 6,
+   fontsize_row = 6,           # Tamaño de texto para los genes
+   fontsize_col = 8,           # Tamaño de texto para las muestras
+   angle_col = 45,             # Rotar nombres de las columnas
+   color = colorRampPalette(c("navy", "white", "firebrick3"))(100),
+   main = "Mapa de calor: Genes sobreexpresados y compartidos"
+ )
>
```

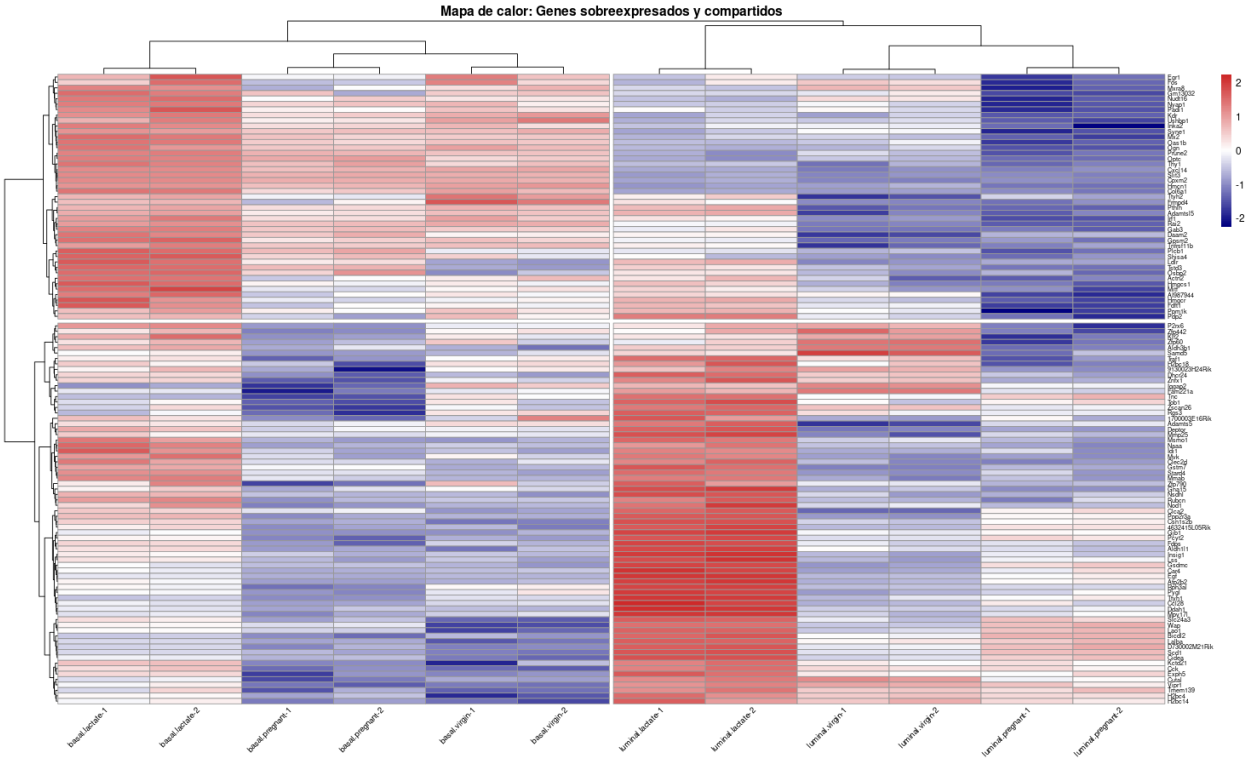


Figura 6: Mapa de calor, método “ward.D2”, de los genes sobreexpresados y compartidos entre los tipos celulares basal y luminal.

Las muestras se agrupan según la similitud de expresión los genes seleccionados, por lo que las muestras que son réplicas de los mismos grupos muestrales se agrupan juntan, ya que al ser réplicas biológicas es esperable que los resultados sean muy similares. Por otro lado, con los genes seleccionados para el análisis, genes compartidos que se encuentren sobreexpresados en ambas líneas celulares, se agrupan juntos los grupos *pregnant* y *virgin*, manteniéndose separado *lactate*, tanto en basal como luminal. Esto indica que las muestras de estadío *lactate* sugieren un perfil de expresión genética distinto, en comparación con *pregnant* y *virgin,* que comparten un perfil de expresión genética más similar.

Comandos empleados para la realización del pheatmap con el método “average”:

pheatmap(logCPM_compartidos_up, scale = "row", cluster_rows = TRUE, cluster_cols = TRUE,

clustering_method = "average", cutree_cols = 2, cutree_rows = 2, fontsize_number = 6,

fontsize_row = 6, fontsize_col = 8, angle_col = 45, color = colorRampPalette(c("blue", "white", "red"))(100), main = "Mapa de calor: Genes sobreexpresados y compartidos"

)

```
Console   Terminal ×   Background Jobs ×
R 4.0.2 · ~/Documentos/Analisis_transcriptomicos/Proyecto_JULIO_2024/
> pheatmap(
+   logCPM_compartidos_up,
+   scale = "row",
+   cluster_rows = TRUE,        # Permite agrupar las filas (genes)
+   cluster_cols = TRUE,        # Agrupa las columnas (muestras)
+   clustering_method = "average",
+   cutree_cols = 2,
+   cutree_rows = 2,
+   fontsize_number = 6,
+   fontsize_row = 6,           # Tamaño de texto para los genes
+   fontsize_col = 8,           # Tamaño de texto para las muestras
+   angle_col = 45,             # Rotar nombres de las columnas
+   color = colorRampPalette(c("blue", "white", "red"))(100),
+   main = "Mapa de calor: Genes sobreexpresados y compartidos"
+ )
>
```

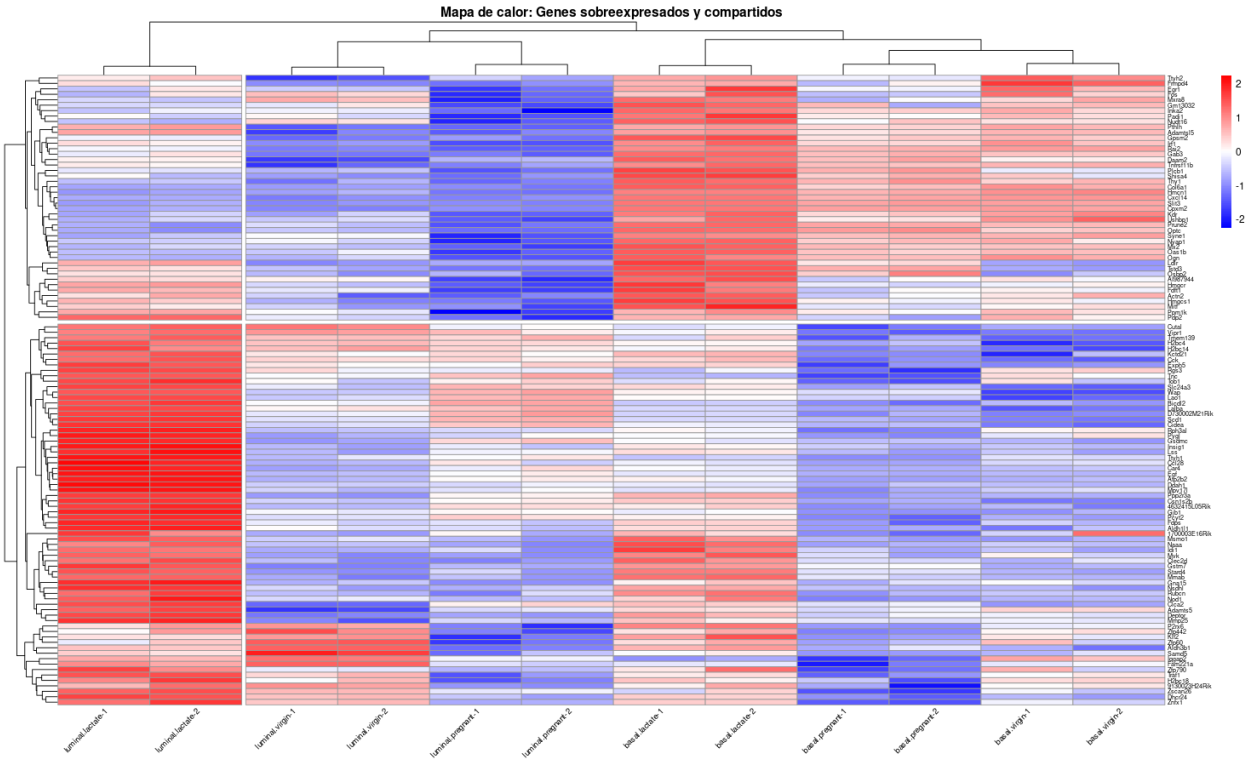


Figura 7: Mapa de calor, método "average", de los genes sobreexpresados y compartidos entre los tipos celulares basal y luminal.

Con el método de *linkage* "average" el resultado a nivel muestral es el mismo, sigue agrupando los grupos v*irgin y pregnant* y mantiene separado el grupo *lactate* tanto en basal como en luminal. Lo único que parece cambiar es que las ramas son menos compactas con el método "average" que con el método "ward.D2" y que cambia de sitio los grupos de muestras basales y luminales, que ahora los luminales están a la izquierda y antes no.

P7. Finalmente, selecciona el grupo de genes anterior y realiza un análisis de enriquecimiento funcional indicando los 10 primeros resultados que reporta topGO en las categorías BP y MF. Comenta con detalle los resultados obtenidos y añade referencias bibliográficas si es necesario (1 pts).