

Resolución Actividad 2, Secuenciación y Ómicas de Próxima Generación, Máster Bioinformática UNIR (2025). Análisis de expresión diferencial de genes relacionados con la obesidad mediante RNA-seq

Oriana Batista Ceballos

2025-12-21

I. Introducción

- En este se introduce el script 2 con la visualización de la expresión diferencial génica a través de un gráfico Heatmap.

I. Introducción

```
# CARGAR LIBRERÍAS
library(tximport)
library(DESeq2)
library(ggplot2)
library(pheatmap)
library(dplyr)

# 1. CONFIGURACIÓN DE ARCHIVOS Y METADATOS
samples <- c("AbrahamSimpson", "HomerSimpson", "MargeSimpson", "PattyBouvier", "SelmaBouvier")
files <- file.path("Salmon", samples, "quant.sf")
names(files) <- samples

# 2. RESUMEN DE CUANTIFICACIÓN (Parte A)
tx2gene <- read.table("Transcrito_a_Gen.tsv", header = TRUE)
txi <- tximport(files, type = "salmon", tx2gene = tx2gene, countsFromAbundance = "no")
counts_matrix <- round(txi$counts) # Enteros
write.csv(counts_matrix, "resumen_cuantificacion.csv")

# 3. PREPARAR METADATOS
colData <- data.frame(
  sample = samples,
  grupo = c("obeso1", "obeso1", "obeso2", "obeso2", "obeso2")
)
rownames(colData) <- samples

# 4. ANÁLISIS DIFERENCIAL CON DESeq2 (Parte B)
dds <- DESeqDataSetFromMatrix(
  countData = counts_matrix,
  colData = colData,
  design = ~ grupo
)

# Añadir esta verificación
print(paste("Número de genes:", nrow(dds)))
```

```
## [1] "Número de genes: 37"
```

```
print(paste("Número de muestras:", ncol(dds)))
```

```
## [1] "Número de muestras: 5"
```

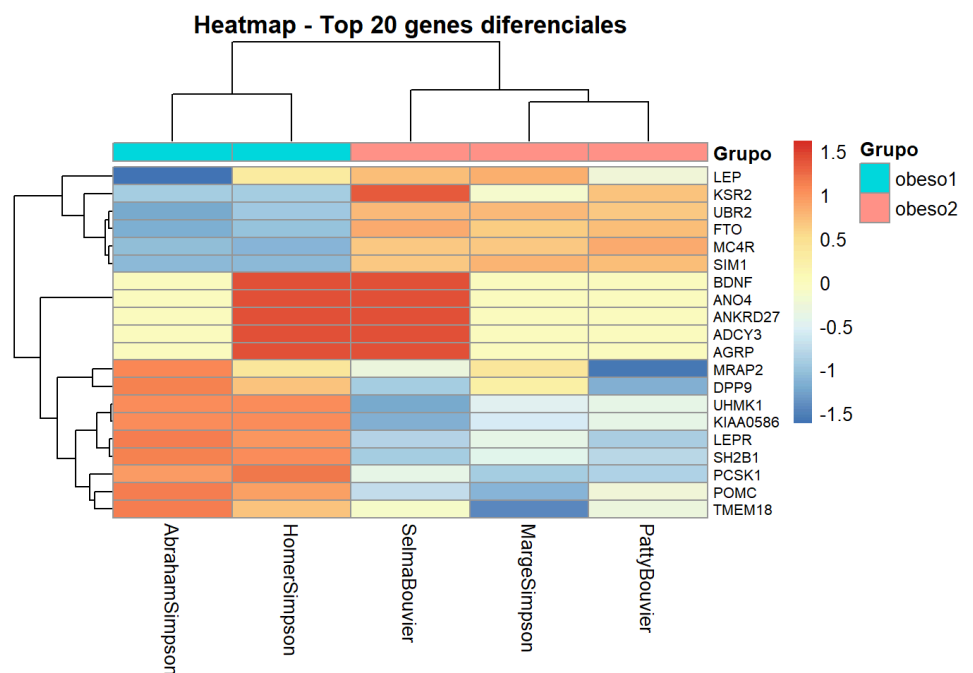
I. Introducción

```
dds <- DESeq(dds, minReplicatesForReplace = Inf) # Si  
n filtrado por simulación  
res <- results(dds, contrast = c("grupo", "obeso1", "o  
beso2"))  
  
# Heatmap - CORRECCIÓN: Usar varianceStabilizingTransf  
ormation en lugar de vst  
# Para pocos genes (37), usar rlog o vst con blind=TRU  
E  
if (nrow(dds) < 1000) {  
  vsd <- rlog(dds, blind = FALSE)  
} else {  
  vsd <- varianceStabilizingTransformation(dds, blind  
= FALSE)  
}  
  
# Seleccionar top genes (máximo 20 o todos si hay meno  
s)  
n_top <- min(20, nrow(res))  
top_genes <- rownames(res)[order(res$padj, na.last = N  
A)][1:n_top]  
  
# Filtrar genes que no tengan NA en padj  
valid_genes <- !is.na(res$padj)  
if (sum(valid_genes) > 0) {  
  top_genes <- rownames(res)[order(res$padj, na.last =  
NA)][1:min(20, sum(valid_genes))]  
  heatmap_data <- assay(vsd)[top_genes, ]  
  
  # Crear anotaciones para el heatmap  
  annotation_col <- data.frame(Grupo = colData$grupo)  
  rownames(annotation_col) <- colnames(heatmap_data)  
  
  pheatmap(heatmap_data,  
            scale = "row",  
            main = paste("Heatmap - Top", length(top_ge  
nes), "genes diferenciales"),  
            cluster_rows = TRUE,  
            cluster_cols = TRUE,  
            annotation_col = annotation_col,  
            show_rownames = TRUE,  
            fontsize_row = 8)  
} else {  
  print("No hay genes con valores de padj no-NA para e  
l heatmap")
```

I. Introducción

```
# Heatmap alternativo con todos los genes
heatmap_data <- assay(vsd)
annotation_col <- data.frame(Grupo = colData$grupo)
rownames(annotation_col) <- colnames(heatmap_data)

pheatmap(heatmap_data,
          scale = "row",
          main = "Heatmap - Todos los genes (37)",
          cluster_rows = TRUE,
          cluster_cols = TRUE,
          annotation_col = annotation_col,
          show_rownames = TRUE,
          fontsize_row = 6)
}
```



```
# 6. RESUMEN DE RESULTADOS
```

```
print("Resumen de resultados diferenciales:")
```

```
## [1] "Resumen de resultados diferenciales:"
```

```
print(summary(res))
```

I. Introducción

```
##  
## out of 37 with nonzero total read count  
## adjusted p-value < 0.1  
## LFC > 0 (up)      : 4, 11%  
## LFC < 0 (down)    : 4, 11%  
## outliers [1]      : 0, 0%  
## low counts [2]    : 0, 0%  
## (mean count < 11)  
## [1] see 'cooksCutoff' argument of ?results  
## [2] see 'independentFiltering' argument of ?results  
##  
## NULL
```

```
# Guardar resultados  
write.csv(as.data.frame(res), "resultados_diferenciales.csv")  
  
# Mostrar los genes más significativos  
print("Top 10 genes más significativos:")
```

```
## [1] "Top 10 genes más significativos:"
```

```
print(head(res[order(res$padj), ], 10))
```

I. Introducción

```
## log2 fold change (MLE): grupo obeso1 vs obeso2
## Wald test p-value: grupo obeso1 vs obeso2
## DataFrame with 10 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat
pvalue      padj
##      <numeric>      <numeric> <numeric> <numeric>
<numeric>  <numeric>
## MC4R          42.0      -3.336649  0.415986  -8.02107
1.04831e-15 3.87874e-14
## LEPR          31.0       3.180812  0.489115   6.50319
7.86329e-11 1.45471e-09
## SIM1          36.4      -2.000000  0.337008  -5.93457
2.94609e-09 3.17389e-08
## UBR2          78.8      -0.990252  0.167569  -5.90951
3.43123e-09 3.17389e-08
## SH2B1         53.4       0.987400  0.178397   5.53485
3.11503e-08 2.30512e-07
## POMC          56.0       0.708698  0.172632   4.10525
4.03879e-05 2.49059e-04
## FTO           32.6      -1.563430  0.395002  -3.95802
7.55721e-05 3.99453e-04
## PCSK1         55.8       0.636677  0.172813   3.68420
2.29419e-04 1.06106e-03
## KSR2          50.8      -0.432111  0.192632  -2.24319
2.48846e-02 1.02303e-01
## MRAP2         36.0       0.424498  0.269959   1.57245
1.15845e-01 4.28628e-01
```

I. Introducción

I. Introducción

I. Introducción