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

Esta parte contiene el Script 3, en el cual se encuentra información acerca de la red de coexpresión y enriquecimiento.

```
# ==================== ANÁLISIS C y D: RED DE COEXPRES
IÓN Y ENRIQUECIMIENTO ====================

# 1. CARGAR LIBRERÍAS
library(tximport)
library(DESeq2)
library(ggplot2)
library(dplyr)
library(corrplot)
library(igraph)
library(reshape2)
library(gprofiler2)

# 2. CONFIGURACIÓN Y LECTURA DE DATOS
cat("\n▶ INICIANDO ANÁLISIS...\n")
```

```
##
## ▶ INICIANDO ANÁLISIS...
```

```
samples <- c("AbrahamSimpson", "HomerSimpson", "MargeS
impson", "PattyBouvier", "SelmaBouvier")
files <- file.path("Salmon", samples, "quant.sf")
names(files) <- samples

# Verificar archivos
if(!file.exists("Transcrito_a_Gen.tsv")) {
  stop("Archivo 'Transcrito_a_Gen.tsv' no encontrad
o.")
}

tx2gene <- read.table("Transcrito_a_Gen.tsv", header =
TRUE, sep = "\t")
txi <- tximport(files, type = "salmon", tx2gene = tx2g
ene, countsFromAbundance = "no")
counts_matrix <- round(txi$counts)

cat("✓ Genes cargados:", nrow(counts_matrix), "\n")
```

```
## ✓ Genes cargados: 37
```

```
# 3. ANÁLISIS DIFERENCIAL
cat("\n▶ ANÁLISIS DIFERENCIAL...\n")
```

```
##
## ▶ ANÁLISIS DIFERENCIAL...
```

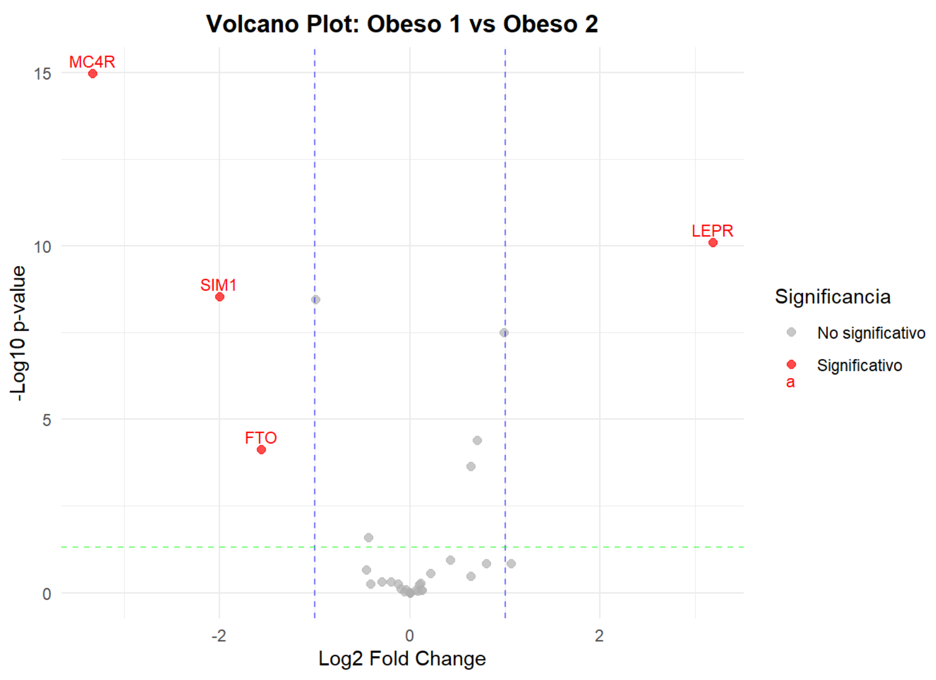
```
colData <- data.frame(
  row.names = samples,
  Grupo = factor(c("obeso1", "obeso1", "obeso2", "obes
o2", "obeso2"))
)

dds <- DESeqDataSetFromMatrix(countData = counts_matri
x, colData = colData, design = ~ Grupo)
dds <- DESeq(dds)
res <- results(dds, contrast = c("Grupo", "obeso1", "o
beso2"))

res_df <- as.data.frame(res)
res_df$gene <- rownames(res_df)
res_df$significant <- ifelse(
  !is.na(res_df$padj) & res_df$padj < 0.05 & abs(res_d
f$log2FoldChange) > 1,
  "Significativo",
  "No significativo"
)

sig_genes <- res_df$gene[res_df$significant == "Signif
icativo"]
cat("✓ Genes significativos (padj<0.05 & |FC|>1):", le
ngth(sig_genes), "\n")
```

```
## ✓ Genes significativos (padj<0.05 & |FC|>1): 4
```

```
# Guardar resultados (CORREGIDO)
write.csv(res_df, "differential_expression_results.cs
v")

# ==================== GRÁFICA 1: VOLCANO PLOT =======
=============
cat("\n▶ GRÁFICA 1: Volcano Plot\n")
```

```
##
## ▶ GRÁFICA 1: Volcano Plot
```

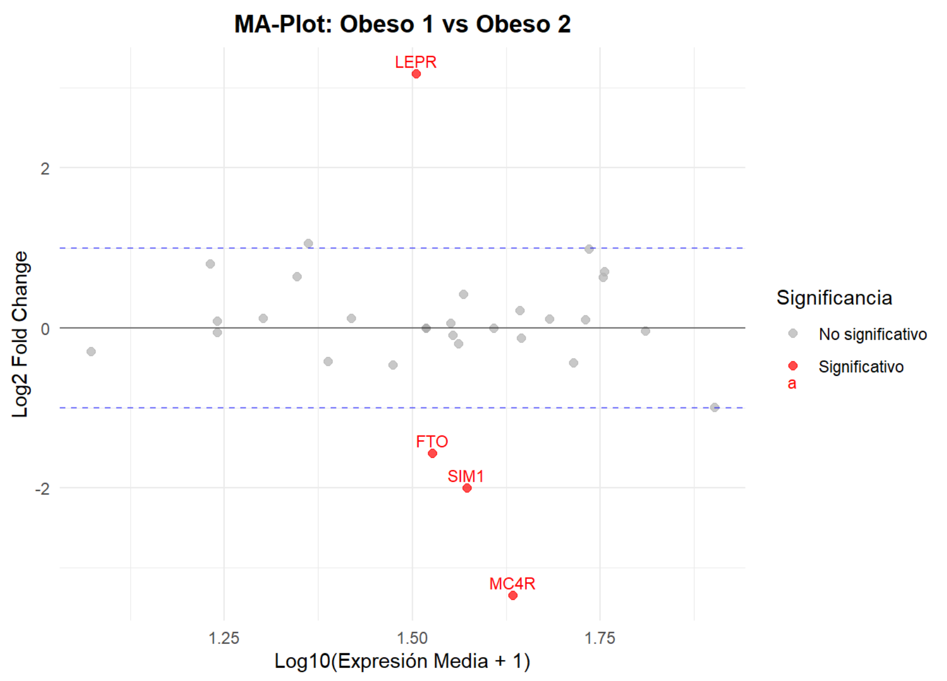
```
# Crear Volcano Plot
volcano_plot <- ggplot(res_df, aes(x = log2FoldChange,
y = -log10(pvalue), color = significant)) +
  geom_point(alpha = 0.7, size = 2) +
  scale_color_manual(values = c("No significativo" =
"gray70", "Significativo" = "red")) +
  geom_vline(xintercept = c(-1, 1), linetype = "dashe
d", color = "blue", alpha = 0.5) +
  geom_hline(yintercept = -log10(0.05), linetype = "da
shed", color = "green", alpha = 0.5) +
  labs(title = "Volcano Plot: Obeso 1 vs Obeso 2",
       x = "Log2 Fold Change",
       y = "-Log10 p-value",
       color = "Significancia") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face =
"bold"))

# Añadir etiquetas
if(length(sig_genes) > 0) {
  sig_data <- res_df[res_df$significant == "Significat
ivo", ]
  volcano_plot <- volcano_plot +
    geom_text(data = sig_data, aes(label = gene),
              vjust = -0.5, size = 3, check_overlap =
TRUE)
}

# MOSTRAR Volcano Plot
print(volcano_plot)
```



```
ggsave("01_volcano_plot.png", volcano_plot, width = 1
0, height = 7)

# =================== GRÁFICA 2: MA-PLOT ============
========
cat("\n▶ GRÁFICA 2: MA-Plot\n")
```

```
##
## ▶ GRÁFICA 2: MA-Plot
```

```
# Crear MA-Plot
normalized_counts <- counts(dds, normalized = TRUE)
ma_data <- res_df
ma_data$mean_expression <- rowMeans(normalized_counts)

ma_plot <- ggplot(ma_data, aes(x = log10(mean_expressi
on + 1), y = log2FoldChange, color = significant)) +
  geom_point(alpha = 0.7, size = 2) +
  geom_hline(yintercept = 0, color = "black", alpha =
0.5) +
  geom_hline(yintercept = c(-1, 1), linetype = "dashe
d", color = "blue", alpha = 0.5) +
  scale_color_manual(values = c("No significativo" =
"gray70", "Significativo" = "red")) +
  labs(title = "MA-Plot: Obeso 1 vs Obeso 2",
       x = "Log10(Expresión Media + 1)",
       y = "Log2 Fold Change",
       color = "Significancia") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face =
"bold"))

# Añadir etiquetas
if(length(sig_genes) > 0) {
  sig_ma_data <- ma_data[ma_data$significant == "Signi
ficativo", ]
  ma_plot <- ma_plot +
    geom_text(data = sig_ma_data, aes(label = gene),
              vjust = -0.5, size = 3, check_overlap =
TRUE)
}

# MOSTRAR MA-Plot
print(ma_plot)
```



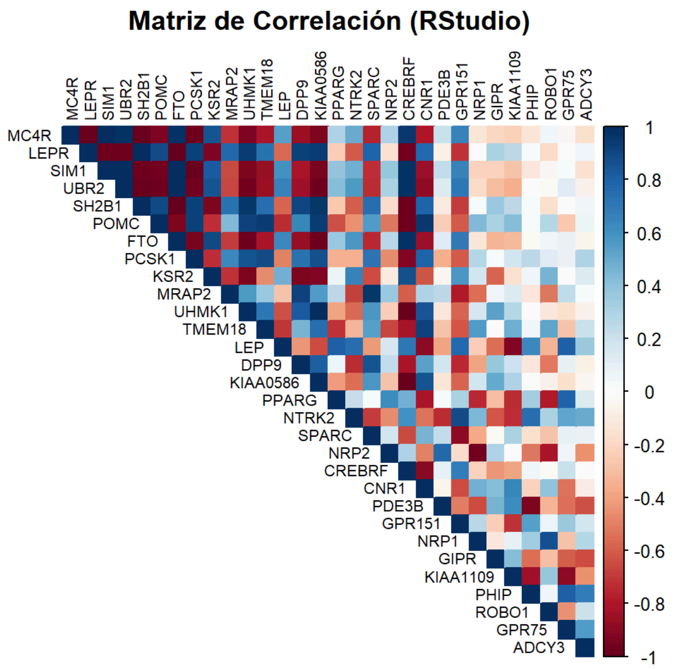
```
ggsave("02_ma_plot.png", ma_plot, width = 10, height =
7)

# ==================== ANÁLISIS C: RED DE COEXPRESIÓN
====================
cat("\n▶ ANÁLISIS C: RED DE COEXPRESIÓN\n")
```

```
##
## ▶ ANÁLISIS C: RED DE COEXPRESIÓN
```

```
# Transformar datos
expr_norm <- assay(rlog(dds, blind = FALSE))

# Calcular matriz de correlación
cor_matrix <- cor(t(expr_norm), method = "pearson")

# Guardar matriz como data.frame
cor_df <- as.data.frame(cor_matrix)
cor_df$Gene <- rownames(cor_df)
write.csv(cor_df, "C_correlation_matrix.csv", row.name
s = FALSE)

# Seleccionar genes para visualización
top_genes <- res_df %>%
  arrange(pvalue) %>%
  head(min(30, nrow(res_df))) %>%
  pull(gene)

# ==================== GRÁFICA 3: HEATMAP DE CORRELACI
ÓN ====================
cat("\n▶ GRÁFICA 3: Heatmap de Correlación\n")
```

```
##
## ▶ GRÁFICA 3: Heatmap de Correlación
```

```
if(length(top_genes) >= 3) {
  cor_subset <- cor_matrix[top_genes, top_genes]

  # MOSTRAR en RStudio
  corrplot(cor_subset,
           method = "color",
           type = "upper",
           tl.col = "black",
           tl.cex = 0.7,
           title = "Matriz de Correlación (RStudio)",
           mar = c(0,0,2,0))

  # Guardar PNG
  png("C1_correlation_heatmap.png", width = 800, heigh
t = 700)
  corrplot(cor_subset,
           method = "color",
           type = "upper",
           tl.col = "black",
           tl.cex = 0.7,
           title = "Matriz de Correlación entre Gene
s",
           mar = c(0,0,2,0))
  dev.off()
}
```



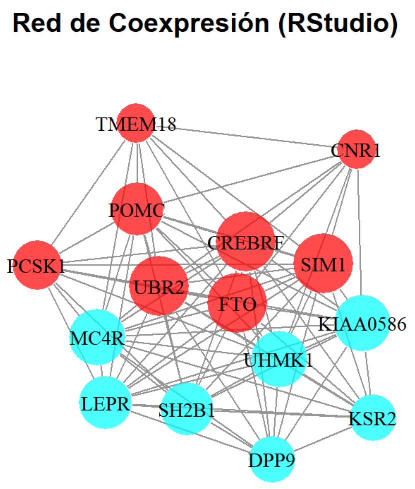
```
## png
##   2
```

```
# ==================== GRÁFICA 4: RED DE COEXPRESIÓN =
===================
cat("\n▶ GRÁFICA 4: Red de Coexpresión\n")
```

```
##
## ▶ GRÁFICA 4: Red de Coexpresión
```

```
# Construir red
cor_threshold <- 0.80
adj_matrix <- ifelse(abs(cor_matrix) > cor_threshold,
1, 0)
diag(adj_matrix) <- 0

g <- graph_from_adjacency_matrix(adj_matrix, mode = "u
ndirected", diag = FALSE)

if(vcount(g) > 0) {
  # Calcular medidas de red
  degree_vals <- degree(g)

  # Crear data.frame de genes importantes
  important_genes_df <- data.frame(
    Gene = names(degree_vals),
    Degree = degree_vals
  ) %>%
    arrange(desc(Degree)) %>%
    filter(Degree > 0)

  # Guardar como CSV
  write.csv(important_genes_df, "C_network_centrality.
csv", row.names = FALSE)

  cat("✓ Red con", vcount(g), "nodos y", ecount(g), "c
onexiones\n")

  # Visualizar red si hay suficientes nodos
  if(nrow(important_genes_df) >= 3) {
    top_hubs <- important_genes_df$Gene[1:min(15, nrow
(important_genes_df))]

    if(length(top_hubs) >= 3) {
      sub_adj <- adj_matrix[top_hubs, top_hubs]
      sub_g <- graph_from_adjacency_matrix(sub_adj, mo
de = "undirected", diag = FALSE)

      if(vcount(sub_g) >= 3) {
        # Detectar comunidades
        communities <- cluster_louvain(sub_g)

        # MOSTRAR en RStudio
        plot(sub_g,
             vertex.size = degree(sub_g) * 2 + 8,
             vertex.color = rainbow(max(communities$me
mbership), alpha = 0.7)[communities$membership],
             vertex.label.cex = 0.9,
             vertex.label.color = "black",
             vertex.frame.color = NA,
             edge.color = "gray60",
             edge.width = 1,
             main = "Red de Coexpresión (RStudio)",
             layout = layout_with_fr)

        # Guardar PNG
        png("C2_gene_coexpression_network.png", width
= 900, height = 800)
        plot(sub_g,
             vertex.size = degree(sub_g) * 2 + 8,
             vertex.color = rainbow(max(communities$me
mbership), alpha = 0.7)[communities$membership],
             vertex.label.cex = 0.9,
             vertex.label.color = "black",
             vertex.frame.color = NA,
             edge.color = "gray60",
             edge.width = 1,
             main = "Red de Coexpresión Génica",
             layout = layout_with_fr)
        dev.off()
      }
    }
  }
}
```

```
## ✓ Red con 37 nodos y 134 conexiones
```



```
## png
##   2
```

```
# ==================== ANÁLISIS D: ENRIQUECIMIENTO FUN
CIONAL ====================
cat("\n▶ ANÁLISIS D: ENRIQUECIMIENTO FUNCIONAL\n")
```

```
##
## ▶ ANÁLISIS D: ENRIQUECIMIENTO FUNCIONAL
```

```
if(length(sig_genes) >= 3) {
  # Realizar análisis de enriquecimiento
  tryCatch({
    gost_results <- gost(query = sig_genes,
                         organism = "hsapiens",
                         sources = c("GO:BP", "GO:MF",
"KEGG"),
                         evcodes = FALSE,
                         user_threshold = 0.05,
                         correction_method = "g_SCS")

    if(!is.null(gost_results) && nrow(gost_results$res
ult) > 0) {
      # Guardar resultados como data.frame
      enrichment_df <- as.data.frame(gost_results$resu
lt)
      write.csv(enrichment_df, "D_enrichment_results.c
sv", row.names = FALSE)

      # ==================== GRÁFICA 5: GO BIOLOGICAL
PROCESS ====================
      cat("\n▶ GRÁFICA 5: GO Biological Process\n")

      go_bp <- enrichment_df %>%
        filter(source == "GO:BP") %>%
        arrange(p_value) %>%
        head(10)

      if(nrow(go_bp) > 0) {
        # MOSTRAR en RStudio
        par(mar = c(5, 15, 4, 2))
        barplot(height = -log10(go_bp$p_value),
                names.arg = substr(go_bp$term_name, 1,
50),
                horiz = TRUE,
                las = 1,
                col = "steelblue",
                main = "GO Biological Process (RStudi
o)",
                xlab = "-log10(p-value)")

        # Guardar PNG
        png("D1_GO_BP_enrichment.png", width = 1000, h
eight = 700)
        par(mar = c(5, 15, 4, 2))
        barplot(height = -log10(go_bp$p_value),
                names.arg = substr(go_bp$term_name, 1,
50),
                horiz = TRUE,
                las = 1,
                col = "steelblue",
                main = "GO Biological Process",
                xlab = "-log10(p-value)")
        dev.off()
      }

      # ==================== GRÁFICA 6: KEGG PATHWAYS 
====================
      cat("\n▶ GRÁFICA 6: KEGG Pathways\n")

      kegg <- enrichment_df %>%
        filter(source == "KEGG") %>%
        arrange(p_value) %>%
        head(10)

      if(nrow(kegg) > 0) {
        # MOSTRAR en RStudio
        par(mar = c(5, 15, 4, 2))
        barplot(height = -log10(kegg$p_value),
                names.arg = substr(kegg$term_name, 1, 
50),
                horiz = TRUE,
                las = 1,
                col = "darkred",
                main = "KEGG Pathways (RStudio)",
                xlab = "-log10(p-value)")

        # Guardar PNG
        png("D2_KEGG_enrichment.png", width = 1000, he
ight = 700)
        par(mar = c(5, 15, 4, 2))
        barplot(height = -log10(kegg$p_value),
                names.arg = substr(kegg$term_name, 1, 
50),
                horiz = TRUE,
                las = 1,
                col = "darkred",
                main = "KEGG Pathways",
                xlab = "-log10(p-value)")
        dev.off()
      }

      cat("✓ Enriquecimiento completado\n")
    }
```

```
  }, error = function(e) {
    cat("⚠ Error en enriquecimiento:", e$message,
"\n")
  })
}
```

```
## ⚠ Error en enriquecimiento: tipo no implementado 'l
ist' en 'EncodeElement'
##
```

```
# ==================== FIGURA INTEGRADA ==============
======
cat("\n▶ CREANDO FIGURA INTEGRADA\n")
```

```
##
## ▶ CREANDO FIGURA INTEGRADA
```

```
png("Figure_integrated_analysis.png", width = 1600, he
ight = 1200)
par(mfrow = c(2, 2), mar = c(5, 4, 4, 2), cex.main =
1.2)

# Panel 1: Volcano Plot
plot(res_df$log2FoldChange, -log10(res_df$pvalue),
     pch = 19, cex = 0.6,
     col = ifelse(res_df$significant == "Significativ
o", "red", "gray70"),
     xlab = "Log2 Fold Change", ylab = "-log10(p-valu
e)",
     main = "A) Volcano Plot")
abline(v = c(-1, 1), lty = 2, col = "blue")
abline(h = -log10(0.05), lty = 2, col = "green")

# Panel 2: MA-Plot
plot(log10(ma_data$mean_expression + 1), ma_data$log2F
oldChange,
     pch = 19, cex = 0.6,
     col = ifelse(ma_data$significant == "Significativ
o", "red", "gray70"),
     xlab = "Log10(Expresión Media + 1)", ylab = "Log2
Fold Change",
     main = "B) MA-Plot")
abline(h = 0, col = "black")
abline(h = c(-1, 1), lty = 2, col = "blue")

# Panel 3: Términos enriquecidos
if(exists("go_bp") && nrow(go_bp) > 0) {
  barplot(height = -log10(head(go_bp$p_value, 5)),
          names.arg = substr(head(go_bp$term_name, 5),
1, 40),
          horiz = TRUE, las = 1, col = "steelblue",
          main = "C) Top 5 GO Terms", xlab = "-log10(p
-value)")
} else {
  plot(1, 1, type = "n", main = "C) Sin términos enriq
uecidos",
       xlab = "", ylab = "", axes = FALSE)
  text(1, 1, "No hay términos\nsignificativos", cex =
1.2)
}

# Panel 4: Heatmap simple
if(length(top_genes) >= 3) {
  exp_data <- expr_norm[top_genes, ]
  exp_scaled <- t(scale(t(exp_data)))

  image(1:ncol(exp_scaled), 1:nrow(exp_scaled),
        t(exp_scaled),
        col = colorRampPalette(c("blue", "white", "re
d"))(100),
        xlab = "Muestras", ylab = "Genes",
        main = "D) Heatmap de Expresión",
        axes = FALSE)

  axis(1, at = 1:ncol(exp_scaled), labels = colnames(e
xp_scaled), las = 2, cex.axis = 0.7)
  axis(2, at = 1:nrow(exp_scaled), labels = rownames(e
xp_scaled), las = 1, cex.axis = 0.6)
} else {
  plot(1, 1, type = "n", main = "D) Heatmap",
       xlab = "", ylab = "", axes = FALSE)
  text(1, 1, "Datos insuficientes", cex = 1.2)
}

dev.off()
```

```
## png
##   2
```

```
# =================== RESUMEN FINAL =================
===
cat("\n")
```

```
cat("======================================================
=============================\n")
```

```
## ======================================================
===========================
```

```
cat("✅ ANÁLISIS COMPLETADO EXITOSAMENTE\n")
```

```
## ✅ ANÁLISIS COMPLETADO EXITOSAMENTE
```

```
cat("======================================================
=============================\n")
```

```
## ======================================================
===========================
```

```
cat("\n")
```

```
cat("📊 GRÁFICAS GENERADAS:\n")
```

```
## 📊 GRÁFICAS GENERADAS:
```

```
cat("1. 01_volcano_plot.png           - Volcano Plot
\n")
```

```
## 1. 01_volcano_plot.png           - Volcano Plot
```

```
cat("2. 02_ma_plot.png                - MA-Plot\n")
```

```
## 2. 02_ma_plot.png                - MA-Plot
```

```
cat("3. C1_correlation_heatmap.png   - Heatmap de corr
elación\n")
```

```
## 3. C1_correlation_heatmap.png   - Heatmap de correl
ación
```

```
cat("4. C2_gene_coexpression_network.png - Red de coex
presión\n")
```

```
## 4. C2_gene_coexpression_network.png - Red de coexpr
esión
```

```
cat("5. D1_GO_BP_enrichment.png      - GO Biological P
rocess\n")
```

```
## 5. D1_GO_BP_enrichment.png      - GO Biological Pro
cess
```

```
cat("6. D2_KEGG_enrichment.png       - KEGG Pathways
\n")
```

```
## 6. D2_KEGG_enrichment.png       - KEGG Pathways
```

```
cat("7. Figure_integrated_analysis.png - Figura integr
ada\n")
```

```
## 7. Figure_integrated_analysis.png - Figura integrad
a
```

```
cat("\n")
```

```
cat("📁 DATOS GUARDADOS:\n")
```

```
## 📁 DATOS GUARDADOS:
```

```
cat("1. differential_expression_results.csv - Resultad
os DE\n")
```

```
## 1. differential_expression_results.csv - Resultados
DE
```

```
cat("2. C_correlation_matrix.csv           - Matriz de
correlación\n")
```

```
## 2. C_correlation_matrix.csv           - Matriz de c
orrelación
```

```
cat("3. C_network_centrality.csv           - Genes cen
trales\n")
```

```
## 3. C_network_centrality.csv           - Genes centr
ales
```

```
cat("4. D_enrichment_results.csv           - Resultado
s enriquecimiento\n")
```

```
## 4. D_enrichment_results.csv           - Resultados
enriquecimiento
```

```
cat("\n")
```

```
cat("📈 RESULTADOS:\n")
```

```
## 📈 RESULTADOS:
```

```
cat(sprintf("• Genes analizados: %d\n", nrow(res_df)))
```

```
## • Genes analizados: 37
```

```
cat(sprintf("• Genes significativos: %d\n", length(sig
_genes)))
```

```
## • Genes significativos: 4
```

```
if(exists("g")) {
  cat(sprintf("• Red: %d nodos, %d conexiones\n", vcou
nt(g), ecount(g)))
}
```

```
## • Red: 37 nodos, 134 conexiones
```

```
if(exists("enrichment_df")) {
  cat(sprintf("• Términos enriquecidos: %d\n", nrow(en
richment_df)))
}
```

```
## • Términos enriquecidos: 4
```

```
cat("\n")
```

```
cat("*************************************************
*****************************\n")
```

```
## *************************************************
***************************
```

```
cat("✅ ANÁLISIS LISTO PARA PUBLICACIÓN\n")
```

```
## ✅ ANÁLISIS LISTO PARA PUBLICACIÓN
```

```
cat("*******************************************************************************\n")
```

```
## *******************************************************************************
```

```
cat("\n")
```