Análisis de Expresión Diferencial

García González Carlos

2025-02-05

Contents

Análisis de Expresión Diferencial: Leucemia Mieloide Aguda

Este documento presenta un análisis de expresión diferencial utilizando datos de pacientes con **Leucemia Mieloide Aguda (LAML)** del proyecto TCGA. Se emplean herramientas de **bioinformática** para la selección, limpieza y visualización de datos de expresión génica.

Configuración del Entorno

Selección del set de datos

Se selecciona el conjunto de datos correspondiente a **LAML (Leucemia Mieloide Aguda)** del repositorio recount3, específicamente del consorcio **TCGA (The Cancer Genome Atlas)**.

```
# Obtener la lista de proyectos disponibles de humanos en recount3
human_projects <- available_projects(organism = "human")
```

```
## 2025-02-08 21:53:03.998909 caching file sra.recount_project.MD.gz.
```

```
## 2025-02-08 21:53:04.723496 caching file gtex.recount_project.MD.gz.

## 2025-02-08 21:53:05.365987 caching file tcga.recount_project.MD.gz.
```

```r
# Filtrar solo los proyectos pertenecientes a TCGA (The Cancer Genome Atlas)
tcga_projects <- human_projects[human_projects$project_home == "data_sources/tcga", ]

# Seleccionar el proyecto LAML (Leucemia Mieloide Aguda)
project_info <- subset(tcga_projects, project == "LAML")

# Crear un objeto RangedSummarizedExperiment con los datos de LAML
rse_LAML <- create_rse(project_info)
```

```
## 2025-02-08 21:53:14.845526 downloading and reading the metadata.

## 2025-02-08 21:53:15.86934 caching file tcga.tcga.LAML.MD.gz.

## 2025-02-08 21:53:16.465047 caching file tcga.recount_project.LAML.MD.gz.

## 2025-02-08 21:53:17.06418 caching file tcga.recount_qc.LAML.MD.gz.

## 2025-02-08 21:53:17.719402 caching file tcga.recount_seq_qc.LAML.MD.gz.

## 2025-02-08 21:53:18.663791 downloading and reading the feature information.

## 2025-02-08 21:53:19.251297 caching file human.gene_sums.G026.gtf.gz.

## 2025-02-08 21:53:20.47889 downloading and reading the counts: 178 samples across 63856 features.

## 2025-02-08 21:53:21.101362 caching file tcga.gene_sums.LAML.G026.gz.

## Warning in grep(pattern, bfr, value = TRUE): unable to translate ' El n<a3>mero
## de serie del volumen es: 20C0-F08A' to a wide string

## Warning in grep(pattern, bfr, value = TRUE): input string 2 is invalid

## Warning in grep(pattern, bfr, value = TRUE): unable to translate ' El n<a3>mero
## de serie del volumen es: 20C0-F08A' to a wide string

## Warning in grep(pattern, bfr, value = TRUE): input string 2 is invalid

## Warning in grep(pattern, bfr, value = TRUE): unable to translate ' El n<a3>mero
## de serie del volumen es: 20C0-F08A' to a wide string

## Warning in grep(pattern, bfr, value = TRUE): input string 2 is invalid

## 2025-02-08 21:53:27.576168 constructing the RangedSummarizedExperiment (rse) object.
```

```r
# Calcular los conteos de lectura y almacenarlos en el assay "counts"
assay(rse_LAML, "counts") <- compute_read_counts(rse_LAML)

# Mostrar el objeto con información de los datos
print(rse_LAML)
```

```
## class: RangedSummarizedExperiment
## dim: 63856 178
## metadata(8): time_created recount3_version ... annotation recount3_url
## assays(2): raw_counts counts
## rownames(63856): ENSG00000278704.1 ENSG00000277400.1 ...
##   ENSG00000182484.15_PAR_Y ENSG00000227159.8_PAR_Y
## rowData names(10): source type ... havana_gene tag
## colnames(178): 984f27ef-d4d7-4e68-bd64-776fdfc04d07
##   8ff9e94a-2ed2-4727-947f-d524d7ece815 ...
##   4c810ffa-ed07-4f4c-9f81-b8f1cf4956f7
##   cebe9594-0f19-46b4-af7d-f8df33e00afb
## colData names(937): rail_id external_id ... recount_seq_qc.errq
##   BigWigURL
```

Formateo de la información del proyecto

Para un análisis más preciso, se filtran **genes de interés** y se seleccionan las **variables clínicas relevantes** del dataset.

```r
# Definir genes de interés para el análisis
genes_interes <- c("SPN", "RUNX1", "CEBPA", "GATA2", "SPI1", "MYB", "FLI1",
                   "ERG", "MECOM", "TAL1", "LMO2", "LDB1", "CBFB", "GFI1",
                   "HOXA9", "MEIS1", "KMT2A", "WT1", "EZH2", "DNMT3A", "TET2",
                   "ASXL1", "IDH1", "IDH2", "NPM1", "FLT3", "KIT", "CSF3R",
                   "MPL", "JAK2", "STAT5A", "STAT3", "ETV6", "NRAS", "KRAS",
                   "PTPN11", "NF1", "CBL", "GATA1", "GATA3", "ZBTB16", "EVI5",
                   "FOXO3", "BCL2", "BCL6", "BAX", "MCL1", "CDKN1A", "CDKN2A",
                   "TP53", "RB1", "MDM2", "IKZF1", "DNTT", "RAG1", "RAG2",
                   "E2A", "HHEX", "ZNF521", "PRDM16", "ARID5B", "KLF4", "KLF5",
                   "MAFB", "IRF8", "IRF4", "NFE2", "NFE2L2", "BACH1", "BACH2",
                   "EP300", "CREBBP", "CBX5", "SUZ12", "SMARCA4", "SMARCB1",
                   "CTCF", "ZEB2", "SNAI1", "SNAI2", "TWIST1", "FOXP1", "FOXP3",
                   "NOTCH1", "NOTCH2", "DLL1", "JAG1", "HES1", "HEY1", "SOCS1",
                   "SOCS3", "PPARG", "NCOR1", "NCOR2", "RXRA", "VDR", "MBD2",
                   "TGFBR1", "SMAD3", "SMAD4", "CD3D", "MYC", "RELA", "NFKB1",
                   "NFKB2", "BCOR", "CD3E", "CD3G", "CEBPB", "CEBPD", "CEBPG",
                   "STAT2", "STAT4", "STAT6", "SOCS2", "SOCS4", "SOCS5",
                   "SOCS6", "SOCS7", "SMAD1", "SMAD2", "SMAD5", "SMAD6",
                   "SMAD7", "TGFB1", "TGFB2", "TGFB3", "TNF", "TNFRSF1A",
                   "TNFRSF1B", "TNFAIP3", "TNIP1", "BIRC2", "BIRC3", "BIRC5",
                   "XIAP", "FAS", "FASLG", "TRAF1", "TRAF2", "TRAF3", "TRAF6",
                   "NLRP3", "NLRP1", "CASP1", "CASP3", "CASP7", "CASP8",
                   "CASP9", "CASP10", "BAK1", "BID", "BAD", "BBC3", "MALT1",
                   "CARD11", "CARD9", "NOD1", "NOD2", "MYD88", "TICAM1", "TLR1",
                   "TLR2", "TLR3", "TLR4", "TLR5", "TLR6", "TLR7", "TLR8",
                   "TLR9", "TLR10", "DOK1", "DOK2", "DOK3", "DOK4", "DOK5",
```

```
                    "DOK6", "SH2B1", "SH2B2", "SH2B3", "CBLB", "CBL2", "UBASH3A",
                    "UBASH3B", "LCP2", "LAT", "FYB", "GRAP", "GRB2", "GAB1",
                    "GAB2", "GAB3", "SHC1", "SHC2", "SHC3", "SHC4", "CRKL",
                    "CRK", "NCK1", "NCK2", "VAV1", "VAV2", "VAV3", "DOCK2",
                    "DOCK8", "ITK", "BTK", "TXK", "TEC", "LCK", "FYN", "HCK",
                    "LYN", "BLK", "YES1", "SYK", "ZAP70", "CSK", "PTK2", "PTK2B",
                    "FER", "FES", "FGR", "EPHA1", "EPHA2", "EPHA3", "EPHA4",
                    "EPHA5", "EPHA6", "EPHA7", "EPHA8", "EPHB1", "EPHB2",
                    "EPHB3", "EPHB4", "EPHB6", "KITLG", "FLT1", "FLT4", "KDR",
                    "PDGFRA", "PDGFRB", "FGFR1", "FGFR2", "FGFR3", "FGFR4",
                    "EGFR", "ERBB2", "ERBB3", "ERBB4", "INSR", "IGF1R", "IGF2R",
                    "MET", "RON", "AXL", "MERTK", "TYRO3", "TEK", "TIE1", "ROR1",
                    "ROR2", "ALK", "ROS1", "NTRK1", "NTRK2", "NTRK3", "DDR1",
                    "DDR2", "EPHA10", "EPHB10","STK11", "MTOR", "PIK3CA",
                    "PIK3CB", "PIK3CD", "PIK3CG", "AKT1", "AKT2", "AKT3", "PTEN",
                    "PDPK1", "PDPK2", "RAC1", "RAC2", "RAC3", "RHOA", "RHOB",
                    "RHOC", "CDC42", "ARHGEF1", "ARHGEF2", "ARHGEF3", "ARHGEF4",
                    "ARHGEF5", "ARHGEF6", "ARHGEF7", "ARHGEF8", "ARHGEF9",
                    "ARHGEF10", "ARHGEF11", "DOCK1", "DOCK3", "DOCK4", "DOCK5",
                    "DOCK6", "DOCK7", "DOCK9", "DOCK10", "DOCK11", "DOCK12",
                    "DOCK13", "DOCK14", "DOCK15", "DOCK16", "DOCK17", "DOCK18",
                    "DOCK19", "DOCK20", "DOCK21", "DOCK22", "DOCK23", "DOCK24",
                    "DOCK25", "DOCK26", "DOCK27"
)


# Filtrar solo los genes de interés en el conjunto de datos
rse_LAML2 <- rse_LAML[which(rowData(rse_LAML)$gene_name %in% genes_interes), ]

# Filtrar columnas relevantes en la metadata del proyecto
columnas_interes <- c("tcga.gdc_cases.diagnoses.age_at_diagnosis",
                      "tcga.gdc_cases.diagnoses.vital_status",
                      "tcga.gdc_cases.samples.sample_type")
colData(rse_LAML2) <- colData(rse_LAML)[, columnas_interes]
```

Análisis de expresión diferencial

Se lleva a cabo un análisis de expresión diferencial utilizando **DESeq2**, que permite identificar genes diferencialmente expresados en función del estado vital de los pacientes.

```
# Asegurar que 'counts' sea la primera matriz en assays
assays(rse_LAML2) <- assays(rse_LAML2)[c("counts", "raw_counts")]

# Filtrar muestras sin NA en vital_status
rse_LAML2 <- rse_LAML2[, !is.na(colData(rse_LAML2)$tcga.gdc_cases.diagnoses.vital_status)]

# Convertir datos a objeto DESeq2
dds <- DESeqDataSet(rse_LAML2, design = ~ tcga.gdc_cases.diagnoses.vital_status)
```

```
## converting counts to integer mode

## Warning in DESeqDataSet(rse_LAML2, design =
```

```
## ~tcga.gdc_cases.diagnoses.vital_status): some variables in design formula are
## characters, converting to factors
```

```
dds <- DESeq(dds)  # Ajuste del modelo
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 19 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

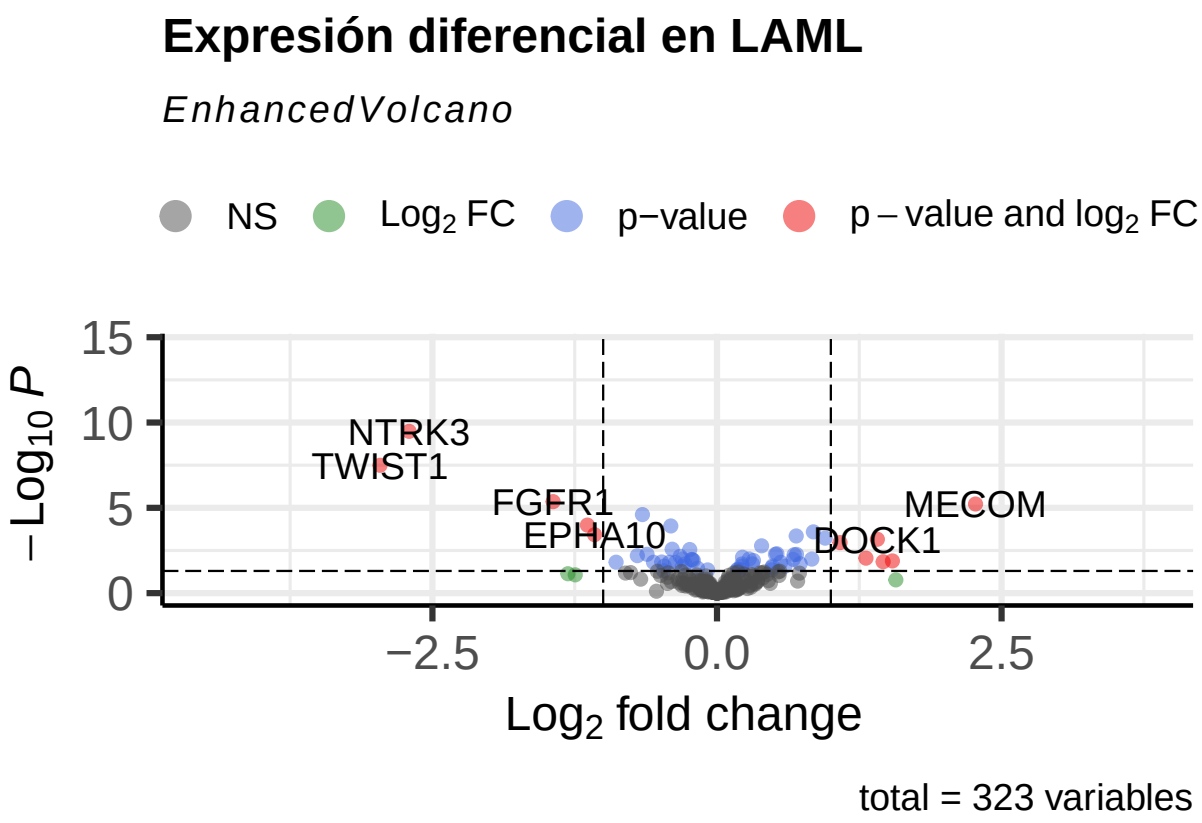
```
# Obtener resultados del análisis diferencial
res <- results(dds)
res$gene <- rowData(rse_LAML2)$gene_name[match(rownames(res), rownames(rse_LAML2))]
res <- res[order(res$padj), ]
```

Visualización de resultados

Gráfico tipo Volcán

Un **gráfico tipo volcán** permite visualizar los genes más significativamente diferencialmente expresados en función del **log2FoldChange** y el valor de **p-value**.

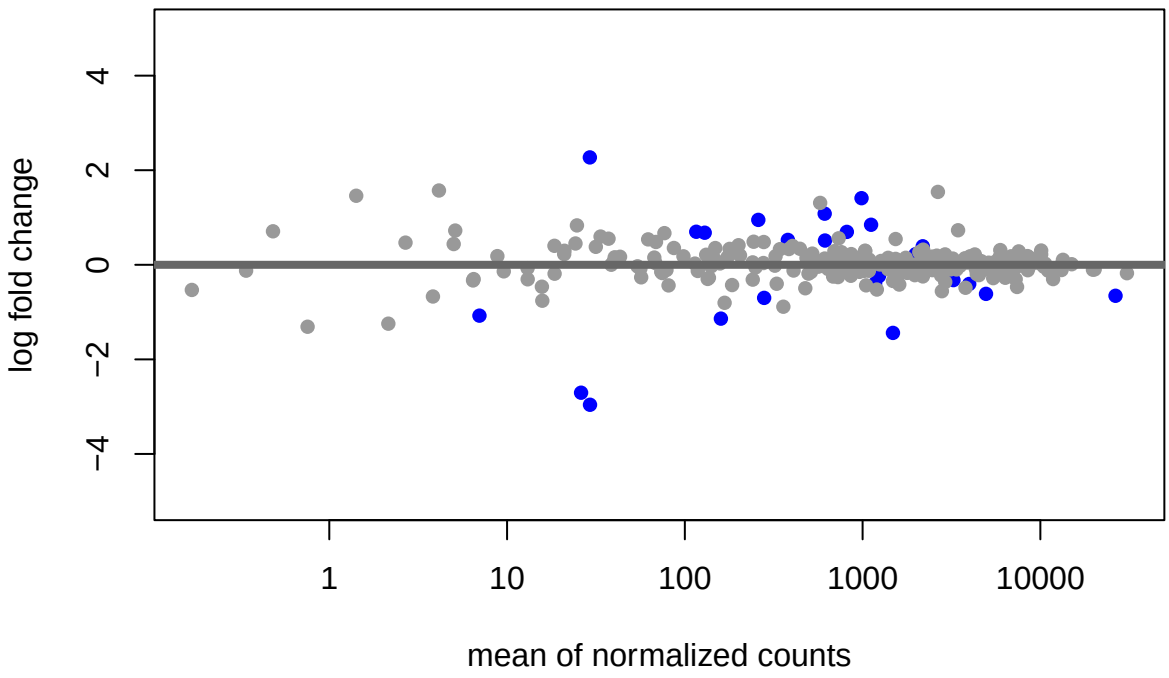
```
EnhancedVolcano(res,
                lab = res$gene,
                x = "log2FoldChange",
                y = "pvalue",
                title = "Expresión diferencial en LAML",
                pCutoff = 0.05)
```



MA plot de las muestras

Un **MA plot** permite visualizar la expresión diferencial de los genes al comparar la **media de expresión normalizada** (Eje X) con el **log2FoldChange** (Eje Y). Este gráfico ayuda a identificar genes diferencialmente expresados al resaltar aquellos con mayor variabilidad y cambios significativos en su expresión.

```
plotMA(res, ylim = c(-5, 5), cex = 1)
```

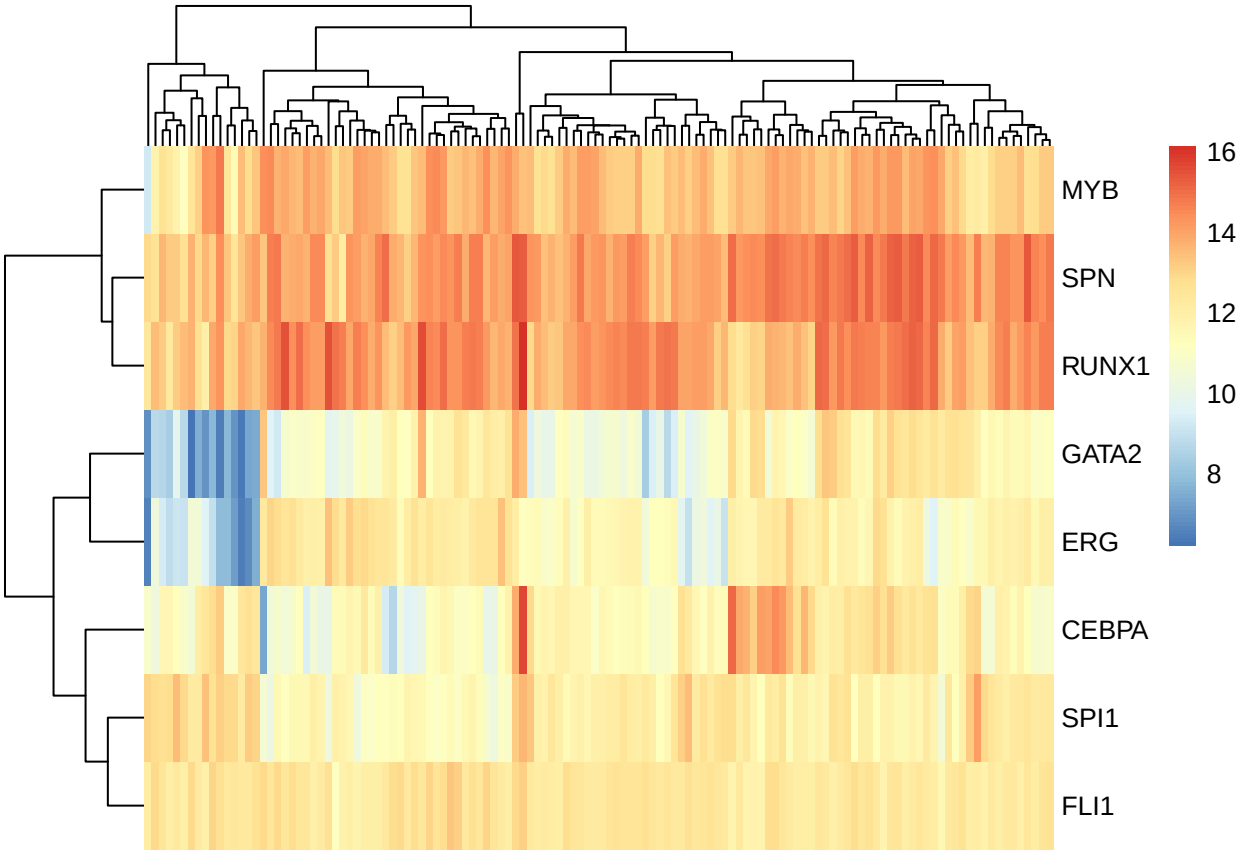


Heatmap de genes de interés

Se genera un **heatmap** para visualizar la expresión de los genes de interés en las distintas muestras.

```
# Transformación de varianza estabilizada
vsd <- varianceStabilizingTransformation(dds)

# Asegurar que los rownames de vsd sean los nombres de los genes
rownames(vsd) <- rowData(vsd)$gene_name

# Crear la matriz de expresión
vsd_matrix <- assay(vsd)

genes_interes_heatmap <- c("SPN", "RUNX1", "CEBPA", "GATA2", "SPI1", "MYB", "FLI1", "ERG")

# Filtrar solo los genes de interés
genes_presentes <- genes_interes_heatmap[genes_interes_heatmap %in% rownames(vsd_matrix)]
vsd_matrix <- vsd_matrix[genes_presentes, , drop = FALSE]

# Verificar que la matriz tenga suficientes genes antes de hacer el heatmap
if (length(genes_presentes) == 0) {
    stop("Ninguno de los genes de interés está presente en la matriz de expresión.")
} else if (length(genes_presentes) == 1) {
    pheatmap::pheatmap(vsd_matrix, show_colnames = FALSE, cluster_rows = FALSE)
} else {
```

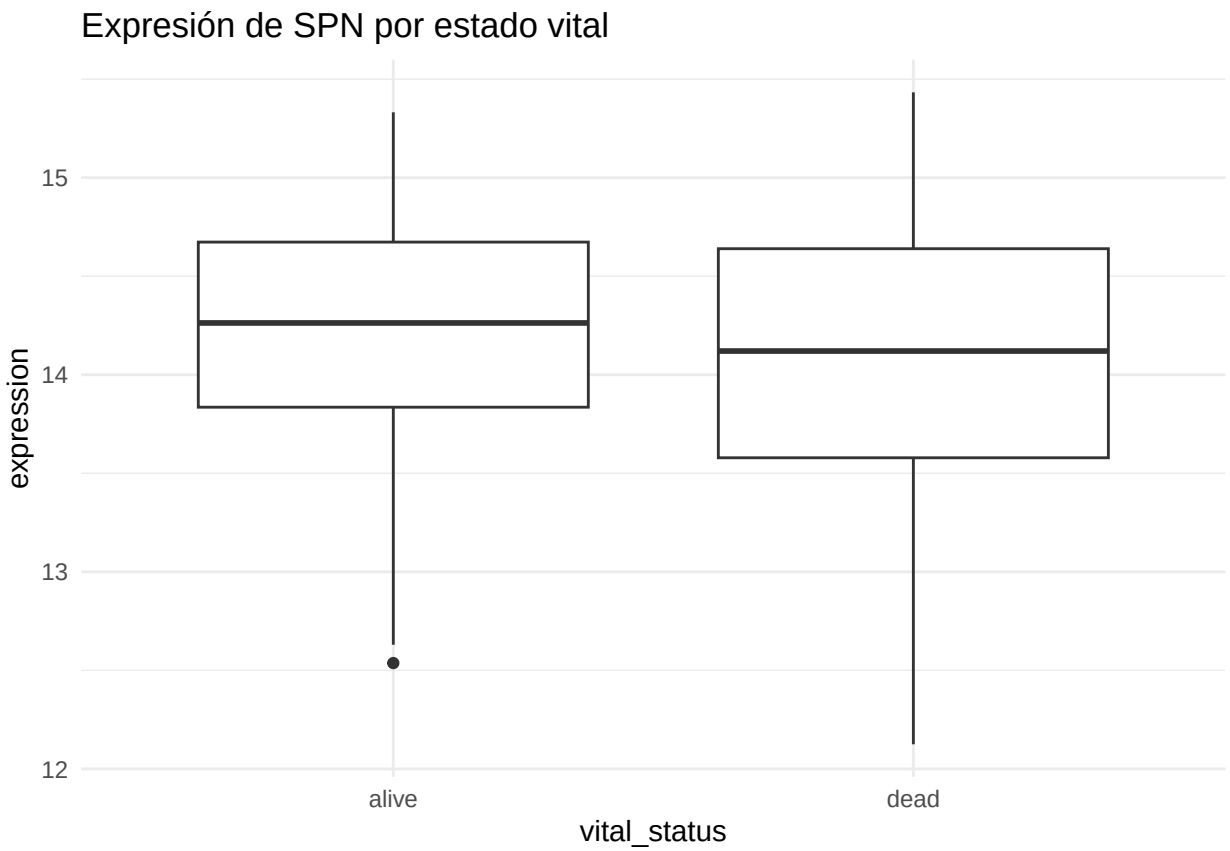
```
    pheatmap::pheatmap(vsd_matrix, show_colnames = FALSE)
}
```



Boxplot de expresión de un gen

Se visualiza la expresión del gen **SPN** por estado vital en un **boxplot**.

```
# Extraer expresión de un gen específico
SPN_expr <- as.vector(assay(vsd)["SPN", ])

df_boxplot <- data.frame(
  vital_status = colData(vsd)$tcga.gdc_cases.diagnoses.vital_status,
  expression = SPN_expr
)

# Generar boxplot
ggplot(df_boxplot, aes(x = vital_status, y = expression)) +
  geom_boxplot() +
  theme_minimal() +
  ggtitle("Expresión de SPN por estado vital")
```



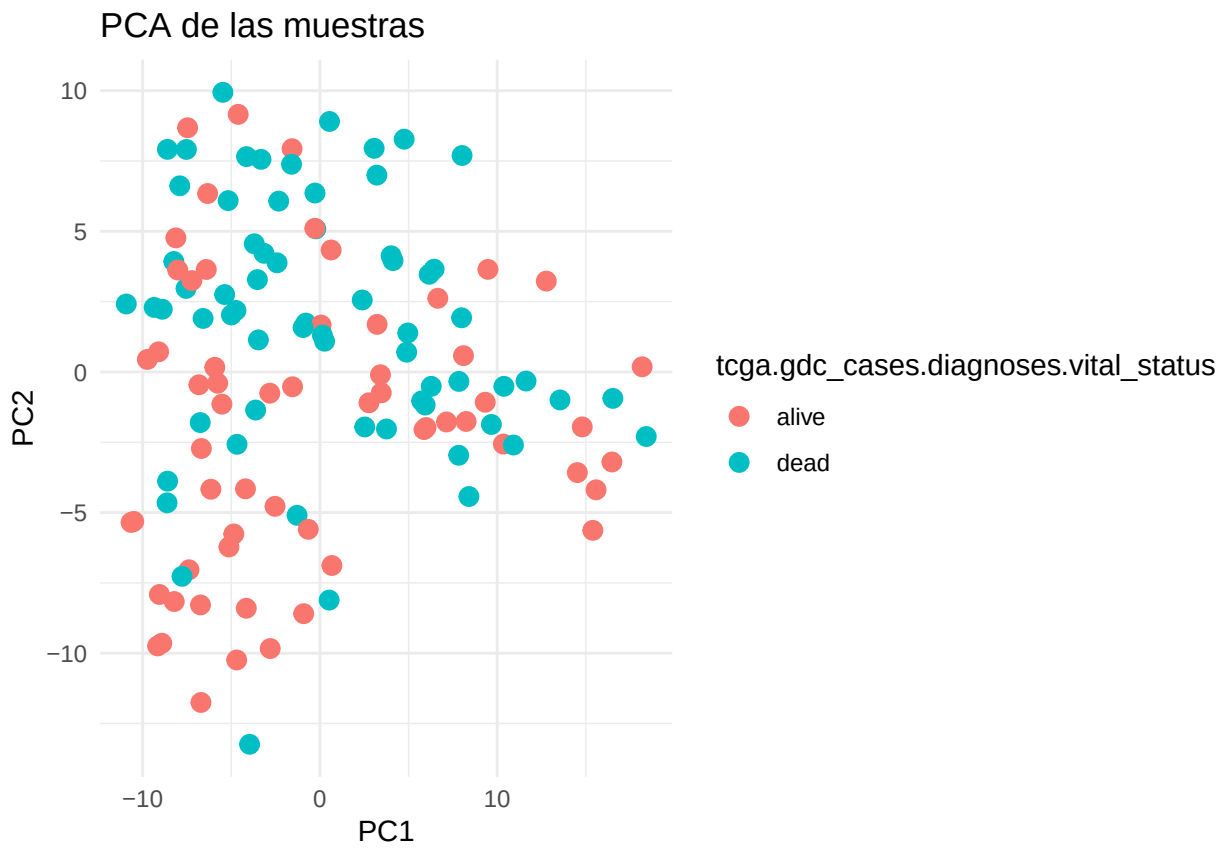
PCA de las muestras

Se realiza un **Análisis de Componentes Principales (PCA)** para explorar la variabilidad entre muestras.

```
vsd2 <- varianceStabilizingTransformation(dds, blind = TRUE)
pcaData <- plotPCA(vsd2, intgroup = "tcga.gdc_cases.diagnoses.vital_status", returnData = TRUE)
```

```
## using ntop=500 top features by variance
```

```
ggplot(pcaData, aes(PC1, PC2, color = tcga.gdc_cases.diagnoses.vital_status)) +
  geom_point(size = 3) +
  theme_minimal() +
  ggtitle("PCA de las muestras")
```



Conclusión

Este análisis proporciona una visión detallada de la expresión diferencial en **Leucemia Mieloide Aguda**, ayudando a identificar genes clave que podrían ser relevantes en la progresión de la enfermedad y posibles objetivos terapéuticos.