



Universitat Oberta  
de Catalunya

# Análisis de datos de RNA-Seq empleando diferentes paquetes desarrollados dentro del proyecto *Bioconductor* para estudios de expresión génica diferencial

**Cristina Chamorro Poyo**

Máster en Bioinformática y Bioestadística  
Estudios genéticos de enfermedades humanas

*Consultora:*

**Helena Brunel Montaner**

*Profesor responsable de la asignatura:*

**David Merino Arranz**

4 de junio de 2019



Esta obra está sujeta a una licencia de  
Reconocimiento-NoComercial-  
SinObraDerivada [3.0 España de Creative  
Commons](#)

**Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)**

**A) Creative Commons:**



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-Compartirlgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-Compartirlgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

**B) GNU Free Documentation License (GNU FDL)**

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

### **C) Copyright**

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Análisis de datos de RNA-Seq empleando diferentes paquetes desarrollados dentro del proyecto Bioconductor para estudios de expresión génica diferencial.</i>
<b>Nombre del autor:</b>	<i>Cristina Chamorro Poyo</i>
<b>Nombre del consultor/a:</b>	<i>Helena Brunel Montaner</i>
<b>Nombre del PRA:</b>	<i>David Merino Arranz</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>06/2019</i>
<b>Titulación::</b>	<i>Máster en Bioinformática y Bioestadística</i>
<b>Área del Trabajo Final:</b>	<i>Estudios genéticos de enfermedades humanas</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>RNA-Seq, análisis de expresión génica diferencial (DGE), Bioconductor</i>

**Resumen del Trabajo (máximo 250 palabras):** Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.

A pesar de que la tecnología RNA-Seq está suponiendo una auténtica revolución de los estudios en transcriptómica y, prácticamente, ha desplazado a técnicas previas como los *microarrays*, todavía no se ha decidido la metodología estándar a seguir para el análisis de los datos derivados, especialmente, de estudios de expresión génica diferencial.

La búsqueda de herramientas para el análisis de los datos en este tipo de estudios ha llevado a la adaptación de paquetes diseñados inicialmente para el análisis de datos de *microarrays* así como al desarrollo de nuevos programas basados en técnicas estadísticas especializadas. Estas herramientas tienen ventajas, limitaciones y metodología a veces poco conocida por los investigadores.

Este trabajo aborda la comparación de los tres paquetes del proyecto *Bioconductor* más populares para el análisis de datos de estudios de expresión génica diferencial: *Limma*, *EdgeR* y *DESeq2*. Para ello, se realizó un exhaustivo análisis de datos procedentes de experimentos reales de RNA-Seq.

Las ventajas e inconvenientes identificadas en cada paquete permitirá a los investigadores tomar decisiones más acertadas de acuerdo a las características del experimento.

**Abstract (in English, 250 words or less):**

Despite the fact *RNA-Seq* technologies suppose a major change within transcriptome analysis, replacing previous technologies like microarrays; a standard method for processing its outputs, especially differential gene expression analysis, has not yet been decided.

Search for data analysis tools, in this type of studies, has led the implementation of previously microarrays data analysis package as well as, new software development based on specialized statistical techniques. Advantages and disadvantages about these tools are not usually well-known by researchers.

This project focuses on the comparison of three most popular *Bioconductor* packages for differential gene expression data analysis: *Limma*, *EdgeR* and *DESeq2*. An exhaustive data analysis, from real *RNA-Seq* experiments, was done in order to achieve this goal.

Identification of advantages and disadvantages within all the three packages will allow researchers to take right decisions according experiment specifications.

# Índice

<b>1. Introducción.....</b>	<b>1</b>
1.1. Contexto y justificación del Trabajo.....	1
1.1.1. Descripción general.....	1
1.1.2. Justificación del Trabajo Final de Máster .....	1
1.2. Objetivos del Trabajo .....	2
1.2.1. Objetivo general .....	2
1.2.2. Objetivos específicos .....	2
1.3. Enfoque y método seguido .....	2
1.4. Planificación del Trabajo.....	3
1.4.1. Tareas .....	3
1.5. Calendario .....	4
1.6. Hitos.....	7
1.7. Análisis de riesgos .....	8
1.8. Breve sumario de productos obtenidos .....	8
1.9. Breve descripción de los otros capítulos de la memoria .....	9
<b>2. Contexto biológico: NGS y RNA-Seq .....</b>	<b>10</b>
<b>3. Materiales y métodos.....</b>	<b>12</b>
3.1. Softwares y paquetes de R usados para el análisis .....	12
3.1.1. Entorno R y <i>R-Studio</i> .....	12
3.1.2. <i>Bioconductor</i> .....	12
3.1.2.1. Paquete <i>Limma</i> .....	13
3.1.2.2. Paquete <i>EdgeR</i> .....	14
3.1.2.3. Paquete <i>DESeq2</i> .....	14
3.2. Búsqueda y obtención de datos para el análisis.....	15
<b>4. Resultados .....</b>	<b>18</b>
4.1. Análisis de expresión génica diferencial empleando los paquetes <i>Limma</i> , <i>EdgeR</i> y <i>DESeq2</i> .....	18
4.1.1. Lectura de los datos .....	18
4.1.2. Preparación de los datos .....	18
4.1.2.1. Filtrado y eliminación de genes con baja expresión.....	18
4.1.2.2. Control de calidad .....	19
Tamaño de las librerías .....	19
Diagramas de caja: análisis de la distribución de los datos .....	21
Gráfico MDS: análisis de escalamiento multidimensional.....	22
Heatmaps .....	24
4.1.3. Análisis con <i>Limma</i> (transformación <i>Voom</i> ).....	26
4.1.3.1. Creación del objeto DGEList .....	26
4.1.3.2. Normalización .....	27
4.1.3.3. Análisis de datos .....	33
Creación de la matriz de diseño y de la matriz de contraste .....	33
Transformación de los datos mediante el método de `voom` .....	34
Análisis de expresión diferencial .....	39

Selección de genes DE .....	41
4.1.4. Análisis con <i>EdgeR</i> .....	43
4.1.4.1. Estimación de las dispersiones .....	43
4.1.4.2. Análisis de expresión diferencial .....	46
Selección de genes DE .....	48
4.1.5. Análisis con <i>DESeq2</i> .....	49
4.1.5.1. Obtención del objeto de la clase DESeq-DataSet.....	50
4.1.5.2. Análisis de expresión diferencial .....	51
Selección de genes DE .....	53
4.2. Comparación de los resultados de expresión diferencial obtenidos con los diferentes paquetes .....	57
4.3. Análisis crítico de los resultados e identificación de las ventajas e inconvenientes implícitas a cada modelo.....	62
<b>5. Conclusiones del estudio .....</b>	<b>67</b>
<b>6. Glosario .....</b>	<b>69</b>
<b>7. Bibliografía .....</b>	<b>71</b>

## **Lista de figuras**

<b>Figura 1.</b> Tamaño de las librerías antes y después del filtrado .....	20
<b>Figura 2.</b> Distribución de los valores de expresión (log2-CPM) de cada una de las muestras en los cuatro estudios seleccionados para el análisis de expresión diferencial.....	21
<b>Figura 3.</b> Gráfico MDS de los valores log2-CPM sobre las dimensiones 1 y 2	23
<b>Figura 4.</b> Mapas de calor de los 1000 genes más variables de cada estudio ..	25
<b>Figura 5.</b> Gráficos MD para las muestras con el mayor ('control2') y menor factor de normalización ('tumor1') del estudio GSE97239 .....	29
<b>Figura 6.</b> Gráficos MD para las muestras con el mayor ('control9') y menor ('control5') factor de normalización del estudio GSE55492 .....	30
<b>Figura 7.</b> Gráficos MD para las muestras con el mayor ('control15') y menor ('paciente9') factor de normalización del estudio GSE92592 .....	31
<b>Figura 8.</b> Gráficos MD para las muestras con el mayor ('tumor20') y menor ('control2') factor de normalización del estudio GSE103001 .....	32
<b>Figura 9.</b> Tendencia de la relación media-varianza observada en los cuatro estudios tras la transformación de los datos mediante el método de Voom	35
<b>Figura 10.</b> Gráfico de la tendencia media-varianza generado a partir de los datos sin filtrar una vez transformada la matriz de conteos brutos mediante Voom vía DGEList .....	37
<b>Figura 11.</b> Gráficos BCV para la visualización de las dispersiones estimadas gen a gen (dispersión tagwise) a partir de la común.....	45
<b>Figura 12.</b> Visualización de los resultados del análisis de expresión génica diferencial llevado a cabo en los cuatro estudios con el paquete <i>EdgeR</i> de <i>Bioconductor</i> .....	49
<b>Figura 13.</b> Visualización de los resultados del análisis de expresión génica diferencial realizado en los cuatro estudios con el paquete <i>DESeq2</i> de <i>Bioconductor</i> .....	55
<b>Figura 14.</b> Diagramas de Venn para la visualización de los genes diferencialmente expresados comunes a dos paquetes .....	60
<b>Figura 15.</b> Comparación de los pasos a seguir y las funciones a ejecutar para completar un análisis de expresión diferencial con los dos paquetes de <i>Bioconductor</i> empleados en este trabajo que se basan en la Binomial Negativa: <i>EdgeR</i> y <i>DESeq2</i> .....	64
<b>Figura 16.</b> Pasos a seguir y las funciones a ejecutar para completar un análisis de expresión diferencial con <i>Limma</i> (con transformación Voom) .....	64

**Figura 17.** Comparación del tiempo de cómputo de los tres paquetes de *Bioconductor* utilizados para el análisis de expresión génica diferencial. Nótese el incremento de tiempo en *EdgeR* y *DESeq2* al aumentar el número de muestras del estudio ..... 65

## Listas de tablas

<b>Tabla 1.</b> Características y diferencias más relevantes entre los principales paquetes empleados por <i>Bioconductor</i> para el análisis de expresión génica diferencial.....	15
<b>Tabla 2.</b> Información de los 4 experimentos seleccionados para el análisis de expresión génica diferencial .....	17
<b>Tabla 3.</b> Condiciones de filtrado establecidas para cada matriz de conteos y porcentaje de genes con baja o nula expresión descartados tras la aplicación del filtro .....	19
<b>Tabla 4.</b> Factor de normalización más bajo y más alto obtenido en cada estudio al normalizar los datos de conteos con la función 'calcNormFactors ()' .....	28
<b>Tabla 5.</b> Número de genes seleccionados como diferencialmente expresados (DE), infra y sobreexpresados, tras el análisis de expresión diferencial llevado a cabo con el paquete <i>Limma</i> de <i>Bioconductor</i> .....	41
<b>Tabla 6.</b> Dispersión común estimada con la función de <i>EdgeR</i> estimateDisp() a partir del correspondiente objeto DGEList de cada estudio.....	44
<b>Tabla 7.</b> Número de genes seleccionados como diferencialmente expresados (DE), infra y sobreexpresados, tras el análisis de expresión diferencial llevado a cabo con el paquete <i>EdgeR</i> de <i>Bioconductor</i> .....	48
<b>Tabla 8.</b> Número de genes seleccionados como diferencialmente expresados (DE), infra y sobreexpresados, tras el análisis de expresión diferencial llevado a cabo con el paquete <i>DESeq2</i> de <i>Bioconductor</i> .....	54
<b>Tabla 9.</b> Número de genes diferencialmente expresados, sobreexpresados e infraexpresados en común entre los diferentes paquetes .....	59
<b>Tabla 10.</b> Número de genes diferencialmente expresados, sobreexpresados e infraexpresados obtenidos tras el análisis con los tres paquetes de <i>Bioconductor</i> más frecuentemente empleados para el análisis de expresión génica diferencial .....	59
<b>Tabla 11.</b> Porcentaje de genes DE incluidos en los paquetes con enfoque más permisivo .....	61
<b>Tabla 12.</b> Resumen de las principales diferencias observadas en el presente estudio entre los tres paquetes de <i>Bioconductor</i> evaluados .....	66

# 1. Introducción

## 1.1. Contexto y justificación del Trabajo

### 1.1.1. Descripción general

En el presente Trabajo Final de Máster (TFM) se comparan los métodos de diferentes paquetes de *Bioconductor* para el análisis de datos de *RNA-Seq* en estudios de expresión génica diferencial. Se describe, para cada uno de los paquetes, los pasos a seguir hasta la identificación de genes diferencialmente expresados y se pone especial énfasis en la comparación de los resultados obtenidos con el fin de identificar las ventajas e inconvenientes implícitas a cada modelo.

### 1.1.2. Justificación del Trabajo Final de Máster

Desde finales de la pasada década, la tecnología NGS (NGS, por sus siglas en inglés *Next Generation Sequencing*) está revolucionando y dominando el ámbito de la investigación genómica al hacer posible la secuenciación masiva a gran escala de ácidos nucleicos de forma fácil y económica (1). Esta tecnología tiene un amplio rango de aplicaciones siendo la más popular su uso en transcriptómica. La secuenciación masiva y profunda de ARNs, cuya técnica se denomina *RNA-Seq*, tiene un alcance mucho mayor a técnicas anteriormente usadas (2). Sin embargo, como toda técnica de secuenciación masiva, *RNA-Seq* genera grandes volúmenes de datos y su análisis supone un enorme desafío a nivel bioinformático. Además, los métodos de análisis no han ido al mismo ritmo que la rápida evolución de estas técnicas (3). Este cuello de botella entre la relación generación de nuevos datos y análisis, ha acelerado precipitadamente el desarrollo de numerosas herramientas con ventajas, limitaciones y metodología a veces poco conocida por los investigadores.

Una buena parte de estas herramientas está enfocada al análisis de datos de uno de los estudios más comunes y relevantes que permite la tecnología *RNA-Seq*: el análisis de expresión génica diferencial o *DGE* (del inglés, *Differential Gene Expression*) (1). Sin embargo, a pesar de que esta tecnología está suponiendo una auténtica revolución en este tipo de estudios y, prácticamente, ha desplazado a los *microarrays*, todavía no se ha decidido la metodología estándar a seguir para el análisis de los datos. De hecho, uno de los pasos principales, la modelización estadística, genera aún hoy en día un intenso debate. Esto es debido a que la mayoría de los métodos estadísticos desarrollados para el análisis de datos de expresión génica se basan en una distribución continua (por ejemplo, datos de *microarrays*) y, como inconveniente, los datos que se obtienen en *RNA-Seq*, al estar relacionados con la abundancia de transcritos de ARNm, son discretos. Por tanto, no pueden aplicarse directamente los algoritmos bien establecidos y consolidados de los *microarrays* y que se basan principalmente en la distribución normal (4).

Para superar esta limitación, se han desarrollado dentro del proyecto *Bioconductor* numerosos métodos basados en distintos modelos probabilísticos. Estos métodos están a disposición de los investigadores en diferentes paquetes siendo los más populares *EdgeR* (5), *Limma* (6) y *DESeq2* (7).

Un mayor conocimiento de estos paquetes permitirá a los investigadores tomar decisiones más acertadas de acuerdo a las características del experimento. Por este motivo, en el presente trabajo se detallan los pasos a seguir para realizar un análisis completo de expresión génica diferencial y se centra el foco de atención en la comparación de los tres programas de *Bioconductor* más destacados.

## 1.2. Objetivos del Trabajo

### 1.2.1. Objetivo general

El objetivo principal de este trabajo fue comparar los métodos de *Bioconductor* actualmente más utilizados para el análisis en R de datos de RNA-seq en estudios de expresión génica diferencial. Para ello, se plantearon los siguientes objetivos específicos:

### 1.2.2. Objetivos específicos

1. Análisis de expresión génica diferencial empleando los paquetes *DESeq2*, *EdgeR* y *Limma*.
2. Comparación de los resultados de expresión génica diferencial obtenidos con los diferentes paquetes.
3. Análisis crítico de los resultados e identificación de las ventajas e inconvenientes implícitas a cada modelo.

## 1.3. Enfoque y método seguido

En este proyecto se llevó a cabo una estrategia bien clara y lineal. En primer lugar, dentro de las distintas opciones que existen a disposición de los investigadores para el análisis de datos de RNA-seq, se eligieron los paquetes más frecuentemente empleados para el análisis de expresión génica diferencial.

Seguidamente, se seleccionaron los datos a analizar en base al tipo de estudio y número de réplicas por grupo experimental. El objeto de estudio fueron matrices de conteos obtenidas de cuatro experimentos reales de RNA-Seq. Estos datos se descargaron de bases de datos de acceso público y se procedió a su lectura en R utilizando las herramientas adecuadas. A partir de aquí se siguió, secuencialmente, una serie de etapas dependiendo del método

de análisis. Hasta este punto se consideró una primera parte e incluyó la consecución del primer objetivo específico.

La segunda parte se centró en la consecución del segundo y tercer objetivo específico. Para ello, se procedió a la comparación de los listados de genes diferencialmente expresados obtenidos con los diferentes paquetes junto con la información recabada de las revisiones y estudios más recientes.

## 1.4. Planificación del Trabajo

En este bloque se concretan las tareas que se realizaron durante el proyecto y se especifican los plazos de entrega así como los posibles factores que podrían haber afectado negativamente a la planificación propuesta.

### 1.4.1. Tareas

De acuerdo a los objetivos establecidos, el proyecto se desglosó en las tareas que se indican a continuación. Se incluye una primera fase de elaboración del Plan de Trabajo así como dos fases finales que corresponden a la creación de la memoria y de la presentación y defensa del Trabajo Final de Máster:

- **Fase 1 (14 días):**

- **Tarea 1.** Elaboración del Plan de Trabajo. Redacción de un documento con los objetivos del proyecto, las tareas a realizar y duración prevista de cada una de ellas, la metodología a seguir e hitos con las fechas clave correspondientes.

- **Fase 2 (37 días):**

- **Tarea 2.** Búsqueda y obtención de datos para el análisis (2 días).
- **Tarea 3.** Búsqueda en la literatura científica de información sobre el funcionamiento de los paquetes *DESeq2*, *EdgeR* y *Limma-Voom* y uso adecuado de *Bioconductor*. Comprobación de nuevas versiones (3 días).
- **Tarea 4.** Control de calidad. Detallada exploración y evaluación de la calidad de los datos antes de proceder al análisis (7 días).
- **Tarea 5.** Normalización de los datos de acuerdo al método empleado por cada paquete (2 días).
- **Tarea 6.** Análisis de expresión diferencial. Obtención de los listados de genes diferencialmente expresados con:
  - *DESeq2* (7 días).
  - *EdgeR* (7 días).
  - *Limma-Voom* (7 días).
- **Tarea 7.** Realización de un informe de seguimiento del proyecto con los resultados obtenidos hasta el momento así como los contratiempos,

dificultades y alteraciones de la planificación surgidos durante la realización de la primera fase (2 días).

▪ **Fase 3 (26 días):**

- **Tarea 8.** Análisis de los resultados obtenidos entre las diferentes metodologías. Comparación de los genes seleccionados como diferencialmente expresados (tanto sobreexpresados como infraexpresados) así como de diferentes parámetros de aplicabilidad (10 días).
- **Tarea 9.** Identificación de las ventajas y desventajas de cada paquete contrastando las conclusiones con la literatura más actual (14 días).
- **Tarea 10.** Realización del segundo informe de seguimiento estructurado de forma similar al primer informe (2 días).

▪ **Fase 4 (+15 días):**

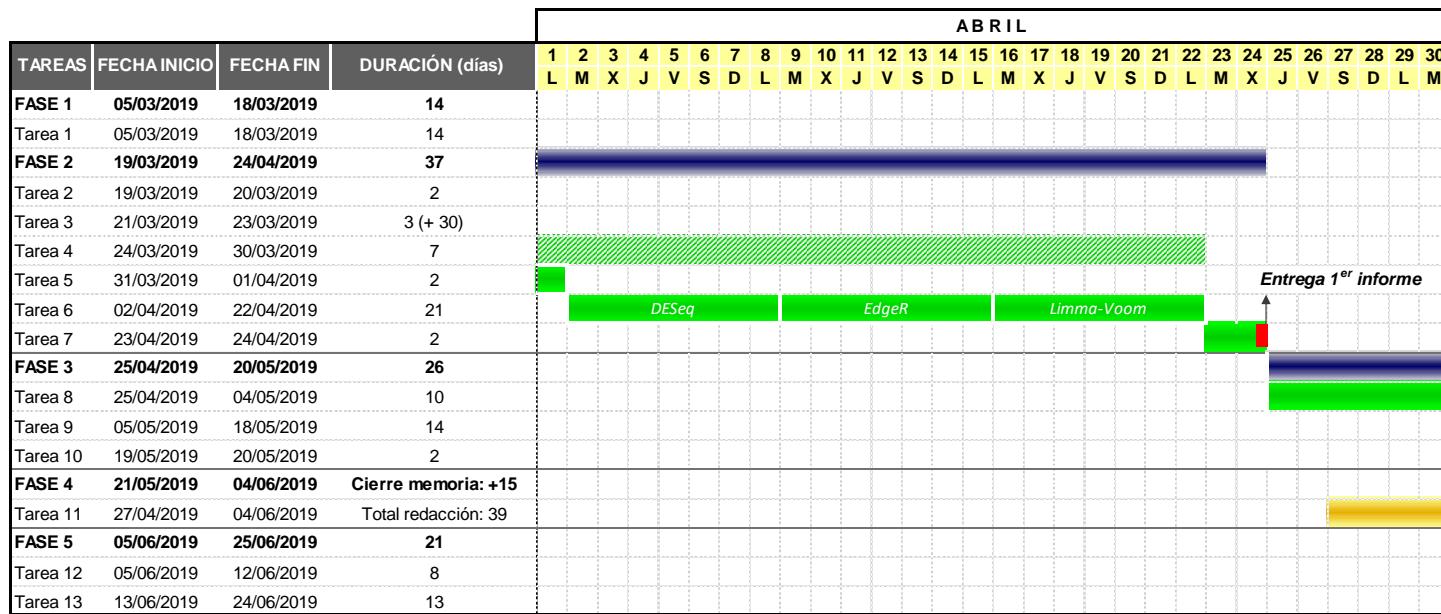
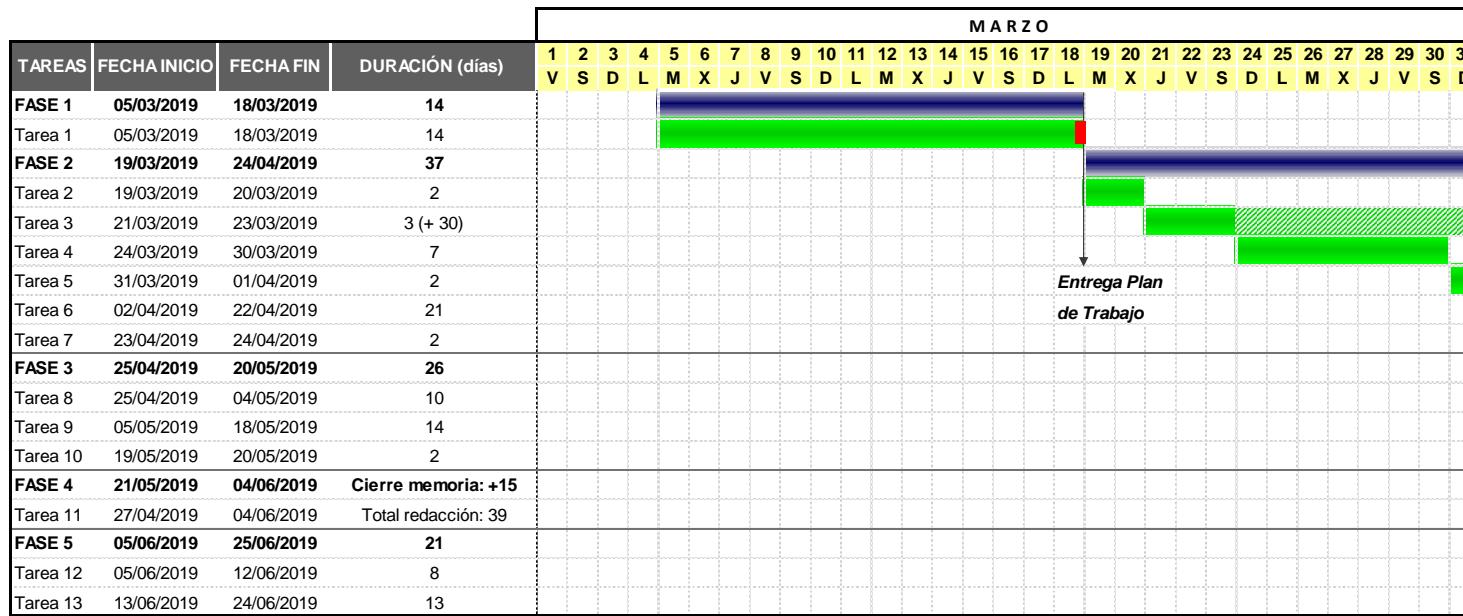
- **Tarea 11.** Redacción de la memoria. Tarea que se inicia a principios de la fase anterior y que se extiende, para el cierre de la memoria, 15 días más desde la finalización de la tercera fase.

▪ **Fase 5 (21 días):**

- **Tarea 12.** Elaboración de la presentación (8 días).
- **Tarea 13.** Defensa virtual del proyecto (diapositivas y vídeo). (13 días).

## 1.5. Calendario

A continuación se muestra el calendario de los meses de marzo a junio con la correspondiente planificación del Trabajo Fin de Máster. Se indican todas las tareas planificadas para la consecución de los objetivos del proyecto así como el tiempo de dedicación para cada una de ellas (diagrama de Gantt). También se indican los hitos y la fecha a la que, de no haberse logrado la/s tarea/s correspondiente/s, se hubiera visto afectado muy negativamente el avance del proyecto.



TAREAS	FECHA INICIO	FECHA FIN	DURACIÓN (días)	M A Y O																																		
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
<b>FASE 1</b>	<b>05/03/2019</b>	<b>18/03/2019</b>	<b>14</b>																																			
Tarea 1	05/03/2019	18/03/2019	14																																			
<b>FASE 2</b>	<b>19/03/2019</b>	<b>24/04/2019</b>	<b>37</b>																																			
Tarea 2	19/03/2019	20/03/2019	2																																			
Tarea 3	21/03/2019	23/03/2019	3 (+ 30)																																			
Tarea 4	24/03/2019	30/03/2019	7																																			
Tarea 5	31/03/2019	01/04/2019	2																																			
Tarea 6	02/04/2019	22/04/2019	21																																			
Tarea 7	23/04/2019	24/04/2019	2																																			
<b>FASE 3</b>	<b>25/04/2019</b>	<b>20/05/2019</b>	<b>26</b>																																			
Tarea 8	25/04/2019	04/05/2019	10																																			
Tarea 9	05/05/2019	18/05/2019	14																																			
Tarea 10	19/05/2019	20/05/2019	2																																			
<b>FASE 4</b>	<b>21/05/2019</b>	<b>04/06/2019</b>	<b>Cierre memoria: +15</b>																																			
Tarea 11	27/04/2019	04/06/2019	Total redacción: 39																																			
<b>FASE 5</b>	<b>05/06/2019</b>	<b>25/06/2019</b>	<b>21</b>																																			
Tarea 12	05/06/2019	12/06/2019	8																																			
Tarea 13	13/06/2019	24/06/2019	13																																			

TAREAS	FECHA INICIO	FECHA FIN	DURACIÓN (días)	J U N I O																																			
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
<b>FASE 1</b>	<b>05/03/2019</b>	<b>18/03/2019</b>	<b>14</b>																																				
Tarea 1	05/03/2019	18/03/2019	14																																				
<b>FASE 2</b>	<b>19/03/2019</b>	<b>24/04/2019</b>	<b>37</b>																																				
Tarea 2	19/03/2019	20/03/2019	2																																				
Tarea 3	21/03/2019	23/03/2019	3 (+ 30)																																				
Tarea 4	24/03/2019	30/03/2019	7																																				
Tarea 5	31/03/2019	01/04/2019	2																																				
Tarea 6	02/04/2019	22/04/2019	21																																				
Tarea 7	23/04/2019	24/04/2019	2																																				
<b>FASE 3</b>	<b>25/04/2019</b>	<b>20/05/2019</b>	<b>26</b>																																				
Tarea 8	25/04/2019	04/05/2019	10																																				
Tarea 9	05/05/2019	18/05/2019	14																																				
Tarea 10	19/05/2019	20/05/2019	2																																				
<b>FASE 4</b>	<b>21/05/2019</b>	<b>04/06/2019</b>	<b>Cierre memoria: +15</b>																																				
Tarea 11	27/04/2019	04/06/2019	Total redacción: 39																																				
<b>FASE 5</b>	<b>05/06/2019</b>	<b>25/06/2019</b>	<b>21</b>																																				
Tarea 12	05/06/2019	12/06/2019	8																																				
Tarea 13	13/06/2019	24/06/2019	13																																				

Como se puede ver, el **mes de marzo** incluyó la elaboración del Plan de Trabajo, búsqueda bibliográfica para la documentación sobre el uso de las diferentes herramientas y métodos de análisis, y la preparación de los datos para la etapa de análisis de expresión diferencial. Aunque pueda parecer excesivo dedicar una semana completa a la Tarea 4 (control de calidad), era fundamental que los datos de partida tuvieran una calidad aceptable para evitar problemas en las siguientes etapas del proceso. En caso de que se detectara algún problema en alguno de los conjuntos de datos seleccionados que no se pudiera depurar, se disponía de un margen de tiempo suficiente como para buscar otro *set* de datos y volver a realizar el análisis de calidad.

El **mes de abril** se dedicó, principalmente, al análisis de expresión diferencial de los datos. Al dedicar una semana completa a cada paquete cualquier retraso se podría haber asumido. Se trata de una de las etapas más complejas del proyecto.

En el **mes de mayo** se trabajó en la consecución del segundo y tercer objetivo específico. Se compararon los listados de genes obtenidos en la Tarea 6 así como diferentes parámetros de aplicabilidad (velocidad de cómputo, facilidad de uso,...) y se identificaron las ventajas e inconvenientes implícitas a cada modelo.

Finalmente, en el **mes de junio** se procedió a la entrega de la memoria, de la presentación y a la defensa del Trabajo Fin de Máster.

## 1.6. Hitos

En este apartado se indican las fechas claves a las que ciertas tareas tenían que estar finalizadas para que el proyecto pudiera seguir avanzando sin poner en riesgo la consecución de los objetivos establecidos. El retraso en el logro de cualquiera de los hitos hubiera afectado a las siguientes tareas así como al cumplimiento de las fechas de entrega previstas. A continuación, se detallan los hitos, fechas clave y fase del proyecto en la que están incluidos:

- ✓ **Entrega del Plan de Trabajo** (18/03/2019). Final de la fase 1.
- ✓ **Entrega del primer informe de seguimiento** (24/04/2019). Final de la fase 2. Listados completos de genes diferencialmente expresados.
- ✓ **Entrega del segundo informe de seguimiento** (20/05/2019). Final de la fase 3. Comparación de los resultados de los diferentes paquetes y conclusiones.
- ✓ **Entrega de la memoria** (04/06/2019). Final de la fase 4.
- ✓ **Entrega de la presentación** (12/06/2019). Final primera semana de la fase 5.
- ✓ **Defensa del Trabajo Fin de Máster** (25/06/2019). Final de la fase 5.

## 1.7. Análisis de riesgos

En la mayoría de los proyectos hay factores y eventos no deseados que pueden amenazar al correcto avance del proyecto o incluso impedir el logro de los objetivos planteados. A continuación, se indican algunos de los problemas que podrían haber surgido durante la realización del proyecto:

1. Falta de tiempo. Es posiblemente el más importante ya que el trabajo se realizó en un plazo muy ajustado. Era importante finalizar las tareas lo antes posible para disponer de un margen y poder asumir eventos imprevistos.
2. Planificación incorrecta. Priorización errónea de las tareas que conlleve a dedicar, innecesariamente, excesivo tiempo a tareas que pueden solaparse con otras más críticas.
3. Mala elección de los experimentos de *RNA-Seq*. Elección de conjuntos de datos problemáticos en la etapa de análisis.
4. Falta de destreza y de conocimientos avanzados en el manejo de las herramientas empleadas en el análisis.
5. Problemas con el ordenador. Lentitud excesiva en el procesamiento de los datos e incluso, poco probable, imposibilidad de la ejecución de ciertos procesos.

## 1.8. Breve sumario de productos obtenidos

Entregables para la calificación del trabajo:

1. **Plan de Trabajo.** Es el presente documento. En él se definen de forma clara los objetivos del proyecto, se establecen las tareas necesarias para alcanzar con éxito los objetivos propuestos y se fija el orden y duración de las mismas. También se incluye una valoración de posibles riesgos que pueden surgir a lo largo del proyecto.
2. **Memoria.** Constituye el documento principal y plasma todo el trabajo realizado. Detalla el procedimiento seguido, los resultados obtenidos y las conclusiones del proyecto.
3. **Presentación virtual.** Exposición oral y visual (formato de vídeo) de los aspectos más importantes del trabajo. Se presentan las distintas fases del proyecto de forma resumida.
4. **Autoevaluación del proyecto.** Documento redactado una vez finalizado el proyecto que incluye una evaluación crítica del trabajo, grado de alcance de los objetivos, y aspectos a mejorar.

## Resultados del estudio:

- Código R y etapas fundamentales para llevar a cabo un análisis de expresión génica diferencial en R empleando diferentes métodos desarrollados por el proyecto *Bioconductor*.
- Listado de genes diferencialmente expresados obtenido a partir de las matrices de conteos de cuatro experimentos reales de RNA-Seq.
- Un mayor conocimiento de las limitaciones, ventajas e inconvenientes de los tres paquetes más frecuentemente utilizados en R para el estudio de expresión génica diferencial.

## 1.9. Breve descripción de los otros capítulos de la memoria

En primer lugar, se realiza una introducción sobre la tecnología de secuenciación NGS, y dentro de ella, se centra el foco de atención en la secuenciación de ARN y los estudios de expresión génica diferencial. Seguidamente, se explica qué es R, R-Studio y *Bioconductor* y se especifican los paquetes usados en este trabajo. También se incluye un apartado donde se indica los datos a analizar, su procedencia e información más relevante que es imprescindible conocer antes de proceder al análisis. A continuación, se expone detalladamente, en un gran bloque denominado *Resultados*, los pasos seguidos para realizar el análisis de los datos con los tres paquetes de *Bioconductor* seleccionados así como la comparación de los resultados obtenidos y la identificación de las limitaciones y ventajas de cada paquete.

## 2. Contexto biológico: NGS y *RNA-Seq*

La secuenciación de alto rendimiento o ultrasecuenciación o también llamada secuenciación de nueva generación (*NGS*), hace referencia a las tecnologías de secuenciación paralela y de alto rendimiento desarrolladas algunas décadas después del método de Sanger (8). Desde que aparecieron a comienzos de la primera década del 2000, estas tecnologías están teniendo importantes repercusiones en la biología humana y la medicina al hacer posible, a diferencia del método tradicional o de Sanger (9), la producción de secuencias a gran escala con una importantísima reducción de costes y del tiempo de secuenciación. Estas ventajas ha llevado a esta tecnología a convertirse en la alternativa a la secuenciación directa de Sanger e irrumpir en los laboratorios de todo el mundo aplicándose a un gran número de campos y empleándose para la resolución de múltiples problemas: determinación de perfiles de expresión génica, detección de cambios epigenéticos, análisis molecular, trastornos inmunitarios, trastornos hereditarios (10), diagnóstico de enfermedades infecciosas (11), diagnóstico prenatal no-invasivo (12) y, más recientemente, en la toma de decisiones terapéuticas para cánceres somáticos (13).

Sin embargo, a pesar de que la ventaja más importante de estas tecnologías es la gran cantidad de datos que producen, su almacenamiento y, sobre todo, el análisis supone un auténtico reto. Los problemas computacionales y bioinformáticos derivados requieren del desarrollo de potentes herramientas de *software* específicas (14).

Actualmente, existen distintas formas de llevar a cabo la secuenciación masiva o ultrasecuenciación. Las principales tecnologías que se utilizan en la actualidad incluyen la secuenciación 454 (de la compañía Roche), la tecnología de Illumina (basada en la tecnología por síntesis de Solexa), la plataforma SOLiD (de la compañía Applied Biosystems), Ion Torrent (de la compañía ThermoFisher) y los secuenciadores de tercera generación manufacturados por Oxford Nanopore y Pacific Biosciences (13).

El rápido progreso de los NGS junto con el desarrollo de herramientas bioinformáticas ha permitido tanto a pequeños como a grandes grupos de investigación usar estas tecnologías en un amplio rango de aplicaciones: secuenciación de ADN (por ejemplo, secuenciación de genomas completos de cualquier organismo de interés), ChIP (interacciones ADN-proteína), epigenómica y secuenciación de ARN (técnica conocida como *RNA-Seq*). Precisamente, esta última es una de las aplicaciones más populares de los NGS (2, 15).

La **tecnología *RNA-Seq*** permite cuantificar la presencia de ARN de una muestra biológica en un momento dado aportando medidas más sensibles y

precisas de la expresión génica, facilitando así, el abordaje de un estudio mucho más fino de lo que se puede conseguir con técnicas previas como los *microarrays*. Esta técnica ha facilitado el hallazgo de *splicing* alternativo, modificaciones postranscripcionales, genes de fusión, mutaciones y polimorfismos de un solo nucleótido (SNPs), y de cambios en la expresión génica en el tiempo o entre diferentes grupos o tratamientos (16-18). Esta última es la aplicación más común del *RNA-Seq* y se conoce como **estudios de expresión génica diferencial o DGE** (2, 19, 20). Estos estudios permiten la elucidación de genes cuyo nivel de expresión es significativamente diferente entre dos o más condiciones experimentales.

El proceso de análisis de los datos de *RNA-Seq* implican diferentes pasos y su correcta realización se antoja fundamental para obtener resultados con significación desde el punto de vista biológico (21, 22).

### 3. Materiales y métodos

#### 3.1. Softwares y paquetes de R usados para el análisis

##### 3.1.1. Entorno R y *R-Studio*

R es un potente lenguaje y entorno de programación destinado al análisis estadístico y la representación de datos (23). Desde la primera versión desarrollada por Robert Gentleman y Ross Ihaka en 1993 (24), este software libre resultado de la implementación GNU del lenguaje de programación S, ha experimentado una expansión irrefrenable hasta el punto de ser elegido por la comunidad científica internacional como la *lingua franca* del análisis de datos. Tanto los desarrolladores como investigadores y usuarios pueden escribir extensiones y distribuir los códigos a otros usuarios. El código se distribuye en forma de librerías constituyendo paquetes listos para su utilización. Precisamente, las aportaciones continuas y desinteresadas de paquetes de propósito tanto de interés general como específico hacen de R un entorno dinámico y versátil formado por una comunidad activa y adherida a la filosofía del software libre.



R se distribuye bajo licencia GNU y el archivo de instalación se puede descargar desde la página oficial <http://www.r-project.org/>.

Para la realización de este trabajo se ha accedido a R a través de la interfaz que ofrece *RStudio* (25), un IDE (*Integrated Development Environment*, o Entorno de Desarrollo Integrado) de código abierto para R. Por su parte, *R-Studio* también puede instalarse de manera gratuita desde el sitio oficial de *RStudio* <https://www.rstudio.com>.



Una de las grandes ventajas de R y *R-Studio* es que al tratarse de softwares libres podemos tener siempre instalada la última versión. En este trabajo se ha usado la versión 3.5.3 de *R-Studio*.

##### 3.1.2. *Bioconductor*

*Bioconductor* es un proyecto de código abierto cuyos orígenes se remontan al 2001 con el objetivo de desarrollar un software que integrara el lenguaje R y proporciona herramientas para el análisis estadístico de datos de laboratorio en biología molecular, especialmente, de datos genómicos de alto rendimiento (26).



Para instalar *Bioconductor*, previamente es necesario tener instalado el programa R. Una vez iniciado R se puede descargar la última versión de *Bioconductor* ejecutando los siguientes comandos tal y como indica la página oficial <https://www.bioconductor.org/install/>.

```

if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install()

```

o bien accediendo directamente a Internet desde R y usar un *script* de instalación ya creado:

```

if(!require(Bioconductor)){
  source("http://bioconductor.org/biocLite.R")      # Buscar Bioconductor en la web
  biocLite(Bioconductor)                            # Instalar la librería Bioconductor
}
library(Bioconductor)                                # Cargar la librería Bioconductor

```

Para este trabajo se usó la versión 3.8 de *Bioconductor*.

Aunque la principal aplicación del proyecto *Bioconductor* ha sido desde su origen proporcionar un conjunto amplio y poderoso de métodos para el análisis de datos de *microarrays*, el anunciado principio del fin de esta técnica ya en 2008 (27, 28) y, prácticamente, hecho realidad a día de hoy debido a las ventajas que ofrece la secuenciación *RNA-Seq* (29), ha precipitado necesariamente la búsqueda de herramientas para el análisis de los datos que genera esta técnica. Para ello, se ha recurrido tanto la adaptación a *RNA-Seq* de paquetes bien consolidados para el análisis de datos de *microarrays* como el desarrollo de nuevos programas basados en técnicas estadísticas especializadas. Gracias a este esfuerzo, actualmente *Bioconductor* dispone de numerosos paquetes para el análisis de datos de *RNA-Seq*.

Los paquetes de *Bioconductor* más utilizados para el análisis de este tipo de datos y que se estudiaron en profundidad en este proyecto se detallan a continuación.

### 3.1.2.1. Paquete *Limma*

Para instalar *Limma*, una opción es iniciar R y ejecutar:

```

source("http://bioconductor.org/biocLite.R")
biocLite("limma")

```

*Limma* proporciona una solución integrada para el análisis de datos de experimentos de expresión génica. Diseñado originalmente para el análisis de datos de *microarrays* (30), sus capacidades se han ampliado en una segunda dirección para permitir el análisis de datos de *RNA-Seq* con *pipelines* muy similares (6).

Debido a que el modelo probabilístico que usa este paquete, modelos lineales, no es fácilmente aplicable a datos discretos como los que se generan en *RNA-Seq*, el análisis bajo *Limma* de los datos de esta última tecnología incluye un paso previo de transformación (*Voom* o *Vst*) que abre el acceso a la aplicación de las herramientas anteriormente restringidas a los *microarrays*. En este trabajo se ha usado, concretamente, el método de *Voom* que se basa en

la estimación de la relación media-varianza de los conteos en escala logarítmica (log2) para generar las correspondientes ponderaciones a nivel de observación (*precision weights*) para su posterior aplicación y estimación bayesiana (31).

Para el filtrado de los transcritos con muy bajo número de lecturas, en *Limma* se establece un mínimo de conteos por millón (CPM).

Respecto al método de normalización por defecto es el denominado TMM (del inglés, *Trimmed mean of M values*, media truncada de M-valores) propuesto por Robinson y Oshlack en 2010 (32). Se trata de una forma simple pero robusta de estimar la producción relativa de ARN de dos muestras. La producción de ARN total no puede estimarse directamente puesto que se desconocen los niveles de expresión y longitudes reales de cada gen.

### 3.1.2.2. Paquete *EdgeR*

Para instalar *EdgeR*, una opción es iniciar R y ejecutar:

```
source("https://bioconductor.org/biocLite.R")
biocLite("edgeR")
```

*Edge R* es un paquete para el análisis de datos de *RNA-Seq* en estudios de expresión génica diferencial basado en la distribución Binomial Negativa (5).

El uso de esta distribución como un modelo para recuentos de lectura ha suscitado mucho interés especialmente en el problema de estimar la variabilidad biológica para experimentos con un pequeño número de réplicas. *EdgeR* cuando ajusta el modelo binomial negativo, estima los coeficientes de variación biológica (BCV) antes de proceder al análisis. El BCV se calcula como la raíz cuadrada del parámetro dispersión bajo la Binomial Negativa lo que equivale a estimar la dispersión del modelo.

El enfoque de ajustar un valor global o tendencia global a las dispersiones del modelo tiene la limitación de no permitir la variación específica de genes. En este contexto, los test de hipótesis para comprobar expresión diferencial sólo son teóricamente precisos cuando la dispersión es pequeña (33).

### 3.1.2.3. Paquete *DESeq2*

Para instalar *DESeq2*, una opción es iniciar R y ejecutar:

```
source("https://bioconductor.org/biocLite.R")
biocLite("DESeq2")
```

Este tercer paquete de *Bioconductor* usado en este proyecto, proporciona las herramientas necesarias para analizar expresión diferencial mediante el uso de modelos de regresión binomial negativos (7).

Aunque *DESeq2* y *EdgeR* usan el mismo modelo probabilístico y siguen estrategias similares difieren en algunos aspectos, siendo el más relevante el método de normalización. *EdgeR* usa el método TMM mientras que *DESeq2* divide los conteos de cada gen en una muestra por el número total de lecturas en dicha muestra. En la práctica, los factores de normalización a menudo son similares pero la diferencial más crucial se encuentra en la forma de estimar la dispersión. *EdgeR* modera las estimaciones de dispersión a nivel de gen hacia una tendencia media de acuerdo a la relación media-dispersión, y *DESeq2* toma el valor máximo entre las estimaciones de dispersión individuales y la tendencia media de la dispersión (22).

Las principales diferencias entre los tres paquetes de *Bioconductor* se resumen en la **Tabla 1**.

**Tabla 1. Características y diferencias más relevantes entre los principales paquetes empleados por *Bioconductor* para el análisis de expresión génica diferencial.**

	<i>Limma</i>	<i>EdgeR</i>	<i>DESeq2</i>
<b>Filtrado</b>	CPM (conteos por millón)	CPM	Media datos para cada transcripto
<b>Método de normalización</b>	TMM (media truncada de M-valores)	TMM	<i>Normalization</i>
<b>Modelo probabilístico</b>	Modelo lineal	Binomial negativo	Binomial negativo
<b>Test</b>	Prueba <i>t</i> -Student	Test exacto basado en la distribución binomial	Paramétrico de Wald

### 3.2. Búsqueda y obtención de datos para el análisis

Para este trabajo se seleccionaron cuatro experimentos reales de RNA-Seq del repositorio público de datos genómicos funcionales **Gene Expression Omnibus** (GEO) (<https://www.ncbi.nlm.nih.gov/geo/>). El título de cada estudio así como el correspondiente identificador GEO se indica a continuación:

- **GSE97239:** *High-throughput RNA sequencing on circular RNA profiles of human bladder cancer tissues and normal bladder tissues.*
- **GSE55492:** *Gene expression profile of calcified and normal tricuspid aortic valves by RNA sequencing.*
- **GSE92592:** *mRNA Sequencing of Ideopathic Pulmonary Fibrosis (IPF) and Control Samples from the Lung Tissue Research Consortium (LTRC).*

- **GSE103001:** *Stranded RNASeq of human mammary primary tumors ER+ and paired adjacent healthy tissues.*

La elección de estos experimentos estuvo motivada por:

- El tipo de estudio
- El número de muestras de cada grupo experimental
- La naturaleza/procedencia de las muestras

Respecto al primer aspecto, los cuatro experimentos se englobaban dentro de los estudios de tipo comparación de grupos, concretamente **estudios de casos y controles**. Por otra parte, y en relación al segundo punto, se prestó especial atención de elegir experimentos que reflejaran una amplia gama de tamaños de muestras: desde estudios con pocas muestras/réplicas por grupo experimental lo cuál es lo más frecuente en la mayoría de los experimentos actuales de RNA-Seq, hasta estudios con un elevado número de muestras en cada grupo. Esto ofrece la posibilidad de analizar si el número de genes diferencialmente expresados (DE) seleccionados por los diferentes paquetes se ve afectado por el número de muestras. Finalmente, respecto al tercer punto, se eligieron dos estudios relacionados con cáncer (GSE97239 y GSE103001), un estudio relacionado con la valvulopatía más frecuente (GSE55492) y un estudio relacionado con una enfermedad inflamatoria intersticial pulmonar (GSE92592).

En la **Tabla 2** se muestra la información más relevante de cada experimento que es importante conocer antes de proceder al análisis de expresión diferencial.

**Tabla 2. Información de los 4 experimentos seleccionados para el análisis de expresión génica diferencial. Se indica el número total de muestras, número de casos y controles y una breve descripción de las mismas.**

GEO.ID	Plataforma	Total de muestras	C A S O S		C O N T R O L E S	
			Muestras	Descripción	Muestras	Descripción
GSE97239	GPL11154	6	3	Tumores de vejiga	3	Tejidos de vejiga adyacentes normales
GSE55492	GPL11154	18	9	Válvulas aórticas tricúspide calcificadas	9	Válvulas aórticas tricúspide no calcificadas
GSE92592	GPL11154	37	19	Biopsias de pulmón de pacientes con fibrosis pulmonar idiopática	18	Biopsias de pulmón de pacientes sanos
GSE103001	GPL11154	40	21	Carcinomas de mama invasivo primario. Receptores tumorales: ER + / HER2	19	Tejidos mamarios adyacentes normales

**Plataforma GPL11154: Illumina HiSeq 2000 (*Homo sapiens*)**

## 4. Resultados

### 4.1. Análisis de expresión génica diferencial empleando los paquetes *Limma*, *EdgeR* y *DESeq2*

En este bloque se expondrán las etapas seguidas secuencialmente para el análisis de datos de *RNA-seq* en estudios de expresión génica diferencial y se mostrarán los resultados obtenidos en cada una de ellas. Se describirá detalladamente la metodología para los tres paquetes de *Bioconductor* seleccionados.

#### 4.1.1. Lectura de los datos

El análisis de expresión diferencial se realizó a partir de las matrices de conteos que se obtienen tras la alineación y el mapeo de las secuencias frente al genoma de referencia. Las matrices de conteos contienen el número de lecturas que se alinean de forma única con los exones de un gen dado en cada una de las muestras analizadas. Los genes se disponen en filas y las muestras en columnas.

Las matrices empleadas en este trabajo para el análisis de expresión diferencial presentaban, concretamente, 35238 filas (genes) y tantas columnas como muestras incluidas en el estudio (ver **Tabla 1** del apartado 3.2. *Búsqueda y obtención de datos para el análisis*).

Los datos de conteos brutos sin normalizar se importaron en R como archivos de extensión .csv una vez cargada la librería que permite trabajar con *Bioconductor*, la librería ‘Biobase’.

#### 4.1.2. Preparación de los datos

Esta etapa incluyó dos pasos muy importantes que deben realizarse antes de proceder al análisis: el filtrado de genes con baja o nula expresión y la evaluación de la calidad de los datos.

##### 4.1.2.1. Filtrado y eliminación de genes con baja expresión

Generalmente, en las matrices de conteos hay un cierto número de genes que presentan pocas o ninguna lectura en todas o la mayoría de las muestras. Estos genes, lógicamente, no son de interés puesto que el objetivo de los estudios de expresión génica diferencial es determinar aquellos genes cuyo nivel de expresión es significativamente diferente entre los distintos grupos experimentales a comparar.

Por este motivo, en este paso se procedió al filtrado de los genes que no mostraban un nivel de expresión significativo ni en el grupo control ni en el grupo de casos.

Para ello, en primer lugar se realizó una transformación de los datos con el fin de tener en una misma escala todas las muestras de un mismo estudio y evitar, así, diferencias debido al distinto tamaño de las librerías. La transformación que se llevó a cabo fue la conversión de las lecturas a CPM (conteos por millón).

Una vez calculados los CPM y con todas las muestras de un mismo experimento con la misma profundidad de secuenciación, se aplicaron las condiciones de filtrado (**Tabla 3**).

**Tabla 3.** Condiciones de filtrado establecidas para cada matriz de conteos y porcentaje de genes con baja o nula expresión descartados tras la aplicación del filtro. El valor de corte se calculó en función de la profundidad de secuenciación y de acuerdo a la regla general que aconseja retener los genes que tienen, al menos, 10 lecturas.

	GSE97239	GSE55492	GSE92592	GSE103001
<b>Mayor profundidad de secuenciación</b>	5243503	34086113	52402249	34251347
<b>Menor profundidad de secuenciación</b>	2666019	4927402	18107245	5351536
<b>Número de genes antes del filtrado</b>	35238	35238	35238	35238
<b>Condición de filtrado</b>	5 CPM en, al menos, 3 muestras	2 CPM en, al menos, 9 muestras	0,6 CPM en, al menos, 18 muestras	2 CPM en, al menos, 19 muestras
<b>Número de genes después del filtrado</b>	14059	13406	16635	15166
<b>Genes descartados del análisis</b>	21179 (60,1%)	21832 (62,0%)	18603 (52,8%)	20075 (57,0%)

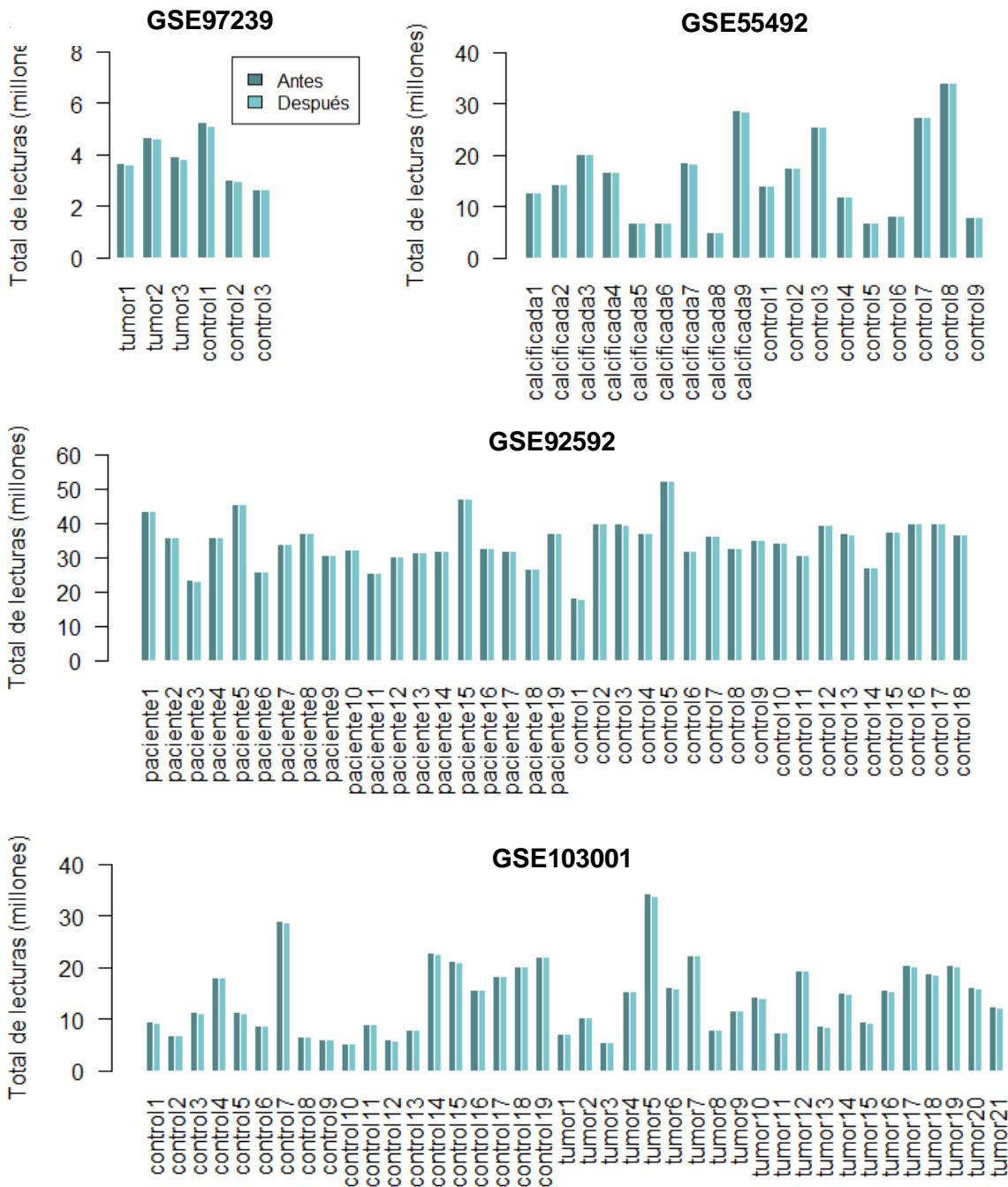
#### 4.1.2.2. Control de calidad

Una vez reducido considerablemente el conjunto de genes a analizar tras la eliminación de aquellos con poca o nula expresión, se procedió a la exploración y evaluación de la calidad de los datos.

Es fundamental partir de datos con una calidad aceptable para evitar problemas en la fase posterior de análisis. En caso detectarse un problema difícil de depurar en alguna muestra, se debe plantear su exclusión del conjunto de datos antes de proceder al análisis de expresión diferencial.

#### Tamaño de las librerías

Aunque previamente se determinó el tamaño de las librerías con una mayor y menor profundidad de secuenciación (**Tabla 3**), se procedió a la realización de un análisis más completo. Para ello, se representó en un diagrama de barras el total de lecturas en cada muestra antes y después del filtrado (**Figura 1**).



**Figura 1. Tamaño de las librerías antes y después del filtrado.**

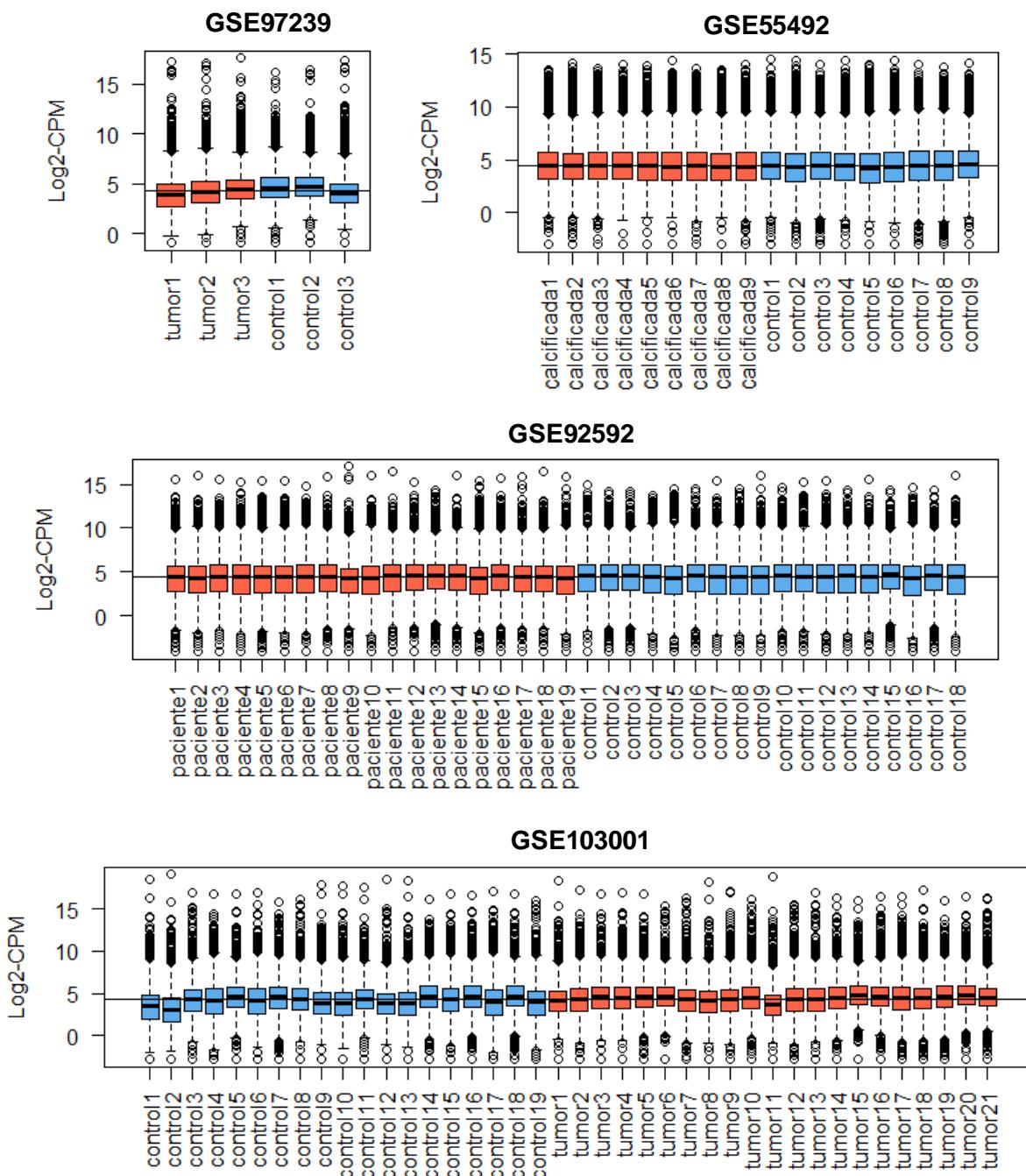
En los cuatro conjuntos de datos se observó diferencias importantes en la profundidad de secuenciación de las muestras que formaban parte de un mismo experimento. Estas diferencias pusieron de manifiesto la necesidad de llevar a cabo una normalización de los datos antes de proceder al análisis de expresión diferencial.

La reducida diferencia del tamaño de las librerías antes y después del filtrado que se apreció en todas las muestras, concuerda con la eliminación del estudio únicamente de los genes con pocas o nulas lecturas. Por este motivo, aunque se descartó del estudio más de la mitad de los genes de cada conjunto

(ver última fila de la **Tabla 3**), el total de lecturas prácticamente no se vio modificado.

#### **Diagramas de caja: análisis de la distribución de los datos**

La distribución de los conteos en cada una de las muestras se analizó mediante la generación de diagramas de caja a partir de los correspondientes valores de CPM en escala logarítmica de base 2 (**Figura 2**).



**Figura 2. Distribución de los valores de expresión (log2-CPM) de cada una de las muestras en los cuatro estudios seleccionados para el análisis de expresión diferencial. Diagramas de caja generados a partir de los datos filtrados pero no normalizados.**

Como se aprecia en la **Figura 2**, todos los *boxplots* obtenidos fueron similares no apreciándose, dentro de un mismo estudio, cajas notablemente desplazadas hacia arriba o hacia abajo con respecto a las demás.

Una vez descartada, por el momento, la presencia de muestras problemáticas se continuó con la evaluación de la calidad de los datos y la generación de uno de los gráficos más importantes en la exploración de los datos: el gráfico MDS (por sus siglas en inglés, *Multidimensional Scaling*).

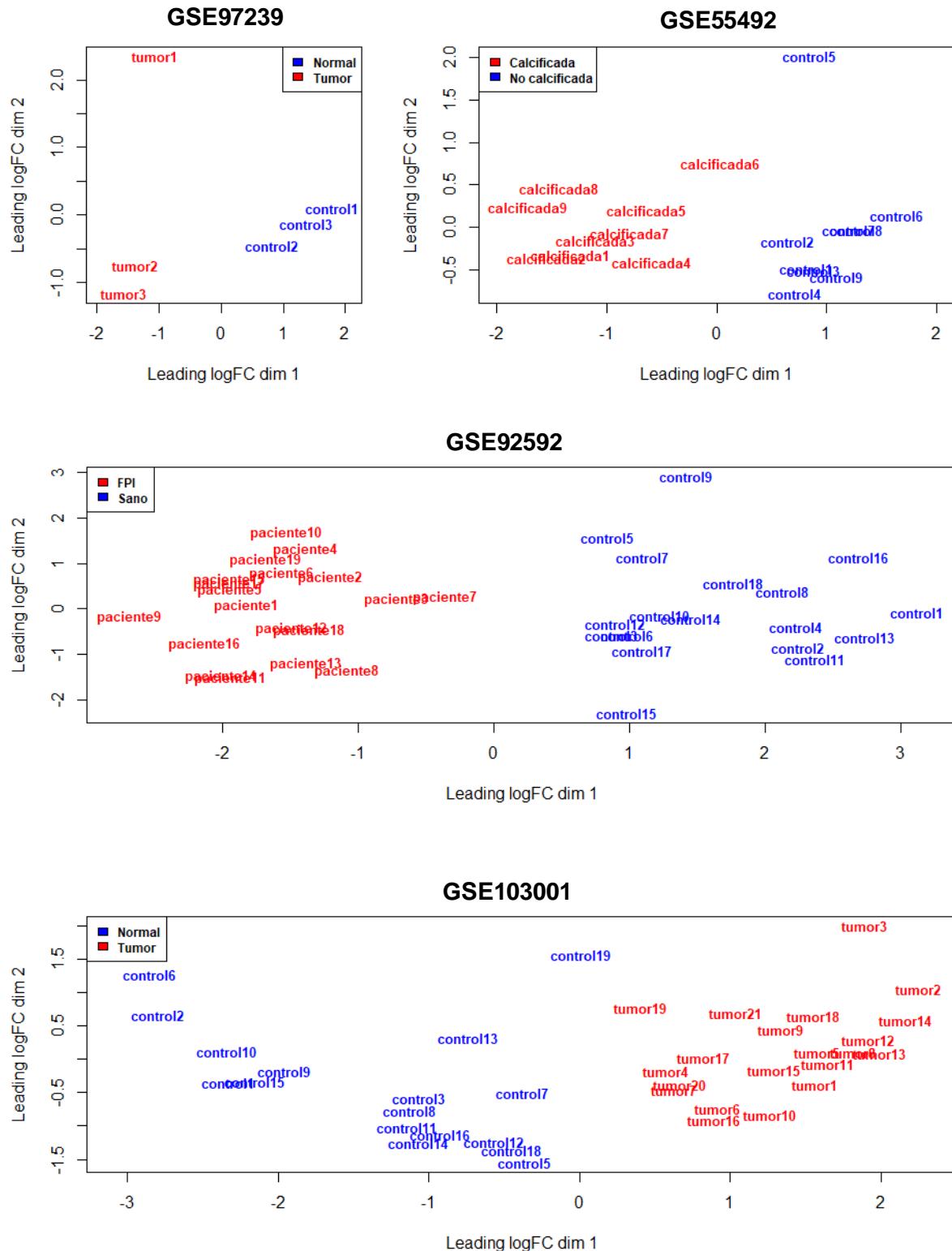
#### **Gráfico MDS: análisis de escalamiento multidimensional**

Para examinar la relación entre muestras y detectar expresión diferencial antes de pasar a las pruebas formales, se recurrió al análisis de escalamiento multidimensional (**Figura 3**), una técnica multivariante que permite analizar visualmente la proximidad entre las muestras de un mismo estudio al situarlas en unas dimensiones determinadas.

En un gráfico MDS, la primera dimensión representa la magnitud del cambio biológico (*fold-change*) inicial que mejor separa las muestras y que explica, por tanto, la mayor proporción de variación en los datos. A la vista de los gráficos generados (**Figura 3**), las muestras se separaron en los cuatro estudios de acuerdo a su grupo experimental. Se descartó la presencia de muestras atípicas puesto que no se observó ninguna con un comportamiento más propio del grupo contrario.

La disposición de los controles y los casos en valores del eje X de signo opuesto apuntó a la existencia de genes diferencialmente expresados entre ambos grupos.

La segunda dimensión es ortogonal a la primera, constituye la segunda fuente de variación y tiene un efecto menor en la expresión diferencial de los genes.



**Figura 3.** Gráfico MDS de los valores log2-CPM sobre las dimensiones 1 y 2. Nótese la separación en los cuatro estudios entre las muestras de casos (en rojo) y las muestras control (en azul).

### **Heatmaps**

Para finalizar la exploración, se generó un *heatmap* o mapa de calor por cada conjunto de datos (**Figura 4**).

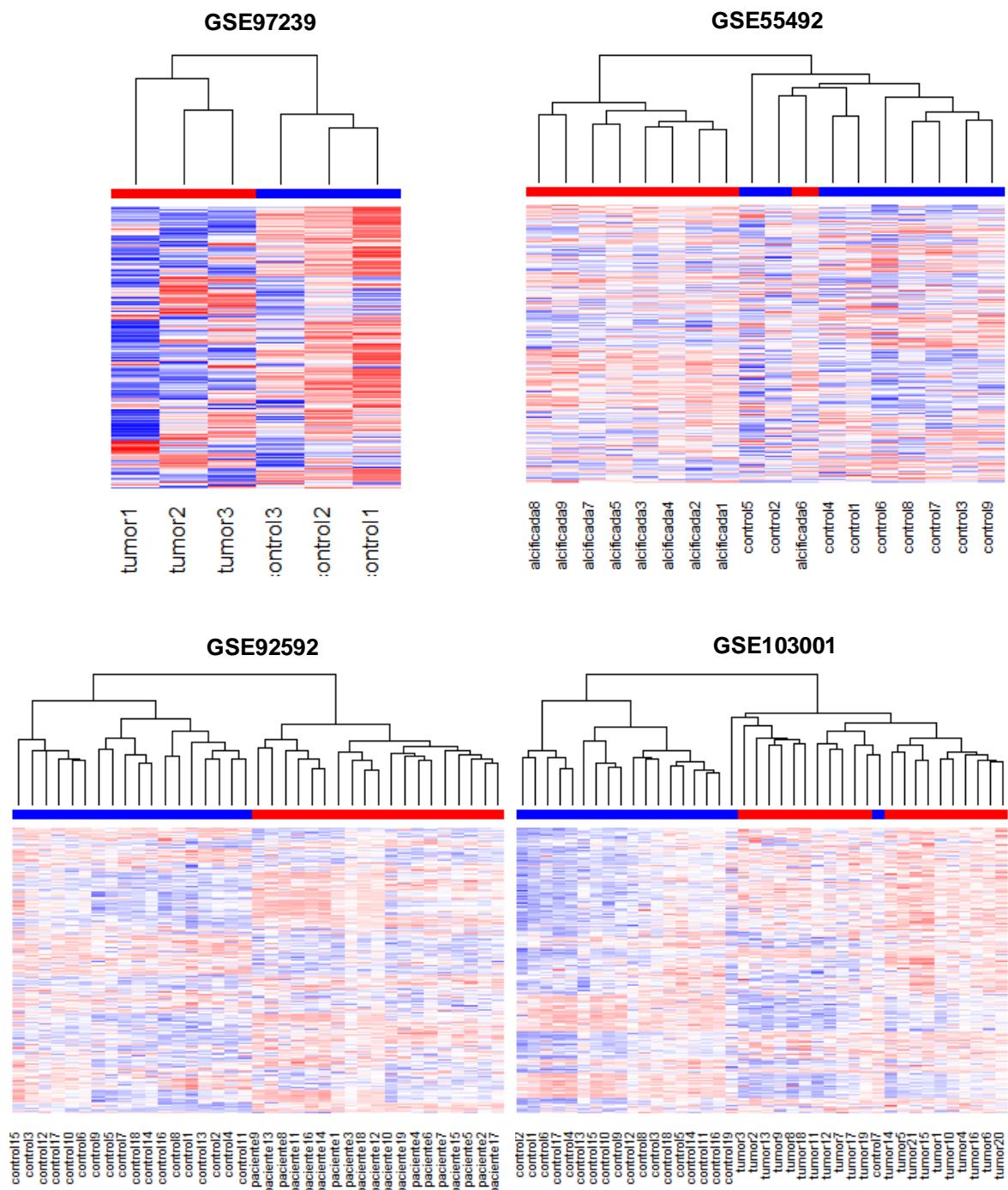
Este tipo de representación permite visualizar la agrupación de las muestras sobre la base de un *cluster* jerárquico junto con los niveles de expresión de genes individuales.

Para generar este tipo de gráfico, previamente se calculó la varianza en cada una de las filas de la matriz log2-CPM y se estableció el número de genes a visualizar.

La selección de los 1000 genes más variables agrupaba a las muestras de acuerdo al grupo experimental (ver *dendograma de la parte superior de los gráficos*, **Figura 4**).

La separación de las muestras en dos grandes grupos (casos y control) fue absolutamente rigurosa en el caso de los estudios GSE97239 y GSE92592. Sin embargo, una muestra del grupo de casos en GSE55492 ('calcificada6') se agrupaba con las muestras control; y, en GSE103001, a la inversa: dos controles ('control19' y 'control7') se agrupaban con las muestras de casos.

Aunque estas tres muestras se encontraban separadas de su grupo en el mapa de calor, no se descartaron del análisis puesto que en el gráfico MDS (**Figura 3**) se disponían en un logFC (dimensión 1) de cero pero, en ningún caso, aparecían en valores de *fold-change* del grupo contrario. Por tanto, no había evidencias claras de asociación atípica como para excluir estas muestras del análisis de expresión génica posterior.



**Figura 4. Mapas de calor de los 1000 genes más variables de cada estudio. Los genes sobreexpresados se representan en rojo, los infrarregulados en azul y el color blanco indica ausencia de cambio de expresión. Cada fila de la cuadricula representa un gen y cada columna una muestra.**

Una vez finalizado el análisis de control de calidad y comprobado que las muestras se comportaban de acuerdo a lo esperado, se procedió al análisis de expresión génica diferencial con los tres paquetes de *Bioconductor* más populares: *Limma*, *EdgeR* y *DESeq2*.

Para simplificar, a partir de este punto se mostrarán únicamente las instrucciones ejecutadas para el análisis de los datos correspondientes al experimento con el menor número de muestras (GSE97239). Este estudio se utilizará a modo de ejemplo para explicar los pasos seguidos con cada método. No obstante, los gráficos con información relevante se mostrarán para cada estudio y al final de este bloque se podrá visualizar una tabla general con los resultados que se obtuvieron con los cuatro estudios.

#### 4.1.3. Análisis con *Limma* (transformación Voom)

##### 4.1.3.1. Creación del objeto DGEList

El primer paso fue la creación de un objeto DGEList a partir de la matriz de conteos y la información de las muestras:

```
y <- DGEList(counts = filtrado_bladder, group= targets)

# Argumento 'counts': matriz de conteos filtrada
# Argumento 'group': información sobre el grupo/condición experimental para
# cada muestra/librería.
```

Una vez creado este objeto se comprobó que en '\$counts' se encontraba correctamente almacenada la matriz numérica con el recuento de lecturas:

```
head(y$counts)

##          tumor1 tumor2 tumor3 control1 control2 control3
## A1BG        48     59     62      76      91      34
## A2M         370    296    203    3926    1749     803
## A2ML1       19     876     55      66      68      21
## A4GALT      57     61     34     127      90      47
## AAAS        28     74     89     112      77      41
## AACs        246    272    329     215     194     143
```

y, en '\$samples', la información necesaria para el análisis: grupo experimental, recuento total (profundidad de secuenciación) para cada librería y factores de normalización:

```
y$sample

##          group lib.size norm.factors
## tumor1    tumor  3614680           1
## tumor2    tumor  4606623           1
## tumor3    tumor  3797256           1
## control1  normal  5122019           1
## control2  normal  2951828           1
## control3  normal  2628740           1
```

En todas las muestras la columna correspondiente al factor de normalización presentaba el valor de 1 puesto que no se especificó en el argumento ‘norm.factors’ de la función `DGEList()` ningún vector numérico de factores de normalización para la ponderación del tamaño de las librerías. Esta columna se actualizó en un paso posterior de acuerdo al ‘*effective library size*’ (tamaño efectivo de las librerías).

#### 4.1.3.2. Normalización

En el punto 4.1.2.1. *Filtrado y eliminación de genes con baja expresión*, se llevó a cabo la transformación de los valores de lecturas a CPM con el objetivo de conseguir una normalización por el tamaño de librería. Al tener en una misma escala todas las muestras se evita la aparición de falsos positivos (a mayor profundidad de secuenciación, mayor es la probabilidad de obtener genes diferencialmente sobreexpresados).

Sin embargo, la decisión de normalizar los datos no sólo se debe basar en el tamaño de las librerías, también es imprescindible tener en cuenta las variaciones biológicas entre las muestras con el fin de eliminar los sesgos de composición. Si no se realiza un ajuste, aparecerán genes falsamente infraregulados en aquellas muestras donde un pequeño número de genes consuman una proporción importante del total de lecturas pero no en otras muestras donde esto no ocurra.

El método más popular de normalización es el TMM. Este método es el que realiza por defecto la función `calcNormFactors()` y el que emplea tanto *Limma* como *EdgeR*:

```
y <- calcNormFactors(object=y)

# Argumento 'object': objeto de clase DGEList que se desea normalizar.
# Demás argumentos mantenidos con los valores asignados por defecto.
```

La ejecución de la instrucción anterior actualizó los factores de normalización:

```
y$samples

##          group lib.size norm.factors
## tumor1    tumor   3614680   0.7403481
## tumor2    tumor   4606623   0.9443866
## tumor3    tumor   3797256   1.1244274
## control1  normal  5122019   1.2158825
## control2  normal  2951828   1.3337480
## control3  normal  2628740   0.7843636
```

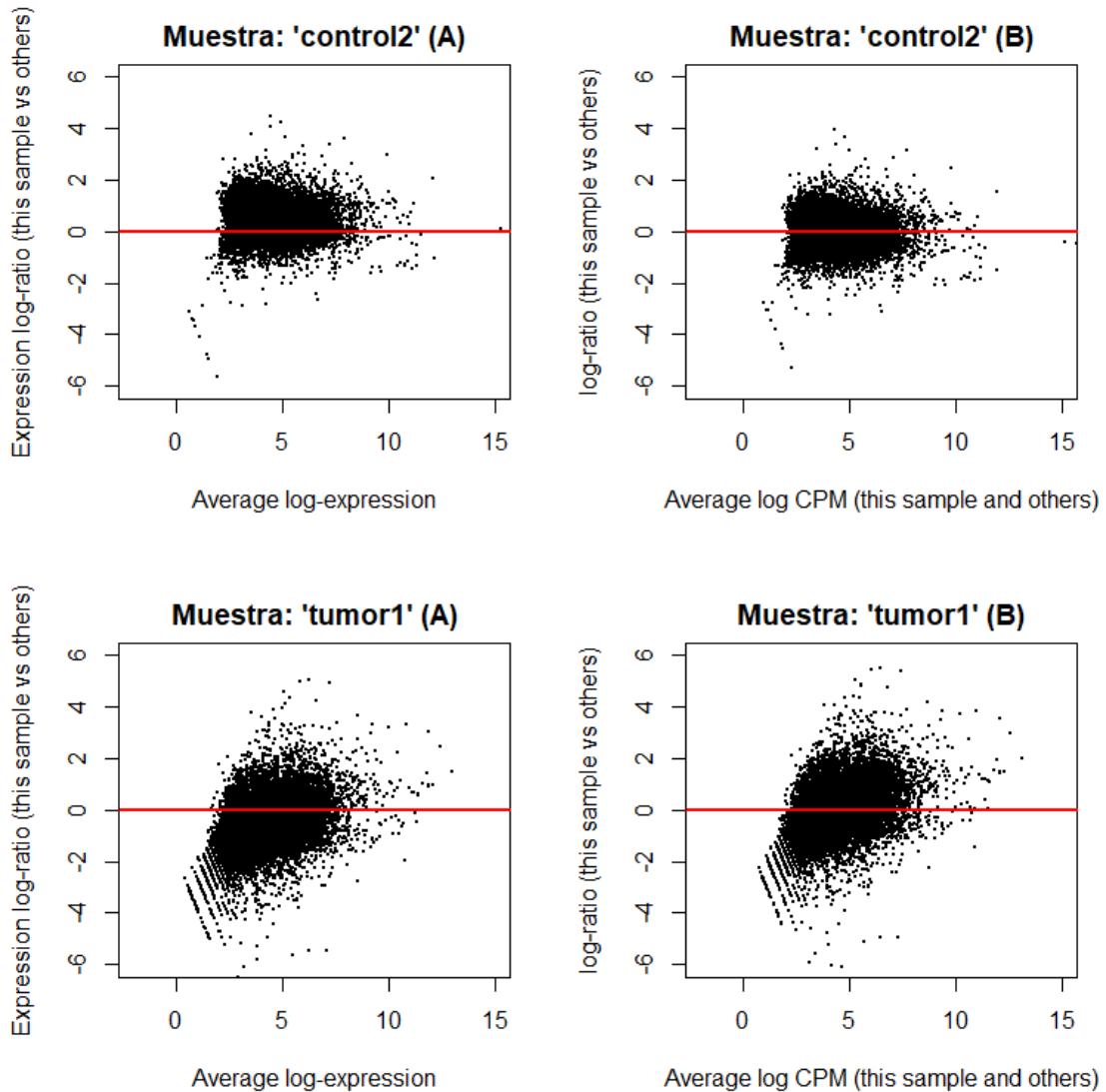
En el caso del estudio GSE97239, se aprecia que los factores de normalización variaron entre 0.7403481 para la muestra con el factor de normalización más bajo ('tumor1') y 1.3337480 para la muestra con el factor más alto ('control2'). La **Tabla 4** recopila esta información en los cuatro estudios.

**Tabla 4. Factor de normalización más bajo y más alto obtenido en cada estudio al normalizar los datos de conteos con la función 'calcNormFactors ()'.**

	GSE97239	GSE55492	GSE92592	GSE103001
<b>Factor normalización mayor</b>	1.3337480	1.1100617	1.1536226	1.4403092
<b>Muestra</b>	'control2'	'control9'	'control15'	'tumor20'
<b>Factor normalización menor</b>	0.7403481	0.8663421	0.8179093	0.4266666
<b>Muestra</b>	'tumor1'	'control5'	'paciente9'	'control2'

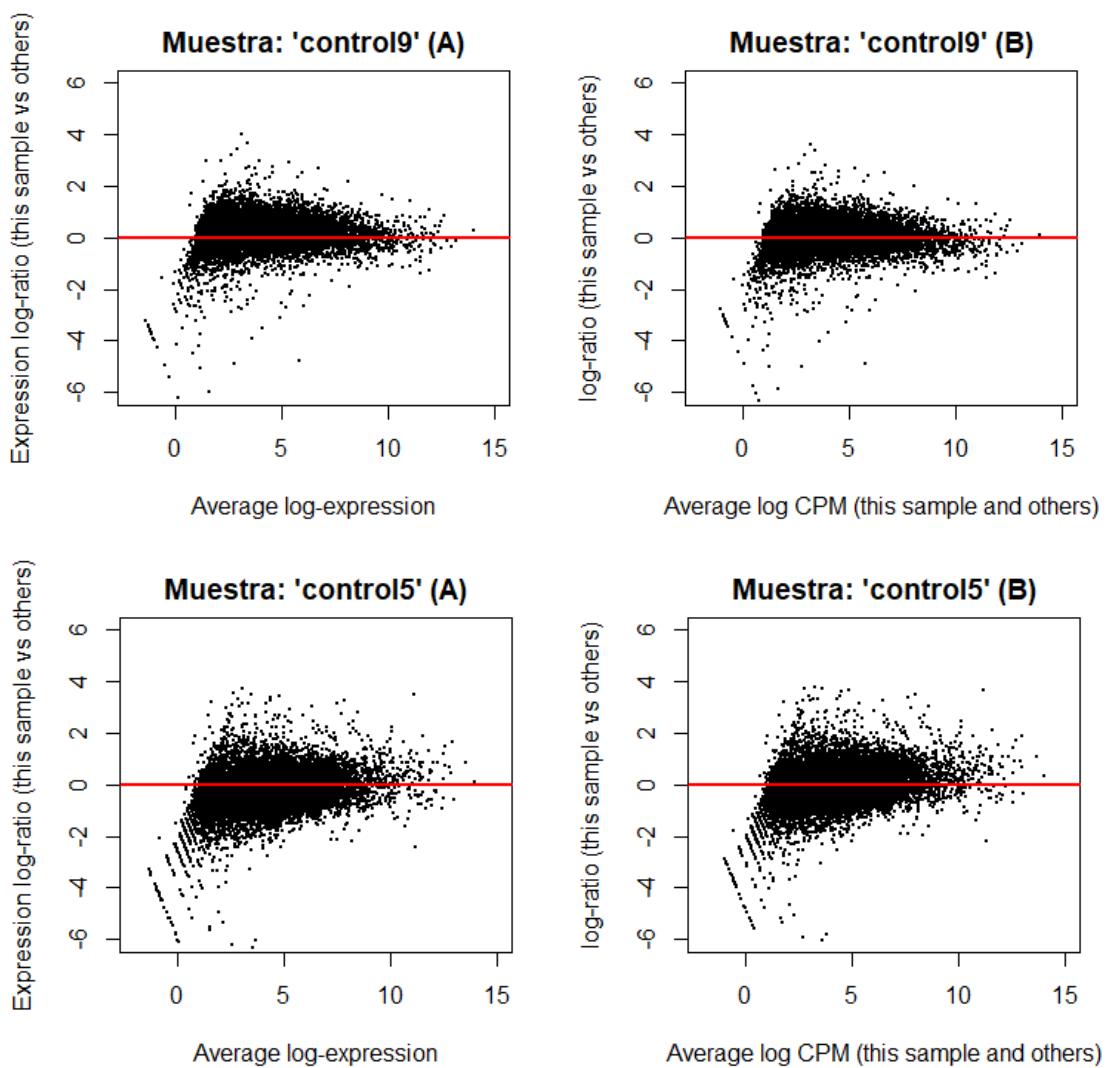
En el caso de GSE55492 y GSE92592 los factores de normalización variaban en un rango de valores entorno al 1, lo que indicaba que el efecto de la normalización TMM en ambos estudios era leve. Sin embargo, en el caso de GSE97239 y, más aún en GSE103001, el rango de los factores de normalización incluía valores muy alejados de 1 lo que significaba que el sesgo de composición en las librerías de estos dos estudios era muy acusado.

Debido a que el efecto de normalización difería entre estudios y, en uno de ellos, era considerable, se decidió explorar a fondo los perfiles de expresión de las muestras indicadas en la **Tabla 4**. Para ello, se procedió a la generación de los correspondientes gráficos MD (diferencia de medias) antes y después de la normalización TMM (**Figuras 5, 6, 7 y 8**).



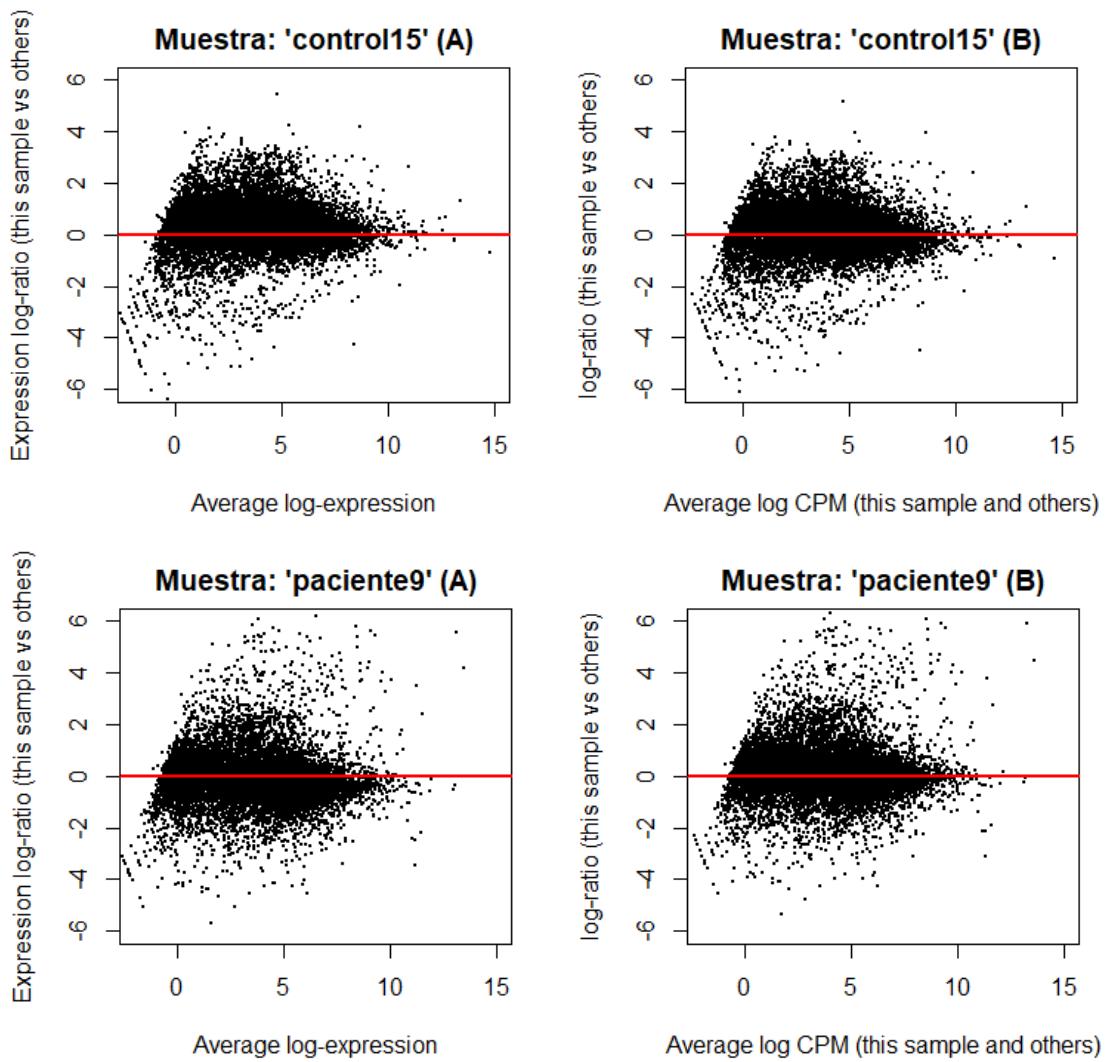
**Figura 5.** Gráficos MD para las muestras con el mayor ('control2') y menor factor de normalización ('tumor1') del estudio GSE97239. (A) Valores de expresión normalizados por el tamaño de librería pero no por el sesgo de composición; (B) Gráficos MD tras la normalización de los valores de expresión mediante el método TMM.

Los gráficos MD del estudio GSE97239 (**Figura 5**) fueron acordes con los valores de los factores de normalización de las muestras representadas. Antes de la normalización TMM se observó, en el caso de la librería con el factor más alto ('control2'), una mayor proporción de genes (cada gen es un punto del gráfico) en valores de *fold-change* positivos. Por el contrario, en la librería con el factor más bajo ('tumor2'), los genes se situaron ligeramente desplazados hacia valores de *fold-change* negativos. La comparación de los gráficos MD previos y posteriores a la normalización permitió verificar la correcta eliminación del sesgo de composición una vez realizado el ajuste de los datos con el método TMM.



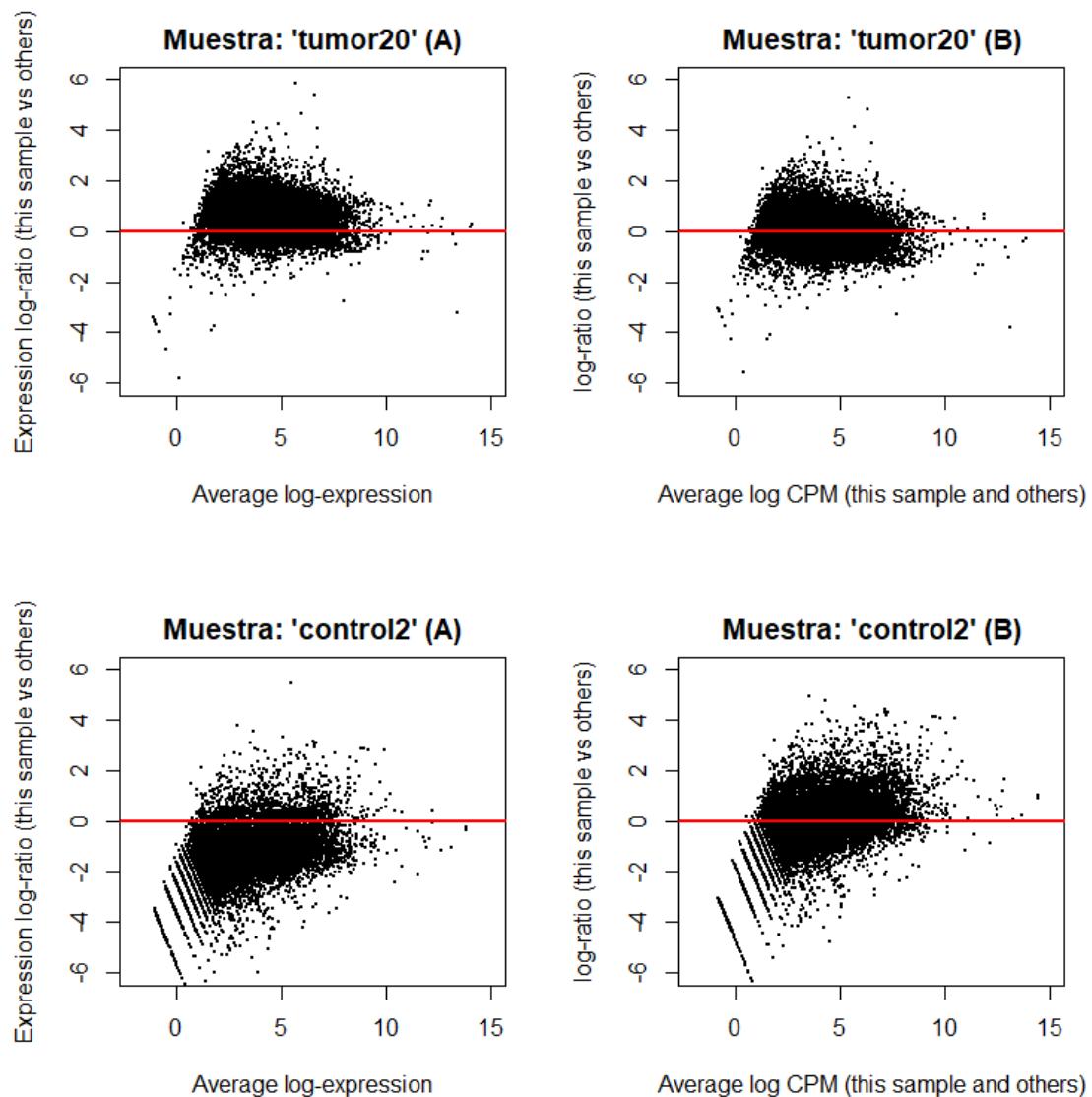
**Figura 6.** Gráficos MD para las muestras con el mayor ('control9') y menor ('control5') factor de normalización del estudio GSE55492. (A) Valores de expresión normalizados por el tamaño de librería pero no por el sesgo de composición; (B) Gráficos MD tras la normalización de los valores de expresión mediante el método TMM.

Debido a que el sesgo de composición en el estudio GS 55492 es muy leve, los gráficos MD que se obtuvieron tras el paso de normalización (**Figura 6 (A)**) fueron muy similares a los generados a partir de los datos no normalizados (**Figura 6 (B)**). Los genes ya se situaban alrededor de la línea que representa el *fold-change* cero (línea roja) antes de la normalización, sin apreciarse un desplazamiento de los puntos hacia la mitad superior o inferior del gráfico.



**Figura 7.** Gráficos MD para las muestras con el mayor ('control15') y menor ('paciente9') factor de normalización del estudio GSE92592. (A) Valores de expresión normalizados por el tamaño de librería pero no por el sesgo de composición; (B) Gráficos MD tras la normalización de los valores de expresión mediante el método TMM.

En el estudio GSE92592 se apreció una situación similar a la descrita en GSE55492. Los gráficos previos y posteriores a la normalización TMM fueron muy parecidos (**Figura 7**) debido a que el sesgo de composición en las librerías de este estudio también resultó ser muy pequeño (**Tabla 4**).



**Figura 8.** Gráficos MD para las muestras con el mayor ('tumor20') y menor ('control2') factor de normalización del estudio GSE103001. (A) Valores de expresión normalizados por el tamaño de librería pero no por el sesgo de composición; (B) Gráficos MD tras la normalización de los valores de expresión mediante el método TMM.

Como se aprecia en la **Figura 8**, los gráficos MD que se obtuvieron antes y después de la normalización TMM fueron considerablemente diferentes. Como ya se ha comentado, cuanto más alejado de 1 estén los factores de normalización, mayor es el sesgo de composición. Por este motivo, los puntos del gráfico MD de la muestra con el factor de normalización más alto presentaban un claro desplazamiento por encima de la línea roja que representa el *fold-change* cero (sesgo negativo) (**Figura 8**, gráfico superior izquierdo) y, en el caso de la muestra con el factor de normalización más bajo, un notable desplazamiento de los puntos por debajo de la línea roja (**Figura 8**, gráfico inferior izquierdo). Tras la normalización TMM, se verificó que la situación acabada de describir fue corregida.

#### 4.1.3.3. Análisis de datos

Como se comentó en el apartado 3.2. *Búsqueda y obtención de datos para el análisis*, la característica común a los cuatro estudios que se eligieron para este trabajo es que se trataba de estudios de casos y controles. Es evidente que la finalidad del análisis era, por tanto, comparar la expresión génica entre estos dos grupos sin entrar a analizar las diferencias que pudieran existir en el perfil transcriptómico entre muestras pertenecientes a un mismo grupo.

Por este motivo, era fundamental definir una matriz de diseño basada en el diseño experimental y una matriz de contraste que estableciera la comparación a realizar.

##### **Creación de la matriz de diseño y de la matriz de contraste**

Para generar la matriz de diseño del modelo lineal a partir del objeto que contiene la información sobre el grupo experimental al que pertenece cada muestra, se empleó la función `model.matrix()` de *Limma*. A continuación, se muestra el código de R empleado para la creación de la correspondiente matriz de diseño del estudio GSE97239:

```
design <- model.matrix(~ 0 + targets) # Eliminación del término constante.
colnames(design) <- levels(y$sample$group)
rownames(design) <- samples
design

##          normal tumor
## tumor1      0     1
## tumor2      0     1
## tumor3      0     1
## control1    1     0
## control2    1     0
## control3    1     0
## attr("assign")
## [1] 1 1
## attr("contrasts")
## attr("contrasts")$targets
## [1] "contr.treatment"
```

En dicha matriz, la presencia de un 1 era indicativo de pertenencia de dicha muestra al grupo experimental en cuestión y, un 0, lo contrario.

A partir de la información especificada en la matriz de diseño se creó la matriz de contraste que establecía la comparación de interés: grupo control *versus* grupo de casos. Para ello, se empleó la función `makeContrast()`, se asignó un nombre a la comparación y se definió el contraste: casos – control.

```

cont.matrix <- makeContrasts (tumorVscontrol = tumor - normal, levels = design)
cont.matrix

##          Contrasts
## Levels   tumorVscontrol
##   normal           -1
##   tumor            1

```

De acuerdo a la comparación establecida se asignó el valor de -1 al grupo control y 1 al de casos.

### ***Transformación de los datos mediante el método de `voom`***

Debido a que en *Limma* el ajuste del modelo lineal no se realiza directamente en los datos de conteos brutos, previamente fue necesario 1º) llevar a cabo una transformación que permitiera asumir una distribución normal de las observaciones, y 2º) realizar un ajuste de cada dato de acuerdo a la tendencia de la varianza respecto de la media.

El propio paquete *Limma* ofrece una función que permite procesar los datos de un experimento de *RNA-seq* antes de realizar el ajuste del modelo. Se trata de la función `voom()` y realiza los siguientes tres pasos:

- Transformación de los conteos en log2-CPM (conteos por millón en escala logarítmica de base 2).
- Estimación de la relación media-varianza de los conteos en escala logarítmica
- Cálculo de la ponderación apropiada para cada observación para su posterior aplicación y estimación bayesiana (eBayes).

*Voom* extrae automáticamente la información necesaria para realizar la transformación (matriz de conteos, tamaño de cada librería, factores de normalización) y produce un objeto de la clase EList:

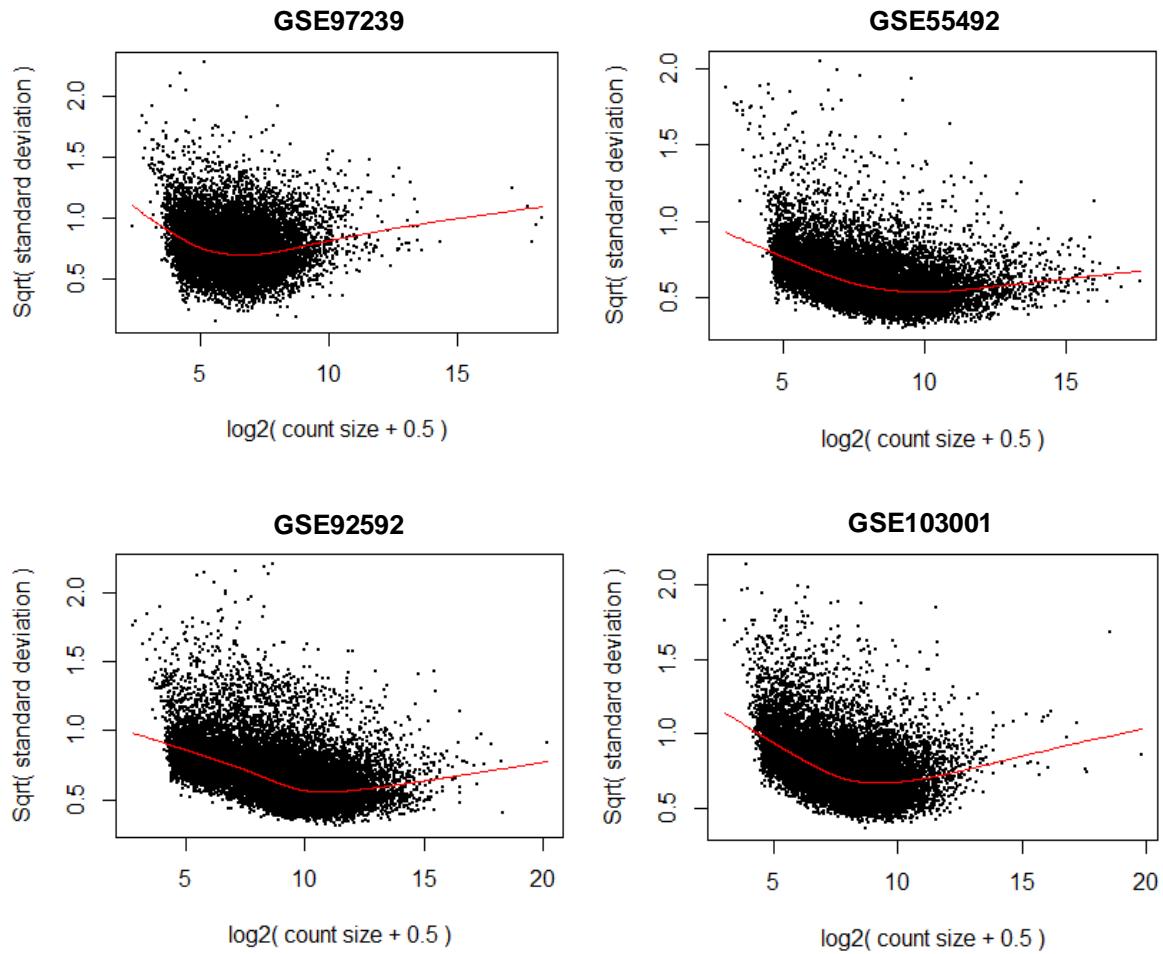
```

v <- voom(counts = y, design = design, plot = TRUE)

# Argumento 'counts': matriz de conteos brutos
# Argumento 'design': matriz de diseño
# Argumento 'plot': si es TRUE, se genera el gráfico de la tendencia media-varianza.

```

Al indicar TRUE en el argumento ‘`plot`’, *Voom* crea automáticamente el correspondiente gráfico de la tendencia media-varianza una vez realizada la transformación de los datos. Este tipo de gráfico se genera al representar, para cada gen, la correspondiente media en el eje x y la correspondiente varianza en el eje y lo que permite obtener una visión general de la dispersión de los datos. Los gráficos obtenidos en los cuatro estudios se muestran en la **Figura 9**.



**Figura 9. Tendencia de la relación media-varianza observada en los cuatro estudios tras la transformación de los datos mediante el método de Voom. Cada punto del gráfico representa un gen.**

Este tipo de gráfico resultó especialmente importante para el análisis de dos aspectos muy relevantes:

- **Variación biológica entre las muestras:** cuanto menor es esta variación, más brusco es el descenso de la tendencia. Como muestra la **Figura 9**, aunque en GSE97239 y GSE103001 el descenso de la varianza a valores de conteos bajos fue algo más acusado que en los otros dos estudios, los perfiles obtenidos en todos los casos eran compatibles con una alta variación biológica entre las muestras. Inicialmente, se aprecia cómo la varianza desciende de forma moderada pero seguidamente se incrementa a medida que aumenta el número de conteos. Esta observación fue compatible con la disposición de las muestras en el gráfico MD generado en el apartado 4.1.2.2. *Control de calidad, Figura 3*. Efectivamente, aunque la dimensión 1 separaba las muestras rigurosamente de acuerdo al grupo experimental (control, casos), la dimensión 2 que constituye la segunda fuente de la variación observada en la expresión

diferencial de los genes, puso de manifiesto una elevada variabilidad entre las muestras.

La menor variabilidad que se observó en los estudios GSE97239 y GSE103001 probablemente fuera debida a que las muestras formaban pares de casos-control: cada par procedente de un mismo individuo. En el caso de GSE55492, cada control se obtuvo a partir de tejido sano de vejiga situado adyacente al tumor y, de forma equivalente, en GSE103001 cada control fue extraído de tejido mamario sano adyacente al carcinoma.

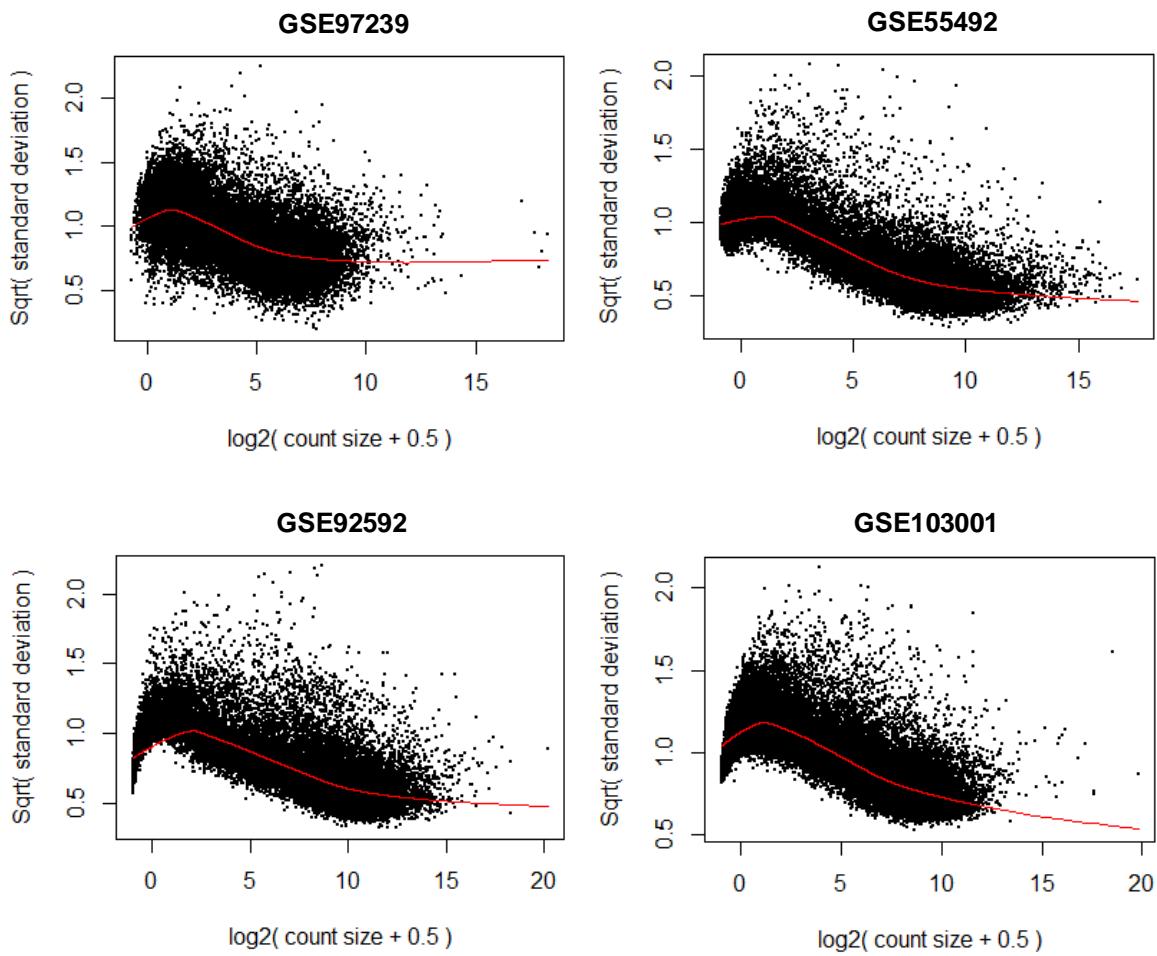
Sin embargo, en los estudios donde se aprecia una ligera mayor variabilidad, GSE55492 y GSE92592, todas las muestras procedían de individuos diferentes independientemente del grupo experimental al que pertenecían.

La descripción de las muestras analizadas en cada estudio se puede consultar en la **Tabla 2**.

- **Calidad del filtrado:** si el filtrado de los genes con baja o nula expresión no se realiza correctamente, aparecerá una caída de la varianza hacia el extremo izquierdo de la curva.

Como no se apreció tal descripción en ninguno de los gráficos, se consideró que el paso de filtrado se realizó correctamente y el umbral de expresión aplicado a la matriz de conteos en el apartado 4.1.2.1. fue el adecuado.

Para comparar los gráficos de la **Figura 9** con los que se hubieran obtenido en caso de no haber eliminado correctamente los genes con pocas o cero lecturas, se procedió a representar la tendencia media-varianza de los datos sin filtrar (**Figura 10**).



**Figura 10.** Gráfico de la tendencia media-varianza generado a partir de los datos sin filtrar una vez transformada la matriz de conteos brutos mediante *Voom* vía DGEList. La tendencia aumenta rápidamente para conteos bajos, disminuyendo gradualmente a medida que aumenta el número de conteos.

Como era de esperar y de acuerdo a la explicación anterior, a conteos bajos el trazado de la línea roja que representa la tendencia, fue notablemente diferente cuando se tomaron para la representación los datos de la matriz sin filtrar. Lógicamente, los recuentos de cero o muy próximos a cero (valor -1 en el eje x) tienen desviaciones estándar bajas pero éstas aumentan inmediatamente en cuanto se incrementa mínimamente el número de lecturas. A partir de un valor del eje x de, aproximadamente 1.5 - 2, la varianza se reduce gradualmente.

La función `voom()` no sólo genera el gráfico de la tendencia media-varianza como se acaba de ilustrar si no que, además, produce un objeto de la clase EList que conserva la información del objeto DGEList:

- ‘targets’ (EList) es equivalente a ‘samples’ (DGEList):

```
v$targets
```

```
##          group lib.size norm.factors
## tumor1      tumor  2676121    0.7403481
## tumor2      tumor  4350433    0.9443866
## tumor3      tumor  4269739    1.1244274
## control1    normal  6227773    1.2158825
## control2    normal  3936995    1.3337480
## control3    normal  2061888    0.7843636
```

- ‘E’ (EList) contiene los valores de expresión guardados en ‘counts’ (DGEList):

```
head(v$E)
```

```
##          tumor1 tumor2 tumor3 control1 control2 control3
## A1BG     4.179769 3.773658 3.871636 3.618671 4.538605 4.064558
## A2M      7.113186 6.090729 5.574737 9.300312 8.795632 8.606188
## A2ML1    2.865258 7.654451 3.700268 3.416566 4.120937 3.382298
## A4GALT   4.425346 3.821355 3.014376 4.355637 4.522751 4.525889
## AAAS     3.412746 4.098009 4.389668 4.175065 4.299029 4.331073
## AACs     6.525300 5.968953 6.269987 5.112827 5.626531 6.120940
```

Adicionalmente, el objeto Elist también guarda la matriz de pesos en ‘weight’:

```
head(v$weights)
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 3.576920 4.181201 4.163729 4.406367 4.173552 3.295990
## [2,] 3.954113 3.406348 3.427885 1.725533 1.953167 2.346483
## [3,] 4.243275 4.392763 4.398115 4.301320 3.827516 2.706786
## [4,] 3.365136 4.046252 4.025992 4.316608 4.363762 3.729901
## [5,] 3.601947 4.197533 4.179975 4.383676 4.284977 3.521511
## [6,] 3.957743 3.410331 3.431640 3.503009 4.017020 4.405642
```

así como la matriz de diseño en ‘design’:

```
v$design
```

```
##          normal tumor
## tumor1      0     1
## tumor2      0     1
## tumor3      0     1
## control1    1     0
## control2    1     0
## control3    1     0
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$targets
## [1] "contr.treatment"
```

Una vez se realizó la transformación mediante el método de *Voom*, los datos ya estaban listos para el ajuste de los modelos lineales empleando, para ello, comandos del paquete *Limma* originalmente diseñados para su aplicación en el análisis de datos de *microarrays*.

### Análisis de expresión diferencial

Para la identificación de genes diferencialmente expresados para la comparación de interés, se realizaron los siguientes pasos:

- Ajuste de los modelos lineales a los datos transformados por *Voom* mediante la función ‘*lmFit()*’.

Esta función que incorpora el paquete *Limma* ajusta el modelo lineal para cada gen a partir de los valores de expresión en escala logarítmica.

```
# 1.- Ajuste de los modelos lineales:  
  
vfit <- lmFit(object = v, design = design)  
# Argumento 'object': matriz con los valores de expresión.  
# Argumento 'design': matriz de diseño.
```

- Estimación del contraste mediante las funciones ‘*contrast.fit()*’ y ‘*eBayes()*’.

```
# 2.- Estimación de los contrastes:  
  
fit.cont <- contrasts.fit(fit = vfit, contrast = cont.matrix)  
# Argumento 'fit': salida de la función 'lmFit()' .  
# Argumento 'contrast': matriz de contraste.  
  
##Análisis diferencial basado en métodos bayesianos:  
efit <- eBayes(fit.cont)
```

- Realización de los test de significación para cada gen empleando la función ‘*decideTests()*’.

```
# 3.- Realización de los test de significación:  
  
dt.fit <- decideTests(object = efit)  
# Argumento 'object': salida de la función 'eBayes()' .  
# Argumento 'lfc': valor de log2(fold change). Por defecto es cero,  
# lo que equivale a un valor de fold change de 1 en valor absoluto.
```

Es importante destacar que la selección de los genes se realiza en base al valor de *p*-ajustado. El nivel de significación fue del 5% que es el que establece la función por defecto.

Tras el análisis de expresión diferencial se procedió a la visualización de los genes más diferencialmente expresados entre los dos grupos experimentales utilizando la función `topTable()`. Una vez tomada como entrada la salida de `eBayes`, esta función generó una lista de genes ordenados de menor a mayor valor de  $p$ . Los genes que se muestran a continuación corresponden a los 10 primeros resultados del listado:

```
topTable(fit = efit, sort.by = "p")
#>
##          logFC AveExpr      t    P.Value adj.P.Val      B
##   FHL1    -4.879290 5.109872 -9.081535 5.088584e-06 0.01891861 4.365735
##   PGM5    -5.073452 6.994604 -9.072015 5.135130e-06 0.01891861 4.327973
##   ANKAR   -3.249913 4.422253 -8.636287 7.856699e-06 0.01891861 4.083704
##   PYCR1    3.691905 4.768896  8.503830 8.971290e-06 0.01891861 3.966945
##   DUSP2   -4.002962 5.895204 -8.361720 1.036238e-05 0.01891861 3.831522
##   SLIT3   -3.106184 5.771688 -8.196708 1.228017e-05 0.01891861 3.715641
##   ITGA8   -3.741230 4.794095 -8.195335 1.229766e-05 0.01891861 3.678407
##   ADAM33  -2.976524 5.010492 -8.103789 1.352812e-05 0.01891861 3.641032
##   CFD     -4.489443 6.680166 -8.057850 1.419567e-05 0.01891861 3.515478
##   MATN2   -3.770635 5.015290 -7.839918 1.789170e-05 0.01891861 3.352713
# Argumento 'fit': salida de la función 'eBayes()' .
# Argumento 'sort.by': estadístico a considerar al ordenar el listado.
# Argumento 'number': por defecto, visualización de los primeros 10 genes.
# Argumento 'coef': identificador de la comparación, por defecto 1
# (una comparación).
```

La salida de `topTable()` indicó, para cada gen, el valor de los siguientes estadísticos:

- `logFC`: *log<sub>2</sub>-fold-change* entre las dos condiciones experimentales.
- `AveExpr`: valor medio de expresión de cada gen en todas las condiciones experimentales
- `t`: valor del estadístico *t* de *Student*.
- `P.Value`: valor de  $p$  asociado al estadístico *t*.
- `adj.P.Value`: valor de  $p$  ajustado para el control de la tasa de falsos positivos.
- `B`: *odds ratio*, representa la probabilidad de que el gen se exprese diferencialmente.

Para una correcta interpretación de los valores del estadístico `logFC`, fue imprescindible tener en cuenta la comparación que se estableció en la matriz de contraste. Una comparación definida como “grupo control – grupo de casos” implica que los valores de `logFC` positivos representan genes cuyo nivel de expresión es inferior en los casos que en los controles, es decir, se trata de genes infraexpresados y, por el contrario, los valores de `logFC` negativos indican que dichos genes se expresan más en los casos que en los controles, es decir, se trata de genes sobreexpresados.

En el siguiente apartado se determinó el número de genes cuyo nivel de expresión era significativamente diferente entre los dos grupos de estudio y se concretó, dentro de estos genes, el número de sobreexpresados e infraexpresados.

### **Selección de genes DE**

Puesto que la función `decideTest()` genera una matriz numérica de elementos -1, 0 y 1 dependiendo de si el valor del estadístico  $t$  obtenido en cada prueba es clasificado, respectivamente, como significativamente negativo, no significativo o significativamente positivo, la obtención del número de genes de cada clasificación se pudo obtener fácilmente mediante el empleo de la función genérica `summary()`:

```
summary(dt.fit)
-----
##          tumorVscontrol
## Down           292
## NotSig        13578
## Up            189
```

Como indica esta salida, de los 14059 genes que pasaron la fase de filtrado (estudio GSE97239), 481 genes presentaban un nivel de expresión estadísticamente diferente entre los dos grupos de la comparación y, dentro de éstos, 292 correspondían a genes infraexpresados y 189 a genes sobreexpresados.

La visualización del resumen de resultados se repitió para el resto de estudios. La **Tabla 5** recoge la información que se extrajo de las respectivas salidas.

**Tabla 5.** Número de genes seleccionados como diferencialmente expresados (DE), infra y sobreexpresados, tras el análisis de expresión diferencial llevado a cabo con el paquete *Limma de Bioconductor*.

	DE	Sobreexpresados	Infraexpresados
GSE97239	481	189	292
GSE55492	1667	976	691
GSE92592	8702	4511	4191
GSE103001	7214	3605	3609

Si bien la obtención del número de genes de cada clasificación fue clave para la consecución del primer objetivo específico, era imprescindible almacenar por separado en nuevas variables el nombre de los genes presentes

en cada clasificación. Una vez guardado el nombre de los genes DE en un vector, los sobreexpresados en otro y los infraexpresados en un tercero y repetido este procedimiento con los otros dos métodos de *Bioconductor*, fue posible determinar los genes seleccionados en común por los diferentes paquetes tal y como se estableció en el objetivo específico 2.

Extraer los genes catalogados como DE, infraexpresados y sobreexpresados a partir de los resultados generados por la función `decideTest()` fue muy sencillo teniendo en cuenta la explicación realizada al principio de este apartado:

- **Genes DE:** genes a los que se les ha asignado el valor de 1 o -1, es decir, todos aquellos que no sean no significativos.

```
DGE_bladder_limma <- names(dt.fit[dt.fit != 0,])  
  
# Número de elementos del vector:  
length(DGE_bladder_limma)  
-----  
## [1] 481
```

- **Genes sobreexpresados:** valor asignado de 1.

```
DGE_sobre_bladder_limma <- names(dt.fit[dt.fit == 1,])  
  
# Número de elementos del vector:  
length(DGE_sobre_bladder_limma)  
-----  
## [1] 189
```

- **Genes infraexpresados:** valor asignado de -1.

```
DGE_infra_bladder_limma <- names(dt.fit[dt.fit == -1,])  
  
# Número de elementos del vector:  
length(DGE_infra_bladder_limma)  
-----  
## [1] 292
```

En los tres casos se comprobó que la longitud del vector se correspondía con el número de genes de cada clasificación.

#### 4.1.4. Análisis con *EdgeR*

Si previamente se ha realizado el análisis de expresión diferencial con *Limma-Voom*, los pasos a seguir para el análisis con el paquete *EdgeR* se simplifican considerablemente.

En *EdgeR* el análisis se realiza, al igual que en *Limma*, a partir de un objeto de la clase ‘*DGEList*’ con los datos de conteos filtrados y normalizados. Tanto la metodología como las funciones utilizadas en las fases de filtrado y normalización son comunes a ambos paquetes: el filtrado de los transcritos de baja expresión se realiza a partir de los CPM (función `cpm()`) y la normalización de los datos se lleva a cabo mediante el método TMM (función `calcNormFactors()`).

Puesto que estos pasos ya se realizaron con *Limma*, se procedió directamente al análisis de expresión génica diferencial.

##### 4.1.4.1. Estimación de las dispersiones

Como se ha explicado en el apartado 3.1.2.2, el modelo probabilístico que emplea *EdgeR* para modelar los datos de conteos es la Binomial Negativa. Por tanto, el primer paso del análisis consistió en estimar la dispersión de cada transcríto a partir de la variabilidad total para todos los genes.

El método que emplea *EdgeR* para el cálculo de la dispersión común en el caso de experimentos que se ajusten al enfoque clásico de *EdgeR* (comparaciones por pares entre grupos, experimentos de un solo factor) es el método qCML (de las siglas en inglés *quantile-adjusted Conditional Maximum Likelihood*).

El estimador qCML, comparado con otros estimadores (estimador de máxima verosimilitud, el estimador de cuasi-verosimilitud, etc), es el que mejor funciona con datos de secuenciación de NGS y el más fiable cuando se trabaja con un número bajo de muestras. Debido al coste de la secuenciación de ADN es frecuente que los experimentos de RNA-seq no cuenten con un elevado número de réplicas.

A partir del objeto *DGEList*, se procedió a la estimación en un único paso de la dispersión común y la dispersión gen a gen o *tagwise* empleando, para ello, la función `estimateDisp()`:

```
x <- estimateDisp(y, design)
# y: matriz de conteos u objeto DGEList.
# Argumento 'design': matriz de diseño.
```

Visualización de la dispersión común (estudio GSE97239):

```
## Dispersión común:  
x$common.dispersion  
  
## [1] 0.1838658
```

Una vez conocida la dispersión común, hallar el coeficiente de variación biológica (BCV) fue tan sencillo como calcular la raíz cuadrada de este parámetro. En este estudio, la dispersión común era de 0.1838658, por lo que situaba al coeficiente de variación biológica en torno a 0.43. La dispersión común así como el valor de BCV obtenido en los cuatro estudios se muestra en la **Tabla 6**.

**Tabla 6.** Dispersión común estimada con la función de *EdgeR estimateDisp()* a partir del correspondiente objeto *DGEList* de cada estudio. Debajo de cada valor de dispersión se indica el respectivo valor del coeficiente de variación biológica (BCV). El BCV es la raíz cuadrada del parámetro de dispersión bajo el modelo binomial negativo.

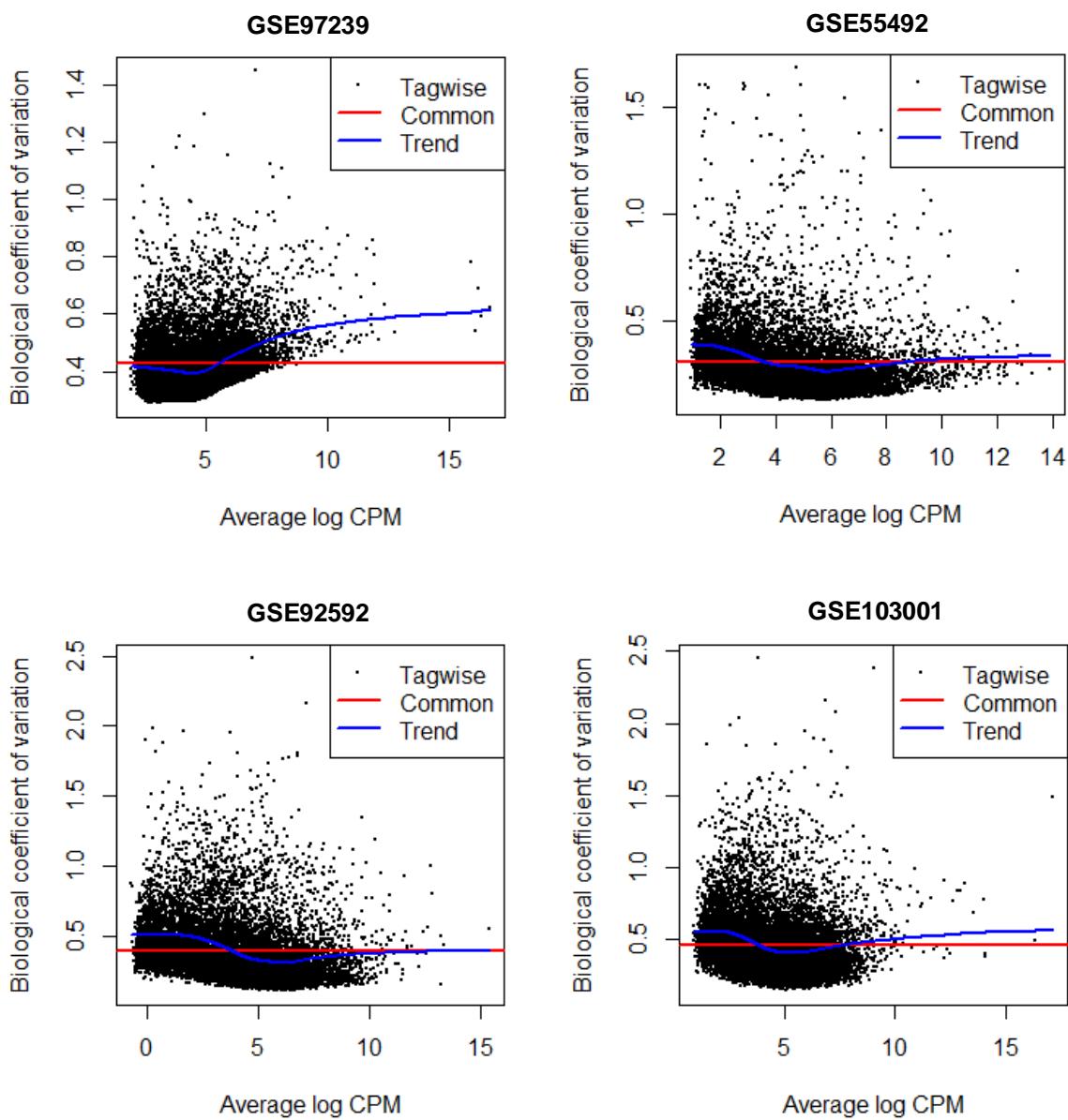
	GSE97239	GSE55492	GSE92592	GSE103001
Dispersión común	0.1838658	0.09236828	0.1489554	0.2171162
BCV	0.4287958	0.30392150	0.3859474	0.4659573

La variación biológica entre las muestras de un mismo grupo/condición es un factor que influye considerablemente en el número de genes DE. Lógicamente, cuanto mayor sea la variación entre grupos con respecto a la variación intragrupo más fácil es la detección de estos genes.

Para la visualización gráfica de las dispersiones estimadas se procedió a la generación de los gráficos BCV mediante la función *plotBCV()*:

```
plotBCV(x)  
# x: salida de la función 'estimateDisp()'.
```

Estos gráficos se obtienen al representar la raíz cuadrada de las dispersiones estimadas frente al logaritmo en base 2 de las lecturas por millón (logCPM) (**Figura 11**).



**Figura 11.** Gráficos BCV para la visualización de las dispersiones estimadas gen a gen (dispersión tagwise) a partir de la común (línea roja). Cada punto del gráfico representa un gen. La línea azul refleja la tendencia de la variación biológica a medida que aumenta el número de lecturas.

Generalmente, se considera óptimo para la detección de genes DE cuando el valor común del coeficiente de variación biológico se encuentra entre 0.2 y 0.4. Como muestra la **Figura 10**, en los gráficos correspondientes a los estudios GSE55492 y GSE92592 este valor (línea roja) se sitúa, claramente, en dicho intervalo. Sin embargo, en los otros dos estudios la línea roja sobrepasa mínimamente el umbral de 0.4 lo que podría afectar al número de genes DE que se encuentren (seleccionándose menos de los que realmente hay).

Una vez obtenidas las estimaciones de dispersión, el siguiente paso consistió en el análisis de expresión diferencial empleando, para ello, el test exacto basado en los métodos qCML. Fue en 2008 cuando Robinson y Smyth desarrollaron un test exacto apropiado para el análisis de expresión diferencial para conteos distribuidos según la Binomial Negativa (34). Resumidamente, al conocer la distribución condicional de la suma de conteos en un grupo, es posible calcular los valores de  $p$  exactos sumando todas las sumas de conteos que tienen una probabilidad inferior a la probabilidad bajo la hipótesis nula de la suma de conteos observada. Es evidente, que el test exacto para la distribución Binomial Negativa propuesto por estos dos autores tiene fuertes paralelismos con el test exacto de Fisher.

Como se ha comentado al inicio de este apartado, el método qCML y por tanto, el test exacto, sólo es aplicable a experimentos que se ajustan al enfoque clásico de *EdgeR*.

En el siguiente apartado se indicarán las funciones que permiten realizar las pruebas para encontrar diferencias entre grupos e identificar los genes más diferencialmente expresados.

#### 4.1.4.2. Análisis de expresión diferencial

La función que realiza el test exacto binomial negativo es `exactTest()`. Esta función acepta dos grupos y realiza la prueba en cada fila de datos. Por tanto, para cada fila, la prueba está condicionada a la suma de conteos de la fila en cuestión.

```
etest <- exactTest(x, pair=c("normal","tumor"))

# x: objeto de clase DGEList.

# Argumento 'pair': vector de longitud dos que especifique los dos
# grupos a comparar. Si es un vector de caracteres se debe indicar
# el nombre de los dos grupos.
```

Tras la ejecución de la instrucción anterior, se generó un objeto con todos los componentes necesarios para la aplicación de otras funciones como, por ejemplo, `topTags()` que permite la visualización de los transcritos más diferencialmente expresados, o `decideTest()` para la clasificación de los genes como no significativos, sobreexpresados o infraexpresados dependiendo del nivel de significación establecido.

La función `topTags()` es equivalente a `topTable()` de *Limma* y también ordena los genes de menor a mayor valor de  $p$ :

```

topTags(etest)

## Comparison of groups: tumor-normal
##          logFC      logCPM      PValue        FDR
## PI16    -4.622160 5.109955 3.148516e-16 4.426498e-12
## SDPR    -4.902330 4.530182 7.618600e-14 5.355495e-10
## FHL1    -5.034822 6.760935 1.245135e-13 5.835117e-10
## SCARA5  -4.779535 5.738365 7.022180e-13 2.468121e-09
## ANKAR   -3.341426 5.253878 1.065798e-12 2.996811e-09
## SPARCL1 -3.825861 5.048747 3.917118e-12 9.178460e-09
## ITGA8   -3.858112 5.886831 5.315947e-12 1.067670e-08
## PGM5    -5.324013 8.837872 6.991122e-12 1.228602e-08
## CCNE1   3.778288 3.930482 8.347005e-12 1.303895e-08
## PYCR1   3.738507 5.788973 1.033480e-11 1.452970e-08

# 'etest': salida de exactTest().
# Argumento 'n': número de genes/transcritos a mostrar. Por defecto, 10.
# Argumento 'adjust.method': método para ajustar los p-valores. Por defecto, "BH".
# Argumento 'sort.by': estadístico a considerar para ordenar la tabla. Por defecto,
# "PValue".

```

- **logFC:** logaritmo en base 2 del *Fold-change*.  
Valor logFC positivo: genes sobreexpresados.  
Valor logFC negativo: genes infraexpresados.
- **logCPM:** logaritmo en base 2 de la media del cpm.
- **PValue:** valores de  $p$  según una distribución la Binomial Negativa con los parámetros anteriormente estimados.
- **FDR:** valores de  $p$  ajustados según el método indicado. Si no se especifica, por defecto la función utiliza el método de Benjamí y Hochberg (abreviado BH) para el control de la tasa de falsos positivos o FDR (del inglés, *False Discovery Rate*). Se trata de un método algo menos conservador que la corrección de Bonferroni.

Tras la visualización de los 10 genes más diferencialmente expresados, se advirtió que algunos de ellos también aparecían en los primeros puestos en el listado que se muestra en el análisis con *Limma* (apartado 4.1.3.3.). Efectivamente, ya entre los 10 primeros genes, 5 fueron identificados por ambos paquetes:

```

## Estudio GSE 97239.

top10_Limma_EdgeR <- intersect(rownames(topTable(efit)), rownames(topTags(etest)))
# topTable(efit): 10 genes más diferencialmente expresados identificados con Limma.
# topTags(etest): 10 genes más diferencialmente expresados identificados con EdgeR.

# Número de genes identificados por ambos paquetes:
length(top10_Limma_EdgeR)

## [1] 5

# Nombre de los genes en común:
top10_Limma_EdgeR

## [1] " FHL1" " PGM5" " ANKAR" " PYCR1" " ITGA8"

```

No obstante, en apartados posteriores se procedió a una comparación más completa considerando la totalidad de los genes seleccionados con cada paquete.

### **Selección de genes DE**

Para finalizar el análisis de expresión diferencial con *EdgeR* se determinó el número de genes de cada clasificación utilizando, para ello, la función `decideTestDGE()`. Esta función es similar a `decideTests()` empleada en *Limma*.

```
dTest <- decideTestsDGE(etest)    # Nivel de significación por defecto: 0.05.
summary(dTest)
-----
##          tumor-normal
## Down           936
## NotSig        12351
## Up            772
```

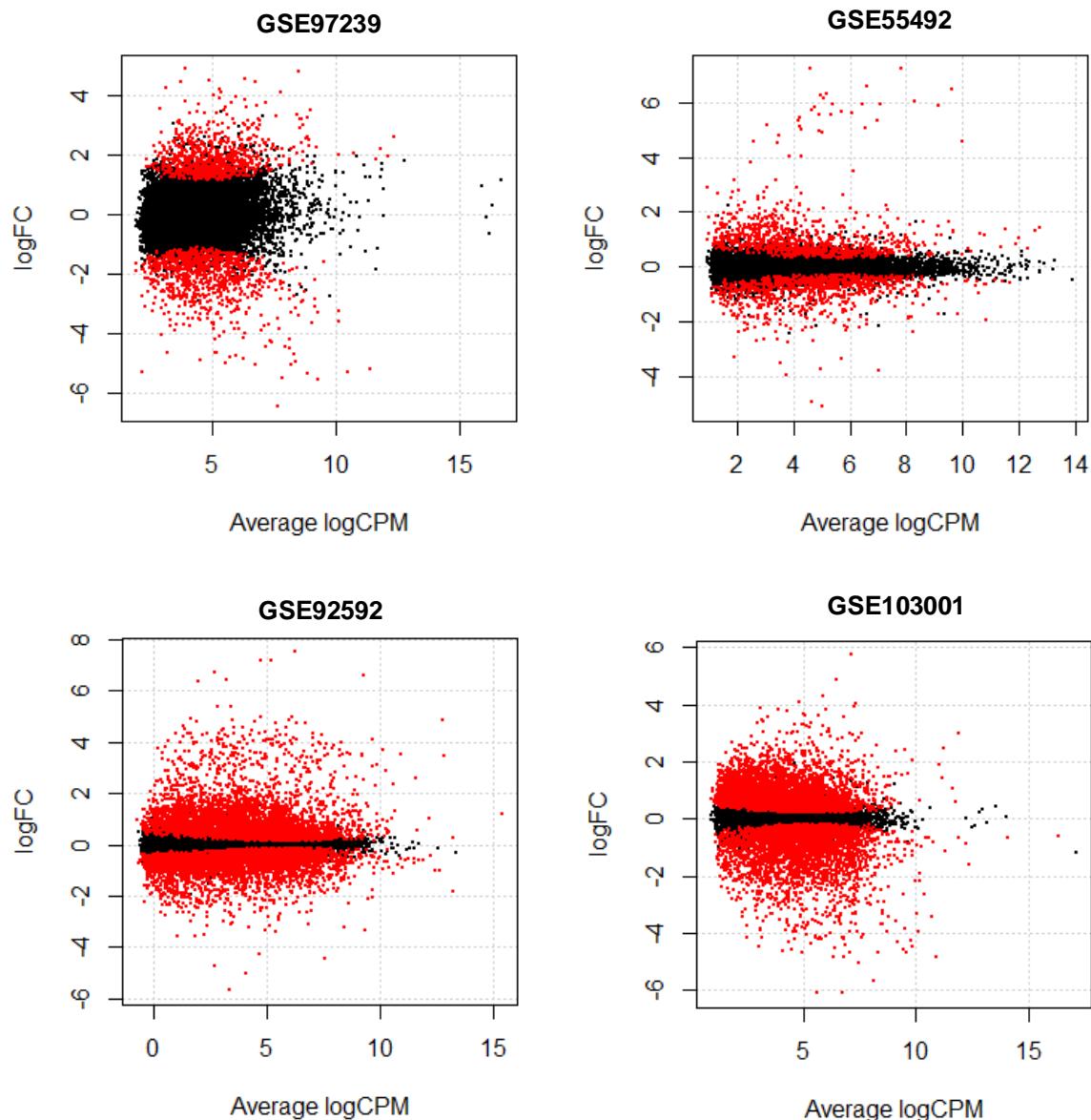
El número total de genes con un valor de FDR inferior al nivel de significación ( $\alpha = 0,05$ ) fue de 1708. De estos genes, 936 se identificaron como infrarregulados y 772 como sobreexpresados.

La visualización del resumen de resultados se repitió para los otros tres estudios. La **Tabla 7** recoge la información que se extrajo de las respectivas salidas.

**Tabla 7.** Número de genes seleccionados como diferencialmente expresados (DE), infra y sobreexpresados, tras el análisis de expresión diferencial llevado a cabo con el paquete *EdgeR* de *Bioconductor*.

	DE	Sobreexpresados	Infraexpresados
<b>GSE97239</b>	<b>1708</b>	772	936
<b>GSE55492</b>	<b>1782</b>	912	870
<b>GSE92592</b>	<b>8769</b>	4339	4430
<b>GSE103001</b>	<b>7616</b>	4155	3461

Para visualizar gráficamente los genes identificados como DE respecto a la totalidad, se decidió generar el llamado gráfico 'plotSmear' (**Figura 12**). Este tipo de gráfico se basa en la representación del logaritmo en base 2 del *fold-change* (logFC) correspondiente a cada gen frente a la media del logaritmo en base 2 de los conteos por millón.



**Figura 12.** Visualización de los resultados del análisis de expresión génica diferencial llevado a cabo en los cuatro estudios con el paquete *EdgeR* de *Bioconductor*. Los genes diferencialmente expresados se muestran resaltados en rojo.

#### 4.1.5. Análisis con *DESeq2*

El tercer paquete de *Bioconductor* en el que se centró nuestro estudio fue *DESeq2*. El modelo probabilístico que utiliza este paquete es, al igual que *EdgeR*, la Binomial Negativa. Sin embargo, cada uno emplea su propia estrategia de normalización como se explicó en el apartado 3.1. *Software y paquetes de R usados para el análisis*.

El análisis de expresión diferencial con *DESeq2* es todavía más cómodo y sencillo que con *EdgeR* ya que no es necesario realizar los diferentes pasos del análisis uno a uno. La mayoría de las funciones se han unificado y, una vez creado el objeto de la clase *DESeq-DataSet* con los datos de conteos y con la correspondiente información sobre las muestras, realiza todo el análisis a través de la función *DESeq()*. Esta función lleva a cabo de forma secuencial todos los pasos necesarios para realizar un análisis completo de expresión diferencial. Finalizado el análisis, es posible visualizar los resultados a través de la función *results()*.

#### 4.1.5.1. Obtención del objeto de la clase *DESeq-DataSet*

El objeto *DESeq-DataSet* se creó a partir de la matriz de conteos empleando, para ello, la función *DESeqDataSetFromMatrix()*.

Previamente fue necesario crear un *data frame* a partir del cuál la función pudiera extraer a través de sus diferentes argumentos información sobre las muestras del estudio, como por ejemplo, el grupo experimental de cada una de ellas.

```
## Estudio GSE 97239

# Información de las muestras:
info_bladder

##          targets
## tumor1      tumor
## tumor2      tumor
## tumor3      tumor
## control1   normal
## control2   normal
## control3   normal
class(info_bladder)

## [1] "data.frame"
```

```

## Estudio GSE 97239

ds_bladder <- DESeqDataSetFromMatrix(countData = filtrado_bladder,
                                       colData = info_bladder,
                                       design = ~targets)

# Argumento 'countData': matriz de conteos
# Argumento 'colData': data frame con la información sobre las muestras.
# Argumento 'design': filas de 'colData'. Se corresponden a las columnas de
# 'countData'.

ds_bladder

```

---

```

## class: DESeqDataSet
## dim: 14059 6
## metadata(1): version
## assays(1): counts
## rownames(14059): A1BG A2M ... ZZZ3 BP-21201H5.1
## rowData names(0):
## colnames(6): tumor1 tumor2 ... control2 control3
## colData names(1): targets

```

Una vez generado el objeto DESeq-DataSet, se procedió al análisis de expresión diferencial.

#### 4.1.5.2. Análisis de expresión diferencial

Como se ha comentado al inicio de esta sección, el paquete *DESeq2* dispone de una función que realiza de forma predeterminada todos los pasos necesarios de un análisis de expresión génica diferencial basado en la distribución Binomial Negativa. Estos pasos incluyen:

- La normalización de los datos mediante la estimación de los tamaños de muestra y los factores de normalización.
- La estimación de la dispersión.
- El ajuste de los datos a un modelo lineal generalizado (GLM) binomial negativo.
- La comprobación de la expresión diferencial de cada transcripto mediante la realización del correspondiente test paramétrico de Wald.

```

## Estudio GSE 97239

ds_bladder <- DESeq(ds_bladder)

```

---

```

## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing

```

El paso en el que se encontraba trabajando la función `DESeq()` era conocido en todo momento puesto que va proporcionado esa información por pantalla a medida que va avanzando en el análisis.

La tabla de resultados se obtuvo a través de la función `results()`:

```
## Estudio GSE 97239

res_bladder <- results(ds_bladder)
head(res_bladder)

## log2 fold change (MLE): targets tumor vs normal
## Wald test p-value: targets tumor vs normal
## DataFrame with 6 rows and 6 columns
##           baseMean    log2FoldChange      lfcSE
##           <numeric>     <numeric>     <numeric>
## A1BG     60.151818649391 -0.128752279879431 0.458928440720162
## A2M      1040.74145111347 -2.46651969809092 0.509868323546766
## A2ML1    163.371254770255  2.60583515047036 0.921977758517278
## A4GALT   67.1310230600856 -0.561722590139428 0.490116246372706
## AAAS     64.6536521554108 -0.195874435256601 0.440033378411662
## AACs     237.086677101968  0.647992836715202 0.44246579919325
##           stat          pvalue      padj
##           <numeric>     <numeric>     <numeric>
## A1BG    -0.280549794816354 0.779055727873309 0.908609237692739
## A2M     -4.83756213944262 1.31441299238609e-06 9.15053781216842e-05
## A2ML1    2.82635359302057 NA             NA
## A4GALT   -1.14610073486989 0.251753473579285 0.524927187571886
## AAAS    -0.445135403054256 0.656221905471485 0.844213765124803
## AACs     1.46450378288376 0.143056350228886 0.380257087917469
```

Como se puede apreciar, la tabla generada como salida de la función mostraba asociada la siguiente información:

- `baseMean`: media de lecturas normalizadas de cada transcripto.
- `log2foldChange`: cambio en la proporción de lecturas, en función del log2, para el diseño propuesto (grupos experimentales a testar).
- `lfcSE`: hace referencia al error estándar.
- `stat`: valor del estadístico de Wald.
- `pvalue` y `padj`: los correspondientes *p*-valores y *p*-valores ajustados, a partir de los cuales obtenemos los transcriptos diferencialmente expresados.

Sin embargo, los genes se muestran ordenados alfabéticamente, tal y como aparecen en la matriz de conteos. Para identificar los genes más diferencialmente expresados, al igual que se realizó con *Limma* y *EdgeR*, se ordenaron los genes de acuerdo al valor de *p* y se visualizaron las 10 primeras filas del listado:

```

head(res_bladder[order(res_bladder$padj),],n=10)

## log2 fold change (MLE): targets tumor vs normal
## Wald test p-value: targets tumor vs normal
## DataFrame with 10 rows and 6 columns
##           baseMean    log2FoldChange        lfcSE
##           <numeric>      <numeric>      <numeric>
## PGM5     1644.29660428714 -5.26387958921964 0.572664654654834
## FHL1     389.825841563348 -4.98458871929542 0.551722434583615
## PI16     122.968382842028 -4.59318293048356 0.544005606949473
## ATF3     611.23436240093 -4.27436199955032 0.514981260890385
## DUSP2    463.36172823239 -3.8573061575246 0.474028898160322
## SCARA5   191.489721969202 -4.74929667488053 0.587800074095566
## SYNP02   1056.87929346579 -4.9915984313141 0.629455821267665
## C7       275.453255738117 -4.80319316936351 0.614402371056993
## CFD      1010.90787284888 -4.42997578472955 0.572437201322684
## ITGA8    212.688510520033 -3.82778837050132 0.494984409419522
##           stat          pvalue        padj
##           <numeric>      <numeric>      <numeric>
## PGM5    -9.19190585001683 3.859417971813e-20 5.40048356795793e-16
## FHL1    -9.03459494638329 1.64610918660274e-19 1.15170029240661e-15
## PI16    -8.44326395134044 3.08597188128066e-17 1.43940015115868e-13
## ATF3    -8.30003404815177 1.0408156176853e-16 3.6410332345676e-13
## DUSP2   -8.13728060144555 4.04255051190185e-16 1.13134818626085e-12
## SCARA5  -8.0797823685003 6.48824595769176e-16 1.51316709476635e-12
## SYNP02  -7.93002187391244 2.19107289752086e-15 4.37995472214419e-12
## C7      -7.8176670462718 5.38112948610035e-15 9.41226811237528e-12
## CFD     -7.73879785327294 1.00361200024377e-14 1.46811865137325e-11
## ITGA8   -7.73314936320973 1.04918076993729e-14 1.46811865137325e-11

## Estudio GSE 97239

```

Como era de esperar, la mayoría de los genes que aparecieron en los primeros puestos del listado coincidían con los identificados por los otros dos paquetes. Concretamente, se determinó que siete de ellos también se encontraban entre los diez genes más diferencialmente expresados seleccionados bien por *Limma* o bien por *EdgeR*.

### **Selección de genes DE**

Para conocer el número de genes que se expresaban diferencialmente entre los dos grupos experimentales, se recurrió de nuevo a la función `summary()`. En este caso fue necesario especificar el nivel de significación ya que el paquete *DESeq2* emplea, por defecto, un valor de  $\alpha = 0.01$ . Lógicamente, para poder comparar resultados, el nivel de significación tiene que ser el mismo para los tres paquetes.

```

## Estudio GSE 97239

summary(res_bladder, alpha = 0.05)
-----
## 
## out of 14059 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 937, 6.7%
## LFC < 0 (down)    : 887, 6.3%
## outliers [1]       : 66, 0.47%
## low counts [2]     : 0, 0%
## (mean count < 12)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

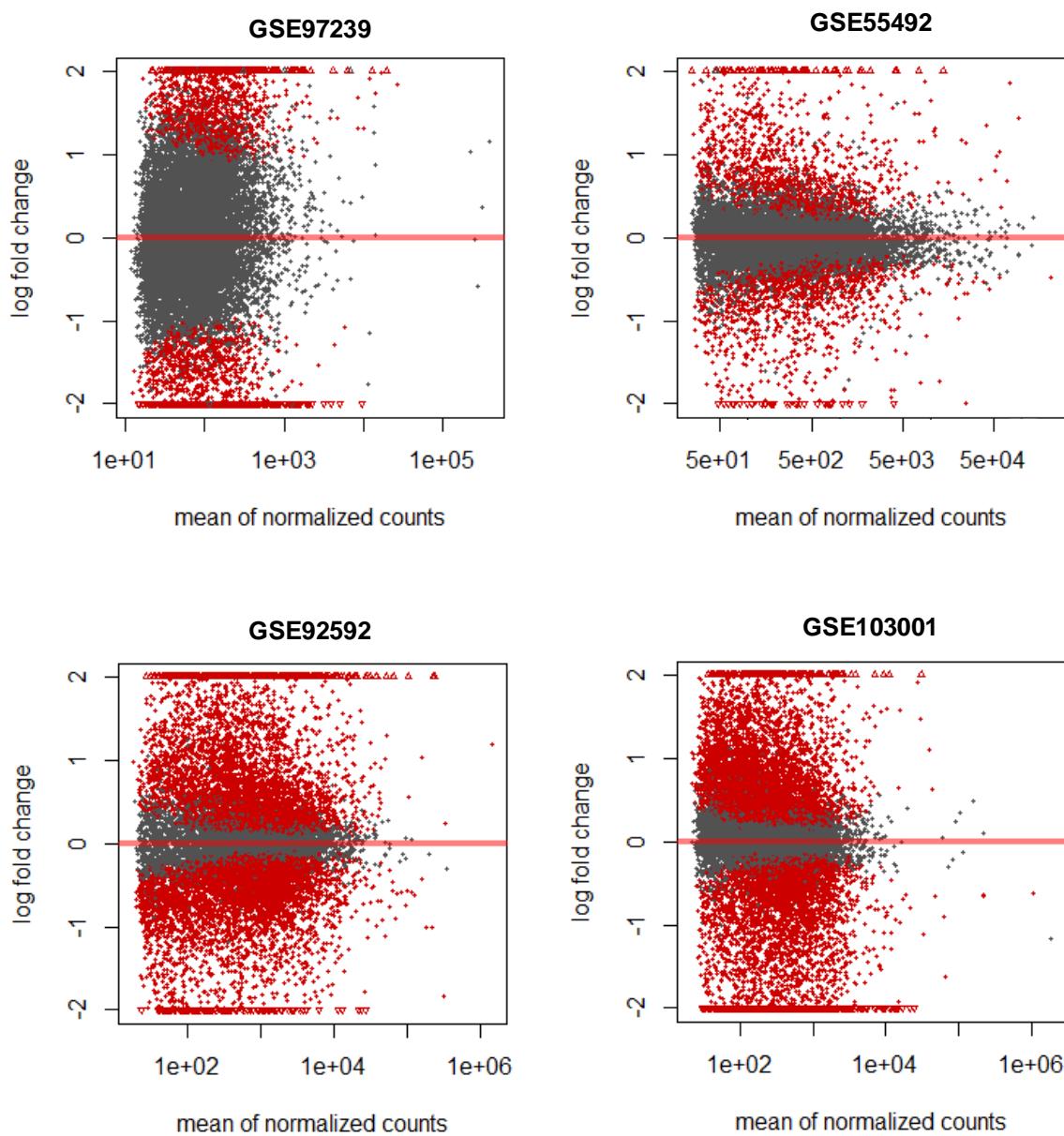
La salida de esta función proporcionó el número de genes que presentaban un valor de logFC significativamente negativo (887 genes) o significativamente positivo (937 genes) a un nivel de significación de 0,05.

El procedimiento descrito para GSE97239 se repitió para el resto de estudios determinándose, en cada uno de ellos, el número de genes con un valor de  $p$  ajustado inferior al nivel de significación establecido y con un logFC positivo o negativo. Dicha información se resume en la **Tabla 8**.

**Tabla 8.** Número de genes seleccionados como diferencialmente expresados (DE), infra y sobreexpresados, tras el análisis de expresión diferencial llevado a cabo con el paquete *DESeq2* de *Bioconductor*.

	DE	Sobreexpresados	Infraexpresados
<b>GSE97239</b>	<b>1824</b>	937	887
<b>GSE55492</b>	<b>2025</b>	1016	1009
<b>GSE92592</b>	<b>9061</b>	4402	4659
<b>GSE103001</b>	<b>7966</b>	4491	3475

Para finalizar, los resultados del análisis de expresión diferencial bajo *DESeq2* se visualizaron mediante gráficos MA-Plot. Este tipo de gráficos se genera al representar la media de lecturas normalizadas de cada gen frente al logaritmo en base 2 del *fold-change* (**Figura 13**).



**Figura 13.** Visualización de los resultados del análisis de expresión génica diferencial realizado en los cuatro estudios con el paquete *DESeq2* de *Bioconductor*. Los puntos correspondientes a genes identificados como diferencialmente expresados (valor de  $p$  ajustado inferior a 0,05) se destacan en color rojo. Los puntos que caen fuera de la ventana se trazan como triángulos abiertos que apuntan hacia arriba o hacia abajo dependiendo de si el valor de logFC es superior a 2 o inferior a -2, respectivamente.

Al comparar estos gráficos con los obtenidos con la función `plotSmear()` de *EdgeR* (**Figura 12**), se advirtió un notable parecido entre ellos. Efectivamente, aunque en *EdgeR* no existe una función como en *DESeq2* para la creación de gráficos MA, la función `plotSmear()` permite obtener un

gráfico muy parecido al representar el logaritmo de las medias de los conteos por millón en relación al *log2fold-change*.

Una vez finalizada la primera parte del trabajo se obtuvo, fruto del análisis de expresión génica diferencial llevado a cabo en los cuatro conjuntos de datos reales de *RNA-Seq*, un listado de genes DE por cada uno de los paquetes de *Bioconductor* utilizado para dicho análisis. Estos listados catalogaban los genes como sobreexpresados o infraexpresados dependiendo del nivel de expresión en comparación con el correspondiente control. En cada estudio, se determinó el número de genes de cada categoría y se recopiló dicha información en forma de tablas al finalizar la sección dedicada a cada paquete (**Tablas 5, 7 y 8**)

A partir de este punto se procedió a la realización de las tareas establecidas en la segunda parte del trabajo para la consecución del segundo y tercer objetivo específico.

## 4.2. Comparación de los resultados de expresión diferencial obtenidos con los diferentes paquetes

En este apartado se abordó el segundo objetivo específico que consistió en la comparación de los genes identificados como diferencialmente expresados por los tres paquetes de *Bioconductor* elegidos para este trabajo por su popularidad y metodología de análisis diferente.

La comparación se centró en la determinación del número de genes DE en común entre dos paquetes (*Limma-Voom/EdgeR*, *EdgeR/DESeq2* y *Limma-Voom/DESeq2*) y comunes a los tres paquetes. Las diferentes comparaciones se pudieron realizar gracias a que se almacenó, una vez finalizado el análisis, el nombre de los genes de cada categoría (DE, sobreexpresados o infraexpresados) en vectores distintos.

La función de R que permitió extraer los genes coincidentes entre los diferentes listados fue `intersect()`. A continuación se indica el código que se empleó para la obtención de esta información en el caso del estudio GSE97239 que se tomó como ejemplo a lo largo de todo el trabajo para mostrar los comandos utilizados en cada paso.

### - Genes en común entre *Limma-Voom* y *EdgeR*:

```
# Diferencialmente expresados:  
DGE_Limma_EdgeR_bladder <- intersect(DGE_bladder_limma, DGE_bladder_edgeR)  
length(DGE_Limma_EdgeR_bladder)  
## [1] 480  
  
# Sobreexpresados:  
DGE_sobre_Limma_EdgeR_bladder <- intersect(DGE_sobre_bladder_limma, DGE_sobre_bladder_edgeR)  
length(DGE_sobre_Limma_EdgeR_bladder)  
## [1] 188  
  
# Infraexpresados:  
DGE_infra_Limma_EdgeR_bladder <- intersect(DGE_infra_bladder_limma, DGE_infra_bladder_edgeR)  
length(DGE_infra_Limma_EdgeR_bladder)  
## [1] 292
```

### - Genes en común entre *Limma-Voom* y *DESeq2*:

```
# Diferencialmente expresados:  
DGE_Limma_DESeq2_bladder <- intersect(DGE_bladder_limma, DGE_bladder_DESeq2)  
length(DGE_Limma_DESeq2_bladder)  
## [1] 478  
  
# Sobreexpresados:  
DGE_sobre_Limma_DESeq2_bladder <- intersect(DGE_sobre_bladder_limma, DGE_sobre_bladder_DESeq2)  
length(DGE_sobre_Limma_DESeq2_bladder)  
## [1] 187
```

```

# Infraexpresados:
DGE_intra_Limma_DESeq2_bladder <- intersect(DGE_intra_bladder_limma, DGE_intra_bladder_DESeq2)
length(DGE_intra_Limma_DESeq2_bladder)
## [1] 291

```

- Genes en común entre **EdgeR** y **DESeq2**:

```

# Diferencialmente expresados:
DGE_EdgeR_DESeq2_bladder <- intersect(DGE_bladder_edgeR, DGE_bladder_DESeq2)
length(DGE_EdgeR_DESeq2_bladder)
## [1] 1611

```

```

# Sobreexpresados:
DGE_sobre_EdgeR_DESeq2_bladder <- intersect(DGE_sobre_bladder_edgeR, DGE_sobre_bladder_DESeq2)
length(DGE_sobre_EdgeR_DESeq2_bladder)
## [1] 758

```

```

# Infraexpresados:
DGE_intra_EdgeR_DESeq2_bladder <- intersect(DGE_intra_bladder_edgeR, DGE_intra_bladder_DESeq2)
length(DGE_intra_EdgeR_DESeq2_bladder)
## [1] 853

```

- Genes comunes a los tres paquetes:

```

# Diferencialmente expresados:
DGE_all_bladder <- intersect(intersect(DGE_bladder_limma, DGE_bladder_edgeR), DGE_bladder_DESeq2)
length(DGE_all_bladder)
## [1] 478

```

```

# Sobreexpresados:
DGE_sobre_all_bladder <- intersect(intersect(DGE_sobre_bladder_limma, DGE_sobre_bladder_edgeR),
                                         DGE_sobre_bladder_DESeq2)
length(DGE_sobre_all_bladder)
## [1] 187

```

```

# Infraexpresados:
DGE_intra_all_bladder <- intersect(intersect(DGE_intra_bladder_limma, DGE_intra_bladder_edgeR),
                                         DGE_intra_bladder_DESeq2)
length(DGE_intra_all_bladder)
## [1] 291

```

Este procedimiento se repitió con el resto de estudios. La **Tabla 9** recopila la información que se extrajo de cada una de las salidas. Para facilitar la comparación de los genes en común respecto al total de genes seleccionados por cada paquete se muestra, inmediatamente a continuación de esta tabla, el número de genes DE, sobreexpresados e infraexpresados identificados en cada estudio bajo cada uno de los tres paquetes (**Tabla 10**). Esta última tabla se creó a partir de la información de las **Tablas 5, 7 y 8** con el objetivo de poder visualizar, conjuntamente, los resultados de todos los análisis realizados.

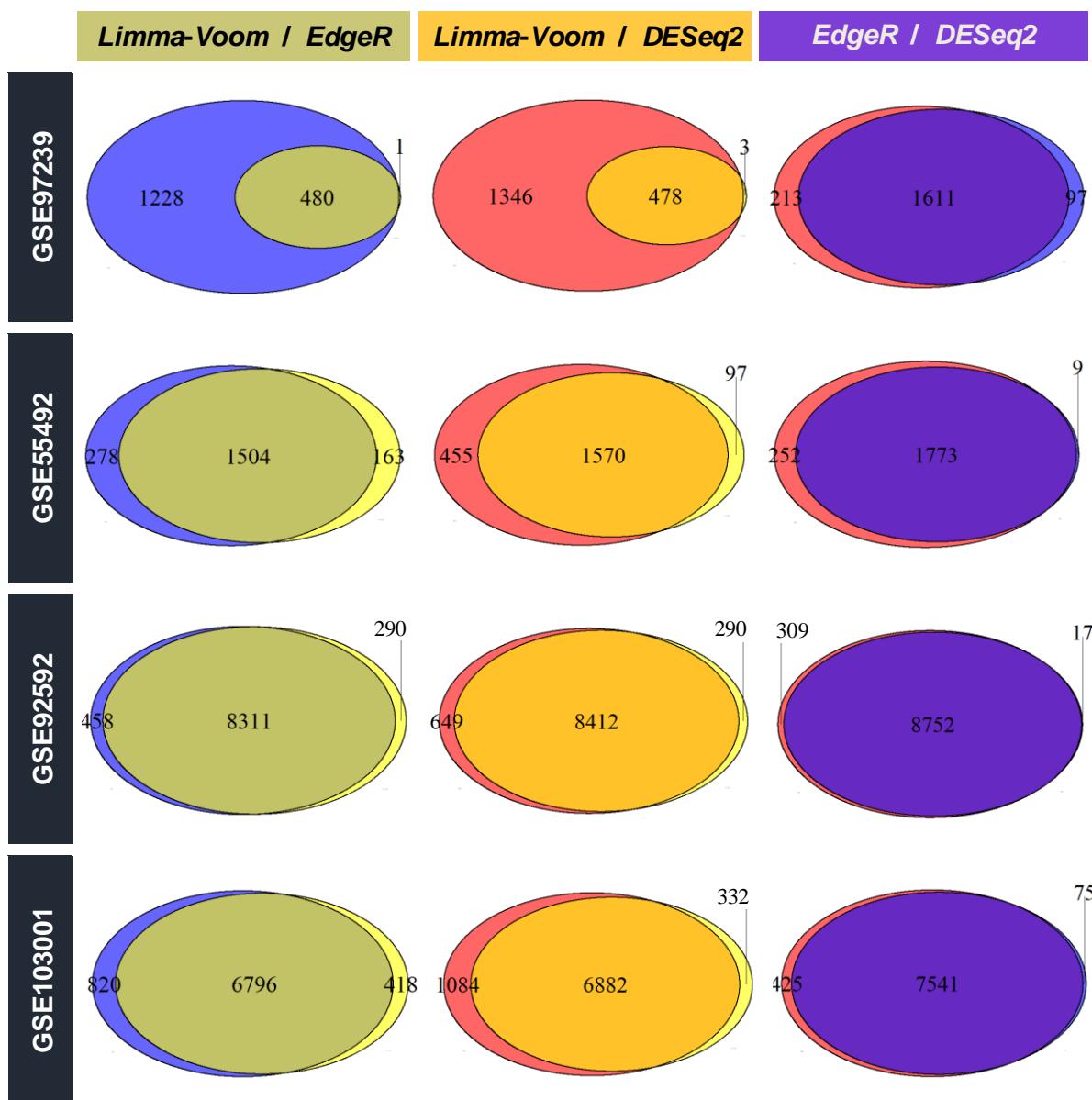
Tabla 9. Número de genes diferencialmente expresados (DE), sobreexpresados e infraexpresados en común entre los diferentes paquetes.

GSE97239			GSE55492			GSE92592			GSE103001			
	DE	Sobreexp.	Infraexp.									
<i>Limma-Voom/EdgeR</i>	480	188	292	1504	854	650	8311	4211	4100	6796	3455	3341
<i>Limma-Voom/DESeq2</i>	478	187	291	1570	892	678	8412	4269	4143	6882	3527	3355
<i>EdgeR/DESeq2</i>	1611	758	853	1773	909	864	8752	4327	4425	7541	4117	3424
<i>Limma-Voom/EdgeR/DESeq2</i>	478	187	291	1500	852	648	8305	4206	4099	6773	3454	3319

Tabla 10. Número de genes diferencialmente expresados (DE), sobreexpresados e infraexpresados obtenidos tras el análisis con los tres paquetes de *Bioconductor* más frecuentemente empleados para el análisis de expresión génica diferencial.

GSE97239			GSE55492			GSE92592			GSE103001			
Filtrados: 14059			Filtrados: 13406			Filtrados: 16635			Filtrados: 15166			
	DE	Sobreexp.	Infraexp.	DE	Sobreexp.	Infraexp.	DE	Sobreexp.	Infraexp.	DE	Sobreexp.	Infraexp.
<i>Limma-Voom</i>	481	189	292	1667	976	691	8702	4511	4191	7214	3605	3609
<i>EdgeR</i>	1708	772	936	1782	912	870	8769	4339	4430	7616	4155	3461
<i>DESeq2</i>	1824	937	887	2025	1016	1009	9061	4402	4659	7966	4491	3475

Aunque un simple vistazo a las tablas anteriores fue suficiente para percatarse de que un elevado número de genes DE fueron seleccionados por dos o incluso los tres paquetes, resultó especialmente útil la representación gráfica de las relaciones entre los conjuntos de genes mediante la generación de los correspondientes diagramas de Venn (**Figura 14**).



**Figura 14.** Diagramas de Venn para la visualización de los genes diferencialmente expresados comunes a dos paquetes. Se muestran las tres combinaciones posibles: *Limma-Voom - EdgeR*, *Limma-Voom - DESeq2* y *EdgeR - DESeq2*. Conjunto de genes de *Limma-Voom* representado en amarillo, *EdgeR* en azul y *DESeq2* en rojo. Las zonas de superposición de los óvalos representan genes comunes a los dos paquetes considerados. Cuanto mayor sea la zona de solapamiento (intersección) entre dos conjuntos mayor es el número de genes en común.

Efectivamente, en los cuatro estudios se observó que el área conjunta que representaba genes hallados paralelamente en dos listados, suponía la mayor parte del área del diagrama, lo que permitió comprobar que los tres paquetes eran capaces de detectar correctamente genes diferencialmente expresados entre los grupos experimentales de interés.

Sin embargo, a pesar de que un importante porcentaje de genes DE eran los mismos, se apreció una clara tendencia que se fue confirmando a medida que se completaba el análisis con las tres metodologías en los diferentes estudios: *Limma - Voom* era el paquete con el que se obtenía un menor número de genes DE y *DESeq2*, por el contrario, con el que obtenía un mayor número. *EdgeR* ocupaba claramente una posición intermedia entre ambos paquetes.

Como se discute en el siguiente apartado, *Limma-Voom* ofrece un enfoque más conservador y *DESeq2* un enfoque más arriesgado.

Lógicamente, el paquete que toma un enfoque más arriesgado deberá incluir la mayoría de los genes identificados como DE por el paquete menos permisivo.

Efectivamente, aunque los diagramas de Venn (**Figura 14**) no dejaron lugar a duda, se calculó el porcentaje de genes del paquete más conservador que fueron declarados también como DE por los paquetes con mayor grado de permisividad (**Tabla 11**).

**Tabla 11. Porcentaje de genes DE incluidos en los paquetes con enfoque más permisivo.**

	Genes DE seleccionados por <i>Limma-Voom</i> incluidos en <i>EdgeR</i>	Genes DE seleccionados por <i>Limma-Voom</i> incluidos en <i>DESeq2</i>	Genes DE seleccionados por <i>EdgeR</i> incluidos en <i>DESeq2</i>
<b>GSE97239</b>	<b>99,80 %</b>	<b>99,38 %</b>	<b>94,32 %</b>
<b>GSE55492</b>	<b>90,22 %</b>	<b>94,17 %</b>	<b>99,50 %</b>
<b>GSE92592</b>	<b>96,63 %</b>	<b>96,67 %</b>	<b>99,81 %</b>
<b>GSE103001</b>	<b>94,21 %</b>	<b>95,40 %</b>	<b>99,02 %</b>

Los resultados de la tabla anterior junto con los diagramas de Venn confirmaron que un altísimo porcentaje de los genes DE seleccionados por *Limma-Voom* también fueron identificados como DE por los otros dos paquetes con un enfoque algo menos conservador.

#### **4.3. Análisis crítico de los resultados e identificación de las ventajas e inconvenientes implícitas a cada modelo**

Como se ha descrito en la primera parte del trabajo, el análisis de expresión génica diferencial llevado a cabo con conjuntos de datos reales permitió identificar, en cada estudio, los genes cuya expresión era significativamente diferente entre los dos grupos de la comparación.

Sin embargo, aunque una buena parte estos genes fueron comunes a los tres paquetes de *Bioconductor*, el número de genes seleccionados por cada uno de ellos difirió ligeramente como era de esperar. Como se muestra en la **Tabla 1** de la sección 3.1. *Software y paquetes de R usados para el análisis*, existen importantes elementos distintivos entre los tres paquetes empleados para el análisis. Estas diferencias se encuentran desde el primer punto clave del análisis que implica la estrategia de normalización empleada, hasta el último paso que hace referencia a los correspondientes test utilizados para el análisis de la expresión diferencial; sin olvidar, lógicamente, la principal diferencia: el modelo probabilístico.

Precisamente, la modelización estadística es el punto que genera más debate entre los científicos que trabajan en NGS y más concretamente con datos de *RNA-Seq*. Esto se pone de manifiesto por la cantidad de artículos científicos que tratan este aspecto y los numerosos paquetes de R/*Bioconductor* basados en diferentes modelos probabilísticos que se han desarrollado para el análisis de este tipo de datos, entre ellos: *EdgeR* (5) y *DESeq2* (7) (Binomial Negativa), *baySeq* (35) y *EBSeq* (36) (enfoques Bayesianos basados en Binomial Negativa), *NO/Seq* (37) (enfoque no paramétrico basado en distribuciones empíricas), *SAMSeq* (38) (enfoque no paramétrico basado en el estadístico de rangos de Wilcoxon), *DEGSeq* (39) (distribución de Poisson), *Limma* (6) (Modelo Lineal), etc.

Entre los diferentes enfoques que se acaban de citar, suscita mucho interés la distribución Binomial Negativa y los modelos lineales. Por este motivo, se decidió comparar los paquetes más populares que actualmente permiten el análisis de datos basados en dichos enfoques: *EdgeR* y *DESeq2* para la Binomial Negativa y *Limma* en el caso de modelos lineales.

De acuerdo a los resultados obtenidos, una diferencia clara entre los paquetes analizados es el nivel de permisividad a la hora de decidir si un gen debe considerarse DE o no. Los dos paquetes más permisivos, es decir, con los que se obtuvo un mayor número de genes DE, resultaron ser aquellos basados en la distribución Binomial Negativa: *EdgeR* y *DESeq2*. Efectivamente, los métodos basados en este tipo de distribución, al tratar las dispersiones estimadas como si fueran parámetros conocidos (sin tener en consideración la incertidumbre de la estimación), conduce a test estadísticos más arriesgados (40).

Por otra parte, aunque se apreció que *DESeq2* era algo más permisivo que *EdgeR*, ofreció resultados muy parecidos en los cuatro estudios

independientemente del número de muestras analizadas. Como muestra la **Tabla 11**, el número de genes identificados como DE por ambos paquetes fue muy similar tanto en el estudio con el menor número de muestras (GSE97239: 3 casos y 3 controles) como en el estudio con el mayor número de muestras/réplicas analizadas por grupo experimental (GSE103001: 21 casos y 19 controles). Además, prácticamente la totalidad de los genes seleccionados por *DESeq2* incluía a los de *EdgeR*. Los porcentajes que aparecen en la última columna de la **Tabla 11**, dan buena muestra de ello.

Por tanto, los resultados derivados de este trabajo indican, claramente, que *EdgeR* y *DESeq2*, a pesar de utilizar algoritmos diferentes para la estimación de la dispersión, ofrecen resultados muy similares. Dado que esta observación se confirma con trabajos de otros autores (40, 41), cualquiera de los dos paquetes se podría utilizar indistintamente siempre que no exista alguna otra limitación o particularidad del experimento que lo desaconseje.

Para el usuario y, sobre todo para aquellos menos experimentados, un punto importante que marcará la diferencia entre trabajar con un paquete u otro será la facilidad de uso lo cuál está directamente relacionado con la cantidad de pasos a seguir hasta completar el análisis de expresión diferencial. Debido a la rápida y creciente irrupción de las tecnologías de secuenciación masiva en la mayoría de los laboratorios de biología molecular, es frecuente encontrar usuarios con poca experiencia en el análisis de este tipo de datos, por lo que es fundamental proporcionales herramientas de análisis cómodas de manejar capaz de detectar rigurosamente genes DE.

Efectivamente, una clara ventaja de *DESeq2* es la facilidad con la que se trabaja en este paquete. Como se comentó en el apartado 4.1.5. *Análisis con DESeq2*, una única función realiza todos los pasos necesarios para el análisis: desde la estimación de las dispersiones, hasta el ajuste del modelo y la realización de los test para la determinación de expresión diferencial. Sin embargo, en *EdgeR* el usuario debe emplear más comandos ya que debe realizar esos pasos uno a uno (**Figura 15**). De forma similar ocurre con *Limma* aunque con un considerable mayor número de pasos (**Figura 16**).

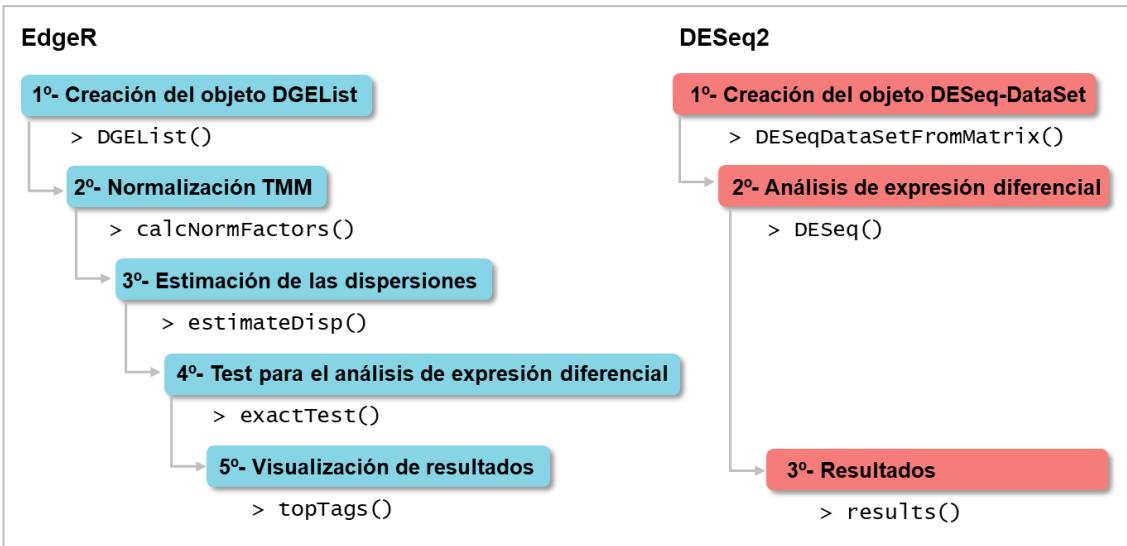


Figura 15. Comparación de los pasos a seguir y las funciones a ejecutar para completar un análisis de expresión diferencial con los dos paquetes de *Bioconductor* empleados en este trabajo que se basan en la Binomial Negativa: *EdgeR* y *DESeq2*. Nótese la simplicidad del análisis con *DESeq2*.

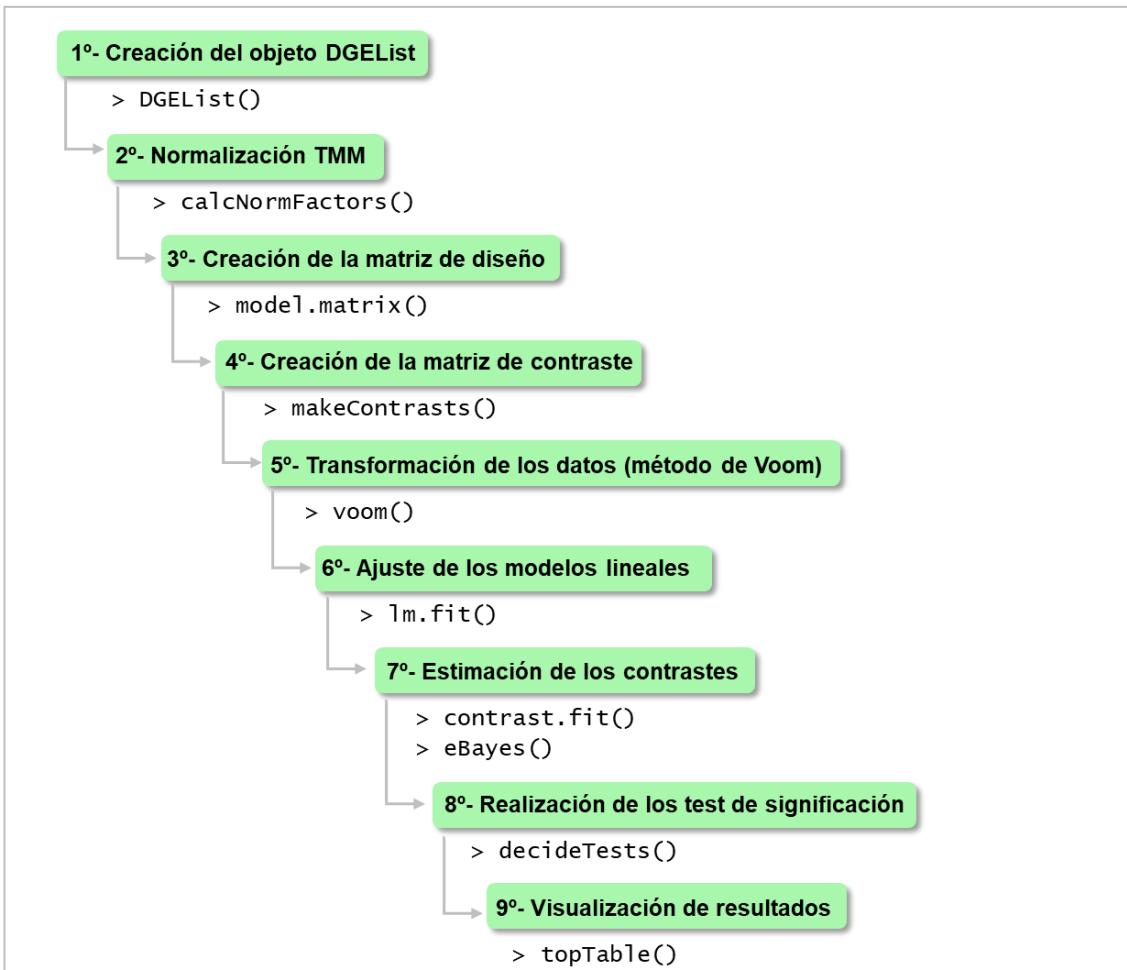
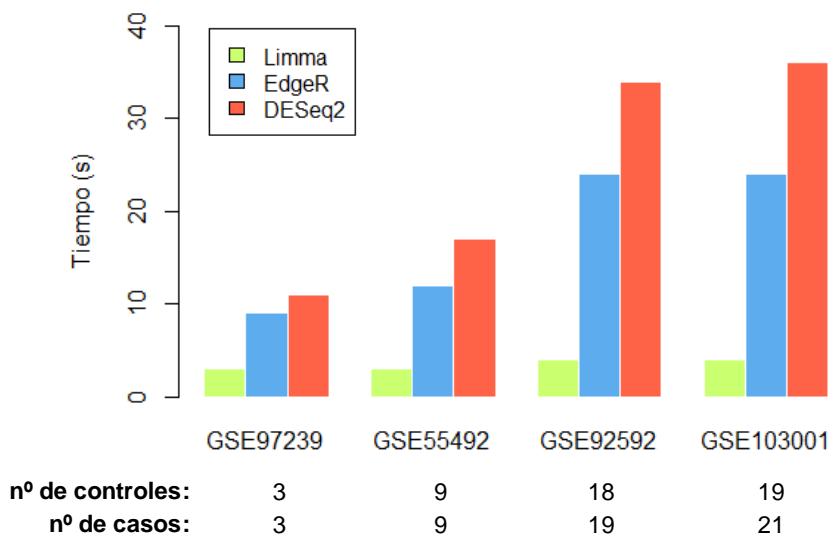


Figura 16. Pasos a seguir y las funciones a ejecutar para completar un análisis de expresión diferencial con *Limma* (con transformación *Voom*).

A parte de la facilidad de uso, otro de los parámetros de aplicabilidad que se comparó fue la velocidad de cómputo. Como muestra la **Figura 17**, aunque el paquete *DESeq2* facilita enormemente el trabajo al usuario al poder realizar el análisis empleando tan sólo un reducido número de comandos, el tiempo de cómputo es mayor en comparación con *EdgeR* aunque ambos emplean el mismo modelo probabilístico. Efectivamente, tal y como apuntan otros autores el tiempo de computación es medio/alto en *DESeq2* (33, 40) y medio en *EdgeR* (42).

Respecto al paquete basado en modelos lineales, el tiempo de cómputo es considerablemente menor en comparación con *EdgeR* y *DESeq2* y, a diferencia de estos dos últimos, no se ve afectado por el número de muestras del experimento. Esta observación coincide con la de otros autores que confirman que *Limma-Voom* es computacionalmente rápido independientemente del tamaño de muestra (33, 40, 43).



**Figura 17.** Comparación del tiempo de cómputo de los tres paquetes de *Bioconductor* utilizados para el análisis de expresión génica diferencial. Nótese el incremento de tiempo en *EdgeR* y *DESeq2* al aumentar el número de muestras del estudio.

Por tanto, el paquete más sencillo también es el más lento y éste es un aspecto que debe valorar el usuario y más aún cuando el número de muestras sea elevado.

De acuerdo a los resultados obtenidos en este trabajo, los dos paquetes basados en la Binomial Negativa ofrecen resultados muy similares con independencia del número de muestras por condición. Sin embargo, esto no es aplicable al paquete basado en modelos lineales. Claramente, el número de genes identificados como DE en el caso del estudio GSE97239 (tan sólo 3 réplicas/grupo) fue considerablemente inferior en *Limma-Voom* que bajo el análisis con *EdgeR* y *DESeq2* (**Tabla 10**). En efecto, se ha descrito que el

paquete basado en modelos lineales tiene bajo poder estadístico para muestras pequeñas (40).

Por tanto, el mejor método para analizar los datos dependerá, en buena parte, de las características propias del experimento siendo clave el número de réplicas del experimento, es decir, del número de muestras en cada grupo experimental o condición.

En la **Tabla 12** se resume el presente estudio por medio de las principales observaciones y características para cada uno de los métodos evaluados.

**Tabla 12. Resumen de las principales diferencias observadas en el presente estudio entre los tres paquetes de *Bioconductor* evaluados.**

	<i>Limma-Voom</i>	<i>EdgeR</i>	<i>DESeq2</i>
<b>Facilidad de uso</b>	+	+++	++++
<b>Tiempo de cómputo</b>	Muy bajo e independiente del número de muestras	Medio. Aumenta con el tamaño de muestra.	Alto. Aumenta con el tamaño de muestra.
<b>Enfoque conservador</b>	<i>Limma-Voom &gt; EdgeR &gt; DESeq2</i>		
<b>Poder estadístico</b>	Bajo para pocas muestras/réplicas. Alto para más muestras.	Alto	Alto

## 5. Conclusiones del estudio

- *Limma*, *EdgeR* y *DESeq2* permiten llevar a cabo el análisis de datos de RNA-Seq para estudios de expresión génica diferencial ofreciendo resultados similares excepto cuando el número de réplicas/muestras por grupo o condición experimental es muy bajo. Para muestras muy pequeñas, *Limma* demuestra tener bajo poder estadístico.
- Los dos paquetes con el enfoque más permisivo son los basados en la Binomial Negativa: *EdgeR* y *DESeq2*. Por el contrario, *Limma* presenta un enfoque más conservador obteniéndose un menor número de genes DE en comparación con los otros dos paquetes.
- *DESeq2* permite realizar un análisis de expresión diferencial completo de manera cómoda y sencilla con tan sólo tres pasos, sin embargo, necesita mayor tiempo de cómputo y éste se incrementa considerablemente con el número de muestras.
- El análisis con *Limma* requiere seguir un elevado número de pasos y ejecutar al menos diez funciones distintas, sin embargo, computacionalmente es muy rápido independientemente del número de muestras.
- *EdgeR* ocupa una posición intermedia entre *Limma* y *DESeq2* en cuanto a grado de permisividad, facilidad de uso y tiempo de cómputo. Este último parámetro depende, al igual que en *DESeq2*, del número de muestras a analizar.

De las conclusiones anteriores se extrae que la elección del mejor paquete para el análisis de los datos de un experimento de RNA-Seq estará determinado, en gran medida, por el número de muestras del experimento.

Aparte de lo anteriormente expuesto, simplemente comentar que, personalmente, este trabajo me ha brindado la oportunidad de reforzar las habilidades en el entorno R y, utilizando datos de experimentos reales, poder conocer de cerca el proceso de análisis de este tipo de datos generado por una tecnología cada vez más extendida y presente en los laboratorios de investigación. Adentrarme en este proyecto me ha aportado una mayor destreza y conocimiento sobre las diferentes herramientas así como una mayor capacidad de decisión sobre la mejor metodología de análisis de acuerdo a las características del experimento.

## **Cumplimiento de los objetivos y seguimiento de la planificación y metodología.**

Todos los objetivos que se plantearon al inicio del proyecto se han cumplido con éxito y la planificación de las tareas establecidas en el plan de trabajo se ha seguido rigurosamente en todo momento.

## **Líneas de trabajo futuro**

Puesto que la metodología de análisis de datos de *RNA-Seq* no es un tema cerrado y, a día de hoy, sigue existiendo un intenso debate y, sobre todo, en torno al modelo probabilístico a usar para la modelización de este tipo de datos, periódicamente surgen nuevos trabajos así como nuevos paquetes de R fruto del esfuerzo de encontrar la mejor forma de análisis y de superar la reducida gama de herramientas estadísticas asociadas con las distribuciones de conteos que actualmente existen en comparación con la distribución normal.

El presente trabajo tiene una clara continuidad que implicará:

- Comparar las nuevas versiones que se publiquen de los paquetes evaluados en este trabajo.
- Incluir en el estudio nuevos paquetes basados en otros modelos probabilísticos.
- Revisar periódicamente los resultados y conclusiones de otros autores.

## 6. Glosario

**Análisis DGE:** acrónimo del inglés *Differential Gene Expression* (en español, análisis de expresión génica diferencial). Es una de las aplicaciones más comunes de la tecnología RNA-Seq. Permite la elucidación de genes diferencialmente expresados entre dos o más condiciones.

**BCV:** acrónimo del inglés *Biological Coefficient of Variation* (en español, coeficiente de variación biológica). Se calcula como la raíz cuadrada del parámetro de dispersión bajo el modelo binomial negativo.

**Biobase (software package):** funciones básicas de *Bioconductor* que son requeridas por muchos otros paquetes o que reemplazan ciertas funciones de R.

**Bioconductor:** es un proyecto de código abierto y gratuito basado en el lenguaje de programación R que proporciona herramientas para el análisis y la comprensión de datos genómicos de alto rendimiento. Tiene dos lanzamientos cada año y una comunidad de usuarios activa.

**Boxplot:** diagrama de caja.

**CPM:** acrónimo del inglés *Counts Per Million* (en español, conteos por millón).

**Data frame:** marco de datos. Clase de objeto especial en R. Se trata de estructuras de datos de dos dimensiones que pueden, a diferencia de las matrices, contener datos de diferentes tipos.

**DE:** gen diferencialmente expresado. Gen cuyo nivel de expresión es significativamente diferente entre dos o más condiciones experimentales.

**FDR.** Acrónimo del inglés *False Discovery Rate* (en español, tasa de falsos descubrimientos).

**GLM:** acrónimo del inglés *Generalized Linear Model* (en español, modelo linear generalizado).

**GNU:** acrónimo recursivo de "GNU's Not Unix" (GNU no es Unix). Es un sistema operativo caracterizado por estar compuesto completamente por software libre.

**Heatmap:** mapa de calor de representación de datos.

**IDE:** acrónimo del inglés *Integrated Development Environment* (en español, Entorno de Desarrollo Integrado). Hace referencia al entorno de programación que ha sido empaquetado como un programa de aplicación y proporciona al usuario un marco de trabajo amigable para los lenguajes de programación.

**Interfaz** (interfaz de usuario): es el medio con que el usuario puede comunicarse con el ordenador.

**MDS**: acrónimo del inglés *Multidimensional Scaling* (en español, escalamiento multidimensional).

**NGS**. Acrónimo del inglés *Next Generation Sequencing*. Tecnología de secuenciación de gran potencia que ha revolucionado el mundo de la genómica y la biología molecular. Se trata de un método de alto rendimiento que utiliza procesos paralelos masivos para secuenciar ADN o ARN de manera más rápida y rentable que las tecnologías anteriores de secuenciación.

**Paquete de R**: es una colección de código R, funciones y datos que se almacenan en una carpeta conforme a una estructura bien definida y fácilmente accesible para R. Estos paquetes se distribuyen listos para su utilización por otros usuarios.

**Pipeline**. Conjunto de técnicas o procesos ejecutados en secuencia.

**qCML**: acrónimo del inglés *quantile-adjusted Conditional Maximum Likelihood*.

**RNA-Seq o secuenciación de RNA**: es una tecnología que utiliza la secuenciación masiva (NGS) para revelar la presencia y cantidad de ARN en una muestra biológica en un momento dado.

**TMM**: acrónimo del inglés *Trimmed mean of M values* (en español, media truncada de M-valores).

**Voom**: acrónimo para el modelado de media-varianza a nivel de observación. Se trata de un método para la transformación de los datos de *RNA-Seq* previamente al modelado lineal en *Limma*. La función que realiza esta transformación recibe el mismo nombre que el método y está disponible en el paquete *Limma*.

**Vst**: acrónimo del inglés *Variance Stabilizing Transformation*. *Vst* junto con *Voom* constituyen los métodos de transformación más importantes que se llevan a cabo antes del modelado lineal con *Limma*.

## 7. Bibliografía

1. Metzker ML. Sequencing technologies - the next generation. *Nature reviews Genetics*. 2010 Jan;11(1):31-46. PubMed PMID: 19997069.
2. Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews Genetics*. 2009 Jan;10(1):57-63. PubMed PMID: 19015660. Pubmed Central PMCID: 2949280.
3. McDermaid A, Monier B, Zhao J, Liu B, Ma Q. Interpretation of differential gene expression results of RNA-seq data: review and integration. *Briefings in bioinformatics*. 2018 Aug 6. PubMed PMID: 30099484.
4. Zararsiz G, Goksuluk D, Korkmaz S, Eldem V, Zararsiz GE, Duru IP, et al. A comprehensive simulation study on classification of RNA-Seq data. *PLoS one*. 2017;12(8):e0182507. PubMed PMID: 28832679. Pubmed Central PMCID: 5568128.
5. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2010 Jan 1;26(1):139-40. PubMed PMID: 19910308. Pubmed Central PMCID: 2796818.
6. Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research*. 2015 Apr 20;43(7):e47. PubMed PMID: 25605792. Pubmed Central PMCID: 4402510.
7. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*. 2014;15(12):550. PubMed PMID: 25516281. Pubmed Central PMCID: 4302049.
8. Mardis ER. Next-generation DNA sequencing methods. *Annual review of genomics and human genetics*. 2008;9:387-402. PubMed PMID: 18576944.
9. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*. 1977 Dec;74(12):5463-7. PubMed PMID: 271968. Pubmed Central PMCID: 431765.
10. Rehm HL. Disease-targeted sequencing: a cornerstone in the clinic. *Nature reviews Genetics*. 2013 Apr;14(4):295-300. PubMed PMID: 23478348. Pubmed Central PMCID: 3786217.
11. Besser J, Carleton HA, Gerner-Smidt P, Lindsey RL, Trees E. Next-generation sequencing technologies and their application to the study and control of bacterial infections. *Clinical microbiology and infection : the official publication of the European Society of Clinical Microbiology and Infectious Diseases*. 2018 Apr;24(4):335-41. PubMed PMID: 29074157. Pubmed Central PMCID: 5857210.

12. Peters DG, Yatsenko SA, Surti U, Rajkovic A. Recent advances of genomic testing in perinatal medicine. *Seminars in perinatology*. 2015 Feb;39(1):44-54. PubMed PMID: 25444417. Pubmed Central PMCID: 4883661.
13. Di Resta C, Galbiati S, Carrera P, Ferrari M. Next-generation sequencing approach for the diagnosis of human diseases: open challenges and new opportunities. *Ejifcc*. 2018 Apr;29(1):4-14. PubMed PMID: 29765282. Pubmed Central PMCID: 5949614.
14. Kulkarni P, Frommolt P. Challenges in the Setup of Large-scale Next-Generation Sequencing Analysis Workflows. *Computational and structural biotechnology journal*. 2017;15:471-7. PubMed PMID: 29158876. Pubmed Central PMCID: 5683667.
15. Lam HY, Clark MJ, Chen R, Chen R, Natsoulis G, O'Huallachain M, et al. Performance comparison of whole-genome sequencing platforms. *Nature biotechnology*. 2011 Dec 18;30(1):78-82. PubMed PMID: 22178993. Pubmed Central PMCID: 4076012.
16. Wang K, Kim C, Bradfield J, Guo Y, Toskala E, Otieno FG, et al. Whole-genome DNA/RNA sequencing identifies truncating mutations in RBCK1 in a novel Mendelian disease with neuromuscular and cardiac involvement. *Genome medicine*. 2013;5(7):67. PubMed PMID: 23889995. Pubmed Central PMCID: 3971341.
17. Li S, Tighe SW, Nicolet CM, Grove D, Levy S, Farmerie W, et al. Multi-platform assessment of transcriptome profiling using RNA-seq in the ABRF next-generation sequencing study. *Nature biotechnology*. 2014 Sep;32(9):915-25. PubMed PMID: 25150835. Pubmed Central PMCID: 4167418.
18. Chu Y, Corey DR. RNA sequencing: platform selection, experimental design, and data interpretation. *Nucleic acid therapeutics*. 2012 Aug;22(4):271-4. PubMed PMID: 22830413. Pubmed Central PMCID: 3426205.
19. Finotello F, Di Camillo B. Measuring differential gene expression with RNA-seq: challenges and strategies for data analysis. *Briefings in functional genomics*. 2015 Mar;14(2):130-42. PubMed PMID: 25240000.
20. Ozsolak F, Milos PM. RNA sequencing: advances, challenges and opportunities. *Nature reviews Genetics*. 2011 Feb;12(2):87-98. PubMed PMID: 21191423. Pubmed Central PMCID: 3031867.
21. Anders S, Huber W. Differential expression analysis for sequence count data. *Genome biology*. 2010;11(10):R106. PubMed PMID: 20979621. Pubmed Central PMCID: 3218662.
22. Anders S, McCarthy DJ, Chen Y, Okoniewski M, Smyth GK, Huber W, et al. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nature protocols*. 2013 Sep;8(9):1765-86. PubMed PMID: 23975260.

23. Team RC. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing Vienna, Austria 2019. URL <https://www.R-project.org/>.
24. Ihaka R, Gentleman R. R: A language for data analysis and graphics. *J Comput Graph Stat.* 1996;5(3):299-314.
25. Team. R. RStudio: Integrated Development for R. RStudio, Inc, Boston, MA. 2019. URL <http://rstudio.com/>.
26. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature methods.* 2015 Feb;12(2):115-21. PubMed PMID: 25633503. Pubmed Central PMCID: 4509590.
27. Shendure J. The beginning of the end for microarrays? *Nature methods.* 2008 Jul;5(7):585-7. PubMed PMID: 18587314.
28. Shendure J, Ji H. Next-generation DNA sequencing. *Nature biotechnology.* 2008 Oct;26(10):1135-45. PubMed PMID: 18846087.
29. Kukurba KR, Montgomery SB. RNA Sequencing and Analysis. *Cold Spring Harbor protocols.* 2015 Apr 13;2015(11):951-69. PubMed PMID: 25870306. Pubmed Central PMCID: 4863231.
30. Smyth GK. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology.* 2004;3:Article3. PubMed PMID: 16646809.
31. Law CW, Chen Y, Shi W, Smyth GK. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome biology.* 2014 Feb 3;15(2):R29. PubMed PMID: 24485249. Pubmed Central PMCID: 4053721.
32. Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome biology.* 2010;11(3):R25. PubMed PMID: 20196867. Pubmed Central PMCID: 2864565.
33. Seyednasrollah F, Laiho A, Elo LL. Comparison of software packages for detecting differential expression in RNA-seq studies. *Briefings in bioinformatics.* 2015 Jan;16(1):59-70. PubMed PMID: 24300110. Pubmed Central PMCID: 4293378.
34. Robinson MD, Smyth GK. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics.* 2008 Apr;9(2):321-32. PubMed PMID: 17728317.
35. Hardcastle TJ, Kelly KA. baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. *BMC bioinformatics.* 2010 Aug 10;11:422. PubMed PMID: 20698981. Pubmed Central PMCID: 2928208.
36. Leng N, Dawson JA, Thomson JA, Ruotti V, Rissman AI, Smits BM, et al. EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq

- experiments. *Bioinformatics*. 2013 Apr 15;29(8):1035-43. PubMed PMID: 23428641. Pubmed Central PMCID: 3624807.
37. Tarazona S, Furio-Tari P, Turra D, Pietro AD, Nueda MJ, Ferrer A, et al. Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package. *Nucleic acids research*. 2015 Dec 2;43(21):e140. PubMed PMID: 26184878. Pubmed Central PMCID: 4666377.
38. Li J, Tibshirani R. Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. *Statistical methods in medical research*. 2013 Oct;22(5):519-36. PubMed PMID: 22127579. Pubmed Central PMCID: 4605138.
39. Wang L, Feng Z, Wang X, Wang X, Zhang X. DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics*. 2010 Jan 1;26(1):136-8. PubMed PMID: 19855105.
40. Soneson C, Delorenzi M. A comparison of methods for differential expression analysis of RNA-seq data. *BMC bioinformatics*. 2013 Mar 9;14:91. PubMed PMID: 23497356. Pubmed Central PMCID: 3608160.
41. Li X, Brock GN, Rouchka EC, Cooper NGF, Wu D, O'Toole TE, et al. A comparison of per sample global scaling and per gene normalization methods for differential expression analysis of RNA-seq data. *PloS one*. 2017;12(5):e0176185. PubMed PMID: 28459823. Pubmed Central PMCID: 5411036.
42. Rapaport F, Khanin R, Liang Y, Pirun M, Krek A, Zumbo P, et al. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome biology*. 2013;14(9):R95. PubMed PMID: 24020486. Pubmed Central PMCID: 4054597.
43. Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, et al. A survey of best practices for RNA-seq data analysis. *Genome biology*. 2016 Jan 26;17:13. PubMed PMID: 26813401. Pubmed Central PMCID: 4728800.