Grado en Estadística

TRABAJO FIN DE GRADO

Análisis de datos de RNA-Seq: comparación de métodos para el estudio de expresión génica diferencial.

Sara del Carmen Sánchez Santana

Departamento de Estadística e Investigación Operativa

Tutora: Mª Dolores Cubiles de la Vega

Junio 2015

Índice

1. ABSTRACT

RNA-Seq is a next generation sequencing (NGS) procedure of the DNA for discovering, profiling and quantifying RNA transcripts, so this technology allows for measure gene expression. The analysis of RNA-Seq data, which comprise discrete counts of reads mapped to genes or transcripts, is made up of different parts: quality control checks on raw sequence data, mapping reads to a reference genome, counts of reads and the detection of differentially expressed genes across different biological conditions.

The Bioconductor project provides tools for identifying differential expression for RNA-Seq data by means of the use of R statistical programming language. We focus our attention on four Bioconductor packages: EdgeR, NOISeq, DESeq2 and Limma. Now let's see the main features of these packages.

- EdgeR: This package is based on the negative binomial distribution. The default method for the normalization of data is the trimmed mean of M-values (TMM). For filter out transcripts with very low counts, EdgeR establishes a minimum of counts per millions (CPM). Finally, it uses the exact test for the binomial negative to determinate differentially expressed genes.

- NOISeq: It is based on non-parametric approaches for the differential expression analysis of RNA-Seq data. In this case, the normalization default technique implemented is RPKM (reads per kilobase per million). The method to filter out transcripts with low counts is CPM (counts per millions). The differential expression analysis between two experimental conditions is based on the probability of each transcript or gene of being differently expressed: it is obtained by comparing the differential expression statistics, M-D values, of that transcript or gene against distribution of changes in expression values when comparing replicates within the same condition, noise distribution.

- DESeq2: It is based on negative binomial generalized linear models. The technique used for the normalization is normalisation based on the estimation of the effective library size. DESEq2 filters out the transcripts which very low counts by means of the mean of normalized counts for each gene. Finally, it uses the Wald test for determining differentially expressed genes.

- Limma: This Bioconductor package was designed originally for the analysis of Microarray data, but it has been adapted for the analysis of RNA-Seq data. It is based on the use of linear models. The normalization default technique is the trimmed mean of M-values (TMM). To filter

out transcripts with very low counts, Limma uses the minimum of counts per millions (CPM). This package uses the t-test for determining differentially expressed genes.

We analyze RNA-Seq data of eight patients with asthma. There are four untreated patients and four patients with dexamethasone, a potent glucocorticoid. We want to identify the number of differentially expressed transcripts between these two experimental conditions: untreated-dexamethasone. It can be done by using the four Bioconductor packages seen previously. The obtained results are: 853 differentially expressed transcripts with EdgeR, 415 transcripts with NOISeq, 1212 transcripts with DESeq2, and 719 with Limma. These four packages agree on 277 differentially expressed transcripts between untreated patients and treated (dexamethasone) patients.

2. OBJETIVO DEL TRABAJO

El objetivo principal de este trabajo consiste en describir el procedimiento a seguir en el análisis estadístico de datos de expresión génica de RNA-Seq, desde el análisis previo de dichos datos hasta la obtención de aquellos genes o tránscritos diferencialmente expresados bajo distintas condiciones biológicas. Pondremos especial énfasis en la comparación estadística de cuatro paquetes de Bioconductor en la plataforma R utilizados para el análisis de expresión génica diferencial. Bioconductor es un software que proporciona herramientas para el análisis y compresión de datos genómicos utilizando el lenguaje de programación estadística R. Los paquetes de Bioconductor que vamos a estudiar y comparar en nuestro trabajo son: EdgeR, NOISeq, DESeq2 y Limma.

Para llevar a cabo dicho objetivo, se realizará un análisis de datos reales procedentes de 8 pacientes que padecen una enfermedad respiratoria crónica, asma, entre los cuales 4 han sido tratados con dexametasona, un potente glucocorticoides sintético, y 4 no han recibido ningún tipo de tratamiento. Dispondremos de una serie de datos de conteo obtenidos mediante RNA-Seq sobre los cuales aplicaremos los cuatro paquetes de Bioconductor mencionados anteriormente. Lo que se pretende con ello es obtener para cada paquete tanto el número de tránscritos diferencialmente expresados como conocer qué tránscritos presentan mayor diferencia de expresión entre pacientes tratados con dexametasona y pacientes sin tratamiento.

Como veremos en las conclusiones, obtendremos para cada paquete un número diferente de tránscritos diferencialmente expresados, pudiendo incluso variar la identificación de aquellos que presentan diferencias más representativas entre distintas condiciones biológicas. Esto es debido, principalmente, a que los paquetes difieren en el modelado probabilístico de los datos de conteo procedentes del número de reads obtenidas mediante RNA-Seq para cada uno de los pacientes. Por consiguiente, en dicho trabajo se realizará una descripción de cada paquete, centrándonos en el estudio del modelo empleado en cada uno de ellos para los correspondientes datos de conteo, y se realizará, posteriormente, una comparación estadística entre ellos en base a los resultados obtenidos tras el análisis de los datos reales de los que disponemos.

3. CONTEXTO BIOLÓGICO: RNA-SEQ

La Bioinformática es una rama de la ciencia cuyo objetivo fundamental es el uso de base de datos biológicos, médicos o sanitarios, y algoritmos computacionales para analizar proteínas, genes y la completa colección de ADN que forma un organismo, es decir, el genoma. Dichas herramientas computacionales permiten detectar los distintos mecanismos fundamentales que hay detrás de ciertos problemas biológicos relacionados con la estructura y función de las macromoléculas, enfermedades, etc.

Dentro de las diversas áreas de investigación de la Bioinformática se encuentra el Análisis de datos de Expresión Genética, es decir, el estudio de los ARNm trancritos por un conjunto de genes en distintas condiciones experimentales. La expresión genética es un proceso en el que la información presente en una secuencia de ADN se transforma en proteínas. Para ello, una parte de dicho ADN se transcribe creando una molécula de ARN que, posteriormente, dará paso a la síntesis de proteínas. La secuencia de ADN constituye la información genética heredable del núcleo celular, los plásmidos, la mitocondria y, en el caso de las plantas, los cloroplastos, que forman la base de los programas de desarrollo de los seres vivos. Dentro del núcleo, se encuentra un numeroso conjunto de genes que codifican proteínas, pero sólo un subconjunto de ellos serán expresados. Esto puede variar en respuesta a una gran variedad de condiciones o circunstancias: enfermedad, paso del tiempo, condiciones ambientales…

Muchas de las preguntas biológicas que se plantea la Bioinformática en cuanto al Análisis de expresión génica, se resuelven a partir de la comparación entre distintos perfiles de expresión genética. Actualmente, se disponen de diferentes técnicas que permiten secuenciar ADN a gran velocidad. Dicha secuenciación proporciona representaciones lineales (secuencias) que resumen la estructura atómica (nucleótidos: A, C, G y T) de la molécula secuenciada. Cuando el ADN es secuenciado, una de las principales tareas es conocer en qué tramo de dicha secuencia se encuentran los genes y qué función desempeñan. Dicha secuenciación se puede clasificar en función de cómo se desarrolle el proceso de secuenciado: métodos básicos, métodos avanzados y métodos de secuenciación masiva (NGS). Las técnicas de secuenciación masiva de ADN permiten secuenciar mayor número de nucleótidos, en comparación con el resto de métodos. RNA-Seq es una tecnología que utiliza la ultrasecuenciación, una de las técnicas de secuenciación masiva de ADN.

Los experimentos de expresión génica diferencial se realizan con el fin de detectar y analizar diferencias en la expresión de los genes de un organismo bajo diferentes condiciones. Este estudio permite, además, realizar anotaciones sobre los genomas secuenciados hasta el momento. Con los nuevos métodos de secuenciación masiva, se ha conseguido facilitar la identificación de aquellos genes que son muy expresados o, simplemente, expresados, en algún estado biológico concreto, y, además, analizar la relación de comportamiento de un conjunto de genes bajo ciertas condiciones experimentales. En conclusión, estos experimentos constituyen una herramienta fundamental para el estudio de la relación entre genes y procesos biológicos o patologías.

Concepto RNA-Seq:

RNA-Seq es un método de secuenciación masiva (NGS) de ADN que permite analizar el transcriptoma, es decir, el conjunto de ARN mensajero transcrito. A diferencia con la secuenciación exclusiva del ADN, el análisis del transcriptoma nos permite conocer más información sobre las muestras ya que en un mismo gen podemos encontrar distintos tránscritos y, por lo tanto, podemos proceder al análisis de isoformas alternativas.

Mediante la transcripción de ADN a ARN se obtiene el ARN mensajero primario. En un gen existen fragmentos de ADN cuya secuencia no será traducida a aminoácidos, conocidos con el nombre de intrones. Por el contrario, las regiones de este gen que codificarán la secuencia de aminoácidos de la futura proteína son los exones. El splicing alternativo provoca la generación de distintas isoformas ya que permite que en un mismo gen se codifique la información necesaria para sintetizar distintas proteínas. Este proceso se basa, a partir del ARN mensajero primario, en combinar los distintos exones obteniendo diferentes ARN mensajeros (tránscritos) que permitan la generación de distintas proteínas partiendo de un mismo gen.

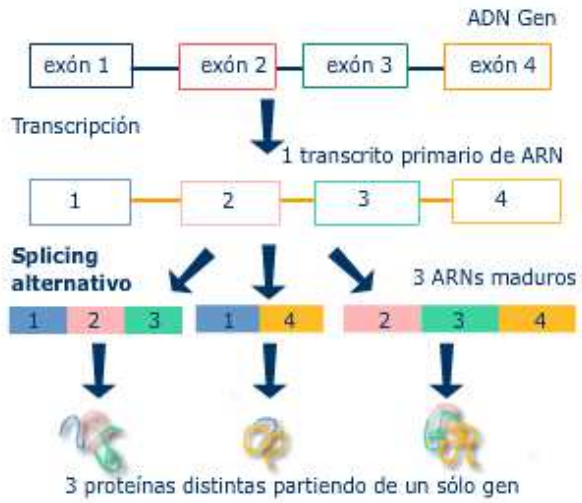


Figura 1. Splicing alternativo

La tecnología RNA-Seq, además de proporcionar una aproximación al perfil transcriptómico, permite medir los niveles de tránscritos bajo distintas condiciones experimentales, de forma más precisa que otros métodos como los Microarrays, posibilitando la comparación entre distintos perfiles transcriptómicos.

El proceso de trabajo de RNA-Seq se basa, en primer lugar, en convertir un conjunto de ARN mensajero en una librería de fragmentos de ADNc con adaptadores en uno o ambos extremos. Mediante el splicing alternativo que comentábamos anteriormente, obtenemos los ARN mensajeros que contienen, en teoría, sólo secuencia codificante. Se realiza la secuenciación complementaria de dichos ARNm, cambiando los Uracilos por Timinas, obteniendo una secuencia de ADN del genoma original que contiene únicamente los exones que hay en el genoma de referencia. Esto es lo que recibe el nombre de ADNc.

Mediante las tecnologías de secuenciación masiva, obtenemos, finalmente, pequeñas secuencias de nucleótidos, llamadas reads, que corresponden a cada fragmento perteneciente a la librería de fragmentos de ADNc obtenida. El número de reads varía en función de la expresión de un determinado gen, es decir, si un gen se expresa más, tendrá más ADNc y, por tanto, es más probable que tenga un mayor número de reads. Estas reads son recogidas en archivos .fastq, que incluyen, principalmente, un identificador, el tamaño, la calidad por cada nucleótido y la secuencia en sí.

El primer paso en el análisis de datos de RNA- Seq es el análisis de calidad de las reads a través de los archivos .fastq. Una vez realizado el estudio de calidad correspondiente, estas secuencias se alinean y mapean frente al genoma humano de referencia. Existen distintas herramientas de mapeo: TopHat (usa Bowtie1), TopHat2 (usa Bowtie1/Bowtie2), HPG Aligner… Estos programas generan archivos de extensión .bam. En cada uno de estos archivos se encuentra la posición de cada read en el genoma de referencia, indicando el cromosoma, el gen y el tránscrito correspondiente, además de la secuencia y la calidad. Una vez realizado el alineamiento, se procede a contar el número de reads por gen o tránscrito para nuestra muestra. Para ello se emplea el software HTSeq-count, el cual genera un archivo .counts que cuantifica el número de reads en cada gen o tránscrito de la muestra en relación al genoma de referencia. De dicha herramienta obtenemos un archivo de extensión .tab cuya estructura es la siguiente: por fila, los genes o tránscritos, y por columna, nuestra muestra o muestras, en el caso de que nuestro objetivo sea observar la expresión génica diferencial entre distintos organismos o un mismo organismo bajo diferentes condiciones. Una vez obtenidos los archivos .tab, existen numerosas herramientas para proceder a su respectivo análisis. El siguiente esquema muestra, a modo de resumen, los pasos a seguir en el análisis de datos de RNA-Seq.

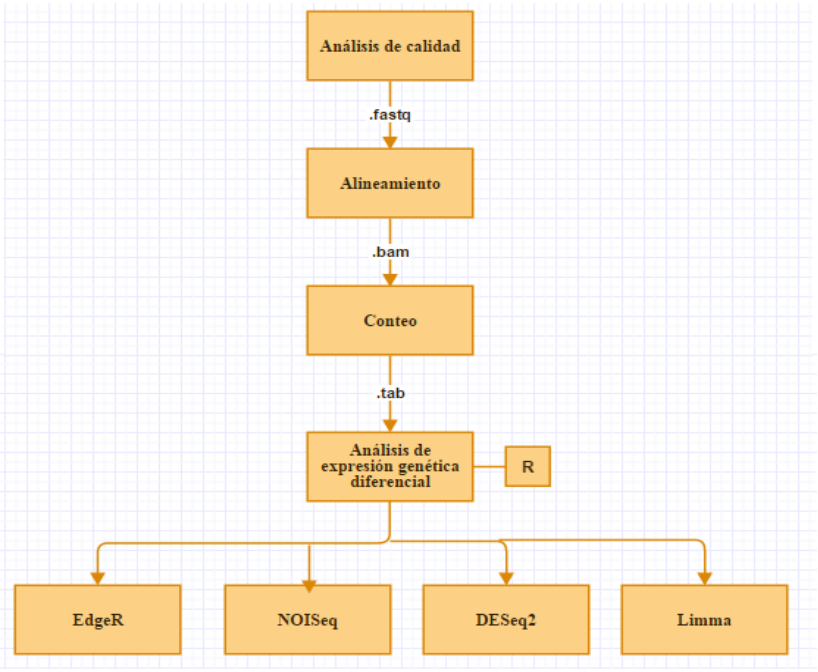


Figura 2. Pasos a seguir en el análisis de datos de RNA-Seq

A continuación, veremos de forma detallada los distintos pasos del análisis de datos de RNA-Seq y nos sumergiremos en el estudio estadístico de cuatro paquetes de la plataforma R para el correspondiente análisis de los datos obtenidos.

Como hacíamos referencia en las áreas de investigación de la Bioinformática, el objetivo de los experimentos de expresión génica diferencial es estudiar las diferencias en la expresión de los genes o los tránscritos de un organismo bajo distintas condiciones. Para ello necesitamos tener distintas muestras dentro del conjunto de datos, siendo importante conocer la procedencia y las características de dichas muestras. A la hora de estudiar la expresión diferencial debemos saber si disponemos de datos con réplicas técnicas y/o biológicas. Las réplicas técnicas son aquellas que se obtienen del mismo individuo, es decir, se realiza la secuenciación un número determinado de veces sobre el mismo individuo. Las réplicas biológicas, por el contrario, son aquellas que se obtienen de distintos individuos pero que comparten una determinada característica, por ejemplo, se realiza la secuenciación de tres pacientes a los que se les suministra un determinado medicamento. A continuación se muestra, a modo de ejemplo, una matriz de conteo obtenida mediante RNA-Seq, que corresponde al archivo de extensión .tab mencionado anteriormente.

	Condición 1				Condición 2			
	Muestra 1	Muestra 2	…	Muestra k	Muestra k+1	Muestra k+2	…	Muestra p
Gen 1	145	98		55	23	36		72
Gen 2	12	7		18	14	24		21
Gen 3	0	0		2	5	6		3
…	…	…		…	…	…		…
Gen n	576	758		647	358	247		176

Tabla 1. Matriz de conteo

4. ANÁLISIS DE DATOS DE RNA-SEQ

Vamos a ver, de forma detallada, los distintos pasos que se llevan a cabo en el análisis de datos de RNA-Seq.

4.1. Análisis de calidad

El primer paso de nuestro análisis de datos de RNA-Seq, es el análisis de calidad de nuestras reads. Para ello se utiliza el programa FastQC, una herramienta de control de calidad de datos de secuenciación de nueva generación.

El objetivo fundamental de FastQC es proporcionar una manera sencilla de hacer algunas comprobaciones de control de calidad de las reads obtenidas a través de las tecnologías de secuenciación masiva (NGS). Dicha herramienta aporta un conjunto de análisis que permiten obtener, principalmente a través de gráficos y tablas, una primera impresión de posibles problemas en las reads, a tener en cuenta antes de seguir con nuestro análisis, permitiendo la depuración de las mismas en caso de ser necesario. A continuación, vamos a describir cada una de dichas comprobaciones de calidad.

- Basic Statistics

Measure	Value
Filename	SRR1039508_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	22935521
Filtered Sequences	0
Sequence length	63
%GC	50

Tabla 2. Basic Statistics

En primer lugar, FastQC proporciona una tabla con algunos datos estadísticos de nuestras reads, entre los que cabe destacar: el nombre del fichero .fastq, el número de reads procesadas, la longitud de dichas reads (pueden tener distinto tamaño) y el porcentaje de GC global (es

decir, el contenido de los nucleótidos GC de todas las bases de todas las secuencias). Este porcentaje de GC se considera aceptable cuando supera el 45%. Además, es aconsejable que la longitud de las reads sea la misma para evitar complicaciones en el resto de comprobaciones de calidad.

- Per Base Sequence Quality

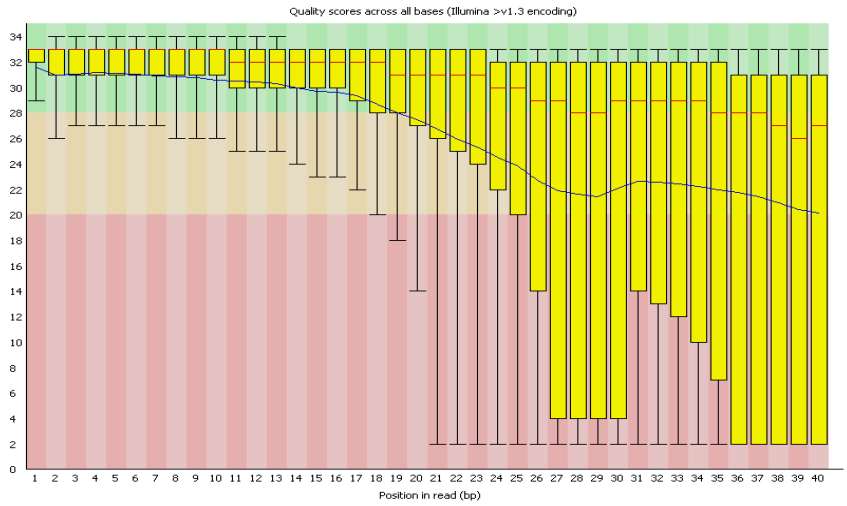


Figura 3. Per Base Sequence Quality

Dicho gráfico nos permite obtener una visión de la calidad por base (nucleótido) de nuestras reads. Se trata del gráfico de comprobación de calidad que, en general, recibe más importancia.

Este gráfico presenta un Box-Plot para cada base. Las líneas rojas se corresponden con la mediana y la línea azul con la calidad media de las bases. El eje y representa la calidad de cada base. A mayor valor en el eje y, mejor será la calidad de dicha base. El fondo del gráfico aparece dividido en tres colores: en color verde, la zona de muy buena calidad; en naranja, la zona de calidad razonable; y en rojo, la zona de mala calidad. Es normal que a medida que la secuenciación avanza a lo largo del tamaño de la read, ésta vaya cometiendo más errores. Esto es debido a que la tecnología de secuenciación utilizada por los secuenciadores más comunes (como la plataforma Ilumina), cuando va incorporando nucleótidos a la read, aumenta la posibilidad de error; el proceso de secuenciación mientras va avanzado es más probable que falle.

FastQC dará una señal de advertencia cuando el cuantil más pequeño de cualquier base sea inferior a 10, o si la mediana es inferior a 25, y señalará un fracaso en la calidad de las reads cuando el cuantil inferior de cualquier base sea inferior a 5, o la mediana inferior a 20. En este

último caso, un procedimiento a seguir es cortar las reads a partir de aquellas bases donde obtenemos muy mala calidad.

- Per Sequence Quality Scores

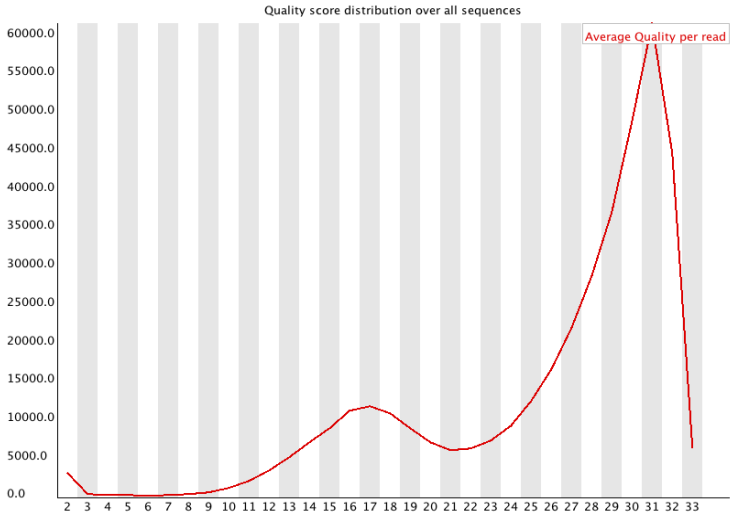


Figura 4. Per Sequence Quality Scores

Este gráfico permite comprobar la calidad en subconjuntos de reads. Se trata de una distribución de las calidades medias del conjunto de reads.

FastQC dará una señal de advertencia cuando la calidad media observada con mayor frecuencia esté por debajo de 27, lo que equivaldría a una tasa de error de 0'2%, y señalará un fracaso cuando la calidad media observada con mayor frecuencia esté por debajo de 20, lo que equivaldría a una tasa de error de 1%.

- Per Base Sequence Content

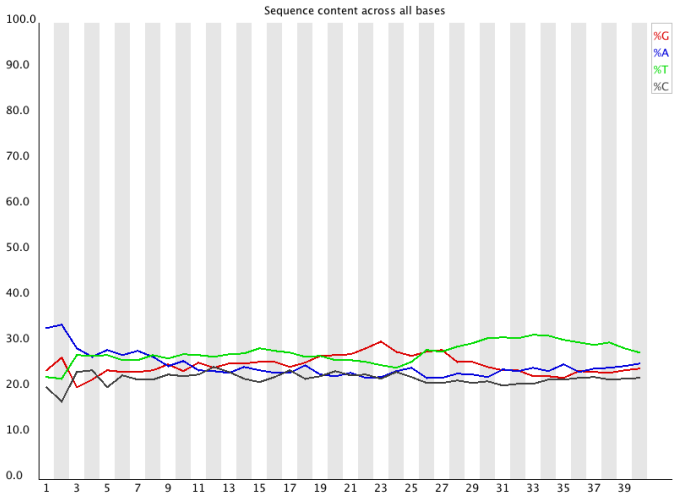


Figura 5. Per Base Sequence Content

Este gráfico muestra la proporción de cada base de los nucleótidos del ADN (G, A, T, C) en cada base de las reads Lo ideal, en dicho gráfico, es que las líneas sean paralelas. Las cantidades relativas a cada base en cada posición de las reads no deberían ser muy desequilibradas, pues dichas cantidades deben reflejar, en cierto modo, la proporción de dichas bases en el genoma.

FastQC dará una señal de advertencia cuando la diferencia entre A y T, o C y G, sea mayor que 10% para cualquier posición, y señalará un fracaso cuando dicha diferencia sea mayor que 20% en cualquier posición.

- Per Sequence GC Content

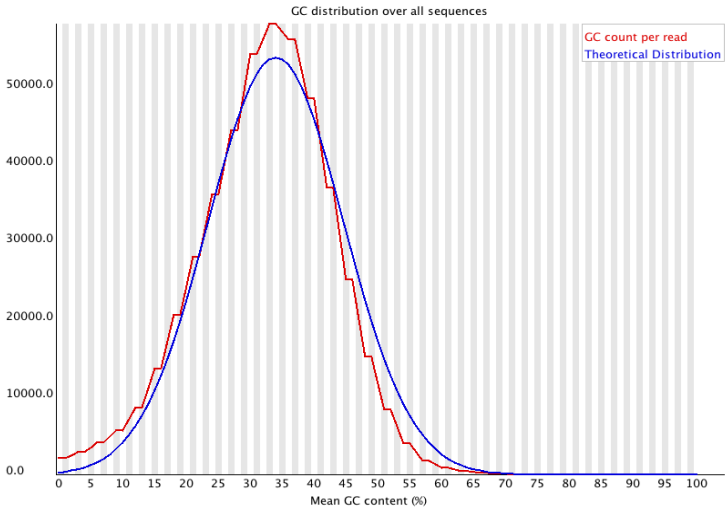


Figura 6. Per Sequence GC Content

Este gráfico presenta la media de contenido de GC en las reads y compara dicho contenido con la distribución normal. Lo correcto es observar una distribución prácticamente normal de contenido de GC, correspondiendo el pico central al contenido de GC del genoma.

FastQC dará una señal de advertencia si la suma de las desviaciones de la distribución normal representa más del 15% de las reads, y señalará un fracaso si dicha suma representa más del 30% de las reads.

- Per base N Content

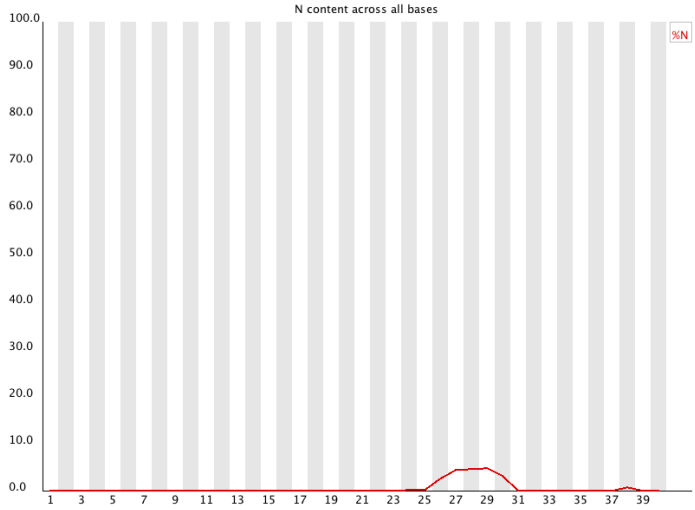


Figura 7. Per Base N Content

Este gráfico muestra la proporción de N (nucleótido desconocido) que se observa en cada posición de las reads.

En ocasiones, las herramientas de secuenciación no consiguen incorporar con certeza una de las bases de los nucleótidos del ADN, por lo que se ven forzadas a agregar una N en dicha posición. Es muy probable que la aparición de Ns ocurra cuando vamos avanzando en las posiciones de las reads. Esto es debido a que, como comentábamos anteriormente, a medida que cualquier herramienta de secuenciación avanza en dicho procedimiento, se produce una disminución de la calidad de secuenciación.

FastQC dará una señal de advertencia si en cualquier posición obtenemos un porcentaje de N mayor del 5%, y señalará un fracaso si dicho porcentaje de N es mayor del 20%.

- Sequence Length Distribution

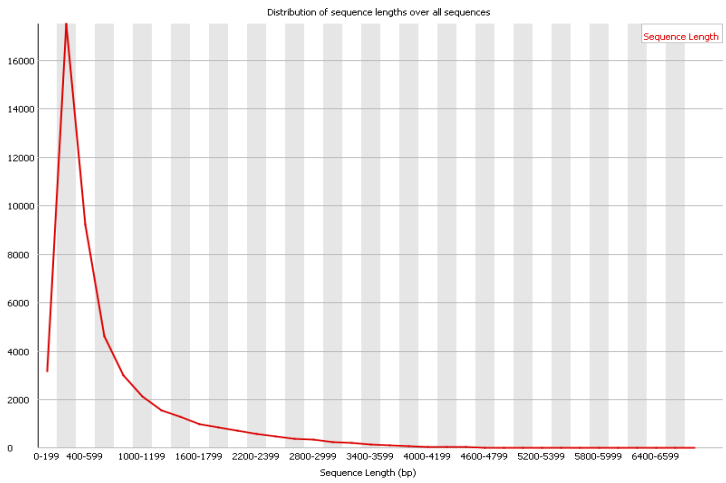


Figura 8. Sequence Length Distribution

Este gráfico muestra la distribución de tamaño de las reads.

Como bien mencionábamos anteriormente, lo aconsejable es tener reads de la misma longitud. No obstante, este gráfico no presenta especial relevancia pues para distintas herramientas de uso posterior el hecho de contar con reads de distintos tamaño no supone ningún problema.

FastQC dará una señal de advertencia si todas las reads no presentan la misma longitud y señalará un fracaso si existen reads con longitud cero.

- Duplicate Sequences

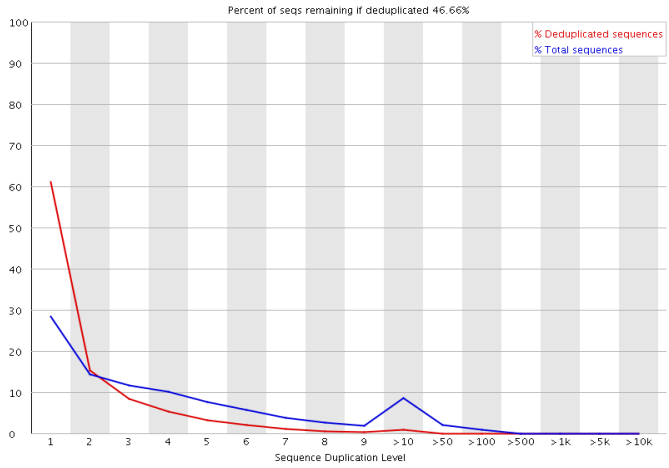


Figura 9. Duplicate Sequences

Este gráfico muestra el porcentaje de reads con distintos grados de duplicidad. En el eje y se representa el porcentaje de reads repetidas y en el eje x, el número de repeticiones.

La interpretación e importancia de dicho gráfico depende del fin que se desea alcanzar en el estudio que se está llevando a cabo, es decir, en determinadas ocasiones, la duplicidad de reads puede suponer un problema en el análisis y, sin embargo, en otras casos, esta duplicidad no supone ningún inconveniente. Por ejemplo, si el objetivo es secuenciar el genoma humano, lo normal es que el porcentaje de duplicación sea constante, pues de no ser así, sería señal de que existen partes del genoma que se han secuenciado en abundancia y zonas que no, dado que el número de reads disponibles es el factor limitante.

FastQC dará una señal de advertencia si el porcentaje de secuencias que presentan duplicidad es mayor del 20% del total de reads y señalará un fracaso si dicho porcentaje supera el 50% del total.

- Overrepresented Sequences

En este caso, obtendremos una tabla en la que aparecerán aquellas secuencias que se repiten un número elevado de veces.

Sequence	Count	Percentage	Possible Source
ACACGTCTGAACTCCAGTCACCGATGTATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAA	57182	0.24931633338523243	TruSeq Adapter, Index 2 (100% over 51bp)

Tabla 3. Overrepresented Sequences

Al igual que ocurría en el gráfico anterior, la importancia de dicha tabla es relativa al fenómeno que estamos estudiando. Puede ocurrir que en nuestro estudio, el hecho de obtener un porcentaje elevado para una secuencia determinada sea lo "normal", como ocurre, por ejemplo, en el caso del tumor de mama, que existe una secuencia de 24 nucleótidos que aparece repetida un 30% aproximadamente. Se trata, en este caso, de un microRNA. Este porcentaje afectaría de la misma forma que veíamos en el anterior gráfico, pero, como hemos mencionado, sabemos, dado que es en el caso del tumor de mama, que es algo aceptable. El problema ocurre cuando la secuencia que aparece repetida, no es una secuencia biológica (es decir, no está presente en el genoma que estamos secuenciando), lo que puede llevar a pensar que ha habido algún tipo de problema en la secuenciación de las reads.

Otra posible interpretación de la repetición de secuencias es debida a la existencia de adaptadores en las reads. En el proceso de secuenciación, antes de secuenciar la read, se genera una secuencia de entre 10 y 20 nucleótidos complementarios a la zona del genoma que vamos a secuenciar. Este adaptador se une al genoma y posteriormente se empieza a secuenciar. A mayor tamaño de las reads, mayor será el tamaño de los adaptadores y, por tanto, mayor posibilidad de que dicha secuencia aparezca frecuentemente repetida en el total de reads.

FastQC dará una señal de advertencia si el porcentaje que representa una secuencia repetida constituye más del 0'1% del total de secuencias y señalará un fracaso si dicho porcentaje supera el 1% del total de secuencias.

-Adapter Content

Esta parte de nuestro análisis de calidad hace énfasis en la detección de posibles adaptadores en las reads, a modo, en cierta medida, de complemento a la tabla anteriormente vista.

Como ya hemos mencionado en otra ocasión, lo normal es que en cada posición de las reads aparezca un nucleótido distinto entre ellas. Dado que el número de nucleótidos es 4, el

porcentaje de cada nucleótido en cada posición del total de reads debe ser, aproximadamente, de un 25%. Sin embargo, en el caso de una repetición abundante de ciertos adaptadores, vamos a obtener en las primeras posiciones de las reads, porcentajes en torno al 50%.

FastQC dará una señal de advertencia si una determinada secuencia aparece repetida en más del 5% del total de reads y señalará un fracaso si aparece repetida en más del 10% del total.

- Kmer Content

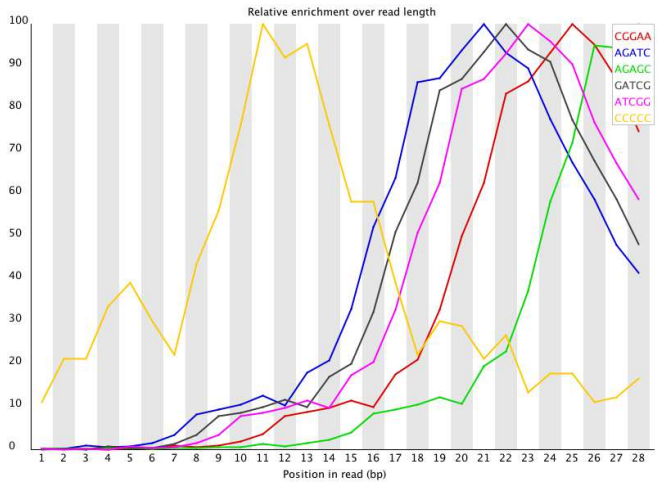


Figura 10. Kmer Content

Este gráfico está integrado por el conjunto de secuencias de entre 5 y 7 nucleótidos que aparecen un mayor número de veces en el total de reads. En dicho análisis, una vez determinadas dichas secuencias, se representa la posición que ocupan éstas respecto a todas la reads.

El hecho de que una determinada secuencia aparezca repetida, por ejemplo, tras un adaptador en, prácticamente, el total de reads, puede ayudarnos a detectar posibles problemas en la secuenciación, ya que se están introduciendo los mismos nucleótidos en las reads tras el adaptador.

FastQC dará una señal de advertencia si cualquiera de las secuencias que integra el gráfico aparece repetida más de un 3% sobre la longitud de las reads o más de un 5 % en una determinada posición, y señalará un fracaso si aparece repetida más de un 10% en una posición determinada de las reads.

4.2. Alineamiento de las reads

Una vez llevado a cabo el análisis de calidad de las reads, junto con la respectiva depuración de las mismas, se procede al alineamiento y mapeo de las reads frente a un genoma de referencia. El objetivo fundamental de este paso es conocer la ubicación de las correspondientes reads con respecto a dicha referencia. Como bien comentábamos anteriormente, existen diversas herramientas para dicha función: TopHat (usa Bowtie1), TopHat2 (usa Bowtie1/Bowtie2), BWA, Novoalign, HPG Aligner…

TopHat es una herramienta que permite, entre otras funciones, mapear reads por tránscritos, dando la posibilidad de descubrir splicing alternativo con dichos datos. En un primer lugar, realiza un alineamiento usando Bowtie (1/2). Sin embargo, en dicho alineamiento puede ocurrir que algunas reads no se alineen/mapeen a alguna secuencia perteneciente al genoma de referencia, bien por problemas de contaminación de las reads o cuando partimos de muestras muy alteradas (como ocurre en el caso de tumores muy agresivos). TopHat intenta alinear dichas reads no alineadas mediante un algoritmo más preciso que permita la existencia de huecos en el alineamiento.

Dichas herramientas generan archivos de extensión .bam. Se trata de archivos que contienen datos de alineamientos separados por tabulación. Cada uno de ellos presenta la siguiente estructura por filas: nombre/identificador de la read; posición de dicha read en el genoma de referencia, indicando el cromosoma en el que se encuentra, el punto inicial y final del lugar que ocupa dentro del genoma de referencia, permitiendo identificar el gen y el tránscrito, o tránscritos, correspondientes; la secuencia y la calidad de dicha secuencia. Es conveniente aclarar que, por lo general, cada read va a corresponder a un único exón, pero, sin embargo, dicho exón puede formar parte de uno o varios tránscritos. Dependiendo de los parámetros que seleccionemos, en la etapa posterior de conteo, dicha read puede considerarse en cada uno de los tránscritos que presentan el exón correspondiente o descartar la correspondiente read. Debido a esta controversia, en muchos casos se habla de genes y no de tránscritos, para evitar este tipo de situaciones. A continuación, se muestra, a modo de ejemplo, parte de un archivo de extensión .bam.

Figura 11. Parte de archivo de extensión .bam

4.3. Conteo de las reads

Tras el alineamiento y mapeo de las reads frente al genoma de referencia, se procede a realizar el conteo de reads por gen o tránscrito para cada muestra. En dicho proceso se genera un archivo .counts que cuantifica las reads de la muestra en relación a los genes o tránscritos que aparecen en el genoma de referencia. Una vez obtenido dicho fichero para cada muestra, se obtiene un fichero .tab que agrupa el conteo de reads para todas las muestras y que se utilizará para el posterior análisis de expresión génica diferencial. En definitiva, se obtiene, finalmente, una matriz de conteo para cada uno de los genes o tránscritos y para cada muestra, que permitirá contabilizar el nivel de expresión de genes o tránscritos, de modo que para aquella muestra que un gen o tránscrito se expresa más que en otra, el número de reads alineadas contra ese gen o tránscrito será mayor que para la otra muestra.

Para el conteo de las reads se emplea el software HTSeq-count. Dicha herramienta permite cuantificar la superposición de reads a genes o tránscritos, midiendo, por lo tanto, el número de reads mapeadas a un determinado gen o tránscrito. El archivo de extensión .tab contiene los datos de conteo separados por tabulación y presenta la siguiente estructura: por fila, los distintos genes o tránscritos sobre los cuales han sido mapeadas las reads, y por columna, las distintas muestras sobre las que se desea realizar el análisis de expresión genética diferencial. La siguiente imagen presenta, a modo de ejemplo, una matriz de conteo, integrada en un archivo de extensión .tab. La primera fila, correspondiente a los nombres de las distintas muestras, aparece trasladada debido a que la longitud de dichos nombres nos impide verlos en la cabecera de las columnas.

File Path ▾ : ~/Projects/miARma/miARma_RNA_examples/3.Read_count/HtseqFormat_results/top_bw2-ReadCount.tab

top_bw2-ReadCount.tab

	SRR1039508	SRR1039509	SRR1039510	SRR1039511	SRR1039512	SRR1039513	SRR1039514	SRR1039515	SRR1039516	SRR1039517	SRR1039518	SRR1039519	SRR1039520	SRR1039521	SRR1039522	SRR1039523
ENST00000000233	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ENST00000000412	6	4	0	0	0	0	4	0	2	4	2	4	0	0	0	2
ENST00000000442	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENST00000001008	266	225	246	266	266	154	332	196	266	277	284	207	211	170	251	247
ENST00000001146	2	0	1	0	2	0	5	4	1	1	1	0	7	2	2	0
ENST00000002125	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENST00000002165	1358	1073	1441	964	1752	901	1985	1556	1388	1409	1315	1199	1306	1061	1777	1354
ENST00000002501	1	1	1	0	0	0	0	0	1	2	1	3	5	1	2	2
ENST00000002596	3	0	0	2	1	0	2	0	0	3	0	1	2	2	3	1
ENST00000002829	3	2	3	6	10	0	8	6	13	7	8	8	4	7	8	6
ENST00000003084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENST00000003100	133	116	146	108	140	39	200	121	177	175	130	123	64	120	193	128

Figura 12. Parte de archivo de extensión .tab

4.4. Análisis de expresión génica diferencial

El conteo de reads genera una archivo .tab, una matriz de conteo a partir de la cual procederemos al análisis de expresión genética diferencial. Existen numerosas herramientas para llevar a cabo dicho análisis, teniendo como elementos distintivos entre ellas los métodos de normalización, el filtrado de características (genes, tránscritos) con bajo nivel de expresión, los modelos probabilísticos, los correspondientes test para el análisis de la expresión diferencial, etc. Nos centraremos en el estudio detallado de cuatro paquetes de Bioconductor en la plataforma R para llevar a cabo dicho análisis, profundizando, principalmente, en los modelos estadísticos propios de cada uno de los paquetes. El objetivo fundamental de estos paquetes es comparar el conteo de reads para cada tránscrito/gen bajo diferentes condiciones biológicas, mediante test estadísticos. Dichos paquetes son: EdgeR, NOISeq, DESEq2 y Limma.

Para poder decidir si, para un gen determinado, existen diferencias significativas en el número de reads mapeadas a dicho gen bajo diferentes condiciones biológicas, es necesario realizar un test, para el cual, a su vez, es necesario modelar el conteo de reads a una distribución determinada. En los modelos de conteo, la variable dependiente es discreta y no negativa y, de forma clásica, se suele utilizar la distribución de Poisson para modelar este tipo de datos. Dicha distribución presenta un solo parámetro, determinado por su media. Además, la varianza de dicha distribución es igual a su media. El hecho de que la media coincida con la varianza es una característica demasiado restrictiva: puede predecir variaciones más pequeñas en comparación con la realidad mostrada por los datos. Cuando se observa una varianza superior a la esperada, decimos que estamos ante un problema de sobredispersión, bajo lo cual se aconseja no hacer uso de la distribución de Poisson en dicho caso. Por este motivo, se han planteado diferentes modelos que intentan recoger la sobredispersión de los datos.

Por otro lado, otro de los problemas que provocan las diferencias entre los distintos paquetes de la plataforma Bioconductor, es el exceso de ceros. Para el caso de datos de conteo con exceso de ceros se utilizan modelos Cero Inflados y Hurdle, basados en la distribución de Poisson o Binomial Negativa. Estos modelos consideran que para una proporción de unidades de estudio se observan ceros, y para el resto los datos provienen de una distribución de Poisson o una Binomial Negativa. La mezcla de ambas distribuciones cuenta con la limitación de tener que presentar como mínimo algún cero y provoca que se realice inferencia de forma independiente en cada distribución. En genética, el uso de modelos Cero Inflados no es adecuado pues los investigadores no están interesados en conocer si el porcentaje de ceros difiere entre dos muestras modelando las unidades de estudio con ceros y sin ceros de forma independiente,

desean conocer si en promedio la expresión de un gen está alterada entre dos condiciones, donde la presencia de ceros indica que ese gen no se expresa.

En conclusión, no disponemos de un único modelo probabilístico para analizar datos de conteos, por lo que, a continuación, detallaremos aquellos modelos utilizados por los cuatro paquetes de Bioconductor en la plataforma R y realizaremos, posteriormente, una comparativa entre ellos.

4.4.1 EdgeR

EdgeR es un paquete del proyecto Bioconductor, diseñado para el análisis de expresión de genes (a partir de ahora usaremos el término genes aplicable, también, a tránscritos o exones), dentro de la plataforma R.

Para poder llevar a cabo nuestro análisis de datos de RNA-Seq necesitamos modelar los correspondientes datos de conteo. Como bien comentábamos anteriormente, en los modelos de conteo la variable dependiente es discreta y no negativa, asumiendo, tradicionalmente, que dicha variable sigue una distribución de Poisson. Dada que la distribución de Poisson se caracteriza por la equidispersión, es decir: sea $Y \sim Po(\lambda)$,

$$E(Y) = Var(Y) = \lambda,$$

hablamos de sobredispersión cuando ocurre que la varianza observada es superior a la media. Una solución para abordar dicho problema de sobredispersión es modelar los datos de conteo mediante una distribución Binomial Negativa. EdgeR asume que los datos de conteo siguen dicha distribución.

En el paquete EdgeR se supone que el número de reads en cada muestra j asignado a un gen i se modela a través de una distribución Binomial Negativa con dos parámetros, la media μ_{ij} y el parámetro de sobredispersión θ_{ij}:

$$Y_{ij} \sim BN(\mu_{ij}, \theta_{ij}).$$

Y_{ij} corresponde al número entero no negativo de reads en cada muestra j asignado a un gen i. Los valores de la media y la sobredispersión, en la práctica, no son conocidos por lo que debemos estimarlos a partir de los datos.

En general, se describe la distribución Binomial Negativa con dos parámetros, μ y θ. El parámetro μ se corresponde con la media y θ se corresponde con el parámetro de

sobredispersión. Dicha distribución es equivalente a la Poisson cuando $\theta = 0$. Este parámetro representa el coeficiente de variación de las correspondientes variaciones biológicas entre las muestras. Como comentábamos al principio de este trabajo, las variaciones biológicas son aquellas que se obtienen de muestras distintas que comparten una determinada característica/condición.

Para el modelo binomial negativo, la variable respuesta se distribuye según la función de densidad:

$$f(y;\mu,\theta) = \frac{\Gamma(y+\theta)}{\Gamma(\theta)\cdot y!}\cdot\frac{\mu^{y}\theta^{\theta}}{(\mu+\theta)^{y+\theta}},$$

siendo la correspondiente función de verosimilitud, a partir de la cual podemos obtener estimadores de los distintos parámetros, $\beta = (\mu,\theta)$:

$$L(\beta|y,X) = \prod_{i=1}^{N}\Pr(y_i|x_i) = \prod_{i=1}^{N}\frac{\Gamma(y+\alpha^{-1})}{y!\,\Gamma(\alpha^{-1})}\left(\frac{\alpha^{-1}}{\alpha^{-1}+\mu_i}\right)^{\alpha^{-1}}\left(\frac{\mu_i}{\alpha^{-1}+\mu_i}\right)^{y_i}, y = 0,1,2\ldots,$$

donde $\alpha = \frac{1}{\theta}$.

En conclusión, EdgeR considera que los datos de conteo siguen una distribución Binomial Negativa, ya que ésta permite analizar con un mayor ajuste datos de conteo que presentan sobredispersión, pues cuenta con el parámetro θ que contiene dicha información.

4.4.2 NOISeq

El paquete NOISeq de Bioconductor en la plataforma R permite identificar aquellos genes diferencialmente expresados mediante una aproximación no paramétrica de los datos de conteo disponibles, es decir, dicho paquete permite computar la expresión diferencial entre dos condiciones mediante la identificación del nivel de expresión de los distintos genes bajo dichas condiciones.

NOISeq analiza y calcula la correspondiente expresión diferencial de datos con réplicas técnicas y/o biológicas (NOISeq-real) o sin réplicas (NOISeq-sim). Dicho paquete presenta, además, una variante, llamada NOISeqBIO, la cual computa la expresión diferencial de datos con réplicas, únicamente, biológicas.

Para cada gen, NOISeq calcula dos estadísticos de expresión diferencial: M (el logaritmo en base 2 del ratio entre dos condiciones) y D (el valor absoluto de la diferencia entre condiciones).

Este método se basa, fundamentalmente, en la hipótesis de que cambios en la expresión de un gen entre dos condiciones no, necesariamente, deben deducir que dicho gen esté diferencialmente expresado bajo dichas condiciones experimentales. Esto ocurre cuando la magnitud de dicho cambio coincide con los cambios en la expresión de ese gen entre réplicas de una misma condición. Por ello, además de computar los cambios en la expresión de cada gen entre dos condiciones experimentales, se debe obtener, mediante la comparación de todas las réplicas dentro de la misma condición, la distribución del ruido, es decir, la distribución de los cambios en los valores de expresión de los genes cuando comparamos réplicas de la misma condición.

Se denota c_{gj}^{i} al número de reads (counts) para cada gen *i* en la *j*-ésima muestra o réplica de la condición experimental g (para NOISeq, g=1 ó 2), donde j varía de 1 hasta el número de muestras o réplicas de la condición g. Para realizar el cálculo de los valores M y D, NOISeq realiza la normalización de los counts de cada muestra o réplica por la media truncada de los valores M (trimmed mean of M values, TMM). En el 2010, Robinson *et al.* crearon dicha corrección para eliminar el ruido introducido por la variaciones biológicas naturales entre las muestras de cada condición no atribuibles a las condiciones en sí. Este método de normalización se basa en la aceptación de que la mayoría de los genes no están diferencialmente expresados. Además, antes de proceder a la normalización, los niveles de expresión iguales a 0 son reemplazados por una constante dada $k>0$, con el objetivo de evitar valores M infinitos o indeterminados. Los nuevos valores corregidos se denotan x_{gj}^{i}. Para el cálculo de los dos estadísticos de expresión diferencial, M y D, x_{g}^{i} se define, en el caso de disponer de réplicas técnicas, como la suma de los valores de todas las muestras o réplicas de cada gen i para la condición experimental g; en el caso de disponer de réplicas biológicas, como la media de todos los valores; y, en el caso de no disponer de réplicas, se define como los valores de cada gen i para la condición experimental g. Luego, para cada gen i, se calculan los estadísticos de expresión diferencial como:

$$M^{i} = \log_2\left(\frac{x_1^i}{x_2^i}\right) \; y \; D^{i} = \left|x_1^i - x_2^i\right|.$$

Como bien comentábamos anteriormente, la distribución del ruido se obtiene mediante la comparación de todas las réplicas dentro de la misma condición. Para su creación, en el caso de disponer de réplicas, se acumulan todos los valores (M, D) de todas las comparaciones entre réplicas de los distintos genes para cada condición. En el caso de no existir réplicas, éstas son simuladas y se sigue el mismo procedimiento que en el caso de disponer de réplicas.

NOISeq considera que un gen está diferencialmente expresado si sus correspondientes valores (M, D) son más altos que el ruido. La probabilidad de expresión diferencial de cada gen se obtiene a través de la comparación de los valores (M, D) de dicho gen y la distribución del ruido. Si el cociente entre la probabilidad de expresión diferencial y la probabilidad de expresión no diferencial es mayor que un determinado umbral q, es decir,

$$\frac{P(expresión\ diferencial)}{P(expresión\ no\ diferencial)} > q,$$

dicho gen se considera diferencialmente expresado entre las dos condiciones. En general, el valor de dicho umbral será q=0.8, dado que es equivalente a considerar que un gen está diferencialmente expresado cuando existe un odds ratio de 4:1, es decir, cuando podemos concluir que es 4 veces más probable que dicho gen esté expresado de forma diferencial entre ambas condiciones a que no lo esté. Por consiguiente, el valor de q será el umbral por el cual podemos considerar si hay más o menos genes diferencialmente expresados, dado que a mayor valor de q, mayor umbral y, por tanto, menos genes diferencialmente expresados entre ambas condiciones. De forma viceversa ocurre para menor valor de q: menor umbral y mayor número de genes diferencialmente expresados entre las dos condiciones.

4.4.3 DESeq2

DESeq2, uno de los cuatro paquetes de Bioconductor en los que centramos nuestro estudio, permite realizar el análisis de expresión diferencial de genes en la plataforma R, basándose, al igual que el paquete EdgeR, en la distribución Binomial Negativa. El paquete DESeq2 proporciona métodos de análisis de expresión diferencial mediante el uso de modelos de regresión binomiales negativos

Como bien comentábamos anteriormente, cuando trabajamos con datos de conteo reales, la hipótesis de igualdad de media y varianza de la distribución de Poisson no suele ser veraz. Asumir que los datos de conteo siguen una distribución Binomial Negativa soluciona el problema de la sobredispersión. En este caso, la variable respuesta, Y, presenta la siguiente función de probabilidad:

$$P(y|\theta,\mu) = \frac{\Gamma(\mathrm{y}+\theta)}{\Gamma(\theta)\Gamma(\mathrm{y}+1)}\left(\frac{\theta}{\mu+\theta}\right)^{\theta}\left(\frac{\mu}{\mu+\vartheta}\right)^{y}, y = 0,1,2\ldots,$$

donde μ y θ son los parámetros de la distribución. Se tiene que:

$$E(Y) = \mu; Var(Y) = \mu + \frac{\mu^2}{\theta}.$$

La distribución Binomial Negativa converge a una distribución de Poisson cuando el parámetro $\frac{1}{\theta} \to 0$, dado que, en este caso, $Var(Y) \to \mu$.

Para un valor fijo de θ, la distribución Binomial Negativa pertenece a la familia exponencial natural, es decir, al conjunto de distribuciones cuya formulación se expresa de la siguiente manera, siendo Y la variable aleatoria y k el parámetro de la correspondiente distribución:

$$f(y_i|k_i) = a(k_i) \cdot b(y_i) \cdot \exp[y_i Q(k_i)],$$

donde $Q(k)$recibe el nombre de parámetro natural; de modo que se puede definir un modelo lineal generalizado (GLM) binomial negativo. Una de las componentes del modelo GLM es la función enlace o función link, la cual es una función del valor esperado de la variable respuesta $Y, E(y)$, como una combinación lineal de las s variables independientes (explicativas o predictoras). La combinación lineal de estas variables,

$$\alpha + \beta_1 x_1 + \cdots + \beta_s x_s,$$

se denomina predictor lineal. En general, para datos de conteo, la función de enlace que se usa habitualmente en el modelo lineal generalizado es el logaritmo de la media, de forma que, considerando una única variable explicativa, se expresa el modelo log-lineal como:

$$\log(\mu) = \alpha + \beta x,$$

de tal forma que

$$\mu = \exp[\alpha + \beta x] = e^{\alpha}(e^{\beta})^x.$$

4.4.4 Limma

El uso de Microarrays es una técnica que permite medir datos de expresión génica y comparar la abundancia relativa de ARN mensajero generado bajo distintas condiciones biológicas. Un Microarray es un soporte sólido construido normalmente en cristal o en membrana de nylon. A pesar de ser la tecnología más usada y de más fácil acceso, actualmente RNA-Seq es generalmente reconocido como mejor método para el análisis del transcriptoma, debido a que, entre otras razones, al contrario que los Microarrays que se limitan a identificar tránscritos que corresponden a secuencias genéticas ya conocidas, RNA-Seq puede determinar secuencias aún

desconocidas; y, otra diferencia a considerar, es que los Microarrays son soportes sólidos, mientras que la tecnología RNA-Seq utiliza lecturas digitales, simplificando el estudio.

Aunque RNA-Seq y los Microarrays son usados con el mismo propósito, los métodos estadísticos para detectar la expresión diferencial en datos procedentes de Microarrays, no son fácilmente aplicables al análisis de datos de RNA-Seq. Esto es debido a que los datos obtenidos mediante los Microarrays constituyen una matriz en la cual obtendremos celdas de diferentes colores en función del nivel de expresión de los genes, mientras que en RNA-Seq dispondremos de datos de conteo procedentes del número de reads mapeadas en cada gen.

Limma es un paquete de Bioconductor en la plataforma R que permite realizar el análisis de expresión diferencial de genes para datos de Microarrays. Como bien se ha comentado anteriormente, los métodos estadísticos para dicho estudio no suelen trasladarse a datos de RNA-Seq. Sin embargo, dicho paquete ha sido adaptado al análisis de expresión diferencial de datos de RNA-Seq. No se dispone de información suficiente acerca de su eficacia ni de su metodología. No obstante, podemos señalar que el paquete Limma se basa en el uso de modelos lineales para realizar dicho análisis.

5. ANÁLISIS DE DATOS REALES

Hemos visto de forma detallada los pasos a seguir en el análisis de datos de RNA-Seq. A continuación, nos dispondremos a realizar un estudio basado en datos reales que nos permitirá llevar a cabo el objetivo principal de nuestro trabajo, obteniendo posteriormente en las conclusiones la comparación entre los distintos paquetes de Bioconductor para el análisis de la expresión génica diferencial.

El asma es una enfermedad crónica del sistema respiratorio causada por la inflamación de las vías respiratorias que afecta a más de 300 millones de personas en todo el mundo. Los glucocorticoides son fármacos antiinflamatorios usados para el tratamiento de muchas afecciones respiratorias. Un potente glucocorticoide es la dexametasona. Para nuestro estudio disponemos de datos obtenidos mediante RNA-Seq referentes a 8 pacientes que padecen asma, de los cuales 4 han sido tratados con dicho glucocorticoide y el resto no han recibido ningún tipo de tratamiento. Contamos con una serie de ficheros .fastq, los cuales recogen las reads obtenidas mediante la secuenciación de cada uno de los pacientes. Como bien comentábamos anteriormente, los ficheros .fastq incluyen, principalmente, para cada read, un identificador, su correspondiente tamaño, la calidad para cada nucleótido y la secuencia de nucleótidos de dicha read. Dichos datos están disponibles en:

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE52778.

Cada archivo .fastq presenta un nombre diferente para cada paciente. En la siguiente tabla se muestra a qué paciente corresponde cada archivo:

Nombre del archivo	Paciente
SRR1039508	Sin tratamiento 1
SRR1039509	Dexametasona 1
SRR1039512	Sin tratamiento 2
SRR1039513	Dexametasona 2
SRR1039516	Sin tratamiento 3
SRR1039517	Dexametasona 3
SRR1039520	Sin tratamiento 4
SRR1039521	Dexametasona 4

Tabla 4. Nombre del archivo - Paciente

En primer lugar, nos disponemos a realizar el análisis de calidad de las reads. Para ello, como comentábamos en el apartado del estudio de calidad, utilizamos el programa FastQC. Dicho programa proporcionará para cada archivo .fastq una serie de comprobaciones de calidad, entre las cuales destacaremos la tabla de datos estadísticos básicos (Basic Statistics), haciendo especial énfasis en la longitud de dichas reads y el porcentaje de GC global, y el gráfico de la calidad por base de dichas secuencias (Per Base Sequence Quality). A continuación se muestran los resultados obtenidos para cada archivo .fastq correspondiente a la secuenciación de cada uno de los pacientes.

Measure	Value
Filename	SRR1039508_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	22935521
Filtered Sequences	0
Sequence length	63
%GC	50

Tabla 5. Paciente sin tratamiento 1

Measure	Value
Filename	SRR1039509_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	21155707
Filtered Sequences	0
Sequence length	63
%GC	49

Tabla 6. Paciente dexametasona 1

Measure	Value
Filename	SRR1039512_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	28136282
Filtered Sequences	0
Sequence length	63
%GC	49

Tabla 7. Paciente sin tratamiento 2

Measure	Value
Filename	SRR1039513_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	16823088
Filtered Sequences	0
Sequence length	63
%GC	48

Tabla 8. Paciente dexametasona 2

Measure	Value
Filename	SRR1039516_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	27298970
Filtered Sequences	0
Sequence length	63
%GC	49

Tabla 9. Paciente sin tratamiento 3

Measure	Value
Filename	SRR1039517_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	34298260
Filtered Sequences	0
Sequence length	63
%GC	49

Tabla 10. Paciente dexametasona 3

Measure	Value
Filename	SRR1039520_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	21275888
Filtered Sequences	0
Sequence length	63
%GC	49

Tabla 11. Paciente sin tratamiento 4

Measure	Value
Filename	SRR1039521_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	23487860
Filtered Sequences	0
Sequence length	63
%GC	49

Tabla 12. Paciente dexametasona 4

Podemos observar que la longitud de las reads para cada archivo .fastq es de 63 nucleótidos. Recordar que para evitar dificultades en el resto de comprobaciones de calidad, era aconsejable disponer de secuencias con la misma longitud. Además, los porcentajes de GC global están entre el 48% y el 50%. Dichos porcentajes se consideran aceptables pues superan el 45%.

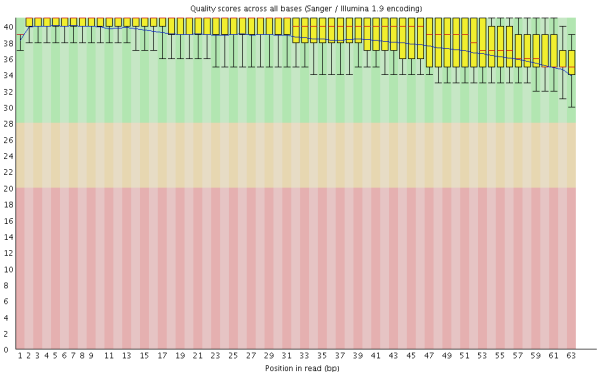


Figura 13. Paciente sin tratamiento 1

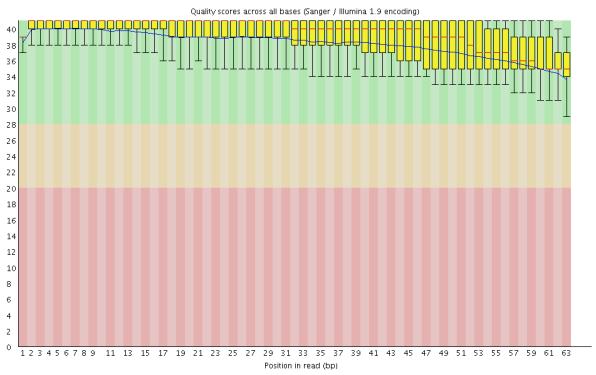


Figura 14. Paciente dexametasona 1

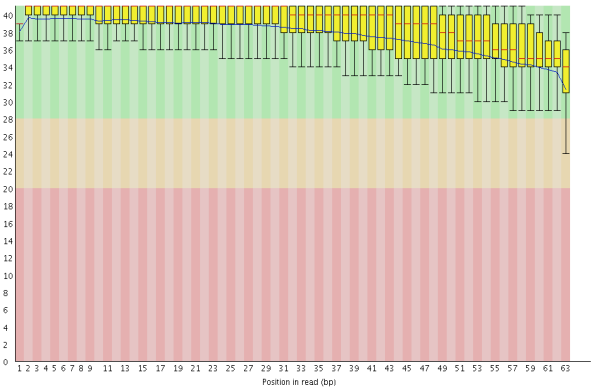


Figura 15. Paciente sin tratamiento 2

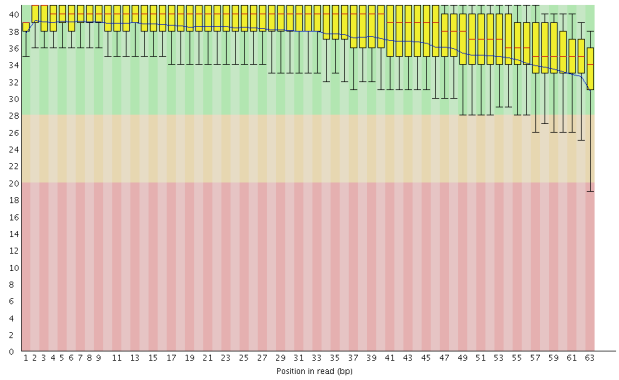


Figura 16. Paciente dexametasona 2

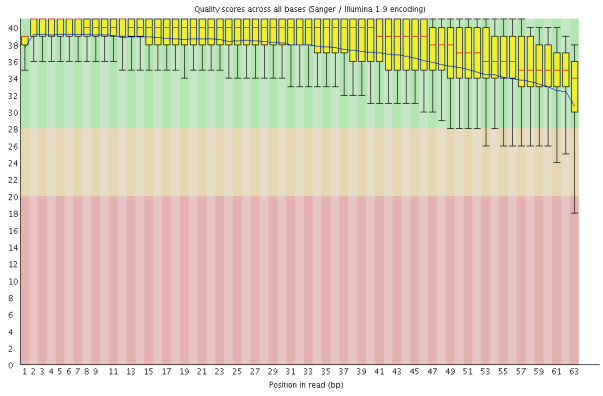


Figura 17. Paciente sin tratamiento 3

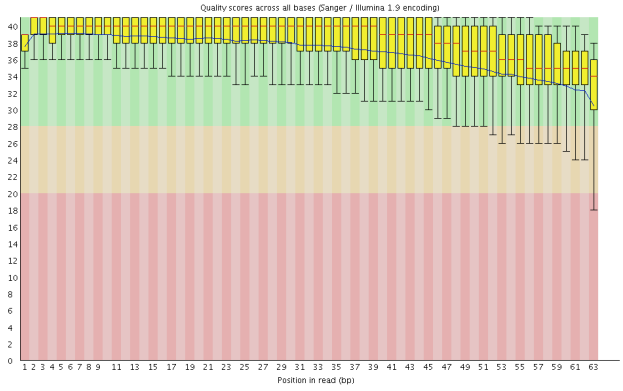


Figura 18. Paciente dexametasona 3

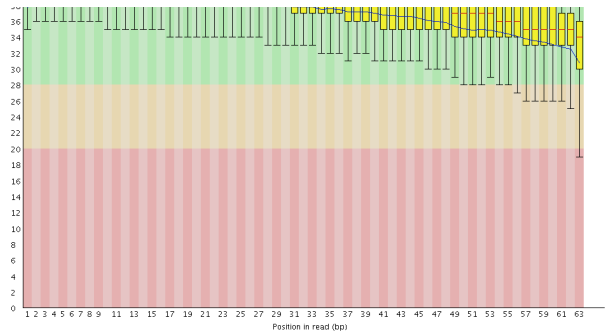


Figura 19. Paciente sin tratamiento 4

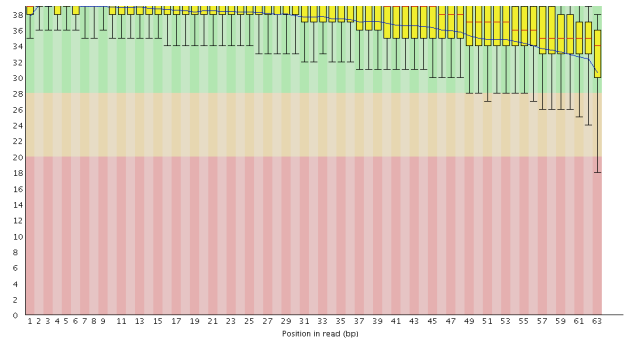


Figura 20. Paciente dexametasona 4

Dichos gráficos nos permiten observar la calidad por base (nucleótido) de las reads de cada archivo .fastq. Para ninguno de dichos archivos el programa FastQC muestra señal de advertencia o fracaso. Como se puede observar, esto es debido a que las bases presentan altos niveles de calidad, claramente identificable en el hecho de que el cuantil más pequeño de cada Box-Plot para cada base se encuentra dentro de la zona de muy buena calidad, en color verde. Además, las medianas, correspondientes a las líneas rojas, no toman valores inferiores a 25. Podemos ver que a medida que avanzamos en las posiciones (bases) de las reads, dicha calidad disminuye ligeramente. Como comentábamos, esto es debido a que cuando la tecnología de secuenciación va incorporando nucleótidos a la read, aumenta la posibilidad de cometer errores.

El siguiente paso en dicho estudio es el alineamiento de las reads frente al genoma de referencia. Para ello, haremos uso de la herramienta TopHat, la cual permite realizar el mapeo de dichas reads por tránscritos. El genoma de referencia es el GRChr37, obtenido a través del proyecto Illumina's iGenomes que recoge una colección de diversas secuencias de referencia. El alineamiento de las reads se realiza usando Bowtie2. TopHat, además, intenta alinear, mediante un algoritmo más preciso, aquellas reads que no se han podido mapear a alguna secuencia del genoma de referencia

Mediante el uso de TopHat, obtenemos para cada archivo .fastq un archivo de extensión .bam. En estos archivos de texto se encuentran los datos del alineamiento presentando la siguiente estructura, por filas, para cada read: el identificador de dicha read; su posición en el genoma de referencia, especificando el cromosoma en el que se encuentra y el punto inicial y final del lugar que ocupa dentro de dicho genoma, lo que permite, a su vez, identificar el gen y el tránscrito, o tránscritos, correspondientes a dicha posición; la secuencia de nucleótidos de dicha read y su correspondiente calidad. La siguiente imagen muestra parte del archivo .bam obtenido a partir del archivo SRR1039508.fastq, que se corresponde con la secuenciación realizada del paciente sin tratamiento 1:

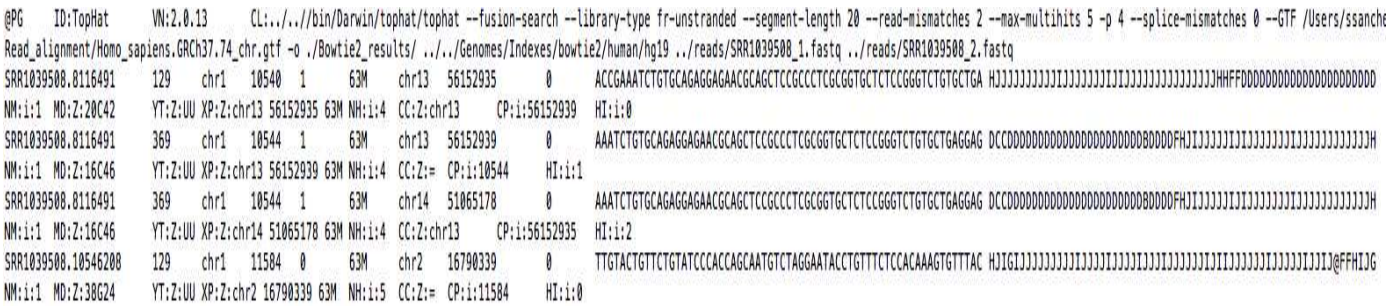
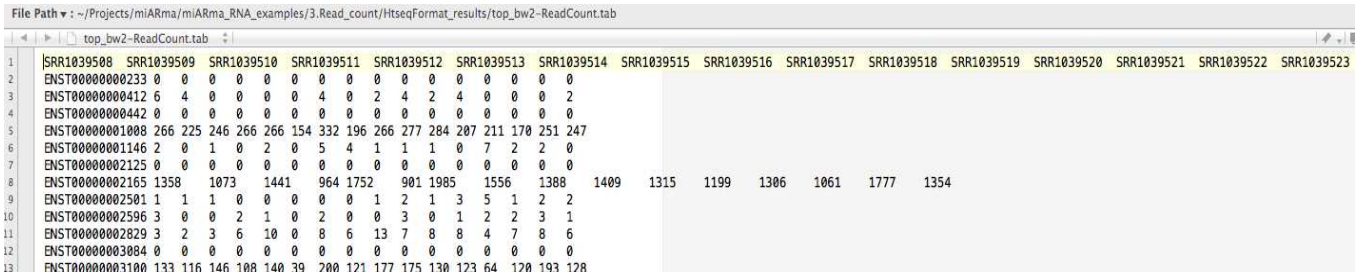


Figura 21. Parte del archivo extensión .bam del paciente sin tratamiento 1

Una vez llevado a cabo el alineamiento, nos disponemos a contar el número de reads por tránscrito para cada paciente. Para realizar dicha tarea empleamos el software HTSeq-count, el cual genera un archivo .counts a partir del archivo .bam de cada paciente, por lo que

obtendremos, finalmente, 8 archivos de extensión .counts. En dichos archivos se cuantifica el número de reads alineadas en relación a cada tránscrito del genoma de referencia. Los archivos .counts de los distintos pacientes se agrupan formando un único archivo .tab que recoge el conteo de reads para cada paciente en cada tránscrito, es decir, dicho archivo .tab presenta la siguiente estructura: por fila, los tránscritos sobre los cuales han sido alineadas las reads, y por columna, los distintos pacientes sobre los que estamos realizando el correspondiente estudio. A continuación, se muestra parte de la matriz de conteo, integrada en nuestro archivo .tab, que permite contabilizar el nivel de expresión de los distintos tránscritos para cada paciente:


File Path ▾ : ~/Projects/miARma/miARma_RNA_examples/3.Read_count/HtseqFormat_results/top_bw2-ReadCount.tab

top_bw2-ReadCount.tab

	SRR1039508	SRR1039509	SRR1039510	SRR1039511	SRR1039512	SRR1039513	SRR1039514	SRR1039515	SRR1039516	SRR1039517	SRR1039518	SRR1039519	SRR1039520	SRR1039521	SRR1039522	SRR1039523
ENST00000000233	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ENST00000000412	6	4	0	0	0	0	4	0	2	4	2	4	0	0	0	2
ENST00000000442	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENST00000001008	266	225	246	266	266	154	332	196	266	277	284	207	211	170	251	247
ENST00000001146	2	0	1	0	2	0	5	4	1	1	1	0	7	2	2	0
ENST00000002125	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENST00000002165	1358		1073		1441		964	1752		901	1985		1556		1388	
ENST00000002501	1	1	1	0	0	0	0	0	1	2	1	3	5	1	2	2
ENST00000002596	3	0	0	2	1	0	2	0	0	3	0	1	2	2	3	1
ENST00000002829	3	2	3	6	10	0	8	6	13	7	8	8	4	7	8	6
ENST00000003084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENST00000003100	133	116	146	108	140	39	200	121	177	175	130	123	64	120	193	128

	1409	1315	1199	1306	1061	1777	1354

Figura 22. Parte del archivo extensión .tab

El último paso es el análisis de expresión génica diferencial, el objetivo principal del estudio de datos de RNA-Seq. Para ello, como bien comentábamos anteriormente, utilizaremos cuatro paquetes de Bioconductor en la plataforma R, obteniendo para cada uno de ellos el número de tránscritos diferencialmente expresados e identificando entre dichos tránscritos los 15 que presentan mayor diferencia de expresión entre las dos condiciones consideradas: pacientes sin tratamiento y con tratamiento. Posteriormente, basándonos en los resultados obtenidos y en el contenido estadístico de cada paquete (métodos de normalización, modelos probabilísticos…), realizamos una comparación entre dichos paquetes.

- **EdgeR:**

A continuación, para llevar a cabo el análisis de expresión génica diferencial haremos uso del paquete EdgeR de Bioconductor en la plataforma R, el cual, como hemos visto anteriormente, se basa en el supuesto de que los datos de conteo se distribuyen según una Binomial Negativa. Realizaremos un resumen de los distintos pasos a seguir para realizar el estudio a partir de dicho paquete, señalando algunas funciones de R fundamentales para su ejecución. El código R completo para el análisis de expresión génica diferencial haciendo uso del paquete EdgeR de Bioconductor se encuentra en el Anexo de este trabajo.

En primer lugar, para poder hacer uso de dicho paquete en R necesitamos ejecutar los siguientes comandos para su instalación:

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

> biocLite("edgeR")

> library(edgeR)

Una vez hemos importado los datos de conteo del archivo .tab, obteniendo un objeto llamado *reads*, a través de la función *read.table*, y hemos creado el objeto *group*, factor con dos niveles: Untreated (sin tratamiento) y Dex (con dexametasona), generamos el objeto *y* de clase *DGElist*, el cual almacena la matriz de conteos y un data.frame con información de cada paciente, principalmente sobre la condición (grupo) a la que pertenece y el número de reads alineadas en el total de tránscritos.

> y <- DGEList(counts=reads,group=group)

Para obtener una primera visión general de la relación entre los datos disponibles de cada paciente, realizamos un dendograma, utilizando las funciones *plot* y *hclus*t:

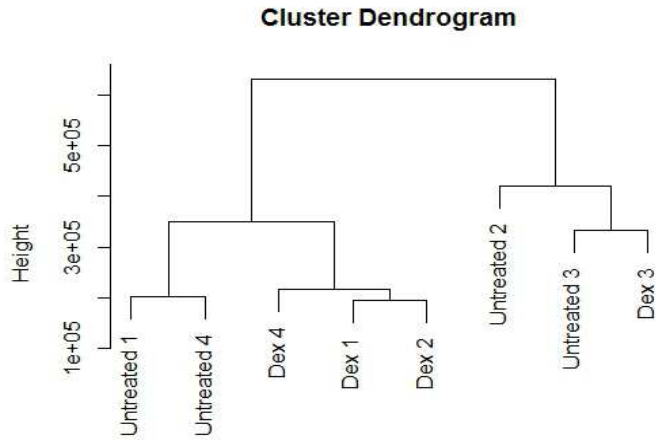


Figura 23. Dendograma

Además del dendograma, podemos utilizar la función *plotMDS*, que utiliza una técnica similar al análisis de componentes principales.

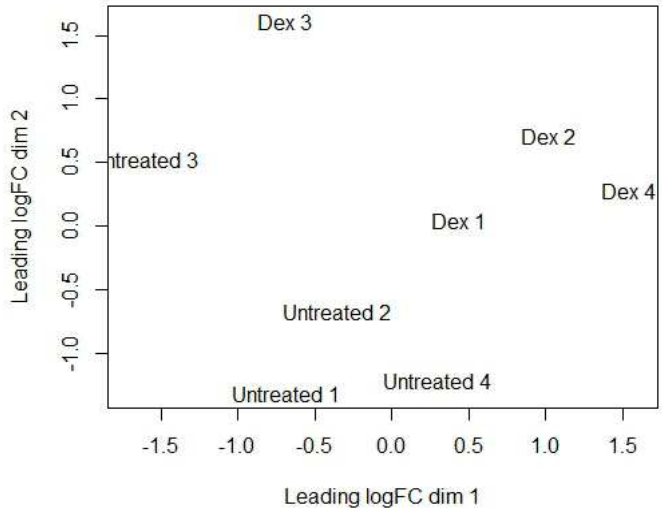


Figura 24. *plotMDS*, similar al análisis de componentes principales

A continuación se procede a la normalización de los datos mediante la función *calcNormFactors*, la cual proporciona un conjunto de factores de normalización que minimizan el *log-fold-change* (cambio en la proporción de reads) entre los pacientes, según el total de reads de cada uno de ellos, para la mayoría de tránscritos. El cálculo de dichos factores de normalización se realiza por la media truncada de los valores M (trimmed mean of M values, TMM) entre cada par de pacientes. Obtenemos factores de normalización cercanos a 1, lo cual nos indica que no existen grandes diferencias entre la composición de los distintos pacientes.

> ynorm<- calcNormFactors(y)

> ynorm$samples

	group	*lib.size*	*norm.factors*
Untreated 1	*Untreated*	*8081189*	*1.0821788*
Dex 1	*Dex*	*7519093*	*1.0348602*
Untreated 2	*Untreated*	*10257728*	*0.9896933*
Dex 2	*Dex*	*6485406*	*0.9807291*
Untreated 3	*Untreated*	*10139822*	*0.9998334*
Dex 3	*Dex*	*12345535*	*0.9641628*
Untreated 4	*Untreated*	*7719875*	*1.0316173*
Dex 4	*Dex*	*8655242*	*0.9250665*

Por otro lado, vamos a filtrar aquellos tránscritos que tienen un número muy bajo de reads, para evitar posibles problemas en el posterior uso de las funciones logarítmicas del paquete EdgeR. El método empleado para ello es el CPM. Dicho método elimina aquellos tránscritos que presentan un número de reads inferior a un determinado número de reads por millón (cpm), es decir, dependiendo del total de reads en cada paciente, dicho límite corresponde a un número

distinto de reads. En nuestro caso, vamos a poner dicho límite en 5 reads por millón. Nos quedaremos con aquellos tránscritos para los que, en al menos dos pacientes, el número de reads alineadas a dicho tránscrito superen dicho límite.

```
> keep <- rowSums(cpm(ynorm)>5) >= 2

> y <- ynorm[keep, ]

> table(keep)

keep
 FALSE   TRUE
206443   8727
```

Finalmente, nos quedamos con 8.727 tránscritos, 206.443 menos de los que disponíamos inicialmente.

Como hemos visto anteriormente, EdgeR se basa en el supuesto de los datos de conteo se modelan mediante una Binomial Negativa, por lo que antes de proceder a realizar el correspondiente test para determinar los tránscritos diferencialmente expresados, debemos estimar el parámetro de dispersión. En primer lugar, estimamos la dispersión común para todos los tránscritos a través de la función *estimateCommonDisp,* es decir, dicho parámetro proporciona una idea general de la variabilidad de los datos. Esta función calcula el coeficiente de variación biológica (BCV), que constituye la raíz cuadrada de dicho parámetro de dispersión y representa el coeficiente de variación entre réplicas de la misma condición. Posteriormente, estimamos la dispersión para cada uno de los tránscritos mediante la función e*stimateTagwiseDisp*.

La función *plotBCV* permite representar gráficamente la raíz cuadrada de los parámetros de dispersión calculados anteriormente (BCVs) respecto al log_2 de reads por millón (logCPM).

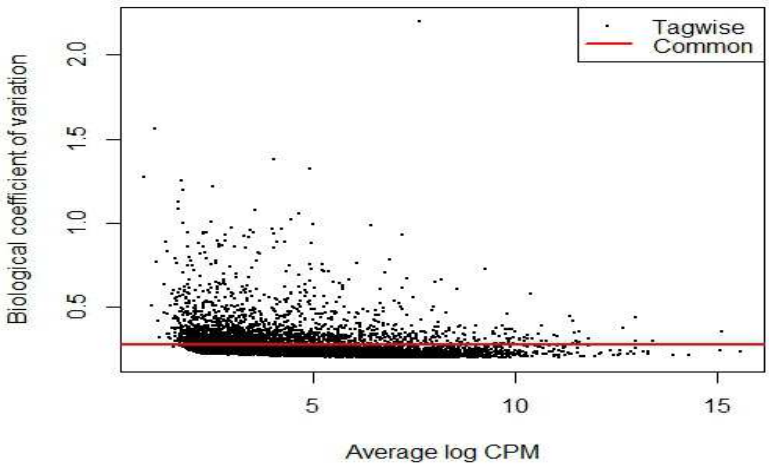


Figura 25. *plotBCV,* raíz cuadrado de los BCVs respecto al logCMP

Gráficamente podemos apreciar que los BCVs son superiores a la dispersión común a bajas concentraciones de reads.

Una vez estimada la dispersión, procedemos a realizar los correspondientes test para determinar la expresión diferencial de tránscritos. Para ello, usaremos el test exacto basado en la distribución Binomial Negativa (test exacto binomial negativo), propuesto por Robinson y Smyth, el cual presenta grandes similitudes con el test exacto de Fisher, a través de la función *exactTest.* La función *topTags* nos permite identificar los *n* tránscritos más diferencialmente expresados:

```
> et <- exactTest(y,,pair=c("Dex","Untreated"))
> topTags(et,n=15)
Comparison of groups:  Untreated-Dex
                    logFC     logCPM       PValue          FDR
ENST00000262878 -3.839657 10.795086 1.131113e-29 9.871225e-26
ENST00000338702 -3.352720  7.394273 7.376986e-29 3.218948e-25
ENST00000274711  4.071636  2.752961 2.812146e-28 8.180533e-25
ENST00000239223 -2.923648  9.556158 6.150308e-28 1.341843e-24
ENST00000377474  2.535384  7.899155 8.938970e-24 1.560208e-20
ENST00000296233 -4.549002  6.750725 3.601536e-23 5.238435e-20
ENST00000374426  3.651896  3.283437 6.746927e-22 8.411490e-19
ENST00000377482 -2.449120  8.510076 6.462653e-20 7.049946e-17
ENST00000542713 -3.475013  3.259211 2.017440e-19 1.956245e-16
ENST00000262424 -2.734678  8.411289 3.667001e-19 3.200192e-16
ENST00000287814 -2.865269  5.232512 5.379590e-19 4.267971e-16
ENST00000427716 -2.316656  7.003612 7.936262e-19 5.771647e-16
ENST00000486554 -5.395703  1.766072 1.465476e-18 9.360324e-16
ENST00000491322 -3.426138  4.921424 1.501599e-18 9.360324e-16
ENST00000379706  3.424798  1.995338 3.432928e-18 1.997277e-15
```

Dicha matriz presenta el nombre de los 15 tránscritos más diferencialmente expresados entre los pacientes sin tratamiento y con tratamiento, ordenados en función del p-valor obtenido para cada test realizado en dichos tránscritos. La columna FDR proporciona la probabilidad de error de tipo I, la cual debe ser inferior a 0’05. Por otro lado, la columna logFC, correspondiente al log_2*-fold-change*, muestra el cambio en la proporción de reads para ambas condiciones en función del log_2. Los tránscritos para los cuales el valor logFC es negativo, constituyen aquellos tránscritos que se expresan más en pacientes con tratamiento. Ocurriendo de forma contraria para aquellos tránscritos cuyo valor logFC es positivo, pues determina que dichos tránscritos se expresan más en aquellos pacientes sin tratamiento.

Obtenemos que el número de tránscritos diferencialmente expresados entre ambas condiciones es de 853, para un nivel de significación α =0'05, proporcionando dicho dato los siguientes comandos:

> summary(de <- decideTestsDGE(et, p=0.05, adjust="BH"))

> detags <- rownames(y)[as.logical(de)]

> length(detags)

853

Finalmente, a través de la función *plotSmear*, podemos obtener una representación gráfica de los distintos logFC en relación con el logCPM, el log_2 de reads por millón, de cada tránscrito, resaltando en color rojo los puntos correspondientes a aquellos tránscritos diferencialmente expresados.

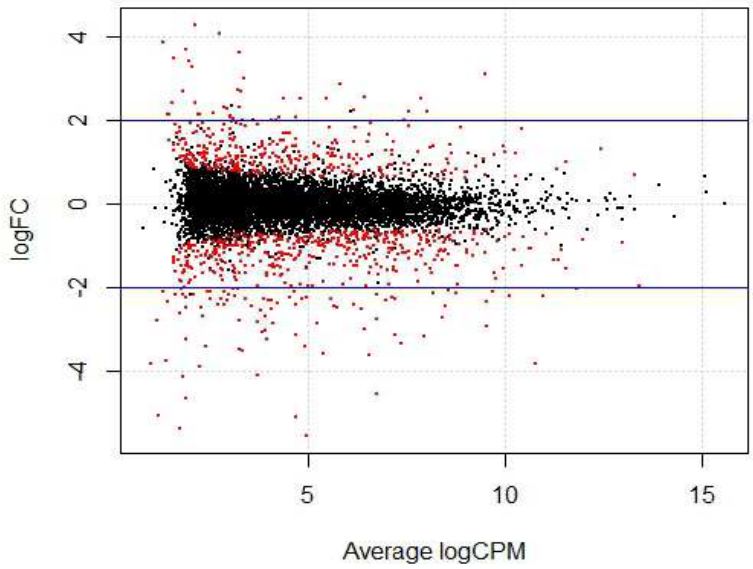


Figura 26. *plotSmear,* logFC en relación con el logCPM de cada tránscrito

Las líneas azules representan el logFC 2. Dicho logFC respresenta un cambio en la proporción de reads entre condiciones equivalente a 4, es decir, el tránscrito al que le corresponde dicho logFC se expresa cuatro veces más en una de las dos condiciones.

- **NOISeq:**

El paquete NOISeq de Bioconductor en la plataforma R nos va a permitir identificar aquellos tránscritos diferencialmente expresados entre las dos condiciones consideradas (con tratamiento y sin tratamiento), a través de aproximaciones no paramétricas de los datos de conteo.

Al igual que para el resto de paquetes a estudiar, necesitamos instalar el paquete NOISeq a través de los siguientes comandos:

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

> biocLite("NOISeq")

> library(NOISeq)

Recordar que el código R completo para el análisis de expresión génica diferencial haciendo uso del paquete NOISeq de Bioconductor se encuentra disponible en el Anexo de este trabajo.

Una vez importados los datos de conteo del archivo .tab, almacenados en el objeto *reads*, a través de la función *read.table*, y creado el data.frame *factores*, que incluye el factor *Tratamiento* el cual tiene dos niveles: Untreated (sin tratamiento) y Dex (con dexametasona), los cuales aparecen ordenados en función de la condición a la que pertenece cada paciente (la ordenación de los pacientes se observa en relación a las posiciones que ocupan en las columnas del archivo .tab, referidas en la *tabla 4*), se lleva a cabo, en primer lugar, el filtro de tránscritos con bajo número de reads. NOISeq incluye tres métodos, igualmente válidos, para realizar dicho procedimiento. Haremos uso del método CPM (método 1), tomando cpm=5, es decir, estableciendo en 5 reads por millón el número mínimo de reads que debe presentar dicho tránscrito en un determinado paciente. El límite para una condición con s réplicas es $cpm \times s$. Los tránscritos cuya suma de reads sea inferior a dicho límite en ambas condiciones, serán eliminados. NOISeq incluye la función *filtered.data*, la cual realiza el filtro de dichos tránscritos, como se muestra en la siguiente instrucción:

> myfilt <- filtered.data(reads, factor = factores$Treatment, norm = FALSE, depth = 2, method = 1, cv.cutoff = 100, cpm = 5)

Además, el argumento cv.cutoff permite limitar el coeficiente de variación (en porcentaje) por condición, permitiendo depurar aquellos tránscritos que presentan valores incoherentes entre sí.

Tras ejecutar dicha función, nos quedamos con 8.095 tránscritos, de los 215.170 que disponíamos inicialmente.

Con los datos filtrados, para poder trabajar con las distintas funciones de NOISeq debemos crear un objeto *y,* a través de la función *readData,*que contendrá información relativa a los tránscritos, los pacientes y las condiciones.

> y <- readData(data=myfilt,factors=factores)

Posteriormente, procedemos a analizar la expresión diferencial de tránscritos. Es conveniente indicar que la normalización se realiza dentro de la función *noiseq* utilizada para el correspondiente estudio de la expresión diferencial, a través del argumento norm. La técnica implementada para llevar a cabo dicha normalización será RPKM (reads por kilobase por millón de reads mapeadas). RPKM se define como el número de reads de cada tránscrito sobre el total de reads de la muestra (en millones) multiplicado por la longitud del tránscrito (en kilobases). La función *noiseq* nos permite obtener los valores de los dos estadísticos de expresión diferencial para cada tránscrito: M (el logaritmo en base 2 del ratio entre dos condiciones) y D (el valor absoluto de la diferencia entre condiciones). Debemos indicar en dicha función, en el argumento replicates, que disponemos de réplicas biológicas. NOISeq calculará la media de reads de dichas réplicas para cada condición en cada tránscrito.

```
> mynoiseq <- noiseq(y, k = 0.5, norm = "rpkm", factor = "Treatment", pnr = 0.2, nss = 5, v = 0.02, lc = 1, replicates = "biological")
```

```
> head(mynoiseq@results[[1]])
```

	Dex_mean	Untreated_mean	M	D	prob	ranking
ENST00000002165	157.2324	197.25387	-0.3271550	40.02146	0.4852219	-40.02280
ENST00000003912	500.8225	570.38556	-0.1876379	69.56301	0.4180538	-69.56326
ENST00000005257	137.7097	222.21060	-0.6902974	84.50090	0.7231114	-84.50371
ENST00000011619	222.1797	144.11648	0.6244915	78.06319	0.6987303	78.06569
ENST00000013222	164.6035	76.85421	1.0987990	87.74931	0.7463722	87.75619
ENST00000014930	122.8187	98.77616	0.3142950	24.04250	0.3547802	24.04456

La columna prob proporciona la estimación de la probabilidad de expresión diferencial de cada tránscrito. Una vez obtenidas dichas estimaciones, queremos conocer aquellos tránscritos para los cuales el cociente entre su probabilidad de expresión diferencial y su probabilidad de no expresión diferencial es mayor que un umbral *q*. En este caso, dichos tránscritos se considerarán diferencialmente expresados entre ambas condiciones. Para ello, haremos uso de la función *degenes*. Dicha función presenta un argumento M, que nos permite especificar si queremos conocer todos los tránscritos diferencialmente expresados (NULL) o sólo aquellos que son más expresados en la condición 1 que en la condición 2 (M=”up”) o aquellos que se expresan más en la condición 2 que en la condición 1 (M=”down”). La condición 1 hace referencia a la condición para la cual aparece la media de reads de sus réplicas biológicas en la primera columna de la salida de la función *head(mynoiseq@results[[1]])*. La segunda columna de dicha salida correspondería a la condición 2. Por otro lado, cuando se usa NOISeq con réplicas, el manual de uso de Bioconductor recomienda que el valor de *q* esté en torno a 8. A continuación, se presentan los comandos que nos permiten obtener el número de tránscritos diferencialmente expresados (415) e identificar aquellos 15 con mayor nivel de expresión diferencial.

> mynoiseq.deg <- degenes(mynoiseq, q = 0.8, M = NULL)

[1] "415 differentially expressed features"

> mynoiseq.deg1 <- degenes(mynoiseq, q = 0.8, M = "up")

[1] "236 differentially expressed features (up in first condition)"

> mynoiseq.deg2 <- degenes(mynoiseq, q = 0.8, M = "down")

[1] "179 differentially expressed features (down in first condition)"

> head(mynoiseq.deg,n=15)

	Dex_mean	Untreated_mean	M	D	prob	ranking
ENST00000262878	3308.96184	245.37003	3.753348	3063.5918	0.9967573	3063.5941
ENST00000374429	149.42906	1372.70791	-3.199492	1223.2789	0.9939469	-1223.2830
ENST00000239223	1330.09032	184.51294	2.849730	1145.5774	0.9923410	1145.5809
ENST00000295927	3333.98506	767.30248	2.119380	2566.6826	0.9877085	2566.6835
ENST00000256637	1242.28302	258.7312	2.263468	983.5518	0.9874717	983.5544
ENST00000313164	447.43299	51.65963	3.114562	395.7734	0.9873996	395.7856
ENST00000262424	596.42215	93.92159	2.666805	502.5006	0.9871629	502.5076
ENST00000284984	1821.43750	412.31211	2.143269	1409.1254	0.9870805	1409.1270
ENST00000421865	5908.49641	1516.01453	1.962507	4392.4819	0.9858246	4392.4823
ENST00000260356	17486.20585	4696.60906	1.896526	12789.5968	0.9849599	12789.5969
ENST00000377482	614.00805	118.72546	2.370628	495.2826	0.9846922	495.2883
ENST00000284987	1367.79570	331.92196	2.042937	1035.8737	0.9845378	1035.8757
ENST00000377474	70.72023	430.54940	-2.605984	359.8292	0.9839922	-359.8386
ENST00000338702	307.80720	31.45960	3.290455	276.3476	0.9838481	276.3672
ENST00000343575	613.68858	2285.94223	-1.897210	1672.2537	0.9834260	-1672.2547

Puede resultar de interés observar gráficamente la media de reads de las réplicas biológica de cada condición en cada tránscrito, resaltando aquellos que han sido considerados diferencialmente expresados. Para ello, hacemos uso de la función *DE.plot*.

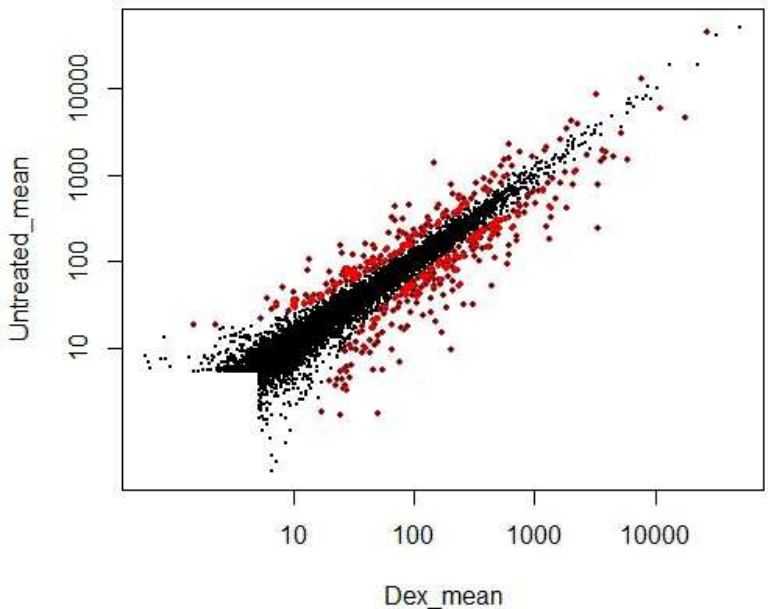


Figura 27. *DE.plot,* media de reads por condición

Dicha función nos permite, además, representar los pares de valores (M, D) de cada tránscrito, resaltando, en color rojo, aquellos pares que corresponden a los tránscritos diferencialmente expresados.

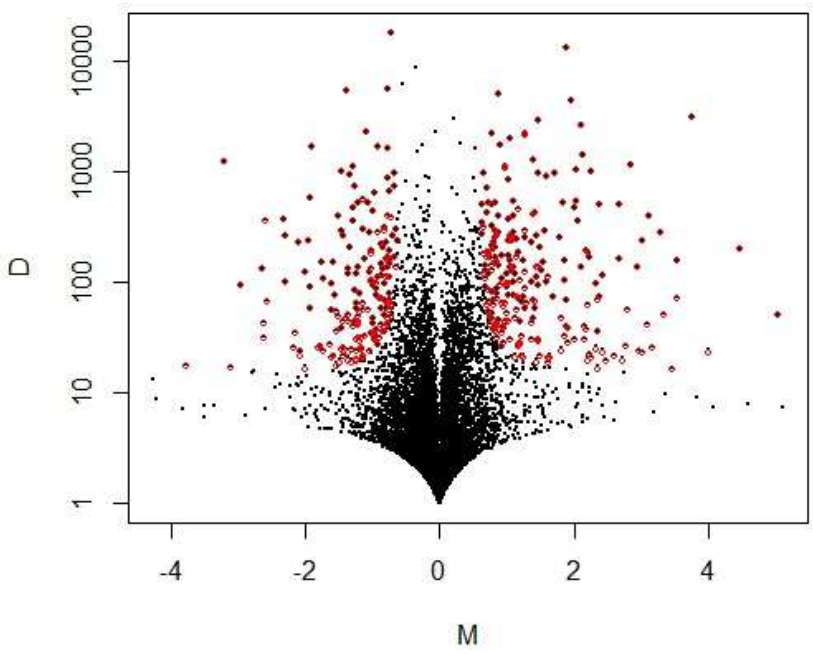


Figura 28. *DE.plot,* pares de valores *(M,D)* de cada tránscrito

- **DESeq2.**

En este punto, llevaremos a cabo el análisis de expresión diferencial de tránscritos por medio del paquete DESeq2 de Bioconductor en la plataforma R, el cual se basa, como bien hemos comentado anteriormente, en el uso del modelo de regresión binomial negativo. En primer lugar, procedemos a instalar dicho paquete a través de los siguientes comandos de R:

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

> biocLite("DESeq2")

> library(DESeq2)

Una vez hemos almacenado los datos de conteo en el objeto *reads*, importados del archivo .tab, y creado el data.frame *treatment*, formado por el factor *Treatment*, que presenta dos niveles: Untreated y Dex, los cuales aparecen ordenados en función de la condición a la que pertenece cada paciente (ordenación de los pacientes en la *tabla 4*), creamos el objeto *y* a través de la función *DESeqDataSetFromMatrix*, añadiendo como argumentos: la matriz de conteo, *reads;* el data.frame *treatment*, coincidiendo el nombre de sus filas con el nombre de las

columnas del objeto *read;* y una fórmula que defina la agrupación de los pacientes en función de las dos condiciones consideradas en nuestro estudio, especificadas en el factor *Treatment*, *design=~Treatment.*

Cuando obtengamos, finalmente, el log_2*-fold-change*, logFC, nos mostrará el cambio en la proporción de reads, en función del log_2, para las dos condiciones consideradas en el factor *Treatment*, el cual presenta dichos niveles ordenados alfabéticamente. La primera de las condiciones consideradas en dicho factor es Dex, con tratamiento, y se corresponde con el nivel de referencia. Por defecto, el valor del logFC será la comparación de una condición con dicho nivel de referencia, $log_2(condición/nivel\ de\ referencia)$, por lo que puede resultar de interés cambiar este nivel. Esto ocurre principalmente cuando tenemos un mayor número de condiciones, para las cuales, en general, es interesante comparar pacientes con diversos tratamientos frente a un paciente de control o sin tratamiento. En nuestro estudio, vamos a considerar como nivel de referencia la condición Untreated, sin tratamiento. Para realizar dicho cambio haremos uso de la función *relevel.*

El paquete DESeq2, una vez generado el objeto *y*, realiza el análisis de expresión diferencial de tránscritos a través de la función *DESeq*. Pasos como la normalización de los datos (a través del cálculo de los factores de normalización), la estimación de la dispersión (dado que DESeq2 se basa en el modelo lineal generalizado binomial negativo) y los correspondientes test para determinar los tránscritos diferencialmente expresados, son llevados a cabo en dicha función. La tabla de resultados se obtiene a través de la función *results*. A continuación, se muestran los 15 tránscritos más diferencialmente expresados en nuestro estudio, ordenados en función de los p-valores ajustados.

```
> dds <- DESeq(y)

> res <- results(dds)

> resOrdered <- res[order(res$padj),]

> head(resOrdered,n=10)

log2 fold change (MAP): Treatment Dex vs Untreated
Wald test p-value: Treatment Dex vs Untreated
DataFrame with 10 rows and 6 columns
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENST00000239223	6484.43811	2.822921	0.1979928	14.25770	4.014632e-46	5.465118e-42
ENST00000262878	15368.12148	3.580092	0.2588879	13.82874	1.710009e-43	1.076418e-39
ENST00000377474	2029.11217	-2.435712	0.1764348	-13.80517	2.372183e-43	1.076418e-39
ENST00000338702	1451.26202	3.176371	0.2330070	13.63208	2.581087e-42	8.784085e-39
ENST00000296233	925.31336	3.963693	0.3321930	11.93190	8.071161e-33	2.197454e-29
ENST00000286713	11914.57326	1.476530	0.1249756	11.81455	3.283057e-32	7.448710e-29
ENST00000427716	1102.48436	2.238713	0.1994957	11.22186	3.185006e-29	6.193926e-26
ENST00000377482	3147.77243	2.356721	0.2119110	11.12128	9.884437e-29	1.681961e-25
ENST00000274711	54.85784	-3.539461	0.3269980	-10.82411	2.646511e-27	4.002995e-24
ENST00000289166	134.40371	2.217172	0.2080716	10.65581	1.638145e-26	2.230007e-23

> *resOrdered[11:16,]*

log2 fold change (MAP): Treatment Dex vs Untreated
Wald test p-value: Treatment Dex vs Untreated
DataFrame with 6 rows and 6 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENST00000287814	319.7023	2.674501	0.2527975	10.57962	3.704702e-26	4.584737e-23
ENST00000262424	2938.4702	2.580190	0.2442725	10.56275	4.434641e-26	5.030731e-23
ENST00000344327	295.8662	1.924680	0.1869240	10.29659	7.300213e-25	7.644446e-22
ENST00000377126	3277.2922	2.010074	0.1990938	10.09612	5.747301e-24	5.588429e-21
ENST00000304698	2500.2831	-1.332701	0.1325089	-10.05744	8.518347e-24	7.730684e-21

La primera columna de dicha salida, baseMean, muestra la media de reads normalizadas de cada tránscrito. El log_2-*fold-change*, logFC, como bien comentábamos anteriormente, muestra el cambio en la proporción de reads, en función del log_2, para las dos condiciones. En este caso, recordar que hemos tomado como nivel de referencia Untreated, sin tratamiento, por lo tanto, valores positivos indicarán que dichos tránscritos se expresan más en pacientes con tratamiento (Dex) y valores negativos determinarán que se expresan más en pacientes sin tratamiento (Untreated). La tercera columna, lfcSE, hace referencia al error estándar. El test utilizado para comprobar la expresión diferencial de cada tránscrito es el test paramétrico de Wald. El estadístico correspondiente a dicho test aparece en la cuarta columna, stat. Por otro lado, obtenemos los correspondientes p-valores y p-valores ajustados, a partir de los cuales obtenemos los tránscritos diferencialmente expresados.

Para conocer el número de tránscritos que se expresan de forma distinta en las dos condiciones consideradas, hacemos uso de la función *summary*, la cual nos dará el número de tránscritos que presentan un logFC positivo o negativo con un p-valor ajustado inferior al nivel de significación α =0’05.

> summary(res,alpha=0.05)

out of 65979 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up) : 746, 1.1%
LFC < 0 (down): 466, 0.71%
outliers [1] : 98, 0.15%
low counts [2] : 52268, 79%
(mean count < 12.7)

En conclusión, se obtienen 1.212 tránscritos diferencialmente expresados. A continuación, realizamos una representación gráfica de los distintos logFC en relación con la media de reads normalizadas de cada tránscrito, destacando en color rojo los puntos correspondientes a aquellos tránscritos diferencialmente expresados, mediante la función *plotMA*.

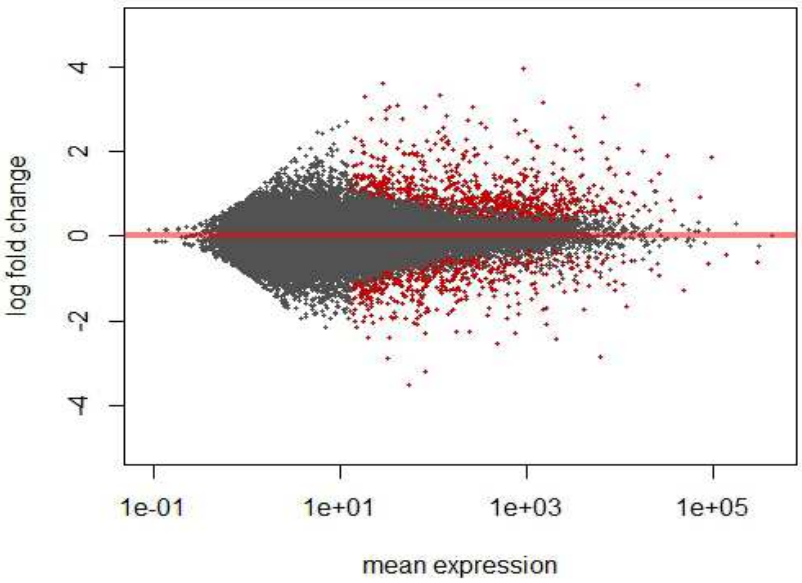


Figura 29. *plotMA*, logFC en relación con la media de reads normalizadas de cada tránscrito

Para finalizar, es interesante destacar que dicho paquete incluye la función *plotCounts*, la cual permite representar gráficamente la expresión diferencial de tránscritos entre condiciones. Vamos a mostrar la variación entre los datos normalizados de las dos condiciones consideradas para el tránscrito que ocupa la primera posición de los tránscritos diferencialmente expresados, dado que cuenta con el p-valor más bajo. Aquella condición para la que dicho tránscrito cuenta con mayor número de reads, los valores correspondientes a cada paciente de dicha condición tendrán mayor valor respecto al eje y (número de reads normalizado), ocurriendo lo contrario para los pacientes de la otra condición considerada en el estudio.

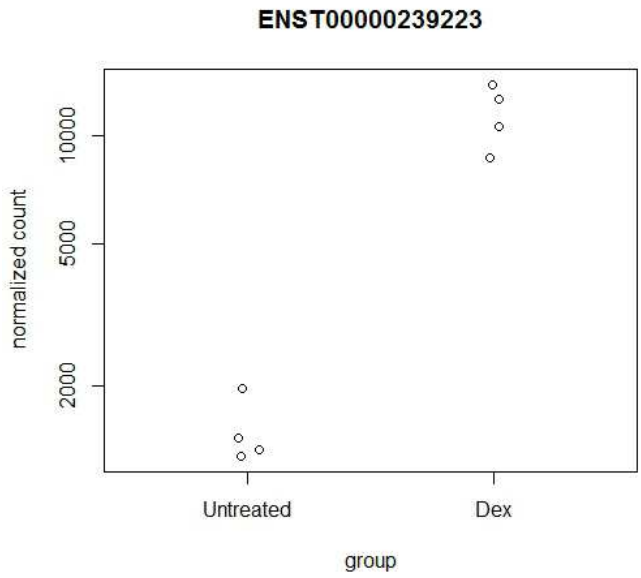


Figura 30. Expresión diferencial del tránscrito con menor p-valor.

- **Limma.**

Por último, realizaremos el análisis de expresión diferencial de tránscritos a través del uso del paquete Limma de Bioconductor en la plataforma R. Como bien comentábamos anteriormente, el propósito fundamental de dicho paquete es llevar a cabo el estudio de expresión diferencial de genes para datos de Microarrays. No obstante, se ha adaptado este paquete al análisis de datos de RNA-Seq. El manual de uso del paquete Limma, disponible en la página de Bioconductor y desarrollado principalmente para el análisis de datos de Microarrays, cuenta con un breve capítulo sobre la aproximación de su metodología al estudio de expresión diferencial de datos de RNA-Seq, la cual se basa, fundamentalmente, en el uso de modelos lineales.

Una vez importados los datos de conteo del archivo .tab, almacenados en el objeto *reads*, debemos crear el objeto *y* de clase *DGElist*. Para ello, haremos uso del paquete EdgeR, por lo que cargamos dicho paquete mediante los comandos de R mencionados con anterioridad. Esta acción provoca la instalación simultánea del paquete Limma. Tanto la normalización de los datos como el filtro de tránscritos con bajo número de reads, se realiza exactamente igual que en el EdgeR.

```
> y <- DGEList(counts=reads)

> keep <- rowSums(cpm(y)>5) >= 2

> y <- y[keep, ]
```

> y <- calcNormFactors(y)

Para poder hacer uso de las funciones del paquete Limma, los datos de conteo deben ser transformados a través de la función *voom*, la cual convertirá dichos datos al logaritmo del número de reads por millón. Cada dato dispondrá de un peso de precisión, que constituye la inversa de la dispersión estimada de dicha observación. Para poder aplicar dicha función debemos crear la matriz *design*, la cual incluye, por columnas, las condiciones a comparar en nuestro estudio y, por filas, los pacientes (el orden de los pacientes está especificado en la *tabla 4*). Dicha matriz contiene el diseño experimental, es decir, indica que muestras son réplicas biológicas de la misma condición, incluyendo un 1 cuando un paciente pertenece a una determinada condición y un 0 en caso de no pertenecer a dicha condición.

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

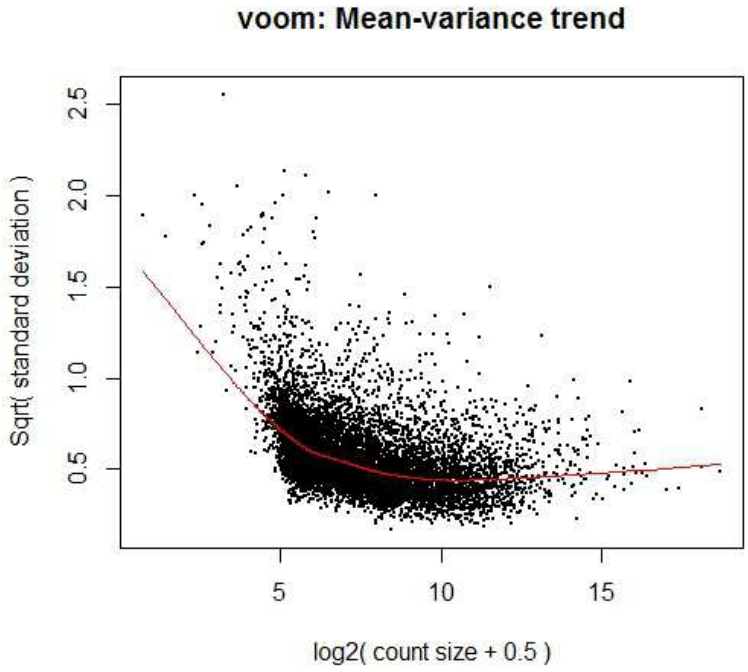


Figura 30. Dispersión de los datos, tendencia de la relación media-varianza.

El gráfico obtenido a través de la función *voom* nos permite obtener una visión general de la dispersión. Una línea recta sería sinónimo de que la media y la varianza tienden a ser iguales; sin embargo, obtenemos una curva, lo cual quiere decir que dichos datos presentan sobredispersión: varianza superior a la media.

Dicha función *voom* crea un objeto *EList* a partir del cual podemos realizar el estudio de expresión diferencial de tránscritos, siguiendo las mismas instrucciones que para el análisis de datos de Microarrays. Recordar que, como bien hemos comentado anteriormente, el paquete Limma se basa en el uso de modelos lineales. A través de la función *lmFit*, ajustamos los datos

de expresión de cada tránscrito del objeto *v* a un modelo lineal teniendo en cuenta el diseño experimental (matriz *design).* Por otro lado, especificamos en el objeto *contrastes*, mediante la función *makeContrasts,* las comparaciones que vamos a considerar: Untreated-Dex. Las funciones *contrasts.fit* y *eBayes* nos permiten realizar el estudio de expresión diferencial, basado en métodos bayesianos, entre las dos condiciones consideradas, especificadas en el objeto *contrastes.* La función *topTable* nos permite identificar los 15 tránscritos más diferencialmente expresados.

```
> fit <- lmFit(v,design)

> fit.main<-contrasts.fit(fit,contrast=contrastes)

> fit.bayes <- eBayes(fit.main)

> a<-topTable(fit.bayes, n=nrow(y$counts))

> head(a,n=15)
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
ENST00000377474	2.576535	7.383557	15.98125	4.026592e-08	0.0001899339	9.240460
ENST00000239223	-2.911518	8.893842	-15.84350	4.355783e-08	0.0001899339	9.167268
ENST00000286713	-1.458084	10.253581	-14.41313	1.025106e-07	0.0002396897	8.366085
ENST00000338702	-3.382033	6.543061	-14.30169	1.099368e-07	0.0002396897	8.288926
ENST00000262878	-3.742767	9.721849	-13.85404	1.463506e-07	0.0002503003	8.055285
ENST00000304698	1.393165	8.033456	13.60525	1.722053e-07	0.0002503003	7.948475
ENST00000389617	1.673652	6.339061	12.69873	3.190341e-07	0.0003932468	7.349856
ENST00000427716	-2.266847	6.559671	-12.44231	3.825744e-07	0.0003932468	7.180861
ENST00000323534	1.303472	7.001788	12.16602	4.670399e-07	0.0004073055	7.004043
ENST00000370763	-1.570059	6.790729	-11.95906	5.437636e-07	0.0004099645	6.856384
ENST00000377482	-2.410884	8.017683	-11.80230	6.111156e-07	0.0004099645	6.744136
ENST00000344327	-1.961144	4.780690	-11.84612	5.914090e-07	0.0004099645	6.652248
ENST00000289166	-2.289698	3.540038	-12.36001	4.058275e-07	0.0003932468	6.585652
ENST00000377126	-2.042245	8.206235	-11.14864	1.009735e-06	0.0006289926	6.257828
ENST00000375856	-2.074437	8.384758	-11.05548	1.086914e-06	0.0006319318	6.185618

Dicha salida presenta, en la primera columna, el correspondiente log_2*-fold-change,* logFC, de cada tránscrito, el cual muestra el cambio en la proporción de reads para las dos condiciones experimentales en función del log_2. Es conveniente señalar que estamos contrastando Untreated-Dex, por lo que valores positivos del logFC indicarán que dichos tránscritos se expresan más en pacientes sin tratamiento, ocurriendo de forma contraria para valores negativos, los cuales indicarán que se expresan más en pacientes con dexametasona. La columna AveExpr da el nivel medio de expresión de cada tránscrito, en función del log_2, en las dos condiciones consideradas. La siguiente columna, t, se corresponde con el estadístico correspondiente a cada test realizado para determinar la expresión diferencial de cada tránscrito

(prueba t-Student). Por otro lado, obtenemos los correspondientes p-valores y p-valores ajustados, a partir de los cuales, con un nivel de significación α =0’05, hallaremos los tránscritos diferencialmente expresados. La última columna, referida al estadístico B, proporciona el odds ratio, en función del log, es decir, el valor sobre el que podemos concluir que es más probable que dicho tránscrito esté diferencialmente expresado a que no lo esté. Por ejemplo, para el primer tránscrito, podemos concluir que la probabilidad de que esté diferencialmente expresado es de $\frac{e^{9.240460}}{(1+e^{9.240460})} = 0.9999$.

```
> nrow(a[a$adj.P.Val<=0.05,])

    719
```

A través del paquete Limma, obtenemos 719 tránscritos diferencialmente expresados.

6. CONCLUSIONES

Como resultado del análisis de datos reales, obtenemos para cada paquete un número diferente de tránscritos diferencialmente expresados. Anteriormente comentábamos que esto es debido, principalmente, a que los paquetes difieren en el modelado probabilístico de los datos de conteo correspondientes al número de reads de cada tránscrito para cada paciente. Entre dichos paquetes, como hemos visto, existen, además, diferencias en cuanto a los métodos de normalización, el filtrado de tránscritos con bajo nivel de expresión, los test para llevar a cabo el análisis de expresión diferencial, etc. La siguiente tabla muestra, a modo de resumen, las principales diferencias entre los 4 paquetes.

	EdgeR	**NOISeq**	**DESeq2**	**Limma**
Modelo probabilístico	Binomial Negativa	Aproximaciones no paramétricas	Modelo de regresión binomial negativo	Modelos lineales
Método de normalización	TMM	RPKM	Normalisation	TMM
Filtrado	CPM	CPM	Basado en la media de los datos normalizados para cada tránscrito	CPM
Test	Test exacto basado en la distribución Binomial Negativa	Estimación de la probabilidad de expresión diferencial	Test paramétrico de Wald	Prueba t-Student

Tabla 13. Diferencias entre los 4 paquetes empleados en nuestro estudio.

En nuestro estudio partimos inicialmente de 215.170 tránscritos. A continuación se muestra la reducción de dicho número tras el filtro de aquellos con bajo número de reads y el número obtenido de tránscritos diferencialmente expresados en cada paquete.

	EdgeR	NOISeq	DESeq2	Limma
Filtrado	8.727	8.095	162.902	8.727
Diferencialmente expresados	853	415	1.212	719

Tabla 14. Número de tránscritos tras el filtrado y el análisis de expresión diferencial.

En el paquete EdgeR, el cual se basa en la distribución Binomial Negativa, obtenemos 853 tránscritos diferencialmente expresados. Con NOISeq, basado en aproximaciones no paramétricas de los datos de conteo, que computa la expresión diferencial en función de los valores de los dos estadísticos M y D para cada tránscrito, se obtienen 415 tránscritos. Por otro lado, el paquete DESeq2, que analiza la expresión diferencial mediante el uso de modelos de regresión binomiales negativos, muestra que hay 1.212 tránscritos diferencialmente expresados. Por último, a través del paquete Limma, basado en el uso de modelos lineales y adaptado a partir del análisis de Microarrays para datos de RNA-Seq, se han obtenido 719 tránscritos. Es conveniente señalar que estos resultados se han obtenido para un nivel de significación α =0'05.

A continuación, la *figura 31*, correspondiente al diagrama de Venn, muestra la relación entre los tránscritos diferencialmente expresados obtenidos en cada paquete.

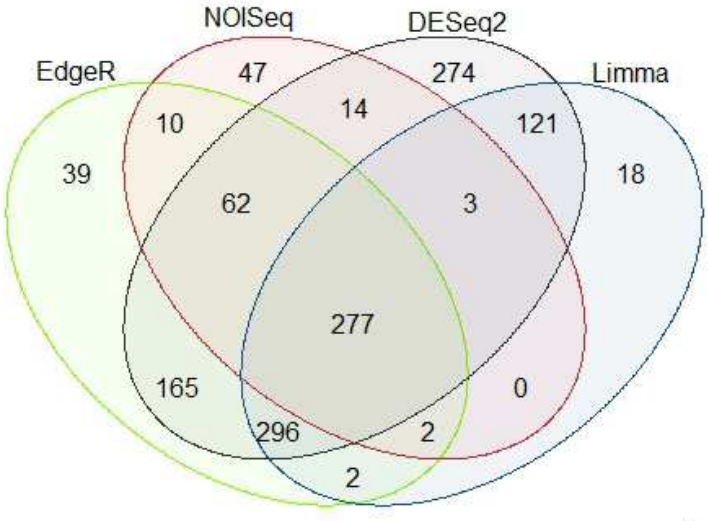


Figura 31. Diagrama de Venn: tránscritos diferencialmente expresados.

Podemos señalar que 277 tránscritos son considerados diferencialmente expresados entre pacientes sin tratamiento y pacientes con dexametasona para los 4 paquetes de Bioconductor utilizados en nuestro estudio de expresión diferencial.

Es importante destacar que si observamos los 15 tránscritos más diferencialmente expresados obtenidos en cada paquete, podemos apreciar que son prácticamente los mismos. Dado que están ordenados en función de los p-valores, salvo en el caso de NOISeq que están ordenados por probabilidad de expresión diferencial, y que dichos p-valores toman valores muy pequeños, la ordenación de dichos tránscritos varía de un paquete a otro.

Como línea futura de trabajo, basándonos en los resultados obtenidos en dicho estudio y mediante asesoramiento biológico, se debería completar con la interpretación biológica de dichos resultados. Para ello, el estudio continuaría con el paso de tránscritos a genes y la descripción de la funcionalidad de estos últimos, obtenida a partir de la base de datos del proyecto *Gene Ontology (GO).*

7. BIBLIOGRAFÍA

ANDERS, Simon; HUBER, Wolfgang. Differential expression analysis for sequence count data. *Genome Biology*, 2010, 11:R106.

BERNET FERNÁNDEZ, José Mª. Introducción a RNA-Seq. *MoleQla: revista de Ciencias de la Universidad Pablo de Olavide.* Marzo 2014, nº 13, p. 41-43. ISSN-e 2173-0903.

CARRASCO PEÑA, Manuel. *Modelización de conteos mediante la distribución Poisson-Tweedie (PT).* Proyecto final Diplomatura de Estadística. Universitat Autònoma de Barcelona, 38 p. (distr. Dipòsit Digital de Documents de la UAB).

CHEN, Yunshun *et al. edgeR: differential expression analysis of digitar gene expression data. User's Guide.* Biocondutor, 17 septiembre 2008. Úlltima revisión: 25 septiembre 2014. 77 p. [Consulta: mayo 2015] <http://www.bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>

HIMES, Blanca E. *et al.* Transcriptome Profiling Identifies CRISPLD2 as a Glucocorticoid Responsive Gen that Modulates Cytokine Function in Airway Smooth Muscle Cells. *PLOS ONE,* junio 2014, 9(6): e99625.

ILLUMINA, INC. *Getting Started with RNA-Seq Data Analysis.* Pub. nº 470-2011-003. California, Estados Unidos: 14 abril 2011.

LAW, Charity W *et al.* voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, 2014, 15(2): R29.

LORENZ, Douglas J. *et al.* Using RNA-seq Data to Detect Differentially Expressed Genes. En DATTA, Somnath; NETTLETON, Dan (eds.). *Statistical Analysis of Next Generation Sequencing Data.* Springer, 2014. p. 25-49.

LOVE, Michael; ANDERS, Simon; HUBER, Wolfgang. *Differential analysis of count data – the DESeq2 package.* Bioconductor, 1 mayo 2015. 48 p. [Consulta: mayo 2015] <http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.pdf>

ROBINSON, Mark D.; McCARTHY, Davis J; SMYTH, Gordon K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. En *Bioinformatics*, vol. 26 nº 1, 2010. p. 139-140

RODRÍGUEZ BAENA, Domingo S. *Análisis de datos de Expresión Genética mediante técnicas de Biclustering*. Memoria del periodo de investigación. Universidad de Sevilla, Mayo 2006. 101 p. (distr. Departamento de Lenguajes y Sistemas Informático de la Universidad de Sevilla).

SÁEZ GARCÍA, Zara. *Secuenciación por RNA-Seq y análisis del transcriptoma de células de neuroblastoma humano SH-SY5Y y tratadas con ácido retinoico*. Trabajo fin de máster. Universitat Politènica de València, marzo 2014, 74 p. (distr. Consejo Superior de Investigaciones Científicas, CSIC).

SMYTH, Gordon K. *et al. Linear Models for Microarray and RNA-Seq Data. User's Guide.* Biocondutor, 2 diciembre 2002. Última revisión: 9 junio 2015. p 69-73. [Consulta: junio 2015] <http://bioconductor.org/packages/release/bioc/vignettes/limma/inst/doc/usersguide.pdf>

TARAZONA, Sonia *et al.* Differential expression in RNA-seq: a matter or depth. *Genome Research*, 2011, 21: 2213-2223.

TARAZONA, Sonia *et al. NOISeq: Differential Expression in RNA-seq.* Biocondutor, 24 febrero 2014. 26 p. [Consulta: mayo 2015] <http://www.bioconductor.org/packages/release/bioc/vignettes/NOISeq/inst/doc/NOISeq.pdf>

TARAZONA, Sonia *et al.* NOISeq: a RNA-seq differential expression method robust for sequencing depth biases. *EMBnet.journal*, febrero 2012, vol. 17.B, p. 18-19. ISSN 2226-6089.

8. ANEXO

A continuación se muestra el código R completo de los 4 paquetes de Bioconductor utilizados, en dicho trabajo, para el estudio de la expresión génica diferencial.

- **EdgeR.**

> #Instalación EdgeR.

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

> biocLite("edgeR")

> library(edgeR)

> #Cargamos el archivo .tab con el número de reads para cada tránscrito en cada paciente.

> reads <- read.table(file="top_bw2-ReadCount.tab",header=T,sep="\t",row.names=1)

> colnames(reads) <- c("Untreated 1","Dex 1","Alb 1","Alb_Dex 1","Untreated 2","Dex 2","Alb 2","Alb_Dex 2","Untreated 3","Dex 3","Alb 3","Alb_Dex 3","Untreated 4","Dex 4","Alb 4", "Alb_Dex 4")

> #Eliminamos las muestras de tratamiento Alb y las muestras de Alb_Dex. Sólo vamos a trabajar con Untreated y Dex.

> reads<-reads[,-c(3,4,7,8,11,12,15,16)]

> head(reads)

>

> y <- DGEList(counts=reads)

> str(y)

>

> #Añadimos el factor grupo. Diferenciamos los dos grupos: Untreated y Dex. (Réplicas biológicas)

> nombres <- read.table(file="nombres.txt",header=TRUE)

> nombres <- nombres[-c(3,4,7,8,11,12,15,16),]

> est <- as.character(nombres[,3])

> group <- as.factor(est)

> group

>

```
> y <- DGEList(counts=reads,group=group)
> head(y)
> levels(y$samples$group)
> y$samples$group <- relevel(y$samples$group,ref = "Untreated")
> levels(y$samples$group)
>
> #Estructura, resumen, dimensión... del objeto y.
> str(y)
> y$samples
> head(y$counts)
> length(y$counts)
> summary(y$counts)
>
> plot(hclust(dist(t(y$counts))))
> plotMDS(y)
> dim(y) #Filas: número de tránscritos.
>
> #Normalización.
> ynorm <- calcNormFactors(y)
> ynorm$samples
> boxplot(log2(ynorm$counts),las=2)
>
> #Filtrado:
> keep <- rowSums(cpm(ynorm)>5) >= 2
> y <- ynorm[keep, ]
> table(keep)
> dim(y)
>
> #Estimación de la dispersión.
> y <- estimateCommonDisp(y,verbose=T)
> y <- estimateTagwiseDisp(y,trend="none")
```

> plotBCV(y)

>

> #Expresión diferencial: exactTest.

> et <- exactTest(y,,pair=c("Dex","Untreated"))

> topTags(et,n=15)

> topTags

> detags <- rownames(topTags(et, n=15))

> head(cpm(y)[detags,])

> summary(de <- decideTestsDGE(et, p=0.05, adjust="BH"))

> detags <- rownames(y)[as.logical(de)]

> length(detags)

>

> plotSmear(et, de.tags=detags)

> abline(h = c(-2, 2), col = "blue")

>

> reedger <- detags

- **NOISeq.**

> #Instalación NOISeq.

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

> biocLite("NOISeq")

> browseVignettes("NOISeq")

> library("NOISeq")

>

> #Cargamos el archivo .tab con el número de reads para cada tránscrito en cada paciente.

> reads <- read.table(file="top_bw2-ReadCount.tab",header=T,sep="\t",row.names=1)

> colnames(reads) <- c("Untreated 1","Dex 1","Alb 1","Alb_Dex 1","Untreated 2","Dex 2","Alb 2","Alb_Dex 2","Untreated 3","Dex 3","Alb 3","Alb_Dex 3","Untreated 4","Dex 4","Alb 4", "Alb_Dex 4")

> #Eliminamos las muestras de tratamiento Alb y las muestras de Alb_Dex. Sólo vamos a trabajar con Untreated y Dex.

> reads <- reads[,-c(3,4,7,8,11,12,15,16)]

> head(reads)

```
>
> nombres <- read.table(file="nombres.txt",header=TRUE)
> nombres <- nombres[-c(3,4,7,8,11,12,15,16),]
> est <- as.character(nombres[,3])
> group <- as.factor(est)
> group
> factores <- data.frame(Treatment=group)
> factores
>
> #Filtrado
> myfilt <- filtered.data(reads, factor = factores$Treatment, norm = FALSE, depth = 2, method
= 1, cv.cutoff = 100, cpm = 5)
> str(myfilt)
>
> y <-readData(data=myfilt,factors=factores)
> y
> str(y)
> head(assayData(y)$exprs)
> head(pData(y))
>
> #Expresión diferencial.
> mynoiseq <- noiseq(y, k = 0.5, norm = "rpkm", factor = "Treatment", pnr = 0.2, nss = 5, v =
0.02, lc = 1, replicates = "biological")
> head(mynoiseq@results[[1]])
> str(mynoiseq)
> mynoiseq.deg <- degenes(mynoiseq, q = 0.8, M = NULL)
> mynoiseq.deg1<- degenes(mynoiseq, q = 0.8, M = "up")
> mynoiseq.deg2 <- degenes(mynoiseq, q = 0.8, M = "down")
> str(mynoiseq.deg)
> head(mynoiseq.deg,n=15)
> DE.plot(mynoiseq, q = 0.8, graphic = "expr", log.scale = TRUE)
> DE.plot(mynoiseq, q = 0.8, graphic = "MD")
```

>

> renoiseq <- rownames(mynoiseq.deg)

- **DESeq2.**

> #Instalación DESeq2

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

> biocLite("DESeq2")

> library(DESeq2)

>

> #Cargamos el archivo .tab con el número de reads para cada tránscrito en cada paciente.

> reads <- read.table(file="top_bw2-ReadCount.tab",header=T,sep="\t",row.names=1)

> colnames(reads) <- c("Untreated 1","Dex 1","Alb 1","Alb_Dex 1","Untreated 2","Dex 2","Alb 2","Alb_Dex 2","Untreated 3","Dex 3","Alb 3","Alb_Dex 3","Untreated 4","Dex 4","Alb 4", "Alb_Dex 4")

> #Eliminamos las muestras de tratamiento Alb y las muestras de Alb_Dex. Sólo vamos a trabajar con Untreated y Dex.

> reads <- reads[,-c(3,4,7,8,11,12,15,16)]

> head(reads)

>

> nombres <- read.table(file="nombres.txt",header=TRUE)

> nombres <- nombres[-c(3,4,7,8,11,12,15,16),]

> nombres <- nombres[,-2]

> str(nombres)

> b <- as.character(nombres$Treatment)

> c <- as.factor(b)

> nombres[,2] <- c

> str(nombres)

>

> colnames(reads) <- nombres[,2]

> head(reads)

> treatment <- data.frame(Treatment=nombres[,2])

> str(treatment)

>

> attach(nombres)

> y <- DESeqDataSetFromMatrix(reads, treatment,design=~Treatment)

> detach(nombres)

> y

>

> y$Treatment <- relevel(y$Treatment,"Untreated")

> y$Treatment <- droplevels(y$Treatment)

> y$Treatment

>

> #Expresión diferencial.

> dds <- DESeq(y)

> res <- results(dds)

> mcols(res)$description

> resOrdered <- res[order(res$padj),]

> head(resOrdered,n=10)

> resOrdered[11:15,]

> summary(res,alpha=0.05)

> plotMA(res, alpha=0.05, main="DESeq2", ylim=c(-5,5))

>

> plotCounts(dds, gene=which.min(res$padj), intgroup="Treatment")

>

> redeseq2 <- rownames(head(resOrdered,n=1212))

- **Limma.**

> #Cargamos el archivo .tab con el número de reads para cada tránscrito en cada paciente.

> reads <- read.table(file="top_bw2-ReadCount.tab",header=T,sep="\t",row.names=1)

> colnames(reads) <- c("Untreated 1","Dex 1","Alb 1","Alb_Dex 1","Untreated 2","Dex 2","Alb 2","Alb_Dex 2","Untreated 3","Dex 3","Alb 3","Alb_Dex 3","Untreated 4","Dex 4","Alb 4", "Alb_Dex 4")

> reads <- reads[,-c(3,4,7,8,11,12,15,16)]

> head(reads)

>

> #Para Limma instalamos edgeR.

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("edgeR")
> library("edgeR")
>
> y <- DGEList(counts=reads)
> head(y$counts)
> dim(y)
>
> #Filtrado y normalización.
> keep <- rowSums(cpm(y)>5) >= 2
> y <- y[keep, ]
> dim(y)
> y <- calcNormFactors(y)
>
> nombres <- read.table(file="nombres.txt",header=TRUE)
> nombres <- nombres[-c(3,4,7,8,11,12,15,16),-2]
> str(nombres)
> b <- as.character(nombres$Treatment)
> c <- as.factor(b)
> nombres[,2] <- c
> str(nombres)
> nombres
>
> attach(nombres)
> design <- model.matrix(~0+Treatment)
> design
>
> contrastes <- makeContrasts(
>       UntreatedvsDex=TreatmentUntreated-TreatmentDex,
>       levels=design)
>
```

```
> #Transformación.
> v <- voom(y,design,plot=TRUE)
> str(v)
> head(v)
> summary(v)
>
> #Expresión diferencial.
> fit <- lmFit(v,design)
> fit.main <- contrasts.fit(fit,contrast=contrastes)
> fit.bayes <- eBayes(fit.main)
> a <- topTable(fit.bayes, n=nrow(y$counts))
> head(a,n=15)
> nrow(a[a$adj.P.Val<=0.05,])#Diferencialmente expresados
>
> relimma<-rownames(head(a,n=719))
```

- Diagrama de Venn.

```
> a <- reedger;length(a)
> b <- renoiseq;length(b)
> c <- redeseq2;length(c)
> d <- relimma;length(d)
>
> ?vennDiagram
> library(limma)
>
> extra <- c("A","B","C","D")
> universe <- union(a,union(b,union(c,union(d,extra))))
> Counts <- matrix(0,nrow=length(universe),ncol=4)
> colnames(Counts) <- c("EdgeR","NOISeq","DESeq2","Limma")
> for(i in 1:length(universe))
> {
```

```
>          Counts[i,1] <- universe[i]%in% a
>          Counts[i,2] <- universe[i]%in% b
 >        Counts[i,3] <- universe[i]%in% c
 >        Counts[i,4] <- universe[i]%in% d
> }
>  vennDiagram(vennCounts(Counts),
>          cex = 1,circle.col=c("chartreuse","firebrick3","gray19","dodgerblue4"))
```