

EJERCICIO DE RNASEQ

DAVID RUBIO MANGAS

TRABAJO DE RNASEQ

**MASTER EN BIOINFORMÁTICA APLICADA A
MEDICINA PERSONALIZADA Y SALUD 2023**



Ejercicio final de transcriptómica (curso 2022-2023)

El objetivo de este trabajo es que el alumno demuestre los conocimientos obtenidos acerca del análisis de RNA-seq. Para ello deberá **redactar un informe** en el que se expliquen los datos de partida y se extraiga una **conclusión de los resultados**. Así mismo, el alumno debe **detallar el proceso de análisis** indicando el *software* (incluida la versión) empleado, así como los parámetros utilizados en cada uno de los pasos. En caso de que hubiera que eliminar muestras por motivos técnicos o biológicos, el alumno debe indicar y justificar el por qué en cada caso. Se pueden introducir *code chunks* e imágenes para apoyar el informe. De manera alternativa, puede aportarse todo el código en forma de repositorio público. Se han planteado **5 preguntas (10 puntos en total)** para guiar la redacción del informe.

El trabajo consta de dos apartados en los que se utilizarán datos de un experimento en el que disponemos de 24 cultivos primarios de tumores paratiroides negativos para receptores de estrógenos alfa (ER α). Las muestras, procedentes de 4 pacientes diferentes, se han tratado con dos fármacos diferentes: diarilpropionitrilo (DPN) o 4-hidroxitamoxifeno (OHT) a 24h o 48h. El DPN es un agonista del ER α mientras que el OHT es un inhibidor competitivo de los receptores de estrógenos.

El **primer apartado (3 preguntas)** abarca los pasos de control de calidad y de fuentes de contaminación, *trimming*, alineamiento y cuantificación para obtener cuentas crudas y normalizadas a partir de un **subset de ficheros fastq**. El **segundo apartado (2 preguntas)** parte de la **matriz completa de cuentas crudas** y está enfocado a realizar un control de calidad biológico, detectar los genes diferencialmente expresados entre condiciones y los pathways enriquecidos en cada una de ellas.

Apartado 1

El dataset original consta de 27 muestras paired-end depositadas en SRA. Con el fin de poder abordar las cuestiones planteadas a en el primer apartado sólo es necesario descargar 2 muestras (SRR479052 y SRR479054), es decir cuatro ficheros fastq. Además, en este repositorio se proporciona:

- Un fichero fasta con la secuencia de la referencia genómica, en este caso correspondiente al cromosoma 21 humano (ensamblaje GRCh38).
- Un fichero GTF con la anotación génica para los genes del cromosoma 21 (GRCh38.ensembl.109).

Se pide realizar un análisis de dichas muestras similar al realizado en clase, considerando los siguientes puntos:

Pregunta 1 (1.5 puntos): Realizar control de calidad de dichas muestras con el programa FastQC, incluir plots más reseñables y comentar cada uno de los apartados. De manera complementaria se podrá realizar un análisis de contaminación con el programa FastQScreen.

Pregunta 2 (1.5 puntos): Para poder llevar a cabo el alineamiento de las muestras en vuestros ordenadores será necesario trabajar con archivos reducidos correspondientes al cromosoma 21. Se requiere el indexado de la secuencia de este cromosoma, así como el alineamiento de las muestras a dicha referencia. Para ello se podrá utilizar el alineador HISAT2 utilizado en clase u otros (alineadores o pseudoalineadores). Comentar cada uno de los comandos y parámetros empleados, justificando su uso.

Pregunta 3 (1.5 puntos): Una vez generados los archivos alineados se reportarán las **estadísticas de alineamiento** y se procederá a la cuantificación de la expresión utilizando el archivo GTF correspondiente. Para ello se podrá utilizar HTSeq u otras herramientas de cuantificación. En cualquier caso, detallar y justificar los comandos y parámetros empleados para ello.

Apartado 2

En este repositorio se proporcionan todos los inputs del Apartado 2:

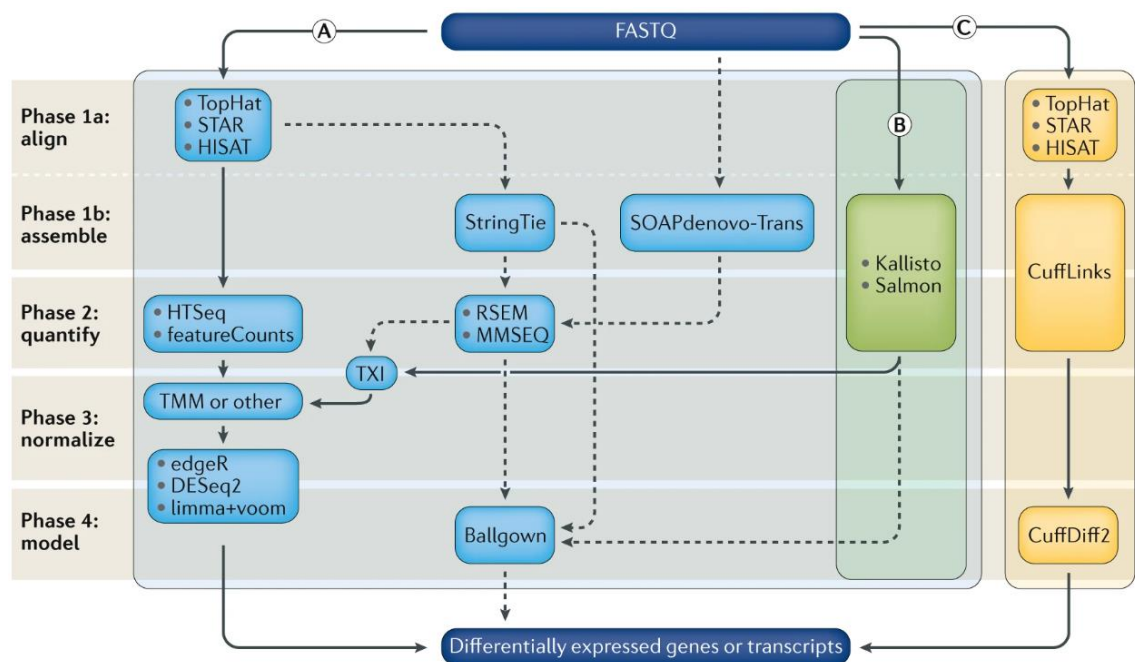
- La matriz de cuentas crudas para los 24 cultivos analizados.
- Data frame con los metadatos asociados al experimento.
- GMT para realizar un GSEA.

Pregunta 4 (3 puntos): ¿Qué genes se encuentran diferencialmente expresados entre las muestras pertenecientes al grupo tratado con OHT con respecto al control tras 24h? ¿Y en el caso de las muestras tratadas con DPN, también tras 24h? Como parte de la respuesta a esta pregunta, podéis entregar una o más tablas adjuntas donde se incluyan los genes diferencialmente expresados, indicando el criterio o los criterios que habéis seguido para filtrar los resultados, así como cualquier otro gráfico o gráficos que hayáis elaborado durante el análisis.

Pregunta 5 (2.5 puntos): Nuestro colaborador ha comparado las muestras tratadas con DPN tras 48h con las muestras control. Nos ha llamado para contarnos que los cambios de expresión tras 48h son mucho más evidentes y nos preguntamos si el DPN produce algún efecto en las primeras 24h. Para contestar esta pregunta, le pedimos que genere un GMT (input) con los genes más expresados en las muestras tratadas tras 48h (DPN_perturbed) y los genes más expresados en la muestra control (DPN_unperturbed). Realiza un análisis con GSEA para determinar el efecto del tratamiento a las 24h. ¿A qué conclusión llegas? Incluid una tabla con los resultados del análisis, destacando las columnas con los valores utilizados para extraer vuestras conclusiones. También incluid los gráficos característicos de este tipo de análisis.

Introducción

Next-generation sequencing (NGS) usando **RNA**, es una herramienta muy efectiva para la investigación de los transcritos presentes en células y tejidos, tanto en estudios experimentales como observacionales. Su objetivo es identificar y cuantificar todos los transcritos presentes en un tipo de célula o tejido en un estado determinado, lo que proporciona información sobre los genes e isoformas que se expresan en condiciones específicas. Esta técnica es crucial para entender cómo los cambios en la expresión génica o transcripcional afectan las funciones celulares, tejidos u organismos. Permite analizar la regulación transcripcional, las vías de señalización y la organización de las redes génicas. El flujo de trabajo seguido se representa en la siguiente figura [1]:



RNA-seq data analysis workflow for differential gene expression. Imagen sacada de: Stark, R., Grzelak, M. & Hadfield, J. RNA sequencing: the teenage years. *Nat Rev Genet* **20**, 631–656 (2019). <https://doi.org/10.1038/s41576-019-0150-2>

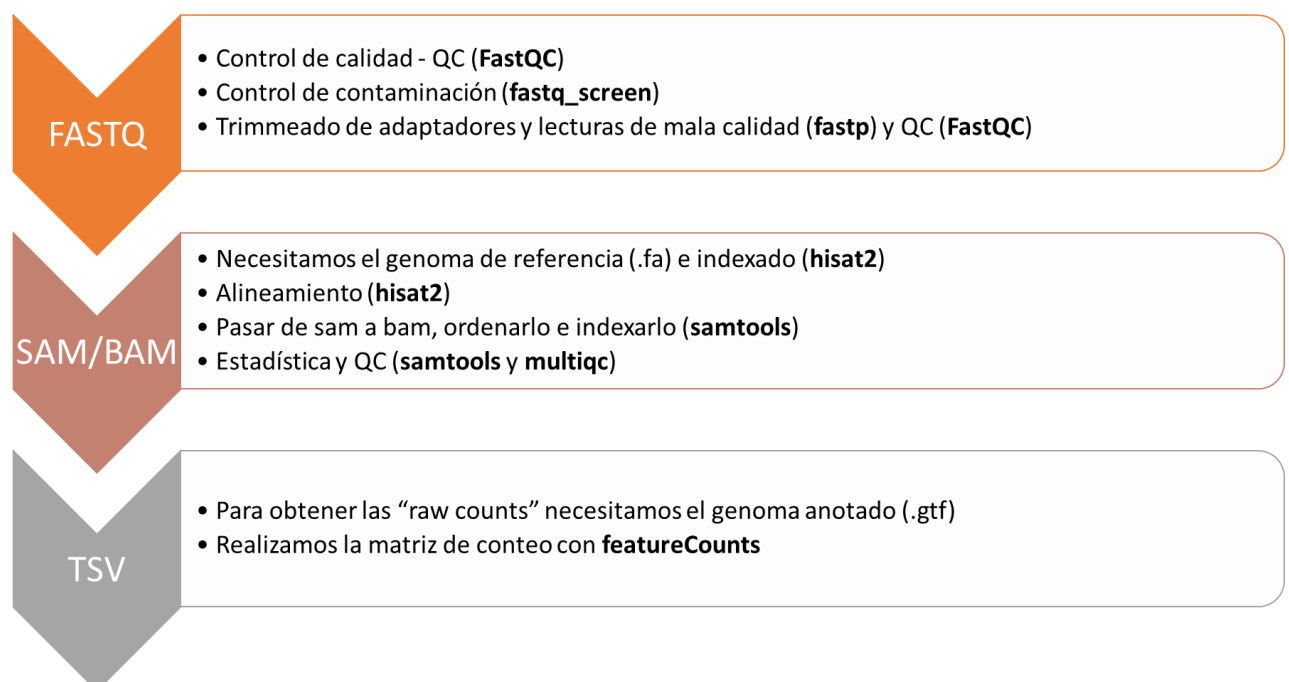
El análisis computacional de la expresión génica diferencial comienza con lecturas de secuenciación de ARN (ARN-seq) en formato FASTQ y puede seguir varios caminos. Se ofrecen como ejemplo tres flujos de trabajo populares (A, B y C, representados por las líneas continuas) y se indican algunas de las herramientas alternativas más comunes (representadas por las líneas discontinuas). En el flujo de trabajo A, alineadores como TopHat, STAR o HISAT utilizan un genoma de referencia para asignar las lecturas a ubicaciones genómicas y, a continuación, herramientas de cuantificación como HTSeq y featureCounts asignan las lecturas a características. Tras la normalización (normalmente utilizando métodos integrados en las herramientas de cuantificación o modelización de la expresión, como la media recortada de valores M (TMM), la expresión génica se modela utilizando herramientas como edgeR, DESeq2 y limma+voom, y se genera una lista de genes o transcritos expresados diferencialmente para su posterior visualización e interpretación. En el flujo de trabajo B, las herramientas más recientes y sin alineación, como Kallisto y Salmon, ensamblan un transcriptoma y cuantifican la abundancia en un solo paso. El resultado de estas herramientas se convierte normalmente en

estimaciones de recuento (utilizando tximport (TXI)) y se somete a la misma normalización y modelización utilizada en el flujo de trabajo A, para obtener una lista de genes o transcritos expresados diferencialmente. Alternativamente, el flujo de trabajo C comienza alineando las lecturas (normalmente se realiza con TopHat, aunque también pueden utilizarse STAR e HISAT), seguido del uso de CuffLinks para procesar las lecturas en bruto y el paquete CuffDiff2 para obtener estimaciones de abundancia de transcripción y una lista de genes o transcripciones expresados diferencialmente. Otras herramientas de uso común son StringTie, que ensambla un modelo de transcriptoma a partir de TopHat (o herramientas similares) antes de pasar los resultados a RSEM o MMSEQ para estimar la abundancia de transcritos, y después a Ballgown para identificar los genes o transcritos expresados diferencialmente, y SOAPdenovo-trans, que alinea y ensambla simultáneamente las lecturas para su análisis a través de la ruta elegida.

Para realizar el RNAseq debemos tener en cuenta información acerca del experimento;

- El dataset original consta de 27 muestras paired-end depositadas en SRA
- En el análisis se utilizarán **2 muestras (SRR479052 y SRR479054)**, es decir cuatro ficheros **fastq** (debido a que se utilizó la aproximación paired-end).
- Un fichero **fasta** con la secuencia de la **referencia** genómica, en este caso correspondiente al **cromosoma 21** humano (ensamblaje **GRCh38**).
- Un fichero **GTF** con la **anotación** génica para los genes del **cromosoma 21 (GRCh38.ensembl.109)**.
- Los resultados provienen de 24 cultivos primarios de tumores paratiroides negativos para receptores de estrógenos alfa (ER α). Se utilizaron dos fármacos diferentes: diarilpropionitrilo (DPN) o 4-hidroxitamoxifeno (OHT) a 24h o 48h. DPN es un agonista del ER α mientras que el OHT es un inhibidor competitivo de los receptores de estrógenos

Previamente, los datos de secuenciación se demultiplexaron con **bcl2fastq** que convierte los basecalls de los secuenciadores de **Illumina** a archivos **Fastq**. El **workflow** principal que se ha seguido para realizar este experimento ha sido el siguiente:



Para esta práctica se ha creado un script general llamado **pipeline.sh** que realiza el workflow de manera automática para estas dos muestras (**SRR479052 y SRR479054**), en el caso de que se introduzca nuevas muestras al estudio, se tendrán que hacer modificaciones pertinentes en el script. Este script llama dentro de él a otros script secundarios y algunos de ellos son; **preproc.sh** (para el preprocesamiento de las muestras y trimmeado), **screen.sh** (para detectar posibles contaminaciones de las muestras), **align.sh** (para el alineamiento de las muestras con hisat2), **posproc.sh** (para convertir los archivos .sam generados del alineamiento en binarios (.bam), además de ordenarlos e indexarlos), **stadistic.sh** (para obtener métricas y estadísticas del alineamiento (samtools) y general de las muestras en conjunto (multiqc)), **merged.sh** (para guardar la matriz de conteo de cada muestra en un único archivo donde tenemos en la primera columna los genes y en las siguientes columnas el conteo de cada muestra por gen) y, finalmente, un script llamado **remove.sh** para eliminar los .sam para liberar espacio (se puede modificar para eliminar todo lo que no queramos utilizar). También se ha generado un script para saber las versiones que se han utilizado para estos programas llamados **versión.sh**.

Para replicar todo el proceso, en primer lugar, debemos de crear un nuevo entorno de **conda** o **mamba** donde instalaremos los programas utilizados.

1. **Instala conda:** Si aún no tienes Conda instalado en tu sistema, puedes descargar e instalar la versión adecuada desde <https://conda.io/miniconda.html>
2. **Descargar RNAseq_DavidRubio.yml.** Aquí encontraras los softwares y las dependencias usadas.
3. **Crear un nuevo ambiente y utilizar las dependencias/softwares de RNAseq_DavidRubio.yml.** Para ello, necesitas usar el siguiente comando “conda env create -n {nombre_de_tu_nuevo_ambiente} -f RNAseq_DavidRubio.yml”
4. **Activar el ambiente.** Para ello, debes ejecutar el siguiente comando “conda activate {nombre_de_tu_nuevo_ambiente}”.

Asegúrate de que estás en el directorio que contiene el archivo **RNAseq_DavidRubio.yml**, para realizar los pasos anteriores. Además, en el caso de que no quieras realizar esto, te recomiendo ejecutar “bash version.sh” para ver las versiones utilizadas e instalarlas de forma libre mediante “conda install -c bioconda fastqc –versionXXX”. En el caso de **multiqc** se tiene que instalar mediante “pip install”.

1. Control de calidad y *trimming*

FASTQC – QUALITY CONTROL (QC)

Durante esta etapa, se evalúa la calidad de la secuenciación a partir de archivos crudos en formato FASTQ. El **control de calidad** de la secuenciación de las muestras se realizó con **FASTQC** software [2]. Para ello, primero tenemos que definir dos variables y correr el comando:

```
#define variables
fastq_path='transcriptomic-final-exercise/Apartado1/input/'
fastqc_path='transcriptomic-final-exercise/Apartado1/output/fastqc/'

#FastQC quality control to raw data
for seq_fastq in $(find "$fastq_path" -name '*.fastq')
do
    fastqc "$seq_fastq" -o "$fastqc_path"
done
```

En este bucle “for” introducimos al software **fastqc** cada una de las muestras para que nos genere un resultado y lo guarde en la carpeta **/fastq**. En este sentido, el programa genera dos tipos de archivos de salida:

1. **Archivo .html:** Este archivo contiene un informe detallado de los resultados del análisis de calidad. El archivo HTML se puede abrir en cualquier navegador web y proporciona una visualización gráfica de los resultados de calidad. El informe contiene diferentes módulos que resumen la calidad de los datos en diferentes aspectos, como la distribución de la calidad de las bases en la secuencia, la presencia de adaptadores, la calidad de las bases en diferentes posiciones de la lectura, entre otros.
2. **Archivo .zip:** Este archivo contiene los gráficos y las estadísticas detalladas de cada módulo del análisis de calidad, en formato de imagen. Los gráficos se pueden abrir en cualquier programa de visualización de imágenes y se pueden utilizar para hacer presentaciones o informes más detallados.

Ahora veremos los resultados obtenidos para las dos muestras:

MUESTRA SRR479052

Resultados del módulo "Basic Statistics" de FastQC.

Basic Statistics

Encoding	Sanger / Illumina 1.9
Total Sequences	15340
Total Bases	1.5 Mbp
Sequences flagged as poor quality	0
Sequence length	101
%GC	52

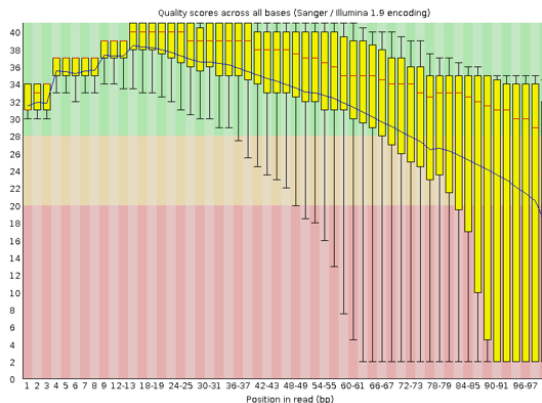
- Encoding: Se utilizó el estándar de codificación Sanger / Illumina 1.9 para la secuenciación.
- Total Sequences: Se secuenciaron un total de 15340 secuencias.
- Total Bases: La longitud total de todas las secuencias es de 1.5 millones de pares de bases (Mbp).
- Sequences flagged as poor quality: Ninguna de las secuencias fue identificada como de baja calidad.
- Sequence length: La longitud de todas las secuencias es de 101 bases.
- %GC: El porcentaje de nucleótidos G y C en la secuencia es del 52%.

Resultados del módulo "Per base sequence quality" de FastQC.

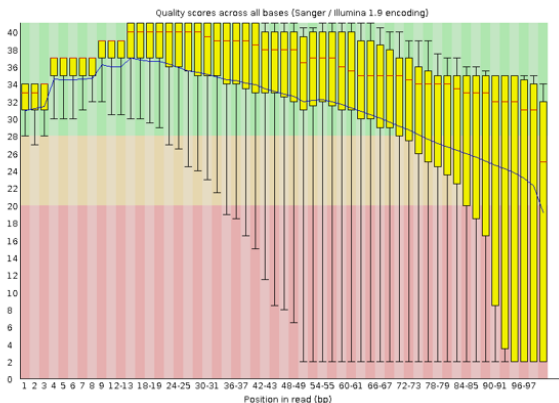
Este módulo proporciona información sobre la calidad de las bases en la secuencia en cada posición de la secuencia, lo que puede ser útil para identificar posibles problemas en la calidad de los datos, como regiones de baja calidad

SRR479052

READ 1



READ 2



Observamos la calidad media (en escala Phred 0-40) y su desviación en cada base a lo largo de todas las lecturas del archivo FastQ. Como se observa en la imagen, a lo largo de 66 pb deja de ser considerada "buena" y, por lo tanto, apunta de que se necesita un preprocesamiento de la muestra para que el alineamiento de la muestra vaya bien (al menos se debería de eliminar las secuencias que se encuentran en la franja roja).

La conclusión de este apartado: Se necesita eliminar las secuencias de mala calidad para que no influyan en los próximos análisis.

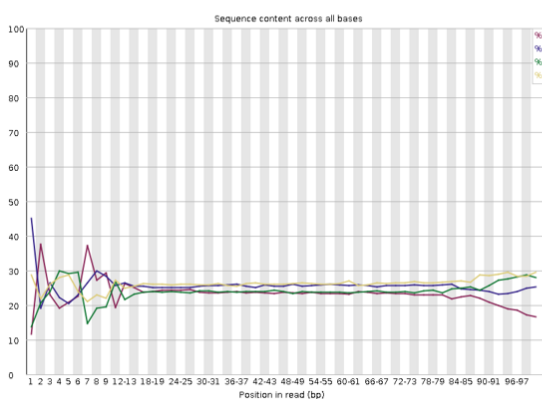
Resultados del módulo "Per base sequence quality" de FastQC.

El módulo "Per base sequence content" de FastQC es otro de los módulos que se utiliza para evaluar la calidad de los datos de secuenciación generados por los secuenciadores de nueva generación (NGS).

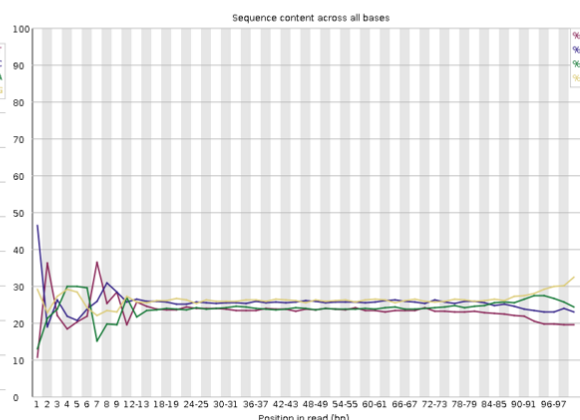
Este módulo analiza la distribución de los nucleótidos A, C, G y T en cada posición de la secuencia, lo que puede ser útil para identificar posibles problemas en la calidad de los datos, como desequilibrios en la composición de nucleótidos o adaptadores presentes en las secuencias.

SRR479052

READ 1



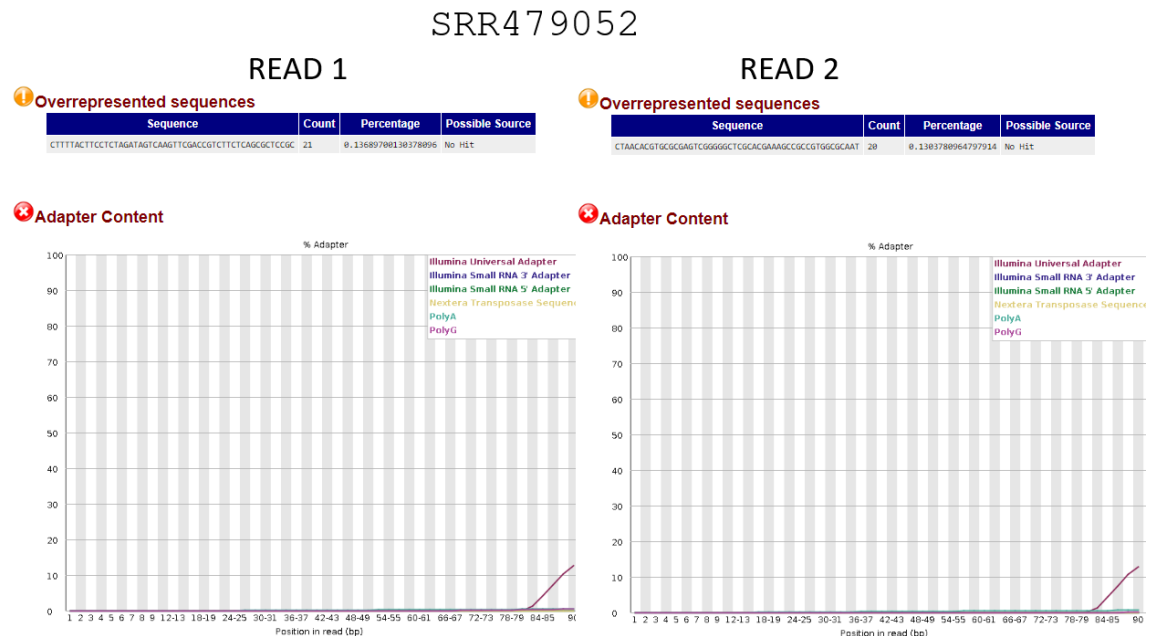
READ 2



En este caso, no hay ningún tipo de error, ya que estos gráficos presentan ciertas características que son típicas de este tipo de datos (RNA-seq). En la generación de cDNA para las librerías de RNA-seq se utilizan "random hexamer primers" para iniciar la síntesis de cDNA, lo que puede resultar en una distribución no uniforme de los nucleótidos y en un contenido de GC variable en las secuencias. En general, esto se considera normal en los datos de RNA-seq y no indica un problema de calidad de los datos en sí mismos.

Resultados del módulo "Overrepresented sequences y adapter content" de FastQC.

El módulo "Overrepresented sequences" de FastQC es una herramienta que se utiliza para identificar secuencias que aparecen en una cantidad inusualmente alta en los datos de secuenciación. El módulo "Adapter Content" es una herramienta similar que también se utiliza para identificar la presencia de adaptadores en las secuencias de los datos de secuenciación y nos muestra un gráfico sobre los adaptadores comúnmente utilizados.



Se ha detectado la presencia de dos secuencias sobrerepresentadas tanto en la read1 como en la read2, se utilizará BLAST y fastq_screen para observar si hay contaminación. También, se ha detectado la presencia de adaptadores universales de Illumina desde 79 pb llegando a estar presente hasta en un 10%.

La conclusión de este apartado: Se necesita conocer si tenemos contaminación en esta muestra con fastq_screen y se necesita eliminar la secuencia de adaptadores con fastp.

Las demás métricas salen correctas:

- ✅ [Basic Statistics](#)
- ❌ [Per base sequence quality](#)
- ✅ [Per sequence quality scores](#)
- ❌ [Per base sequence content](#)
- ✅ [Per sequence GC content](#)
- ✅ [Per base N content](#)
- ✅ [Sequence Length Distribution](#)
- ✅ [Sequence Duplication Levels](#)
- 🚩 [Overrepresented sequences](#)
- ❌ [Adapter Content](#)

Para más información abrir el **.html** de la carpeta **/fastqc**.

MUESTRA SRR479054

Resultados del módulo "Basic Statistics" de FastQC.



Basic Statistics

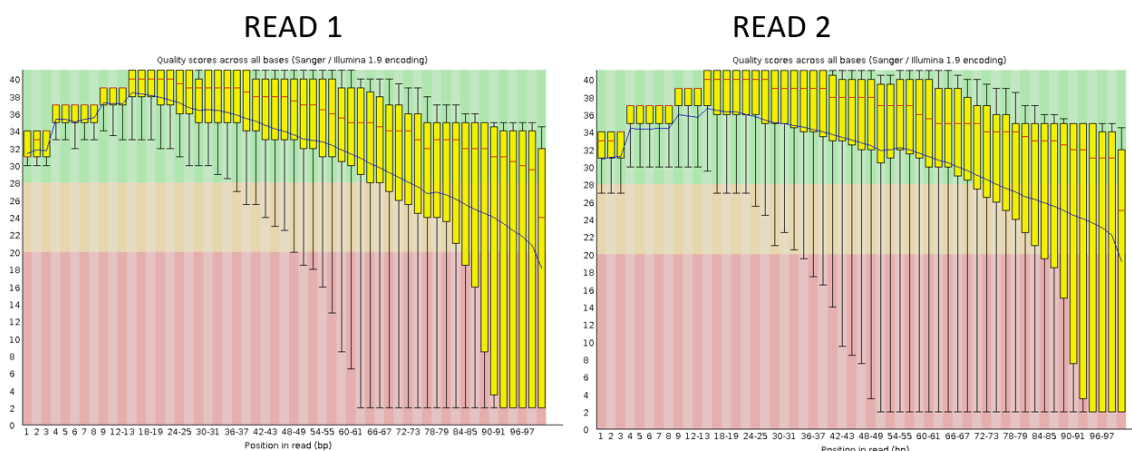
Encoding	Sanger / Illumina 1.9
Total Sequences	9746
Total Bases	984.3 kbp
Sequences flagged as poor quality	0
Sequence length	101
%GC	51

En esta tabla se encuentra la siguiente información:

- Encoding: Se utilizó el estándar de codificación Sanger / Illumina 1.9 para la secuenciación.
- Total Sequences: Se secuenciaron un total de 9746 secuencias.
- Total Bases: La longitud total de todas las secuencias es de 984.3 kilobases pares (kbp).
- Sequences flagged as poor quality: Ninguna de las secuencias fue identificada como de baja calidad.
- Sequence length: La longitud de todas las secuencias es de 101 bases.
- %GC: El porcentaje de nucleótidos G y C en la secuencia es del 51%.

Resultados del módulo " Per base sequence quality" de FastQC.

SRR479054



En la figura previa, se puede observar el análisis de calidad de los datos de secuenciación FastQ. Observamos que la calidad media se encuentra en escala Phred 0-40, así como su desviación, dicha calidad disminuye a lo largo de las 66 pb. Esto sugiere que las lecturas en esta región no cumplen con los estándares de calidad deseados y, por lo tanto, podrían afectar negativamente los resultados de alineación y análisis posteriores.

Las lecturas que se encuentran en la franja roja indican que estas secuencias tienen baja calidad y deberían ser eliminadas para asegurar una buena calidad de los datos y evitar que influyan en los próximos análisis. Se debería realizar un preprocesamiento de la muestra para eliminar estas secuencias.

En conclusión, se recomienda eliminar las secuencias de mala calidad para garantizar la calidad de los datos y obtener resultados confiables en los análisis posteriores.

Resultados del módulo " Overrepresented sequences and adapter content" de FastQC.

SRR479054

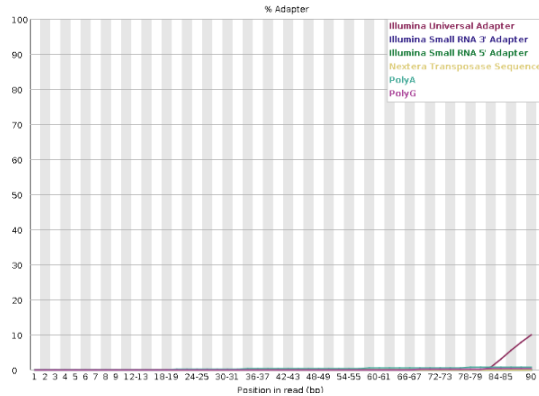
READ 1

READ 2

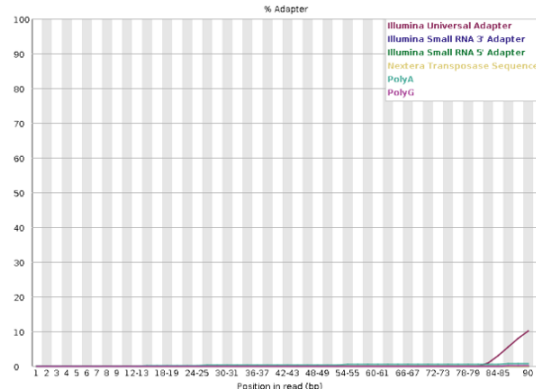
✔ Overrepresented sequences
No overrepresented sequences

✔ Overrepresented sequences
No overrepresented sequences

✘ Adapter Content



✘ Adapter Content



En este caso, no se han detectado secuencias sobrerrepresentadas, pero si se ha detectado la presencia de adaptadores universales de Illumina desde 79 pb y su presencia llega hasta un 10%.

En conclusión: *Estos resultados, sugieren que las lecturas de secuencia contienen adaptadores de secuenciación que deben ser eliminados antes de continuar con el análisis*

Las demás métricas salen correctas (aunque el módulo “per base sequence content” aparezca como incorrecto, la gráfica es correcta para este tipo de método – RNAseq):

- ✔ Basic Statistics
- ✘ Per base sequence quality
- ✔ Per sequence quality scores
- ✘ Per base sequence content
- ✔ Per sequence GC content
- ✔ Per base N content
- ✔ Sequence Length Distribution
- ✔ Sequence Duplication Levels
- ✔ Overrepresented sequences
- ✘ Adapter Content

Para más información abrir el **.html** de la carpeta **/fastqc**.

FASTP - TRIMMING

Después del control de calidad, se realizó el **PREPROCESAMIENTO** o **TRIMMING** con **Fastp** [3]. Fastp es una herramienta de preprocesamiento de datos de secuenciación de alta velocidad y eficiencia. Se utiliza para eliminar adaptadores, recortar bases de baja calidad, filtrar secuencias de baja calidad y realizar otras operaciones de preprocesamiento en los datos de secuenciación. Se caracteriza por ser muy rápido, eficiente y por ser fácil de usar gracias a su interfaz de línea de comandos y sus numerosas opciones de configuración.

En nuestro caso, se realizó un script (**preproc.sh**) para el preprocesamiento de las muestras de manera automática. Se definió las variables de adaptadores, el primero, "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA", es el adaptador de secuencia para la lectura 1 y el

segundo, "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT", es el adaptador de secuencia para la lectura 2 en una secuenciación de tipo paired-end.

El script que se utilizó fue el siguiente:

```
# Definir variables
adapter_seq="AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
adapter_seq_r2="AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
input_dir="transcriptomic-final-exercise/Apartado1/input"
output_dir="transcriptomic-final-exercise/Apartado1/output/trimmed"
mkdir -p "$output_dir"

# Iterar sobre cada muestra y ejecutar fastp
for r1 in "$input_dir"/*.chr21_1.fastq
do
    # Obtener el nombre base de la muestra sin la extensión
    sample=$(basename "$r1" .chr21_1.fastq)

    # Construir los nombres de archivo de entrada y salida para la muestra actual
    r2="$input_dir/${sample}.chr21_2.fastq"
    r1_trimmed="$output_dir/${sample}.chr21_1.trimmed.fastq"
    r2_trimmed="$output_dir/${sample}.chr21_2.trimmed.fastq"

    # Ejecutar fastp
    fastp -i "$r1" -I "$r2" -o "$r1_trimmed" -O "$r2_trimmed" \
        --adapter_sequence "$adapter_seq" \
        --adapter_sequence_r2 "$adapter_seq_r2" \
        --cut_mean_quality 20 \
        --cut_front \
        --cut_tail \
        --length_required 50 \
        --detect_adapter_for_pe \
        --thread 8
        --html "$output_dir/${sample}_fastp.html"
done
# Esperar a que todos los procesos finalicen
wait

echo "Procesamiento de las muestras completado."
```

Se extrajo el nombre base de la muestra sin la extensión ".chr21_1.fastq" y esto se realizó para construir los nombres de archivo de entrada y salida para la muestra actual y se construyen los nombres de archivo de entrada y salida para la muestra actual. Una vez realizado, se ejecutó el **fastp** con los siguientes argumentos:

- "-i" indica el archivo de entrada read_1 (chr21_1.fastq)
- "-I" indica el archivo de entrada read_2 (chr21_2.fastq)
- "-o" indica el archivo de salida read_1 después del preprocesamiento.
- "-O" indica el archivo de salida read_2 después del preprocesamiento.
- "--adapter_sequence" indica la secuencia del adaptador a eliminar en read_1 (estos adaptadores son los universales de illumina Truseq).
- "--adapter_sequence_r2" indica la secuencia del adaptador a eliminar en read_2 (estos adaptadores son los universales de illumina Truseq).
- "--cut_mean_quality 20" indica que se deben recortar las bases de baja calidad con una calidad media inferior a 20.
- "--cut_front" y "--cut_tail" indican que se deben recortar las bases de baja calidad al principio y al final de la secuencia.
- "--length_required 50" indica que se deben eliminar las secuencias que sean más cortas que 50 pares bases después del preprocesamiento.

- "--detect_adapter_for_pe" indica que se debe detectar la secuencia del adaptador automáticamente para las lecturas paired-end.
- "--thread 8" indica que se deben usar 8 hilos/cores para el procesamiento.

Los resultados de fastp para cada muestra están disponibles en el archivo *_**fastp**.html. Además de los resultados específicos de cada muestra, fastp también proporciona un resumen sobre estadísticas, estimación del insert-size, read-quality, base-content, antes y después del filtrado. Sin embargo, Para evaluar los efectos del preprocesamiento en la calidad de los datos, se volvió ejecutar el fastqc para observar las gráficas generadas. Se puede observar las estadísticas generadas por fastp en la carpeta de /trimmed.

FASTQC – QC-TRIMMING

MUESTRA SRR479052 trimmed

Resultados del módulo "Basic Statistics" de FastQC.



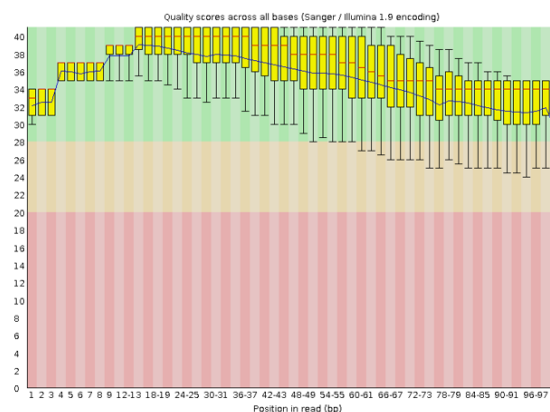
Basic Statistics

Encoding	Sanger / Illumina 1.9
Total Sequences	12790
Total Bases	1.1 Mbp
Sequences flagged as poor quality	0
Sequence length	50-101
%GC	50

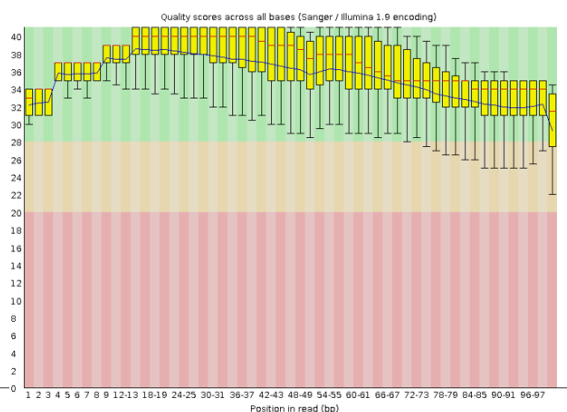
Resultados del módulo " Per base sequence quality" de FastQC.

SRR479052

READ 1



READ 2



Resultados del módulo " Overrepresented sequences y adapter content" de FastQC.

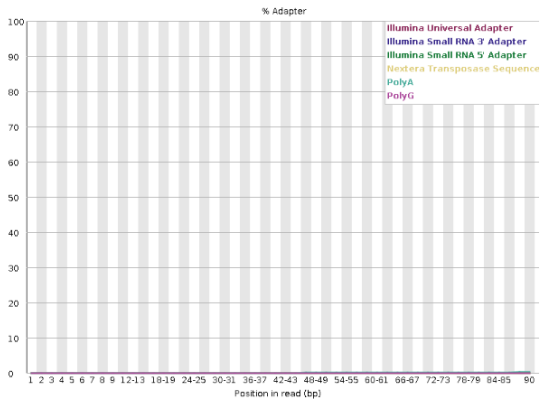
SRR479052

READ 1

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTTTTACTTCTCTAGATAGTCAGGCTCTCTCAGGCTCCGC	19	0.1485535574667799	No Hit

Adapter Content

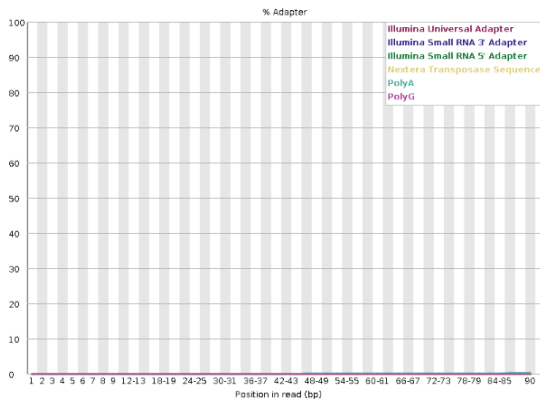


READ 2

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTAACACGTGCGGAGTCGGGGCTCGACCAAGCCCGCTGGCGAAT	18	0.12509773268359855	No Hit

Adapter Content



MUESTRA SRR479054 trimmed

Resultados del módulo "Basic Statistics" de FastQC.



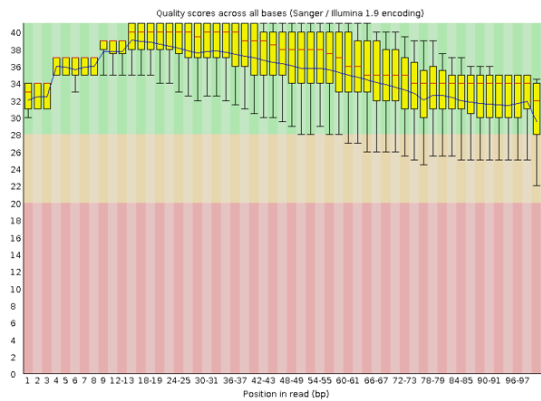
Basic Statistics

Encoding	Sanger / Illumina 1.9
Total Sequences	8143
Total Bases	751.5 kbp
Sequences flagged as poor quality	0
Sequence length	50-101
%GC	49

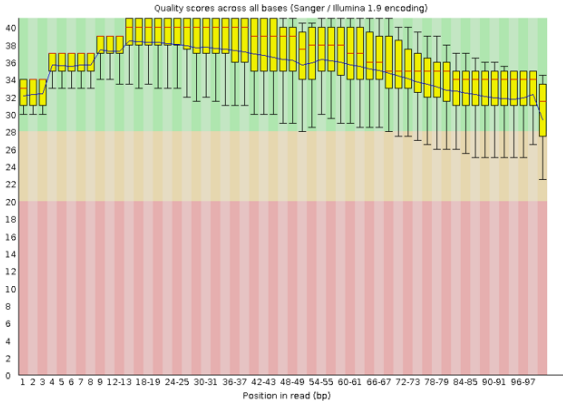
Resultados del módulo "Per base sequence quality" de FastQC.

SRR479054

READ 1



READ 2



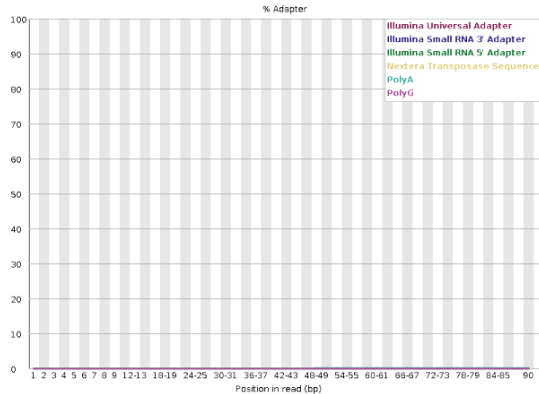
Resultados del módulo "Overrepresented sequences y adapter content" de FastQC.

SRR479054

READ 1

✔ Overrepresented sequences
No overrepresented sequences

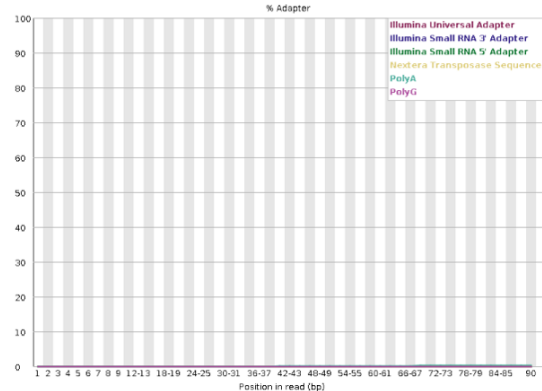
✔ Adapter Content



READ 2

✔ Overrepresented sequences
No overrepresented sequences

✔ Adapter Content



Como se puede observar en ambas muestras, ya tenemos una calidad aceptable y hemos eliminado secuencias adaptadoras en ambas muestras. Sin embargo, en la muestra SRR479052 se observa secuencias sobrerrepresentadas. Por ello, hemos utilizado dos herramientas: Fastq-screen [4] y BLAST (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>) [5].

BLAST AND FASTQ SCREEN – ANALISIS CONTAMINACIÓN

BLAST (Basic Local Alignment Search Tool) encuentra regiones de similitud entre secuencias biológicas. El programa compara secuencias de nucleótidos o proteínas con bases de datos de secuencias y calcula la importancia estadística. En la read_1 tenemos “CTTTTACTTCTCTAGATAGTCAAGTTCGACCGTCTTCTCAGCGCTCCGC” y en la read_2 tenemos “CTAACACGTGCGCGAGTCGGGGCTCGCACGAAAGCCGCGTGGCGCAAT”

Los resultados obtenidos de BLAST de la read_1 parece provenir de muestras de ratón (Mus musculus).

Job Title

Nucleotide Sequence

RID

[2URSH6XD013](#) Search expires on 04-06 19:44 pm [Download All](#) ▼

Program

BLASTN [Citation](#) ▼

Database

nt [See details](#) ▼

Query ID

lcl|Query_35055

Description

None

Molecule type

dna

Query Length

50

Other reports

[Distance tree of results](#) [MSA viewer](#) [?](#)

Filter Results

Organism

only top 20 will appear ☐ exclude

Type common name, binomial, taxid or group name

[+ Add organism](#)

Percent Identity

to

E value

to

Query Coverage

to

Filter

Reset

Descriptions

Graphic Summary

Alignments

Taxonomy

Sequences producing significant alignments

Download ▼

Select columns ▼

Show 100 ▼

[?](#)

☒ select all 100 sequences selected

[GenBank](#) [Graphics](#) [Distance tree of results](#) [MSA Viewer](#)

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
<input checked="" type="checkbox"/> Mus musculus genome assembly chromosome_18	Mus musculus	93.5	93.5	100%	1e-15	100.00%	89877872	OX439032.1
<input checked="" type="checkbox"/> Mus musculus genome assembly chromosome_16	Mus musculus	93.5	93.5	100%	1e-15	100.00%	96079412	OX439031.1

Y la del read_2 parece provenir de una secuencia de rRNA conservada en chimpancé común (Pan troglodytes)

Job Title

Nucleotide Sequence

RID

2URUNSPS013 Search expires on 04-06 19:45 pm [Download All](#) ▼

Program

BLASTN [Citation](#) ▼

Database

nt [See details](#) ▼

Query ID

lcl|Query_8517

Description

None

Molecule type

dna

Query Length

50

Other reports

[Distance tree of results](#) [MSA viewer](#) [?](#)

Filter Results

Organism only top 20 will appear ☐ exclude

Type common name, binomial, taxid or group name

[+ Add organism](#)

Percent Identity

to

E value

to

Query Coverage

to

Filter

Reset

Descriptions

Graphic Summary

Alignments

Taxonomy

Sequences producing significant alignments

Download ▼ Select columns ▼ Show 100 ▼ [?](#)

☒ select all 100 sequences selected
[GenBank](#) [Graphics](#) [Distance tree of results](#) [MSA Viewer](#)

	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per Ident	Acc Len	Accession
<input checked="" type="checkbox"/>	PREDICTED: Pan troglodytes 28S ribosomal RNA (LOC129138352) .rRNA	Pan troglodytes	93.5	93.5	100%	1e-15	100.00%	5076	XR_008541587.1
<input checked="" type="checkbox"/>	PREDICTED: Pan troglodytes 28S ribosomal RNA (LOC129137948) .rRNA	Pan troglodytes	93.5	93.5	100%	1e-15	100.00%	5074	XR_008540792.1

Para realizar un análisis más profundo, se utilizó Fastq_screen. Fastq_screen es una herramienta bioinformática que se utiliza para realizar un análisis de detección de contaminantes en datos de secuenciación. Permite detectar y cuantificar la presencia de secuencias de origen biológico o artificial, como bacterias, virus, hongos, cloroplastos, mitocondrias, entre otros. La herramienta utiliza una base de datos de referencia que contiene secuencias de genomas de organismos conocidos (tenemos que ejecutar fastqscreen --get_genomes para obtenerlo), para comparar los datos de secuenciación y detectar cualquier secuencia que coincida con las secuencias de la base de datos. Esta comparación se realiza mediante algoritmos de alineación y clasificación de secuencias. La salida de Fastq_screen proporciona una visión general de la cantidad y el tipo de contaminantes presentes en los datos de secuenciación, lo que ayuda a evaluar la calidad de los datos y a tomar decisiones sobre el tratamiento de los mismos antes de realizar análisis posteriores.

Se diseñó un script para correr fastq_screen. El script tiene como nombre **“screen.sh”** y contiene la siguiente información:

```
#!/bin/bash

#specifica el directorio donde se encuentran tus archivos fastq
FASTQ_DIR="transcriptomic-final-exercise/Apartado1/output/trimmed"

#specifies the directory where fastq_screen results will be saved
FASTQSCREEN_DIR="transcriptomic-final-exercise/Apartado1/output/fastqscreen"
mkdir -p "$FASTQSCREEN_DIR"

#in the fastqscreen folder to obtain the reference genomes.
#fastqscreen --get_genomes

#specifies the path to the fastq_screen database you want to use, in this case all of them
DATABASE="transcriptomic-final-exercise/Apartado1/output/fastqscreen/FastQ_Screen_Genomes/fastq_screen.conf"

#iteration over each pair of fastq files in the directory FASTQ_DIR
for file in $FASTQ_DIR/*_1.trimmed.fastq
do
    #get the name of the read 1 file and delete the part "_1.trimmed.fastq"
    filename=$(basename "$file")
    sample_name=${filename%%*_1.trimmed.fastq}

    #run fastq_screen
    fastq_screen --paired --outdir "$FASTQSCREEN_DIR" --aligner bowtie2 \
        --threads 8 --conf "$DATABASE" \
        "$FASTQ_DIR/${sample_name}_1.trimmed.fastq" "$FASTQ_DIR/${sample_name}_2.trimmed.fastq"
done
```

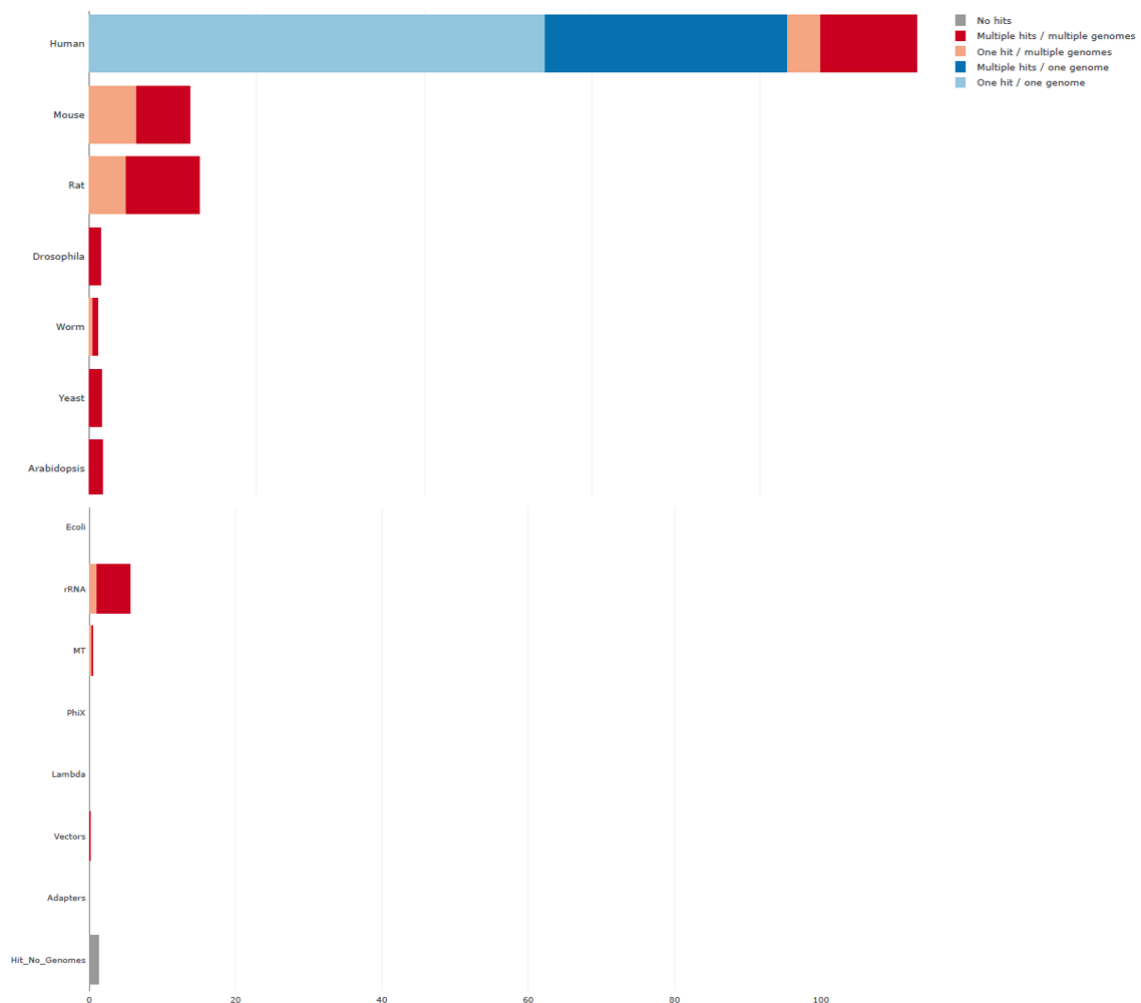
Los parámetros utilizados son los siguientes:

- --paired: indica que se están procesando archivos de lecturas pareadas.
- --outdir "\$FASTQSCREEN_DIR": indica el directorio de salida donde se guardarán los archivos de salida generados por fastq_screen.

- `--aligner bowtie2`: especifica el programa de alineación que se utilizará para el análisis. En este caso se está utilizando Bowtie2.
- `--threads 8`: indica el número de hilos que se utilizarán para el análisis.
- `--conf "$DATABASE"`: especifica el archivo de configuración que contiene los genomas de referencia para la alineación.

El comando `fastqscreen` genera dos tipos de archivos: un archivo **HTML** y un archivo de texto plano (**.txt**).

1. El archivo **HTML** (*.html) contiene un resumen visual de los resultados del análisis, incluyendo gráficos interactivos y tablas. Es útil para una rápida visualización de los resultados y para compartirlos con otros usuarios. Un ejemplo de este archivo para la muestra SRR479052 de la `read_2` (que es la que tiene secuencias sobrerrepresentadas)



2. El archivo de texto plano (*.txt) contiene los mismos resultados que el archivo HTML, pero en un formato más fácil de procesar y analizar. Este archivo incluye información detallada sobre la alineación de las secuencias con las diferentes bases de datos de referencia utilizadas por `fastqscreen`.

Las reads de ambas muestras tienen patrones similares. En FastQ Screen, "One hit" se refiere a la cantidad de lecturas que se alinearon con éxito a una sola secuencia de referencia en la base de datos de referencia. "Multiple hits" significa que una lectura se alineó a múltiples secuencias de referencia en la base de datos. "One hit / one genome" se refiere a la cantidad de lecturas que se alinearon a una sola secuencia de referencia que pertenece a un solo organismo genómico, mientras que "One hit / multiple

genomes" significa que las lecturas se alinearon a una sola secuencia de referencia que pertenece a múltiples organismos genómicos diferentes.

Se pueden observar todos los resultados de esta parte en el directorio `"/output/fastqscreen"`

La presencia de regiones sobrerrepresentadas en FastQC no siempre indica contaminación. Pueden ser el resultado de ciertas características biológicas de la muestra, como la presencia de regiones altamente transcritas o repetitivas, por lo que la conclusión de este apartado es que las secuencias sobrerrepresentadas presentes en la muestra SRR479052 no indica a priori que haya contaminación con las secuencias estudiadas (ya que no alinea únicamente en ratón), lo que indica que puede que sea regiones conservadas repetitivas que también pueden estar en otros organismos.

La conclusión de este apartado; no eliminamos esas regiones sobrerrepresentadas debido a lo mencionado anteriormente y se procede a realizar el alineamiento de las muestras.

2. Alineamiento de secuencias

El objetivo de esta etapa fue alinear las lecturas procesadas de RNA-seq contra el genoma de referencia utilizando la herramienta de alineamiento HISAT2 [6]. HISAT2 fue seleccionado debido a su alta precisión y su capacidad para minimizar el número de falsos positivos y negativos en el alineamiento. Esta está basado en la transformación de Burrows-Wheeler (BWT) y, además, HISAT2 tiene una eficiencia computacional significativamente mayor en comparación con otros alineadores y lo hemos visto en clase, por lo que me parece un alineador ideal para el procesamiento de datos de RNA-seq, aunque se pueden utilizar otro tipo de alineadores como STAR o pseudoalineadores como KALLISTO o SALMON.

Antes de realizar el proceso de alineamiento de las lecturas de RNA-seq contra el genoma de referencia, es necesario realizar un paso previo conocido como **indexado del genoma**. Este proceso permite una alineación más rápida y eficiente de las lecturas frente al genoma de referencia. En nuestro caso, se utilizó una versión del genoma humano que solo incluye el cromosoma 21 para el indexado y posterior alineamiento mediante HISAT2.

Indexado del genoma

```
#define variables
index_path='transcriptomic-final-exercise/Apartado1/index_hisat2_chr21/index'
ref_genome='transcriptomic-final-exercise/Apartado1/input/Homo_sapiens.GRCh38.dna.chromosome.21.fa'

#create folder for fastqc and index
mkdir -p transcriptomic-final-exercise/Apartado1/index_hisat2_chr21
mkdir -p "$qc_trim_path"

#check if the index exists inside the directory and if not create index
if [ "$(ls -A transcriptomic-final-exercise/Apartado1/index_hisat2_chr21)" ]; then
    echo -e "Index already built, skipping...\n"
else
    hisat2-build --seed 123 -p 8 "$ref_genome" "$index_path"
fi
```

El indexado del genoma se tiene que hacer una sola vez, por lo que se ha creado el comando anterior (en **pipeline.sh**) que comprueba si la carpeta `index_hisat2_chr21` en el path `transcriptomic-final-exercise/Apartado1/` ya contiene archivos utilizando el comando `ls -A`. Si la carpeta está vacía, lo que indica que no se ha construido previamente el índice, entonces el script utiliza la herramienta `hisat2-build` (de `hisat2`) para indexar el genoma de referencia que se encuentra en la variable `$ref_genome`. El índice se guarda en la carpeta `index_hisat2_chr21` utilizando la variable `$index_path`.

El parámetro `--seed` es la semilla y se utiliza para reproducibilidad. El parámetro `-p` se utiliza para especificar el número de hilos de procesamiento que se utilizarán, que en este caso he usado 8 (ya que

el ordenador tiene 8 cores). Si la carpeta /index_hisat2_chr21 ya contiene archivos, lo que significa que el indexado del genoma ya se ha construido previamente, entonces el script da por pantalla un mensaje indicando que el índice ya está construido y no lo creará nuevamente.

Alineamiento con hisat2

Una vez indexado el genoma de referencia humano (Chr21), se realizó el alineamiento de las muestras con hisat2. Para ello, se ejecutó un script llamado “*aligh.sh*”

```
#!/bin/bash

# Path to input directory
samples_dir="transcriptomic-final-exercise/Apartado1/output/trimmed/"

# Path to output directory
sam_dir="transcriptomic-final-exercise/Apartado1/output/sam_sample"
mkdir -p "$sam_dir"

# Loop over all files in the input directory
for file in "$samples_dir"/*.fastq; do

    # Extract the sample name from the file name
    sample=$(basename "$(basename "$file" .chr21_1.trimmed.fastq)" .chr21_2.trimmed.fastq)

    # Run HISAT2 on the sample
    echo "Aligning sample $sample ..."
    hisat2 --seed 123 -k 1 --no-spliced-alignment --new-summary -p 4 \
        -x transcriptomic-final-exercise/Apartado1/index_hisat2_chr21/index \
        -1 "$samples_dir/$sample".chr21_1.trimmed.fastq \
        -2 "$samples_dir/$sample".chr21_2.trimmed.fastq \
        -S "$sam_dir/$sample.sam" \
        --summary-file "$sam_dir/$sample.summary.txt"

done
```

El script anterior genera los archivos .sam y además genera un archivo resumen en .txt para cada una de las muestras.

- Primero definimos las variables que vamos a utilizar y definimos el path de input y output;
 - *samples_dir="transcriptomic-final-exercise/Apartado1/output/trimmed/"*: Define la ruta al directorio donde se encuentran los archivos de lecturas de RNA-Seq procesados y recortados con fastp (en la carpeta trimmed).
 - *sam_dir="transcriptomic-final-exercise/Apartado1/output/sam_sample"*: Define la ruta al directorio donde se guardarán los archivos de salida en formato SAM.
- *mkdir -p "\$sam_dir"*: Crea el directorio de salida (si no existe) donde se almacenarán los archivos de salida en formato SAM.
- *for file in "\$samples_dir"/*.fastq; do*: Inicia un bucle for que iterará sobre todos los archivos de lecturas trimeadas (SRR479052 y SRR479054, tanto read1 y read2) que se encuentren en el directorio de entrada.
- *sample=\$(basename "\$(basename "\$file" .chr21_1.trimmed.fastq)" .chr21_2.trimmed.fastq)*: Extrae el nombre de la muestra de los nombres de archivo de las lecturas trimmeadas.
- *hisat2 --seed 123 -k 1 --no-spliced-alignment --new-summary -p 4 -x transcriptomic-final-exercise/Apartado1/index_hisat2_chr21/index* -1 "\$samples_dir/\$sample".chr21_1.trimmed.fastq -2 "\$samples_dir/\$sample".chr21_2.trimmed.fastq -S "\$sam_dir/\$sample.sam" --summary-file "\$sam_dir/\$sample.summary.txt": Este comando ejecuta HISAT2 en la muestra actual, utilizando el genoma de referencia indexado previamente con HISAT2.
 - La opción **-k 1** especifica que **se permitirá como máximo una alineación por lectura**.

- La opción `--no-spliced-alignment` especifica que no se permitirán alineaciones en regiones de empalme.
- La opción `--new-summary` especifica que se utilizará un formato de salida resumido para el archivo de resumen.
- La opción `-p 4` especifica que se utilizarán 4 hilos de procesamiento.
- **-x especifica la ruta del índice del genoma de referencia.**
- **-1 y -2 especifican la ruta a las lecturas de RNA-Seq recortadas (read 1 y read 2, respectivamente).**
- **-S especifica la ruta donde se guardará el archivo de salida en formato SAM.**
- `--summary-file` especifica la ruta donde se guardará el archivo de resumen.

Control de alineamiento con hisat2

Las métricas de rendimiento del alineamiento se obtienen a partir del script anterior y se almacenan en los archivos `.summary`. Los resultados obtenidos son los siguientes:

MUESTRA SRR479052

```
HISAT2 summary stats:
  Total pairs: 12790
    Aligned concordantly or discordantly 0 time: 3682 (28.79%)
    Aligned concordantly 1 time: 8980 (70.21%)
    Aligned concordantly >1 times: 0 (0.00%)
    Aligned discordantly 1 time: 128 (1.00%)
  Total unpaired reads: 7364
    Aligned 0 time: 5029 (68.29%)
    Aligned 1 time: 2335 (31.71%)
    Aligned >1 times: 0 (0.00%)
  Overall alignment rate: 80.34%
```

MUESTRA SRR479054

```
HISAT2 summary stats:
  Total pairs: 8143
    Aligned concordantly or discordantly 0 time: 2483 (30.49%)
    Aligned concordantly 1 time: 5557 (68.24%)
    Aligned concordantly >1 times: 0 (0.00%)
    Aligned discordantly 1 time: 103 (1.26%)
  Total unpaired reads: 4966
    Aligned 0 time: 3413 (68.73%)
    Aligned 1 time: 1553 (31.27%)
    Aligned >1 times: 0 (0.00%)
  Overall alignment rate: 79.04%
```

- *Total pairs*: el número total de pares de lecturas que se procesaron en el alineamiento.
- *Aligned concordantly o discordantly 0 time*: el número de pares de lecturas que no se alinearon con éxito con el genoma de referencia.
- *Aligned concordantly 1 time*: el número de pares de lecturas que se alinearon exactamente una vez con el genoma de referencia.
- *Aligned concordantly >1 times*: el número de pares de lecturas que se alinearon con éxito más de una vez con el genoma de referencia.
- *Aligned discordantly 1 time*: el número de pares de lecturas que se alinearon discordantemente exactamente una vez con el genoma de referencia.
- *Total unpaired reads*: el número total de lecturas no pareadas que se procesaron en el alineamiento.
- *Aligned 0 time*: el número de lecturas no pareadas que no se alinearon con éxito con el genoma de referencia.

- *Aligned 1 time*: el número de lecturas no pareadas que se alinearon exactamente una vez con el genoma de referencia.
- *Aligned >1 times*: el número de lecturas no pareadas que se alinearon con éxito más de una vez con el genoma de referencia.
- *Overall alignment rate*: la tasa de alineación general para todas las lecturas, tanto pareadas como no pareadas.

En conclusión;

1. Para la muestra SRR479052 la tasa de alineación general es del 80,34%, lo que significa que el 80,34% de las lecturas se alinearon con éxito con el genoma de referencia.
2. Para la muestra SRR479054 la tasa de alineación general es del 79,04%, lo que significa que el 79,04% de las lecturas se alinearon con éxito con el genoma de referencia.

3. Posprocesamiento con SAMTOOLS y estadísticas

El siguiente paso fue convertir los archivos SAM generados por HISAT2 en archivos BAM ordenados y luego indexarlos, gracias a la herramienta **samtools** [7]. A continuación, se reportaron nuevas estadísticas del alineamiento gracias a **samtools** [7].

Samtools es una suite de herramientas de bioinformática para procesar y manipular archivos **SAM/BAM** (Sequence Alignment/Map). SAM es un formato de archivo de texto plano que contiene la información de alineamiento de las lecturas de secuenciación a una referencia genómica, en este caso a la referencia del chr 21 del genoma humano GRCh38. BAM es una versión binaria del formato SAM que se utiliza para el almacenamiento de grandes volúmenes de datos de alineamiento de secuencias. He utilizado samtools debido a que permite filtrar, ordenar y manipular archivos BAM y SAM, así como convertir archivos de un formato a otro. Además, he utilizado argumentos de samtools, que explicaré posteriormente, para extraer estadísticas de alineamiento.

Para esta parte se ha generado dos script; **preproc.sh** (1) y **stadistic.sh** (2).

(1)

```
#!/bin/bash

# Path to input directory containing SAM files
sam_dir="transcriptomic-final-exercise/Apartado1/output/sam_sample/"

# Path to output directory
bam_dir="transcriptomic-final-exercise/Apartado1/output/bam_sample/"
mkdir -p "$bam_dir"

# Convert SAM files to BAM files and sort them
for file in "$sam_dir"/*.sam; do
    # Extract the sample name from the file name
    sample=$(basename "$file" .sam)

    # Convert the SAM file to a sorted BAM file
    echo "Converting and sorting SAM file $sample ..."
    samtools view -@ 5 -bS "$file" | samtools sort -@ 5 -o "${bam_dir}/${sample}.sorted.bam"
    samtools index "${bam_dir}/${sample}.sorted.bam"
    # rm "$sam_dir/$sample.sam"
done
```

En este script primero, se definió la ruta del directorio que contiene los archivos SAM y se estableció el directorio de salida donde se almacenarán los archivos BAM ordenados. Luego, mediante un bucle "for", se recorren los archivos SAM uno a uno (de las dos muestras) y se extrajo el nombre de muestra del nombre del archivo. Posteriormente, se utilizó la herramienta **samtools** para convertir cada archivo SAM a un archivo BAM ordenado. **La opción -b indica que se desea obtener un archivo BAM en lugar de SAM, mientras que la opción -S especifica que la entrada es un archivo SAM.** A continuación, se utiliza

la función **samtools sort** para ordenar el archivo BAM y guardarlo en el directorio de salida especificado. Finalmente, **se indexa el archivo BAM utilizando la función samtools index**.

Adicionalmente, se puede observar que la línea `# rm "$sam_dir/$sample.sam"` está comentada, lo que indica que no se está eliminando el archivo SAM original después de la conversión a BAM. Sin embargo, esta línea se puede descomentar si se desea eliminar los archivos SAM para ahorrar espacio, aunque de manera adicional cree un script final llamado **remove.sh** que elimina los archivos SAM.

(2)

```
#!/bin/bash
mkdir -p transcriptomic-final-exercise/Apartado1/output/stadistic
for file in transcriptomic-final-exercise/Apartado1/output/sam_sample/SRR479052.sam transcriptomic-final-exercise/Apartado1/output/sam_sample/SRR479054.sam
do
    echo "Archivo: ${file}"
    base=$(basename ${file})
    samtools flagstat ${file} > transcriptomic-final-exercise/Apartado1/output/stadistic/${base}.flagstat.txt
    samtools stats ${file} > transcriptomic-final-exercise/Apartado1/output/stadistic/${base}.stats.tsv
done
```

Este script, llamado **"stadistic.sh"** contiene información para realizar las estadísticas de alineamiento con la herramienta **samtools**. En el caso de que haya más muestras, se tiene que modificar el script, ya que como entrada el bucle **"for"** solo tiene en cuenta las muestras SRR479052 y SRR479054. ¡Esto lo tenemos que tener en cuenta en el caso de que se aumente el número de muestras en el estudio!.

La herramienta **samtools** se utiliza en este script para generar estadísticas a partir de los archivos SAM. En este caso se han utilizado dos comandos: **flagstat** y **stats**.

- El comando **samtools flagstat** calcula varias estadísticas básicas para el archivo SAM, incluyendo el número total de lecturas, el número de lecturas alineadas y el porcentaje de lecturas alineadas.
- El comando **samtools stats** proporciona estadísticas más detalladas, como la distribución de la longitud de las lecturas, la calidad de la base y otros aspectos de la alineación.

Estas estadísticas se guardan en archivos de texto separados para cada archivo SAM en un directorio de salida especificado. En el script, los archivos de salida se guardan en el directorio `transcriptomic-final-exercise/Apartado1/output/stadistic/`, con el nombre del archivo de entrada como prefijo y una extensión diferente para cada estadística generada (`.flagstat.txt` para el comando **flagstat** y `.stats.tsv` para el comando **stats**). Los resultados de `stat.tsv` se muestra en el Anexo.

MUESTRA SRR479052

```
25580 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
20551 + 0 mapped (80.34% : N/A)
25580 + 0 paired in sequencing
12790 + 0 read1
12790 + 0 read2
17960 + 0 properly paired (70.21% : N/A)
18488 + 0 with itself and mate mapped
2063 + 0 singletons (8.06% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

- "25580 + 0 in total": indica que hay un total de 25,580 fragmentos de lectura (pares de lectura) en el archivo.
- "0 + 0 secondary" y "0 + 0 supplementary": se refiere a los fragmentos de lectura que son secundarios o complementarios en el archivo, que en este caso no hay ninguno.
- "0 + 0 duplicates": indica que no hay fragmentos de lectura duplicados en el archivo.

- "20551 + 0 mapped": significa que de los 25,580 fragmentos de lectura, 20,551 están correctamente alineados en el genoma de referencia (80.34%).
- "25580 + 0 paired in sequencing": indica que todos los fragmentos de lectura son pares (en este caso, de tipo Illumina).
- "12790 + 0 read1" y "12790 + 0 read2": se refiere al número de lecturas que provienen del primer y segundo extremo del fragmento, respectivamente.
- "17960 + 0 properly paired": indica el número de fragmentos de lectura en los que tanto el primer como el segundo extremo están correctamente alineados y tienen la orientación correcta en relación con el genoma de referencia.
- "18488 + 0 with itself and mate mapped": significa que el primer extremo y su pareja (mate) están correctamente alineados en el genoma de referencia.
- "2063 + 0 singletons": se refiere a los fragmentos de lectura en los que solo un extremo se alinea correctamente en el genoma de referencia.
- "0 + 0 with mate mapped to a different chr" y "0 + 0 with mate mapped to a different chr (mapQ>=5)": se refiere al número de fragmentos de lectura en los que el extremo y su pareja se alinean en diferentes cromosomas. En este caso no hay ninguno.

MUESTRA SRR479054

```
16286 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
12873 + 0 mapped (79.04% : N/A)
16286 + 0 paired in sequencing
8143 + 0 read1
8143 + 0 read2
11114 + 0 properly paired (68.24% : N/A)
11514 + 0 with itself and mate mapped
1359 + 0 singletons (8.34% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

- "16286 + 0 in total": este es el número total de lecturas en el archivo BAM.
- "0 + 0 secondary" y "0 + 0 supplementary": estas son las lecturas secundarias y complementarias, que no se incluyen en el recuento total.
- "0 + 0 duplicates": el número de lecturas duplicadas encontradas en el archivo BAM. En este caso, no se encontraron duplicados.
- "12873 + 0 mapped": el número de lecturas mapeadas correctamente al genoma de referencia. En este caso, 12873 lecturas mapearon correctamente, lo que representa el 79.04% del total de lecturas.
- "16286 + 0 paired in sequencing": el número de pares de lecturas en el archivo BAM. En este caso, hay 16286 pares de lecturas.
- "8143 + 0 read1" y "8143 + 0 read2": el número de lecturas en la primera y segunda posición de cada par, respectivamente.
- "11114 + 0 properly paired": el número de pares de lecturas que se alinearon correctamente y que están en la orientación correcta.
- "11514 + 0 with itself and mate mapped": el número de pares de lecturas que se alinearon correctamente a la misma región del genoma.
- "1359 + 0 singletons": el número de lecturas que se alinearon correctamente, pero que no tienen una pareja en el archivo.
- "0 + 0 with mate mapped to a different chr" y "0 + 0 with mate mapped to a different chr (mapQ>=5)": el número de pares de lecturas que se alinearon a diferentes cromosomas o regiones del genoma con una puntuación de mapeo de al menos 5. En este caso, no hay ningún par de lecturas que se hayan alineado a diferentes cromosomas o regiones del genoma.

4. Conteo de lecturas con featureCounts

En clase hemos visto HTSeq. Sin embargo, he tenido problemas para descargar HTSeq, por lo que he optado por hacer el conteo de las lecturas con otra herramienta de cuantificación llamada **featureCounts** [8].

He estado leyendo y algunas publicaciones determinan que **featureCounts** es más rápido que HTSeq en términos de tiempo de ejecución. Además, **featureCounts** puede realizar el recuento de lecturas a nivel de subcaracterística, lo que permite el recuento de lecturas de exones individuales y que admite una amplia variedad de tipos de datos de entrada. Por todo lo mencionado anteriormente, realicé el conteo de lecturas con **featureCounts**.

Para ello, se ha creado un script que tiene como nombre “**featurecounts.sh**”

```
#!/bin/bash

# Path to the annotation file
annotation_file="transcriptomic-final-exercise/Apartado1/input/Homo_sapiens.GRCh38.109.chr21.gtf"

# Path to the output directory
counts_dir="transcriptomic-final-exercise/Apartado1/output/counts"
mkdir -p "$counts_dir"

# Path to the sorted BAM files
bam_file1="transcriptomic-final-exercise/Apartado1/output/bam_sample/SRR479052.sorted.bam"
bam_file2="transcriptomic-final-exercise/Apartado1/output/bam_sample/SRR479054.sorted.bam"

# Run featureCounts
echo "Counting features for BAM file 1 ..."
featureCounts -p -T 4 -B --countReadPairs -t exon -g gene_id -a "$annotation_file" -o "$counts_dir/SRR479052_counts.tsv" "$bam_file1"

echo "Counting features for BAM file 2 ..."
featureCounts -p -T 4 -B --countReadPairs -t exon -g gene_id -a "$annotation_file" -o "$counts_dir/SRR479054_counts.tsv" "$bam_file2"
```

La ruta al archivo de anotación GTF (annotation_file) y la ruta al directorio de salida (counts_dir) se establecen al principio del script. Luego, se definen las rutas de los dos archivos BAM ordenados e indexados que se van a analizar (bam_file1 y bam_file2). Y luego he generado dos comandos, uno por cada muestra (en el caso de que se aumente el número de muestras estos scripts tienen que modificarse). En ambos casos se utilizan las siguientes opciones:

- -p: indica que el análisis se realizará en paralelo.
- -T 4: especifica el número de hilos/cores que se utilizarán (en este caso 4).
- -B: indica que se contarán las lecturas apareadas.
- --countReadPairs: indica que se contará cada par de lecturas apareadas una sola vez.
- -t exon: especifica que se contarán las características a nivel de exón.
- -g gene_id: indica que se agruparán las características por gene_id.
- -a "\$annotation_file": especifica la ruta al archivo de anotación GTF.
- -o "\$counts_dir/SRR479052_counts.tsv" o -o "\$counts_dir/SRR479054_counts.tsv": indica la ruta de salida del archivo de conteo para el archivo BAM correspondiente.

NO se ha indicado el parámetro -s, ya que desconocemos la orientación de las lecturas. **FeatureCounts** utilizará la configuración predeterminada, que es 0, lo que significa que todas las lecturas se contarán independientemente de su orientación.

Finalmente, se ejecutó el script “**merged.sh**”;

```
#!/bin/bash

output_dir="transcriptomic-final-exercise/Apartado1/output"
counts_dir="$output_dir/counts"
merged_dir="$counts_dir/merged"
mkdir -p "$merged_dir"

cut -f1 "$counts_dir/SRR479052_counts.tsv" | tail -n +1 > "$merged_dir/common_column.txt"
tail -n +1 "$counts_dir/SRR479052_counts.tsv" > "$merged_dir/SRR479052_counts_no_header.tsv"
tail -n +1 "$counts_dir/SRR479054_counts.tsv" > "$merged_dir/SRR479054_counts_no_header.tsv"
paste "$merged_dir/common_column.txt" "$merged_dir/SRR479052_counts_no_header.tsv" "$merged_dir/SRR479054_counts_no_header.tsv" | cut -f1,8,15 > "$merged_dir/merged_counts.tsv"
```

Este script tiene como objetivo unir los resultados de conteo de expresión generados por **featureCounts** para los dos archivos BAM procesados en una única tabla de conteo. En primer lugar, se define la ruta de salida y el directorio donde se almacenarán los conteos combinados. Luego, se crea un archivo

llamado "common_column.txt" que contiene la columna de ID de gen común a ambos archivos. Para hacer esto, se extrae la primera columna del archivo de conteo de uno de los archivos y se la guarda en un archivo separado. A continuación, se eliminan las filas de encabezado de los archivos de conteo y se los guarda en archivos separados. Después, se combina el contenido del archivo "common_column.txt" con los dos archivos de conteo que no tienen encabezado, usando el comando "paste". Por último, se utiliza el comando "cut" para seleccionar las columnas de ID de gen y los recuentos de expresión correspondientes a los dos archivos BAM procesados y se los guarda en un archivo llamado "merged_counts.tsv" dentro del directorio "merged".

5. Métricas de calidad del experimento con multiQC

Finalmente, se realizó un análisis de calidad final con **multiQC** [9]. **MultiQC** es una herramienta que permite generar un informe de calidad final a partir de los resultados obtenidos en diferentes herramientas de análisis de datos. MultiQC recopila la información de calidad y rendimiento de diversas herramientas de análisis, como FastQC, HISAT2, etc., y genera un informe HTML con gráficos y tablas que permiten una rápida visualización y evaluación de la calidad de los datos.

```
#!/bin/bash
mkdir -p transcriptomic-final-exercise/Apartado1/output/multiqc
multiqc . -o transcriptomic-final-exercise/Apartado1/output/multiqc/multiqc_report
```

Al ejecutar el MultiQC, se recopila información de los programas que se han usado y se genera un reporte con todas las estadísticas, es muy útil para ver de una manera visual y en un solo vistazo toda la información que se obtiene del análisis. *El reporte está en /multiqc_report.html.*

CONCLUSIONES

Tras realizar un control de calidad con FastQC, se ha detectado la presencia de adaptadores, secuencias sobrerrepresentadas y extremos de baja calidad, lo que indica la necesidad de realizar preprocesamiento (cutadapt, fastp, trimmomatic...) y análisis de contaminantes (fastq_screen). Una vez obtenido las secuencias con una calidad aceptable, se realizó el alineamiento con hisat2 obteniendo una tasa de alineación del 79-80%.

Para mejorar los resultados se podría probar a alinear con diferentes alineadores o pseudoalineadores para ver cuál obtiene un mejor rendimiento, o utilizar otras herramientas de cuantificación como HT-seq. Además de probar diferentes parámetros que están descritos en los manuales.

Apartado 2. PREGUNTA 4. Introducción y creación del objeto dds.

En este caso nos han proporcionado inputs independientes del apartado 1 para realizar la expresión diferencial y GSEA:

- La matriz de cuentas crudas para los 24 cultivos analizados.
- Data frame con los metadatos asociados al experimento.
- GMT para realizar un GSEA.

Para llevar a cabo esta sección, se ha creado un nuevo entorno virtual llamado "trabajo_RNAseq_2" y se han copiado las dependencias del entorno virtual "dge_lessons". Se abrió Rstudio y se creó un nuevo archivo de tipo R-markdown, el cual fue guardado para su posterior uso.

En esta sesión de R, se activaron las librerías necesarias (entre ellas `Deseq2`[10]) y se realizó la importación de los **raw.counts**, y **metadata** para generar el objeto **DESeqDataSet**, que es requerido por el paquete `DESeq2` para el análisis de expresión diferencial. Previamente, se comprobó si las columnas de 'countData' tienen el mismo número de filas que 'colData'. Si devuelve TRUE seguimos con la creación del objeto dds.

El código '`dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData, design = ~ patient + agent)`' es utilizado para crear un objeto de tipo `DESeqDataSet`. La función `DESeqDataSetFromMatrix()` toma como argumentos:

- **countData**: una matriz con los recuentos de lecturas para cada gen y muestra.
- **colData**: un data frame con información de metadatos sobre cada muestra, como las condiciones experimentales y factores que pueden influir en la expresión génica.
- **design**: una fórmula que especifica cómo se modela la relación entre las condiciones experimentales y la expresión génica.

En este caso, la fórmula `~ patient + agent` especifica que se utilizarán dos factores para modelar la variación en la expresión génica: **patient** y **agent**, ya que sólo queremos saber lo que ocurre en las muestras de 24h y no tenemos en cuenta el tiempo, ya que vamos a hacer un subset de los datos. El factor **patient** se refiere a los pacientes de los cuales se obtuvieron las muestras, y el factor **agent** se refiere a los tratamientos (OHT y DNT).

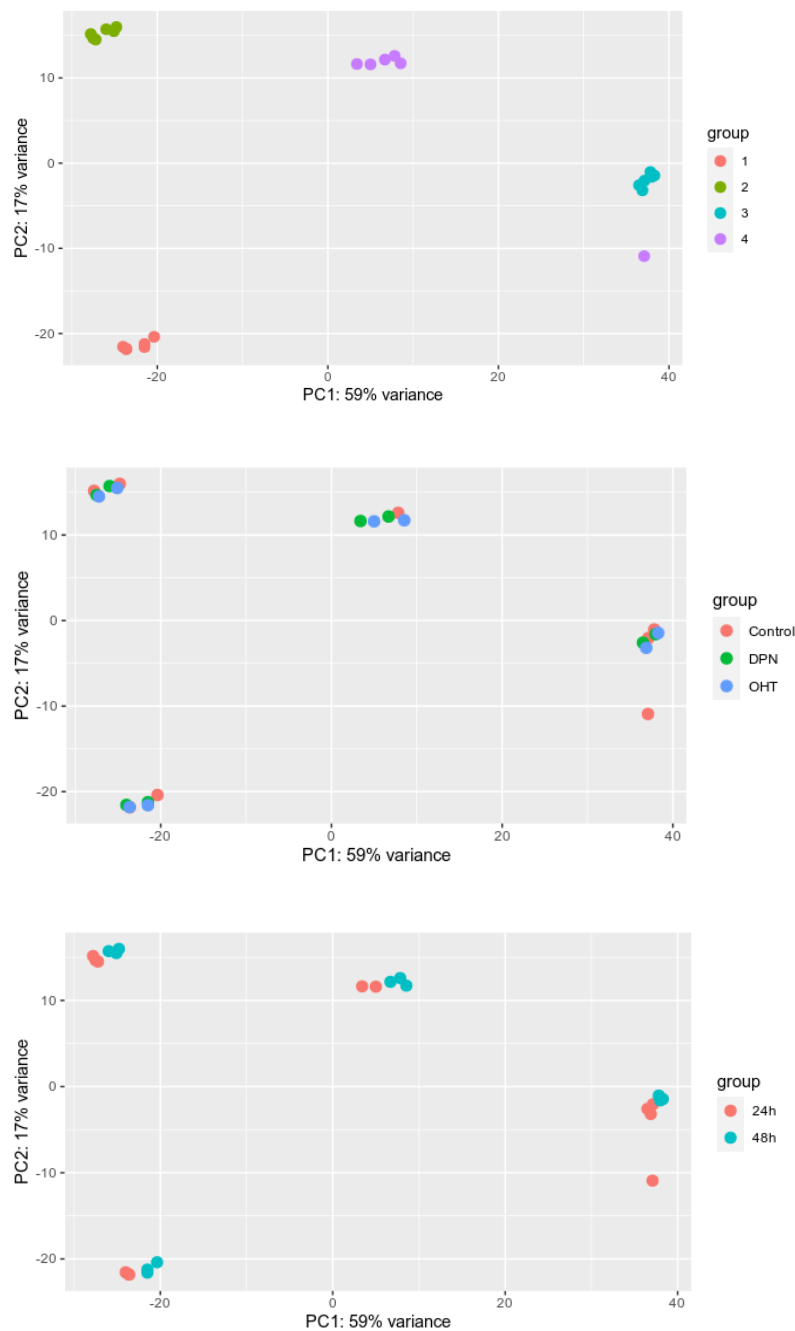
Una vez generado el objeto de `DESeq2`, se realizó un análisis exploratorio de los datos, para ello se tiene que reducir la heterocedasticidad de los datos y lo realizamos con la función **vst**. **VST (Variance Stabilizing Transformation)** es una técnica utilizada en el análisis de datos de expresión génica para reducir la heterocedasticidad (varianza no constante) de los datos, lo que puede mejorar la detección de genes diferencialmente expresados. La transformación **vst** utiliza una función matemática para estabilizar la varianza y hacer que sea constante en todo el rango de expresión. La transformación **vst** se basa en el modelo Poisson y una función logarítmica, por lo que es adecuada para datos de conteo como los de RNA-seq.

Después de aplicar la transformación **vst**, se realizó el análisis exploratorio de los datos a través de **PCA (Análisis de Componentes Principales)** y **análisis de distancia** para visualizar la similitud o diferencia entre muestras y para identificar patrones en los datos de expresión génica.

1. Análisis exploratorio: análisis de componentes principales (PCA) y análisis de distancia.

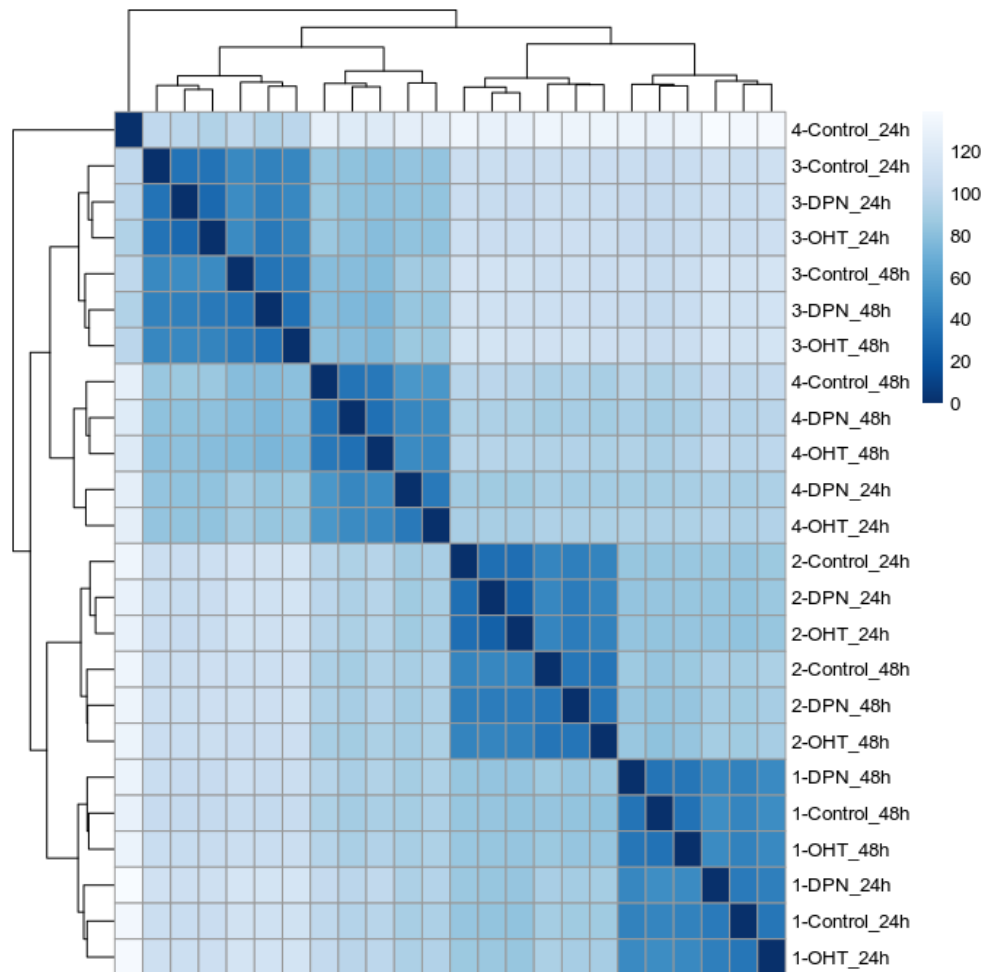
El análisis de componentes principales (PCA) se fundamenta en la generación de componentes (PC) que son combinaciones lineales y ortogonales de las variables experimentales originales, y además son no correlacionadas o independientes entre sí. Este enfoque proporciona una descripción de un conjunto de datos en términos de las componentes creadas previamente, basadas en la expresión de cada elemento genómico. Se lleva a cabo a partir de una matriz normalizada, en la cual los datos se escalan y centran, y las componentes se ordenan en función de la cantidad de varianza explicada. Es decir, la primera componente, PC1, explica una proporción determinada de la varianza total, mientras que la segunda componente, PC2, explica el porcentaje de la varianza que no ha sido explicada por la primera componente, y así sucesivamente. Por lo tanto, **el PCA es una técnica exploratoria que permite reducir la dimensionalidad de un conjunto de datos y es una herramienta increíblemente útil para verificar valores atípicos y efectos por lotes.**

Los resultados obtenidos se muestran a continuación en un gráfico bidimensional:



Las muestras fueron coloreadas según el paciente, tratamiento y tiempo. **Se observó que las muestras se agrupan principalmente en función del tipo de paciente, lo que sugiere la existencia de factores biológicos relevantes que contribuyen a las diferencias observadas entre ellas. No obstante, se ha identificado una muestra en particular del paciente 4 que se aleja significativamente de las demás muestras del mismo paciente, lo que indica la posibilidad de algún tipo de problema técnico o biológico en dicha muestra en particular.**

Después se realizó el plot del análisis de distancia. `sampleDistMatrix` se utiliza como entrada para generar un gráfico de mapa de calor con la función `heatmap`, en el que **las muestras se agrupan por similitud en la expresión génica.**



Se identificó que los pacientes 1 y 2 parecen estar más relacionados entre sí, mientras que los pacientes 3 y 4 están más relacionados entre ellos. La presencia del outlier (paciente 4 control 24h) también se confirmó por este método y se encontró que se agrupa más cercano a los datos del paciente 3 que a los de su propio grupo, tal como se demostró en el análisis PCA.

Como conclusión: la muestra del paciente 4 control a las 24 horas es un **outlier**.

2. Filtrado de los datos y análisis exploratorio tras el filtrado.

Cómo en el enunciado nos piden solamente los genes diferencialmente expresados entre OHT vs control y DPN vs control en las muestras tratadas 24horas, se utilizó el siguiente código;

```
keep <- rowSums(counts(dds)) >= 10 ## Select genes with more than 10 counts in all samples
```

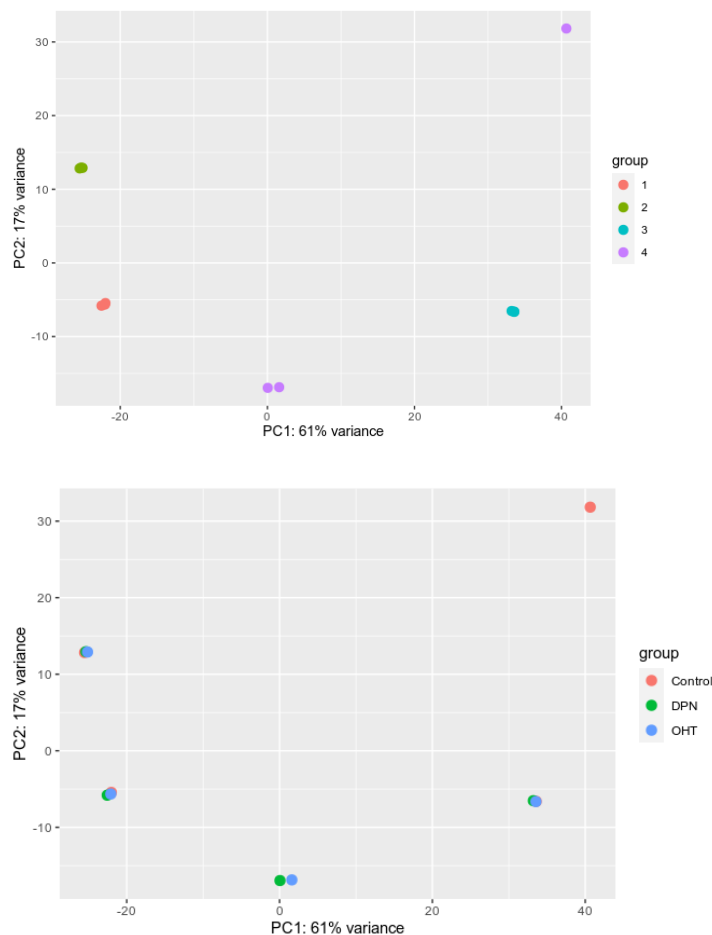
```
dds_filter <- dds[keep, ]
```

```
dds_24 <- dds_filter[, dds_filter$time == "24h"]
```

Este código es responsable de filtrar el conjunto de datos original DESeqDataSet y seleccionar únicamente aquellos genes que tienen un conteo mínimo de 10 en cada muestra. Los genes que cumplen este criterio se mantienen en la variable "**keep**". Luego, se creó un nuevo objeto DESeqDataSet llamado "dds_filter" que contiene solamente los genes que cumplen con el criterio de recuento mínimo. En la práctica habitual el descartar genes y/o transcritos con muy pocas o ninguna lectura a lo largo de todas las muestras es común, ya que estos no tienen relevancia en la expresión diferencial y además, reduce el procesamiento computacional de las muestras.

Finalmente, se seleccionan todas las muestras de "dds_filter" que corresponden a un tiempo de tratamiento de 24 horas y se almacenan en un objeto llamado "dds_24".

Antes de realizar el análisis exploratorio, se volvió a realizar la normalización de las matrices de expresión mediante **vst** (variance stabilizing transformation algorithm) de DESeq2. Los resultados fueron los siguientes:



Durante este análisis exploratorio, **se identificó que la muestra 24h control del paciente 4 era un outlier**, lo que implica que difiere significativamente de las demás muestras del mismo paciente y puede afectar la precisión de los resultados. Al excluir esta muestra atípica, se evita cualquier posible sesgo en los resultados y se garantiza que no estén influenciados por esta muestra fuera de lo común.

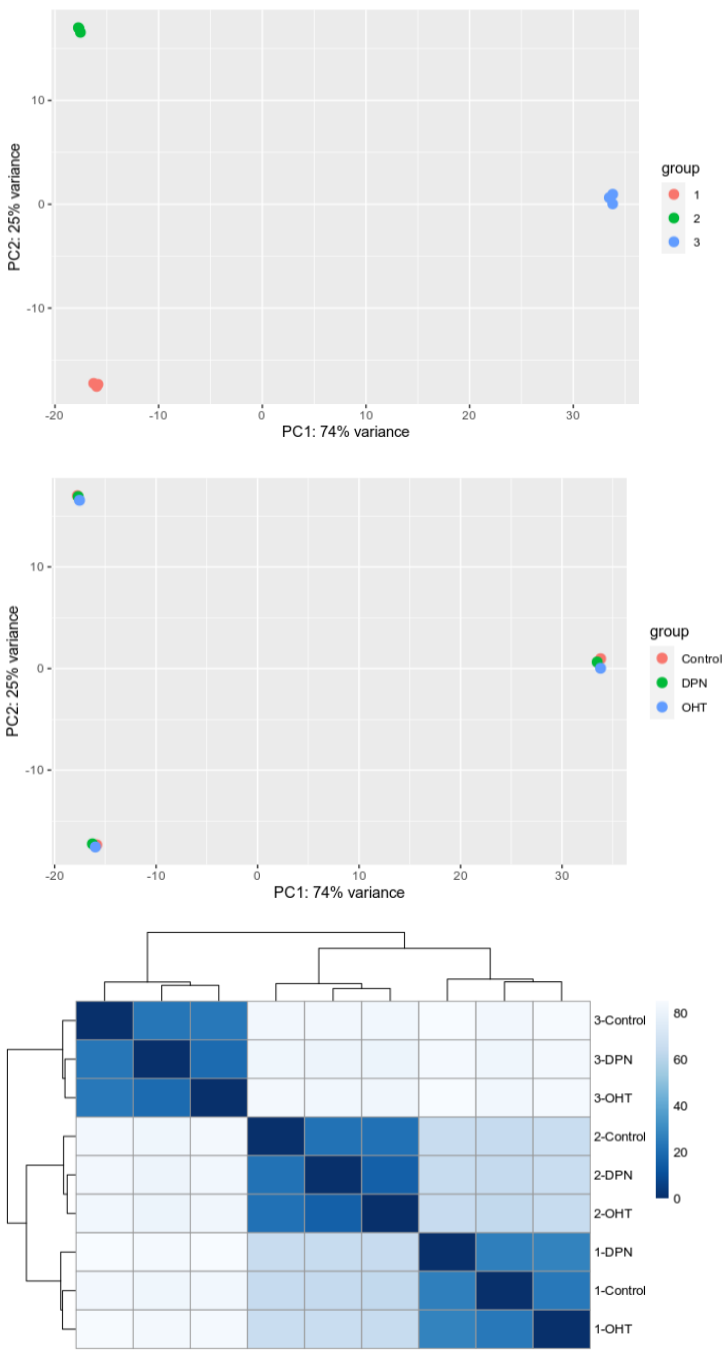
Es importante destacar que, a pesar de que el número de muestras es limitado, la eliminación de esta muestra puede tener un impacto significativo en el análisis. Por lo tanto, para obtener resultados más robustos, **se necesitaría un mayor número de muestras**. No obstante, en este caso, he tomado la

decisión de ser rigurosos y eliminar la muestra para garantizar la precisión y fiabilidad de los resultados obtenidos.

CONCLUSION: ELIMINACIÓN DEL PACIENTE 4.

Al eliminar esta muestra, se puede reducir el impacto de este posible sesgo en los resultados y mejorar la precisión de los resultados generales. Por supuesto, es importante tener en cuenta **que el número de muestras en este análisis es pequeño**, y la eliminación de una muestra podría tener un gran impacto en los resultados. Por lo tanto, es importante ser cuidadoso al eliminar muestras y considerar si se necesitan muestras adicionales para garantizar que los resultados sean lo más precisos y confiables posible.

Al eliminar la muestra, se volvió a realizar la transformación **vst** y se realizó un análisis exploratorio y los resultados fueron los siguientes;



Por lo tanto, parece que todo está listo para realizar el análisis de expresión diferencial con DESeq2 y como conclusión del análisis exploratorio podemos destacar que los datos se agrupan por su similitud en relación a los **pacientes**.

3. Análisis de expresión diferencial.

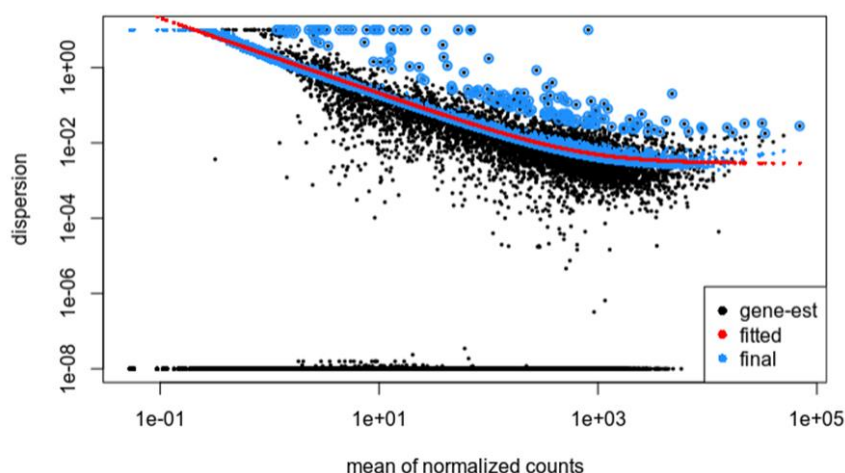
Para el análisis de expresión diferencial se usó el paquete mencionado anteriormente llamado **DESeq2** [10]. **DESeq2** asume que los recuentos de lecturas siguen una distribución binomial negativa. La estimación la expresión diferencial entre dos condiciones se llevó a cabo mediante el **estadístico de Wald** que evalúa si la expresión de genes/transcritos se ajusta una la distribución binomial negativa, determinando si la medida de efecto, log2FC (logaritmo en base dos del fold-change), es significativamente diferente de cero. En este análisis, para el problema de comparaciones múltiples (multiple testing), se ajustó el p-value mediante el procedimiento de Benjamini-Hochberg (BH).

Se realiza el análisis de expresión diferencial utilizando el método DESeq y se especifica que se quiere utilizar la prueba de Wald. `dds_24_filt <- DESeq(dds_24_filt, test = "Wald")`. Este análisis se realiza con los datos filtrados previamente para incluir solo aquellos genes con más de 10 conteos en todas las muestras.

DESeq realiza los siguientes pasos principales:

1. Estimación de los factores de escala o size factors: los factores de escala se utilizan para ajustar las diferencias en la profundidad de secuenciación entre las muestras. Esto permite una comparación más precisa de los niveles de expresión entre las muestras. DESeq utiliza una mediana geométrica para calcular los factores de escala.
2. Estimación de la dispersión: la dispersión se refiere a la variabilidad de los datos entre las muestras. DESeq utiliza un modelo de Binomial negativo para estimar la dispersión.
3. Ajuste del modelo de GLM - Binomial negativo: DESeq utiliza un modelo de regresión lineal generalizado (GLM) para ajustar los datos a un modelo de Binomial negativo. Este modelo tiene en cuenta los factores de escala y la dispersión.

Posteriormente con la función `plotDispEst(dds_24_filt)` se realizó un gráfico de dispersión para estimar la variabilidad biológica y técnica de los datos.



3.1. Análisis de expresión diferencial: OHT vs Control

Para comprobar el efecto del tratamiento con OHT a 24 horas se realizó un análisis de expresión diferencial utilizando el modelo lineal y enfoque de contrasta para identificar los genes que están

significativamente regulados en el grupo tratado con OHT en comparación con el grupo control. El comando utilizado fue:

```
res_OHT <- results(dds_24_filt, contrast = c("agent", "OHT", "Control"),  
  lfcThreshold = 1,  
  alpha=0.05,  
  pAdjustMethod = "BH")  
  
res_OHT$symbol <- mapIds(org.Hs.eg.db, keys = row.names(res_OHT),  
  keytype = "ENSEMBL", column = "SYMBOL")
```

res_OHT

summary(res_OHT)

Se utilizó el objeto “dds_24_filt” como entrada, que es el objeto DESeqDataSet creado previamente. El argumento contrast especifica el grupo de tratamiento ("OHT") y el grupo de control ("Control") para realizar el contraste de interés, donde "agent" es el nombre del factor en el que se basa la comparación. El argumento lfcThreshold especifica el umbral de cambio de plegamiento mínimo para considerar un gen como diferencialmente expresado, que en este caso se utilizó 1. El argumento alpha especifica el nivel de significación utilizado para ajustar los p-valores, mientras que el argumento pAdjustMethod especifica el método de corrección de múltiples comparaciones utilizado para ajustar los p-valores, en este caso hemos utilizado corrección de Benjamini-Hochberg. El resultado del análisis se guarda en el objeto res_OHT.

Posteriormente, se agrego una columna de símbolo de genes al objeto res_OHT y se observo un resumen con el comando “summary”.

El parámetro lfcThreshold establecido en 1 indica que solo se considerarán como diferencialmente expresados aquellos genes que tienen un cambio de plegamiento (log2FoldChange) de al menos 1 unidad. Esto significa que solo los genes cuya expresión se haya aumentado o disminuido al menos 2 veces entre los grupos de tratamiento y control se considerarán significativamente regulados. Se ha elegido un valor de 1, ya que cuando se tienen pocas muestras, es importante ser más conservador en la elección del umbral de cambio de plegamiento (lfcThreshold), ya que las estimaciones de la expresión génica pueden ser más ruidosas debido a la variabilidad biológica y técnica inherente a los datos de RNA-seq. Por lo tanto, es recomendable utilizar un umbral de cambio de plegamiento más alto para minimizar la cantidad de falsos positivos.

El parámetro alpha establecido en 0.05 es el nivel de significación utilizado para ajustar los p-valores. Significa que se considerarán significativos aquellos genes que tengan un valor de p ajustado por debajo de 0.05.

'select()' returned 1:many mapping between keys and columns

log2 fold change (MLE): agent OHT vs Control

Wald test p-value: agent OHT vs Control

DataFrame with 24416 rows and 7 columns

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<character>
ENSG000000000003	599.90666	0.0891077	0.106523	0	1	1	TSPAN6
ENSG000000000005	1.46579	-0.2001296	1.877963	0	1	1	TNMD
ENSG000000000419	264.84588	-0.1866093	0.135852	0	1	1	DPM1
ENSG000000000457	195.95228	-0.1460213	0.152427	0	1	1	SCYL3
ENSG000000000460	486.02888	0.1444583	0.112234	0	1	1	C1orf112
...
ENSG00000257097	2.421169	1.205788	1.356182	0.151741	0.879391	1	NA
ENSG00000257104	0.334348	-0.507235	3.472882	0.000000	1.000000	1	NA
ENSG00000257106	0.903789	-0.746293	2.197893	0.000000	1.000000	1	NA
ENSG00000257108	27.409934	0.157528	0.387442	0.000000	1.000000	1	NHLRC4
ENSG00000257112	8.027149	0.293747	0.686671	0.000000	1.000000	1	NA

out of 24392 with nonzero total read count

adjusted p-value < 0.05

LFC > 1.00 (up) : 0, 0%

LFC < -1.00 (down) : 0, 0%

outliers [1] : 0, 0%

low counts [2] : 0, 0%

(mean count < 0)

[1] see 'cooksCutoff' argument of ?results

[2] see 'independentFiltering' argument of ?results

Como hemos sido muy restrictivos y conservadores, los resultados muestran que **NO existen genes diferencialmente expresados para la comparativa OHT vs Control**, según este análisis. Sin embargo, podemos jugar con los parámetros como el "Alpha" para ser menos restrictivos, ya que una elección de Alpha de 0,05 es restrictiva y puede aumentar la tasa de falsos negativos y perder genes diferencialmente expresados importantes.

NOTA: Los resultados obtenidos van a depender en gran medida del investigador del estudio, deberíamos tener reuniones con él para saber que es lo que quiere, cuales son sus objetivos y podemos modificar los parámetros según lo que el investigador nos indique. El investigador debe ser quien decida, nosotros como bioinformáticos podemos aconsejar, pero creo que finalmente es el investigador el que conoce el estudio, como se ha realizado la parte técnica, las limitaciones y fortalezas del estudio y debería ser él el que nos indique la pregunta biológica que esta abordando, la complejidad del sistema biológico que esta estudiando y la cantidad de datos disponibles. Quizás, con un numero de muestras mayor y con los parámetros que hemos utilizado si lleguemos a ver genes diferencialmente expresados.

Finalmente, se generó un mapa de calor (heatmap) para visualizar la expresión génica de los 30 genes más significativamente diferencialmente expresados en la comparación entre el grupo de tratamiento (OHT) y el grupo de control.


```

out of 24392 with nonzero total read count
adjusted p-value < 0.05
LFC > 1.00 (up) : 0, 0%
LFC < -1.00 (down) : 0, 0%
outliers [1] : 0, 0%
low counts [2] : 0, 0%
(mean count < 0)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

```

```

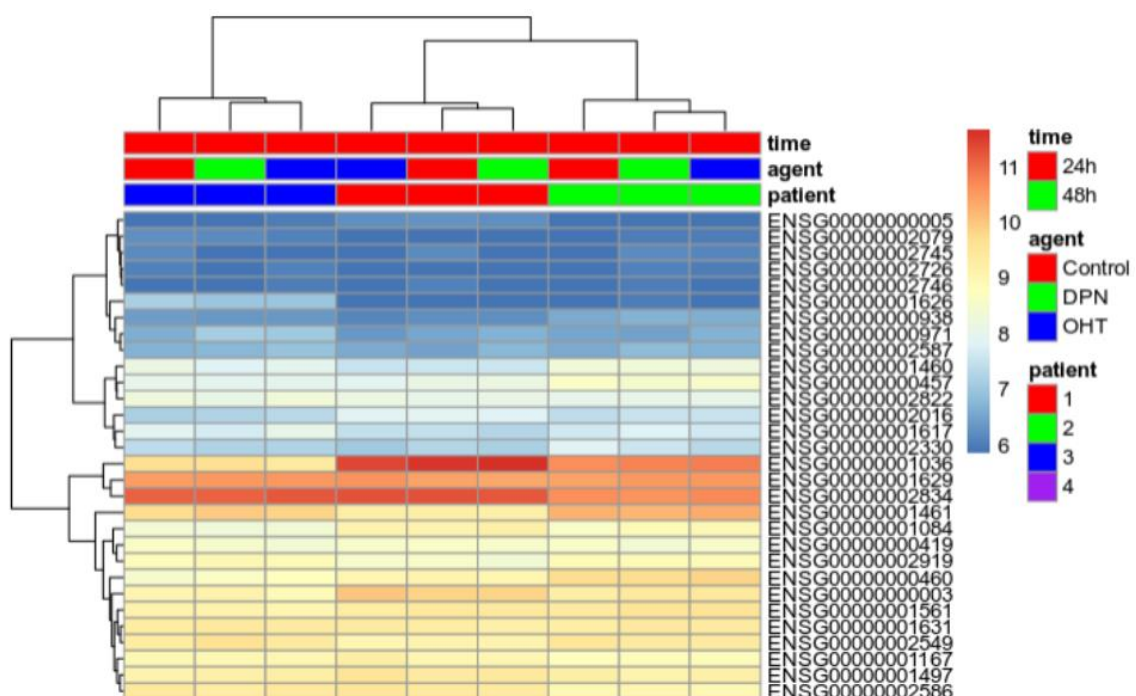
log2 fold change (MLE): agent DPN vs Control
Wald test p-value: agent DPN vs Control
DataFrame with 24416 rows and 7 columns

```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<character>
ENSG000000000003	599.90666	0.1182829	0.107442	0	1	1	TSPAN6
ENSG000000000005	1.46579	-0.6614618	1.920338	0	1	1	TNMD
ENSG0000000000419	264.84588	-0.1760605	0.137472	0	1	1	DPM1
ENSG0000000000457	195.95228	-0.1047530	0.154333	0	1	1	SCYL3
ENSG0000000000460	486.02888	0.0431445	0.114029	0	1	1	C1orf112
...
ENSG000000257097	2.421169	0.1718994	1.438615	0	1	1	NA
ENSG000000257104	0.334348	-0.5863602	3.497232	0	1	1	NA
ENSG000000257106	0.903789	-0.0232297	2.144175	0	1	1	NA
ENSG000000257108	27.409934	0.2859502	0.389701	0	1	1	NHLRC4
ENSG000000257112	8.027149	-0.3419144	0.722468	0	1	1	NA

Como en el apartado anterior, para la comparativa entre tratamiento con DPN vs Control, **NO** existen a priori genes diferencialmente expresados. Sin embargo, es recomendable discutir con el investigador para saber cómo de restrictivos se deben ser en el análisis y qué es lo que se espera encontrar en términos de genes diferencialmente expresados. De esta manera, se pueden tomar decisiones informadas sobre los valores de lfcThreshold y alpha que se deben utilizar, y se pueden interpretar adecuadamente los resultados del análisis en el contexto del diseño experimental y la literatura existente.

Finalmente, se generó un mapa de calor (heatmap) para visualizar la expresión génica de los 30 genes más significativamente diferencialmente expresados en la comparación entre el grupo de tratamiento (DPN) y el grupo de control.



CONCLUSIONES

El análisis exploratorio muestra que un mismo paciente tiene valores de expresión similares independientemente del tiempo y tratamiento, y que existen diferencias entre individuos debido a la variabilidad individual biológica de cada paciente.

La muestra 4 control 24 horas es un **outlier**, ya que se separa demasiado en el análisis exploratorio de las muestras del mismo paciente (cuando todas las muestras de un mismo paciente se agrupan juntas), por lo que se decidió eliminar dicha muestra del análisis (ya que puede influir en el resultado final).

NO existen genes diferencialmente expresados entre el tratamiento vs Control, para los dos tipos de tratamientos. Esto puede deberse a los parámetros restrictivos y conservados que hemos puesto y al bajo número de muestras analizadas durante el análisis. Se recomienda aumentar el número de muestras para obtener un resultado más fiable.

Apartado 2. PREGUNTA 5. GSEA

GSEA (**Gene Set Enrichment Analysis**) es una técnica bioinformática ampliamente utilizada para identificar patrones de expresión génica comunes en conjuntos de genes predefinidos. GSEA utiliza una lista de genes clasificada por su puntuación de expresión diferencial (por ejemplo, logaritmo del cambio de plegamiento o p-valor) y busca conjuntos de genes que muestren cambios coordinados en la expresión entre las condiciones de estudio [11].

Hay dos tipos clásicos de GSEA:

- GSEA de clasificación de genes: En este tipo de análisis, los genes se clasifican en función de su nivel de expresión diferencial entre dos grupos de muestra. Luego, los genes se ordenan en función de su puntuación de clasificación (por ejemplo, el logaritmo del cambio de plegamiento). A continuación, se realiza un análisis de enriquecimiento de conjunto de genes utilizando conjuntos de genes predefinidos, como los conjuntos de genes de la base de datos de Ontología de Términos Biológicos (GO) o los conjuntos de genes de vías metabólicas de la base de datos de la vía metabólica de Kyoto (KEGG).
- GSEA preranked: En este tipo de análisis, los genes se ordenan según su puntuación de expresión diferencial, independientemente de su clasificación entre los grupos de muestra. Luego, se realiza un análisis de enriquecimiento de conjunto de genes utilizando conjuntos de genes predefinidos, como se describe anteriormente.

En ambos tipos de GSEA, se utiliza un análisis estadístico para evaluar la significancia del enriquecimiento de conjunto de genes y se genera un gráfico de enriquecimiento de conjunto de genes que muestra la correlación entre la puntuación de expresión diferencial y la posición del gen en el conjunto de genes predefinido. El resultado del análisis de GSEA puede proporcionar información valiosa sobre los procesos biológicos que están regulados en la muestra y puede ayudar a identificar las vías y los mecanismos biológicos más relevantes para la condición estudiada.

En este caso vamos a utilizar el método **PRERANKED** ya que le vamos a meter una lista de genes que nos interesa, que es la lista de genes "**DPN_response.gmt**", correspondiente al GMT (input) con los genes más expresados en las muestras tratadas tras 48h (DPN_perturbed) y los genes más expresados en la muestra control (DPN_unperturbed)

En este caso hemos hecho un ranqueo de los genes por el LFC (log-fold-change), ya que tiene diferentes ventajas y es una medida comúnmente utilizada en análisis de expresión génica diferencial para cuantificar el cambio en la expresión génica. Las ventajas que presenta usar LFC para hacer un ranqueo son:

1. LFC tiene en cuenta tanto la magnitud como la dirección del cambio de expresión génica. Esto es importante porque algunos genes pueden estar regulados en la misma dirección en la condición de estudio, mientras que otros pueden estar regulados en direcciones opuestas.
2. LFC tiene una escala logarítmica que permite la comparación de cambios de expresión génica en diferentes rangos de expresión. La escala logarítmica también proporciona una mayor sensibilidad para detectar pequeños cambios de expresión génica.
3. LFC es una medida fácilmente interpretable que permite una comparación directa entre diferentes genes

Tenemos el archivo `res_DPN_vs_Control_24.csv` ordenado por LFC. Sin embargo, se recomienda usar un método de reducción de los datos conocido como `shrunken`, el cual se basa en técnicas estadísticas para mejorar la exactitud de las estimaciones de los parámetros al reducir el error de la varianza, pero al utilizar este método estadístico tenía errores de dependencias al utilizar el método `"apeglm"`, por lo que me quede con los datos sin ajustar.

Se guardo el archivo `.rnk` para utilizarlo en GSEA, este archivo estaba ordenado por el LFC.

```

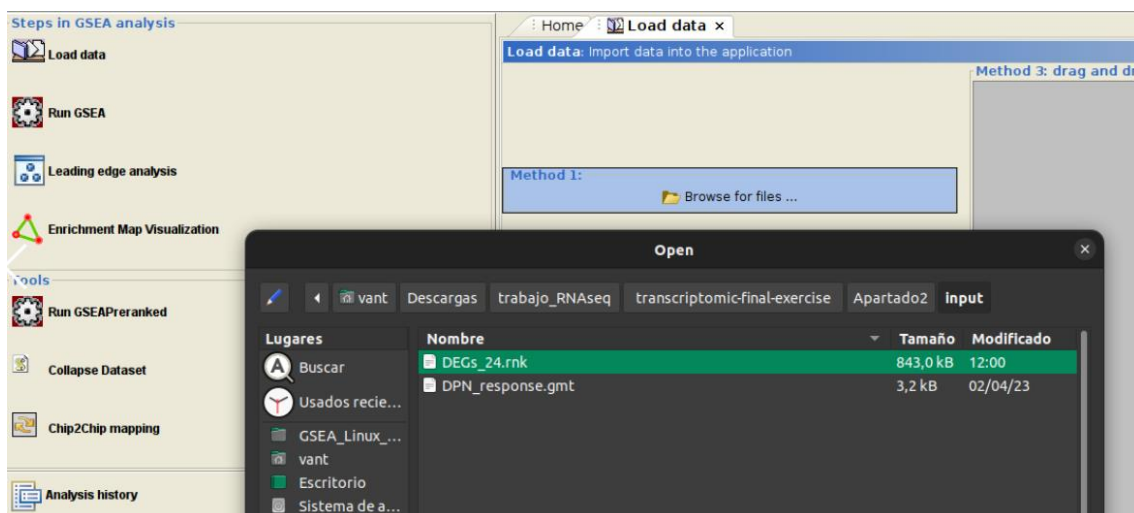
{r}
rnk_24 <- data.frame(Feature = rownames(res_DPN_order), LFC = res_DPN_order$log2FoldChange)
head(rnk_24)
write.table(rnk_24, file = "DEGs_24.rnk", sep = "\t", quot = FALSE, col.names = FALSE, row.names = FALSE)

```

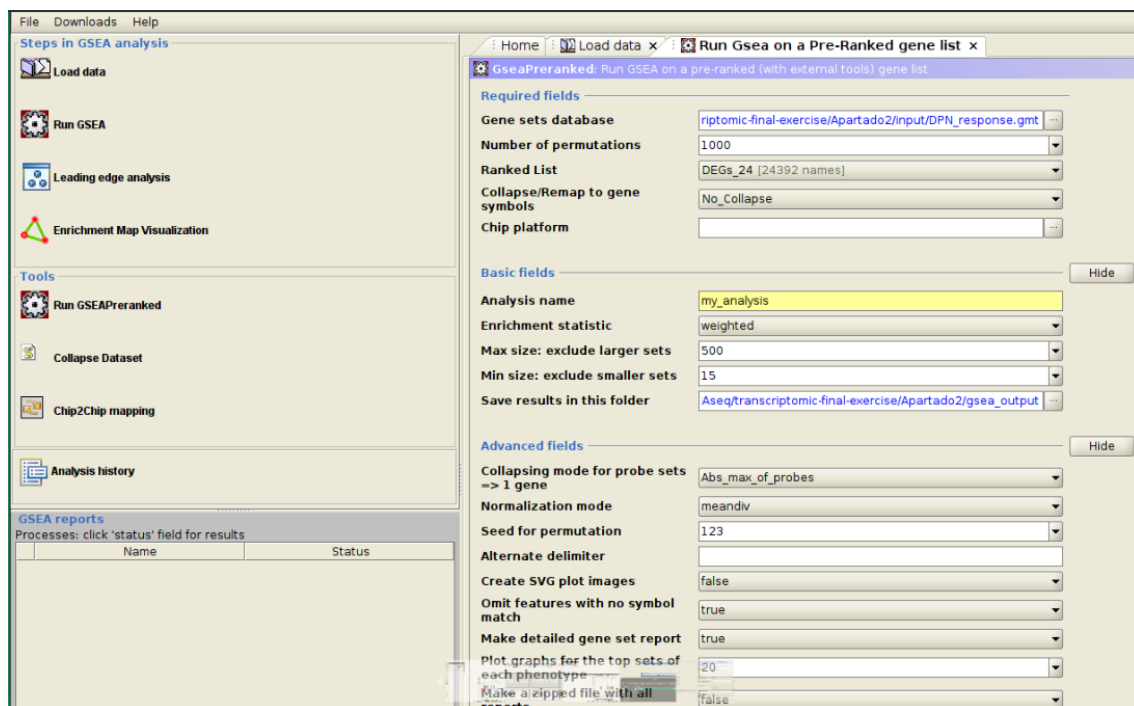
Description: df [6 × 2]

	Feature <chr>	LFC <dbl>
1	ENSG000000256312	-4.252539
2	ENSG000000249839	-4.107633
3	ENSG000000215325	-3.820298
4	ENSG000000248185	-3.522051
5	ENSG000000234393	-3.416684
6	ENSG000000120664	-3.395160

Finalmente, se abrió el software GSEA y se seleccionaron los dos archivos;



Y se seleccionaron el gene sets database (gmt), el ranked list, se puso NO COLLAPSE ya que no usamos el CHIP. Y se selecciono el output donde queremos que se guarden nuestros archivos.



Gene set DPN PERTURBED

GSEA ha identificado **un enriquecimiento** en el conjunto de genes analizado. Los resultados resumidos que se muestran a continuación indican que el valor de p nominal es de 0 (lo que equivale a 1/1000), y su FDR (valor ajustado para múltiples pruebas) también es de 0.

El Enrichment Score (ES) es una medida que indica el grado de enriquecimiento de un conjunto de genes en la condición experimental o patológica en comparación con la condición de control. El valor del ES oscila entre -1 y 1, donde un valor cercano a 1 indica un alto enriquecimiento del conjunto de genes en la condición de interés. En este caso, un ES de 0.57 indica que el conjunto de genes analizado tiene un grado moderado de enriquecimiento en la condición experimental (DPN a las 24h respecto a las 48h).

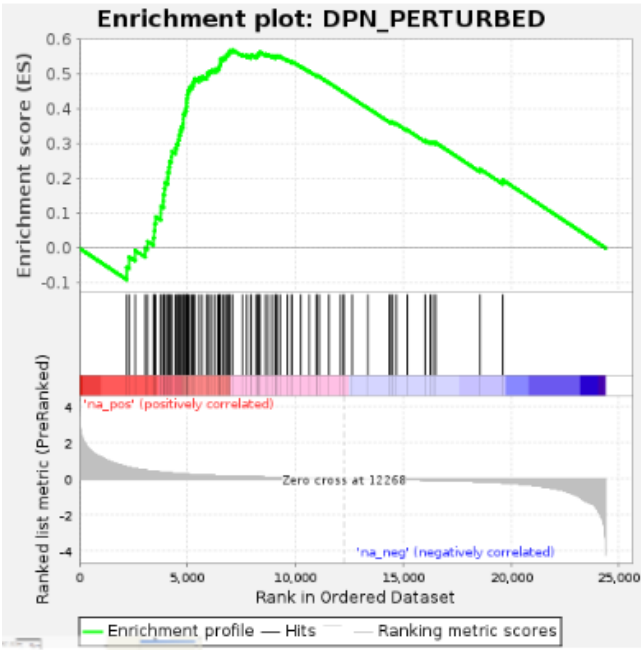
Por otro lado, el NES (Normalized Enrichment Score) es una medida del ES normalizada por la variabilidad de los ES de los conjuntos de genes en la base de datos de GSEA. Un NES de 1.72 indica que el conjunto de genes analizado tiene un enriquecimiento significativo.

	GS follow link to MSigDB	GS DETAILS	SIZE	ES	NES	NOM p-val	FDR q-val	FWER p-val	RANK AT MAX	LEADING EDGE
1	DPN_PERTURBED	Details...	100	0.57	1.72	0.000	0.000	0.000	7117	tags=59%, list=29%, signal=83%

Dataset	DEGs_24
Phenotype	NoPhenotypeAvailable
Upregulated in class	na_pos
GeneSet	DPN_PERTURBED
Enrichment Score (ES)	0.56991017
Normalized Enrichment Score (NES)	1.7207301
Nominal p-value	0.0
FDR q-value	0.0
FWER p-Value	0.0

En la sección de "Details" podemos ver que genes del gene-set forman parte del "core enrichment" (core.enrichmen == yes)

Finalmente, se observó el enrichment plot, que es una visualización que se utiliza en el análisis de enriquecimiento de conjuntos de genes para ilustrar la posición y la magnitud del enriquecimiento de un conjunto de genes en una lista ordenada de genes, en relación con una condición experimental



CONCLUSIÓN: El resultado del análisis de enriquecimiento indica que el conjunto de genes en cuestión está enriquecido en los valores altos de la tabla, es decir, aquellos que se presumen sobreexpresados en la condición experimental de exposición al agonista del receptor de estrógeno alfa (ER α), DPN en comparación con la condición control (sin tto). Este hallazgo es consistente con la información previa disponible sobre el conjunto de genes, ya que se sabe que los genes pertenecientes a este conjunto están sobreexpresados a las 48 horas y están implicados en diversos procesos biológicos relacionados con la respuesta celular a los estrógenos.

El enriquecimiento en los valores altos de la tabla sugiere que el conjunto de genes enriquecido puede estar involucrado en procesos biológicos asociados con el tratamiento con DPN y estos genes podrían estar involucrados en procesos biológicos específicos relacionados con la respuesta celular a los estrógenos, tales como la proliferación celular o la diferenciación celular.

Además, el conocimiento previo sobre la regulación de estos genes por el receptor de estrógeno alfa (ER α) y su implicación en procesos biológicos específicos a las 48h, refuerza la hipótesis de que este conjunto de genes tiene un papel importante en la respuesta celular a la exposición a DPN, y puede proporcionar pistas valiosas para futuras investigaciones sobre los mecanismos subyacentes a la acción del agonista del receptor de estrógeno alfa (ER α) en diferentes tejidos y condiciones patológicas.

Gene set DPN UNPERTURBED

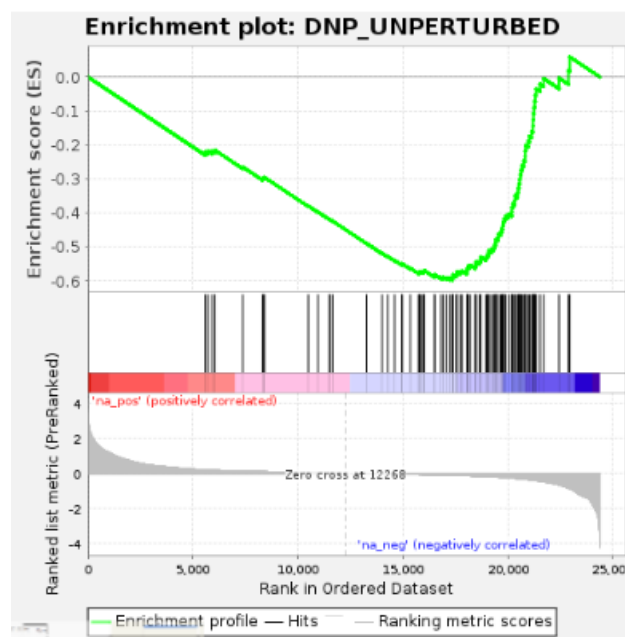
Nuestro análisis GSEA también ha detectado un enriquecimiento significativo en el segundo conjunto de genes, que consiste en los 100 genes más down-regulated en la condición experimental a las 48 horas, en relación a las 24h. El valor nominal de p-valor es 0 y su transformación para múltiple test (FDR) también, lo cual refleja su significancia estadística.

En este caso se obtuvo un NES de -1.83 y un ES de -0,60, lo cual sugiere que este dataset está enriquecido en los que son down-regulated.

	GS follow link to MSigDB	GS DETAILS	SIZE	ES	NES	NOM p-val	FDR q-val	FWER p-val	RANK AT MAX	LEADING EDGE
1	DPN_UNPERTURBED	Details...	100	-0.60	-1.83	0.000	0.000	0.000	7019	tags=68%, list=29%, signal=95%

Dataset	DEGs_24
Phenotype	NoPhenotypeAvailable
Upregulated in class	na_neg
GeneSet	DNP_UNPERTURBED
Enrichment Score (ES)	-0.5969526
Normalized Enrichment Score (NES)	-1.8348129
Nominal p-value	0.0
FDR q-value	0.0
FWER p-Value	0.0

Estos datos se pueden comprobar en el enrichment plot:



Nuestro análisis GSEA ha identificado un enriquecimiento significativo de los genes incluidos en este conjunto en la cola inferior (down-reg) de la distribución de genes y esto concuerda con el gene.set aportado. En la sección de “Details” podemos ver que genes del gene-set forman parte del “core enrichment” (core.enrichmen == yes)

Como conclusión mencionar que los genes down-regulated en 24h la mayoría de ellos también aparecen down-regulated en 48h, por lo que se observa en el GSEA el enriquecimiento de estos genes y podrían suponer una hipótesis de que estos genes NO se expresan cuando se le da el tratamiento con DPN.

Como conclusión final, el enriquecimiento en estos gene set a las 48h concuerda con nuestros resultados a las 24h. Proponiendo que a lo largo del tiempo el efecto del tratamiento con DPN tiene un efecto mayor y significativo para estos genes, y que estos genes (tanto los up como los down) pueden estar alterados en el tratamiento con DPN.

La identificación de estos genes puede proporcionar información valiosa sobre los procesos biológicos específicos que son afectados por el tratamiento, y podría ser útil para entender los mecanismos subyacentes a la respuesta celular al tratamiento. Se debería de realizar un análisis de enriquecimiento en rutas para saber si los genes up-regulated podrían promover la división celular, apoptosis, senescencia, etc. A partir de una lista de genes y conocer si están enriquecidos (PARA FUTURO).

En general, el análisis de genes up-regulated y down-regulated después del tratamiento con DPN a lo largo del tiempo puede proporcionar pistas valiosas sobre los mecanismos moleculares subyacentes a

la respuesta celular al tratamiento, lo que puede ser útil para desarrollar terapias más efectivas y personalizadas en el futuro.

REFERENCIAS

- [1] R. Stark, M. Grzelak, and J. Hadfield, 'RNA sequencing: the teenage years', *Nat. Rev. Genet.*, vol. 20, no. 11, Art. no. 11, Nov. 2019, doi: 10.1038/s41576-019-0150-2.
- [2] 'Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data'. <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> (accessed Jan. 25, 2022).
- [3] S. Chen, Y. Zhou, Y. Chen, and J. Gu, 'fastp: an ultra-fast all-in-one FASTQ preprocessor', *Bioinforma. Oxf. Engl.*, vol. 34, no. 17, pp. i884–i890, Sep. 2018, doi: 10.1093/bioinformatics/bty560.
- [4] S. W. Wingett and S. Andrews, 'FastQ Screen: A tool for multi-genome mapping and quality control', *F1000Research*, vol. 7, p. 1338, 2018, doi: 10.12688/f1000research.15931.2.
- [5] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezuk, S. McGinnis, and T. L. Madden, 'NCBI BLAST: a better web interface', *Nucleic Acids Res.*, vol. 36, no. suppl_2, pp. W5–W9, Jul. 2008, doi: 10.1093/nar/gkn201.
- [6] D. Kim, B. Langmead, and S. L. Salzberg, 'HISAT: a fast spliced aligner with low memory requirements', *Nat. Methods*, vol. 12, no. 4, Art. no. 4, Apr. 2015, doi: 10.1038/nmeth.3317.
- [7] H. Li *et al.*, 'The Sequence Alignment/Map format and SAMtools', *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009, doi: 10.1093/bioinformatics/btp352.
- [8] Y. Liao, G. K. Smyth, and W. Shi, 'featureCounts: an efficient general purpose program for assigning sequence reads to genomic features', *Bioinformatics*, vol. 30, no. 7, pp. 923–930, Apr. 2014, doi: 10.1093/bioinformatics/btt656.
- [9] P. Ewels, M. Magnusson, S. Lundin, and M. Käller, 'MultiQC: summarize analysis results for multiple tools and samples in a single report', *Bioinformatics*, vol. 32, no. 19, pp. 3047–3048, Oct. 2016, doi: 10.1093/bioinformatics/btw354.
- [10] M. I. Love, W. Huber, and S. Anders, 'Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2', *Genome Biol.*, vol. 15, no. 12, p. 550, Dec. 2014, doi: 10.1186/s13059-014-0550-8.
- [11] A. Subramanian *et al.*, 'Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles', *Proc. Natl. Acad. Sci.*, vol. 102, no. 43, pp. 15545–15550, Oct. 2005, doi: 10.1073/pnas.0506580102.

Anexo.


Resultados de stat.tsv

MUESTRA SRR479052

 SRR479052.sam.stats.tsv: Bloc de notas

```
Archivo Edición Formato Ver Ayuda
# This file was produced by samtools stats (1.6+htslib-1.6) and can be plotted using plot-bamstats
# This file contains statistics for all reads.
# The command line was: stats transcriptomic-final-exercise/Apartado1/output/sam_sample/SRR479052.sam
# CHK, Checksum [2]Read Names [3]Sequences [4]Qualities
# CHK, CRC32 of reads which passed filtering followed by addition (32bit overflow)
CHK 330c9b5a e9239de5 12706f04
# Summary Numbers. Use `grep ^SN | cut -f 2-` to extract this part.
SN raw total sequences: 25580
SN filtered sequences: 0
SN sequences: 25580
SN is sorted: 0
SN 1st fragments: 12790
SN last fragments: 12790
SN reads mapped: 20551
SN reads mapped and paired: 18488 # paired-end technology bit set + both mates mapped
SN reads unmapped: 5029
SN reads properly paired: 17960 # proper-pair bit set
SN reads paired: 25580 # paired-end technology bit set
SN reads duplicated: 0 # PCR or optical duplicate bit set
SN reads MQ0: 485 # mapped and MQ=0
SN reads QC failed: 0
SN non-primary alignments: 0
SN total length: 2344224 # ignores clipping
SN bases mapped: 1881337 # ignores clipping
SN bases mapped (cigar): 1861900 # more accurate
SN bases trimmed: 0
SN bases duplicated: 0
SN mismatches: 4981 # from NM fields
SN error rate: 2.675224e-03 # mismatches / bases mapped (cigar)
SN average length: 91
SN maximum length: 101
SN average quality: 35.6
SN insert size average: 155.4
SN insert size standard deviation: 102.1
SN inward oriented pairs: 5340
SN outward oriented pairs: 75
SN pairs with other orientation: 54
SN pairs on different chromosomes: 0
```

MUESTRA SRR479054

 SRR479054.sam.stats.tsv: Bloc de notas

```
Archivo Edición Formato Ver Ayuda
# This file was produced by samtools stats (1.6+htslib-1.6) and can be plotted using plot-bamstats
# This file contains statistics for all reads.
# The command line was: stats transcriptomic-final-exercise/Apartado1/output/sam_sample/SRR479054.sam
# CHK, Checksum [2]Read Names [3]Sequences [4]Qualities
# CHK, CRC32 of reads which passed filtering followed by addition (32bit overflow)
CHK eb74e06a 17c56d04 cada629e
# Summary Numbers. Use `grep ^SN | cut -f 2-` to extract this part.
SN raw total sequences: 16286
SN filtered sequences: 0
SN sequences: 16286
SN is sorted: 0
SN 1st fragments: 8143
SN last fragments: 8143
SN reads mapped: 12873
SN reads mapped and paired: 11514 # paired-end technology bit set + both mates mapped
SN reads unmapped: 3413
SN reads properly paired: 11114 # proper-pair bit set
SN reads paired: 16286 # paired-end technology bit set
SN reads duplicated: 0 # PCR or optical duplicate bit set
SN reads MQ0: 314 # mapped and MQ=0
SN reads QC failed: 0
SN non-primary alignments: 0
SN total length: 1507177 # ignores clipping
SN bases mapped: 1192337 # ignores clipping
SN bases mapped (cigar): 1179918 # more accurate
SN bases trimmed: 0
SN bases duplicated: 0
SN mismatches: 3191 # from NM fields
SN error rate: 2.704425e-03 # mismatches / bases mapped (cigar)
SN average length: 92
SN maximum length: 101
SN average quality: 35.5
SN insert size average: 174.4
SN insert size standard deviation: 186.7
SN inward oriented pairs: 3667
SN outward oriented pairs: 55
SN pairs with other orientation: 36
SN pairs on different chromosomes: 0
```