

Ejercicio final de transcriptómica (curso 2022-2023)

Informe del Apartado 1

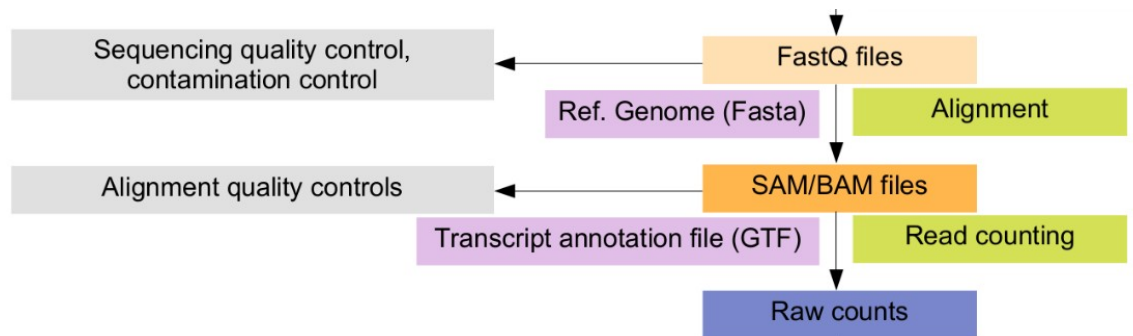
Alumno: Marcos Rubio Fernández

Sumario

Objetivo.....	3
Desarrollo.....	4
Control de calidad FastQ - FastQC.....	4
Calidad de la secuencia por base (Per base sequence quality).....	5
Contenido de la secuencia por base (Per base sequence content).....	6
Secuencias sobre-representadas (Overrepresented sequences).....	7
Contenido de adaptadores (Adapter content).....	8
Alineamiento.....	8
Indexado del genoma de referencia.....	9
Alineamiento.....	9
Control de calidad del alineamiento.....	10
Procesado con SAMTOOLS.....	11
Conteo de las lecturas.....	12
Conclusiones.....	15
Anexo: Mejoras al script.....	16
Eliminación de las secuencias adaptadoras.....	16

Objetivo

En este apartado tiene como objetivo final obtener el archivo con la matriz de cuentas de un experimento de RNAseq, partiendo para ello de una serie de secuencias en formato .fastq. Las secuencias son de tipo pair-end y corresponden con un control de adenoma de 24h (sec) y con un tratamiento de DPN?? de 24horas.



Para la consecución del objetivo marcado debemos completar una serie de apartados que se muestran en la imagen anterior:

1. Control de calidad de los archivos FastQ
2. Alineamiento
3. Control de calidad del alineamiento
4. Conteo de las lecturas

Adicionalmente a estos cuatros pasos básicos podríamos incluir algun otro, como por ejemplo:

- Pre-procesado de las secuencias: si tras el análisis de calidad de los ficheros FastQ se detecta que la calidad u otros parámetros que presentan podrían influir en la calidad del alineamiento, se puede realizar un pre-procesado. Esto puede incluir, entre otras cosas, la eliminación de secuencias adaptadoras que hayan quedado o el recorte (trimmeo) de las secuencias para eliminar baja calidad en alguno de los extremos.
- Pre-procesado de los alineamientos: en este punto lo más común suele ser el marcaje (y eliminación) de duplicados de PCR. Estos duplicados pueden sobreestimar el número de lecturas y suelen marcarse y eliminarse (colapsarse).

Desarrollo

Para este apartado nos hemos centrado inicialmente en cubrir el workflow básico de cuatro puntos que hemos visto anteriormente, utilizando para ello los programas estudiados en la fase teórica:

1. Control calidad FastQ → FastQC
2. Alineamiento → HISAT2
3. Control calidad Alineamiento → SamTools / FastQC
4. Conteo de las lecturas → HTSEQ

Para llevarlo a cabo de manera más programática, en consonancia con el objetivo de un máster en bioinformática, se ha desarrollado un pipeline que automatiza el uso de estos programas partiendo como input de las cuatro secuencias de FastQ que se nos aportan. A este pipeline se le ha denominado como Apartado1_pipeline-basico.sh.

A continuación iremos detallando los puntos principales de este pipeline, el código completo puede consultarse en el repositorio público de GitHub creado para esta práctica (<https://github.com/MarcRubFer/transcriptomic-final-exercise>).

Tal y como se especifica en README, debe activarse el ambiente “Apartado1_env” para poder llevar a cabo los procesos que se detallan a continuación. Dentro de este ambiente se pueden consultar las versiones de los programas utilizados

Control de calidad FastQ - FastQC

Para realizar el primer análisis de calidad de las secuencias en formato FastQ, utilizamos el programa FastQC mediante las siguientes líneas de código:

```
» for seqs in $(find Apartado1/input/ -name "*.fastq")
» do
»   fastqc "$seqs" -o Apartado1/output/fastqc_results/2>>Apartado1/output/fastqc_results/log.txt
» done
```

En ellas mediante este bucle for introducimos al programa FastQC cada uno de los archivos para que nos genere el resultado y lo guarde en la carpeta destino output/fastqc_results. El programa genera dos tipos de archivos:

- Archivos .html: los cuales nos conducen a una dirección web donde podemos analizar cada uno de los estudios de cada secuencia. Estos archivos se encuentran en el repositorio para su consulta.
- Archivos .zip: que contienen la información que aparece en el .html pero de manera separada (imagenes, resúmenes, etc.)

A continuación mostraremos algunos de los datos y gráficos más reseñables, centrándonos en aquellos que parecen tener problemas en las secuencias analizadas.

Apartado 1 – Informe trabajo final Transcriptómica

Estadísticas generales (Basis Statistics)

Comenzaremos con mostrar una tabla resumen de los datos que aparecen en el apartado de “Basis Statistics”, el cual recoge información importante sobre cada una de las secuencias analizadas.

Muestra	Codificación Illumina	Secuencias totales	Bases totales (en Mbp)	Longitud secuencia	%GC
SRR479052_1	1.9	15340	1.5	101	52
SRR479052_2	1.9	15340	1.5	101	52
SRR479054_1	1.9	9746	0.9843	101	51
SRR479054_2	1.9	9746	0.9843	101	51

En esta tabla tenemos los siguientes apartados:

- Muestra: indica el nombre de la muestra así como la direccionalidad de las lecturas _1 (5'-3') y _2 (3'-5').
- Codificación Illumina: codificación de la calidad de las bases dado por la casa comercial Illumina. En el caso de la codificación 1.9 los valores se representan como caracteres ASCII que van desde el número 33 al 73. Esta codificación corresponde con una codificación de Phred score de Phred33, parametro que es importante para pasos posteriores como veremos más adelante.
- Secuencias totales: número de lecturas
- Bases totales: número de bases leídas
- Longitud de la secuencia: es la longitud de la lecturas
- %GC: porcentaje de GC presente en la lecturas.

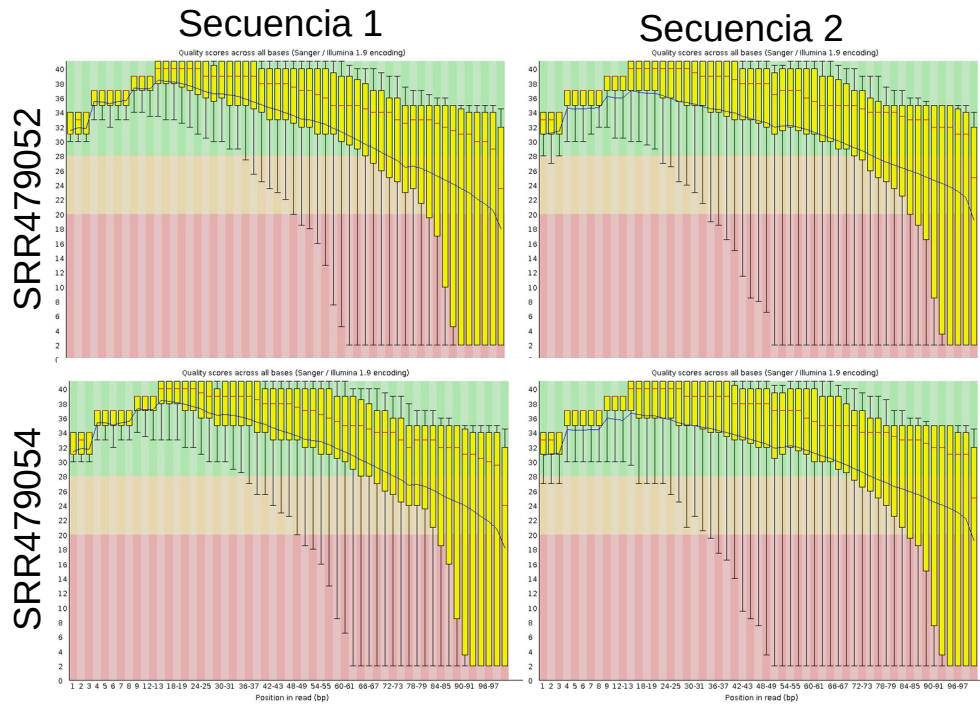
Calidad de la secuencia por base (Per base sequence quality)

En la siguiente figura mostramos los resultados obtenidos del análisis por base en cada una de las muestras y las secuencias.

Este tipo de gráficos representa la calidad media (en escala Phred 0-40) y su desviación en cada base a lo largo de todas las lecturas del archivo FastQ.

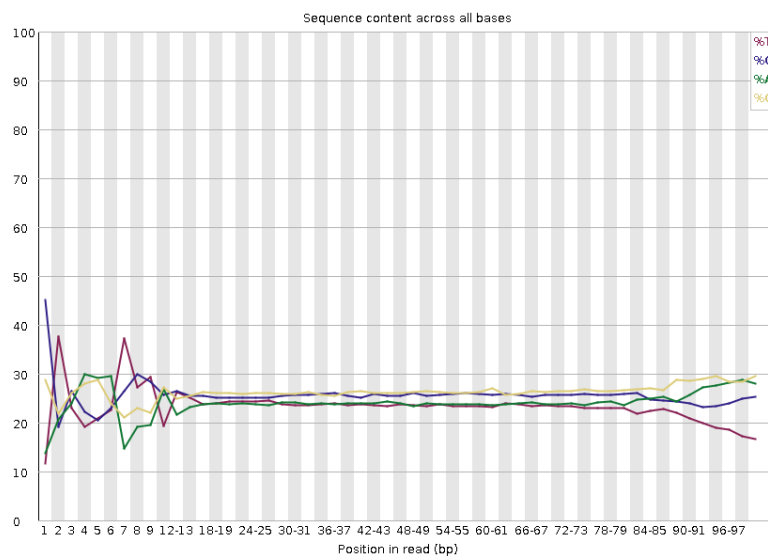
En el caso que nos ocupa podemos ver claramente que la calidad de las secuencias deja de ser considera como buena (franja verde – calidad 28) a partir de las 65 bp aprox., y que deja de ser óptima (franja amarilla – calidad 20) a partir de las 80 pb aproximadamente.

Este análisis puede ser indicativo de que para un correcto alineamiento posterior necesitaríamos recortar las secuencias para que esa mala calidad del extremo 3' no nos influya en los resultados. Como hemos indicado al inicio, este pipeline inicial con el que estamos trabajando no incluye este apartado.



Contenido de la secuencia por base (Per base sequence content)

En este caso todos los estudios FastQC dan como error para este parametro pero esto es algo característico de la técnica de RNAseq debido a la utilización de “random hexamer primers” en la generación de cDNA de las librerías. A continuación mostramos uno de los gráficos de una de las muestras, los otros tres salen similares y se puede comprobar en los archivos .html.



Apartado 1 – Informe trabajo final Transcriptómica

Secuencias sobre-representadas (Overrepresented sequences)

Esta sección nos detalla la presencia en los archivos de secuencias con una frecuencia inusualmente alta, es importante conocer este parámetro para análisis posteriores. En nuestro análisis solo se han detectado secuencias sobre-representadas en la muestra SRR479052, tanto en las secuencias 1 como 2. A continuación se muestran los resultados.

Sample	Secuencia	Conteo	Porcentaje	Posible Fuente
SRR479052_1	CTTTTACTTCCTCTAGATAGTCAAGT TCGACCGTCTTCTCAGCGCTCCGC	21	0,1368	Sin resultados
SRR479052_2	CTAACACGTGCGCGAGTCGGGGGC TCGCACGAAAGCCGCCGTGGCGCA AT	20	0,1303	Sin resultados

Si realizamos un BLASTn a estas secuencias obtenemos como resultado que son secuencias de ratón o ribosomales de otras especies para la primera muestra como se aprecia en la imagen inferior.

	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
<input checked="" type="checkbox"/>	PREDICTED: Desmodus rotundus 18S ribosomal RNA (LOC128779486). rRNA	Desmodus rotundus	93.5	93.5	100%	1e-15	100.00%	1869	XR_008425459.1
<input checked="" type="checkbox"/>	PREDICTED: Desmodus rotundus 18S ribosomal RNA (LOC128780144). rRNA	Desmodus rotundus	93.5	93.5	100%	1e-15	100.00%	1869	XR_008426095.1
<input checked="" type="checkbox"/>	PREDICTED: Desmodus rotundus 18S ribosomal RNA (LOC128780140). rRNA	Desmodus rotundus	93.5	93.5	100%	1e-15	100.00%	1869	XR_008426092.1
<input checked="" type="checkbox"/>	PREDICTED: Desmodus rotundus 18S ribosomal RNA (LOC128780136). rRNA	Desmodus rotundus	93.5	93.5	100%	1e-15	100.00%	1869	XR_008426088.1

O, para el segundo caso, corresponden con secuencias ribosomales de un orangután (Pongo Pygmaeus), como se aprecia en la siguiente imagen.

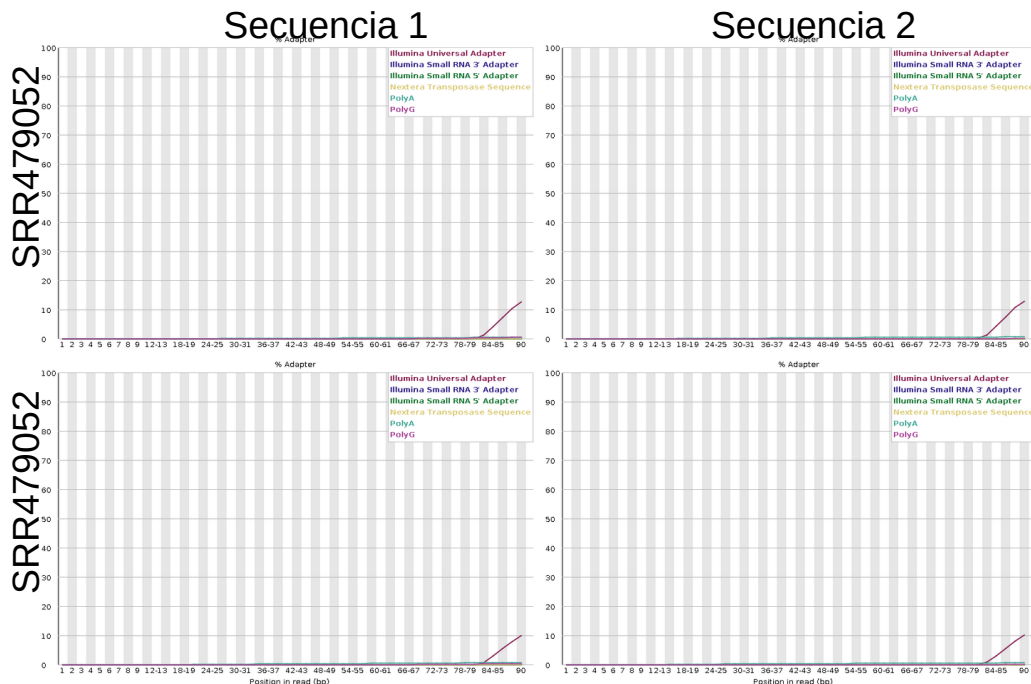
	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
<input checked="" type="checkbox"/>	PREDICTED: Pongo pygmaeus 28S ribosomal RNA (LOC129032324). rRNA	Pongo pygmaeus	93.5	93.5	100%	1e-15	100.00%	5044	XR_008501329.1
<input checked="" type="checkbox"/>	PREDICTED: Pongo pygmaeus 28S ribosomal RNA (LOC129032323). rRNA	Pongo pygmaeus	93.5	93.5	100%	1e-15	100.00%	5065	XR_008501328.1
<input checked="" type="checkbox"/>	PREDICTED: Pongo pygmaeus 28S ribosomal RNA (LOC129032321). rRNA	Pongo pygmaeus	93.5	93.5	100%	1e-15	100.00%	5067	XR_008501326.1
<input checked="" type="checkbox"/>	PREDICTED: Pongo pygmaeus 28S ribosomal RNA (LOC129032320). rRNA	Pongo pygmaeus	93.5	93.5	100%	1e-15	100.00%	5063	XR_008501325.1
<input checked="" type="checkbox"/>	PREDICTED: Pongo pygmaeus 28S ribosomal RNA (LOC129032319). rRNA	Pongo pygmaeus	93.5	93.5	100%	1e-15	100.00%	5046	XR_008501324.1
<input checked="" type="checkbox"/>	PREDICTED: Pongo pygmaeus 28S ribosomal RNA (LOC129032318). rRNA	Pongo pygmaeus	93.5	93.5	100%	1e-15	100.00%	5029	XR_008501323.1

Estos resultados nos llevan a pensar que puede existir algún tipo de contaminación por otra/s especies en la muestra control SRR479052. Para estos casos sería recomendable realizar un estudio con otro software de análisis de calidad, FastQSCREEN, que nos muestra estos resultados. De nuevo este pipeline no lo tiene incorporado.

Contenido de adaptadores (Adapter content)

Esta última sección del reporte de FastQC nos detalla si existe presencia en las secuencias de adaptadores de secuenciación conocidos. En todos los análisis se ha detectado presencia de

adaptadores universales de Illumina desde las 80 pb y llegando a suponer hasta un 10%. A continuación se muestran las gráficas de los resultados.



Alineamiento

Para el alineamiento de las secuencias FastQ vamos a utilizar el software HISAT2 utilizado durante las sesiones teóricas. Este alineador de secuencias NGS está basado en la transformación de Burrows-Wheeler (BWT) modificada para grafos. HISAT2 implementa un diseño propio de índices de grafos FM (GFM). Estos grafos están diseñados tanto de manera global para representar la población general pero también tienen índices locales más pequeños (56Kbp) que permiten cubrir colectivamente todo el genoma. Para más información se puede acudir al manual del software (<http://daehwankimlab.github.io/hisat2/manual/>).

Indexado del genoma de referencia

El primer paso del proceso de alineamiento es la realización de un indexado del genoma (si no se ha realizado ya) para así poder realizar el alineamiento de manera más rápida y efectiva. En condiciones normales, el indexado se realizaría del genoma completo pero para esta práctica se nos ha facilitado el subconjunto del cromosoma 21 del genoma humano, donde alinean nuestras secuencias. Para realizar este paso tenemos el siguiente código en nuestro pipeline:

```
» hisat2-build --seed 123 -p 2 \
»     Apartado1/input/Homo_sapiens.GRCh38.dna.chromosome.21.fa \
»     Apartado1/output/hisat2/index/Homo_sapiens.GRCh38.dna.chromosome.21 >
»     Apartado1/output/hisat2/log/hisat2_index.log
```

- El comando *hisat2-build* es el encargado de generar el indexado del genoma
- *--seed*: simplemente establece una “semilla” para la randomización que nos permite obtener los mismos resultados siempre de volver a realizarse este análisis.

Apartado 1 – Informe trabajo final Transcriptómica

- -p 2 : se refiere al número de núcleos trabajando en paralelo.
- El resto de órdenes son para indicar las rutas al input, el output y el archivo .log

Alineamiento

Antes de comenzar con el alineamiento vamos a crear un archivo de texto que contendrá los nombres de las muestras a analizar, este archivo nos servirá para el bucle que posteriormente explicaremos. El código para ello es el siguiente:

```
» find Apartado1/input/ -name '*.fastq' | xargs basename -s .fastq | cut -d'.' -f1 | uniq >
Apartado1/output/hisat2/sample_id.txt
```

Mediante los comandos find, basename, cut y uniq extraeremos de los archivos .fastq depositados en input el nombre de cada una de las muestras.

A continuación este archivo se introducirá en un bucle *for* que iterará por los nombres e irá realizando el alineamiento, en primer lugar, y posteriormente otros procesos (explicados más adelante). La parte del bucle correspondiente al alineamiento es la siguiente:

```
for sid in $(cat Apartado1/output/hisat2/sample_id.txt);
do
    fw_path=$(find Apartado1/input -name "$sid*_1*")
    rv_path=$(find Apartado1/input -name "$sid*_2*")
    mkdir -p Apartado1/output/hisat2/results/$sid
    hisat2 --new-summary --summary-file Apartado1/output/hisat2/results/$sid/$sid.hisat2.summary \
        --seed 123 --phred33 -p 2 -k 1 \
        -x Apartado1/output/hisat2/index/Homo_sapiens.GRCh38.dna.chromosome.21 \
        -1 $fw_path -2 $rv_path \
        -S Apartado1/output/hisat2/results/$sid/$sid.sam
```

Para cada una de las muestras recogidas en el archivo sample_id:

- Se determina la ruta de los archivos _1
- Se determina la ruta de los archivos _2
- Generamos un directorio específico para esa muestra dentro del de hisat2/results
- Comando *hisat2* es el que comienza el alineamiento:
- --new_summary: creará un resumen con las estadísticas de los alineamientos
- --seed 123: al igual que con el indexado establece una semilla para la aleatorización que nos permita luego reproducir los resultados
- --phred33: codificación de la calidad de las secuencias, parametro que como hemos visto podemos obtener del análisis de FastQC.
- -p 2: indica el número de cores en paralelo
- -k 1: para que el programa muestre para cada pareja de lecturas solo una alineamiento.
- -x: especifica la ruta hasta el índice creado por hisat2-build
- -1 y -2 son las rutas para los archivos .fastq pareados
- -S especifica la ruta de salida para el archivo .SAM

Un parametro/opción que podría ser importante y que se no se ha incluido en el pipeline es la **direccionalidad (--rna-strandness)**. Con este parametro se define la orientación del fragmento de RNA original y si ésta se ha tenido en cuenta o no a la hora de realizar el protocolo. Algunos kits son capaces de mantener esta direccionalidad por lo que nos permiten luego orientar, siendo esto importante en el caso de transcritos solapantes. En nuestro caso no se ha definido la orientación (unstrandness) ya que no se ha encontrado información al respecto de la misma dentro del set de SRA ni de la publicación en la que se describen los datos. Si como en este caso, no se tiene información al respecto es posible tratar de deducir esta información mediante la utilización de algunos paquetes, como por ejemplo RseqQC con su opción infer-experiment.py (<https://rseqqc.sourceforge.net/#infer-experiment-py>).

Control de calidad del alineamiento

De este proceso obtenemos unas métricas de rendimiento del alineamiento guardadas en los archivos .summary. A continuación mostramos los resultados para cada una de las muestras.

SRR479052

HISAT2 summary stats:

Total pairs: 15340

Aligned concordantly or discordantly 0 time: 6663 (43.44%)

Aligned concordantly 1 time: 7061 (46.03%)

Aligned concordantly >1 times: 0 (0.00%)

Aligned discordantly 1 time: 1616 (10.53%)

Total unpaired reads: 13326

Aligned 0 time: 6636 (49.80%)

Aligned 1 time: 6690 (50.20%)

Aligned >1 times: 0 (0.00%)

Overall alignment rate: 78.37%

Este resultado nos informa del total de parejas alineadas, 15340 (al ser parejas multiplicamos por 2 para obtener el número de lecturas, 30680) y del total de lecturas sin aparear, 13326. De las parejas alineadas, el 43,44% no se alineo correctamente, el 46,03% se alineo correctamente una única vez y el 10,53% se alineo una vez pero no correctamente. En este caso no hay multi-alineamientos. Para las lecturas desapareadas, el 49,80% no se alinearon mientras que el 50,20% se alineo correctamente una vez. De nuevo no hay multi-alineamiento.

Todos estos datos arrojan una tasa global de alineamiento del 78,37%.

SRR479054

HISAT2 summary stats:

Total pairs: 9746

Aligned concordantly or discordantly 0 time: 4043 (41.48%)

Aligned concordantly 1 time: 4852 (49.78%)

Aligned concordantly >1 times: 0 (0.00%)

Aligned discordantly 1 time: 851 (8.73%)

Total unpaired reads: 8086

Aligned 0 time: 4044 (50.01%)

Aligned 1 time: 4042 (49.99%)

Aligned >1 times: 0 (0.00%)

Overall alignment rate: 79.25%

Apartado 1 – Informe trabajo final Transcriptómica

Para la segunda muestra el total de pares es de 9746 (19492 total), de las cuales el 41,48 no han alineado, el 49,78 lo han hecho correctamente 1 vez y el 8,73% han alineado incorrectamente una vez. No hay multi-alineamientos. El total de lecturas desapareadas es de 8086, donde el 50,01% no han alineado y el 49,99% lo han hecho una sola vez; de nuevo, no hay multi-alineamientos. La tasa global de alineamiento para esta segunda muestra es del 79,25%.

Procesado con SAMTOOLS

A continuación dentro del mismo bucle for procedemos a formatear los archivos. Sam obtenidos del alineamiento a archivos comprimidos .bam. También ordenaremos e indexaremos los archivos .bam. Todo este proceso se realiza con la herramienta samtools.

Además, añadiremos al archivo de hisat2.summary, donde están las estadísticas de alineamiento realizadas por este programa, el análisis del alineamiento que realiza Samtools. El código para todo esto es el siguiente:

- samtools view -bS: realiza la conversión de formato .sam a formato .bam
- samtools sort: ordena los alineamientos en base a la posición (comenzando por los más a la izquierda). La opción -o determina la ruta para el archivo de salida.
- Samtools index: indexa las coordenadas ordenadas del alineamiento para un acceso más rápido.
- Samtools flagstat: analiza el archivo de entrada de alineamiento para ofrecer unas estadísticas. Se basa en los indicadores FLAG del alineamiento, para más información se puede consultar el manual de samtools aquí (<https://samtools.github.io/hts-specs/SAMv1.pdf>).

```
for sid in $(cat Apartado1/output/hisat2/sample_id.txt);
do
    (...HISAT2...)
    echo -e "\nSamtools processing: SAM -> BAM / Sorting / Indexing / Statistics"
    #1. conversion .sam to .bam
    samtools view -bS Apartado1/output/hisat2/results/$sid/$sid.sam >
Apartado1/output/hisat2/results/$sid/$sid.bam
    #2. sorting
    samtools sort Apartado1/output/hisat2/results/$sid/$sid.bam -o
Apartado1/output/hisat2/results/$sid/$sid.sorted.bam
    #3. indexing
    samtools index Apartado1/output/hisat2/results/$sid/$sid.sorted.bam

    echo -e "\n"

    #SAMTOOLS alignment statistics
    echo -e "\nSAMTOOLS alignment statistics\n" >>
Apartado1/output/hisat2/results/$sid/$sid.hisat2.summary
    samtools flagstat Apartado1/output/hisat2/results/$sid/$sid.sorted.bam >>
Apartado1/output/hisat2/results/$sid/$sid.hisat2.summary
```

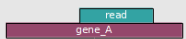
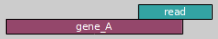


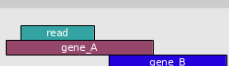

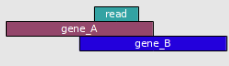
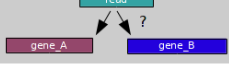
Conteo de las lecturas

Para finalizar el bucle *for* llegamos al conteo de lecturas para el cual utilizamos el programa HTSEQ visto en clase. El código para esta análisis es el siguiente:

```
for sid in $(cat Apartado1/output/hisat2/sample_id.txt);
do
    (...HISAT2...)
    (...SAMTOOLS...)
    #HTseq count
    htseq-count \
        --format=bam \
        --mode=intersection-nonempty \
        --minqual=10 \
        --type=exon \
        --idattr=gene_id \
        --additional-attr=gene_name \
        Apartado1/output/hisat2/results/$sid/$sid.sorted.bam \
        Apartado1/input/Homo_sapiens.GRCh38.109.chr21.gtf \
    > Apartado1/output/htseq/results/"$sid".gene_counts.txt
    2> Apartado1/output/htseq/log/htseq_report.log
```

htseq-count: llamamos a la parte del programa encargada de realizar el conteo.

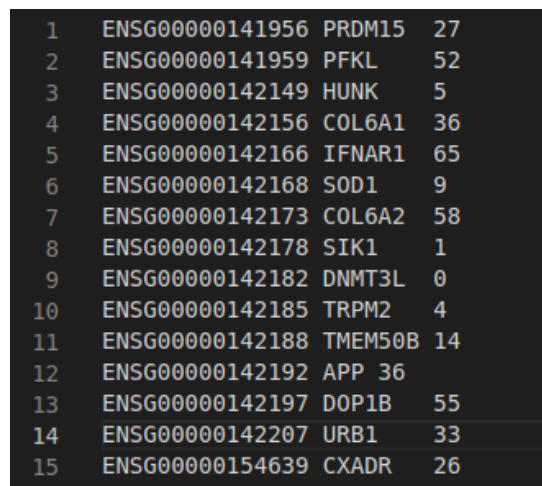
- `--format = bam`: especificamos que el archivo de entrada es formato .bam
- `--mode = intersection-nonempty`: hace relación a como realiza el conteo de la lectura en función de su solapamiento/intersección con el gen o genes. La imagen siguiente resume los diferentes modos de htseq (fuente: manual htseq (https://htseq.readthedocs.io/en/release_0.9.1/count.html#count)).

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

Apartado 1 – Informe trabajo final Transcriptómica

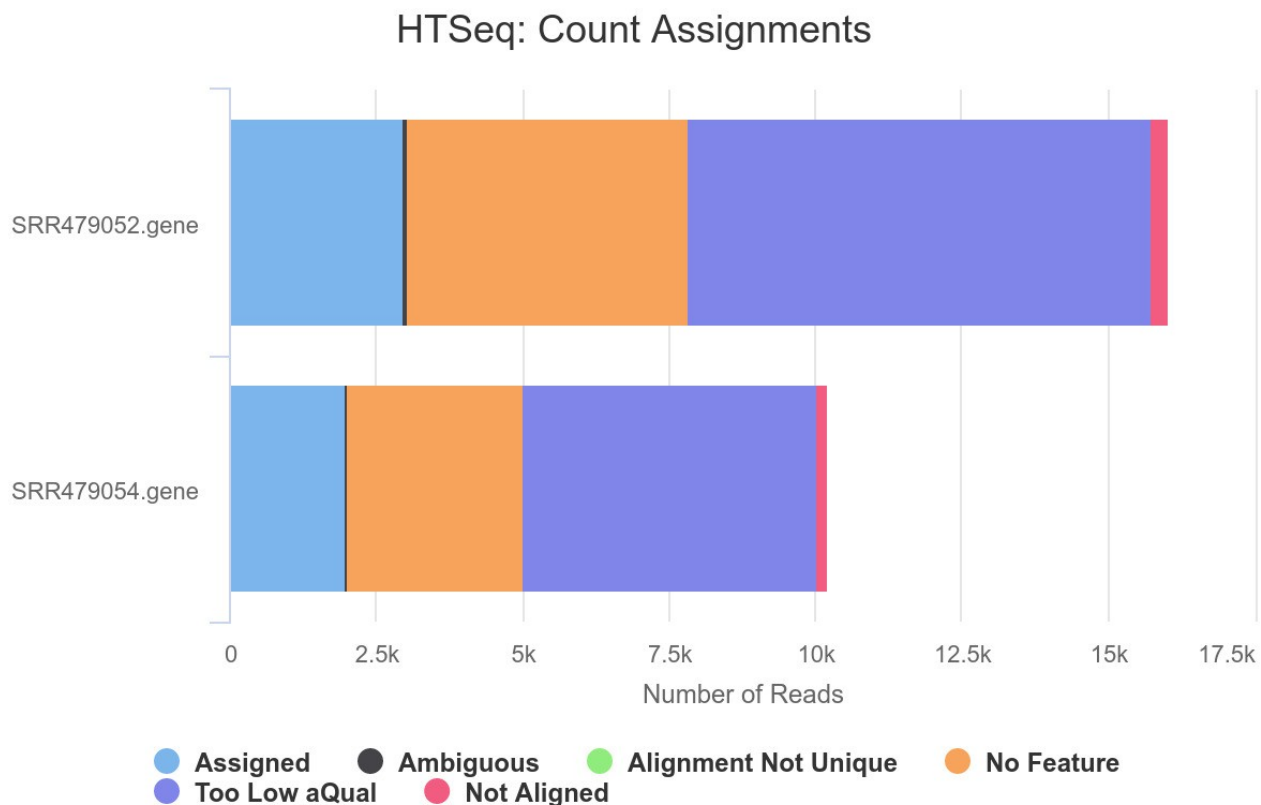
- `--minaaqual = 10`: es el mínimo de calidad del alineamiento, establecido en el archivo `.bam` a partir del cual se consideran las lecturas. Por debajo de él no las considera.
- `--type = exon`: hace referencia a la anotación (feature) que toma como referencia para hacer la asignación/conteo de las lecturas. Esta información está contenida en archivo GTF/GFF que le proporcionamos.
- `--idattr = gene_id`: establece el atributo que actuará como identificador único para cada característica o feature.
- `--additional_attr = gene_name`: son otros atributos que se añadirán al archivo de cuentas.
- La ruta del input (`.bam`)
- La ruta hasta el archivo GTF que contiene las coordenadas y las anotaciones de atributos de cada una.
- La ruta de salida a un archivo `.gene_counts.txt`
- Añadimos una redirección del `stderr` a un archivo de tipo `.log`

De este proceso obtenemos unas matrices de cuentas que tienen el siguiente aspecto



1	ENSG00000141956	PRDM15	27
2	ENSG00000141959	PFKL	52
3	ENSG00000142149	HUNK	5
4	ENSG00000142156	COL6A1	36
5	ENSG00000142166	IFNAR1	65
6	ENSG00000142168	SOD1	9
7	ENSG00000142173	COL6A2	58
8	ENSG00000142178	SIK1	1
9	ENSG00000142182	DNMT3L	0
10	ENSG00000142185	TRPM2	4
11	ENSG00000142188	TMEM50B	14
12	ENSG00000142192	APP	36
13	ENSG00000142197	DOP1B	55
14	ENSG00000142207	URB1	33
15	ENSG00000154639	CXADR	26

En él vemos el identificador del gen en formato Ensembl, el nombre del gen y el número de cuentas que se han asignado. Con la intención de obtener un resultado gráfico más entendible a primera vista se ha realizado un informe con el programa MultiQC (explicado en la siguiente sección) que nos muestra lo siguiente.



En este gráfico podemos observar que la mayoría de las cuentas no han sido asignadas por un bajo score de la calidad (morado) o bien no han podido ser asignadas a ninguna anotación/feature (naranja). El porcentaje en azul son las lecturas que han sido contadas y asignadas a una anotación. En último término, de manera minoritaria se han encontrado lecturas ambiguas a la hora de anotar el conteo o bien no han sido alineadas.

Informe MultiQC

MultiQC es una herramienta en Python que nos permite obtener en un único informe todas las estadísticas que hemos estado viendo anteriormente (FastQC, alineamiento, conteo, etc.), por lo que considero que es muy útil a la hora de evaluar de un solo vistazo toda la información que se obtiene de este pipeline. Para la obtención de este informe se incluye el siguiente código:

```
# MultiQC report

mkdir -p Apartado1/output/multiqc_report

multiqc -o Apartado1/output/multiqc_report/ Apartado1/output
```

Conclusiones

1. Partiendo de unos archivos con las secuencias fastq hemos conseguido alinear un porcentaje adecuado de secuencias, cercano al 80% global, aunque este no ha podido ser asignado a muchas anotaciones, según HTseq debido probablemente a una baja calidad de las lecturas.
2. El control de calidad de FastQC nos indica que sería recomendable realizar:
 1. Eliminación de los adaptadores detectados
 2. Eliminación de secuencias repetidas
 3. Recorte de los extremos debido a su calidad inferior a 20 (escala Phred)
3. Aun con la calidad de las secuencias mencionada el alineamiento ha conseguido alinear de manera única casi el 50% de las secuencias.
4. Las matrices de cuentas generadas para cada una de las muestras tienen pocas cuentas asignadas a un gen, hecho probablemente derivado de una calidad subóptima para el proceso.

Anexo: Mejoras al script

Tras concluir con el pipeline básico vamos a proponer algunas mejoras al mismo e iremos comprobando si nuestros resultados varían al realizar estos cambios o no lo hacen.

Eliminación de las secuencias adaptadoras

La presencia de adaptadores puede causar problemas en el alineamiento de las lecturas de secuenciación con el genoma de referencia, lo que puede resultar en una menor cobertura y precisión de la alineación. Además, los adaptadores pueden introducir sesgos de secuenciación y errores de base, lo que puede afectar la cuantificación de la expresión génica.

Para eliminar esas secuencias que hemos detectado por FastQC en el primer pipeline va a crear uno nuevo basado en el anterior e introduciremos la eliminación de las secuencias adaptadoras. Para esto vamos a utilizar la herramienta BBTools, en concreto la funcionalidad BBDuk.

Dentro del bucle for anteriormente explicado incluiremos esta funcionalidad de la siguiente manera (para verlo en contexto consultar el código en el repositorio GitHub):

```
# BBDuk adapter trimming
bbduk.sh in1=$fw_path in2=$rv_path \
  out1=Apartado1/output_BBDuk/BBDuk_results/"$sid"_1.trimmed.fastq \
  out2=Apartado1/output_BBDuk/BBDuk_results/"$sid"_2.trimmed.fastq \
  ref=Apartado1/input/BBDuk_adapters/adapters.fa \
  ktrim=r k=23 mink=11 hdist=1 tpe tbo \
```

- los parámetros *in1* e *in2* son las rutas de entrada de los Fastq
- Los parámetros *out* especifican donde se localizaran los archivos de salida pareados
- *ref* = es la ruta donde se encuentra el archivo fasta con la secuencia de los adaptadores que utilizará para la eliminación
- *ktrim*= *r* : especifica en cual de los extremos de la secuencia recortamos, en este caso “r” hace referencia al final de la secuenciación
- *k* = 23: es el tamaño de los kmers que se utiliza para la búsqueda y eliminación de las secuencias adaptadoras
- *mink*=11 : es la longitud mínima de k-mers que se permite utilizar, eso se pone ya que puede suceder que no coincida el final con los 23 estipulados anteriormente y quedaría entonces fracciones sin analizar.
- *Hdist* = 1: distancia mínima entre los k-mers y las secuencias adaptadoras, en este caso siendo 1 solo se permite una discordancia
- *tpe* : especifica que se procesen las lecturas apareadas hasta la misma longitud, esto es para casos donde solo se ha recortado una de las 2 secuencias apareadas.
- *Tbo* : especifica que también se deben recortar los adaptadores basándose en la detección de solapamiento de pares utilizando BBMerge (lo que no requiere secuencias de adaptador conocidas)

Apartado 1 – Informe trabajo final Transcriptómica

Tras este recorte miramos las estadísticas de alineamiento que nos da HISAT2:

	SRR479052	SRR479054
Total pairs	15324	9736
Aligned concordantly or discordantly 0 time:	5539 (36.15%)	3514 (36.09%)
Aligned concordantly 1 time:	9715 (63.40%)	6172 (63.39%)
Aligned concordantly >1 times:	0 (0.00%)	0 (0.00%)
Aligned discordantly 1 time:	70 (0.46%)	50 (0.51%)
Total unpaired reads:	11078	7028
Aligned 0 time:	5673 (51.21%)	3603 (51.27%)
Aligned 1 time:	5405 (48.79%)	3425 (48.73%)
Aligned >1 times:	0 (0.00%)	0 (0.00%)
Overall alignment rate:	81.49%	81.50%

Si comparamos con las tablas de HiSAT2 obtenidas anteriormente podemos ver que los porcentajes globales han mejorado, de ser menores del 80% a estar por encima del 80% (no por mucho en ambos casos). Esto sucede ya que tras trimmear las secuencias hemos aumentado el número de secuencias que se alinean correctamente y también se han reducido las lecturas no apareadas.

Alineamiento con pseudoalineador Kallisto

Se ha intentado realizar el alineamiento complementariamente con el pseudoalineador Kallisto en vez de con HISAT2.

No se ha conseguido obtener un resultado, creo que el problema ha sido que no consigo que me reconozca el archivo .bam (del parametro `-pseudobam`) para introducirlo en HTSEQ. He intentado buscar información al respecto pero no encuentro nada concreto y he tenido que desistir de debuggear este problema para continuar con el resto de la práctica.

Lamento no poder ofrecer estos resultados ya que creo que sería interesante comparar el funcionamiento entre alineadores tradicionales y la nueva generación de pseudoalineadores. El código correspondiente está en el repo y de ser posible agradecería al evaluador, si tiene tiempo, que me de un feedback de cómo se solucionaría este problema.