Análisis RNAseq

Descargar genoma de rattus_norvegicus

Necesitamos la secuencia FASTA de referencia para poder comparar nuestras secuencias y poder identificar con qué genes se están alineando.

```
#!/bin/bash

for chr in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 MT X Y

do

echo Starting with chromosome ${chr}

wget https://ftp.ensembl.org/pub/current_fasta/rattus_norvegicus/dna/Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.${chr}.fa.gz
cat Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.${chr}.fa.gz >> Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.chromosome.all.fa.g
echo Added the following bytes to general fasta:
stat -c %s Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.chromosome.all.fa.gz

rm Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.${chr}*

echo Finishing with chromosome ${chr}

done
```

Lo que hace este código es descargar (**wget**) secuencias genómicas de diferentes cromosomas del genoma de Rattus norvegicus (la rata de laboratorio común) desde un servidor FTP de Ensembl.

Después con **cat** la abre y con **>>** concatena las secuencias de los distintos cromosomas en un único archivo general.

stat -c %s lo que está haciendo es mostrar el número de bytes que se añaden al fichero al añadir la información de un nuevo cromosoma.

rm elimina los archivos dercargados para el cromosoma actual. El último echo me sirve para indicar que ya ha terminado con el cromosoma que sea.

Descargar archivo de anotación

```
#!/bin/bash

#Get .gtf file
wget https://ftp.ensembl.org/pub/release-110/gtf/rattus_norvegicus/Rattus_norvegicus.mRatBN7.2.110.gtf.gz

#Get .gff3 file
wget https://ftp.ensembl.org/pub/release-110/gff3/rattus_norvegicus/Rattus_norvegicus.mRatBN7.2.110.gff3.gz
```

Estamos descargando dos archivos distintos (formato **GTF y GFF3**; aunque realmente sólo utilizamos el formato GTF para este experimento) desde un servidor FTP de Ensembl que contienen datos de anotación genómica para Rattus norvegicus.

Los **datos de anotación genómica** son información detallada sobre la estructura, función y características de los elementos presentes en un genoma. Estos datos se refieren a la identificación y descripción de diferentes componentes genéticos y no genéticos dentro del ADN de un organismo.

Procesamiento de archivos FASTQ

Un archivo **FASTQ** es un tipo de archivo de texto que se utiliza comúnmente en bioinformática para almacenar secuencias de ADN o ARN junto con su calidad de lectura correspondiente. Se estructura en 4 líneas:

1. **Línea de identificación**, única para cada secuencia.
2. **Secuencia de ADN o ARN**.

3. **Línea de separación**, ayuda a distinguir claramente dónde termina la secuencia de nucleótidos y dónde comienzan los valores de calidad.
4. **Valores de calidad**, caracteres que representan un valor numérico que indica la confiabilidad o precisión de la lectura para esa base.

Concatenación

Algunos programas de análisis de datos genómicos pueden requerir que las lecturas forward y reverse (también conocidas como R1 y R2 en secuenciación pareada) estén en un solo archivo para realizar análisis conjunto y correcto emparejamiento de lecturas para análisis posteriores.

```
#!/bin/bash
tissue=$1
sample=$2

input_dir=${tissue}/FASTQ
output_dir=${tissue}/FASTQ_concat
whole_path=${input_dir}/${sample}

echo Starting with ${sample} forward
gunzip -c ${whole_path}/*_1.fastq.gz > ${output_dir}/${sample}_1.fastq

echo Starting with ${sample} reverse
gunzip -c ${whole_path}/*_2.fastq.gz > ${output_dir}/${sample}_2.fastq
```

Este script concatena nuestros archivos FASTQ para un tipo de **tejido** (especificado por el argumento 1) y una **muestra** específica (especificada por el argumento 2). Esto lo hago porque estoy utilizando el mismo script que Blanca, pero si lo utilizara sólo para mí podría cambiar la variable tissue por mPFC.

input_dir y **output_dir** establecen los directorios de entrada y salida basados en la variable de tejido. El input parte de los archivos FASTQ y el output guardará los archivos concatenados.

El **whole path** concatena la variable input_dir (que contiene la ruta al directorio de los archivos FASTQ) con la variable sample (que contiene el nombre específico de la muestra). El resultado es una variable llamada whole_path que contiene la ruta completa al directorio que contiene los archivos FASTQ de la muestra seleccionada.

echo me indica con qué muestra ha empezado y la dirección.

gunzip lo que hace es descomprimir los archivos FASTQ correspondientes a las lecturas 1 (*_1.fastq.gz) y 2 (*_2.fastq.gz) respectivamente, concatenando su contenido en archivos de salida.

> Utilizar **>** sobreescribe lo anterior, mientras que >> añade al final.

> El * me indica `todo lo que´, es decir, todo lo que se llame 1.fastq.gz lo va a concatenar. Por eso es muy importante estar dentro de la secuencia de la rata porque si no te va a concatenar secuencias de distintas ratas.

Los archivos resultantes se guardan en el directorio de salida (output_dir) con el nombre de la muestra seguido de _1.fastq y _2.fastq para las lecturas forward y reverse, respectivamente.

Control de calidad FASTQC

El control de calidad **identifica posibles problemas** en los datos de secuenciación, como secuencias de baja calidad, adaptadores contaminantes, secuencias sobre-representadas, desequilibrio en las bases, entre otros. Detectar estos problemas al inicio del análisis permite corregir o filtrar los datos para mejorar la precisión de los resultados.

Además, permite **ajustar los parámetros de recorte** para mejorar la calidad de regiones con problemas de calidad.

Esto ayuda a garantizar que las **conclusiones obtenidas** de los datos de secuenciación son **precisas y están libres de problemas técnicos** que podrían introducir errores en la interpretación de los resultados.

Por último, **facilita la comparación entre muestras y la replicabilidad de los resultados**. Al asegurarse de que todas las muestras tengan una calidad similar, se establece una base sólida para la comparación entre experimentos o replicaciones.

```
#!/bin/bash

#Para el CCC hay que cargar los 'modulos' o softwares que vais a usar en cada script
module load fastqc/0.11.9
tissue=$1

#El argumento -t especifica el numero de trabajos que correr a la vez (threads)
fastqc ${tissue}/FASTQ_concat/*fastq -t 20 -o ${tissue}/FASTQC
```

Primero cargamos el modulo y asignamos el primer argumento que representa el tipo de tejido.

La segunda parte ejecuta la herramienta FASTQC sobre los archivos FASTQ en el directorio ${tissue}/FASTQ_concat/. Lo que hacemos es decirle que dentro del **tejido** (mPFC en nuestro caso) y de la carpeta de FASTQ_concat busque todos los archivos que terminen con la extensión ".fastq" en ese directorio para someterlos a análisis (***fastq**).

> **-t 20** especifica el número de hilos usarán para la ejecución de FASTQC. En este caso, se establece en 20, lo que significa que se ejecutarán hasta 20 análisis simultáneamente (en este caso nos interesa que sea así porque tenemos 40 en total, 20 fw y 20 rv).

> **-o{tissue}/FASTQC** establece el directorio output para los resultados de FASTQC. En este caso, los resultados se guardarán en un directorio llamado FASTQC dentro del directorio ${tissue}. En nuestro caso: mPFC/FASTQC.

Alineamiento

El alineamiento implica asignar lecturas que provienen de transcritos de ARN al genoma de referencia para **determinar de dónde provienen dentro del genoma** y así reconstruir la secuencia de ARN original.

Esto permite **cuantificar la expresión génica** al contar la cantidad de lecturas que se alinean a genes específicos. Esto proporciona información sobre la cantidad relativa ARNm que se está produciendo en la muestra.

Además puede ayudar a **identificar variantes**, como SNPs (polimorfismos de un solo nucleótido), indels (inserciones y deleciones) y fusiones génicas, que podrían ser relevantes en términos de enfermedades o fenotipos específicos.

Asimismo, puede proporcionar información sobre la estructura de los genes y el **splicing alternativo**, identificando los sitios de unión de exones e intrones y las variantes de splicing que podrían estar presentes en las muestras.

Por último, se utiliza para **validar los ensamblajes de novo y las anotaciones** de genes, verificando si las secuencias transcripcionales generadas coinciden con el genoma de referencia conocido.

Index genoma

```
#!/bin/bash
module load hisat2/2.1.0

# unzippear archivo
echo Unzipping
gunzip Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.chromosome.all.fa.gz

# build index
echo Starting index
hisat2-build Rattus_norvegicus.mRatBN7.2.dna.primary_assembly.chromosome.all.fa g_rata
```

Primero **carga la versión** 2.1.0 de **HISAT2**, un programa ampliamente utilizado para alinear secuencias de ARN a un genoma de referencia que al acabar me muestra un mensaje del porcentaje medio de muestras que han alineado con el genoma de referencia.

Después **descomprime el archivo del genoma de referencia de la rata noruega**. Esto es necesario para construir el índice que HISAT2 utilizará para el alineamiento.

Por último, **construye el índice** que HISAT2 utilizará para alinear las lecturas de ARN. El comando hisat2-build toma el genoma de referencia descomprimido y genera un índice que facilitará el alineamiento rápido y eficiente de las lecturas. Esto es para que luego el ordenador pueda trabajar dentro de un apartado de ese índice y no tenga que ir por todo el genoma de la rata.

Alineamiento de las muestras

```
#!/bin/bash
module load hisat2/2.2.1
module load samtools/1.9

#Primer argumento: tejido con el que se trabaja
tissue=$1
#Dependiendo del valor de tissue, que se use un sufijo u otro
sufijo=""

if  [ "$tissue" == "Talamo" ]; then
        sufijo="T"
elif [ "$tissue" == "mPFC" ]; then
        sufijo="mPFC"
fi
#Index del genoma: ver carpeta ref_genome

#Alignment: como las muestras son de las mismas ratas, se hace un bucle con el nombre de cada una y lo que cambia es el tejido

for index in 1.2 2.1 2.2 2.4 2.6 2.7 3.11 3.4 3.6 3.8 3.9 4.2 5.4 5.5 5.6 5.7 6.6 6.7 7.3 7.9
do
        sample=${index}${sufijo}
        echo Starting with sample ${sample}
        hisat2 -q --rna-strandness RF -k 1 -p 4 -x ref_genome/g_rata -1 ${tissue}/FASTQ_concat/${sample}_1.fastq -2 ${tissue}/FASTQ_con

        echo Getting into samtools, sample ${sample}
        samtools view -@ 4 -b ${tissue}/alignment/${sample}_aligned.sam -o ${tissue}/alignment/${sample}_aligned.bam
        samtools sort -@ 4 -o alignment/${sample}_sorted.bam alignment/${sample}_aligned.bam
        samtools index ${tissue}/alignment/${sample}_sorted.bam
        echo Finishing with sample ${sample}
done
```

Este script en bash utiliza **HISAT2 y Samtools** para realizar el alineamiento de secuencias de ARN (RNA-seq) y las tareas relacionadas con la m**anipulación de archivos SAM/BAM**.

Lo primero que hace es **cargar los módulos** de HISAT2 y Samtools.

Lo segundo es **asignar variables**. A ésta se le añade un condicional para determinar el **sufijo** dependiendo del valor de tissue. Si tissue es igual a "Talamo", el sufijo se establece como "T". Si tissue es igual a "mPFC", el sufijo se establece como "mPFC". De nuevo, esta parte sólo es necesaria porque estoy utilizando el mismo script que Blanca.

A continuación se realiza un **bucle** sobre una serie de índices específicos (como las muestras son de las mismas ratas, se hace un bucle con el nombre de cada una y lo que cambia es el tejido). Para cada índice, se construye el nombre de la muestra (**sample**) concatenando el índice y el sufijo determinado anteriormente.

echo me indica con qué muestra está trabajando.

Se ejecuta HISAT2 para alinear las lecturas de ARN. Se especifica la información relacionada con la dirección de la secuencia RNA-strandness (**--rna-strandness RF**), se utiliza un máximo de una alineación (**-k 1**), se especifica el número de hilos (**-p 4**), y se indica el archivo de índice del genoma de referencia (**-x ref_genome/g_rata**). Los archivos FASTQ se toman de directorios específicos para cada muestra y el resultado del alineamiento se guarda como un archivo SAM.

> **-q** me indica que el input está en formato FASTAQ,

> **--rna-stradness RF** me indica que tenemos reverse y forward, siendo los forward el -1 y el reverse el -2,

> **-k** me indica el número máximo con el que quiero que mi secuencia se alinee en el genoma de referencia.

> **-p** me indica que si el ordenador tiene varios procesadores se puede correr el alineamiento en paralelo para que sea más rápido.

> **-x** es el nombre de base para el índice del genoma de referencia.

> **2>>** lo que hace es añadir al final del fichero lo que está pasando, es decir, el fichero de errores. La diferencia que tiene con respecto a >> es que este comando lo que hace es añadir el output al final del fichero, no el archivo de errores.

Samtools es una herramienta que sirve para pasar de formato sam a formato bam para que ocupe menos espacio.

Un archivo SAM (Sequence Alignment/Map) es un formato de archivo de texto plano que almacena información sobre el alineamiento de secuencias de ADN, ARN o proteínas respecto a un genoma de referencia. En un archivo SAM, cada línea representa el alineamiento de una secuencia contra el genoma de referencia. Cada línea contiene múltiples campos separados por tabulaciones, donde cada campo proporciona información específica sobre el alineamiento (identificación de la secuencia, bandera de alineación, nombre del cromosoma, posición de inicio de la alineación, secuencia alineada, calidad de alineamiento).

Un archivo BAM (Binary Alignment/Map) es una versión binaria y comprimida de un archivo SAM (Sequence Alignment/Map). Al estar en formato binario, los archivos BAM son más eficientes en términos de lectura y escritura, lo que facilita el procesamiento y la manipulación de

grandes volúmenes de datos de alineamiento. La compresión de los archivos BAM se realiza utilizando el algoritmo de compresión BGZF (Blocked GNU Zip Format), que divide el archivo en bloques y aplica compresión gzip a cada bloque, lo que permite una compresión eficiente y la posibilidad de acceder a partes específicas del archivo sin necesidad de descomprimir el archivo completo.

Contaje

El conteo en RNA-seq es un proceso fundamental que consiste en cuantificar la cantidad de lecturas de ARN que se asignan a cada gen o región genómica específica. Estos conteos representan la expresión relativa de cada gen en una muestra biológica y son la base para comparar niveles de expresión entre diferentes condiciones experimentales, tejidos o tratamientos.

Se utiliza para:

1. **Análisis de expresión génica**: Los conteos proporcionan una medida de la actividad génica, permitiendo identificar qué genes están activos y en qué medida. Estos datos son esenciales para entender la biología subyacente de un sistema biológico en estudio.

2. **Comparación entre muestras**: Al cuantificar la expresión génica en diferentes condiciones, como muestras de tejido normal y enfermedad, se pueden identificar genes diferencialmente expresados. Esto ayuda a comprender los cambios en la regulación génica asociados con diferentes estados biológicos.

3. **Validación de resultados**: Los conteos pueden utilizarse para validar otros análisis, como los estudios de variantes, para asegurar que las diferencias observadas sean consistentes con los niveles de expresión génica entre muestras.

4. **Identificación de vías biológicas**: Al analizar conjuntos de genes diferencialmente expresados, se pueden identificar vías biológicas o procesos celulares que estén regulados en diferentes condiciones experimentales.

```
#!/bin/bash
tissue=$1

input_dir=${tissue}/alignment
output_dir=${tissue}/counts

sufijo=""
if [ "$tissue" == "Talamo" ]; then
        sufijo="T"
elif [ "$tissue" == "mPFC" ]; then
        sufijo="mPFC"
fi

module load miniconda/3.7

mkdir -p ${tissue}/counts
for index in 1.2 2.1 2.2 2.4 2.6 2.7 3.11 3.4 3.6 3.8 3.9 4.2 5.4 5.5 5.6 5.7 6.6 6.7 7.3 7.9
do
        sample=${index}${sufijo}
        echo Starting with sample ${sample}
        htseq-count -f bam -r pos -m intersection-strict --stranded reverse --minaqual 1 -t gene --idattr gene_id ${input_dir}/${sample}
        echo Finished with sample ${sample}
done
```

Este script en bash está diseñado para realizar conteos de expresión génica a partir de archivos BAM utilizando la herramienta htseq-count, que es comúnmente usada en análisis de RNA-seq.

Primero se **definen las variables** de tejido, input_dir={tissue}/alignment y output_dir={tissue}/counts. El directorio de entrada almacena los archivos BAM de alineamiento, mientras que el de salida será donde se guarden los archivos de conteos.

A continuación hay un bucle para el **sufijo="..."** que determina un sufijo dependiendo del valor de tissue (Tipo de tejido). Si tissue es "Talamo", el sufijo será "T"; si es "mPFC", será "mPFC".

module load miniconda/3.7 carga Miniconda, un gestor de entornos de Python, que probablemente se necesite para ejecutar htseq-count.

mkdir -p ${tissue}/counts crea el directorio de salida para los archivos de conteo.

En el bucle, dentro de cada muestra lo que se va a hacer es:

> Crear el nombre de la muestra combinando el índice con el sufijo determinado anteriormente (**sample={index}${sufijo}**).

> **htseq-count ...** realiza el conteo de expresión génica. Especificando las siguientes características:

>> Esta herramienta utiliza archivos BAM como entrada (**-f bam**).

> **-r pos** indica cómo se deben leer las lecturas desde los archivos BAM. En este caso, pos significa que las lecturas se deben leer comenzando desde la posición de mapeo de cada read.

> **-m intersection-strict** determina cómo se cuentan las alineaciones de las lecturas con las características de anotación del genoma (proporcionadas en el archivo GTF). En este caso sólo se contarán las lecturas que se superpongan estrictamente con una característica de anotación (por ejemplo, un gen).

> **--stranded reverse** especifica la información sobre el tipo de biblioteca RNA-seq utilizada. En este caso, se indica que la biblioteca es "strand-specific", y las lecturas se alinean en la dirección opuesta al gen al que pertenecen. Esto es importante para determinar la dirección de la transcripción.

> **--minaqual 1** establece un valor mínimo de calidad para las bases en las lecturas que se utilizarán en el conteo. En este caso, se aceptarán todas las bases con una calidad mínima de 1.

> **-t gene** indica que se deben contar las lecturas a nivel de genes. Esto significa que las lecturas se asignarán a los genes basándose en la información de anotación proporcionada en el archivo GTF.

> **--idattr gene_id** especifica el atributo del archivo de anotación GTF (Rattus_norvegicus.mRatBN7.2.110.gtf.gz) que contiene información de anotación genómica para asignar las lecturas a genes específicos.

> El resultado se redirige a un **archivo TSV en el directorio de salida**. Un archivo TSV (Tab-Separated Values) es un tipo de archivo de texto plano en el que los datos están organizados en filas y columnas, donde los valores de cada fila están separados por tabulaciones en lugar de comas como en un archivo CSV.

Cuando termine aparecerá un **echo** diciendo que ha terminado de correr el script.