Analaisis RNAseq

Genome download

```
#!/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.gz
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
```

Necesitamos la secuencia FASTA de referencia para poder comparar nuestras secuencias y poder identificar con qué genes se están alineando. Lo que hace este código es descargar la secuencia de referencia (wget) de la página web. 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. 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 el genoma de referencia de la rata noruega en formato GTF y GFF3 (aunque realmente sólo utilizamos el formato GTF para este experimento)

FASTQ treatment

Concatenación

```
#!/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
```

Como estoy utilizando el mismo script que Blanca tenemos la variable de tissue, pero si lo utilizara sólo para mí podría cambiarlo por mPFC. Después tenemos la variable de muestra, el directorio de entrada donde se especifica el tejido y el del output donde meterá las secuencias una vez concatenadas. El whole path está utilizando el directorio de entrada y la muestra en concreto. echo me indica con qué muestra ha empezado y la dirección forward. Gunzip lo que hace es descomprimir los archivos. > sobreescribe lo anterior, a diferencia de >> que añade al final. * 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.

Preprocesamiento

Control de calidad FASTQC

```
#!/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
```

Una vez cargamos el modulo lo que hacemos es decirle que dentro de mPFC y de la carpeta de FASTQ_concat va a coger todo lo que acaba en fastq y va a especificar con -t 20 trabajos al mismo tiempo (en este caso nos interesa que sea así porque tenemos 40, 20 de cada tipo) el output lo va a guardar como mPFC/FASTQC

Alineamiento

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
```

Esto está generando un índice dentro de la secuencia de referencia 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 ${tissue}/FASTQ_concat/${sample}_2.fastq -S ${tissue}/alignment

        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
```

hista2 es un programa de alineamiento que al acabar me muestra un mensaje el porcentaje medio de muestras que han alineado con el genoma de referencia. -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.

Contaje

```
#!/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}_sorted.bam ./ref_genome/Rattus_norvegicus.r
        echo Finished with sample ${sample}
done
```