

Curso de Iniciación a la Secuenciación Masiva

BU-ISCI

Práctica: Control de Calidad y preprocesamiento de ficheros fastq

Descripción

Para esta parte de la práctica vamos a utilizar los datos que se encuentran en la carpeta 02_preprocessing.

Cambiamos al directorio RAW y visualizamos uno de los ficheros. Se trata de datos provenientes de una secuenciación por MiSeq de 2x251, es decir, datos paired-end de 251 nt de longitud.

```
# Comprobamos donde estamos situados.
pwd
# Output:
/home/alumno/ngs_course_exercises/02_handson_preprocessing/01_fastq_format/
prueba_454

#Comprobamos que tenemos activado el environment de conda de las prácticas
y si no lo activamos
conda activate ngs_course

# Nos movemos a la carpeta que contiene la segunda parte de la practica
# Recordatorio: - con .. accedemos a la carpeta inmediatamente superior a
la que nos encontramos.
# - con ../.. subiríamos dos niveles en el árbol de
directorios.
cd
/home/alumno/ngs_course_exercises/02_handson_preprocessing/02_preprocessing

# Comprobamos que estamos donde debemos estar
pwd
# Output:
/home/alumno/ngs_course_exercises/02_handson_preprocessing/02_preprocessing

# Listamos el contenido de la carpeta
ls
# Output:RAW RESULTS RESULTS_CORRECTED

# Nos movemos a la carpeta RAW
cd RAW
pwd
ls

# Visualizamos el contenido del fichero fastq
less CFSAN002083-01_S1_L001_R1_001_fixed.fastq
# Recordatorio: para salir del comando less presionando q
```

Volvemos a la carpeta del análisis. Lo primero que vamos a hacer es utilizar el programa FastQC, que nos permite hacer un control de calidad de los ficheros tal y como se ha explicado en teoría.

```
# Volvemos al directorio de análisis
cd ..

# Comprobamos donde estamos situados
pwd
ls

#Comprobamos que la carpeta donde vamos a generar los resultados está vacía
ls RESULTS/QC/RAW/

#También podemos abrir la carpeta en el explorador de archivos y ver que
también está vacía

# Realizamos el análisis de calidad con fastqc
fastqc -t 4 RAW/CFSAN002083-01_S1_L001_R1_001_fixed.fastq RAW/CFSAN002083-
01_S1_L001_R2_001_fixed.fastq -o RESULTS/QC/RAW
```

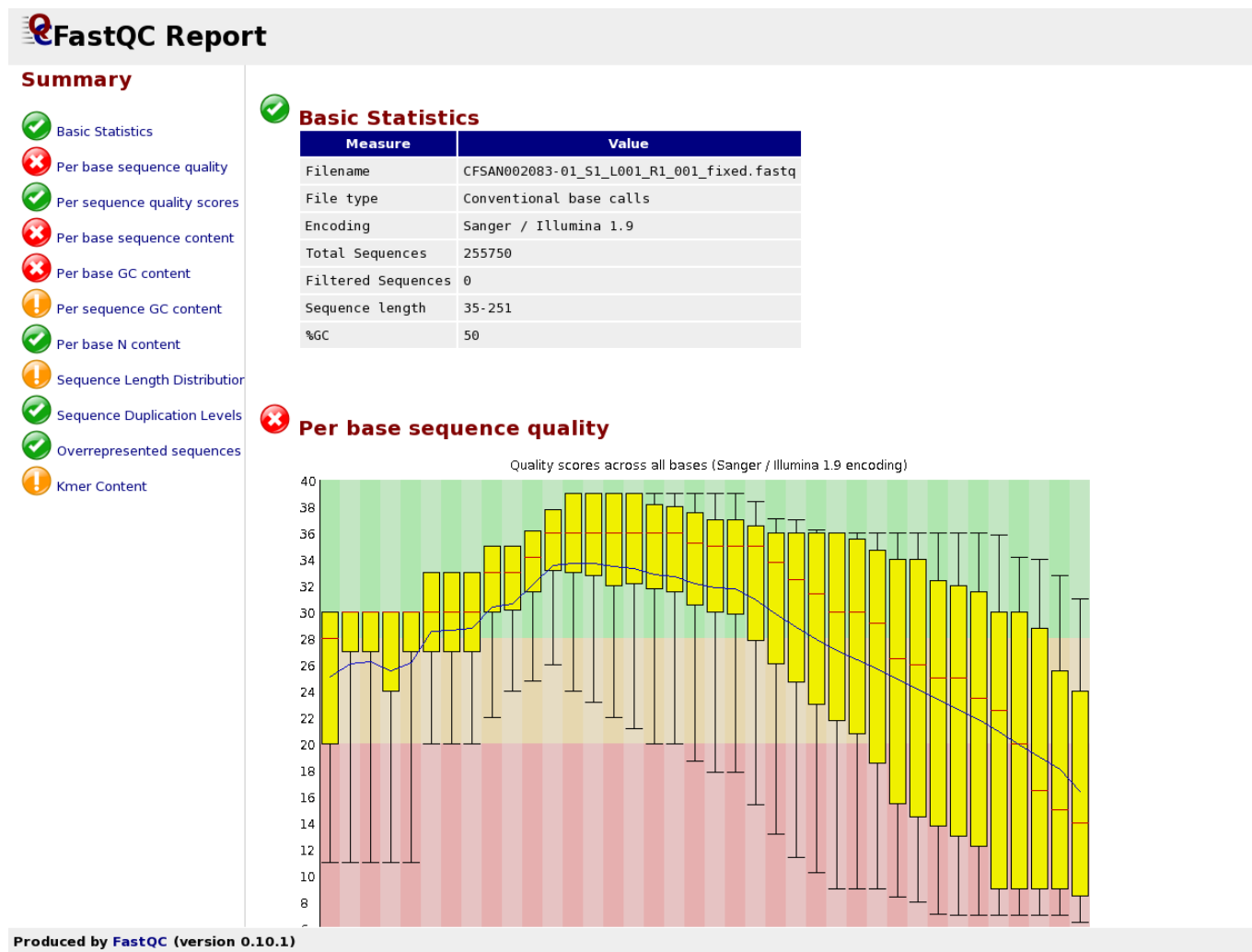
Aquí estamos ejecutando el programa y diciéndole los ficheros que tiene que analizar y con el parámetro “-o” el directorio donde queremos que se guarden los resultados. El parámetro “-t” indica cuántos núcleos del procesador puede utilizar.

En el explorador de ventanas vamos a visualizar los resultados. Recordad que se encuentran en donde habéis dicho al programa que los guarde es decir
/home/alumno/ngs_course_exercises/02_handson_preprocessing/02_preprocessing/RESULTS/QC/RAW.

```
ls RESULTS/QC/RAW/
```

Ahí hay dos ficheros que acaban en fixed_fastqc.html, uno por cada fichero fastq analizado. Hacemos doble click sobre uno de los ficheros html y se abrirá el explorador Firefox.

Estaremos visualizando ahora el resultado del fastQC con todas las gráficas que se han visto en teoría. Estas estadísticas nos permitirán tomar una decisión en cuanto al preprocesamiento de los datos, si tienen buena calidad, si necesitan filtrado, si necesitan trimming en los extremos, ...



El convenio de “buena” calidad de una base en una lectura es de 20 de calidad Phred, si vamos a hacer variant calling, p.e. incluso sería recomendable un mínimo de 30. Según este principio:

- ¿Consideráis que este experimento es de buena calidad?
- ¿De cuántas lecturas partimos?
- ¿Creéis necesario realizar filtrado?
- ¿Creéis necesario realizar trimming?

Este ejemplo ha sido seleccionado por ser especialmente de mala calidad, cuánto más largas sean las lecturas normalmente peor calidad nos encontramos y más complejo se hace el preprocesamiento para poder tener un buen balance entre no perder muchas reads y no utilizar malos datos en nuestro análisis.

En este caso evaluando la calidad nos encontramos con 255570 reads de entre 35 y 251 nt de longitud con una caída de calidad en el inicio 5' típica de la tecnología HiSeq y MiSeq de Illumina, pero también vemos una calidad media bastante baja que cae de forma brusca en el extremo 3'.

Vamos a realizar varias pruebas para ver qué aproximación de preprocesamiento es la mejor.

Filtrado por calidad

Para realizar el filtrado por calidad vamos a utilizar el software [fastp](#).

```
#Comprobamos que la carpeta donde se van a generar las lecturas está vacía
ls RESULTS/QC/FILTERED/

# Realizamos filtrado de lecturas por calidad
fastp -i RAW/CFSAN002083-01_S1_L001_R1_001_fixed.fastq \
      -I RAW/CFSAN002083-01_S1_L001_R2_001_fixed.fastq \
      -o RESULTS/QC/FILTERED/CFSAN002083-01_R1_filtered.fastq \
      -O RESULTS/QC/FILTERED/CFSAN002083-01_R2_filtered.fastq \
      -j RESULTS/QC/FILTERED/fastp.json \
      -h RESULTS/QC/FILTERED/fastp.html \
      --qualified_quality_phred 20 \
      --unqualified_percent_limit 30 \
      --length_required 50

# Realizamos el análisis de calidad
fastqc -t 2 RESULTS/QC/FILTERED/CFSAN002083-01_R1_filtered.fastq \
RESULTS/QC/FILTERED/CFSAN002083-01_R2_filtered.fastq \
-o RESULTS/QC/FILTERED

#Comprobamos que se han generado los resultados
ls RESULTS/QC/FILTERED/
```

Estos son los parámetros que estamos modificando del comando de fastp para este análisis en concreto.
¡No hay que lanzarlo!

```
#####
#      Ayuda del comando - no ejecutar      #
#####

Parámetros:
--qualified_quality_phred: calidad phred mínima necesaria.
--unqualified_percent_limit: porcentaje de las bases que está permitido que
no cumplan el límite de calidad phred.
--length_required: todas las lecturas menores a 50 bases se eliminan.
```

Abrimos en el explorador de ventanas como en el caso anterior en RESULTS/QC/FILTERED

Doble click sobre CFSAN002083-01_R1_filtered_fastqc.html y observamos los resultados.

- ¿Hemos mejorado notablemente la calidad?
- ¿Cuántas lecturas hemos perdido?

Trimming + filtrado por calidad

Ahora vamos a realizar antes del filtrado por calidad un proceso de trimming en los extremos.

```
# Realizamos trimming de las lecturas
fastp -i RAW/CFSAN002083-01_S1_L001_R1_001_fixed.fastq \
```

```

-I RAW/CFSAN002083-01_S1_L001_R2_001_fixed.fastq \
-o RESULTS/QC/TRIMMING_FILTERED/CFSAN002083-
01_R1_filtered_trimmed.fastq \
-O RESULTS/QC/TRIMMING_FILTERED/CFSAN002083-
01_R2_filtered_trimmed.fastq \
-j RESULTS/QC/TRIMMING_FILTERED/fastp.json \
-h RESULTS/QC/TRIMMING_FILTERED/fastp.html \
--qualified_quality_phred 20 \
--unqualified_percent_limit 30 \
--cut_front --cut_tail \
--trim_poly_x \
--cut_mean_quality 30 \
--cut_window_size 10 \
--length_required 50

# Realizamos análisis de calidad
fastqc -t 2 RESULTS/QC/TRIMMING_FILTERED/CFSAN002083-
01_R1_filtered_trimmed.fastq \
RESULTS/QC/TRIMMING_FILTERED/CFSAN002083-01_R2_filtered_trimmed.fastq \
-o RESULTS/QC/TRIMMING_FILTERED/

ls RESULTS/QC/TRIMMING_FILTERED/

```

Estos son los parámetros que estamos modificando del comando de fastp para este análisis en concreto.
¡No hay que lanzarlo!

```

#####
#      Ayuda del comando - no ejecutar      #
#####
--cut_front --cut_tail: cortamos por 5' y por 3'
--trim_poly_x: cortamos extremos con polyX
--cut_mean_quality: valor phred medio que tiene que tener la ventana de nt
para cortar.
--cut_window_size: número de nucleótidos de la ventana para ir evaluando
desde los extremos de la lectura.

```

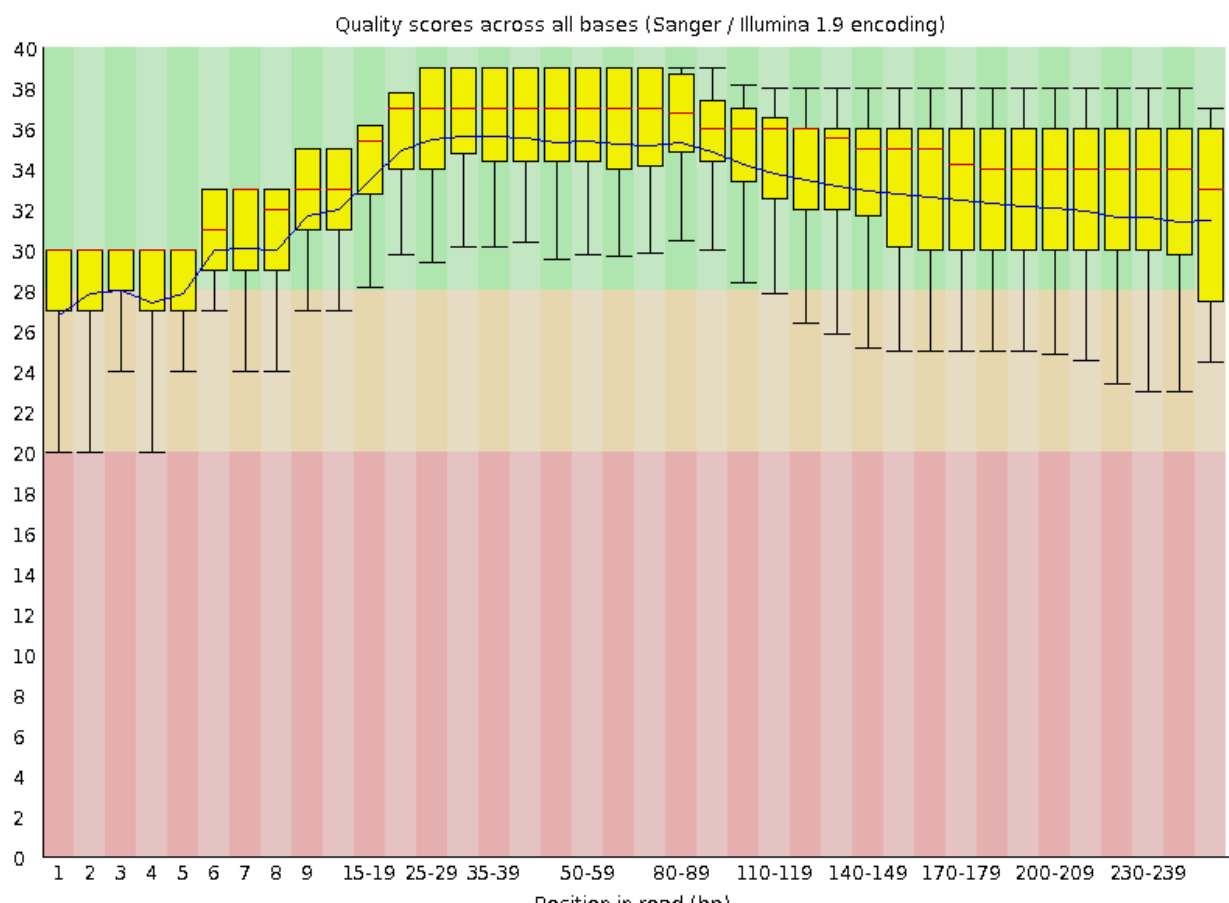
Abrimos en el explorador de ventanas como en el caso anterior en RESULTS/QC/TRIMMING_FILTERED

Doble click sobre CFSAN002083-01_R1_filtered_trimmed_fastqc.html y observamos los resultados.

- ¿Hemos mejorado notablemente la calidad?
- ¿Cuántas lecturas hemos perdido con esta aproximación?



Per base sequence quality



Resolución

Se observa con este ejemplo la importancia del análisis de la calidad y del pre-procesamiento de los datos.

En muchos casos sólo con el paso de filtrado es suficiente para quedarnos con una calidad aceptable para el posterior análisis de los datos.

Pero en otros casos como el ejemplo con el que hemos trabajado hoy es necesario realizar un trimming por los extremos previo al filtrado, ya que el principal problema de calidad de las reads es que los errores se concentran en el final. De manera que sólo con el filtrado no se consigue eliminar el problema ya que las reads malas tienen un alto porcentaje de nucleótidos de buena calidad pero unos extremos muy malos que es lo que estropean el dato.

De hecho, perdemos más reads realizando sólo el paso de filtrado, que realizando trimming y luego filtrado.