Qiime 2: De Ilumina a taxonomía.

Durante todo este manual hemos utilizado la herramienta Qiime2. Es por ello que hay que tenerla descargada, instalada y activada (nosotros la tenemos instalada con MiniConda, para activarla usamos el comando “conda activate qiime2-2022.2”)

Nota: Recomendamos trabajar en una nueva carpeta vacía, nada de trabajar desde el escritorio o el home.

El primer paso es transformar nuestros archivos .fastq.gz en un artefacto.qza con el cual qiime2 pueda trabajar. Para ello, crearemos una carpeta donde introduciremos nuestras secuencias R1 y R2 de la muestra, NO introducir los controles. En mi caso he nombrado a la carpeta como “mtp_pe”. Nosotros estamos trabajando con secuencias demultiplexed paired-end tipo Casava 1.8, es por ello que usaremos el siguiente comando: (para más información)

> qiime tools import \

--type 'SampleData[PairedEndSequencesWithQuality]' \ #Indicamos el formato (no cambiar)

--input-path mtp_pe/ \ #Carpeta con nuestros archivos .fastq.gz R1 y R2

--input-format CasavaOneEightSingleLanePerSampleDirFmt \ #Formato paired-end (no cambiar)

--output-path sample.qza #Output de artefacto.qza

Hay que tener en cuenta que los reads ya vienen demultiplexed, es por ello que nos saltaremos todos los pasos de demultiplexing del manual anterior.

Creamos un archivo.qzv que nos devuelva el resumen con la información de nuestros reads. Este archivo nos será de gran utilidad a lo largo del proceso.

> qiime demux summarize \

--i-data sample.qza \ #Artefacto creado en paso anterior

--o-visualization sample.qzv #Resumen visualizable

Para visualizar archivos.qzv usaremos el comando “qiime tools view”

> qiime tools view sample.qzv

Antes de continuar con el control de calidad hay que eliminar los primers de las secuencias, para ello usaremos la herramienta “cutadapt”.

> qiime cutadapt trim-paired \ #Indicamos que nuestro artefacto es paired-end y demultiplexed.

--i-demultiplexed-sequences prueba.qza \

--p-cores 2 \ #Indicamos el número de núcleos que queremos que utilice para este trabajo.

--p-front-f CCTACGGGNGGCWGCAG \ #Secuencia del FW primer

--p-adapter-f GGATTAGATACCCBDGTAGTC \ #Secuencia inversa del complementario al RV

--p-front-r GACTACHVGGGTATCTAATCC \ #Secuencia del RV primer

--p-adapter-r CTGCWGCCNCCCGTAGG \ #Secuencia inversa del complementario al FW

--p-discard-untrimmed \ #Indicamos que nos descarte aquellas secuencias que no presenten primer (así reducimos los errores)

--o-trimmed-sequences prueba-trimmed.qza

----------Lo del trimm no termina de funcionar----------

Como se puede observar los primers presentan letras que no corresponden a A, T, G, o C, se trata de la nomenclatura IUPAC para bases variables, ej: N significa “cualquier base”

Ahora vamos a realizar las tablas de características y el control de calidad. En el manual anterior ya se explicaron 2 maneras distintas de como realizar este punto, y la función del mismo. En este manual, solo utilizaremos el plugin Dada2.

> qiime dada2 denoise-paired \ #Indicamos que estamos trabajando con paired-end reads

--i-demultiplexed-seqs prueba-trimmed-filtered.qza \

--p-trunc-len-f 250 \

--p-trunc-len-r 250 \

--p-n-threads 0 \ #Al introducir este comando con el valor 0 permite usar el máximo número de nucleos para realizar la tarea, reduciendo el tiempo de espera

--o-representative-sequences rep-seqsP5.qza \

--o-table tableP5.qza \

--o-denoising-stats statsP5.qza \

--verbose #Nos muestra como avanza el programa

Nota: al trabajar con archivos tan grandes es normal que este comando tarde en cargar.

Este es el punto crítico del proceso. Dada2 es una tubería que realiza 4 funciones distintas: filtering (por calidad y longitud (triiming)), denoising (elimina el ruido), merged (formación de los contigs) y chimeric filtering (elimina las quimeras formadas durante el merged).

- Filtrado: Para el filtrado primero se realiza el trimming de las bases indicadas, posteriormente se hace un estudio estadístico de los errores en las bases usando 100.000 de las secuencias en la muestra. Para obtener el filtrado de mejor calidad posible es necesario que las bases no tengan un quality score por debajo de 30, para ello realizamos

el trimming. Lo ideal es realiazar el trimming a partir del punto donde el 25 percentil caiga por debajo de 30. Si no queremos perder tantas bases podemos realizar el trimming en el punto donde la mediana caiga por debajo de 30.

-Denoising: elimina el ruido de fondo.

-Merged: En este punto se forman los contigs que serviran para crear las tablas de características y de frecuencias. Para que se forme un contigs tiene que haber como mínimo un solapamiento de 12pb idénticas, este es el número mínimo por lo que bajo ninguna circunstancia recomendamos reducir dicha cantidad. En nuestro caso estamos trabajando con las regiones V3-V4 del 16S, necesitaremos reads de al menos 250pb para encontrar contigs. Esto se debe a la gran variabilidad de esta región.

-Chimeric filtering: en este paso se eliminan las quimeras. Las quimeras son secuencias de artefactos formadas por dos o más secuencias biológicas unidas incorrectamente.

Podemos crear un resumen visualizable de cada output con los siguientes comandos:

```
> qiime metadata tabulate \
--m-input-file stats-dada2.qza \
--o-visualization stats-dada2.qzv
```

```
> qiime feature-table summarize \
--i-table table.qza \
--o-visualization table.qzv
> qiime feature-table tabulate-seqs \
--i-data rep-seqs.qza \
--o-visualization rep-seqs.qzv
```

En este caso nos saltaremos todos los estudios de diversidad del manual anterior y pasaremos directamente al análisis taxonómico. En este punto nos descargaremos un clasificador taxonómico entrenado basado en secuencias del 16s rRNA.

```
> wget \ #Descargamos el clasificador
-O "silva-138-99-nb-classifier.qza" \
"https://data.qiime2.org/2022.2/common/silva-138-99-nb-classifier.qza"
```

> qiime feature-classifier classify-sklearn \ #Realizamos el análisis taxonómico

--i-classifier ../silva-138-99-nb-classifier.qza\

--i-reads rep-seqs.qza \

--o-classification taxonomy.qza

> qiime metadata tabulate \ #Creamos un artefacto visualizable

--m-input-file taxonomy.qza \

--o-visualization taxonomy.qzv

Una vez obtenido el análisis taxonómico podemos pasar a estudiar la abundancia relativa.

> qiime taxa barplot \

--i-table table.qza \

--i-taxonomy taxonomy.qza \

--o-visualization taxplot.qzv

Nota: Este comando nos devuelve un gráfica con la abundancia relativa poblaciones bacterianas en la muestra.

Ahora vamos a crear una tabla con la abundancia relativa de cada especie. Primero, creamos una tabla con la frecuencia de cada especie dentro de la población bacteriana:

> qiime taxa collapse \

--i-table table.qza \ #Tabla de frecuencias (resultado del dada2)

--i-taxonomy taxonomy.qza \ #Artefacto con la taxonomía

--p-level 7 \ #Nivel taxonómico (1=reino, …, 6=género, 7=especie)

--o-collapsed-table phyla-table.qza #Tabla con la frecuencia de cada especie dentro de la población bacteriana

Para facilitar el estudio, generamos una tabla con la frecuencia relativa de cada especie.

> qiime feature-table relative-frequency \

--i-table phyla-table.qza \

--o-relative-frequency-table rel-phyla-table.qza

Ahora exportamos la información contenida en el artefacto a un archivo.biom. En mi caso lo he guardado en una nueva carpeta denominada “rel-table”

(creo la nueva carpeta: mkdir rel-table)

> qiime tools export \

--input-path ./rel-phyla-table.qza \

--output-path ./rel-table/

(voy a la nueva carpeta: cd rel-table)

Transformo el archivo.biom en un archivo.tsv

> biom convert -i feature-table.biom -o rel-phyla-table.tsv --to-tsv

Este nuevo archivo lo podemos abrir en una hoja de cálculo OpenOffice, hacer cat,…

Finalmente, vamos a crear un gráfico tipo Krona. Para ello usaremos nuestro archivo taxplot.qvz con las abundancias relativas y un script.py para transformarlo en un gráfico Krona. Para utilizar el script es necesario tener instalado previamente el paquete KronaTools.

(El script está en NextCloud (generate_krona) y lo único dato que hay que introducir al script es la ubicación de nuestro taxplot.tsv)

LISTO!! YA TIENES TU ANÁLISIS TAXONÓMICO.

IMPORTANTE: El clasificador basado en Silva está bien entrenado para detectar géneros, sin embargo, no es capaz de llegar a la profundidad de especies. Para conseguir un clasificador eficaz que llegase a este nivel habría que entrenarlo con una base de datos basada en microbiota humana, una posible opción sería GMrepo v2.

¿Por qué hay que entrenar el clasificador con una base de datos específica para llegar hasta el nivel taxonómico de especie? Esto se debe a que, al trabajar con con regiones del 16s, los contigs generados van a ser ambiguos. Esto significa que el contigs va estar conservado en varias especies, por lo que no vamos a poder diferenciarlas. La única manera de diferenciar qué especie tenemos sería teniendo en cuenta el contexto, por ejemplo, si el contig generado matchea con una especie de Prevotella aislada en la flora vaginal de ratón y otra especie de Prevotella aislada en una muestra de heces de un paciente con colón irritable, ¿cuál de las dos es más probable que se encuentre en la microbiota de nuestro paciente? Con gran fiabilidad podríamos afirmar que se trata de la segunda. Es por este motivo que debemos entrenar a nuestros clasificadores en un contexto concreto.

(el clasificador de qiime para silva-132 fue entrenado con scikit-learn)