QIIME 2 TUTORIALES

Importing data tutorial:

Para usar QIIME2, los datos que importes deben ser importados (guardados) en artefactos QIIME2 (.qza files). Esto no es por capricho, sino para que los datos puedan ser rastreados y manipulados correctamente.

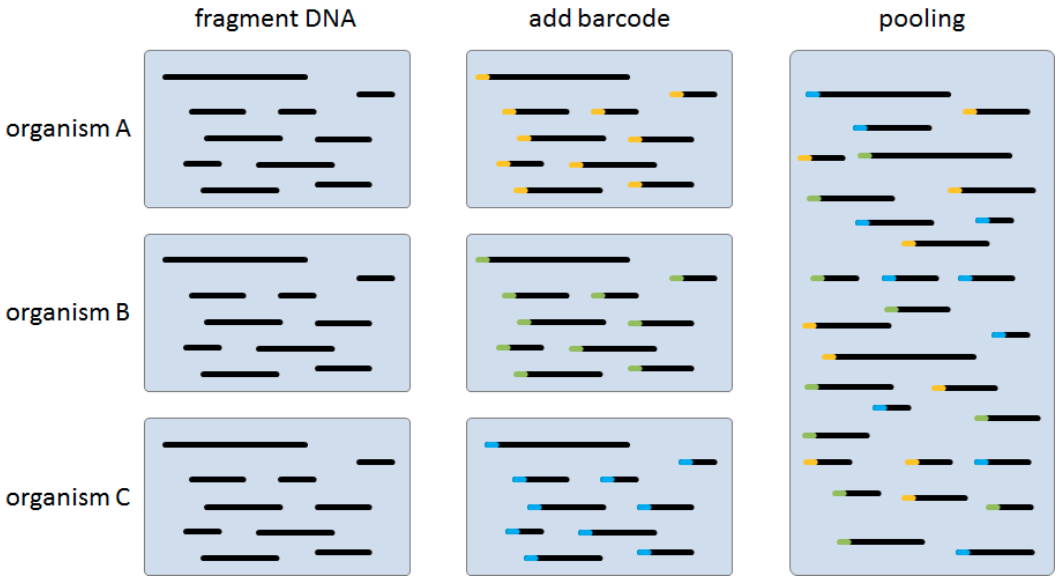
Con QIIME2, hay funciones para importar diferentes tipos de datos FASTQ:

1. FASTQ data usando el "EMP Protocol Format"
2. Multiplexed FASTQ data con códigos de barras en las secuencias.
3. FASTQ data en el formato Casava 1.8 demultiplexed
4. Cualquier otro FASTQ data.

"EMP PROTOCOL" multiplexed single-end fastq

Las secuencias single-end que están en "Earth Microbiomo Project protocol" que se importan desde esa database contienen 2 tipos de archivos:

1. un **forward.fastq.gz** que contiene las secuencias single-end
2. un **barcodes.fastq.gz** que contiene el código de barras de las secuencias (este código de barras se refiere a las secuencias únicas que fueron ligadas a cada una de las secuencias de las muestras genéticas antes de que se mezclen, ojo que no es el id de secuencia)



En este formato, las secuencias están todavía multiplexed, es decir, tenemos un archivo fastq.gz que contiene raw data de todas las muestras, estas están todas mezcladas y las secuencias obtenidas no están separadas y clasificadas por muestras. Por ello, se hace el demultiplexing. Este paso hace referencia al proceso por el cual se usa el barcode para saber qué secuencias provienen de cada muestra.

EJEMPLO:

> mkdir emp-single-end-sequences **# creamos un directorio en donde meteremos las secuencias que vamos a descargar**

```
> wget \
  -0 "emp-single-end-sequences/barcodes.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/moving-pictures/emp-single-end-sequence-barcodes.fastq.gz"
```
 # descargamos los barcodes y los metemos en el directorio "emp-single-end-sequences" bajo el nombre de "barcodes.fastq.gz".

```
> wget \
  -O "emp-single-end-sequences/sequences.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/moving-pictures/emp-single-end-sequences/sequences.fastq.gz"
```
 # descargamos las secuencias y las metemos en el directorio "emp-single-end-sequences" bajo el nombre de "sequences.fastq.gz".

```
> qiime tools import \
  --type EMPSingleEndSequences \
  --input-path emp-single-end-sequences \
  --output-path emp-single-end-sequences.qza
```
 # para trabajar con los datos deben ser artefactos QIIME2. Mediante este comando, estamos estableciendo el tipo de archivo, el camino donde se encuentran los archivos (barcodes y sequences) y el formato en el que las vamos a convertir (.qza)

"EMP PROTOCOL" multiplexed single-end fastq

Las secuencias single-end que están en "**Earth Microbiomo Project protocol**" que se importan desde esa database contienen 2 tipos de archivos:
1. Un archivo forward.fastq.gz que contiene las secuencias forward
2. Un archivo reverse.fastq.gz que contiene las secuencias reverse
3. Un archivo barcodes.fastq.gz que contiene las barcodes asociadas

EJEMPLO:

```
> mkdir emp-paired-end-sequences
> wget \
  -O "emp-paired-end-sequences/forward.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/atacama-soils/1p/forward.fastq.gz"

> wget \
  -O "emp-paired-end-sequences/reverse.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/atacama-soils/1p/reverse.fastq.gz"
> wget \
  -O "emp-paired-end-sequences/barcodes.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/atacama-soils/1p/barcodes.fastq.gz"
> qiime tools import \
  --type EMPPairedEndSequences \
  --input-path emp-paired-end-sequences \
  --output-path emp-paired-end-sequences.qza
```

Multiplexed single-end FASTQ with barcodes in sequence

Hasta ahora, los barcodes han estado separados de las secuencias. En este apartado, se tratarán multiplexed single-ended barcodes que están en secuencias (reads). Estos archivos contendrán lo siguiente:

1. Un archivo fastq.gz que contiene datos de múltiples muestras
2. Un archivo metadata con los barcodes de cada muestra para hacer el FASTQ demultiplexing.

EJEMPLO:

```
> mkdir muxed-se-barcode-in-seq
> wget \
  -O "muxed-se-barcode-in-seq/sequences.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/importing/muxed-se-barcode-in-seq.fastq.gz"
> qiime tools import \
  --type MultiplexedSingleEndBarcodeInSequence \
  --input-path muxed-se-barcode-in-seq/sequences.fastq.gz \
  --output-path multiplexed-seqs.qza
```

Multiplexed single-end FASTQ with barcodes in sequence

Los archivos con multiplexed paired-end barcodes en los reads deberían contener:

1. Un archivo forward.fastq.gz que contenga los reads forward de muchas muestras.
2. Un archivo reverse.fastq.gz que contenga los reads reverse de esas mismas muestras.
3. Un archivo metadata con los barcodes de cada muestra para hacer el FASTQ demultiplexing.

EJEMPLO:

```
> mkdir muxed-pe-barcode-in-seq
> wget \
  -O "muxed-pe-barcode-in-seq/forward.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/importing/muxed-pe-barcode-in-seq/forward.fastq.gz"
> wget \
  -O "muxed-pe-barcode-in-seq/reverse.fastq.gz" \
  "https://data.qiime2.org/2022.2/tutorials/importing/muxed-pe-barcode-in-seq/reverse.fastq.gz"
> qiime tools import \
  --type MultiplexedPairedEndBarcodeInSequence \
  --input-path muxed-pe-barcode-in-seq \
  --output-path multiplexed-seqs.qza
```

Moving pictures

Primera parte: Importar archivos y demultiplexing.

Si trabajamos con secuencias multiplex siempre tendremos que tener 2 archivos, uno con la metadata y otro con los reads. Este último puede tener reads single-end o pair-end, en nuestro caso trabajaremos con single-end.

Qiime no trabaja con cualquier tipo de archivo, sino que crea "artefactos" especificos con los que el programa puede trabajar. Para más información sobre los tipos de artefactos. Para transformar archivos/directorios en artefactos usaremos el comando "tools import".

```
1.  qiime tools import \          #Ahora los flags básicos
--type TipoDeArchivo \            #Seleccionas el tipo de artefacto que vas a crear
--input-path Ruta \               #Path to file or directory that should be imported
--output-path Ruta \              #Path where output artifact should be written.
```

En nuestro caso el input es un directorio con los reads en .fastq.gz y los barcodes en fastq.gz.

A la hora de trabajar con microbiota es necesario que nuestros reads esten demultiplexed, para ello usaremos el comando "demux"

Nota: "*Demultiplexing se refiere al paso en el procesamiento en el que usaría la información del barcode para saber qué secuencias provienen de qué muestras después de haberlas secuenciado todas juntas. Los barcodes se refieren a las secuencias únicas que se ligaron al material genético de cada una de sus muestras individuales antes de que las muestras se mezclaran. Dependiendo de su instalación de secuenciación, es posible que sus muestras ya estén divididas en archivos fastq individuales, o pueden agruparse en un solo archivo fastq con batcodes aún adjuntos para que pueda dividirlos. Si este es el caso, también debe tener un archivo de "mapeo" que le indique qué barcode corresponden a qué muestras (Archivo con la metadata).*"

```
2. qiime demux EndType \                  #Tenemos que indicar si estamos trabajando con single-end o pair-end
--i-seqs ArtefactoConLosReads \           #Artefacto con los reads (archivo.qza)
--m-barcodes-file MetadataArchivo \       #Archivo con la metadata (archivo.tsv)
--m-barcodes-column ColumnaBarcodes \     #Columna del archivo con la metadata en la que se encuentran los barcodes.
--o-per-sample-sequences NuevoArtefacto \ #El nuevo artefacto contiene las secuencias demultiplexed
--o-error-collection-details NuevoArtefacto2 \  #Contiene información sobre los errores en la collección de barcodes.
```

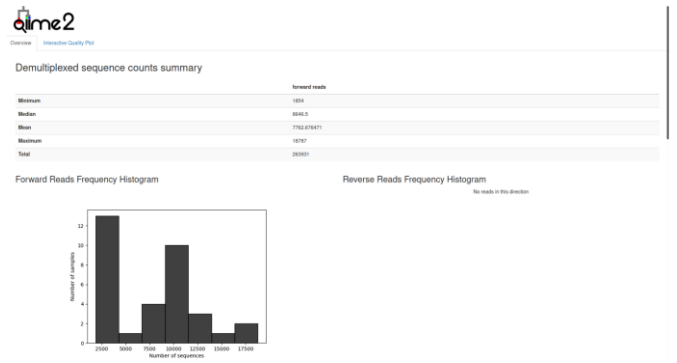
Después del demultiplexing, es útil generar un resumen de los resultados de la demultiplexation. Esto permite determinar cuántas secuencias se obtuvieron por muestra y también obtener un resumen de la distribución de las calidades de secuencia en cada posición en sus datos de secuencia.

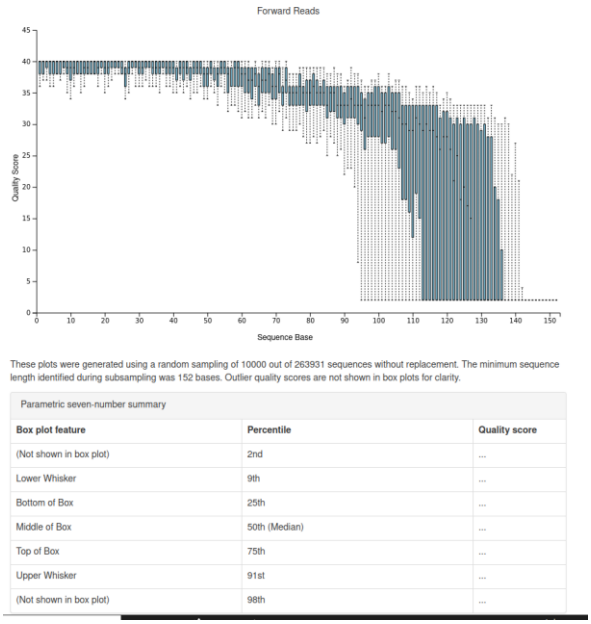
```
3. qiime demux summarize \        #En lugar de poner el tipo de read ponemos summarize
  --i-data demux.qza \            #Indicamos el artefacto demultiplexed
  --o-visualization demux.qzv     #Generamos un archivo.qzv
```

Para visualizar el resumen usaremos el comando “tools view”, este nos abre en el navegador un archivo .html.

4. qiime tools view demux.qzv #Tenemos que indcar que archivo

El .html tiene dos apartados, Overview e Interactive Quality Plot.





Segunda parte: Control de calidad de secuencias y construcción tablas de características.

Este punto es importante para mejorar la calidad de nuestros reads y manipular los mismos en pasos posteriores. Podemos generar tablas de características por varios métodos diferentes, pero en este manual solo explicaremos dos de ellos, Dada2 y Deblur. Ambos programas nos devuelven un artefacto con tabla de características por frecuencias y otro artefacto con una tabla de características por secuencias, esta última nos servirará para mapear las características con secuencias.

Dada2

5. qiime dada2 denoise-single \ #Comando para hacer las tablas de características
  --i-demultiplexed-seqs Artefacto.qza \ #Input del artefacto demultiplexed
  --p-trim-left m \ #Sirve para hacer trim de las primeras bases (m=numero de bases)
  --p-trunc-len n \ #Sirve para hacer trim de las últimas bases (n=apartir de que base trunca la secuencia)
  --o-representative-sequences Artefacto1.qza \ #Artefacto con las sequencias representativas de cada característica.
  --o-table Artefacto2.qza \ #Artefacto con la tabla de características
  --o-denoising-stats stats-dada2.qza #Artefacto con el resumen

En este ejemplo no se hace trimming de las primeras bases ya que presentan una muy buena calidad (m = 0), pero si se truncan los reads a partir de la base 120 (n = 120) porque es, aproximadamente, el punto a partir del cual la secuenciación empieza a perder calidad. Cabe destacar que este comando

tambien realiza algunas mejoras de manera predeterminada, las cuales podemos modificar. Para saber más, usa "--help".

Para visualizar un artefacto este tiene que estar en formato **.qzv**, ya que es el unico tipo de artefacto que utiliza el comando "tools view". Para transformar archivos **.qza** en **.qzv** podemos usar el comando "metadata tabulate". "metadata tabulate" genera tablas visualizables a partir de la información recogida en la metadata.

```
6. qiime metadata tabulate \
--m-input-file Artefacto.qza \          #Artefacto.qza con la metadata
--o-visualization Artefacto.qzv         #Artefacto.qzv con la tabla visualizable
```

Para visualizar el archivo basta con usar "tools view".

Deblur

Este método utiliza perfiles de error de secuencias para asociar sequence reads que sean erróneas con las secuencias biológicas (verdaderas/reales) de las que derivan, lo que nos proporciona datos sobre cada secuencia.

Esto se hace en 2 pasos:

PRIMERO: se aplica un proceso de filtrado de calidad basado en los quality scores

```
7. qiime quality-filter q-score \
   --i-demux demux.qza \  #indicamos la secuencia demultiplexed para ser filtrada por calidad
   --o-filtered-sequences demux-filtered.qza \  #indicamos el nombre de las secuencias resultantes
  --o-filter-stats demux-filter-stats.qza    #indicamos el nombre del resumen estadístico proceso de filtrado
```

SEGUNDO: utiliza el método qiime deblur denoise-16S, el cual requiere de un parámetro (--p-trim-length n) que es usado en filtrado de calidad. Este parámetro trunca las secuencias en la posición "n" que le indiques, es recomendable indicarle que corte en el punto donde el quality score de la mediana comience a bajar (normalmente entre la posición 115 y 130 en la secuencia).
En este tipo de análisis es muy importante que la longitud de los reads sea la misma para todos los experimentos de secuenciación que se comparan con el objetivo de evitar introducir un sesgo específico de estudio.

```
8. qiime deblur denoise-16S \
    --i-demultiplexed-seqs demux-filtered.qza \   #indicamos las secuencias demultiplexed para denoised
    --p-trim-length 120 \    #el 120 indica la posición donde truncará la secuencia, podemos desactivar el trimming indicando "-1"
    --o-representative-sequences rep-seqs-deblur.qza \    #indicamos el nombre de las secuencias resultantes
    --o-table table-deblur.qza \   #indicamos el nombre de la tabla resultante denoised
    --p-sample-stats \   #clasifica los stats por muestra
    --o-stats deblur-stats.qza   #nos da los stats por muestra

#Los siguientes comandos van a generar un resumen estadístico
9.  qiime metadata tabulate \
    --m-input-file demux-filter-stats.qza \       #indicamos el metadata a tabular
    --o-visualization demux-filter-stats.qzv     #para verlo
```

10. qiime deblur visualize-stats \
  --i-deblur-stats deblur-stats.qza \ #resumen estadístico del proceso Deblur
  --o-visualization deblur-stats.qzv #para verlo

11. qiime tools view deblur-stats.qzv #para visualizarlo

Para finalizar este punto vamos a generar unos resumenes visualizables a partir de la metadata obtenida. El comando de "feature-table summarize" dará información sobre cuántas secuencias están asociadas con cada muestra y con cada característica, histogramas de esas distribuciones y algunas estadísticas de resumen relacionadas.

12. qiime feature-table summarize \
  --i-table table.qza \ #Artefacto.qza con la tabla de frecuencias
  --o-visualization table.qzv \ #Artefacto.qzv resultante
  --m-sample-metadata-file sample-metadata.tsv #Archivo con la metadata original (ej: archivo.tsv)

El comando "feature-table tabulate-seqs" proporcionará un mapeo de ID de características a secuencias y proporcionará enlaces para BLAST fácilmente cada secuencia contra la base de datos nt de NCBI.

13. qiime feature-table tabulate-seqs \
  --i-data rep-seqs.qza \ #Artefacto.qza con la tabla de sequencias
  --o-visualization rep-seqs.qzv #Artefacto.qzv resultante

Tercera parte: Arbol filogenético

14. qiime phylogeny align-to-tree-mafft-fasttree \
  --i-sequences rep-seqs.qza \ #Input artefacto.qza con la tabla de secuencias repetidas
  --o-alignment aligned-rep-seqs.qza \ #Artefacto.qza con las secuencias alineadas
  --o-masked-alignment masked-aligned-rep-seqs.qza \ #Artefacto.qza con las secuencias alineadas enmascaras
  --o-tree unrooted-tree.qza \ #Artefacto.qza con el arbol filogenético sin arraigar
  --o-rooted-tree rooted-tree.qza #Artefacto.qza con el árbol filogenético arraigado

Nota: Este programa es una tubería, es decir, cada artefacto generado es utilizado por la siguiente línea de comando, es por ello que todos los outputs son obligatorios.

En este punto creamos un árbol filogenético arraigado, el cual nos será útil más adelante. Primero, la tubería usa el programa mafft para realizar una alineación de secuencias múltiples de las secuencias en nuestro FeatureData[Sequence] para crear un artefacto FeatureData[AlignedSequence] QIIME 2. A continuación, la canalización enmascara (o filtra) la alineación para eliminar las posiciones que son muy variables. Generalmente se considera que estas posiciones agregan ruido a un árbol filogenético resultante. A continuación, la canalización aplica FastTree para generar un árbol filogenético a partir de la alineación enmascarada. El programa FastTree crea un árbol sin raíz, por lo que en el paso final de esta sección se aplica el enraizamiento en el punto medio para colocar la raíz del árbol en el punto medio de la distancia más larga de punta a punta en el árbol sin raíz.

Cuarta parte: Análisis de diversidad alfa y beta

En este punto se realiza análisis de diversidad filogenética tipo alfa y beta, utilizando el árbol filogenético arraigado del punto anterior. Durante este punto se aplican diversos calculos estadísticos y se generan varios gráficos visualizables, los cuales se pueden visualizar usando Emperor. Las diferentes métricas de diversidad que incluye el plugin son:

Diversidad alfa

- Índice de diversidad de Shannon (una medida cuantitativa de la riqueza de la comunidad)
- Características observadas (una medida cualitativa de la riqueza de la comunidad)
- Diversidad filogenética de Faith (una medida cualitativa de la riqueza de la comunidad que incorpora relaciones filogenéticas entre las características)
- Uniformidad (o Uniformidad de Pielou; una medida de la uniformidad de la comunidad)

Diversidad beta

- Distancia de Jaccard (una medida cualitativa de la disimilitud de la comunidad)
- Distancia de Bray-Curtis (una medida cuantitativa de la disimilitud de la comunidad)
- Distancia UniFrac no ponderada (una medida cualitativa de la disimilitud de la comunidad que incorpora relaciones filogenéticas entre las características)
- Distancia UniFrac ponderada (una medida cuantitativa de la disimilitud de la comunidad que incorpora relaciones filogenéticas entre las características)

Para realizar todos estos análisis nos bastaremos de un único plugin, q2-diversity. Este plugin incorpora numerosos análisis de diversisdad, aunque nosotros usaremos concretamente "core-metrics-phylogenetic", el cual incluye todos los análisis de diversidad anteriormente nombrados.

15. qiime diversity core-metrics-phylogenetic \ #Plugin: diversity, herramienta: core-metrics-phylogenetic
--i-phylogeny rooted-tree.qza \ #Artefacto.qza con el árbol filogenético arraigado
--i-table table.qza \ #Artefacto.qza con la tabla de características
--p-sampling-depth n \ #Profundidad del muestreo (n > 500)
--m-metadata-file sample-metadata.tsv \ #Artefacto.tsv con la metadata original
--output-dir core-metrics-results #Crea un directorio con todos los outputs (.qza, .qzv)

NOTA: La profundidad que elijamos va a repercutir en el análisis, de tal modo, que cuanto menor sea la longitud de la secuencia de muestreo, menor confianza tendra el muestreo. Pero si utilizamos longitudes de secuencia muy grandes, estaremos excluyendo numerosas muestras fueras del proceso por lo que el análisis sera poco representativo. Es por ello que escogeremos siempre valores de profundidad cercanos a la longitud mínima pero procurando que sean los mas grande posible. En

nuestro caso, con 34 muestras, escogimos el valor 1103, lo cual nos dejaba a 3 muestras con valores entorno a 900 fuera del muestreo, pero era un valor bastante proximo a las siguientes 5 muestras. IMPORTANTE: los valores de longitud de secuencia aparecen en la tabla de características. IMPORTANTE: la longitud tiene que ser siempre superior a 500 pb.

Es importante añadir el flag –output-dir ya que se van a generar gran cantidad de outputs, por lo que es preferible agruparlos en un nuevo directorio. Dentro del directorio encontraremos ademas de numerosos artefactos.qza, también encontraremos 4 artefactos visualizables los analisis de diversidad tipo beta.

```
16. qiime diversity alpha-group-significance \                    #Más análisis de diversidad
  --i-alpha-diversity core-metrics-results/faith_pd_vector.qza \  #Metemos el faith vector generado en el paso anterio
  --m-metadata-file sample-metadata.tsv \                         #Artefacto.tsv con la metadata
  --o-visualization core-metrics-results/faith-pd-group-significance.qzv #Dirección y nombre del .qzv
```

```
17. qiime diversity alpha-group-significance \                    #Más análisis de diversidad
  --i-alpha-diversity core-metrics-results/evenness_vector.qza \  #Metemos el evennes vector
  --m-metadata-file sample-metadata.tsv \                         #Metadata original artefacto.tsv
  --o-visualization core-metrics-results/evenness-group-significance.qzv #Output artefacto.qzv
```

A continuación, analizaremos la composición de la muestra en el contexto de los metadatos categóricos usando PERMANOVA usando el comando beta-group-significance. Los siguientes comandos probarán si las distancias entre las muestras dentro de un grupo, como las muestras del mismo sitio del cuerpo (p. ej., intestino), son más similares entre sí que con las muestras de los otros grupos (p. ej., lengua, palma izquierda, y palma derecha). Si llama a este comando con el parámetro --p-pairwise, como haremos aquí, también realizará pruebas por pares que le permitirán determinar qué pares específicos de grupos (por ejemplo, lengua y tripa) difieren entre sí. Este comando puede tardar en ejecutarse, especialmente cuando se pasa --p por pares, ya que se basa en pruebas de permutación. Entonces, a diferencia de los comandos anteriores, **ejecutaremos beta-group-significance en columnas específicas de metadatos que nos interesa explorar**, en lugar de todas las columnas de metadatos a las que se aplica. Aquí aplicaremos esto a nuestras **distancias UniFrac no ponderadas**, utilizando dos columnas de metadatos de muestra, de la siguiente manera.

```
18. qiime diversity beta-group-significance \
--i-distance-matrix core-metrics-results/unweighted_unifrac_distance_matrix.qza \ #IMPORTANTE usasmos las distancias NO PONDERADAS (esta dentro de core-metrics-results/)
--m-metadata-file sample-metadata.tsv \  #Metadata original artefacto.tsv
--m-metadata-column body-site \ #Aquí indicamos en función de que categoria de datos (columna) quiero que me realice el análisis, nuestro caso lo hacemos en función de la parte del cuerpo.
--o-visualization core-metrics-results/unweighted-unifrac-body-site-significance.qzv \ #Output artefacto.qzv, los guardamos en core-metrics-results también.
--p-pairwise #Podemos indicar que nos realiza el análisis por nodos de pares de secuencias (si omitimos este flag lo realizara sin pares)
```

```
19. qiime diversity beta-group-significance \ #Igual que el anterior
  --i-distance-matrix core-metrics-results/unweighted_unifrac_distance_matrix.qza \
  --m-metadata-file sample-metadata.tsv \
  --m-metadata-column subject \ #Pero usamos otra clasificación de datos, otra columna.
  --o-visualization core-metrics-results/unweighted-unifrac-subject-group-significance.qzv \
  --p-pairwise
```

Finalmente, la ordenación es un enfoque popular para explorar la composición de la comunidad microbiana en el contexto de los metadatos de muestra. Podemos usar la herramienta Emperor para explorar gráficos de coordenadas principales (PCoA) en el contexto de metadatos de muestra. Si bien nuestro comando core-metrics-phylogenetic ya generó algunos gráficos Emperor, queremos pasar un parámetro opcional, --p-custom-axes, que es muy útil para explorar datos de series temporales. Los resultados de PCoA que se usaron en core-metrics-phylogeny también están disponibles, lo que facilita la generación de nuevas visualizaciones con Emperor. Generaremos gráficos Emperor para UniFrac y Bray-Curtis no ponderados para que el gráfico resultante contenga ejes para la coordenada principal 1, la coordenada principal 2 y los días desde el inicio del experimento. Usaremos ese último eje para explorar cómo estas muestras cambiaron con el tiempo.

20. qiime emperor plot \
 --i-pcoa core-metrics-results/unweighted_unifrac_pcoa_results.qza \ #Utilizamos los ejes principales de las Distancias UniFrac (ejes 1 y 2)
 --m-metadata-file sample-metadata.tsv \
 --p-custom-axes days-since-experiment-start \ #Añadimos el tercer eje, en este caso como varía el experimento en función del tiempo.
 --o-visualization core-metrics-results/unweighted-unifrac-emperor-days-since-experiment-start.qzv

21. qiime emperor plot \
 --i-pcoa core-metrics-results/bray_curtis_pcoa_results.qza \ #Utilizamos los ejes principales de las distancias de Bray-Curtis (ejes 1 y 2)
 --m-metadata-file sample-metadata.tsv \
 --p-custom-axes days-since-experiment-start \ #Añadimos el tercer eje, en este caso como varía el experimento en función del tiempo.
 --o-visualization core-metrics-results/bray-curtis-emperor-days-since-experiment-start.qzv

Quinta parte: Alpha rarefraction plotting

El visualizer que se crea con el siguiente comando nos devuelve varios gráficos de diversidad alfa en distintas profundidades de muestreo múltiple.

qiime diversity alpha-rarefaction \
 --i-table table.qza \ #Artefacto.qza con la tabla de características
 --i-phylogeny rooted-tree.qza \ # Artefacto.qza con el arbol filogenético arraigado
 --p-max-depth n \ #Introducimos la longitud máxima de secuencia para los cálculos estadísticos (n ~ mediana, viene en table.qzv, en nuestro caso 4010,5 ~ **4000**)
 --m-metadata-file sample-metadata.tsv \ #Metadata original
 --o-visualization alpha-rarefaction.qzv #Artefacto visualizable con las gráficas