Análisis del microbioma del oso polar (*Ursus maritimus*) mediante DNA metabarcoding en el contexto de su estado de conservación

Inés González Castellano

2026-02-20

```
#Definir la ruta base de trabajo

base_dir <- "/home/ines/TFM"

#Setear el directorio de trabajo para R

setwd(base_dir)

#Exportar la variable de R a bash

Sys.setenv(BASE_DIR = base_dir)
```

Análisis de metabarcoding

00_Raw_data

Los datos empleados se descargan del EMBL-EBI-ENA.

Study accession: PRJNA773176.

La tabla de metadatos asociada se obtiene desde SRA Run Selector, siendo el ID del BioProject el mismo que el study accession del ENA.

Renombrar archivos

Partiendo del archivo de metadatos, crear un csv con las columnas Run y Library Name (mapping.csv) y renombrar los archivos.

```
mkdir "$BASE_DIR/00_Raw_data"

cd "$BASE_DIR/00_Raw_data"

while IFS=$'\t' read -r srr sample; do
    for i in 1 2; do
        old="${srr}_${i}.fastq.gz"
        new="${sample}_${i}.fastq.gz"
        if [ -f "$old" ]; then
            echo "Renombrando \"$old\" -> \"$new\""
            mv "$old" "$new"
        else
            echo "Archivo \"$old\" no encontrado"
        fi
```

```
    done
done < mapping.csv
```

Cambiar además los _1 y _2 de los nombres por _R1 y _R2, respectivamente.

```
#R1
for f in *_1.fastq.gz; do
    mv "$f" "${f/_1/_R1}"
done

#R2
for f in *_2.fastq.gz; do
    mv "$f" "${f/_2/_R2}"
done
```

01_METADATA

Partiendo del archivo de metadatos original del artículo (SraRunTable.csv), construir el fichero de metadatos reteniendo solo la columna de identificador de muestra y la columna geo_loc_name_country, que son los dos grupos de muestras que se emplearán para las comparaciones ("Greenland" y "Alaska").

```
mkdir "$BASE_DIR/01_METADATA"
cd "$BASE_DIR/01_METADATA"

cut -d, -f20,26 SraRunTable.csv | sed 's/,/\t/g' | awk -F'\t' '{print $2 "\t" $1}' \
> metadata.txt
```

Crear el archivo de metadatos que usará QIIME 2, con el encabezado que reconoce QIIME 2:

```
echo -e "#SampleID\tLocation" > metadata_visual.txt
echo -e "#q2:types\tcategorical" >> metadata_visual.txt
tail -n +2 metadata.txt >> metadata_visual.txt
```

02_QC

Fastqc

```
mkdir "$BASE_DIR/02_QC"
cd "$BASE_DIR/02_QC"
fastqc -t 7 "$BASE_DIR/00_Raw_data/*.fastq.gz" -o .
```

Multiqc

Hacer un multiqc para las R1 y otro para las R2, para evaluarlas por separado porque será más fácil ver diferencias en las calidades.

```
mkdir "$BASE_DIR/02_QC/R1"
mkdir "$BASE_DIR/02_QC/R2"

mv *_R1* "$BASE_DIR/02_QC/R1"
mv *_R2* R2"$BASE_DIR/02_QC/R2"

cd "$BASE_DIR/02_QC/R1"
multiqc .
mv multiqc_report.html multiqc_report_R1.html
```

```
cd "$BASE_DIR/02_QC/R2"
multiqc .
mv multiqc_report.html multiqc_report_R2.html
```

FLASH2

```
mkdir "$BASE_DIR/02_QC/FLASH2"
cd "$BASE_DIR/02_QC/FLASH2"

for r1 in "$BASE_DIR/00_Raw_data/*_R1.fastq.gz"; do
    base=$(basename "$r1" "_R1.fastq.gz")
    r2="$BASE_DIR/00_Raw_data/${base}_R2.fastq.gz"
    flash2 "$r1" "$r2" -o "$base"
done
```

Crear un gráfico para ver de manera visual la distribución de los tamaños.

```
archivos <- list.files(
  file.path(base_dir, "02_QC", "FLASH2"),
  pattern = "\\.hist$",
  full.names = TRUE
)

colores <- rainbow(length(archivos))

datos <- lapply(archivos, function(f) {
  read.table(f, sep = "\t", col.names = c("tamano", "recuento"))
})

x_min <- min(sapply(datos, function(d) min(d$tamano)))
x_max <- max(sapply(datos, function(d) max(d$tamano)))
y_max <- max(sapply(datos, function(d) max(d$recuento)))

plot(NA, xlim = c(x_min, x_max), ylim = c(0, y_max),
     xlab = "Tamaño", ylab = "Frecuencia",
     main = "Distribución de tamaños",
     xaxt = "n")

ticks_x <- seq(from = x_min, to = x_max, by = 10)
axis(1, at = ticks_x)

for(i in 1:length(datos)){
  lines(datos[[i]]$tamano, datos[[i]]$recuento,
        col = colores[i], lwd = 2)
}
```

03_DEMUX

Instalar QIIME 2:

```
conda update conda

conda env create -n qiime2-amplicon-2024.5 \
--file https://data.qiime2.org/distro/amplicon/qiime2-amplicon-2024.5-py39-linux-conda.yml
```

Importar los datos a artefacto de QIIME 2.

```
conda activate qiime2-amplicon-2025.4

mkdir "$BASE_DIR/03_DEMUX"
cd "$BASE_DIR/03_DEMUX"

qiime tools import --type 'SampleData[PairedEndSequencesWithQuality]' \
--input-path manifest.tsv --input-format PairedEndFastqManifestPhred33V2 \
--output-path demux-pe.qza

qiime demux summarize --i-data demux-pe.qza --o-visualization demux-pe.qzv

qiime tools view demux-pe.qzv
```

04_DENOISING

En el denoising se incluye un paso de eliminación de los primers de amplificación por tamaño, es decir, trimeando 17 pb del primer forward y 21 pb del primer reverse.

04_DADA2_notrunc

Sin truncar, el tamaño de amplicón máximo recuperable sería de

300+300-12=588 con primers

300+300-12-17-21=550 pb sin primers.

Nota: se restan 12 pb para calcular el tamaño porque 12 son las posiciones mínimas que se deben aplicar como overlap en dada2.

```
mkdir "$BASE_DIR/04_DADA2_notrunc"
cd "$BASE_DIR/04_DADA2_notrunc"

qiime dada2 denoise-paired --i-demultiplexed-seqs ../03_DEMUX/demux-pe.qza \
--p-trim-left-f 17 --p-trim-left-r 21 --p-trunc-len-f 0 --p-trunc-len-r 0 \
--p-n-reads-learn 1000000 --o-denoising-stats DADA2stats_notrunc.qza \
--o-representative-sequences rep-seqs.qza --o-table asv-table.qza --p-n-threads 7 \
--verbose

qiime feature-table tabulate-seqs --i-data rep-seqs.qza --o-visualization rep-seqs.qzv

qiime tools export --input-path rep-seqs.qza --output-path .

qiime tools export --input-path DADA2stats_notrunc.qza --output-path ./

qiime tools export --input-path asv-table.qza --output-path ./

biom summarize-table -i feature-table.biom -o asv-table.biom.txt

biom convert -i feature-table.biom -o feature-table.tsv --to-tsv

head asv-table.biom.txt
```

-Se han inferido 2073 ASVs de 71-550 pb.

-Estadísticas (stats): al final del proceso de denoising se retienen entre el 3,12% y el 39,39% de las lecturas iniciales por muestra. La mayor parte de las lecturas se pierden en el primer paso de filtrado.

04_DADA2_244-244

En FLASH2 se observaba que el tamaño máximo de amplicones es 470 pb. Se le da un pequeño margen y se asume que es 476 pb con primers, que serían 438 pb sin primers.

244+244-12-17-21=438

```
mkdir "$BASE_DIR/04_DADA2_244-244"
cd "$BASE_DIR/04_DADA2_244-244"

qiime dada2 denoise-paired --i-demultiplexed-seqs ../03_DEMUX/demux-pe.qza \
--p-trim-left-f 17 --p-trim-left-r 21 --p-trunc-len-f 244 --p-trunc-len-r 244 \
--p-n-reads-learn 1000000 --o-denoising-stats DADA2stats_trunc244-244.qza \
--o-representative-sequences rep-seqs.qza --o-table asv-table.qza --p-n-threads 7 \
--verbose

qiime feature-table tabulate-seqs --i-data rep-seqs.qza --o-visualization rep-seqs.qzv

qiime tools export --input-path rep-seqs.qza --output-path .

qiime tools export --input-path DADA2stats_trunc244-244.qza --output-path ./

qiime tools export --input-path asv-table.qza --output-path ./

biom summarize-table -i feature-table.biom -o asv-table.biom.txt

biom convert -i feature-table.biom -o feature-table.tsv --to-tsv

head asv-table.biom.txt
```

-Obtenidos 3742 ASVs de 227-430 pb.

-Las estadísticas son mejores, se retienen al final entre el 41% y el 75% de las lecturas.

Se emplearán los resultados de este denoising con truncado 244-244 para hacer la asignación taxonómica.

05_ASSIGNMENT

La asignación taxonómica se hará contra la base de datos SILVA 138.2.

Preparación de la base de datos

Descargar la última versión de SILVA y formatearla. Manual en https://forum.qiime2.org/t/processing-filtering-and-evaluating-the-silva-database-and-other-reference-sequence-data-with-rescript/15494

```
mkdir "$BASE_DIR/SILVA_database"
cd "$BASE_DIR/SILVA_database"

qiime rescript get-silva-data --p-version '138.2' --p-target 'SSURef_NR99' \
--o-silva-sequences silva-138.2-ssu-nr99-rna-seqs.qza  \
--o-silva-taxonomy silva-138.2-ssu-nr99-tax.qza --p-include-species-labels

#Pasar secuencias de ARN a ADN

qiime rescript reverse-transcribe --i-rna-sequences silva-138.2-ssu-nr99-rna-seqs.qza \
--o-dna-sequences silva-138.2-ssu-nr99-seqs.qza
```

```
#Eliminar secuencias que tengan 5 o más bases ambiguas y homopolímeros de 8 o más bases
#de longitud.

qiime rescript cull-seqs \
--i-sequences silva-138.2-ssu-nr99-seqs.qza \
--o-clean-sequences silva-138.2-ssu-nr99-seqs-cleaned.qza

#Filtrar las secuencias por longitud y taxonomía, aplicando un filtrado diferencial de
#la longitud según la taxonomía. Es decir, se tiene en cuenta el dominio al que
#pertenecen las secuencias para decidir la longitud de filtrado.

#Archaea (16S) >= 900 pb, Bacteria (16S) >= 1200 pb y Eukaryota (18S) >=1400 pb.

qiime rescript filter-seqs-length-by-taxon \
--i-sequences silva-138.2-ssu-nr99-seqs-cleaned.qza \
--i-taxonomy silva-138.2-ssu-nr99-tax.qza --p-labels Archaea Bacteria Eukaryota \
--p-min-lens 900 1200 1400 --o-filtered-seqs silva-138.2-ssu-nr99-seqs-filt.qza \
--o-discarded-seqs silva-138.2-ssu-nr99-seqs-discard.qza

#Derreplicar las secuencias y la taxonomía. Eliminar secuencias duplicadas o
#redundantes de la base de datos para reducir el esfuerzo de computación.

qiime rescript dereplicate --i-sequences silva-138.2-ssu-nr99-seqs-filt.qza \
--i-taxa silva-138.2-ssu-nr99-tax.qza --p-mode 'uniq' \
--o-dereplicated-sequences silva-138.2-ssu-nr99-seqs-derep-uniq.qza \
--o-dereplicated-taxa silva-138.2-ssu-nr99-tax-derep-uniq.qza

#Crear el clasificador. Lo que se conoce como "entrenar".

qiime feature-classifier fit-classifier-naive-bayes \
--i-reference-reads  silva-138.2-ssu-nr99-seqs-derep-uniq.qza \
--i-reference-taxonomy silva-138.2-ssu-nr99-tax-derep-uniq.qza \
--o-classifier silva-138.2-ssu-nr99-classifier.qza
```

Hacer la asignación taxonómica contra el clasificador de SILVA entrenado:

```
mkdir "$BASE_DIR/05_ASSIGNMENT"
cd "$BASE_DIR/05_ASSIGNMENT"

qiime feature-classifier classify-sklearn \
--i-classifier ../SILVA_database/silva-138.2-ssu-nr99-classifier.qza \
--p-confidence 0.7 --i-reads ../04_DADA2_244-244/rep-seqs.qza --p-n-jobs 7 \
--verbose --o-classification taxonomy_16S.qza
```

Crear la tabla de ASVs:

```
qiime tools export --input-path taxonomy_16S.qza --output-path ./

sed -e 's/Feature /#OTU /' -e 's/Taxon/taxonomy/' -e 's/Confidence/confidence/' \
taxonomy.tsv > taxonomy_16S.tsv

biom add-metadata -i ../04_DADA2_244-244/feature-table.biom \
-o asv-table_with_tax_unfiltered.biom --observation-metadata-fp taxonomy_16S.tsv \
--sc-separated taxonomy
```

```
biom convert -i asv-table_with_tax_unfiltered.biom -o asv-table_with_tax_unfiltered.tsv \
--to-tsv --header-key taxonomy --table-type "OTU table"

sed -i 's/#OTU ID/#ASV ID/g' 16S_asv-table_with_tax.tsv
```

El archivo asv-table_with_tax_unfiltered.tsv es la tabla de ASVs con la taxonomía.

Revisar la taxonomía para evaluar cómo ha sido la asignación:

```
grep -v "#" taxonomy_16S.tsv | wc -l
grep -v "#" taxonomy_16S.tsv | grep "Unassigned" | wc -l
grep -v "#" taxonomy_16S.tsv | grep -v ";" | wc -l
grep -v "#" taxonomy_16S.tsv | grep "g__" | wc -l
grep -v "#" taxonomy_16S.tsv | grep "d__Bacteria" | wc -l
grep -v "#" taxonomy_16S.tsv | grep "d__Archaea" | wc -l
grep -v "#" taxonomy_16S.tsv | grep "d__Eukaryota" | wc -l
grep -v "#" taxonomy_16S.tsv | grep "Mitochondria" | wc -l
grep -v "#" taxonomy_16S.tsv | grep "Chloroplast" | wc -l
```

06_FILTERING

Sobre los ASVs inferidos se aplicarán filtros basados en la abundancia y en la taxonomía.

01_singletons

Excluir los singletons, es decir, ASVs que solo contienen una secuencia en todo el conjunto de datos.

```
mkdir -p "$BASE_DIR/06_FILTERING/01_singletons"
cd "$BASE_DIR/06_FILTERING/01_singletons"

qiime feature-table filter-features --i-table ../../04_DADA2_244-244/asv-table.qza \
--p-min-frequency 2 --o-filtered-table asv-table_no_singletons.qza

qiime tools export --input-path asv-table_no_singletons.qza --output-path ./

biom convert -i feature-table.biom -o asv-table_no_singletons.tsv --to-tsv \
--header-key taxonomy --table-type "OTU table"

biom summarize-table -i feature-table.biom -o asv-table_no_singletons.biom.txt

head asv-table_no_singletons.biom.txt
```

Se eliminan 19 ASVs.

02_mistagging

Eliminar los ASVs que aparecen en una frecuencia menor del 0,01% por muestra (0,0001) porque no se puede afirmar fehacientemente que pertenezcan realmente a esa muestra o se hayan generado por mistagging desde otra muestra.

```
mkdir "$BASE_DIR/06_FILTERING/02_mistagging"
cd "$BASE_DIR/06_FILTERING/02_mistagging"
```

QIIME 2 no permite trabajar con frecuencias relativas, solo con conteos absolutos de ASVs, por lo que se hará este paso de filtrado en R y después se volverá a QIIME 2. En R se definirá la frecuencia relativa por la que filtrar los ASVs (0.0001) y se pasarán todos los recuentos que sean menores a esa frecuencia a cero.

```r
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

library(BiocManager)

if (!requireNamespace("phyloseq", quietly = TRUE)) {
  BiocManager::install("phyloseq", ask = FALSE, update = FALSE)
}

library(phyloseq)

if (!requireNamespace("biomformat", quietly = TRUE)) {
  BiocManager::install("biomformat", ask = FALSE, update = FALSE)
}

library(biomformat)


otu_df <- read.table(file.path(
    base_dir,
    "06_FILTERING",
    "01_singletons",
    "asv-table_no_singletons.tsv"
  ),
  header = TRUE,
  row.names = 1,
  sep = "\t",
  check.names = FALSE,
  comment.char = ""
  )

otu_df <- otu_df[!grepl("^#", rownames(otu_df)), ]

otu <- phyloseq(otu_table(as.matrix(otu_df), taxa_are_rows = TRUE))

otu_mat <- as.matrix(otu_table(otu))

threshold_frac <- 0.0001
otu_mat_frac <- sweep(otu_mat, 2, colSums(otu_mat), FUN="/")
otu_mat_filtered <- otu_mat
otu_mat_filtered[otu_mat_frac < threshold_frac] <- 0
write.table(
  otu_mat_filtered,
  file = file.path(
    base_dir,
    "06_FILTERING",
    "02_mistagging",
    "asv-table_filteredbysample_w0.tsv"
  ),
  sep = "\t",
  quote = FALSE,
  col.names = NA
```

```
  )
```

Volver a QIIME 2 para eliminar todos los ASVs que tengan solamente ceros como conteos.

```
biom convert -i asv-table_filteredbysample_w0.tsv -o asv-table_filteredbysample_w0.biom \
--table-type="OTU table" --to-hdf5

qiime tools import --input-path asv-table_filteredbysample_w0.biom \
--type 'FeatureTable[Frequency]' --output-path asv-table_filteredbysample_w0.qza

qiime feature-table filter-features --i-table asv-table_filteredbysample_w0.qza \
--p-min-frequency 1 --o-filtered-table asv-table_filteredbysample.qza

qiime tools export --input-path asv-table_filteredbysample.qza --output-path ./

biom summarize-table -i feature-table.biom -o asv-table_filteredbysample.biom.txt

head asv-table_filteredbysample.biom.txt
```

Se eliminan 1905 ASVs.

Añadir la taxonomía a la tabla de ASVs, para ver qué se debe eliminar en los filtros de taxonomía:

```
biom add-metadata -i feature-table.biom -o asv-table_no_mistagging.biom \
--observation-metadata-fp ../../05_ASSIGNMENT/taxonomy_16S.tsv  --sc-separated taxonomy

biom convert -i asv-table_no_mistagging.biom -o asv-table_no_mistagging.tsv --to-tsv \
--header-key taxonomy --table-type "OTU table"
```

03_coamp_lowresolution

Eliminar en el mismo filtro ASVs asignados a grupos no objetivo (arqueas o eucariotas), ASVs no asignados ("Unassigned"), los asignados a secuencias organulares y aquellos cuya asignación se queda a nivel de dominio (d__Bacteria) porque no aportan información.

Revisar cuántos ASVs de estos hay:

```
grep -v "#" asv-table_no_mistagging.tsv | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "d__Bacteria" | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "d__Bacteria" | grep -v ";" | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "Unassigned" | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "d__Archaea" | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "d__Eukaryota" | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "Mitochondria" | wc -l

grep -v "#" asv-table_no_mistagging.tsv | grep "Chloroplast" | wc -l
```

En este caso habría que eliminar en total 11 ASVs: 1 unassigned, 1 asignado a arqueas, 2 que se quedan con asignación solamente a nivel de dominio Bacteria y 7 que se asignan a secuencias de origen cloroplástico.

Para eliminarlos, en primer lugar se crea el archivo ASVs_toremove.txt con los nombres de los ASVs y se le

añade el encabezado featureID.

```
mkdir "$BASE_DIR/06_FILTERING/03_coamp_lowresolution"
cd "$BASE_DIR/06_FILTERING/03_coamp_lowresolution"

grep -v "#" ../02_mistagging/asv-table_no_mistagging.tsv | grep "d__Bacteria" \
| grep -v ";" | cut -f1 > ASVs_toremove.txt
grep -v "#" ../02_mistagging/asv-table_no_mistagging.tsv | grep "d__Archaea" \
| cut -f1 >> ASVs_toremove.txt
grep -v "#" ../02_mistagging/asv-table_no_mistagging.tsv | grep "Unassigned" \
| cut -f1 >> ASVs_toremove.txt
grep -v "#" ../02_mistagging/asv-table_no_mistagging.tsv | grep "Chloroplast" \
| cut -f1 >> ASVs_toremove.txt

sed -i '1ifeatureID' ASVs_toremove.txt
```

Eliminarlos con QIIME 2:

```
qiime feature-table filter-features \
--i-table ../02_mistagging/asv-table_filteredbysample.qza \
--m-metadata-file ASVs_toremove.txt --p-exclude-ids --p-filter-empty-samples \
--o-filtered-table asv-table_assigned.qza

qiime tools export --input-path asv-table_assigned.qza --output-path ./

biom add-metadata -i feature-table.biom -o asv-table_with_tax_filtered.biom \
--observation-metadata-fp ../../05_ASSIGNMENT/taxonomy_16S.tsv  --sc-separated taxonomy

biom convert -i asv-table_with_tax_filtered.biom -o asv-table_with_tax_filtered.tsv \
--to-tsv --header-key taxonomy --table-type "OTU table"

sed -i 's/#OTU ID/#ASV ID/g' asv-table_with_tax_filtered.tsv

biom summarize-table -i asv-table_with_tax_filtered.biom \
-o asv-table_with_tax_filtered.biom.txt

head asv-table_with_tax_filtered.biom.txt
```

Se han eliminado los once ASVs. Se conservan las 93 muestras iniciales.

El archivo asv-table_with_tax_filtered.tsv es la tabla final de ASVs tras los filtrados.

07_VISUALIZATION

01_rarefaction

Crear curvas de rarefacción para las muestras antes y después de los filtrados.

Se construyen las curvas aplicando el máximo de lecturas por muestra, para asegurarse de no sesgar los resultados.

```
mkdir -p "$BASE_DIR/07_VISUALIZATION/01_rarefaction"
cd "$BASE_DIR/07_VISUALIZATION/01_rarefaction"

#Antes del filtrado
head ../../04_DADA2_244-244/asv-table.biom.txt
```

```
qiime diversity alpha-rarefaction --i-table ../../04_DADA2_244-244/asv-table.qza \
--p-min-depth 1 --p-max-depth 190527  --p-steps 50 \
--o-visualization 16S_raref_before_filtering.qzv

qiime tools export --input-path 16S_raref_before_filtering.qzv \
--output-path 16S_raref_before_filtering

qiime tools view 16S_raref_before_filtering.qzv

#Después del filtrado
head ../../06_FILTERING/03_coamp_lowresolution/asv-table_assigned.biom.txt

qiime diversity alpha-rarefaction \
--i-table ../../06_FILTERING/03_coamp_lowresolution/asv-table_assigned.qza \
--p-min-depth 1 --p-max-depth 189940  --p-steps 50 \
--o-visualization 16S_raref_after_filtering.qzv

qiime tools export --input-path 16S_raref_after_filtering.qzv \
--output-path 16S_raref_after_filtering

qiime tools view 16S_raref_after_filtering.qzv
```

Hacer las representaciones en R. Se guardan como html interactivos:

```
if (!requireNamespace("tidyverse", quietly = TRUE)) {
  install.packages("tidyverse")
}

library(tidyverse)

if (!requireNamespace("ggplot2", quietly = TRUE)) {
  install.packages("ggplot2")
}

library(ggplot2)

if (!requireNamespace("plotly", quietly = TRUE)) {
  install.packages("plotly")
}

library(plotly)

if (!requireNamespace("htmlwidgets", quietly = TRUE)) {
  install.packages("htmlwidgets")
}

library(htmlwidgets)

#Antes del filtrado

raref_before <- read_csv(
  file.path(
    base_dir,
    "07_VISUALIZATION",
```

```
    "01_rarefaction",
    "16S_raref_before_filtering",
    "observed_features.csv"
  )
)

raref_long_before <- raref_before %>%
  pivot_longer(
    cols = -`sample-id`,
    names_to = "depth_iter",
    values_to = "observed_features"
  )

raref_long_before <- raref_long_before %>%
  separate(
    depth_iter,
    into = c("depth", "iter"),
    sep = "_"
  ) %>%
  mutate(
    depth = as.numeric(str_remove(depth, "depth-")),
    iter  = as.numeric(str_remove(iter, "iter-"))
  )

  raref_mean_before <- raref_long_before %>%
    group_by(`sample-id`, depth) %>%
    summarise(
      mean_observed = mean(observed_features, na.rm = TRUE),
      .groups = "drop"
    )

raref_before_plot <-  ggplot(raref_mean_before,
                             aes(
                               x = depth,
                               y = mean_observed,
                               group = `sample-id`,
                               color = `sample-id`,
                               text = paste(
                                 "Muestra:", `sample-id`,
                                 "<br>Profundidad de secuenciación:", depth,
                                 "<br>ASVs observados:", round(mean_observed, 1)
                               )
                             )
) +
  geom_line(alpha = 0.6, size = 2) +
  theme_bw() +
  labs(
    x = "Profundidad de secuenciación",
    y = "ASVs observados",
    color = "Muestra"
  ) +
  theme(
  legend.position = "right",
```

```
    legend.text = element_text(size = 17),
    legend.title = element_text(size = 18),
    axis.text  = element_text(size = 17),
    axis.title =  element_text(size = 17)
  ) +
  guides(
    color = guide_legend(ncol = 2)
  )

raref_before_plot_interactive <- ggplotly(
  raref_before_plot,
  tooltip = "text"
)

saveWidget(
  raref_before_plot_interactive,
  file = file.path(
    base_dir,
    "07_VISUALIZATION",
    "01_rarefaction",
    "rarefaction_before_filtering.html"
  ),
  selfcontained = TRUE
)


#Después del filtrado

raref_after <- read_csv(
  file.path(
    base_dir,
    "07_VISUALIZATION",
    "01_rarefaction",
    "16S_raref_after_filtering",
    "observed_features.csv"
  )
)

raref_long_after <- raref_after %>%
  pivot_longer(
    cols = -`sample-id`,
    names_to = "depth_iter",
    values_to = "observed_features"
  )

raref_long_after <- raref_long_after %>%
  separate(
    depth_iter,
    into = c("depth", "iter"),
    sep = "_"
  ) %>%
  mutate(
    depth = as.numeric(str_remove(depth, "depth-")),
```

```
    iter  = as.numeric(str_remove(iter, "iter-"))
  )

  raref_mean_after <- raref_long_after %>%
    group_by(`sample-id`, depth) %>%
    summarise(
      mean_observed = mean(observed_features, na.rm = TRUE),
      .groups = "drop"
    )

raref_after_plot <- ggplot(raref_mean_after,
                           aes(x = depth,
                               y = mean_observed,
                               group = `sample-id`,
                               color = `sample-id`,
                               text = paste(
                                 "Muestra:", `sample-id`,
                                 "<br>Profundidad de secuenciación:", depth,
                                 "<br>ASVs observados:", round(mean_observed, 1)
                               )
                           )
) +
  geom_line(alpha = 0.6) +
  theme_bw() +
  labs(
    x = "Profundidad de secuenciación",
    y = "ASVs observados",
    color = "Muestra"
  ) +
  theme(legend.position = "none")

raref_before_plot_interactive <- ggplotly(
  raref_after_plot,
  tooltip = "text"
)

saveWidget(
  raref_after_plot_interactive,
  file = file.path(
    base_dir,
    "07_VISUALIZATION",
    "01_rarefaction",
    "rarefaction_after_filtering.html"
  ),
  selfcontained = TRUE
)
```

En la curva después del filtrado se observa que no se cierra la curva para tres muestras: SBS-56-MK, EG-12-MK y SBS-5-MK. Se deben eliminar.

06_FILTERING

04_rarefaction

Hacer el archivo samplestoremove.txt con el nombre de las tres muestras a eliminar y el encabezado sampleID. Se eliminan con QIIME 2.

```
mkdir "$BASE_DIR/06_FILTERING/04_rarefaction"
cd "$BASE_DIR/06_FILTERING/04_rarefaction"

printf "sampleID\nSBS-56-MK\nEG-12-MK\nSBS-5-MK\n" > samplestoremove.txt

qiime feature-table filter-samples \
--i-table ../03_coamp_lowresolution/asv-table_assigned.qza \
--m-metadata-file samplestoremove.txt --p-exclude-ids \
--o-filtered-table asv-table_no_samples_raref.qza

qiime tools export --input-path asv-table_no_samples_raref.qza --output-path .

biom add-metadata -i feature-table.biom -o asv-table_with_tax_filtered_raref.biom \
--observation-metadata-fp ../../05_ASSIGNMENT/taxonomy_16S.tsv  --sc-separated taxonomy

biom convert -i asv-table_with_tax_filtered_raref.biom \
-o asv-table_with_tax_filtered_raref.tsv --to-tsv \
--header-key taxonomy --table-type "OTU table"

sed -i 's/#OTU ID/#ASV ID/g' asv-table_with_tax_filtered_raref.tsv

biom summarize-table -i asv-table_with_tax_filtered_raref.biom \
-o asv-table_with_tax_filtered_raref.biom.txt

head asv-table_with_tax_filtered_raref.biom.txt
```

Al eliminar estas tres muestras se pierden 49 ASVs que eran exclusivos de ellas.

La tabla asv-table_with_tax_filtered_raref.tsv es la tabla de ASVs tras los filtrados de ASVs y muestras.

07_VISUALIZATION

02_barplots

Construir los barplots con QIIME 2 partiendo del archivo asv-table_no_samples_raref.qza:

```
mkdir "$BASE_DIR/07_VISUALIZATION/02_barplots"
cd "$BASE_DIR/07_VISUALIZATION/02_barplots"

qiime taxa barplot \
--i-table ../../06_FILTERING/04_rarefaction/asv-table_no_samples_raref.qza \
--i-taxonomy ../../05_ASSIGNMENT/taxonomy_16S.qza \
--m-metadata-file ../../01_METADATA/metadata_visual_raref.txt --o-visualization QIIME2_16S.qzv

qiime tools view QIIME2_16S.qzv

qiime tools export --input-path QIIME2_16S.qzv --output-path barplots_export/
```

Representar los barplots en R. Se muestra solo el código para hacer uno de los niveles, pero se representarían

igual los 7 niveles. Se guardan como imagen png y como html interactivos.

```
library(tidyverse)
library(ggplot2)

barplot_class <- read.csv(
  file.path(
    base_dir,
    "07_VISUALIZATION",
    "02_barplots",
    "barplots_export",
    "level-3.csv"
  ),
  row.names = 1,
  check.names = FALSE
)

barplot_class_long <- barplot_class %>%
  rownames_to_column("Muestra") %>%
  pivot_longer(
    cols = where(is.numeric),
    names_to = "Taxon",
    values_to = "Recuento"
  )

barplot_class_long <- barplot_class_long %>%
  group_by(Muestra) %>%
  mutate(Frecuencia = Recuento / sum(Recuento) * 100) %>%
  ungroup()

barplot_class_long <- barplot_class_long %>%
  mutate(
    Clase = str_extract(Taxon, "c__[^;]*")
  )

barplot_class_image <- ggplot(barplot_class_long,
                              aes(x = Muestra, y = Frecuencia, fill = Clase,
                                                  text = paste(
                                                    "Muestra:", Muestra,
                                                    "<br>Clase:", Clase,
                                                    "<br>Abundancia (%):",
                                                      round(Frecuencia, 2)
                                                  ))) +
  geom_bar(stat = "identity") +
  ylab("Abundancia relativa") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right",
        legend.text = element_text(size = 17),
    legend.title = element_text(size = 18),
    axis.text.y = element_text(size = 17),
    axis.title =  element_text(size = 17)
  ) +
```

```
  guides(
    fill = guide_legend(ncol = 1)
  )

#Gardarlos como imagen
ggsave(
  file = file.path(
    base_dir,
    "07_VISUALIZATION",
    "02_barplots",
    "barplot_clase.pdf"
  ),
  plot = barplot_class_image,
       width = 25,
       height = 6
   )

#Guardarlos como html interactivos

library(plotly)
library(htmlwidgets)

barplot_class_interactive <- ggplotly(
  barplot_class_image,
  tooltip = "text"
)


saveWidget(
  barplot_class_interactive,
  file = file.path(
    base_dir,
    "07_VISUALIZATION",
    "02_barplots",
    "barplot_class_interactive.html"
  ),
  selfcontained = TRUE
)
```

08_ALPHA_BETA

01_normalization

Normalizar los resultados al número menor de lecturas por muestra (8965).

```
mkdir -p "$BASE_DIR/08_ALPHA_BETA/01_normalization"
cd "$BASE_DIR/08_ALPHA_BETA/01_normalization"

qiime diversity core-metrics \
--i-table ../../06_FILTERING/04_rarefaction/asv-table_no_samples_raref.qza \
--p-sampling-depth 8965 --m-metadata-file ../../01_METADATA/metadata_visual_raref.txt \
--p-n-jobs 7 --output-dir ./core-metrics-results

cd "$BASE_DIR/08_ALPHA_BETA/01_normalization/core-metrics-results"
```

```
qiime tools export --input-path rarefied_table.qza --output-path ./

biom add-metadata -i feature-table.biom -o rarefied_table.biom \
--observation-metadata-fp ../../../05_ASSIGNMENT/taxonomy_16S.tsv \
--sc-separated taxonomy

biom convert -i rarefied_table.biom -o rarefied_table.tsv --to-tsv \
--header-key taxonomy --table-type "OTU table"

biom summarize-table -i rarefied_table.biom -o asv-table_beyond_domain.biom.txt

head asv-table_beyond_domain.biom.txt
```

Se han perdido 96 ASVs pero al rarificar al número menor de lecturas por muestra no se pierde ninguna muestra.

02_alpha

Calcular los índices de Shannon y de Chao1.

```
mkdir "$BASE_DIR/08_ALPHA_BETA/02_alpha"
cd "$BASE_DIR/08_ALPHA_BETA/02_alpha"

#Índice de Shannon (H')

qiime diversity alpha \
--i-table ../01_normalization/core-metrics-results/rarefied_table.qza \
--p-metric 'shannon' --o-alpha-diversity Shannon_diversity.qza

qiime diversity alpha-group-significance --i-alpha-diversity Shannon_diversity.qza \
--m-metadata-file ../../01_METADATA/metadata_visual_raref.txt \
--o-visualization Shannon-group-significance.qzv

qiime tools view Shannon-group-significance.qzv

qiime tools export --input-path Shannon_diversity.qza --output-path .

#Exporta la tabla con el índice por muestra. Construir un Excel con los dos índices
#(Indexes_alpha_per_location.xlsx)


#Índice de Chao1

qiime diversity alpha \
--i-table ../01_normalization/core-metrics-results/rarefied_table.qza \
--p-metric 'chao1' --o-alpha-diversity Chao1_diversity.qza

qiime diversity alpha-group-significance --i-alpha-diversity Chao1_diversity.qza \
--m-metadata-file ../../01_METADATA/metadata_visual_raref.txt \
--o-visualization Chao1-group-significance.qzv

qiime tools view Chao1-group-significance.qzv

qiime tools export --input-path Chao1_diversity.qza --output-path .
```

```
#Índice de Simpson (1-D)

qiime diversity alpha \
--i-table ../01_normalization/core-metrics-results/rarefied_table.qza \
--p-metric 'simpson' --o-alpha-diversity Simpson_diversity.qza

qiime diversity alpha-group-significance --i-alpha-diversity Simpson_diversity.qza \
--m-metadata-file ../../01_METADATA/metadata_visual_raref.txt \
--o-visualization Simpson-group-significance.qzv

qiime tools view Simpson-group-significance.qzv

qiime tools export --input-path Simpson_diversity.qza --output-path .
```

Hay que comparar dos grupos de muestras independientes (no pareadas), por lo que hay que hacer el test de Mann–Whitney–Wilcoxon (MWW). La representación gráfica de los boxplots y el test de MWW se hace en R. Se calcula el p-valor corregido.

```
if (!requireNamespace("ggpubr", quietly = TRUE)) {
  install.packages("ggpubr")
}

library(ggpubr)

df_Chao1 <- read.delim(
  file.path(
    base_dir,
    "08_ALPHA_BETA",
    "02_alpha",
    "alpha-diversity_Chao1.tsv"
  ),
  comment.char = "#"
  )

comparisons <- list(
  c("Greenland", "USA")
)

boxplot_Chao1 <- ggboxplot(
  df_Chao1,
  x = "Población",
  y = "Chao1",
  color = "black",
  fill = "Población",
  add = "jitter",
  add.params = list(color = "black", size = 3)
) +
  stat_compare_means(
    comparisons = comparisons,
    method = "wilcox.test",
    p.adjust.method = "BH",
     size = 8) +
  scale_x_discrete(
    labels = c("Greenland" = "Greenland",
```

```
                "USA" = "Alaska")
  ) +
  scale_fill_manual(
    values = c(
      "Greenland" = "#00BFC4",
      "USA"       = "#F8766D"
    ),
    labels = c(
      "Greenland" = "Greenland",
      "USA"       = "Alaska"
    )
  ) +
  labs(y = "Chao1",
       x = "Población"
  ) +
  theme(legend.position = "none",
        axis.title.x = element_text(size = 24),
        axis.title.y = element_text(size = 24),
        axis.text.x  = element_text(size = 24),
        axis.text.y  = element_text(size = 24)
  )

ggsave(
  file = file.path(
    base_dir,
    "08_ALPHA_BETA",
    "02_alpha",
    "boxplot_Chao1.pdf"
  ),
  plot = boxplot_Chao1,
  width = 12,
  height = 10
)
```

Solo se muestra cómo representar un gráfico pero serían igual los de los otros dos índices.

Guardar los tres gráficos en una misma imagen:

```
boxplots_all <- ggarrange(boxplot_Shannon, boxplot_Chao1, boxplot_Simpson,
  ncol = 3,
  nrow = 1,
  labels = c("A", "B", "C"),
  font.label = list(size = 26, face = "bold")
)

ggsave(
  file = file.path(
    base_dir,
    "08_ALPHA_BETA",
    "02_alpha",
    "boxplots_all.pdf"
  ),
  plot = boxplots_all,
  width = 22,
  height = 10
```

```
)
```

Además, sacar el valor medio por población para cada índice y para construir una tabla (Indexes_alpha_per_location.xlsx) con el p-valor de la comparación. El p-valor lo muestran los gráficos, la media se hace en R (ejemplo para Chao1):

```
if (!requireNamespace("dplyr", quietly = TRUE)) {
  install.packages("dplyr")
}

library(dplyr)

df_summary_Chao1 <- df_Chao1 %>%
  group_by(Población) %>%
  summarise(
    across(Chao1, list(mean = ~mean(., na.rm = TRUE),
                       SD = ~sd(., na.rm = TRUE)))
  )
```

03_beta

Calcular la matriz de distancias Bray-Curtis con QIIME 2.

```
mkdir "$BASE_DIR/08_ALPHA_BETA/03_beta"
cd "$BASE_DIR/08_ALPHA_BETA/03_beta"

qiime diversity beta \
--i-table ../01_normalization/core-metrics-results/rarefied_table.qza \
--p-metric braycurtis --p-n-jobs 7 --o-distance-matrix BC-distance-matrix.qza

qiime tools export --input-path BC-distance-matrix.qza --output-path .
```

Representar el nMDS en R:

```
if (!requireNamespace("vegan", quietly = TRUE)) {
  install.packages("vegan")
}

library(vegan)
library(ggplot2)
library(dplyr)

bc <- read.table(
  file.path(
    base_dir,
    "08_ALPHA_BETA",
    "03_beta",
    "distance-matrix.tsv"
  ),
  header = TRUE,
  row.names = 1,
  sep = "\t",
  check.names = FALSE
)

metadata <- read.table(
```

```
  file.path(
    base_dir,
    "01_METADATA",
    "metadata_visual_raref.txt"
  ),
  header = TRUE,
  sep = "\t"
)

bc_dist <- as.dist(bc)

nmds <- metaMDS(bc_dist, k = 2, trymax = 100)

coords <- as.data.frame(scores(nmds))
coords$sample <- rownames(coords)

coords <- coords %>%
  left_join(metadata, by = "sample")

nmds <- ggplot(coords, aes(x = NMDS1, y = NMDS2, color = Location)) +
  geom_point(size = 5) +
  stat_ellipse(aes(group = Location), type = "t", linetype = 2, level = 0.95) +
  scale_color_manual(
    values = c("Greenland" = "#00BFC4", "USA" = "#F8766D"),
    labels = c("Greenland" = "Greenland", "USA" = "Alaska")) +
  theme_minimal(base_size = 14) +
  theme_classic() +
  labs(title = NULL,
       x = "NMDS1",
       y = "NMDS2",
       color = "Población") +
  theme(legend.position = "right",
        axis.title = element_text(size = 18),
        axis.text  = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text  = element_text(size = 18)
        )

ggsave(
   file = file.path(
    base_dir,
    "08_ALPHA_BETA",
    "03_beta",
    "nmds.pdf"
  ),
  plot = nmds,
  width = 16,
  height = 10
  )
```

Calcular el PERMANOVA:

```
library(vegan)
```

```r
adonis_result <- adonis2(bc_dist ~ Location, data = metadata, permutations = 999)
print(adonis_result)
```

Calcular PERMDISP (dispersión intragrupo):

```r
bc_dist <- as.dist(bc)

bd <- betadisper(bc_dist, metadata$Location)
permutest_result <- permutest(bd)
print(permutest_result)


plot(bd, main = "Dispersión de Bray-Curtis por grupo (PERMDISP)")
```