Proyecto2

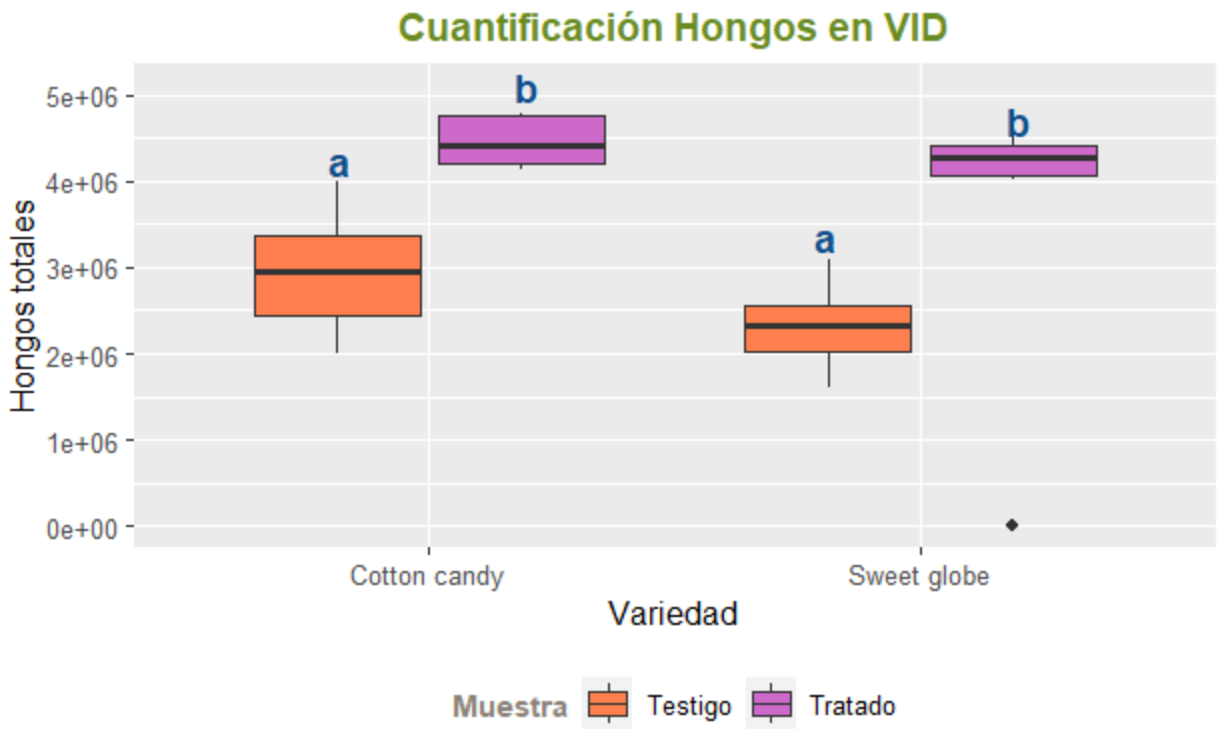
Code

Librería

Hide

```
library(dada2)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(patchwork)
library(RColorBrewer)
library(ggbreak)
library(plotrix)
library(ggsignif)
```

Gráfica



Seleccion y preparacion de archivos

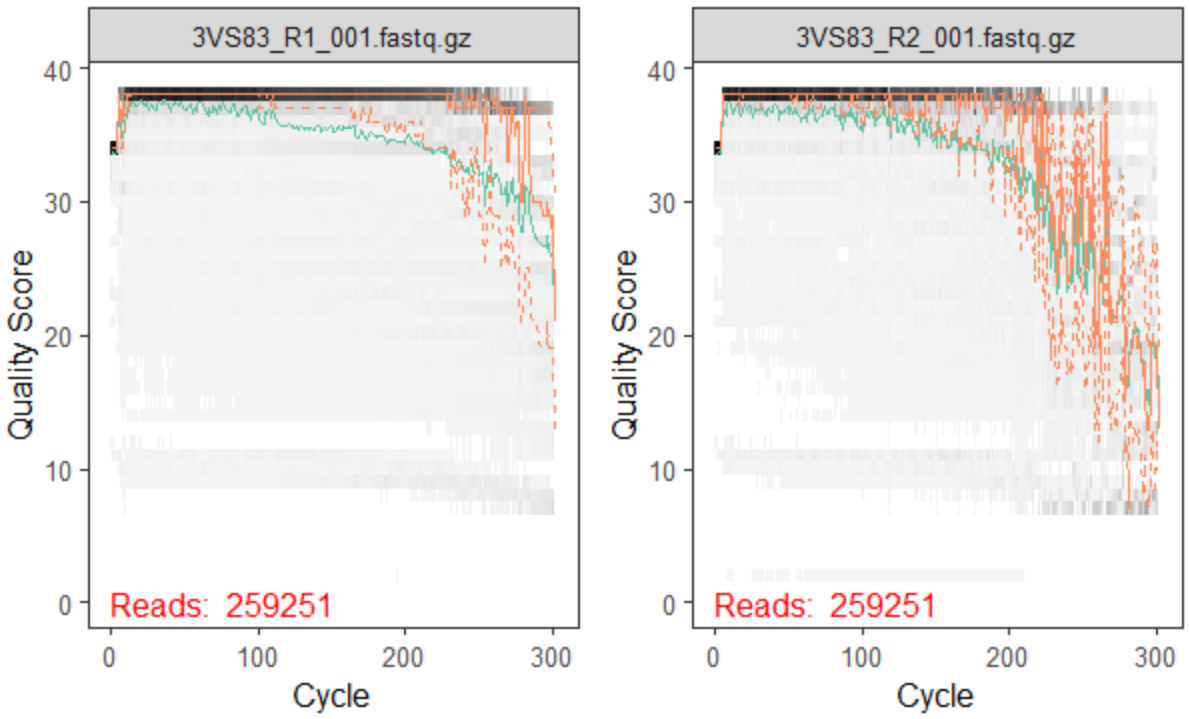
Hide

```
## Fijar el camino al directorio donde estan mis muestras

path <- "~/capR/curso/curso_Innovak/Secuenciacion_proyecto2/"

list.files(path)

## Ahora leeremos los nombres de nuestras muestras y los separaremos en objetos entre forward y reverse reads

# forward

fnFs <- sort(list.files(path, pattern = "_R1_001.fastq.gz", full.names = TRUE))

# REVERSE

fnRs <- sort(list.files(path, pattern = "_R2_001.fastq.gz", full.names = TRUE))
```

Hide

```
## Extract sample names

sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`,1) # Este codigo funciona dependiendo de como estan escrito el nombre de sus datos
```

Inspeccionar perfiles de calidad



En esta parte se ve la calidad de las muestras, en este caso la muestra forward decidí cortar en 260, basandome en el número de bases que tenemos por secuencia entre 30-40 cuando empieza a caer menor a 30 ya que es cuando la calidad de la muestra no es tan buena, fijandome en la línea verde y naranja donde empezaron a caer los picos.

El 260 lo decidí en el segundo intento de corte, ya que en el primero había puesto 240 y al final que corrí todo, hubo muy pocas lecturas (352) por lo cual decidí ampliar un poco más, aún donde no se viera muy bajo los picos y lo deje en 260.

En la muestra reverse empieza a caer los picos a partir del 200 y es por eso que deje ese corte. Y

Estos cortes se realizan con el fin de disminuir el numero de errores, ya que al momento de secuenciar generalmente los últimos nucleotidos ya son de baja calidad.

Filtrar y cortar

Primero se creó una carpeta donde van a estar nuestras muestras filtradas

Hide

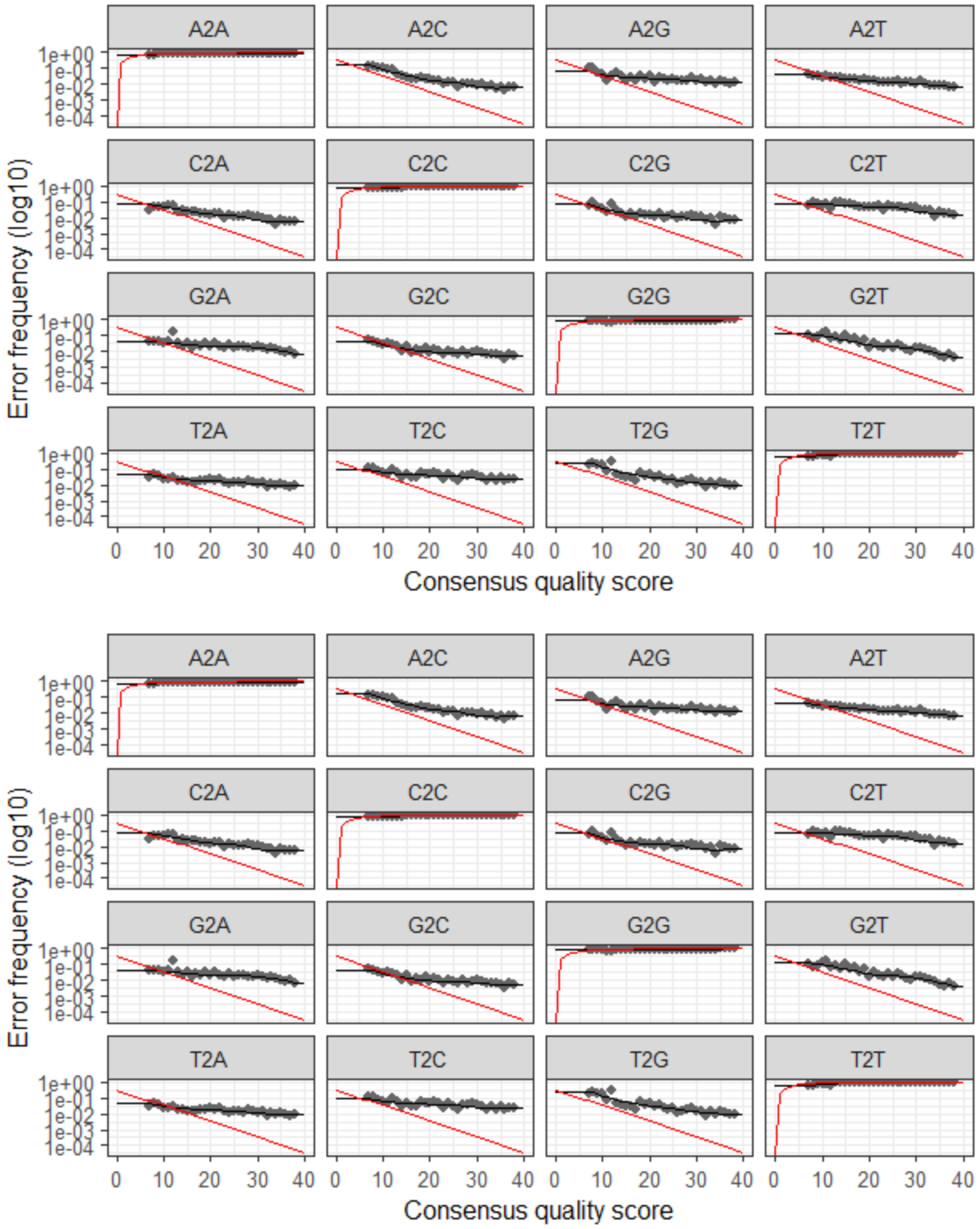
```
# Guardando el camino a nuestras muestras filtradas en un objeto nuevo

filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz")) #forward
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz")) #reverse

# Asignando los nombres de las muestras a nuestros objetos nuevos

names(filtFs) <- sample.names
names(filtRs) <- sample.names
```

Aquí es importante tomar en cuenta como se removeran las lecturas:

* `truncQ= 2` le deje el 2 ya que es el de buena calidad

- `maxN=0` remueve las lecturas que tenga nucleotidos no reconocidos
- `truncLen` Aquí es el paso donde se pone en donde cortar para saber el numero de bases que se van a dejar, yo aquí establecí 260,200.
- `maxEE=c(,)` Es establecer los errores esperados, en esta ocasión decidí dejar (5,5) ya que la momento de ponerme más estricta y bajar el parametro a (2,5), bajaba mucho el número de lecturas.

Esta parte estima que tan probable es que una base en realidad sea otra, obteniendo la tasa de error.



Para la interpretación de los gráficos de error es importante observar la línea negra (tasa de error estimada) se mantenga y que este sobre la línea roja (tasas de error esperadas) y está última vaya disminuyendo

#Inferencia de la muestra

En esta parte es donde se retiran todos los errores de secuenciacion para dejar unicamente los miembros reales de la comunidad que fue secuenciada

Hide

```
dadaRs_nopool <- dada(filtRs, err=errR, multithread = TRUE)
```

```
Sample 1 - 240024 reads in 127746 unique sequences.
```

Hide

```
save(dadaRs_nopool, file = "dadaRs_nopoolP2.RData")
```

Unir las lecturas forward y reverse

Aquí se unen las lecturas conforme al corte que se estableció anteriormente, # Uniendo las lecturas forward y también se ve cuantos pares serán rechazados.

Hide

```
mergers <- mergePairs(dadaFs_nopool, filtFs, dadaRs_nopool, filtRs, verbose = TRUE)
```

```
58888 paired-reads (in 11718 unique pairings) successfully merged out of 187318 (in 78357 pairin
gs) input.
```

Hide

```
save(mergers, file = "mergersP2.RData")
```

Hide

```
table(nchar(getSequences(seqtab)))
```

```
 260  262  273  295  296  297  298  323  327  335  380  381  394
   1    1    1    8    1    3    7    1    1    1    1    1    1
 397  404  409  410  411  416  417  418  419  420  421  422  433
   1    1    1    1    1    2    1    1    1    1    1    1    4
 434  435  437  438  439  440  441  442  443  444  445  446  447
   2    1   22   26  547 6373  969 1035  127  378 2000   72   92
 448
  29
```

Checar la longitud de las secuencias nos sirve para verificar que no pasen del número de bases que vimos en los primeros gráficos que eran 300, aquí se observa que algunas si pasan del 300

Quimeras

Se quitan pedazos de ADN que se unieron cuando no debían

Hide

```
#incluyendo abundancias
sum(seqtab.nochim)/sum(seqtab) # porcentaje de secuencias no quimericas que se mantuvieron
```

```
[1] 0.5111907
```

El porcetanje de secuencias no quimericas es 51%, es decir que es el porcentaje de nuestras lecturas que se mantuvieron.

Seguimiento del proceso

Hide

```
out <- read.csv("~/capR/curso/curso_Innovak/Proyecto2/conteo_reads_proyecto2.csv")

# Primero crearemos una funcion
getN <- function(x) sum(getUniques(x))


# Creamos una nueva tabla llamada track
track <- cbind(out, # Paso1: filtrado y corte
               getN(dadaFs_nopool), #paso2:calculamos errores dentro de dada
               getN(dadaRs_nopool),# paso3: denoising
               getN(mergers),#paso4: unir muestras
               rowSums(seqtab.nochim)) #paso5: quitar quimeras

# Nombramos nuestras filas y columnas
colnames(track) <- c("Sample_names","input", "filtered", "denoisedF", "denoisedR", "merged", "no
nchim")

rownames(track) <- sample.names # no siempre es necesario

#guardamos esta tabla

write.csv(track, "~/capR/curso/curso_Innovak/Proyecto2//Seguimiento_dada.csv") # para guardar un
a tabla
```

Está tabla es un resumen del proceso, en ella nos arroja los siguientes valores:

Input: Número de lecturas obtenidas de la muestra Filtered: Número de lecturas ya filtradas *Denoised:Quitar todos los errores* Merged:Unir las muestras *Nonchim:Quitar quimeras

Asignar taxonomía

Hide

```
taxa <- assignTaxonomy(seqtab.nochim,                     "~/capR/curso/curso_Innovak/Secuenciac
ion/Taxa/silva_nr99_v138.1_train_set.fa.gz", multithread = TRUE)
```

Viendo la tabla donde aparecen los géneros, la mayoría si tiene a cual pertenece, pero si aparecen algunos con NA.

Asignar especies

Está parte no es necesaria, pero quería verificar si me daría alguna especie pero en este caso no se encontraron, por lo que aparece NA.

Hide

```
taxa <- addSpecies(taxa,"~/capR/curso/curso_Innovak/Secuenciacion/Taxa/silva_species_assignment_
v138.1.fa.gz")

save(taxa, file = "taxa_ch_p2.RData")
```