H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course: Int_BT_2019

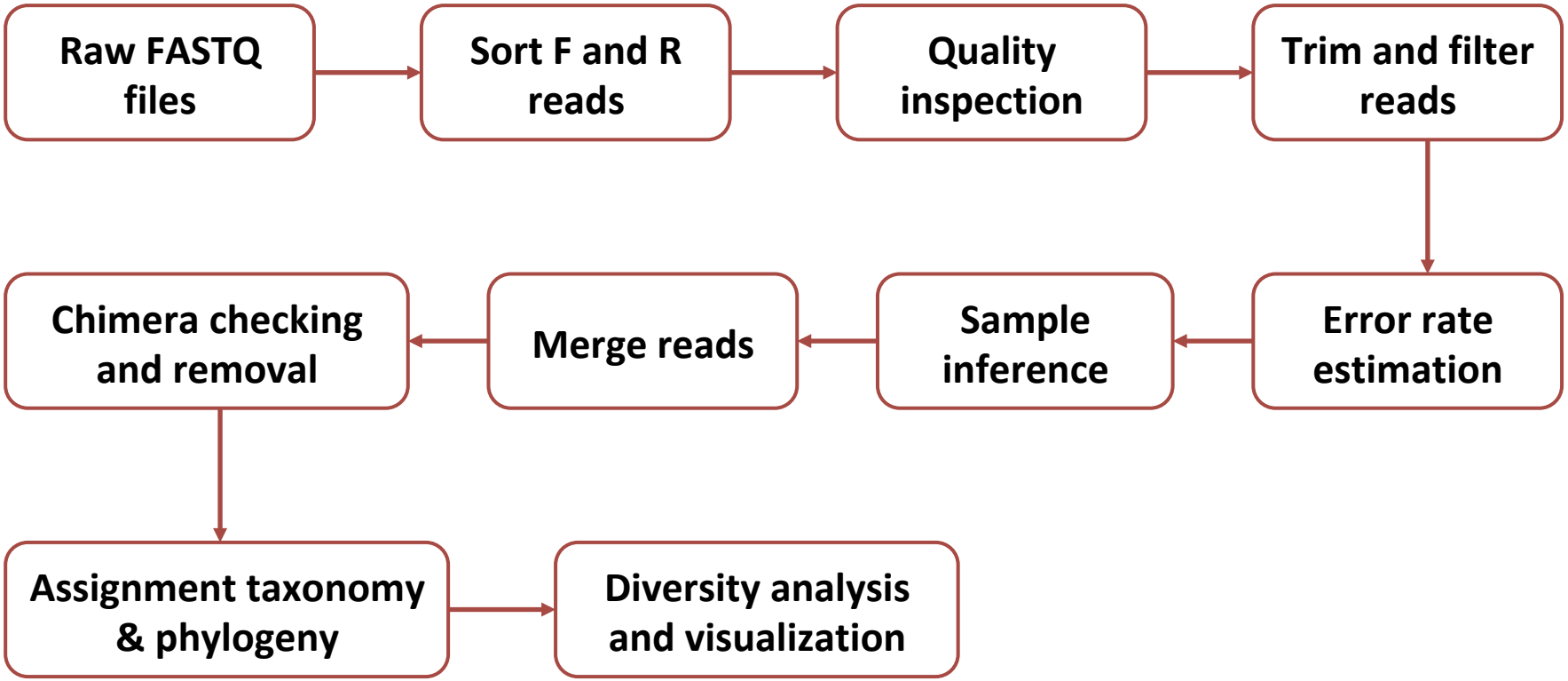
16S analysis pipeline

QC and ASV picking using the dada2 pipeline

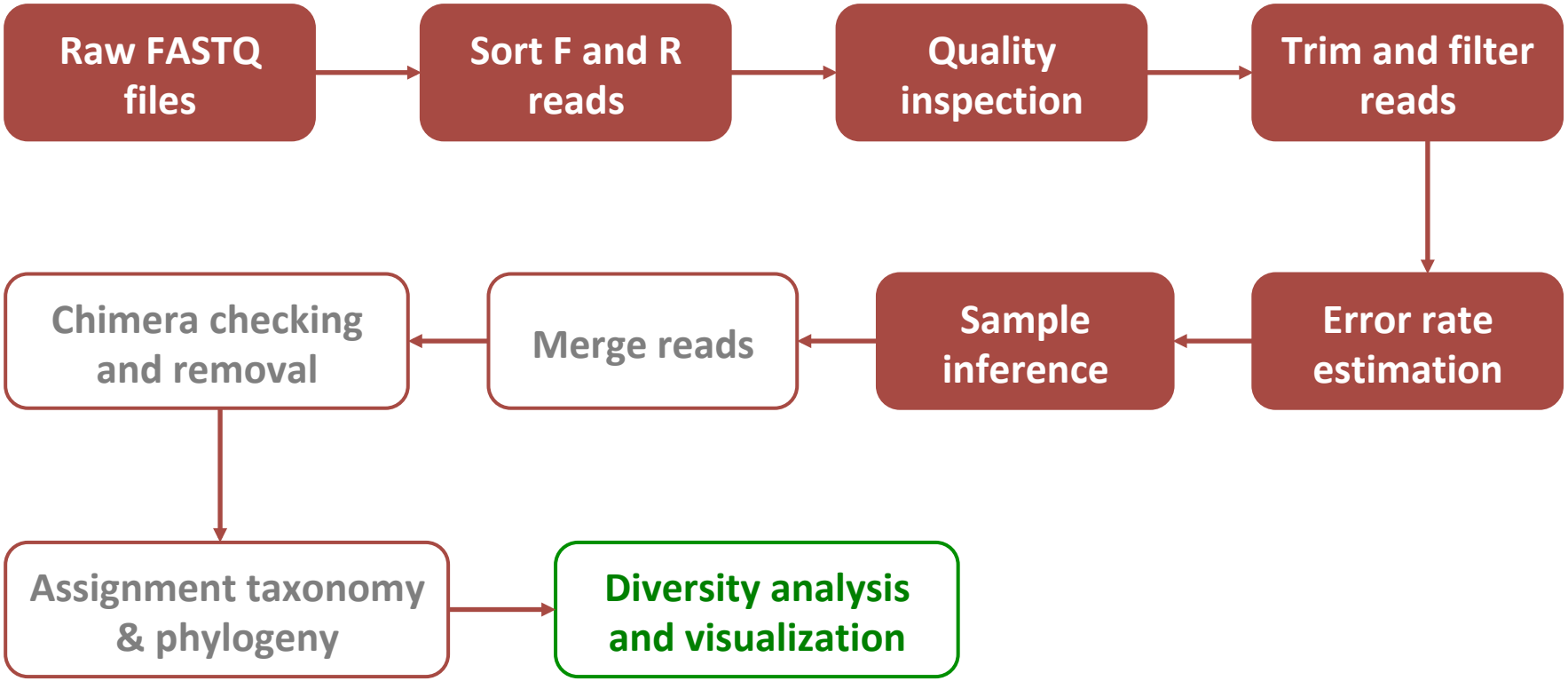
Outline

- Quality Control
- DADA2 background
- **DADA2 workflow**

DADA2 workflow



DADA2 workflow



DADA2 workflow

Before running the pipeline:

- Barcodes, adapters should be removed
  - Cutadapt, Trimmomatic, …
- Samples should be demultiplexed
  - FASTX-Toolkit, idemp, …
- For paired-end data, forward and reverse reads must be in the same order.

DADA2 workflow

The data:

- The data can be accessed here.
- Stool samples.
- Paired-end 300 bp reads.
- Barcodes/Adapters have been removed.

Sample	Dog	Treatment
Dog1	B	2
Dog2	G	3
Dog3	K	3
Dog8	B	4
Dog9	G	0
Dog10	K	4
Dog15	B	1
Dog16	G	4
Dog17	K	0
Dog22	B	3
Dog23	G	1
Dog24	K	2
Dog29	B	0
Dog30	G	2
Dog31	K	1

DADA2 workflow

H3ABioNet

Getting Ready

Load the dada2 package in your R/RStudio

```
library(dada2); packageVersion("dada2")
```

If you do not already have it, see the dada2 installation instructions

DADA2 workflow

H3ABioNet

Getting Ready

Set the path, it points to the **dog samples** directory:

```
MY_HOME <- Sys.getenv("HOME")
data <- paste(MY_HOME, "/dada2_tutorial_dog/dog_samples", sep='')  # change the path
list.files(data)
```

```
##  [1] "Dog1_R1.fastq"  "Dog1_R2.fastq"  "Dog10_R1.fastq" "Dog10_R2.fastq"
##  [5] "Dog15_R1.fastq" "Dog15_R2.fastq" "Dog16_R1.fastq" "Dog16_R2.fastq"
##  [9] "Dog17_R1.fastq" "Dog17_R2.fastq" "Dog2_R1.fastq"  "Dog2_R2.fastq"
## [13] "Dog22_R1.fastq" "Dog22_R2.fastq" "Dog23_R1.fastq" "Dog23_R2.fastq"
## [17] "Dog24_R1.fastq" "Dog24_R2.fastq" "Dog29_R1.fastq" "Dog29_R2.fastq"
## [21] "Dog3_R1.fastq"  "Dog3_R2.fastq"  "Dog30_R1.fastq" "Dog30_R2.fastq"
## [25] "Dog31_R1.fastq" "Dog31_R2.fastq" "Dog8_R1.fastq"  "Dog8_R2.fastq"
## [29] "Dog9_R1.fastq"  "Dog9_R2.fastq"
```

DADA2 workflow

H3ABioNet

Getting Ready

Sort the forward and reverse reads

```
# Forward and reverse fastq filenames have format: SAMPLENAME_R1.fastq and SAMPLENAME_R2.fastq
dataF <- sort(list.files(data, pattern="_R1.fastq", full.names = TRUE))
dataR <- sort(list.files(data, pattern="_R2.fastq", full.names = TRUE))
```

Extract sample names

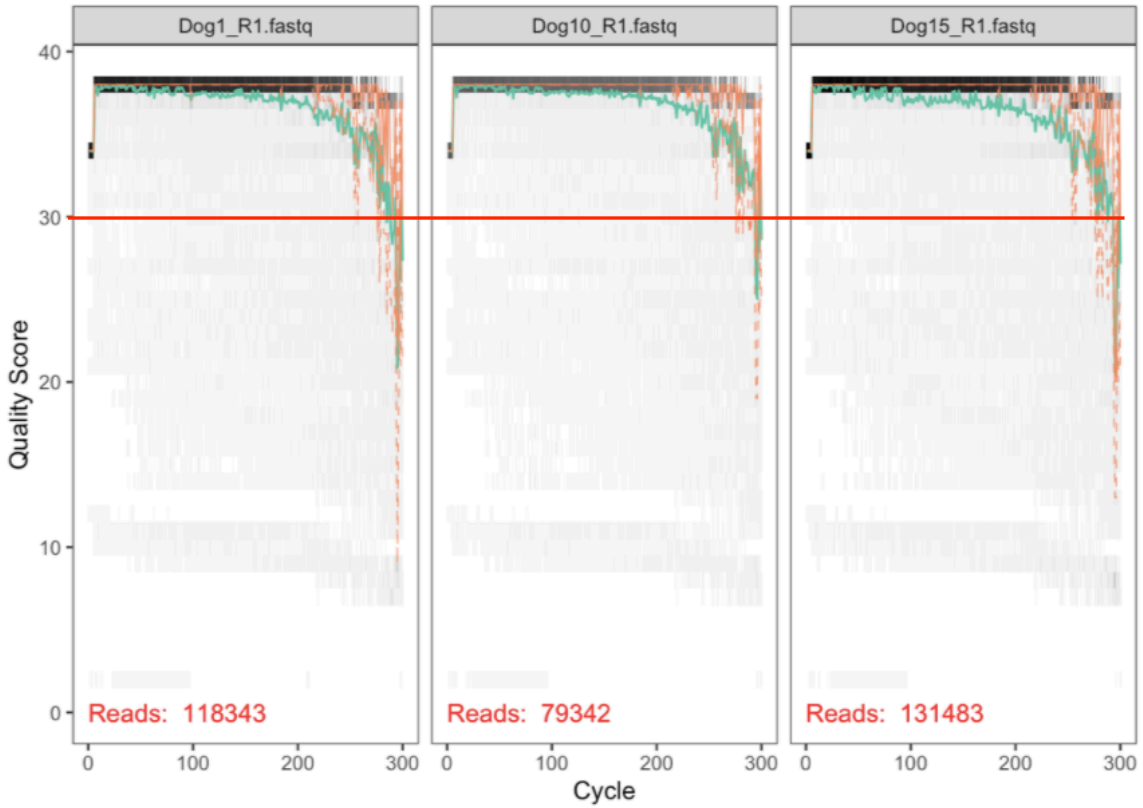
```
# Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq
list.sample.names <- sapply(strsplit(basename(dataF), "_"), `[`, 1)
list.sample.names
```

```
##  [1] "Dog1"  "Dog10" "Dog15" "Dog16" "Dog17" "Dog2"  "Dog22" "Dog23"
##  [9] "Dog24" "Dog29" "Dog3"  "Dog30" "Dog31" "Dog8"  "Dog9"
```

DADA2 workflow

Quality Control

```
plotQualityProfile(dataF[1:3])
```



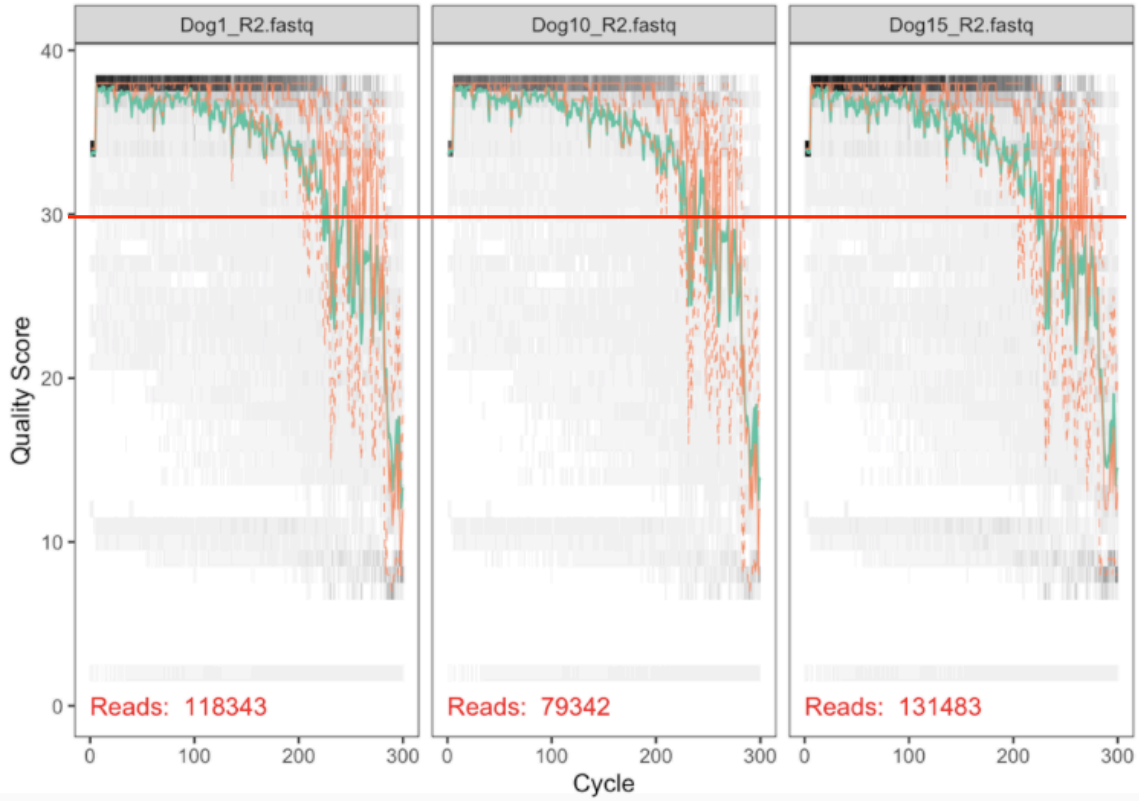
- The quality plot of three forward samples.

- Scores never really go below 30.

DADA2 workflow

Quality Control

```
plotQualityProfile(dataR[1:3])
```



- The reverse reads are slightly different.

- The scores are good but they drop off right around 275 bp.

DADA2 workflow

H3ABioNet

Filter and Trim

Set filtered subdirectory and rename files

```
# Place filtered files in filtered/ subdirectory
filt.dataF <- file.path(data, "filtered", paste0(list.sample.names, "_F_filt.fastq.gz"))
filt.dataR <- file.path(data, "filtered", paste0(list.sample.names, "_R_filt.fastq.gz"))
names(filt.dataF) <- list.sample.names
names(filt.dataR) <- list.sample.names
```

DADA2 workflow

Filter and Trim

```
out <- filterAndTrim(dataF, filt.dataF, dataR, filt.dataR, truncLen=c(290,275),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE) # On Windows set multithread=FALSE
```

truncLen truncates your reads at specific base.

truncLen=c(290,275)

The amplicon length.

The length of your overlap, by default is 20 for DADA2.

DADA2 workflow

Filter and Trim

```
out <- filterAndTrim(dataF, filt.dataF, dataR, filt.dataR, truncLen=c(290,275),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE) # On Windows set multithread=FALSE
```

maxN maximum number of ambiguous nucleotides.

maxN=0

DADA2 requires no Ns.

DADA2 workflow

Filter and Trim

```
out <- filterAndTrim(dataF, filt.dataF, dataR, filt.dataR, truncLen=c(290,275),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE) # On Windows set multithread=FALSE
```

maxEE maximum number of estimated errors allowed in your reads.

maxEE=c(2,2)

The quality of your sequences.

DADA2 workflow

Filter and Trim

```
out <- filterAndTrim(dataF, filt.dataF, dataR, filt.dataR, truncLen=c(290,275),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE) # On Windows set multithread=FALSE
```

truncQ truncates the read at the first nucleotide with a specific quality score.

truncQ=2

Score of 2 means that the probability of the base being incorrect is 63%.

DADA2 workflow

Filter and Trim

```
out <- filterAndTrim(dataF, filt.dataF, dataR, filt.dataR, truncLen=c(290,275),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE) # On Windows set multithread=FALSE
```

rm.phix removes reads that match against the phiX genome.

rm.phix=TRUE

DADA2 workflow

Filter and Trim

```
out <- filterAndTrim(dataF, filt.dataF, dataR, filt.dataR, truncLen=c(290,275),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE) # On Windows set multithread=FALSE
```

Compress if you want to fastq files to be gzipped.

Multithread if you want your files to run in parallel.

DADA2 workflow

H3ABioNet

Learn the Error Rates

It will create an error model that will be used by the DADA2 algorithm.

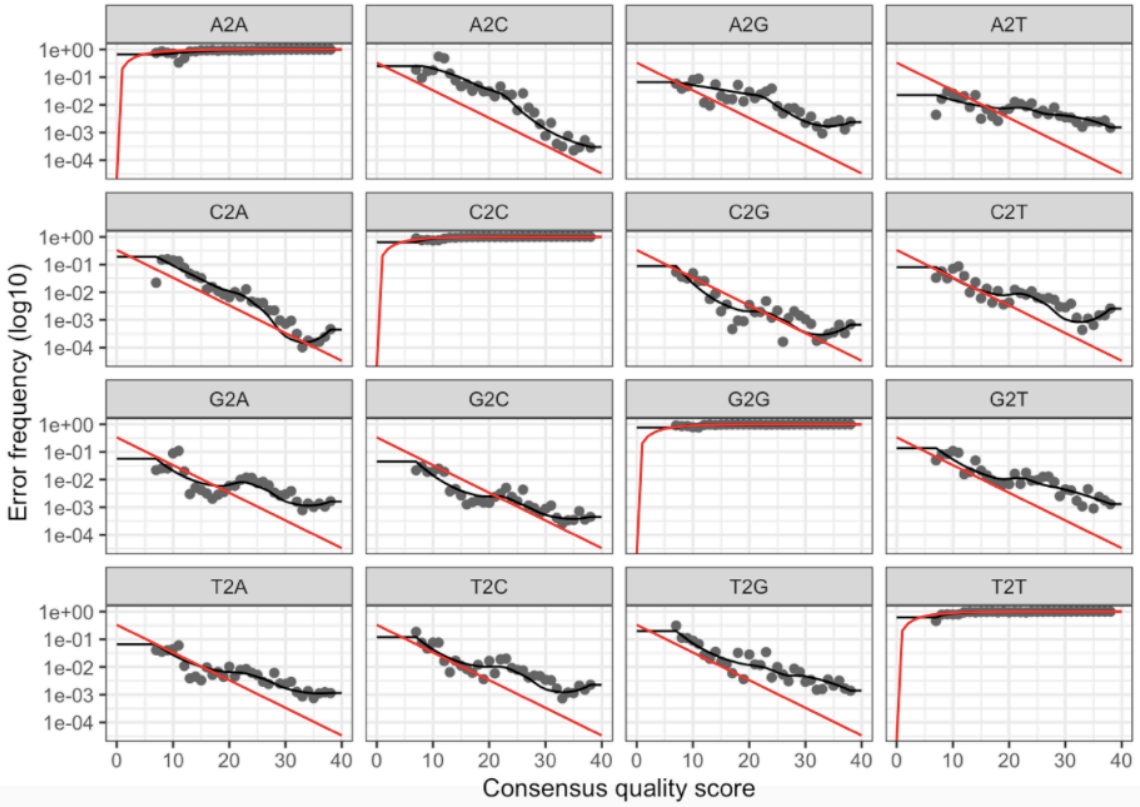
```
errF <- learnErrors(filt.dataF, multithread=TRUE)
```

```
errR <- learnErrors(filt.dataR, multithread=TRUE)
```

DADA2 workflow

Learn the Error Rates

```
plotErrors(errF, nominalQ=TRUE)
```



- The error rates for each possible transition (A -> C).

- As quality score increases, the expected error rate decreases.

DADA2 workflow

Sample Inference

Set filtered subdirectory and rename files

```
# Place filtered files in filtered/ subdirectory
filt.dataF <- file.path(data, "filtered", paste0(list.sample.names, "_F_filt.fastq.gz"))
filt.dataR <- file.path(data, "filtered", paste0(list.sample.names, "_R_filt.fastq.gz"))
names(filt.dataF) <- list.sample.names
names(filt.dataR) <- list.sample.names
```

It uses the error model that was created earlier.

p-value high -> sequence likely caused by errors.
p-value low -> sequence is real.