H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S analysis pipeline

QC and ASV picking using the dada2 pipeline

Module 5: 16S rRNA Analysis Pipeline

- Session 1:

QC and ASV picking using the dada2 pipeline

- Session 2:

Taxonomic classification and alignment using the dada2 pipeline

Learning Objectives

- To give an overview on quality control process
- To give a background on dada2
- To give a general understanding of the 16S rRNA analysis pipeline using dada2
- To complete the dada2 workflow

Learning Outcomes

- To understand each step of 16S rRNA analysis pipeline using dada2
- To run 16S rRNA analysis pipeline from raw reads
- To know how to edit files and use command line
- To choose the best approach and tools to analyze your data

H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S analysis pipeline

QC and ASV picking using the dada2 pipeline

Outline

- Quality Control
- DADA2 background
- DADA2 workflow

Quality Control

- Before analyzing generated sequence to draw biological conclusions, a quality control check should be performed to make sure there is no biases in the data.

- QC gives a quick impression of whether your data has any problems of which you should be aware before doing any analysis.

Quality Control

Potential problems:

- Low confidence bases (Ns)
- Sequence specific bias
- Sequence contamination
- Adapters
- …

Quality Control

Software packages for QC:

- FastQC
- MultiQC
- FastX-Toolkit
- PRINSEQ
- TagCleaner
- NGS QC Tool-Kit
- …

Quality Control

H3ABioNet

FASTQ format

What is a FastQ file?

FASTQ= FASTA + Quality

FastQ format is a text-based format for storing both a biological sequence and its corresponding quality scores.

Quality Control

FASTQ format

- Each FastQ file contains hundreds of millions of rows.
- Each block of 4 lines, starting with " @" represents a read.

Line 1 begins with a '@' character and is followed by a sequence identifier and an optional description (like a FASTA title line).

Line 2 is the raw sequence letters (ATCG).

Line 3 begins with a '+' character and is optionally followed by the same sequence identifier (and any description).

Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

Quality Control

FASTQ format

A FastQ file containing a single sequence might look like this:

```
@read name
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+ read name
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

The character '!' represents the lowest quality while '~' is the highest.

Quality Control

H3ABioNet

Quality measurements

Base-calling error probabilities are reported by sequencers.
Usually in Phred (quality) score.
Usually coded by ASCII characters

Phred score

$$Q = -10log_{10}P$$

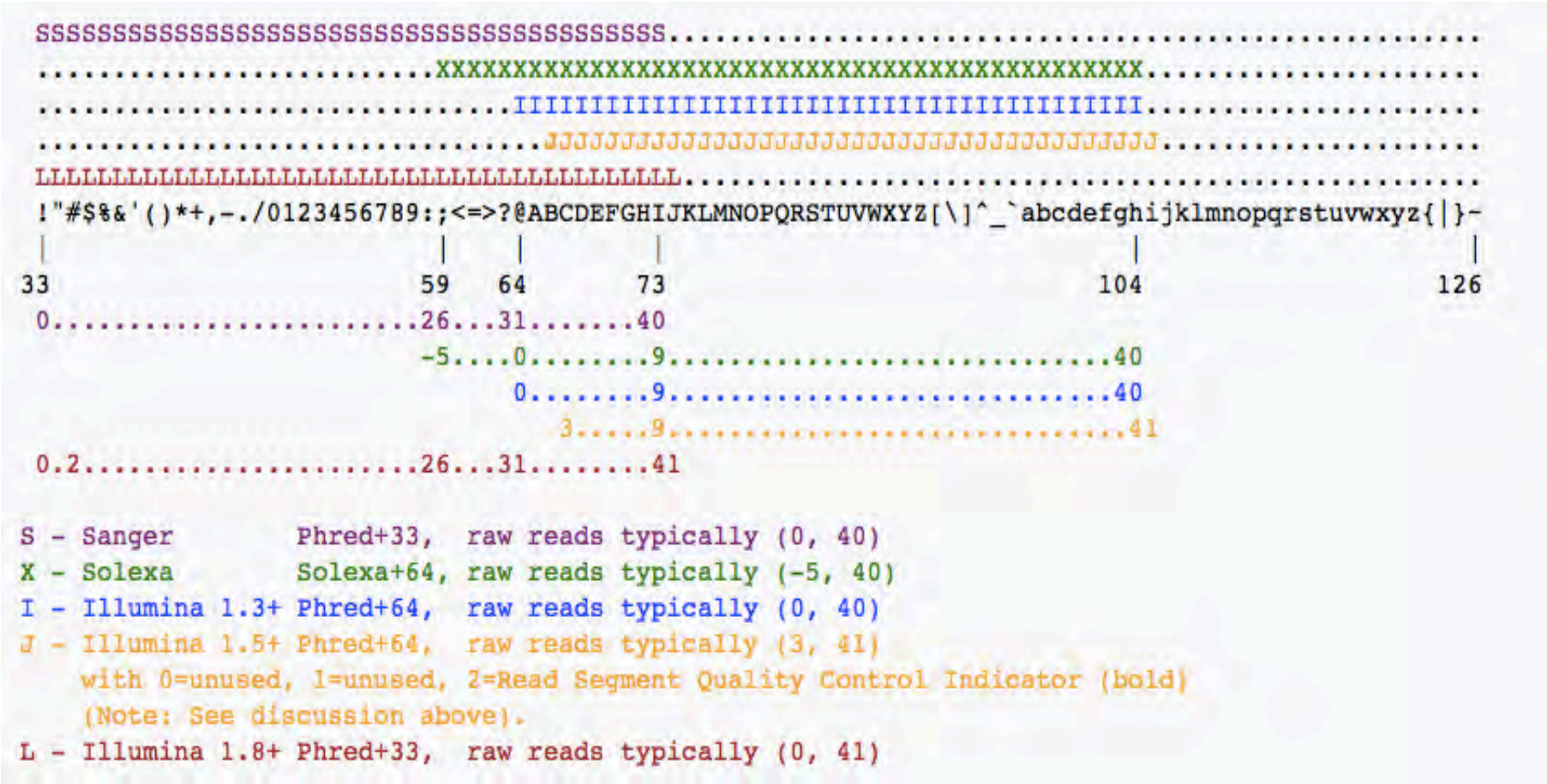
If the quality of a base is 20, the probability that it is wrong is 0.01

T	C	A	G	T	A	C	T	C	G
40	40	40	40	40	40	40	40	37	35

Quality Control

H3ABioNet

Quality measurements



Quality Control

What is FastQC?

FastQC aims to provide a QC report which can spot problems which originate either in the sequencer or in the library material.

Quality Control

FastQC reports

Quality Control

FastQC reports

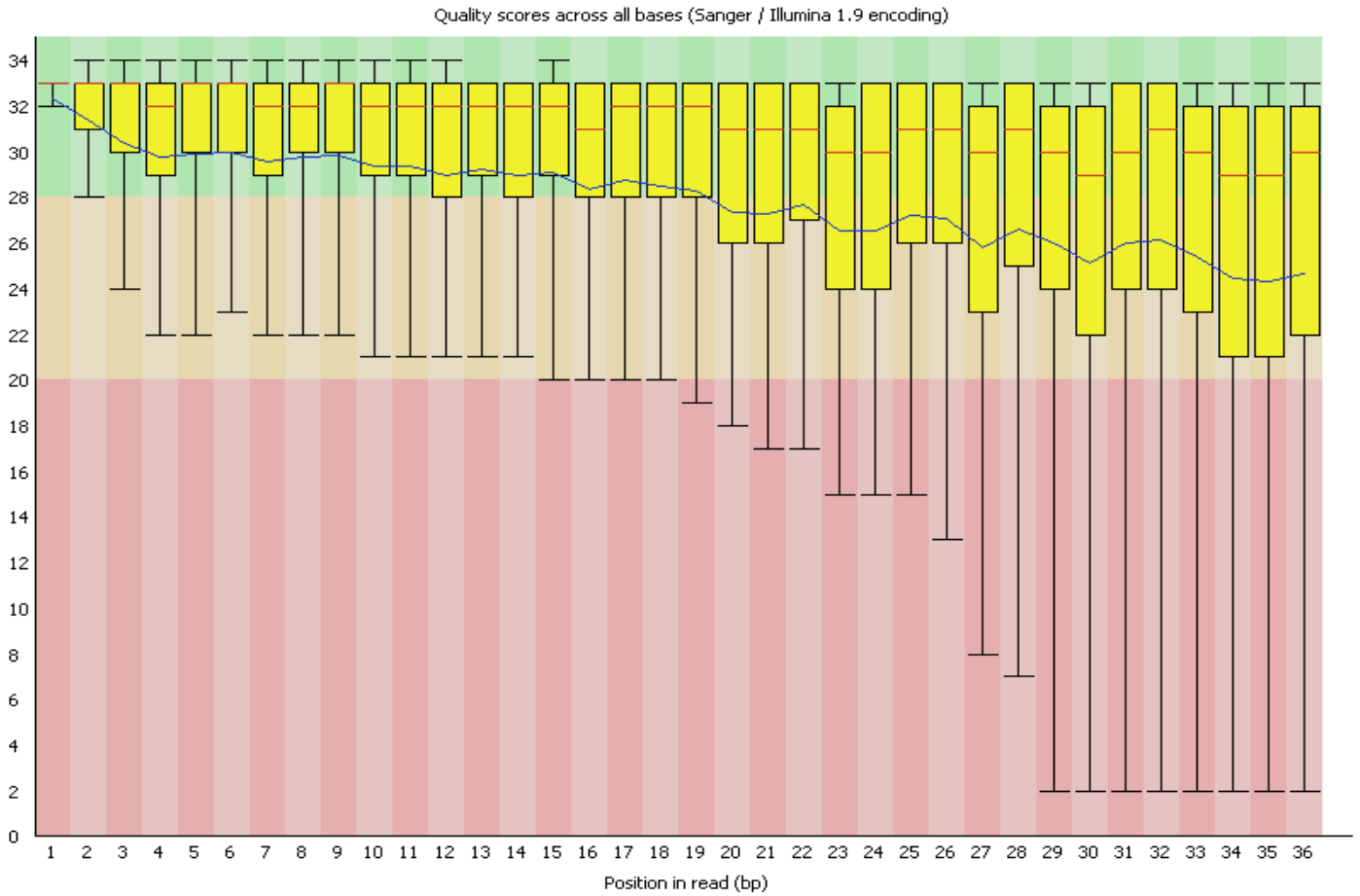
Basic Statistics

Measure	Value
Filename	good_sequence_short.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	40
%GC	45

Quality Control

FastQC reports

Per Base Sequence Quality

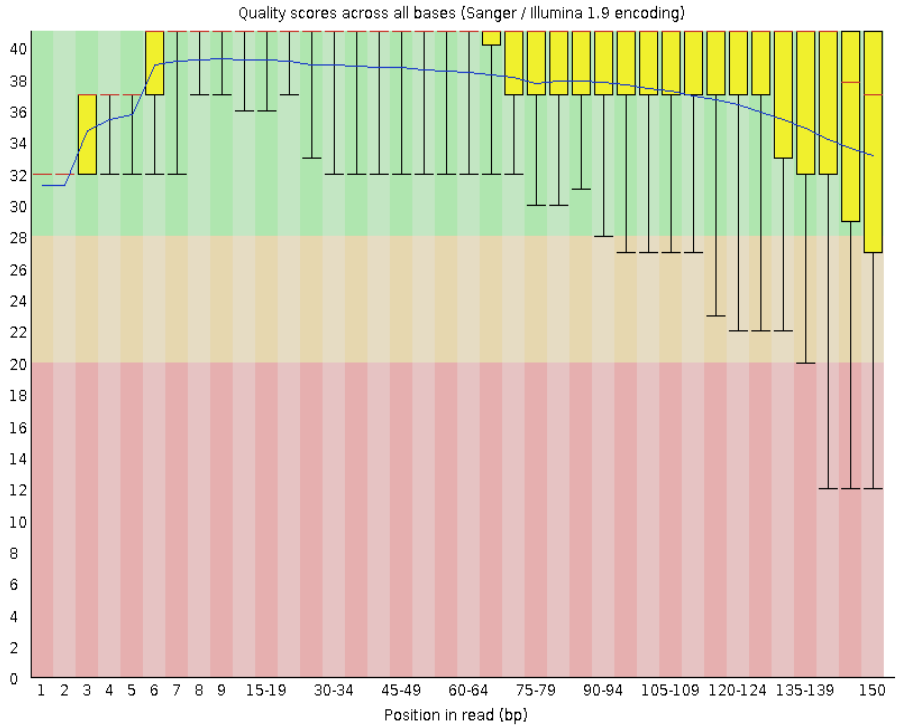


Quality Control

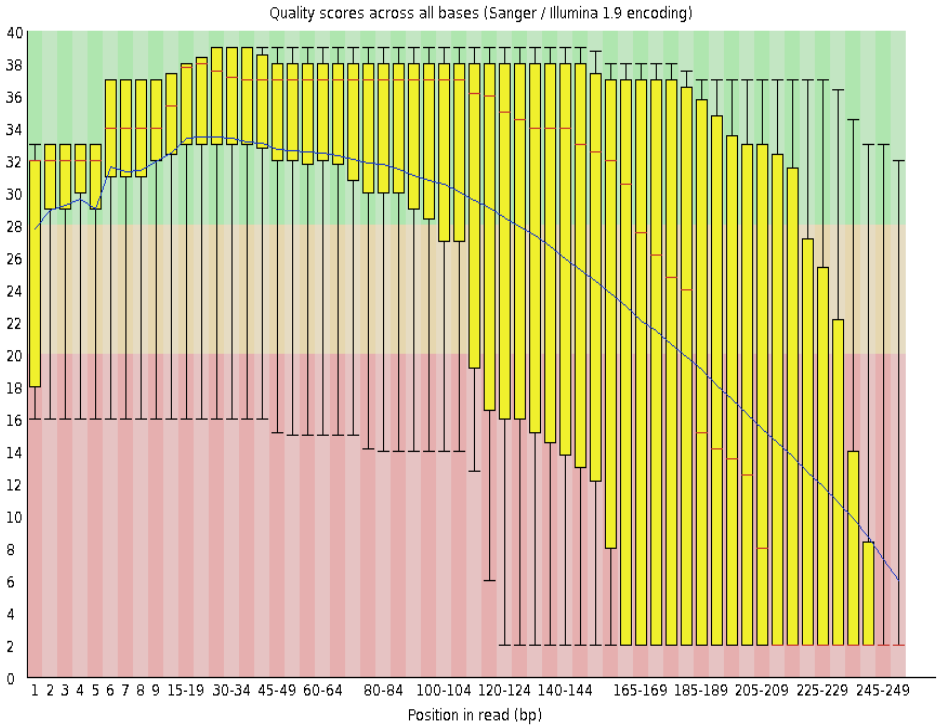
FastQC reports

Per Base Sequence Quality

Good quality FastQC report:



Bad quality FastQC report

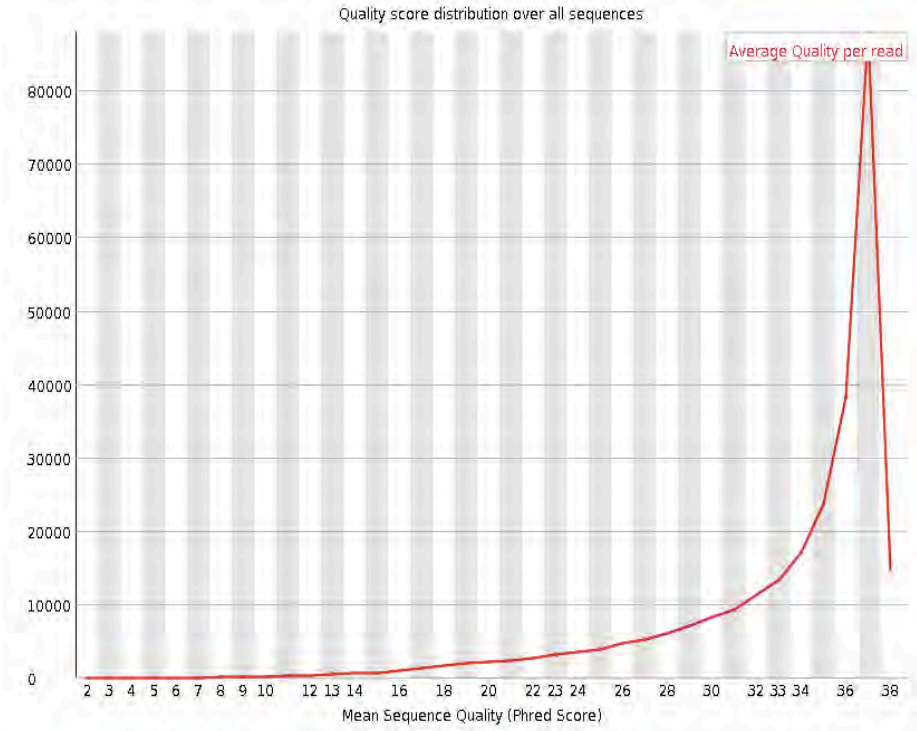


Quality Control

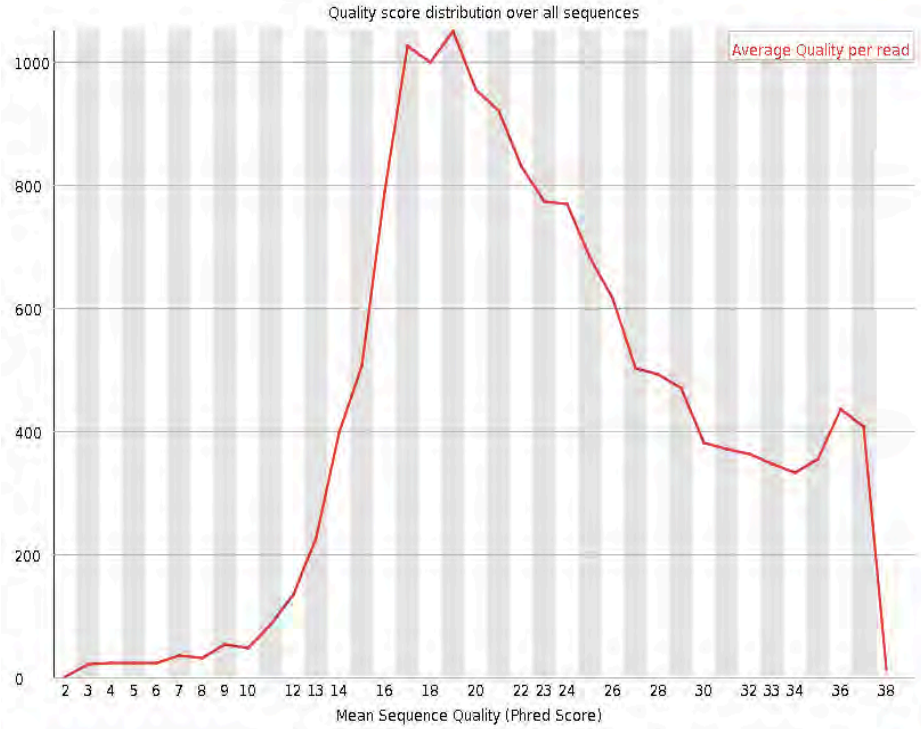
FastQC reports

Per Sequence Quality Scores

Good quality FastQC report:



Bad quality FastQC report

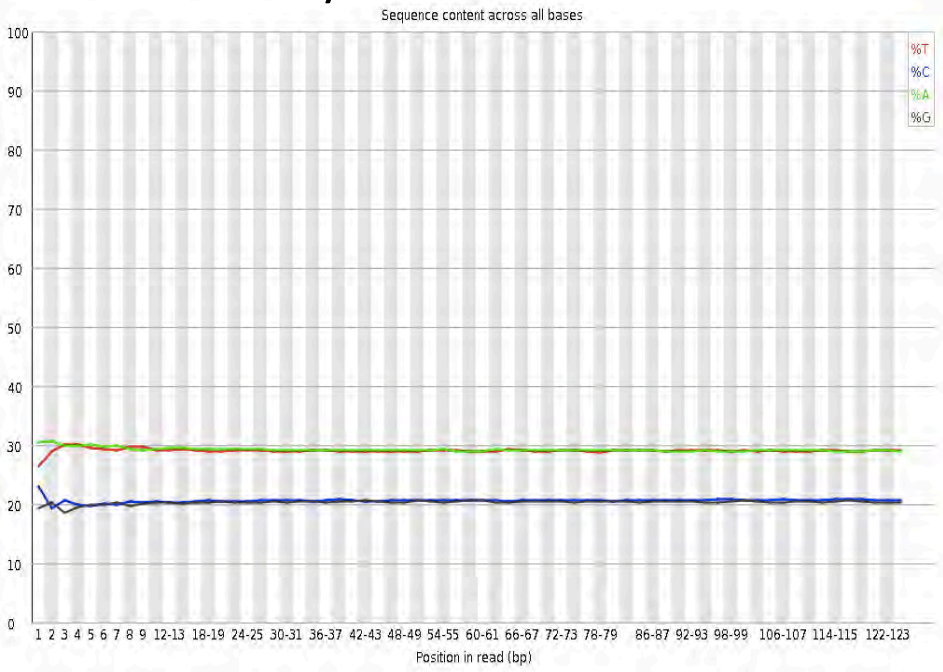


Quality Control

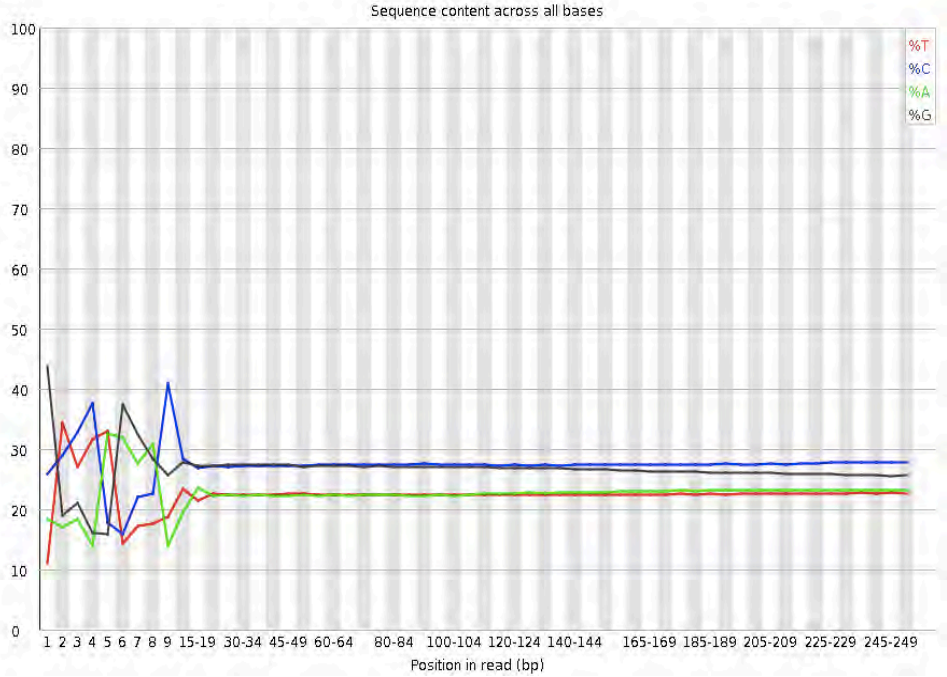
FastQC reports

Per Base Sequence Content

DNA library



RNA library



H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S analysis pipeline

QC and ASV picking using the dada2 pipeline

Outline

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

DADA2 background

Divisive Amplicon Denoising Algorithm 2 (DADA2) is an open source algorithm implemented in R, which uses a statistical inference to correct amplicon errors.

It intends to simplify the study of microbial communities by allowing to reconstruct amplicon-sequenced communities at the highest resolution.

Callahan et al., 2016

DADA2 background

DADA2 implements a complete workflow that takes raw amplicon sequencing data in fastq files as input.

It produces an error-corrected table of the abundances of amplicon sequence variants in each sample (ASV table).

Callahan et al., 2016

DADA2 background

Benefits to DADA2:

- Compatible with all amplicon types

  16S, 18S, ITS,…

- Works on different next generation sequencing platforms

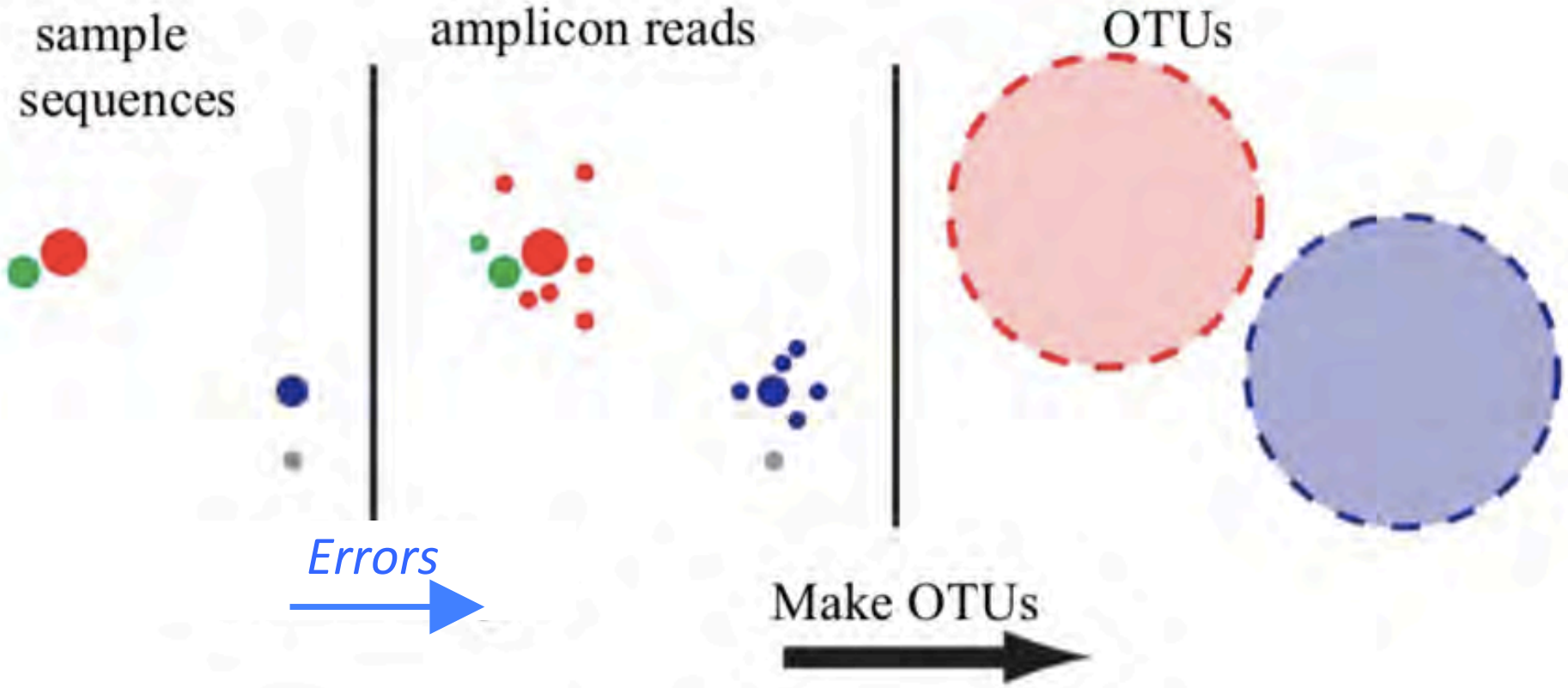
  Illumina, Ion Torrent, 454 pyrosequencing

- Provides single-nucleotide resolution
- Lower false-positive rate

DADA2 background

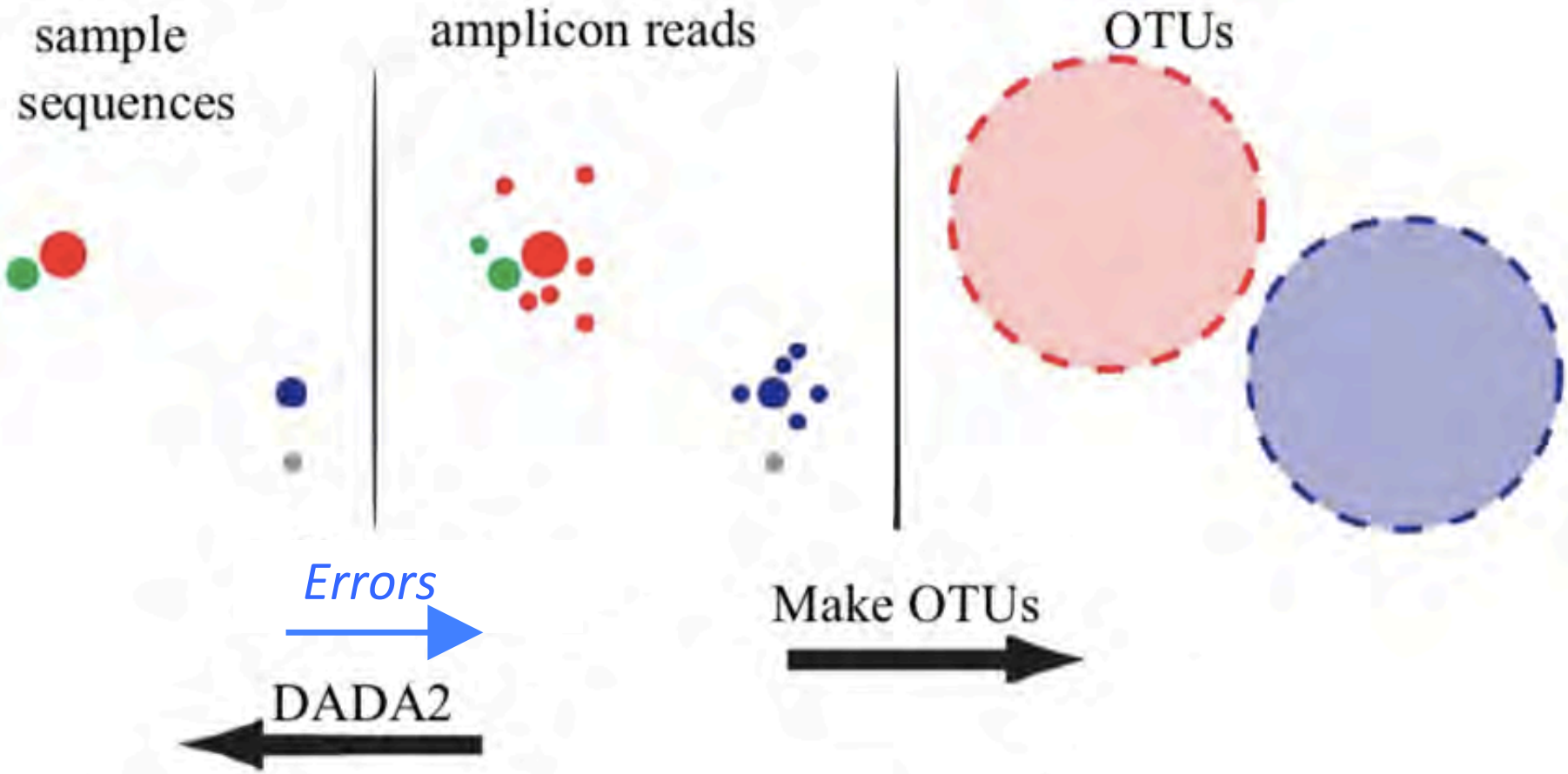
- R or RStudio.
- QIIME2
  - Simplified and condensed the dada2 workflow

DADA2 background



Benjamin Callahan

DADA2 background



Benjamin Callahan

16SrRNA Intermediate Bioinformatics Online Course: Int_BT_2019
Imane Allali

H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

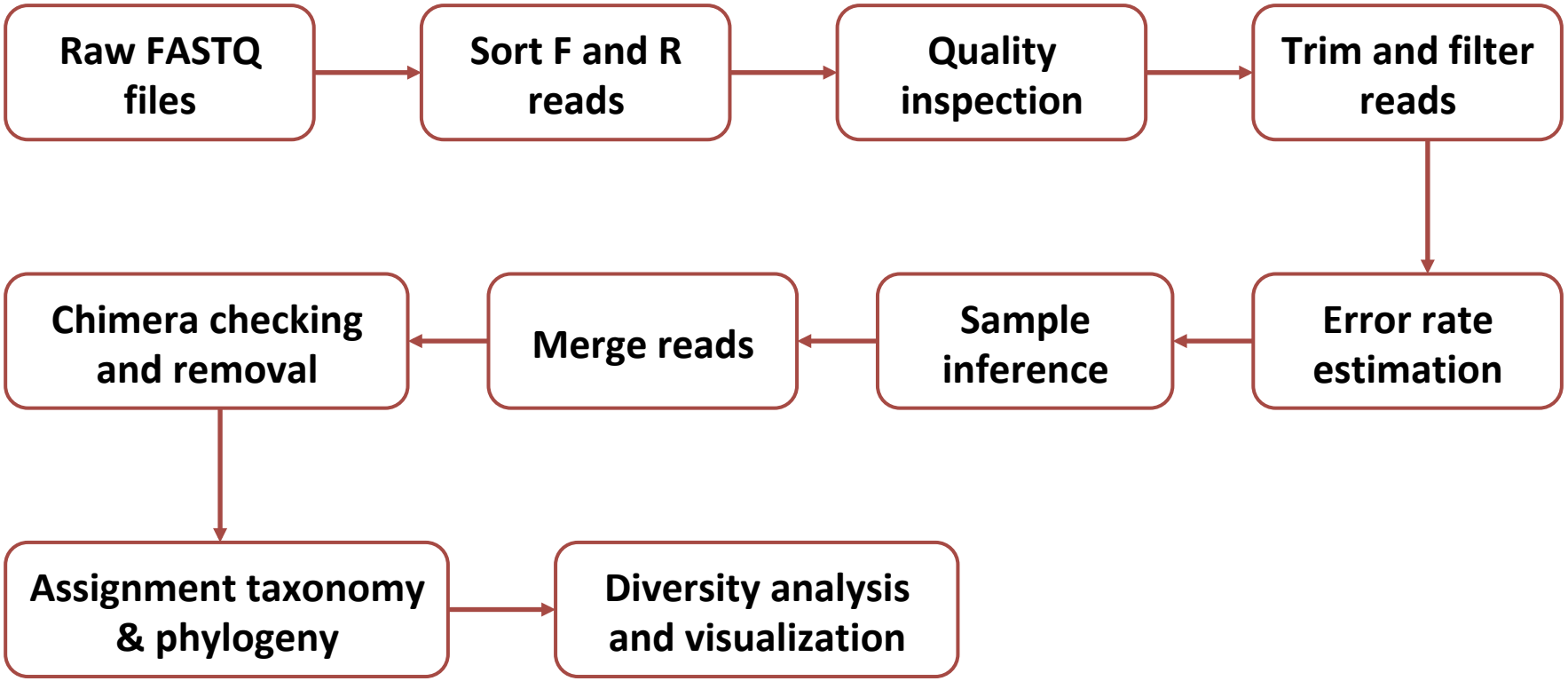
16S analysis pipeline

QC and ASV picking using the dada2 pipeline

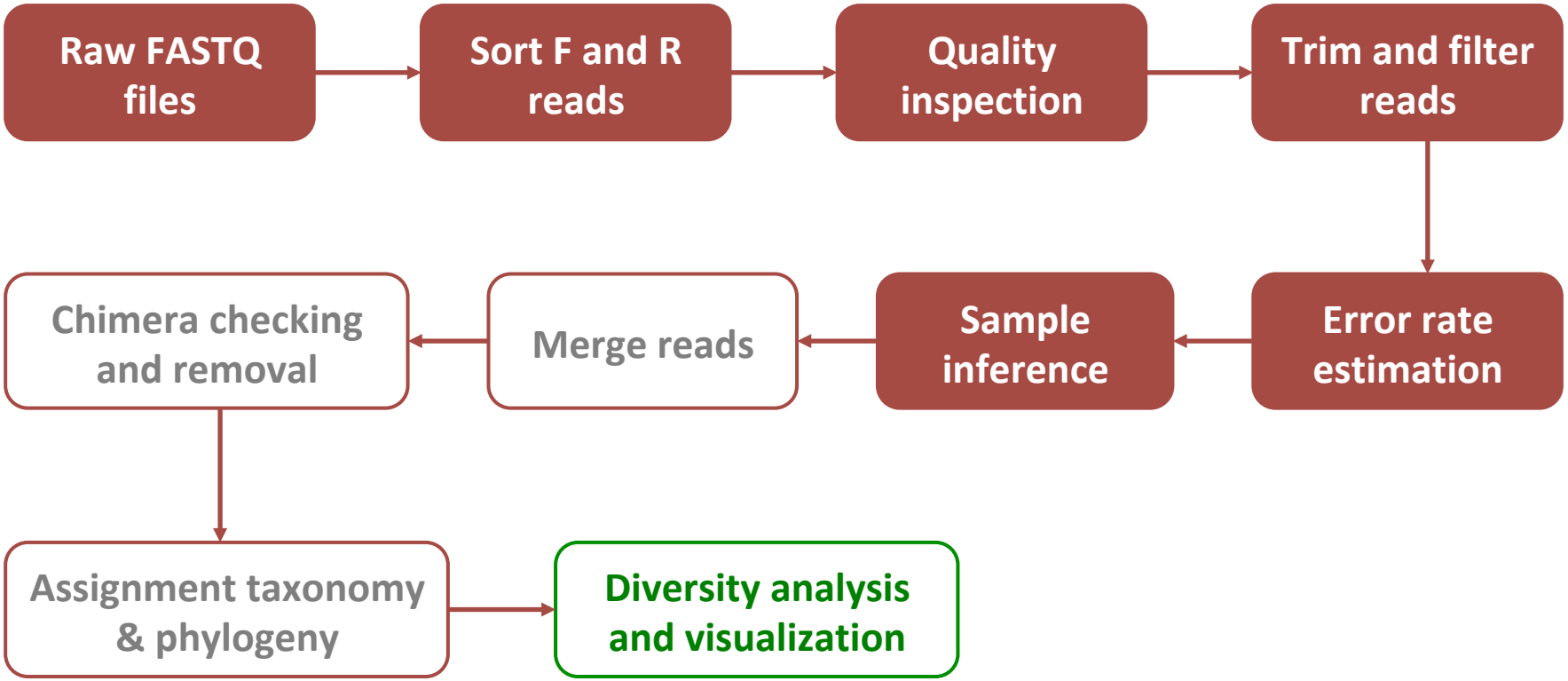
Outline

- 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

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

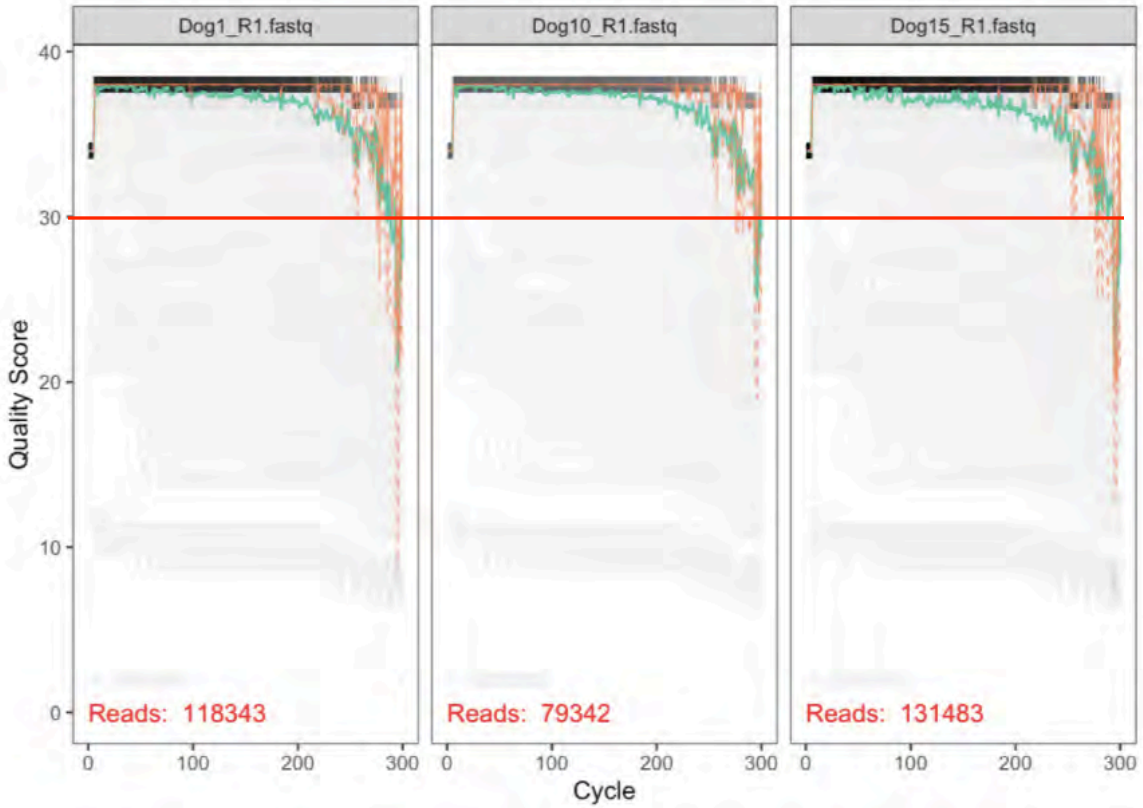
```
# Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq
list.sample.names <- sapply(strsplit(basename(dataF), "_"), `[`, 1)
list.sample.names
```

```
##  [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])
```



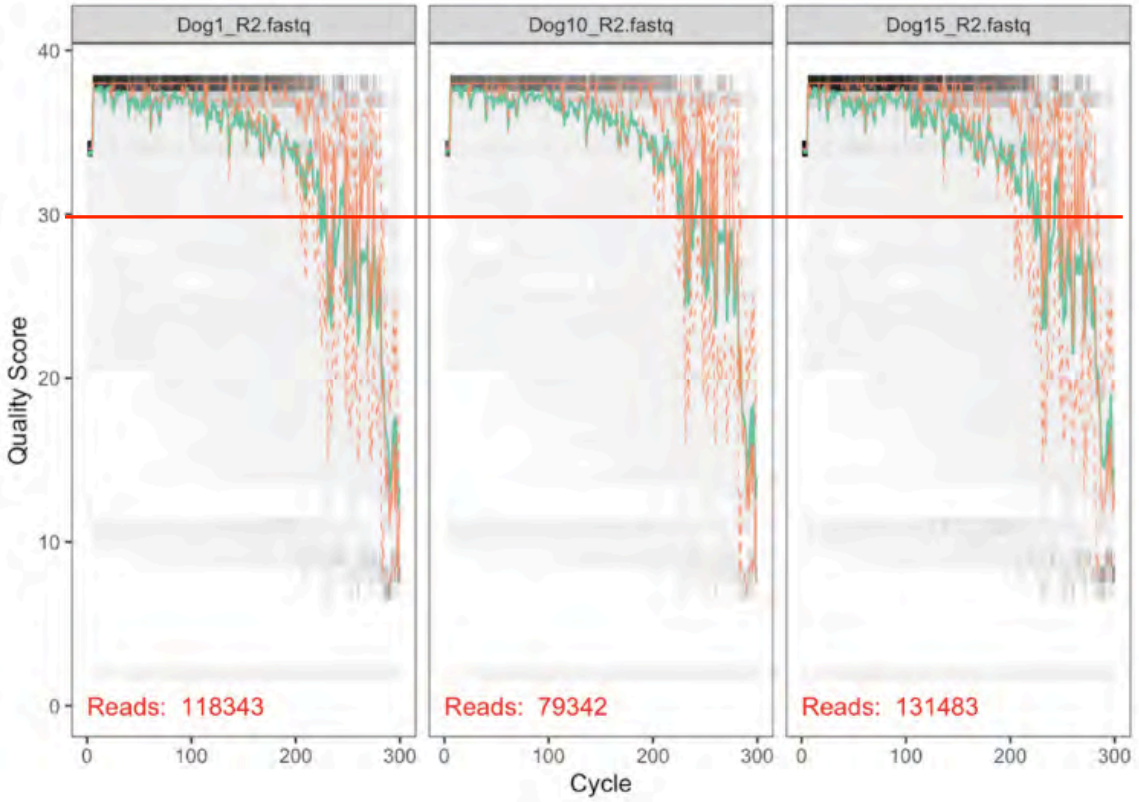
- The quality plot of three forward samples.

- 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

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

Learn the Error Rates

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

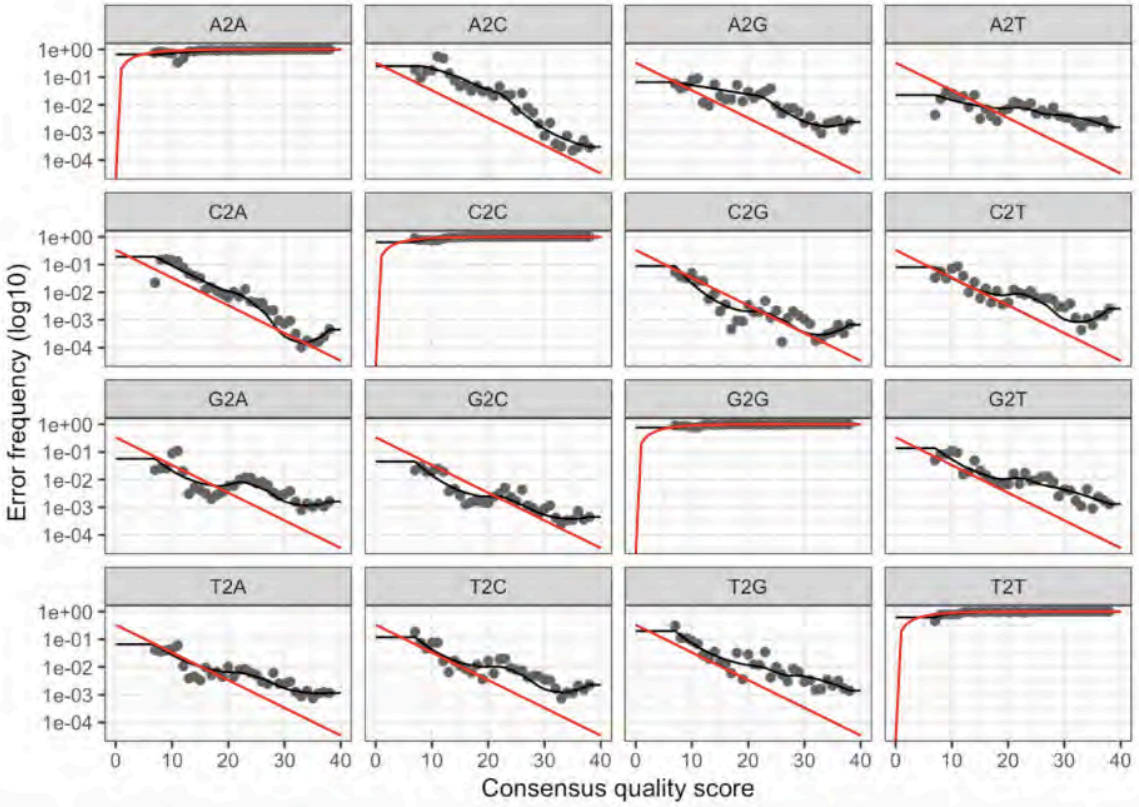
```
errF <- learnErrors(filt.dataF, multithread=TRUE)
```

```
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.

H3ABioNet
Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S analysis pipeline

QC and ASV picking using the dada2 pipeline

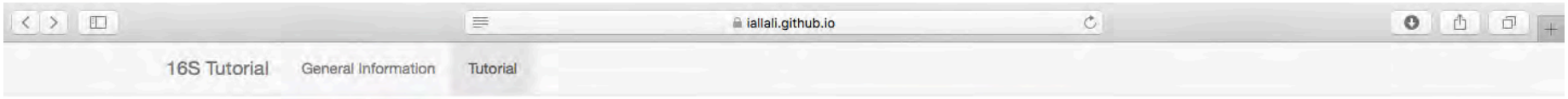
Practical

Practical

The practical is available here:

https://iallali.github.io/DADA2_pipeline/16SrRNA_DADA2_pipeline.html

Practical



Getting Ready

First, we load the dada2 package on your RStudio. if you do not already have it, see the dada2 installation instructions.

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

```
## [1] '1.13.1'
```

We set the path so that it points to the extracted directory of the dataset named "dog_samples" on your computer or cluster:

```
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"  "filtered"
```

If your listed files match those here, you can start running the DADA2 pipeline.

Now, we read in the names of the fastq files and we sort them by forward and reverse. Then, we perform some string manipulation to extract a list of the sample names.

```
# 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, 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"
```

H3ABioNet
Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S rRNA analysis pipeline

Taxonomic classification and alignment using the dada2 pipeline

Module 5: 16S rRNA Analysis Pipeline

- Session 1:

QC and ASV picking using the dada2 pipeline

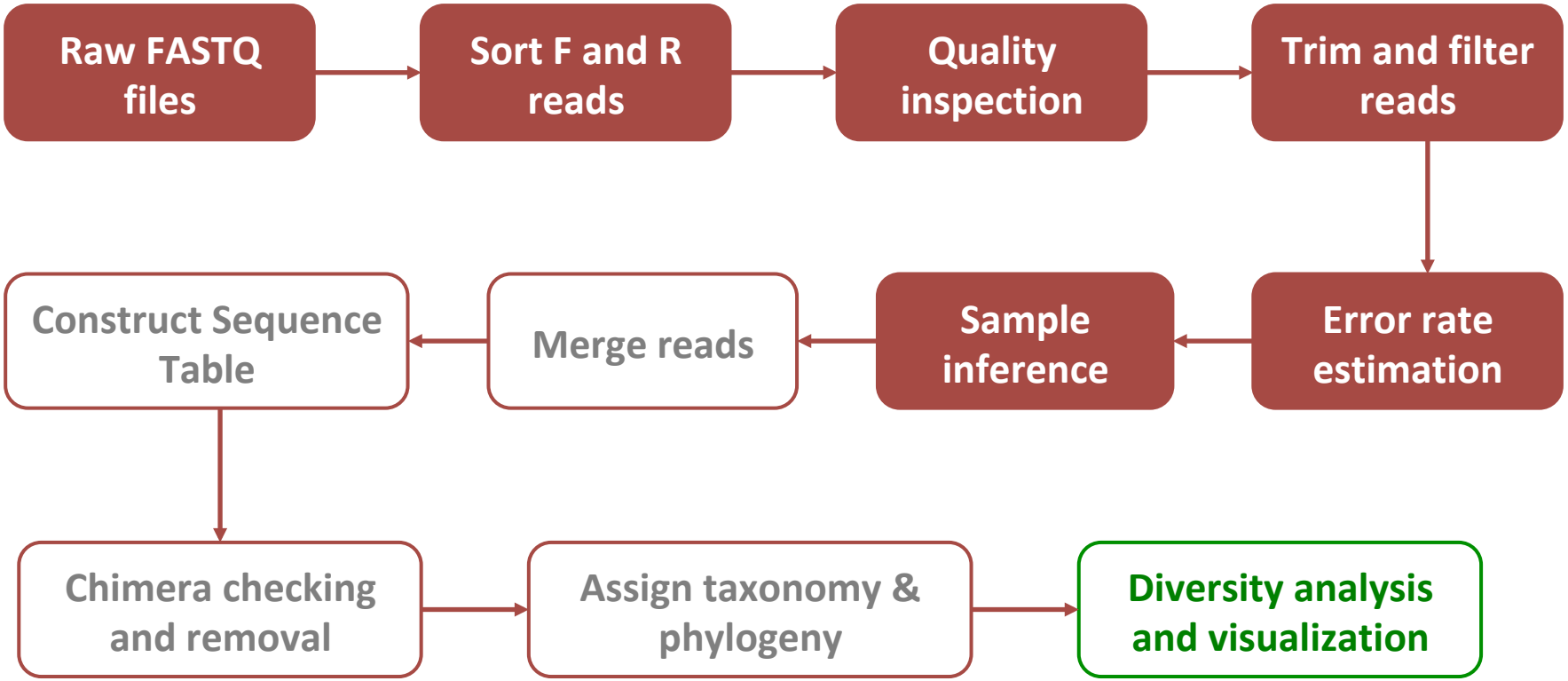
- Session 2:

Taxonomic classification and alignment using the dada2 pipeline

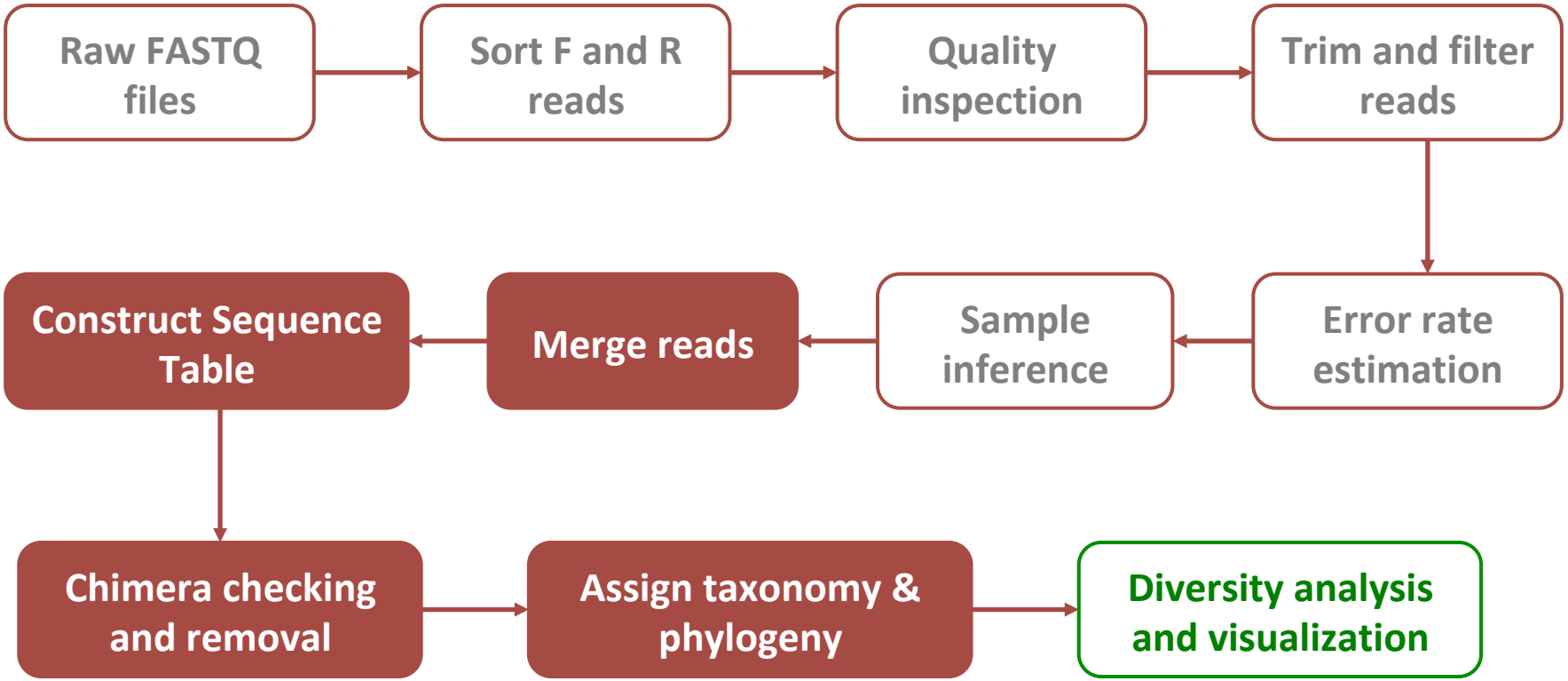
Outline

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

DADA2 workflow



DADA2 workflow



DADA2 workflow

Merge Reads

```
merge.reads <- mergePairs(dadaF, filt.dataF, dadaR, filt.dataR, verbose=TRUE)
```

```
## 76306 paired-reads (in 1153 unique pairings) successfully merged out of 82378 (in 2526 pairings) input.
## 54948 paired-reads (in 720 unique pairings) successfully merged out of 60139 (in 1811 pairings) input.
## 83013 paired-reads (in 843 unique pairings) successfully merged out of 89310 (in 2175 pairings) input.
## 74533 paired-reads (in 1093 unique pairings) successfully merged out of 81135 (in 2674 pairings) input.
## 67364 paired-reads (in 855 unique pairings) successfully merged out of 71053 (in 1743 pairings) input.
## 69262 paired-reads (in 1194 unique pairings) successfully merged out of 76778 (in 2891 pairings) input.
```

mergePairs merges reads only if they exactly overlap.

The length of your overlap, by default is 20 nt for DADA2, you can lower it by using this parameter **minOverlap**.

DADA2 workflow

Merge Reads

```
head(merge.reads[[1]])
```

```
##
sequence
## 1  TGTGTGCCAGCAGCCGCGGTAATACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGC
TCAACCTGGGAACTGCATCTGATACTGGCAAGCTTGAGTCTCGTAGAGGGGGGTAGAATTCCAGGTGTAGCGGTGAAATGCGTAGAGATCTGGAGGAATACCGGTGGCGAAGG
CGGCCCCCTGGACGAAGACTGACGCTCAGGTGCGAAAGCGTGGGGAGCAAACAGGATTAGAAACCCGAGTAGTCCGGCTGACTGACTCGAGA
## 2  TGTGTGCCAGCAGCCGCGGTAATACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGC
TCAACCTGGGAACTGCATCTGATACTGGCAAGCTTGAGTCTCGTAGAGGGGGGTAGAATTCCAGGTGTAGCGGTGAAATGCGTAGAGATCTGGAGGAATACCGGTGGCGAAGG
CGGCCCCCTGGACGAAGACTGACGCTCAGGTGCGAAAGCGTGGGGAGCAAACAGGATTAGAAACCCTAGTAGTCCGGCTGACTGACTCGAGA
## 3  TGTGTGCCAGCAGCCGCGGTAATACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGC
TCAACCTGGGAACTGCATCTGATACTGGCAAGCTTGAGTCTCGTAGAGGGGGGTAGAATTCCAGGTGTAGCGGTGAAATGCGTAGAGATCTGGAGGAATACCGGTGGCGAAGG
CGGCCCCCTGGACGAAGACTGACGCTCAGGTGCGAAAGCGTGGGGAGCAAACAGGATTAGAAACCCCAGTAGTCCGGCTGACTGACTCGAGA
## 4 TTGTGTGCCAGCAGCCGCGGTAATACGTAGGGGGCTAGCGTTATCCGGATTTACTGGGCGTAAAGGGTGCGTAGGCGGTCTTTCAAGTCAGGAGTTAAAGGCTACGGC
TCAACCGTAGTAAGCTCCTGATACTGTCTGACTTGAGTGCAGGAGAGGAAAGCGGAATTCCCAGTGTAGCGGTGAAATGCGTAGATATTGGGAGGAACACCAGTAGCGAAGGC
GGCTTTCTGGACTGTAACTGACGCTGAGGCACGAAAGCGTGGGGAGCAAACAGGATTAGAAACCCTAGTAGTCCGGCTGACTGACTCGAGAG
## 5  TGTGTGCCAGCAGCCGCGGTAATACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGC
TCAACCTGGGAACTGCATCTGATACTGGCAAGCTTGAGTCTCGTAGAGGGGGGTAGAATTCCAGGTGTAGCGGTGAAATGCGTAGAGATCTGGAGGAATACCGGTGGCGAAGG
CGGCCCCCTGGACGAAGACTGACGCTCAGGTGCGAAAGCGTGGGGAGCAAACAGGATTAGAAACCCTTGTAGTCCGGCTGACTGACTCGAGA
## 6  TGTGTGCCAGCAGCCGCGGTAATACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTTTGTTAAGTCAGATGTGAAATCCCCGGGC
TCAACCTGGGAACTGCATCTGATACTGGCAAGCTTGAGTCTCGTAGAGGGGGGTAGAATTCCAGGTGTAGCGGTGAAATGCGTAGAGATCTGGAGGAATACCGGTGGCGAAGG
CGGCCCCCTGGACGAAGACTGACGCTCAGGTGCGAAAGCGTGGGGAGCAAACAGGATTAGAAACCCTGGTAGTCCGGCTGACTGACTCGAGA
##   abundance forward reverse nmatch nmismatch nindel prefer accept
## 1       460       2       1    253          0      0      2   TRUE
## 2       456       1       1    253          0      0      2   TRUE
## 3       421       5       1    253          0      0      2   TRUE
## 4       414       7       2    252          0      0      2   TRUE
## 5       401       6       1    253          0      0      2   TRUE
## 6       400       4       1    253          0      0      2   TRUE
```

DADA2 workflow

Construct Amplicon Sequence Variant (ASV) Table

```
seqtab <- makeSequenceTable(merge.reads)
dim(seqtab)
```

```
## [1]    15 13527
```

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

```
##
##  311  312  313  315
##  107 9136 4283    1
```

DADA2 workflow

Construct Amplicon Sequence Variant (ASV) Table

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		TTGTGTGCC	TTGTGTGCC	TGTGTGCCA	TGTGTGCCA	TTGTGTGCC	TTGTGTGCC	TGTGTGCCA	TGTGTGCCA	TTGTGTGCC	TTGTGTGCC	TGTGTGCCA	TTGTGTGCC
2	Dog1	0	0	242	205	0	0	0	223	0	0	195	0
3	Dog10	0	0	0	0	0	0	0	0	0	0	0	0
4	Dog15	0	0	0	0	0	0	0	0	0	0	0	0
5	Dog16	0	0	0	0	0	0	0	0	0	0	0	0
6	Dog17	0	0	0	0	0	0	0	0	0	0	0	0
7	Dog2	373	0	0	0	276	0	0	0	283	277	0	322
8	Dog22	1926	0	0	0	1516	0	0	0	1453	1459	0	1366
9	Dog23	0	955	0	0	0	805	0	0	0	0	0	0
10	Dog24	0	0	0	0	0	0	1747	0	0	0	0	0
11	Dog29	0	0	0	0	0	0	0	0	0	0	0	0
12	Dog3	0	921	0	0	0	944	0	0	0	0	0	0
13	Dog30	0	0	0	0	0	0	0	0	0	0	0	0
14	Dog31	0	0	1596	1625	0	0	0	1523	0	0	1536	0
15	Dog8	0	0	0	0	0	0	0	0	0	0	0	0
16	Dog9	0	0	0	0	0	0	0	0	0	0	0	0
17													
18													

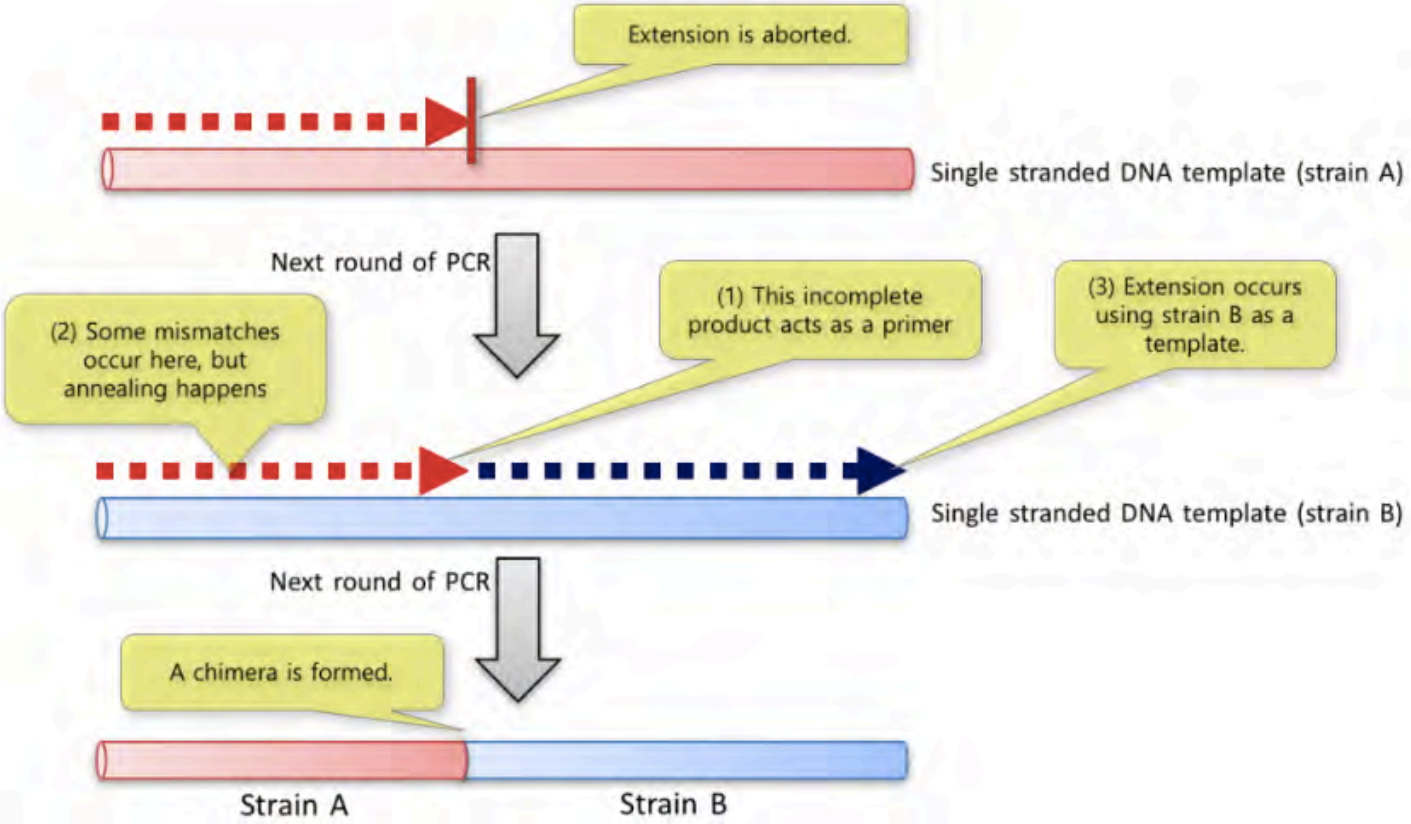
DADA2 workflow

Chimera Sequence

- Chimeras are sequences formed from two or more biological sequences joined together.
- Amplicons with chimeric sequences can be formed during PCR.
- Chimeras are rare with shotgun sequencing but are common in amplicon sequencing when closely related sequences are amplified.

DADA2 workflow

Chimera Sequence



https://help.ezbiocloud.net/

DADA2 workflow

Chimera Checking and Removal

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbose=TRUE)
```

```
## Identified 9112 bimeras out of 13527 input sequences.
```

```
dim(seqtab.nochim)
```

```
## [1]   15 4415
```

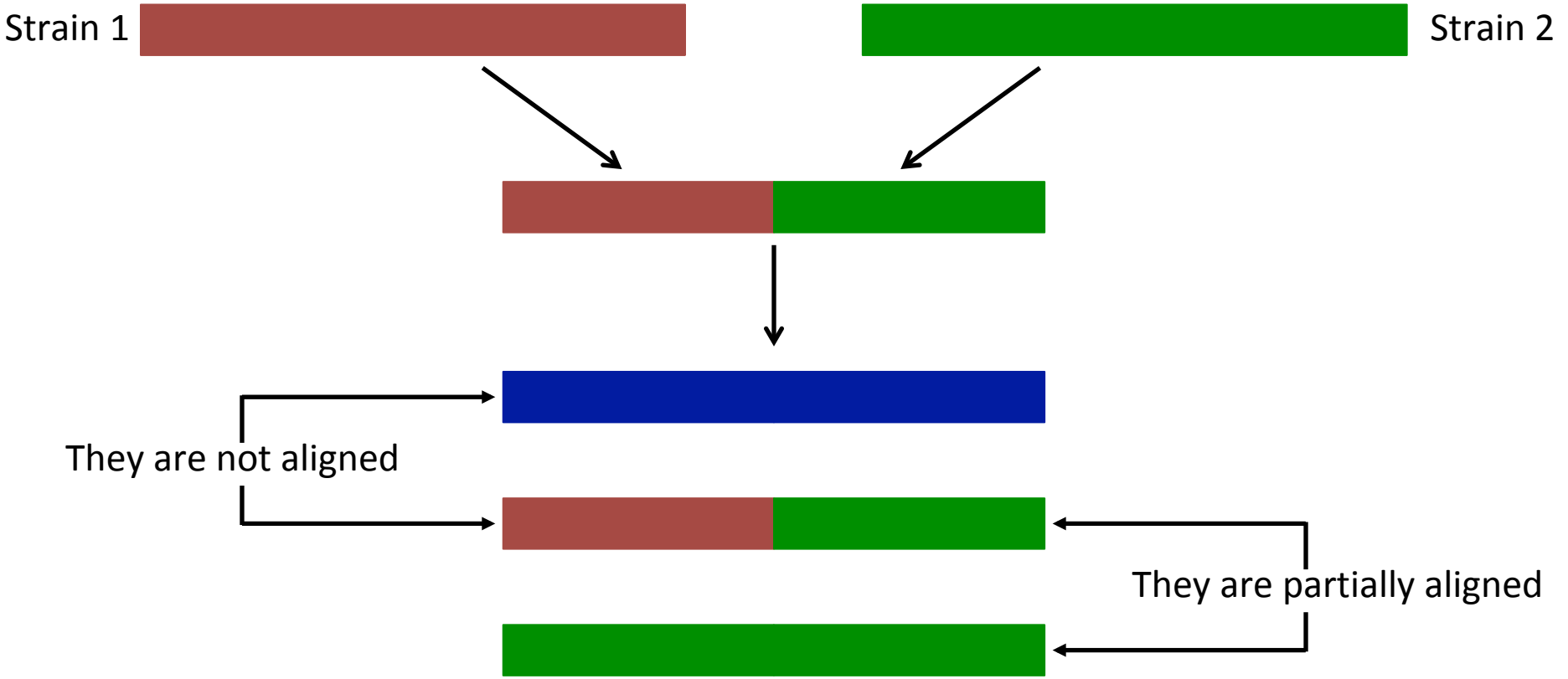
```
sum(seqtab.nochim)/sum(seqtab)
```

```
## [1] 0.5094968
```

- It uses *de novo* to check for two parent chimeras.
- Chimeric sequences are identified if they can be exactly reconstructed by combining a left-segment and a right-segment from two more abundant "parent" sequences.

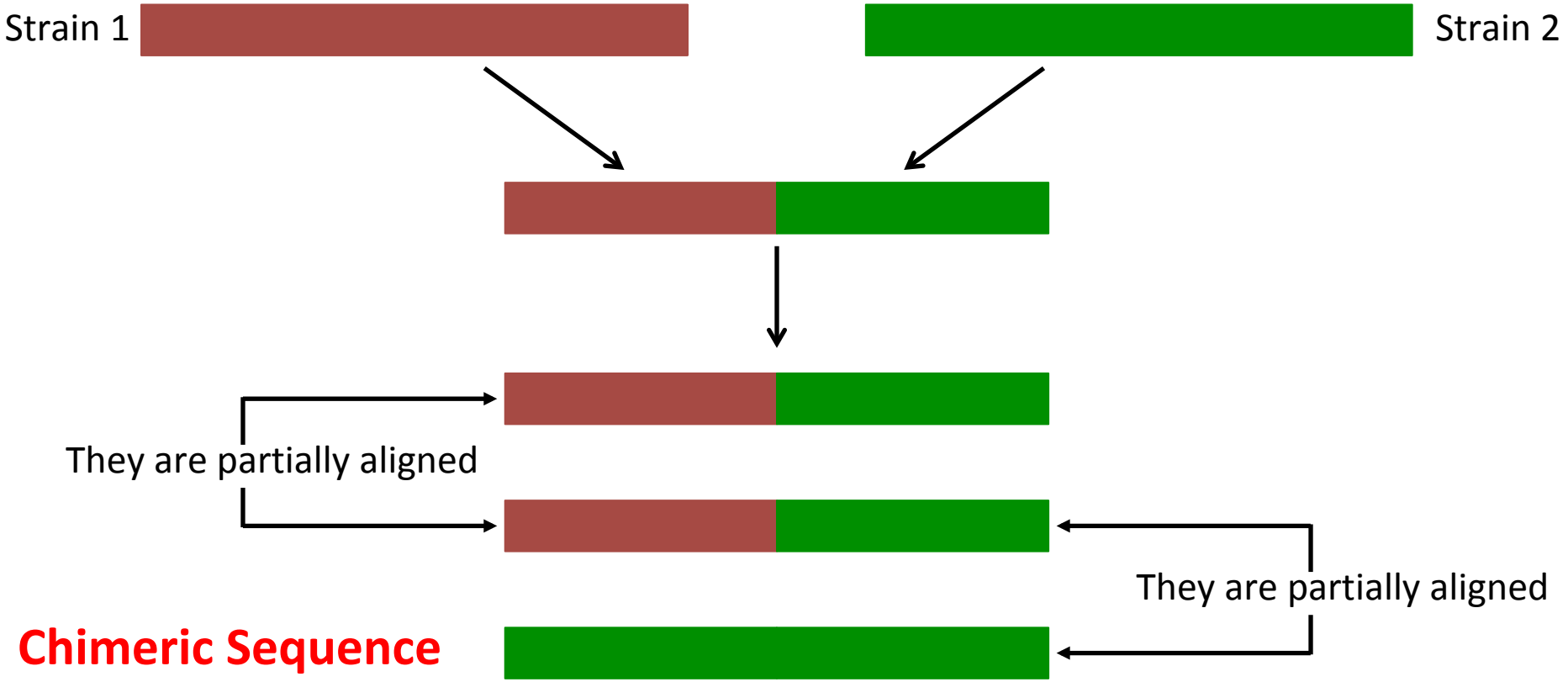
DADA2 workflow

Chimera Checking and Removal



DADA2 workflow

Chimera Checking and Removal



DADA2 workflow

Chimera Checking and Removal

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbose=TRUE)
```

```
## Identified 9112 bimeras out of 13527 input sequences.
```

```
dim(seqtab.nochim)
```

```
## [1]   15 4415
```

```
sum(seqtab.nochim)/sum(seqtab)
```

```
## [1] 0.5094968
```

- It uses *de novo* to check for two parent chimeras.
- Chimeric sequences are identified if they can be exactly reconstructed by combining a left-segment and a right-segment from two more abundant "parent" sequences.

H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S rRNA analysis pipeline

Taxonomic classification and alignment using the dada2 pipeline

DADA2 workflow

Track Reads through DADA2 pipeline

```
getN <- function(x) sum(getUniques(x))
track.nbr.reads <- cbind(out, sapply(dadaF, getN), sapply(dadaR, getN), sapply(merge.reads, getN), rowSums(seqtab
.nochim))

colnames(track.nbr.reads) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track.nbr.reads) <- list.sample.names
head(track.nbr.reads)
```

```
##        input filtered denoisedF denoisedR merged nonchim
## Dog1  118343    84815     82888     83710  76306   35440
## Dog10  79342    61952     60537     61130  54948   24103
## Dog15 131483    91235     89781     90372  83013   48592
## Dog16 114424    83564     81753     82442  74533   46006
## Dog17  99610    72919     71427     72229  67364   42960
## Dog2  108679    79668     77366     78301  69262   30129
```

DADA2 workflow

Assign Taxonomy

```
taxa <- assignTaxonomy(seqtab.nochim, paste(MY_HOME, "/dada2_tutorial_dog/RefSeq-RDP16S_v3_May2018.fa.gz", sep=''
), multithread=TRUE) # change the path
```

RefSeq-RDP16S_v3_May2018.fa.gz

Three most common 16S databases: Silva, RDP and GreenGenes

DADA2 workflow

Assign Taxonomy

Maintained:

- Silva version 132, Silva version 128, Silva version 123 (Silva dual-license)
- RDP trainset 16, RDP trainset 14
- GreenGenes version 13.8
- UNITE (use the General Fasta releases)

Contributed:

- RefSeq + RDP (NCBI RefSeq 16S rRNA database supplemented by RDP)
  - Reference files formatted for `assignTaxonomy`
  - Reference files formatted for `assignSpecies`
- GTDB: Genome Taxonomy Database (More info: http://gtdb.ecogenomic.org/)
  - Reference files formatted for `assignTaxonomy`
  - Reference files formatted for `assignSpecies`
- HitDB version 1 (Human InTestinal 16S rRNA)
- RDP fungi LSU trainset 11
- Silva Eukaryotic 18S, v132 & v128
- PR2 version 4.7.2+. SEE NOTE BELOW.

DADA2 workflow

Assign Taxonomy

```
taxa.print <- taxa # Removing sequence rownames for display only
rownames(taxa.print) <- NULL
head(taxa.print)
```

```
##      Kingdom    Phylum           Class          Order
## [1,] "Bacteria" "Firmicutes"     "Clostridia"   "Clostridiales"
## [2,] "Bacteria" "Firmicutes"     "Clostridia"   "Clostridiales"
## [3,] "Bacteria" "Bacteroidetes"  "Bacteroidia"  "Bacteroidales"
## [4,] "Bacteria" "Bacteroidetes"  "Bacteroidia"  "Bacteroidales"
## [5,] "Bacteria" "Firmicutes"     "Clostridia"   "Clostridiales"
## [6,] "Bacteria" "Firmicutes"     "Clostridia"   "Clostridiales"
##      Family                  Genus
## [1,] "Peptostreptococcaceae" "Clostridium_XI"
## [2,] "Peptostreptococcaceae" "Clostridium_XI"
## [3,] "Prevotellaceae"        "Alloprevotella"
## [4,] "Prevotellaceae"        "Alloprevotella"
## [5,] "Peptostreptococcaceae" "Clostridium_XI"
## [6,] "Peptostreptococcaceae" "Clostridium_XI"
##      Species
## [1,] "Clostridium_hiranonis(AB023970)"
## [2,] "Clostridium_hiranonis(AB023970)"
## [3,] "Prevotellamassilia_timonensis(NR_144750.1)"
## [4,] "Prevotellamassilia_timonensis(NR_144750.1)"
## [5,] "Clostridium_hiranonis(AB023970)"
## [6,] "Clostridium_hiranonis(AB023970)"
```

DADA2 workflow

Assign Taxonomy

```
write.csv(taxa, file="ASVs_taxonomy.csv")
saveRDS(taxa, "ASVs_taxonomy.rds")
```

	A	B	C	D	E	F	G	H	I	J	K
1		Kingdom	Phylum	Class	Order	Family	Genus	Species			
2	ASV_1	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
3	ASV_2	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
4	ASV_3	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
5	ASV_4	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
6	ASV_5	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
7	ASV_6	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
8	ASV_7	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
9	ASV_8	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
10	ASV_9	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
11	ASV_10	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
12	ASV_11	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
13	ASV_12	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
14	ASV_13	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
15	ASV_14	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
16	ASV_15	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
17	ASV_16	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
18	ASV_17	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
19	ASV_18	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
20	ASV_19	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
21	ASV_20	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
22	ASV_21	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
23	ASV_22	Bacteria	Firmicutes	Clostridia	Clostridiales	Peptostreptc	Clostridium_	Clostridium_hiranonis(AB023970)			
24	ASV_23	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
25	ASV_24	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			
26	ASV_25	Bacteria	Bacteroidete	Bacteroidia	Bacteroidale	Prevotellace	Alloprevotell	Prevotellamassilia_timonensis(NR_144750.1)			

DADA2 workflow

Assign Taxonomy

```
asv_headers <- vector(dim(seqtab.nochim)[2], mode="character")
count.asv.tab <- t(seqtab.nochim)
row.names(count.asv.tab) <- sub(">", "", asv_headers)
write.csv(count.asv.tab, file="ASVs_counts.csv")
saveRDS(count.asv.tab, file="ASVs_counts.rds")
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Dog1	Dog10	Dog15	Dog16	Dog17	Dog2	Dog22	Dog23	Dog24	Dog29	Dog3	Dog30	Dog31	Dog8	Dog9
2	ASV_1	0	0	0	0	0	373	1926	0	0	0	0	0	0	0	0
3	ASV_2	0	0	0	0	0	0	0	955	0	0	921	0	0	0	0
4	ASV_3	242	0	0	0	0	0	0	0	0	0	0	0	1596	0	0
5	ASV_4	205	0	0	0	0	0	0	0	0	0	0	0	1625	0	0
6	ASV_5	0	0	0	0	0	276	1516	0	0	0	0	0	0	0	0
7	ASV_6	0	0	0	0	0	0	0	805	0	0	944	0	0	0	0
8	ASV_7	0	0	0	0	0	0	0	0	1747	0	0	0	0	0	0
9	ASV_8	223	0	0	0	0	0	0	0	0	0	0	0	1523	0	0
10	ASV_9	0	0	0	0	0	283	1453	0	0	0	0	0	0	0	0
11	ASV_10	0	0	0	0	0	277	1459	0	0	0	0	0	0	0	0
12	ASV_11	195	0	0	0	0	0	0	0	0	0	0	0	1536	0	0
13	ASV_12	0	0	0	0	0	322	1366	0	0	0	0	0	0	0	0
14	ASV_13	184	0	0	0	0	0	0	0	0	0	0	0	1479	0	0
15	ASV_14	0	0	0	0	0	0	0	806	0	0	857	0	0	0	0
16	ASV_15	170	0	0	0	0	0	0	0	0	0	0	0	1480	0	0
17	ASV_16	0	0	0	0	0	0	0	0	1645	0	0	0	0	0	0
18	ASV_17	0	0	0	0	0	270	1369	0	0	0	0	0	0	0	0
19	ASV_18	220	0	0	0	0	0	0	0	0	0	0	0	1373	0	0
20	ASV_19	0	0	0	0	0	0	0	772	0	0	798	0	0	0	0
21	ASV_20	174	0	0	0	0	0	0	0	0	0	0	0	1376	0	0
22	ASV_21	0	0	0	0	0	0	0	0	1524	0	0	0	0	0	0
23	ASV_22	0	0	0	0	0	0	0	743	0	0	720	0	0	0	0
24	ASV_23	188	0	0	0	0	0	0	0	0	0	0	0	1254	0	0
25	ASV_24	179	0	0	0	0	0	0	0	0	0	0	0	1260	0	0
26	ASV_25	154	0	0	0	0	0	0	0	0	0	0	0	1263	0	0
27	ASV_26	167	0	0	0	0	0	0	0	0	0	0	0	1238	0	0
28	ASV_27	0	0	0	0	0	0	0	689	0	0	716	0	0	0	0

DADA2 workflow

Alignment

```
library(DECIPHER)
```

Tools for curating, analyzing, and manipulating biological sequences.

```
seqs <- getSequences(seqtab.nochim)
names(seqs) <- seqs # This propagates to the tip labels of the tree
alignment <- AlignSeqs(DNAStringSet(seqs), anchor=NA)
```

DADA2 workflow

Construct Phylogenetic Tree

```
library(phangorn)
```

```
phang.align <- phyDat(as(alignment, "matrix"), type="DNA")
dm <- dist.ml(phang.align)
treeNJ <- NJ(dm) # Note, tip order != sequence order
fit = pml(treeNJ, data=phang.align)
```

It constructs a neighbor-joining tree.

1. Change sequence alignment output into **phyDat** structure.
2. Create distance matrix using **dist.ml**.
3. Perform neighbor joining.
4. Perform internal maximum likelihood.

DADA2 workflow

Construct Phylogenetic Tree

```
fitGTR <- update(fit, k=4, inv=0.2)
fitGTR <- optim.pml(fitGTR, model="GTR", optInv=TRUE, optGamma=TRUE,
rearrangement = "stochastic", control = pml.control(trace = 0))
```

It fits a GTR+G+I (Generalized time-reversible with Gamma rate variation) maximum likelihood tree using the neighbor-joining tree as a starting point.

```
saveRDS(fitGTR, "phangorn.tree.RDS")
```

H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course

16S analysis pipeline
Taxonomic classification and alignment using the dada2 pipeline
Practical

Practical

The practical is available here:

https://iallali.github.io/DADA2_pipeline/16SrRNA_DADA2_pipeline.html

Practical

5. Merge the Paired Reads

In this step, we merge the forward and reverse reads to obtain the full sequences.

```
merge.reads <- mergePairs(dadaF, filt.dataF, dadaR, filt.dataR, verbose=TRUE)
```

```
## 76306 paired-reads (in 1153 unique pairings) successfully merged out of 82378 (in 2526 pairings) input.
```

```
## 54948 paired-reads (in 720 unique pairings) successfully merged out of 60139 (in 1811 pairings) input.
```

```
## 83013 paired-reads (in 843 unique pairings) successfully merged out of 89310 (in 2175 pairings) input.
```

```
## 74533 paired-reads (in 1093 unique pairings) successfully merged out of 81135 (in 2674 pairings) input.
```

```
## 67364 paired-reads (in 855 unique pairings) successfully merged out of 71053 (in 1743 pairings) input.
```

```
## 69262 paired-reads (in 1194 unique pairings) successfully merged out of 76778 (in 2891 pairings) input.
```

```
## 97401 paired-reads (in 954 unique pairings) successfully merged out of 106466 (in 2867 pairings) input.
```

```
## 129237 paired-reads (in 1632 unique pairings) successfully merged out of 146610 (in 5263 pairings) input.
```

```
## 111558 paired-reads (in 1143 unique pairings) successfully merged out of 120597 (in 3110 pairings) input.
```

```
## 85319 paired-reads (in 839 unique pairings) successfully merged out of 92957 (in 2348 pairings) input.
```

```
## 68550 paired-reads (in 915 unique pairings) successfully merged out of 74010 (in 2198 pairings) input.
```

```
## 92870 paired-reads (in 1266 unique pairings) successfully merged out of 107043 (in 4252 pairings) input.
```

```
## 101935 paired-reads (in 981 unique pairings) successfully merged out of 108579 (in 2495 pairings) input.
```