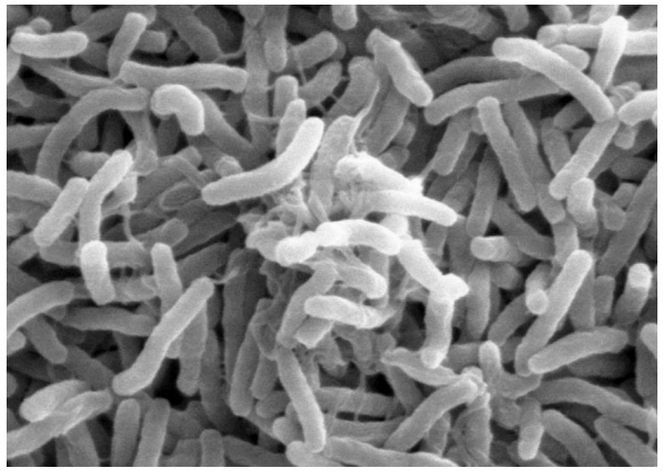
Bioinformatic analysis of metabarcoding data with DADA2

Practical part

Clarisse Lemonnier

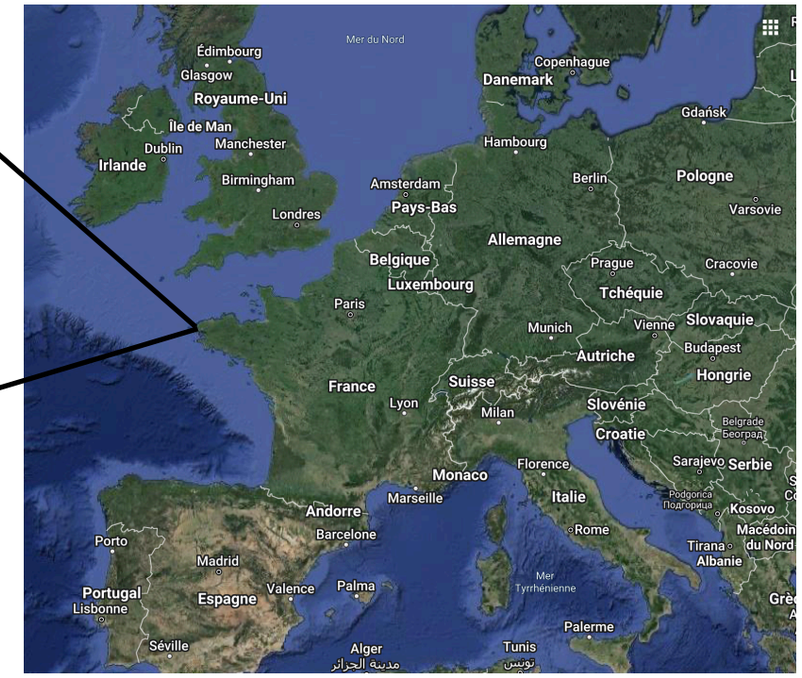
SOME FEW WORDS ON THE TUTORIAL DATASET

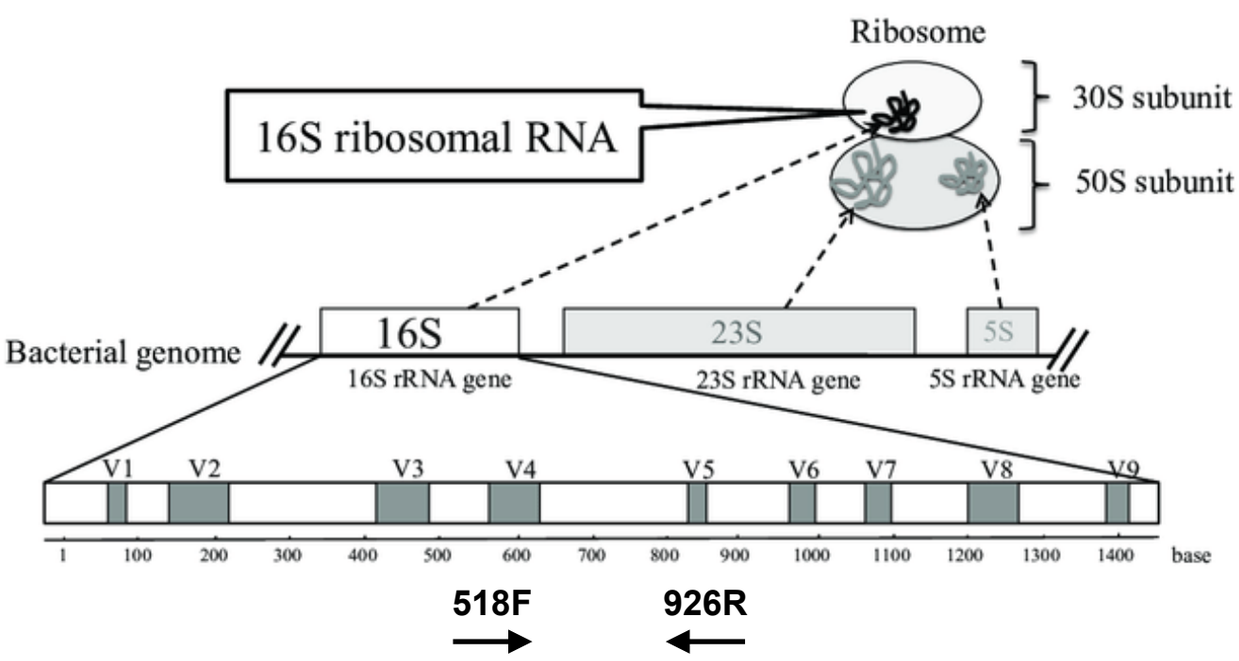
Bacteria



Bay of Brest

France





Primer type	Primer name	Sequence (5'-3')
Forward	518F	NNNNNCCAGCAGCYGCGGTAAN
Reverse	926R_1	CCGTCAATTCNTTTRAGT
	926R_2	CCGTCAATTTCTTTGAGT
	926R_3	CCGTCTATTCCTTTGANT

Barcode of around 373bp - targeting the V4/V5 region of 16S rRNA gene

SOME FEW WORDS ON THE TUTORIAL DATASET

You have 5 samples - Illumina MiSeq 2x250bp

Already demultiplexed by the sequencing platform

S3_1sept14

S3_1sept14_R1.fastq

S3_1sept14_R2.fastq

S3_9janv15

S3_9janv15_R1.fastq

S3_9janv15_R2.fastq

S3_24jui14

S3_24jui_R1.fastq

S3_24jui_R2.fastq

S3_8aou14

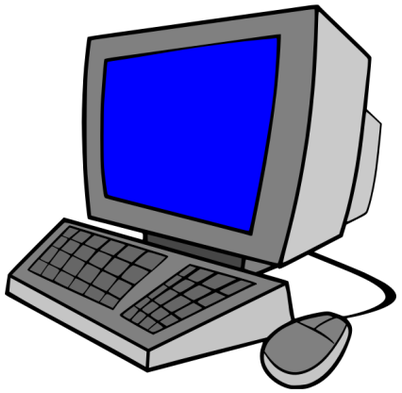
S3_8aou14_R1.fastq

S3_8aou14_R2.fastq

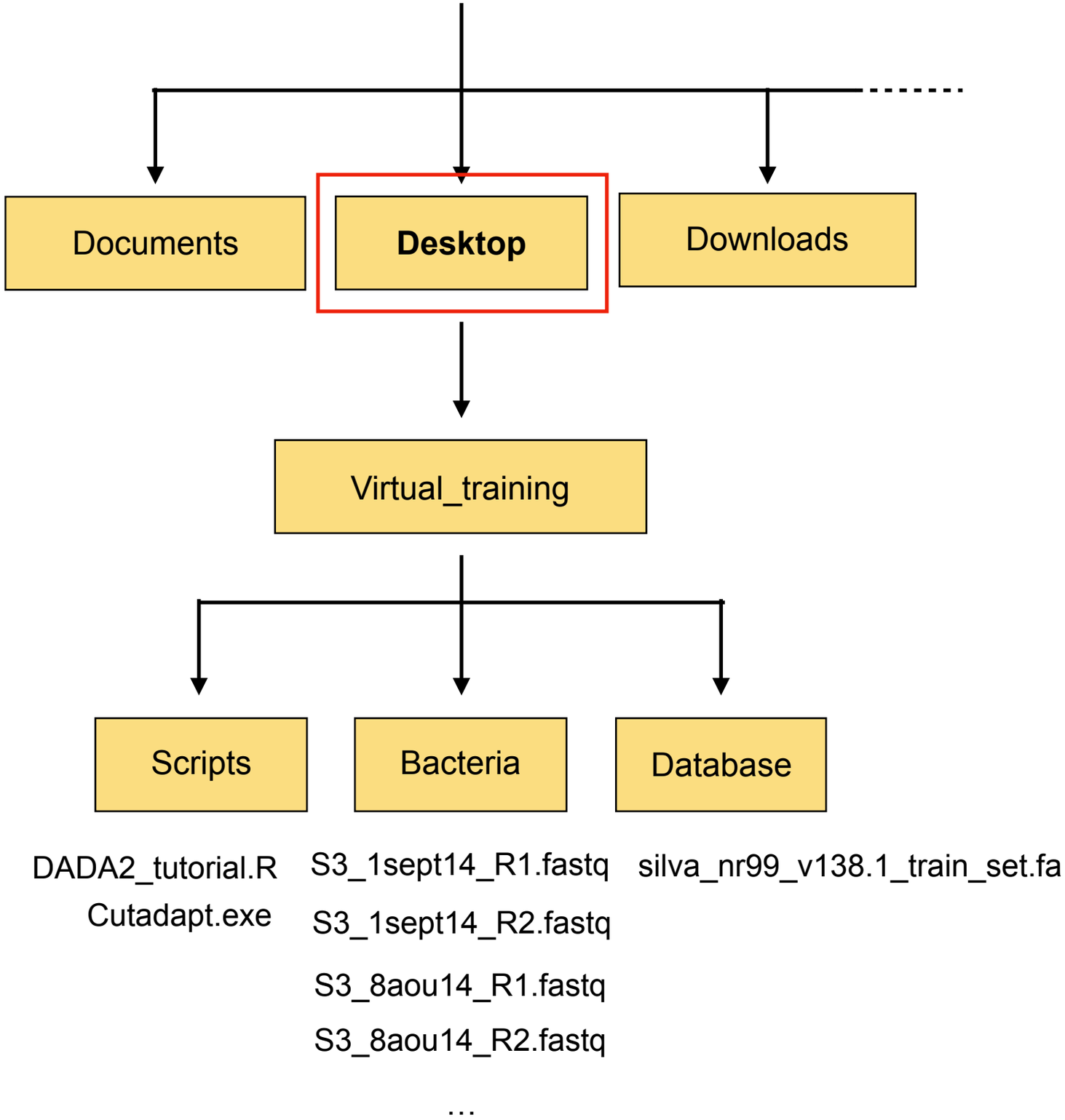
S3_16fev15

S3_16fev15_R1.fastq

S3_16fev15_R2.fastq



BEFORE STARTING…



OPEN THE SCRIPT DADA2_tutorial.R

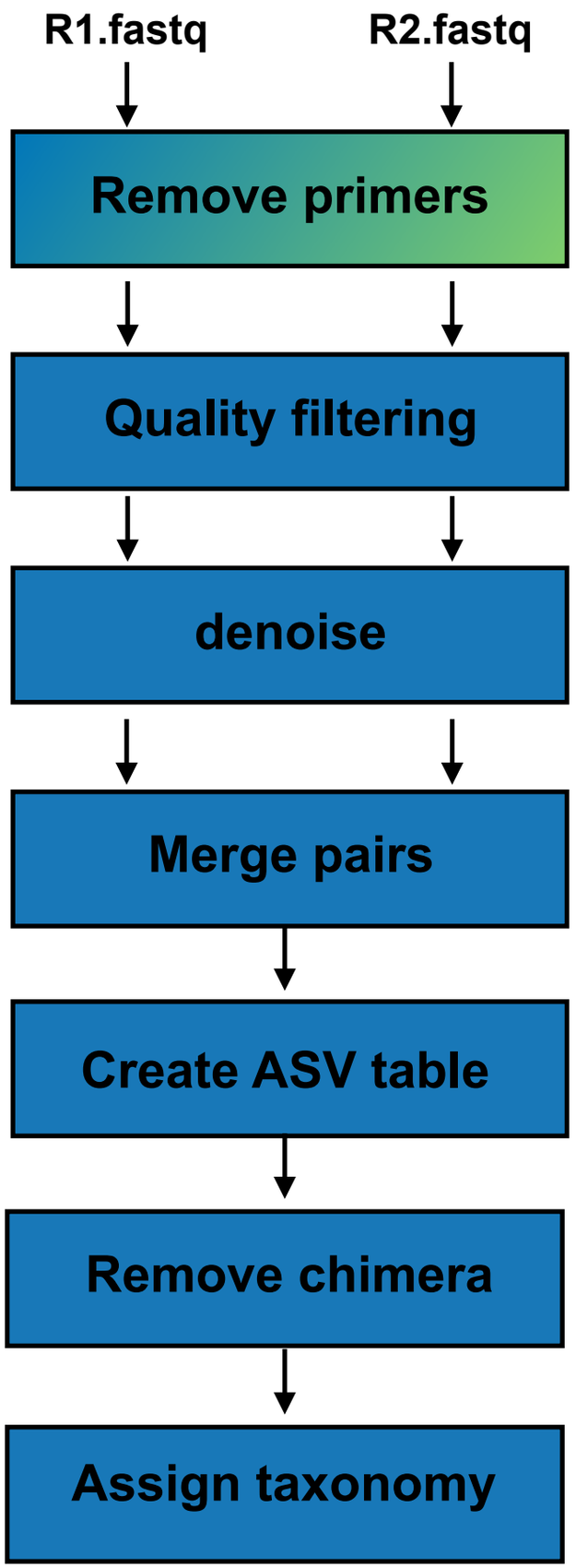
DADA2_tutorial.R*

Source on Save | Run | Source

```
1  ################################
2  ######## DADA2 TUTORIAL ########
3  ################################
4
5  # This is an RScript of DADA2 pipeline for metabarcoding data.
6  # This pipeline is also fully detailed in DADA2 website : https://benjjneb.github.io/dada2/tutori
7
8  # Important informations needed before the analysis :
9
10 # We are analysing V3-V4 region of 16S to assess the diversity of bacteria
11 # Forward primer is : NNNNNCCAGCAGCYGCGGTAAN
12
13 # Reverse primer are : CCGTCAATTCNTTTRAGT
14 #                      CCGTCAATTTCTTTGAGT
15 #                      CCGTCTATTCCTTTGANT
16
17 # Expected barcode length is around 370-375nt (there is variability depending on species)
18
19 # Reference database : for bacteria the best one is SILVA
20
21 # This pipeline is divided into 11 STEPS :
```

…

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

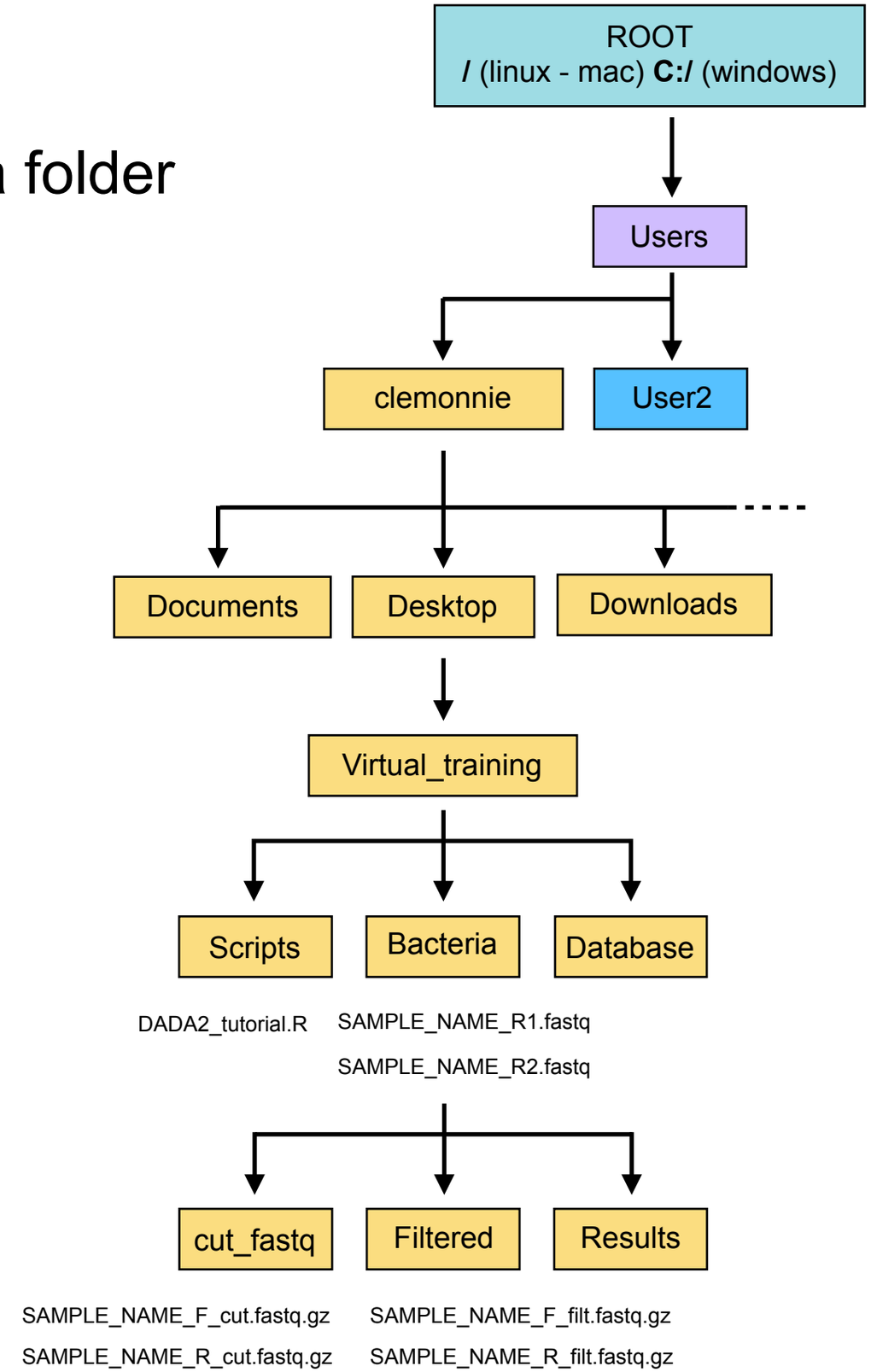
STEP 2 : SET FILE PATHS

path - must be the absolute path to Bacteria folder (where fastq files are)

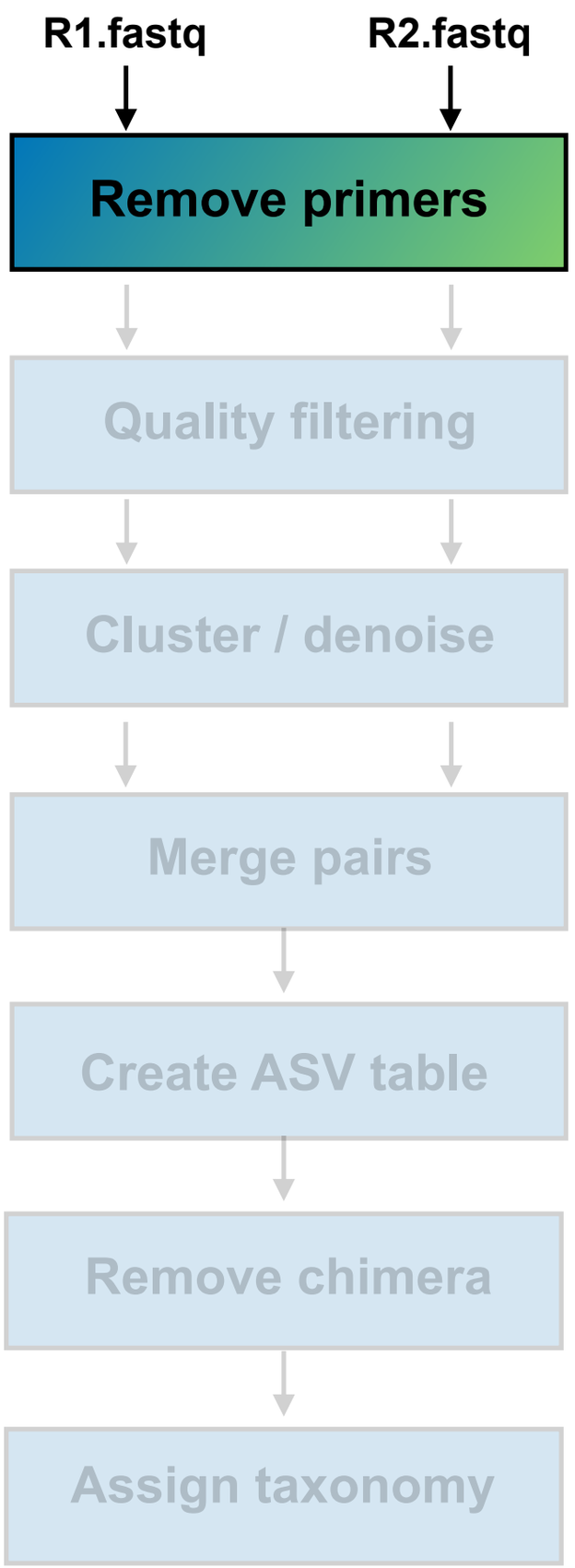
```
path <- "YOURPATH" # CHANGE ME
```

In this step, we will create 3 new subfolders in the folder Bacteria to store intermediate files of the different steps

We will also store the absolute path directing to each files, so they can be used in some commands



DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

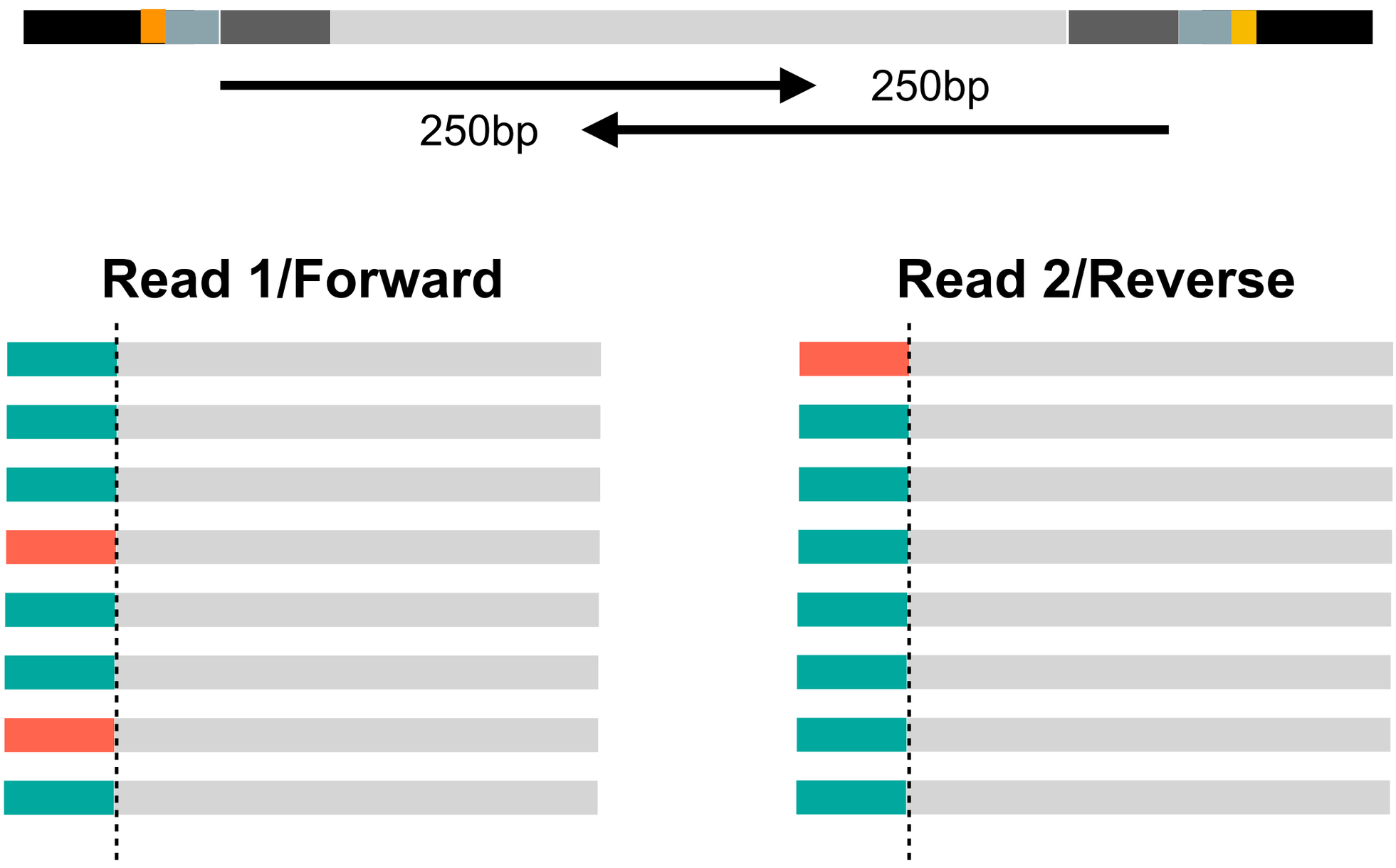
removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

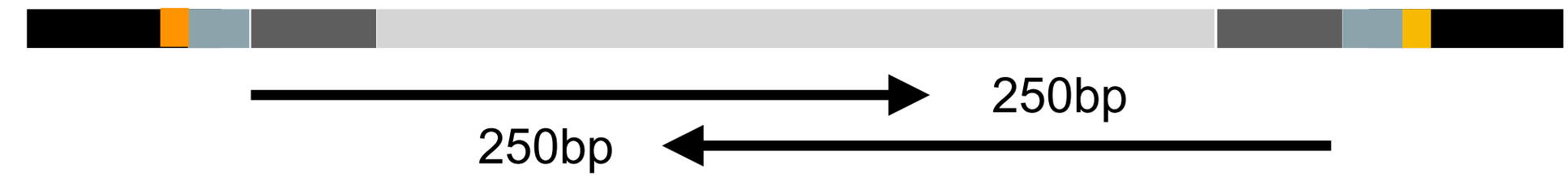
Cutadapt removes adapter sequences from sequencing reads

STEP 3 : REMOVE PRIMERS

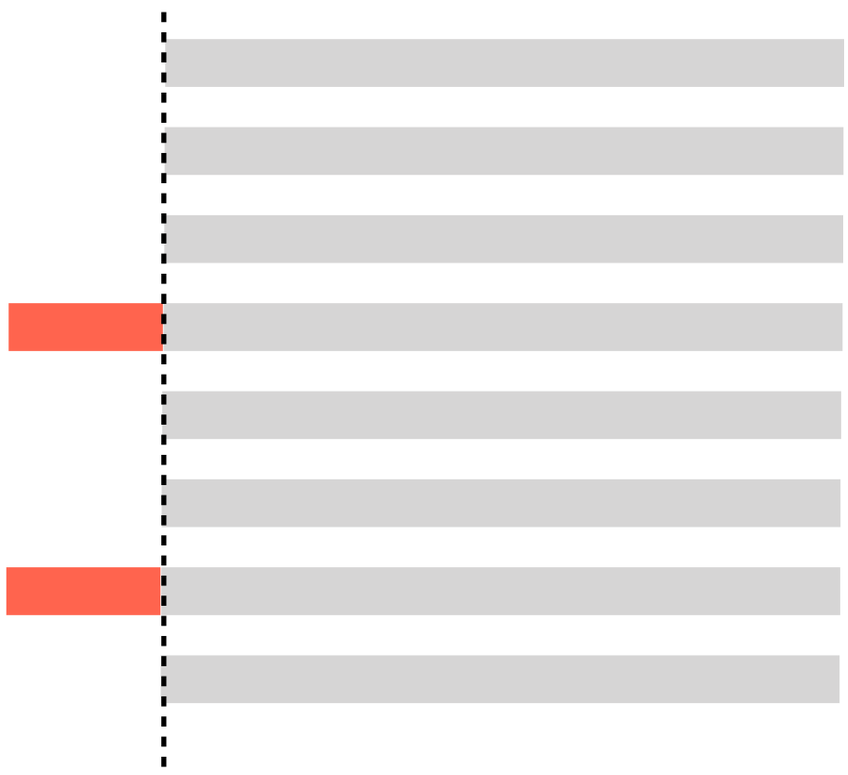


Look reads that start with an exact match to the primer sequence

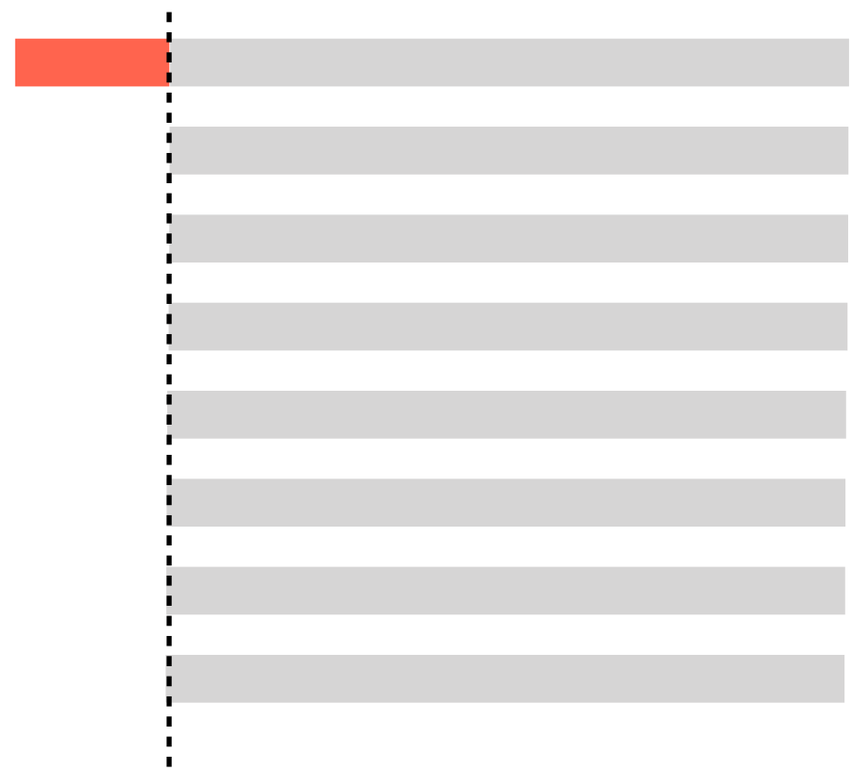
STEP 3 : REMOVE PRIMERS



Read 1/Forward

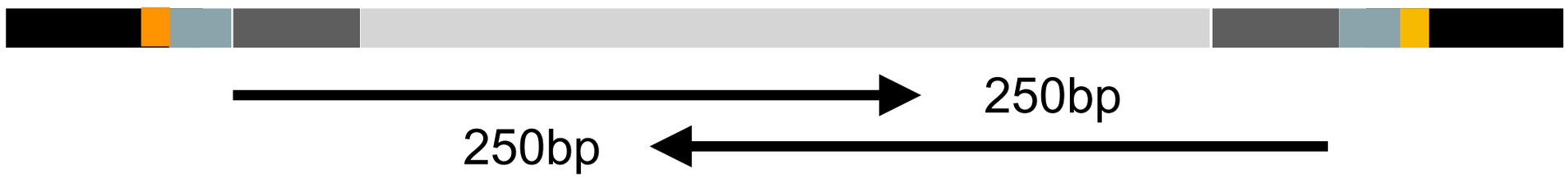


Read 2/Reverse



If you have an exact match, it trims the sequence

STEP 3 : REMOVE PRIMERS

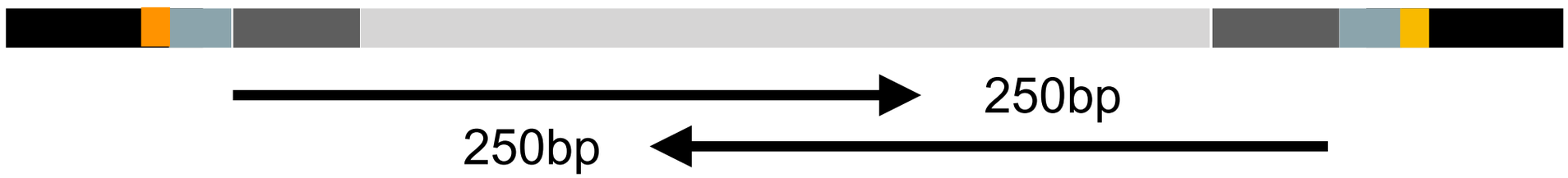


Read 1/Forward

Read 2/Reverse

It removes sequences that do not have a perfect match

STEP 3 : REMOVE PRIMERS



Read 1/Forward

Read 2/Reverse

Remove also the corresponding sequence in the other R1 or R2 file
(Cutadapt only)

STEP 3 : REMOVE PRIMERS

Primer forward sequence

NNNNNCCAGCAGCYGCGGTAAN

For cutadapt : you can combine different versions of a primer directly

For the dada2 removePrimer() function, you can only give one sequence

Primer reverse sequence

CCGTCAATTCNTTTRAGT
CCGTCAATTTCTTTGAGT
CCGTCTATTCCTTTGANT

STEP 3 : REMOVE PRIMERS

Primer forward sequence

NNNNNCCAGCAGCYGCGGTAAN

For cutadapt : you can combine different versions of a primer directly

For the dada2 removePrimer() function, you can only give one sequence

Primer reverse sequence

CCGTCAATTCNTTTRAGT
CCGTCAATTTCTTTGAGT
CCGTCTATTCCTTTGANT

IUPAC Code

Symbol	Nucleotide Base
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
N	A or C or G or T
M	A or C
R	A or G
W	A or T
S	C or G
Y	C or T
K	G or T
V	Not T
H	Not G
D	Not C
B	Not A

STEP 3 : REMOVE PRIMERS

Primer forward sequence

NNNNNCCAGCAGCYGCGGTAAN

For cutadapt : you can combine different versions of a primer directly

For the dada2 removePrimer() function, you can only give one sequence

IUPAC Code

Symbol	Nucleotide Base
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
N	A or C or G or T
M	A or C
R	A or G
W	A or T
S	C or G
Y	C or T
K	G or T
V	Not T
H	Not G
D	Not C
B	Not A

Primer reverse sequence

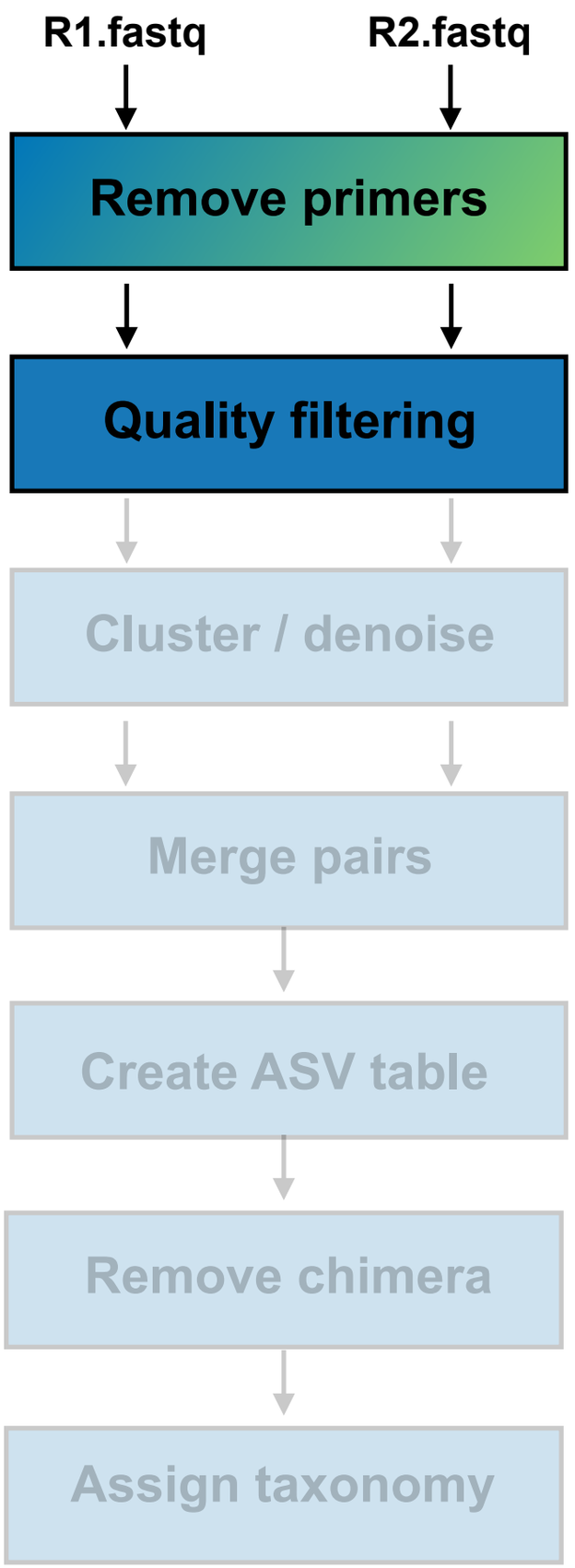
CCGTCAATTCNTTTRAGT
CCGTCAATTTCTTTGAGT
CCGTCTATTCCTTTGANT

CCGTCWATTYNTTTRANT

So we will write a unique sequence that combine all the three version of the reverse primer

Be careful !! If you make a mistake at this step, you will have troubles…

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

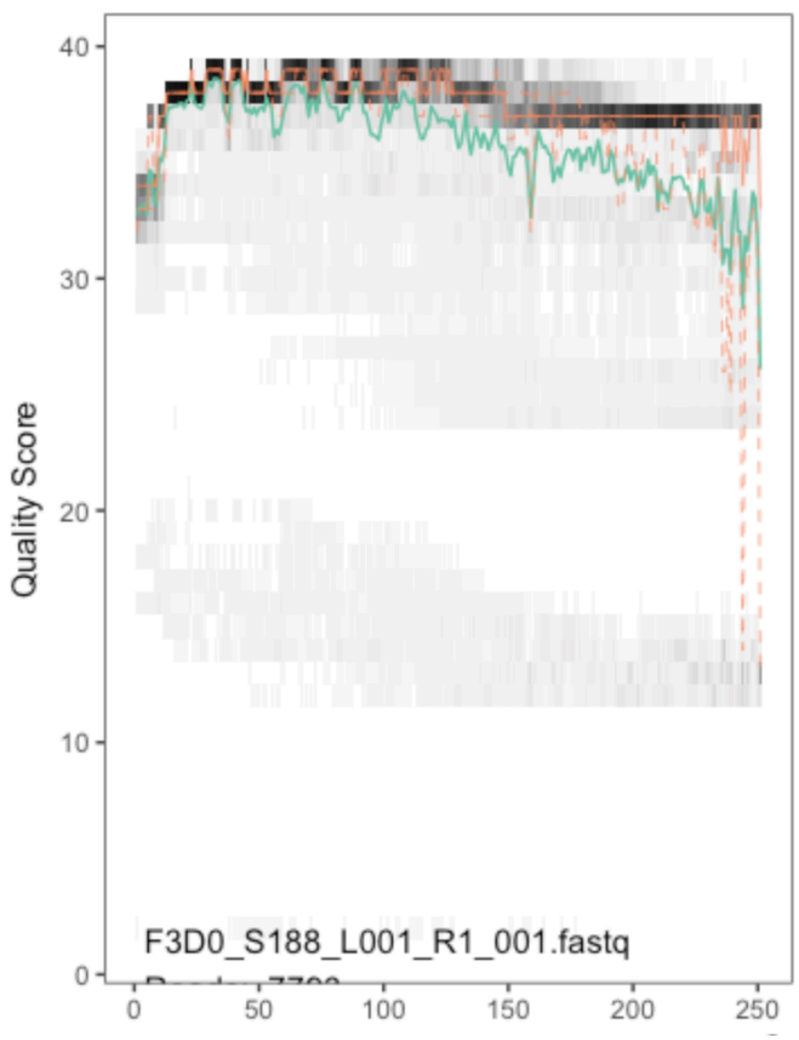
STEP 4 : FILTER AND TRIM

DADA2 algorithm use a model to learn the error rates. So we need to be sure that the reads are of good quality to avoid less well controlled errors that can arise

STEP 4 : FILTER AND TRIM

We will inspect the read quality manually, and trim them if they are parts that are of low quality

```
plotQualityProfile(cut_F_reads[1:5])
```



The sequencing quality score of a given base, Q, is defined by the following equation:

$$Q = -10\log_{10}(e)$$

where e is the estimated probability of the base call being wrong.

- **Higher Q scores** indicate a smaller probability of error.
- **Lower Q scores** can result in a significant portion of the reads being unusable. They may also lead to increased false-positive variant calls, resulting in inaccurate conclusions.

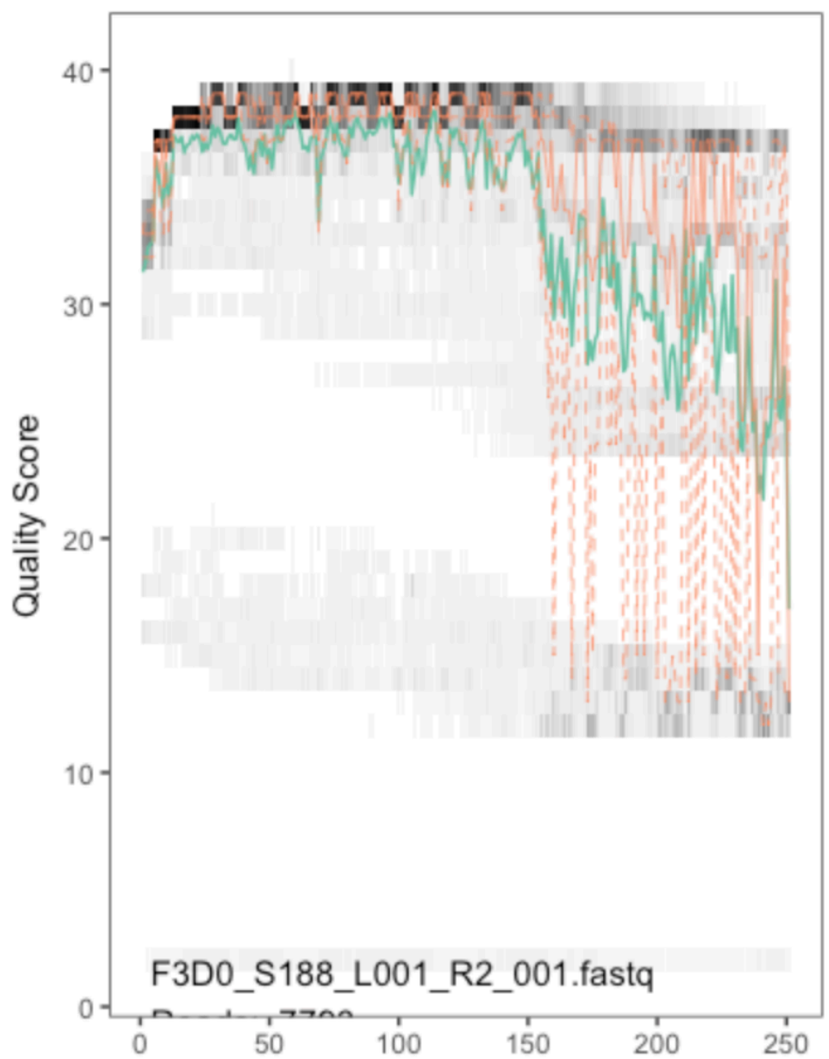
Relationship Between Sequencing Quality Score and Base Call Accuracy

Quality Score	Probability of Incorrect Base Call
10 (Q10)	1 in 10
20 (Q20)	1 in 100
30 (Q30)	1 in 1000

STEP 4 : FILTER AND TRIM

We will inspect the read quality manually, and trim them if they are parts that are of low quality

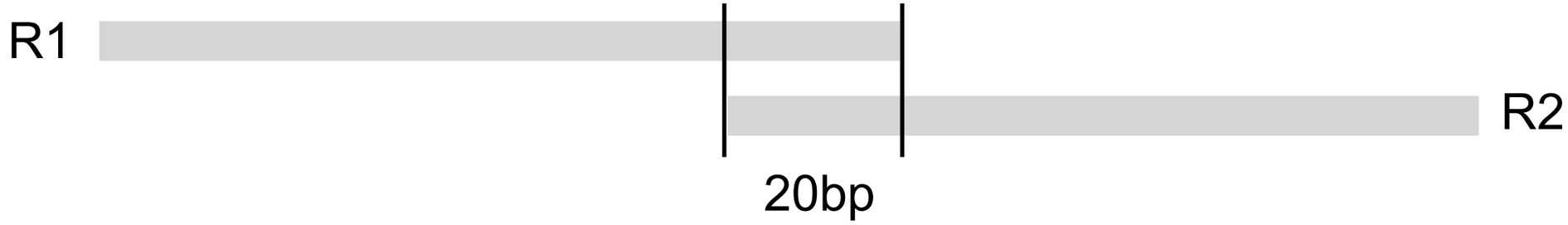
```
plotQualityProfile(cut_F_reads[1:5])
```



STEP 4 : FILTER AND TRIM

Be careful !

Ideally : overlap of 20nt
DADA2 remove reads with overlap less than 12nt



For the tutorial dataset : expected barcode length is around 373
So length of reads trim should follow this equation :

R1-length + R2-length - 12 > 370

If we cannot trim as much bases as we wanted, this is not too worrisome, as DADA2 incorporate quality information into the error models, which makes the algorithm robust to lower quality sequences.

STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

Precise the file names with their absolute path, for R1 and R2 separately

STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

truncLen : where to trim the R1 and R2 read respectively

STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

maxN: maximum number of ambiguous bases

Should always be set to 0 because dada2 model errors cannot deal ambiguous bases.

STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

maxEE = sets maximum number of expected errors in R1 and R2 reads respectively

You can relax it a little bit (increasing the value) if your data is of very low quality

STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

truncQ = truncate read at the first instance of a quality score less than or equal to truncQ

You can lower it or even set truncQ=0 if your data is of very low quality

STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

rm.phix = phix is an internal control of Illumina sequencer

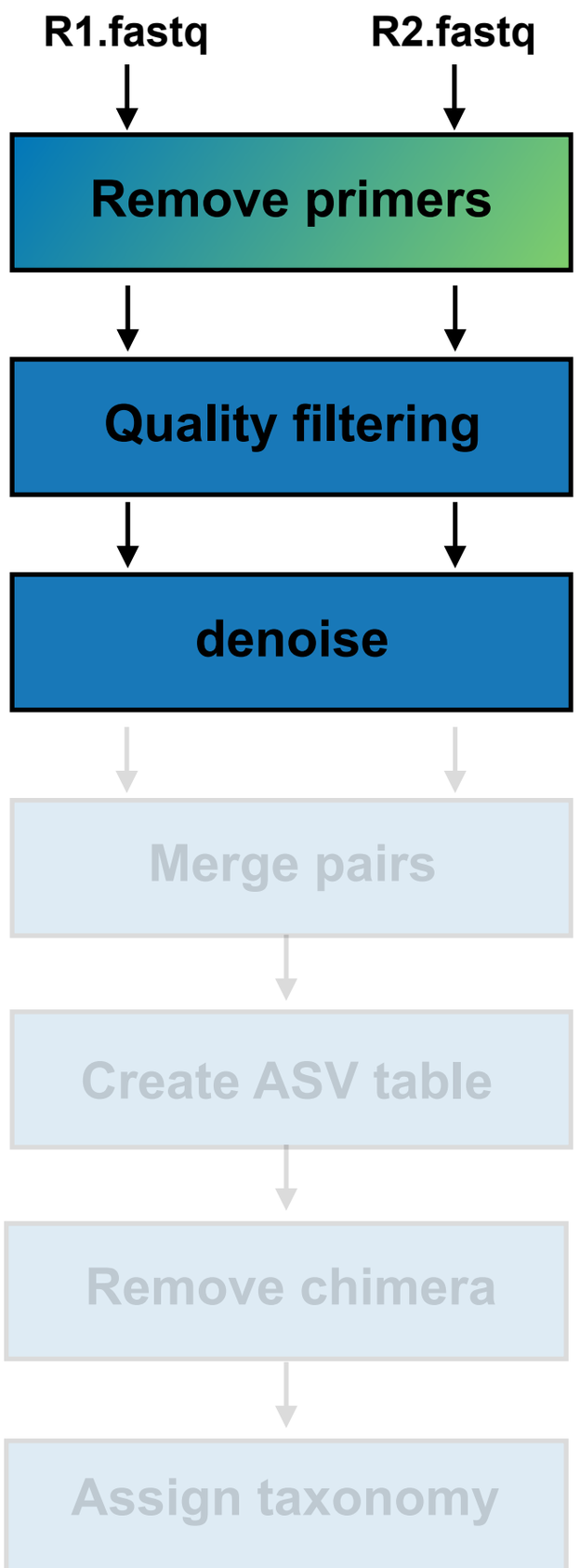
STEP 4 : FILTER AND TRIM

```
filterAndTrim(cut_F_reads, filt_F_reads, cut_R_reads, filt_R_reads,
              truncLen=c(200,190),
              maxN=0,
              maxEE=c(2,2),
              truncQ=2,
              rm.phix=TRUE,
              matchIDs=TRUE,
              compress=TRUE,
              multithread=FALSE)
```

matchIDs = if TRUE, only pairs that shared id field are output

Very important if you trimmed primers with the removePrimers() function

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

STEP 5 : LEARN THE ERROR RATE AND INFER SAMPLES

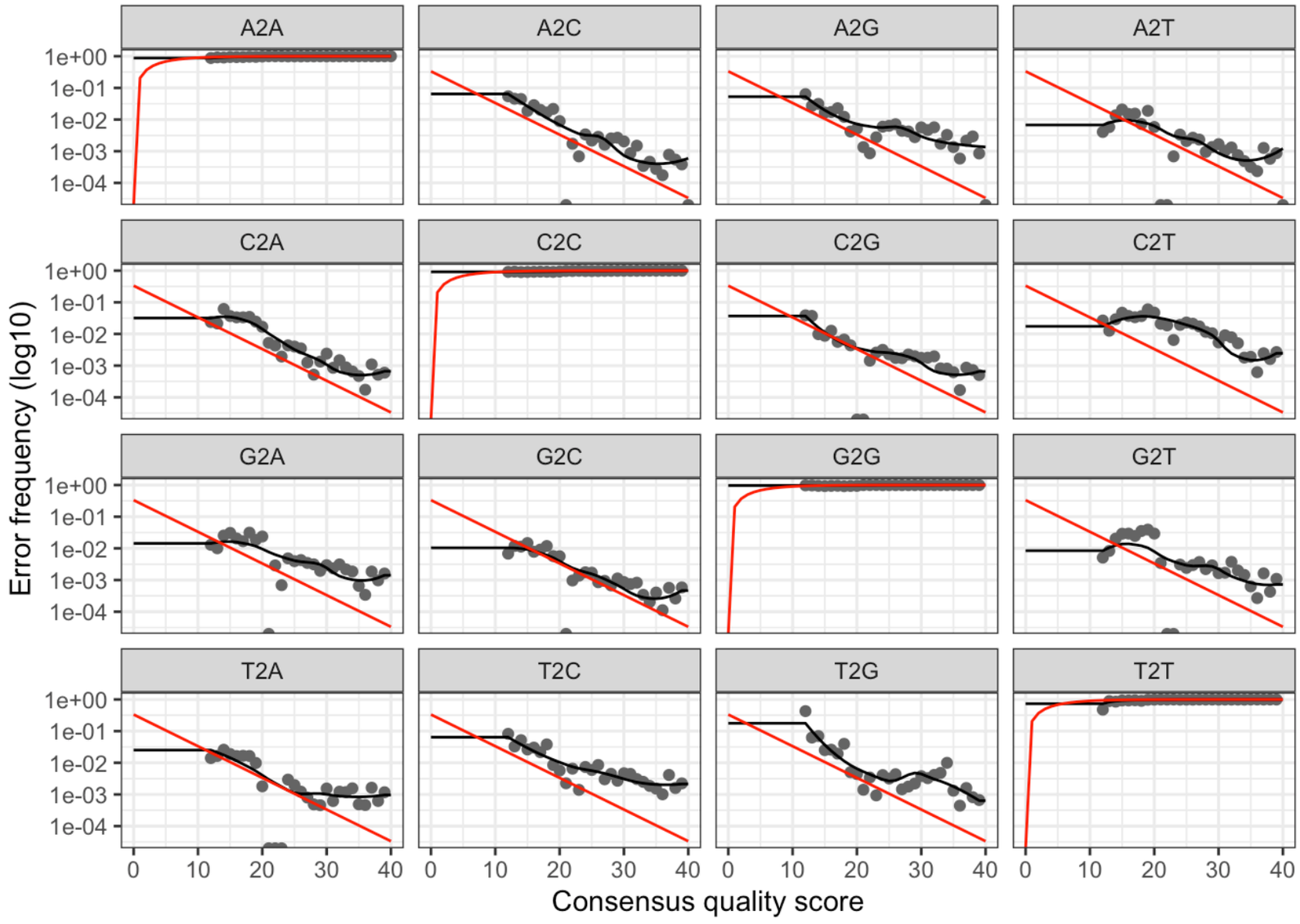
The DADA2 algorithm depends on a parametric error model

Each amplicon dataset will have a different set of error rates

learnErrors method learn the error rates by alternating estimation of the error rates and inference of sample composition until they converge on a jointly consistent solution.

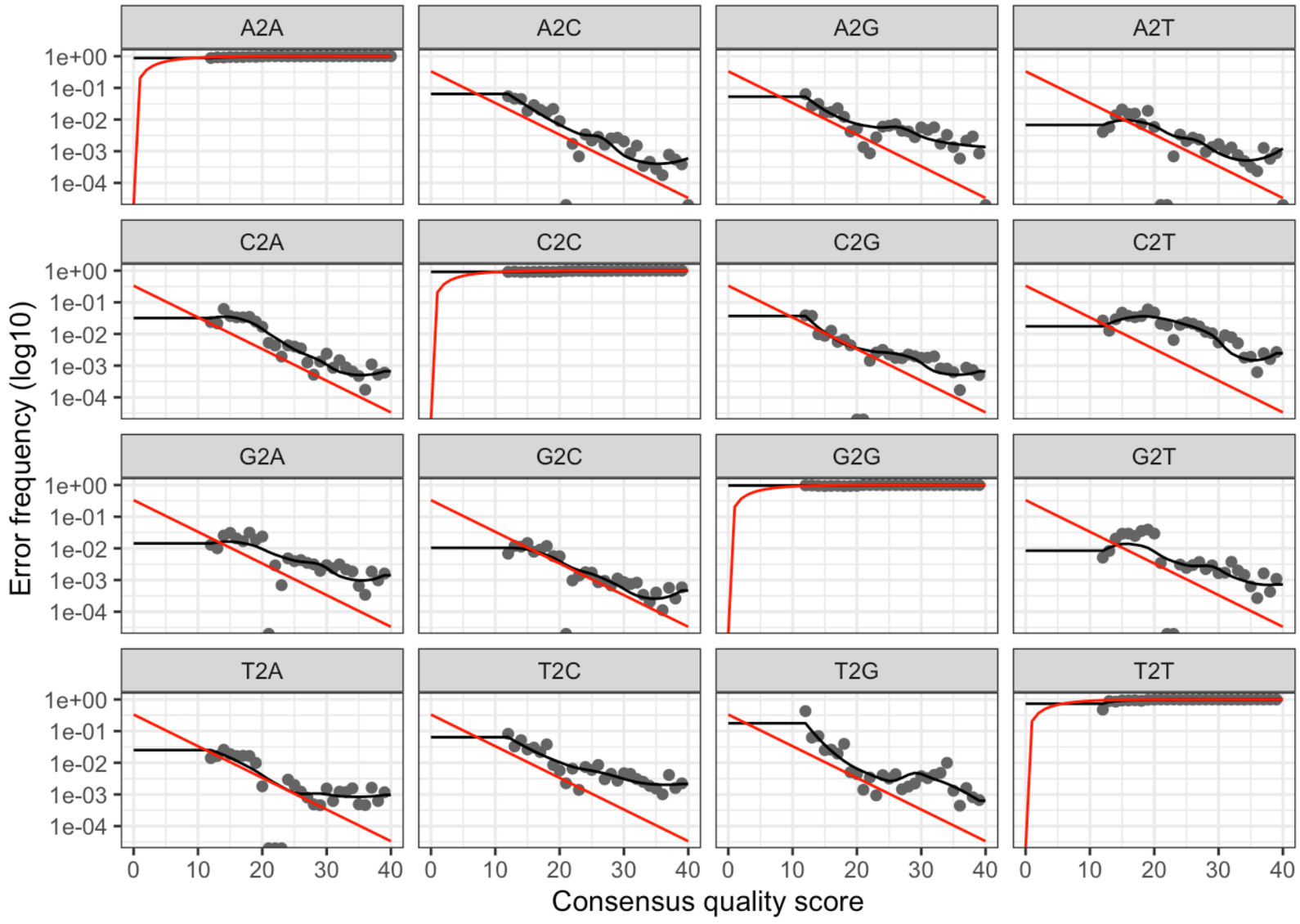
This approach is advantageous as it builds unique error rates for each sequencing runs.

STEP 5 : LEARN THE ERROR RATE AND INFER SAMPLES



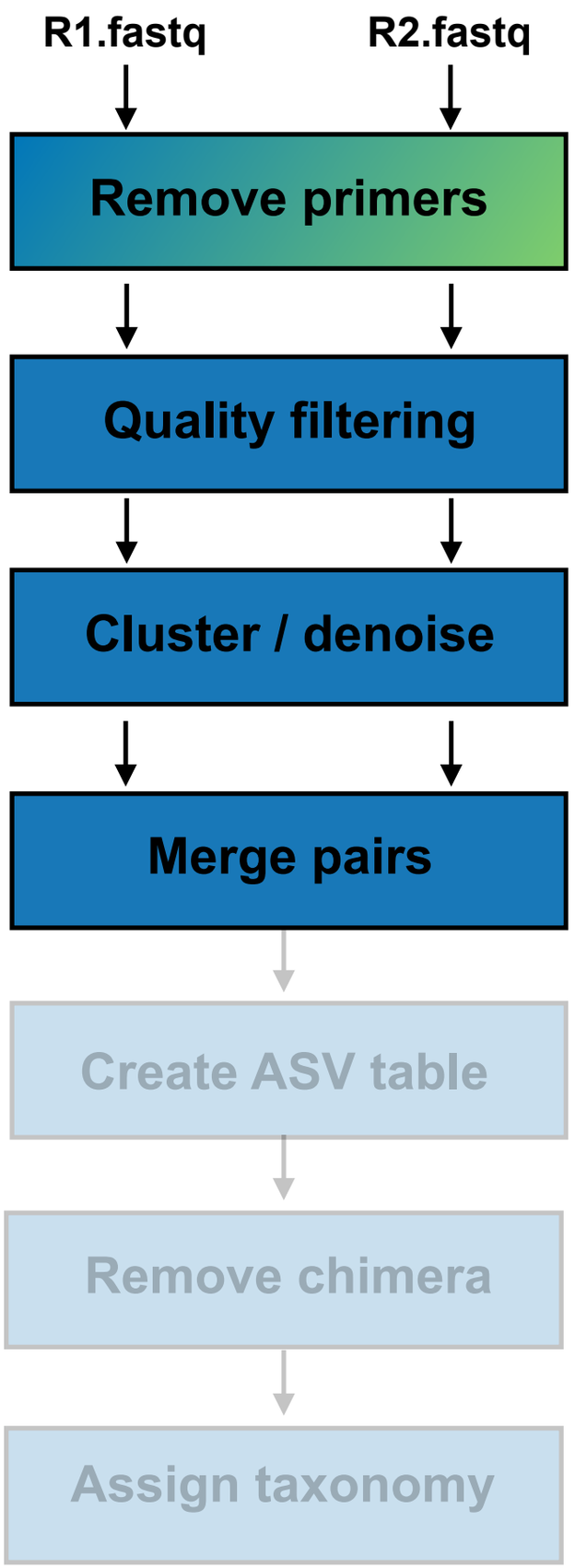
Here is a figure resuming the estimated error rates for each possible transition
Check : black lines (model) fit the observed error rates (point)
And this estimated error rates should decrease with increasing quality score

STEP 5 : LEARN THE ERROR RATE AND INFER SAMPLES



If the model does not give good resultats, dada2 team recommend to increase the number of reads used to learn the error rates (nreads parameter)

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

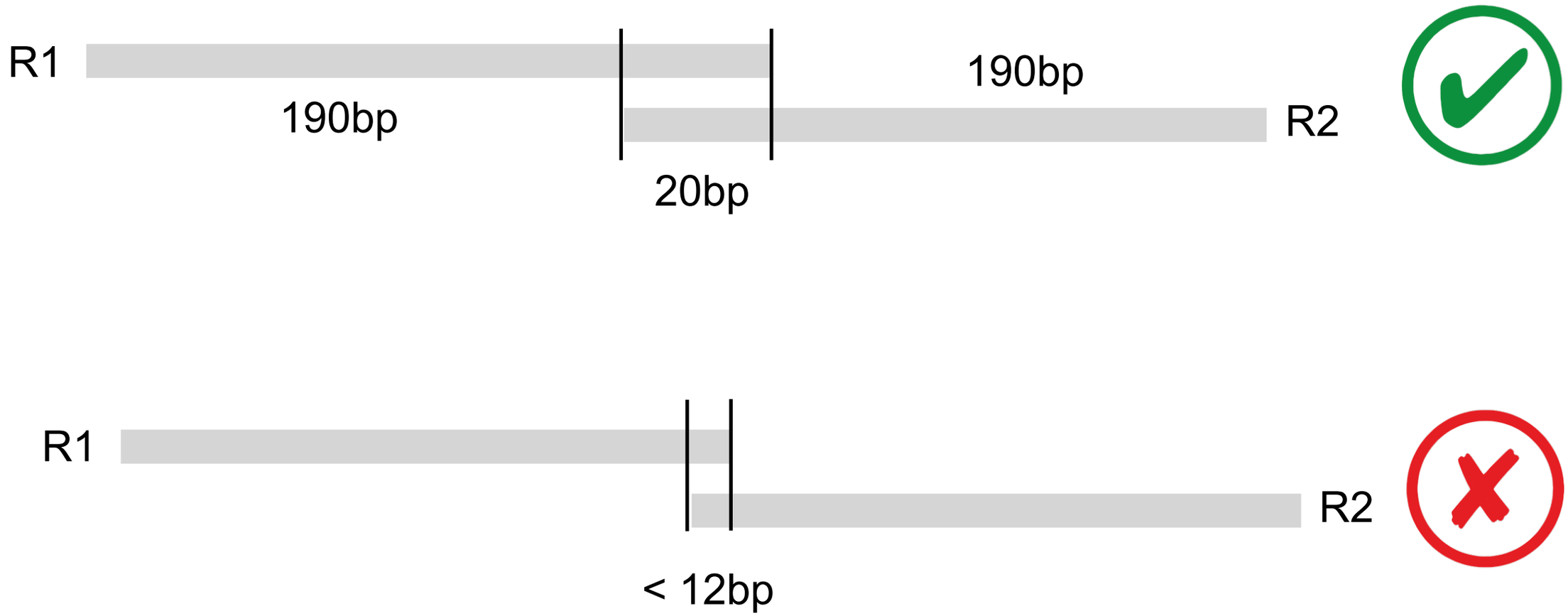
removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

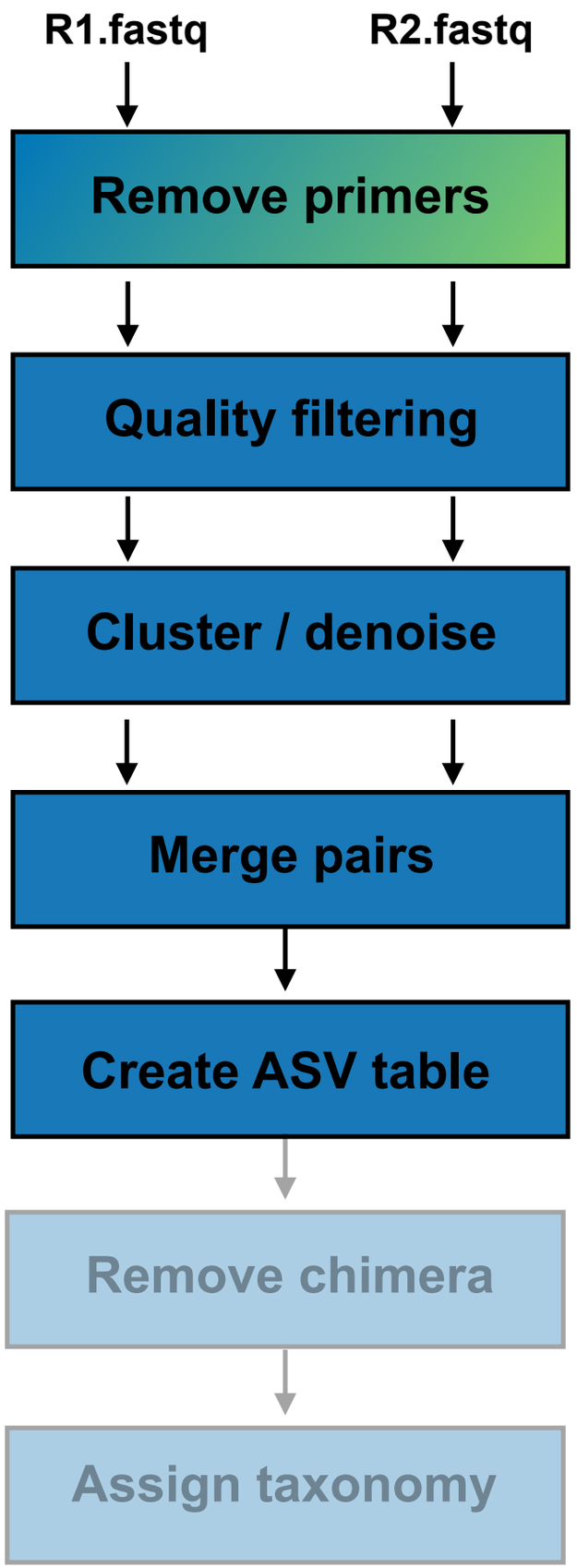
STEP 6 : MERGE PAIRS



You should not loose too many reads at this step.

Otherwise maybe you trimmed to much? DADA2 remove reads that have an overlap smaller than12bp

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

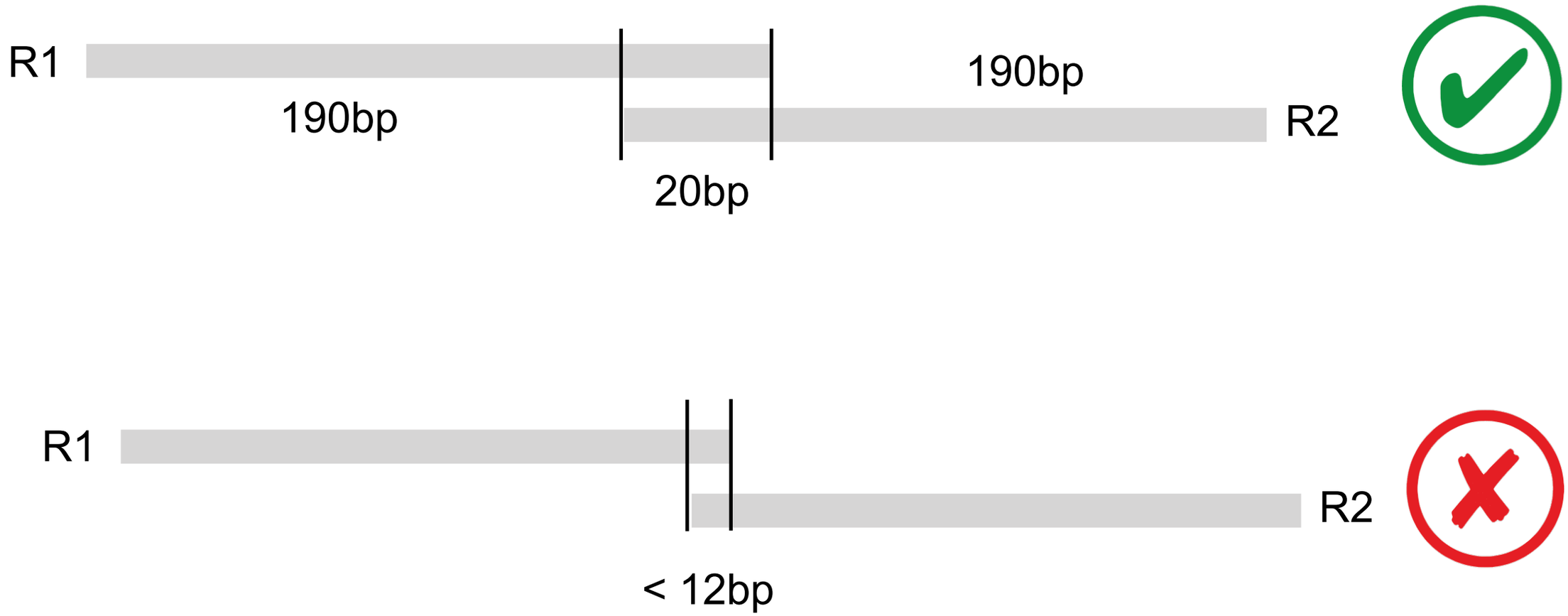
removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

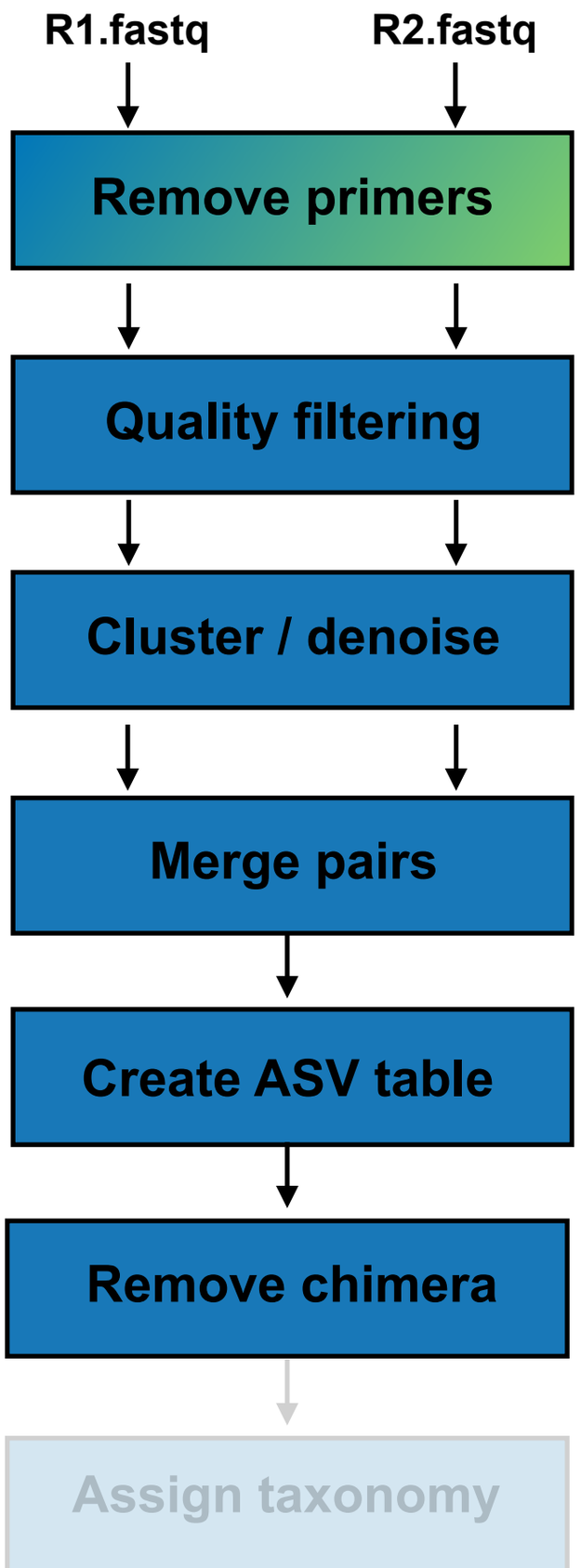
STEP 6 : MERGE PAIRS



You should not loose too many reads at this step.

Otherwise maybe you trimmed to much? DADA2 remove reads that have an overlap smaller than12bp

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

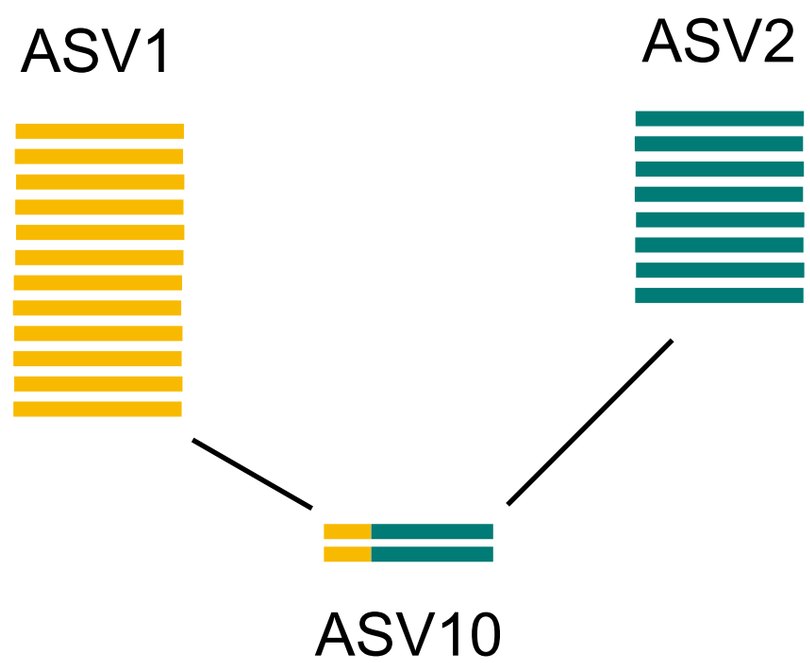
removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

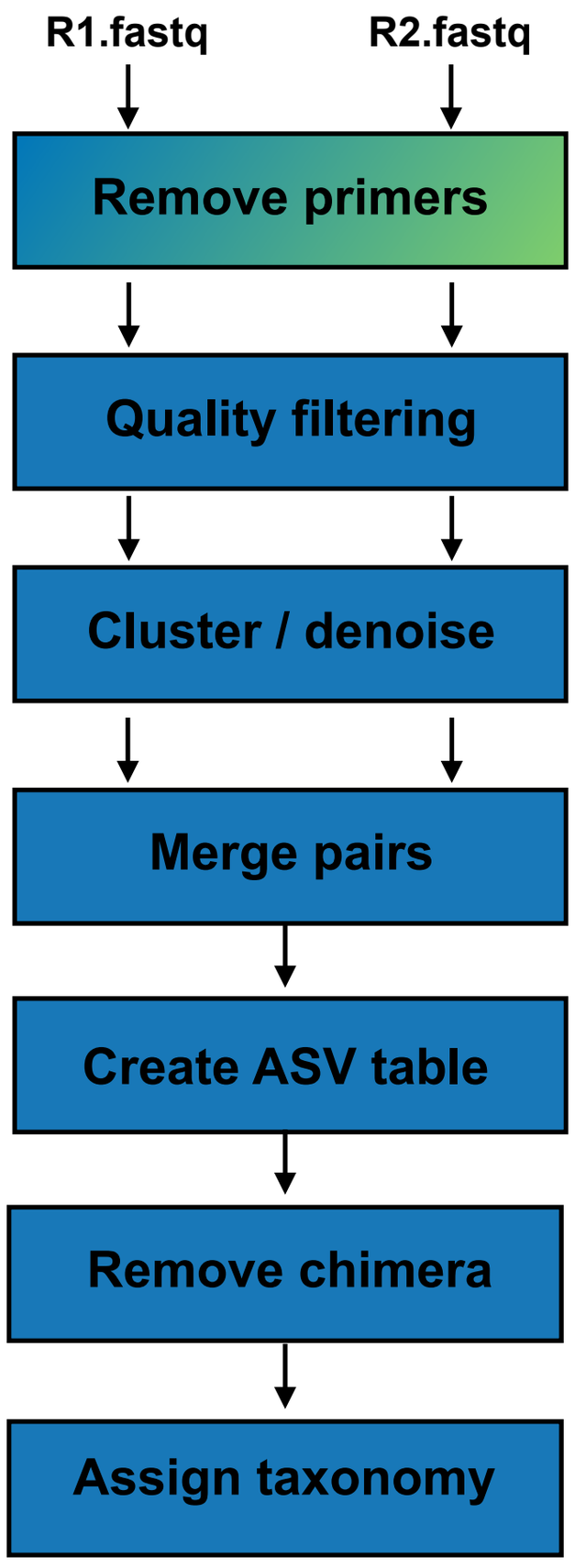
STEP 8 : REMOVE CHIMERA



The principle is to remove any ASV that is the exact combination of 2 more abundant ASV

Here ASV10 will be removed

DADA2 pipeline



removePrimers()
But cutadapt is better

filterAndTrim()

learnErrors()

dada()

mergePairs()

makeSequenceTable()

removeBimeraDenovo()

assignTaxonomy()

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

STEP 10 : ASSIGN TAXONOMY

Taxonomy is assigned using a naive bayesian algorithm

The reference database should be a fasta file like this :

```
>Eubacteria;Cyanobacteria;Cyanophyceae;Oscillatoriales;Oscillatoriaceae;Phormidium;Phormidium tergestinum
GAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGTCTGCTTAACACATGCAAGTCGAACGGGCGCAGAAATGCGCTAGTGGCGGACGGGTGAGTAACACGTGAGAATCTGCCAA
>Eubacteria;Cyanobacteria;Cyanophyceae;Oscillatoriales;Oscillatoriaceae;Blennothrix;Blennothrix sp.
CGGACGGGTGAGTAACGCGTGAGAATCTGCCTTCTGGTCTGGGACAACAGAGGGAAACTTCTGCTAATCCCGGATGAGCCTTTCGGTAAAAGATAAATTGCCTGGAGATGAGCT
>Eubacteria;Cyanobacteria;Cyanophyceae;Oscillatoriales;Oscillatoriaceae;Phormidium;Phormidium irriguum
TGCAAGTCGAACGGACTCTTCGGAGTTAGTGGCGGACGGGTGAGTAACGCGTGAGAATCTAGCTTCTGGATGGGGACAACAGAGGGAAACGTCTGCTAATACCCGATGTGCCGA
>Eubacteria;Cyanobacteria;Cyanophyceae;Synechococcales;Prochlorotrichaceae;Halomicronema;Halomicronema metazoicum
GCGGAGGGTGAGTAACGCGTGAGGATCTGCCTTCAGGAGGGGGACAACAGTTGGAAACGACTGCTAATACCCCATATGCCCATGGGTGAAACGGTGAATTCCGCCTGAAGATGA
>Eubacteria;Cyanobacteria;Cyanophyceae;Oscillatoriales;Oscillatoriaceae;Oscillatoria;Oscillatoria miniata
CGGACGGGTGAGTAACGCGTGAGAATCTGCCTTCAGGTCTGGGACAACAGAAGGAAACTTCTGCTAATCCCGGATGAGCCTAACGGTCAAAGATTAATTGCCTGGAGATGAGCT
>Eubacteria;Cyanobacteria;Cyanophyceae;Pleurocapsales;Hyellaceae;Hyella;Hyella sp.
CTCAGAATGAACGCTGGCGGTATGCTTAACACATGCAAGTCGAACGAATTCTTCGGAATGAGTGGCGGACGGGTGAGTAACGCGTGAGAACCTGCCTTCAGGATGGGGACAACA
>Eubacteria;Cyanobacteria;Cyanophyceae;Nostocales;Microchaetaceae;Fortiea;Fortiea sp.
AGTTGATCCTGGCTCAGGATGAACGCTGGCGGTATGCTTAACACATGCAAGTCGAACGCAGCTTTAGGGCTGAGTGGCGGACGGGTGAGTAACGCGTGAGAATCTGCCTTCAGG
```

By default, assignTaxonomy() function of DADA2 take the following taxonomic levels :

"Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species"

STEP 10 : ASSIGN TAXONOMY

For bacteria, SILVA SSU is the best reference database

SILVA SSU 138.1 update release

	SSU Parc	SSU Ref NR 99	LSU Parc	LSU Ref NR 99
Minimal length	300	1200/900	300	1900
Quality filtering	basic	strong	basic	strong
Guide Tree	no	yes	no	yes
Release date	27.08.20	27.08.20	27.08.20	27.08.20
Aligned rRNA sequences	**9,469,124**	510,508	**1,312,534**	95,286

silva
high quality ribosomal RNA databases

STEP 10 : ASSIGN TAXONOMY

But when you download it directly on SILVA website, it looks like this :

```
>AY846379.1.1791 Eukaryota;Archaeplastida;Chloroplastida;Chlorophyta;Chlorophyceae;Sphaeropleales;
AACCUGGUUGAUCCUGCCAGUAGUCAUAUGCUUGUCUCAAAGAUUAAGCCAUGCAUGUCUAAGUAUAAACUGCUUAUACU
GUGAAACUGCGAAUGGCUCAUUAAAUCAGUUAUAGUUUAUUUGAUGGUACCUCUACACGGAUAACCGUAGUAAUUCUAGA
GCUAAUACGUGCGUAAAUCCCGACUUCUGGAAGGGACGUAUUUAUUAGAUAAAAGGCCGACCGAGCUUUGCUCGACCCGC
GGUGAAUCAUGAUAACUUCACGAAUCGCAUAGCCUUGUGCUGGCGAUGUUUCAUUCAAAUUUCUGCCCUAUCAACUUUCG
AUGGUAGGAUAGAGGCCUACCAUGGUGGUAACGGGUGACGGAGGAUUAGGGUUCGAUUCCGGAGAGGGAGCCUGAGAAAC
GGCUACCACAUCCAAGGAAGGCAGCAGGCGCGCAAAUUACCCAAUCCUGAUACGGGGAGGUAGUGACAAUAAAUAACAAU
GCCGGGCAUUUCAUGUCUGGCAAUUGGAAUGAGUACAAUCUAAAUCCCUUAACGAGGAUCAAUUGGAGGGCAAGUCUGGU
GCCAGCAGCCGCGGUAAUUCCAGCUCCAAUAGCGUAUAUUUAAGUUGUUGCAGUUAAAAAGCUCGUAGUUGGAUUUCGGG
UGGGUUCCAGCGGUCCGCCUAUGGUGAGUACUGCUGUGGCCCUCCUUUUUGUCGGGGACGGGCUCCUGGGCUUCAUUGUC
CGGGACUCGGAGUCGACGAUGAUACUUUGAGUAAAUUAGAGUGUUCAAAGCAAGCCUACGCUCUGAAUACUUUAGCAUGG
AAUAUCGCGAUAGGACUCUGGCCUAUCUCGUUGGUCUGUAGGACCGGAGUAAUGAUUAAGAGGGACAGUCGGGGGCAUUC
GUAUUUCAUUGUCAGAGGUGAAAUUCUUGGAUUUAUGAAAGACGAACUACUGCGAAAGCAUUUGCCAAGGAUGUUUUCAU
UAAUCAAGAACGAAAGUUGGGGGCUCGAAGACGAUUAGAUACCGUCGUAGUCUCAACCAUAAACGAUGCCGACUAGGGAU
UGGAGGAUGUUCUUUUGAUGACUUCUCCAGCACCUUAUGAGAAAUCAAAGUUUUUGGGUUCCGGGGGGAGUAUGGUCGCA
AGGCUGAAACUUAAAGGAAUUGACGGAAGGGCACCACCAGGCGUGGAGCCUGCGGCUUAAUUUGACUCAACACGGGAAAA
CUUACCAGGUCCAGACAUAGUGAGGAUUGACAGAUUGAGAGCUCUUUCUUGAUUCUAUGGGUGGUGGUGCAUGGCCGUUC
UUAGUUGGUGGGUUGCCUUGUCAGGUUGAUUCCGGUAACGAACGAGACCUCAGCCUGCUAAAUAUGUCACAUUCGCUUUU
UGCGGAUGGCCGACUUCUUAGAGGGACUAUUGGCGUUUAGUCAAUGGAAGUAUGAGGCAAUAACAGGUCUGUGAUGCCCU
UAGAUGUUCUGGGCCGCACGCGCGCUACACUGACGCAUUCAGCAAGCCUAUCCUUGACCGAGAGGUCUGGGUAAUCUUUG
AAACUGCGUCGUGAUGGGGAUAGAUUAUUGCAAUUAUUAGUCUUCAACGAGGAAUGCCUAGUAAGCGCAAGUCAUCAGCU
UGCGUUGAUUACGUCCCUGCCCUUUGUACACACCGCCCGUCGCUCCUACCGAUUGGGUGUGCUGGUGAAGUGUUCGGAUU
GGCAGAGCGGGUGGCAACACUUGCUUUUGCCGAGAAGUUCAUUAAACCCUCCCACCUAGAGGAAGGAGAAGUCGUAACAA
GGUUUCCGUAGGUGAACCUGCAGAAGGAUCA
>AB001445.1.1538 Bacteria;Proteobacteria;Gammaproteobacteria;Pseudomonadales;Pseudomonadaceae;Pseu
AACUGAAGAGUUUGAUCAUGGCUCAGAUUGAACGCUGGCGGCAGGCCUAACACAUGCAAGUCGAGCGGCAGCACGGGUAC
UUGUACCUGGUGGCGAGCGGCGGACGGGUGAGUAAUGCCUAGGAAUCUGCCUGGUAGUGGGGGAUAACGCUCGGAAACGG
ACGCUAAUACCGCAUACGUCCUACGGGAGAAAGCAGGGGACCUUCGGGCCUUGCGCUAUCAGAUGAGCCUAGGUCGGAUU
AGCUAGUUGGUGAGGUAAUGGCUCACCAAGGCGACGAUCCGUAACUGGUCUGAGAGGAUGAUCAGUCACACUGGAACUGA
GACACGGUCCAGACUCCUACGGGAGGCAGCAGUGGGGAAUAUUGGACAAUGGGCGAAAGCCUGAUCCAGCCAUGCCGCGU
GUGUGAAGAAGGUCUUCGGAUUGUAAAGCACUUUAAGUUGGGAGGAAGGGCAGUUACCUAAUACGUAUCUGUUUUGACGU
UACCGACAGAAUAAGCACCGGCUAACUCUGUGCCAGCAGCCGCGGUAAUACAGAGGGUGCAAGCGUUAAUCGGAAUUACU
GGGCGUAAAGCGCGCGUAGGUGGUUUGUUAAGUUGAAUGUGAAAUCCCCGGGCUCAACCUGGGAACUGCAUCCAAAACUG
```

Not suitable for DADA2 assignTaxonomy() function !

STEP 10 : ASSIGN TAXONOMY

Hopefully, DADA2 team maintain different formatted database, including SILVA :

DADA2-formatted reference databases

We maintain reference fastas for the three most common 16S databases: Silva, RDP and GreenGenes. The dada2 package recognizes and parses the General Fasta releases of the UNITE project for ITS taxonomic assignment. Formatted versions of other databases can be "contributed" and will be made available through this page if referencable by doi (eg. deposited at Zenodo or Figshare).

Please note that the files provided here are just derivative reformattings of these taxonomic databases. If using these files for taxonomic assignment, the source database should also be cited.

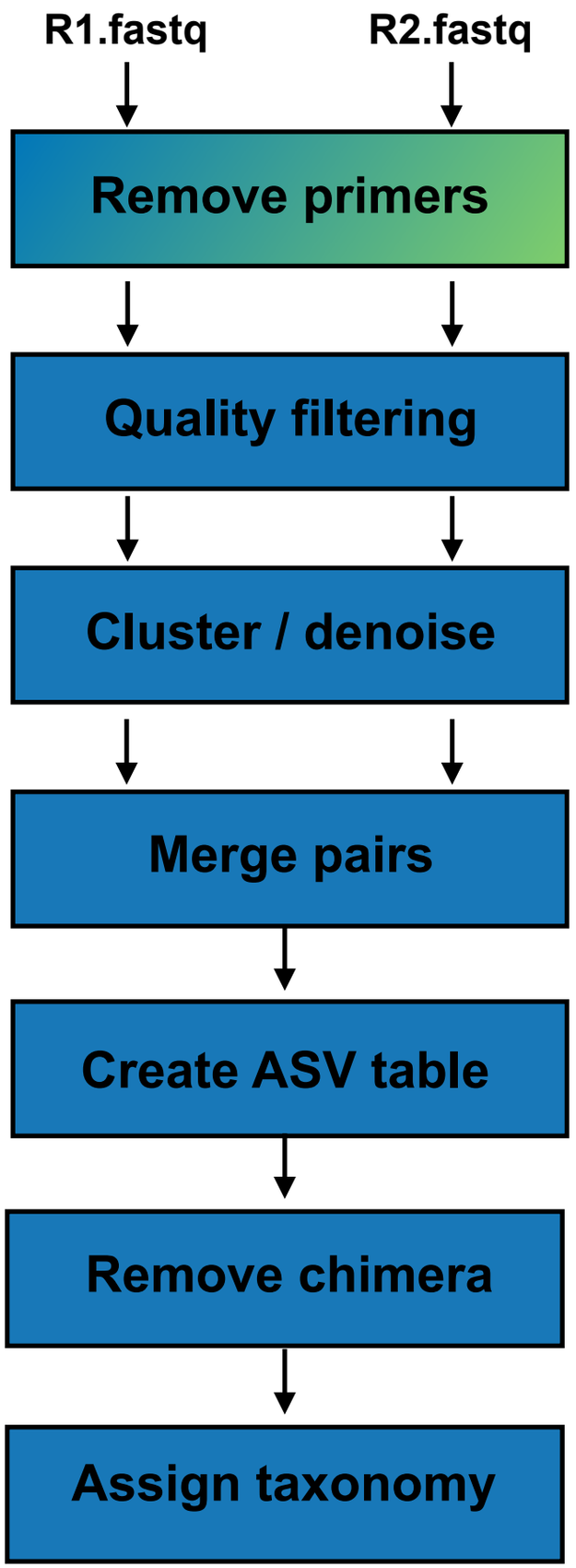
Maintained:

- **Silva version 138.1 - UPDATED Mar 10, 2021**, version 132, version 128, version 123
  - NOTE: As of Silva version 138, the official DADA2-formatted reference fastas are optimized for classification of Bacteria and Archaea, and are not suitable for classifying Eukaryotes.
- **RDP trainset 18**, RDP trainset 16, RDP trainset 14
- **UNITE (use the General Fasta releases, "All eukaryotes")**
- *Deprecated: GreenGenes version 13.8 (the source GreenGenes database is no longer being maintained)*

Contributed:

- **GTDB Version 202: Genome Taxonomy Database** (More info on GTDB)
  - Version 86 for `assignTaxonomy` and `assignSpecies`
- RefSeq + RDP (NCBI RefSeq 16S rRNA database supplemented by RDP)
  - Reference files formatted for `assignTaxonomy`
  - Reference files formatted for `assignSpecies`
- HitDB version 1 (Human InTestinal 16S rRNA)
- Human Oral Microbiome Database: HOMD
- MiDAS: Field Guide to the Microbes of Activated Sludge and Anaerobic Digesters
- MIDORI Reference 2 (for taxonomic assignments of Eukaryota mitochondrial DNA sequences)
- RDP fungi LSU trainset 11

DADA2 pipeline

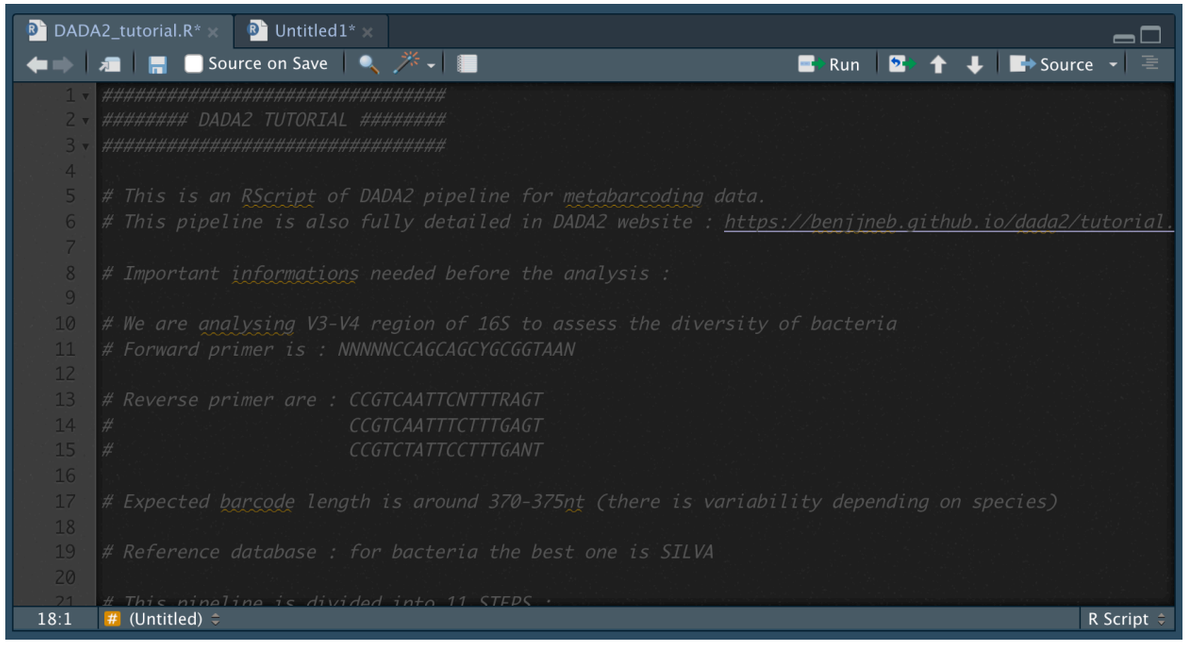


ANY QUESTIONS ?

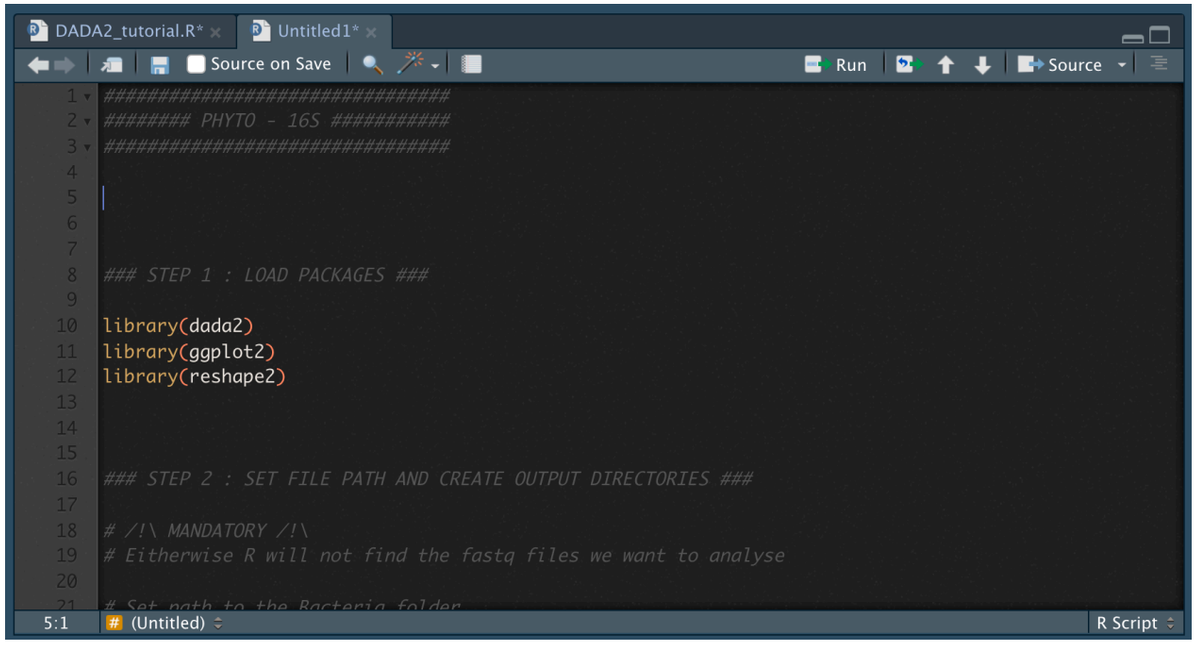
FOR TOMORROW

You will analyse **3** real datasets by group of **3**

For this you will need to adapt the DADA2 tutorial script to this new dataset (primers, path, reference database,…)

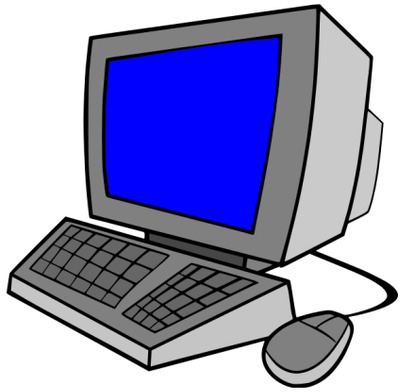


DADA2_tutorial.R

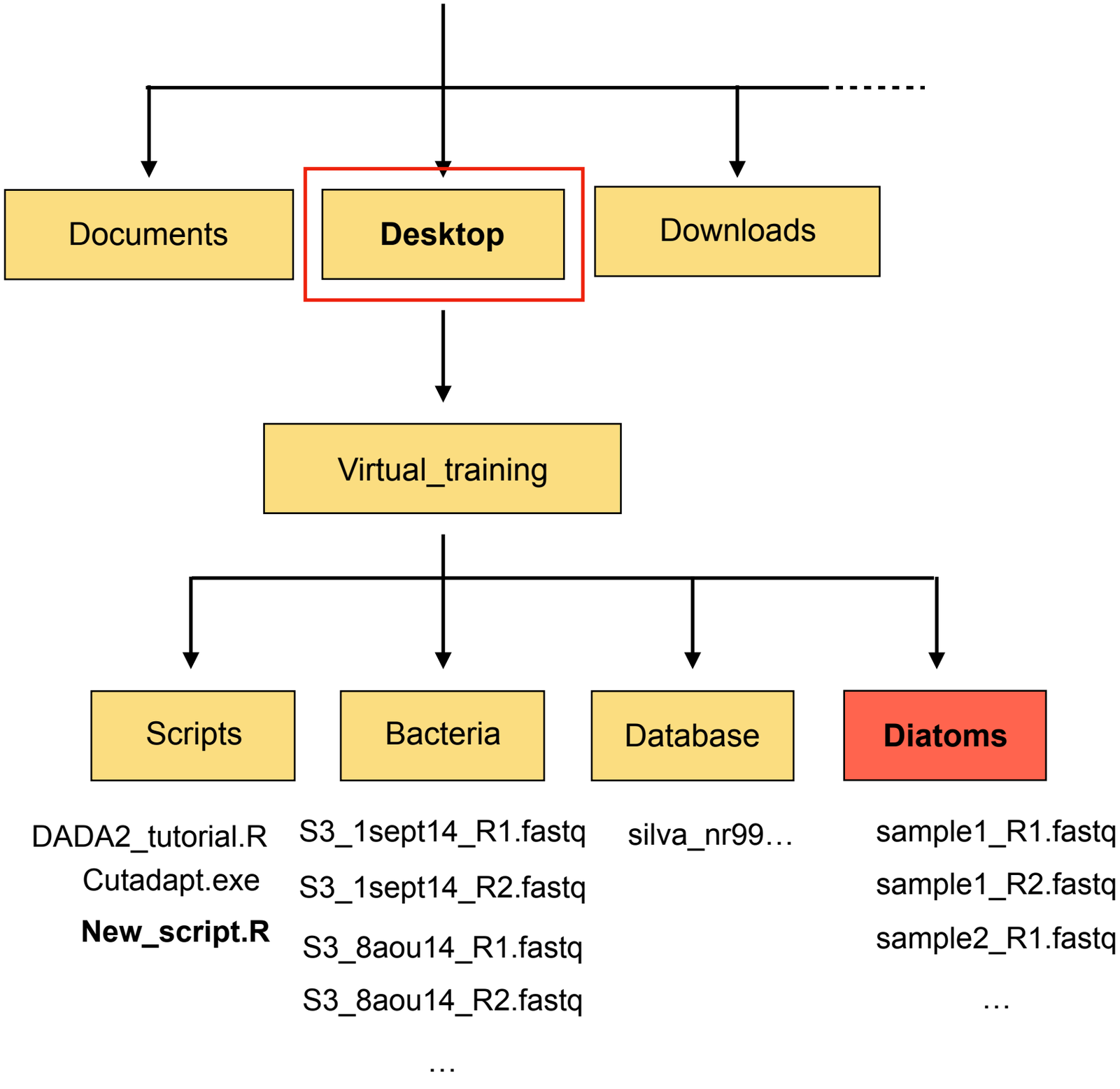


New_script.R

Copy



FOR TOMORROW



FOR TOMORROW

On DAY4 and DAY5 you will present the results for everyone :

Quick presentation of the dataset and targeted barcode

What were the difficulties you met at each step?

What did you need to change in the script?

How many reads did you loose at each step? Is it correct?

What is the database you used?

Any other comments

FOR TOMORROW

There won’t be any zoom.

But if you have any difficulties, please use discord !

We will be connected and answer your questions as much as possible