

**Hello!**

**Welcome to 16S amplicon workshop playground!**

When you are ready,  
check for today's code-along materials on GitHub:

[https://github.com/jinnyyang/16S-  
Analysis-Playground/tree/main/  
pipeline\\_QIIME2-DADA2](https://github.com/jinnyyang/16S-Analysis-Playground/tree/main/pipeline_QIIME2-DADA2)

# Hi, welcome to 16S amplicon workshop playground!

Code-along materials on GitHub:

[https://github.com/jinnyyang/16S-Analysis-Playground/tree/main/pipeline\\_QIIME2-DADA2](https://github.com/jinnyyang/16S-Analysis-Playground/tree/main/pipeline_QIIME2-DADA2)

HPCC dashboard:

<https://ondemand.hpcc.msu.edu/pun/sys/dashboard/>

## **1. Install QIIME2 on HPCC**

Follow “[QIIME2 Installation ScreenShots.docx](#)” and “[QIIME2 Installation script.txt](#)”

## **2. Upload example materials to HPCC and start a job** (will take ~01:25 to run)

1. Copy materials to your own directory:

A. If you have access to Lebeis lab, copy material with: `cp /mnt/research/LebeisLab/Yang/16SAmpliconWorkshop_20250819.zip .`

B. Or download “16SAmpliconWorkshop\_20250819.zip” from my OneDrive and upload to your own directory: [16SAmpliconWorkshop\\_20250819.zip](#)

2. Unzip with: `unzip 16SAmpliconWorkshop_20250819.zip`

3. Enter the file with: `cd 16SAmpliconWorkshop_20250819/HPCC`

4. Start your job: `sbatch cutadapt_Qiime2_DATA2_blastn.sh`

### 3. Get ready with R

Code-along materials on GitHub:

[https://github.com/jinnyyang/16S-Analysis-Playground/  
tree/main/pipeline\\_QIIME2-DADA2](https://github.com/jinnyyang/16S-Analysis-Playground/tree/main/pipeline_QIIME2-DADA2)

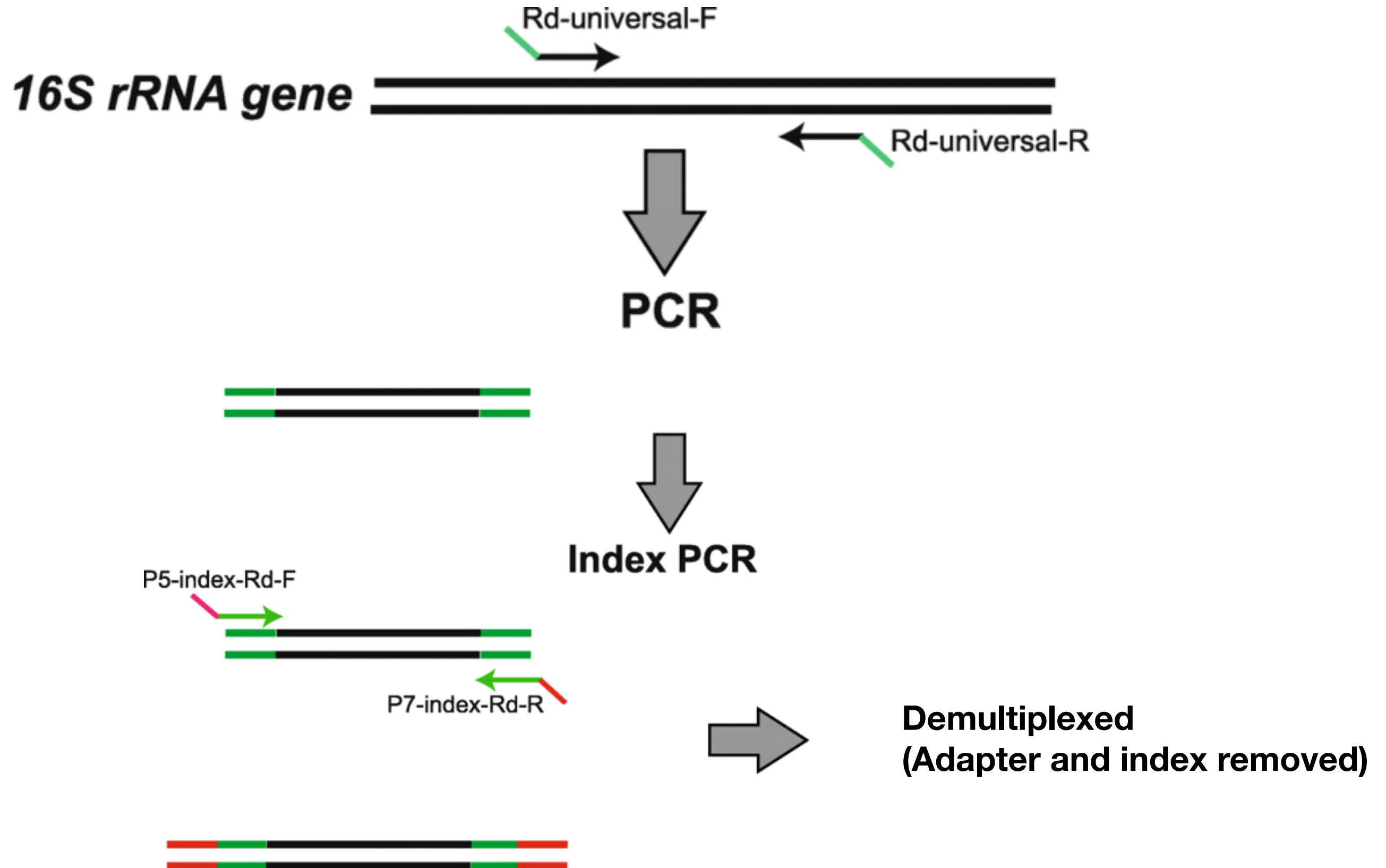
#### 1. Install R and R Studio

2. Download file “R” from

16SAmpliconWorkshop\_20250819/R




3. Open R studio and install R  
packages

# Working on 16S Amplicon reads

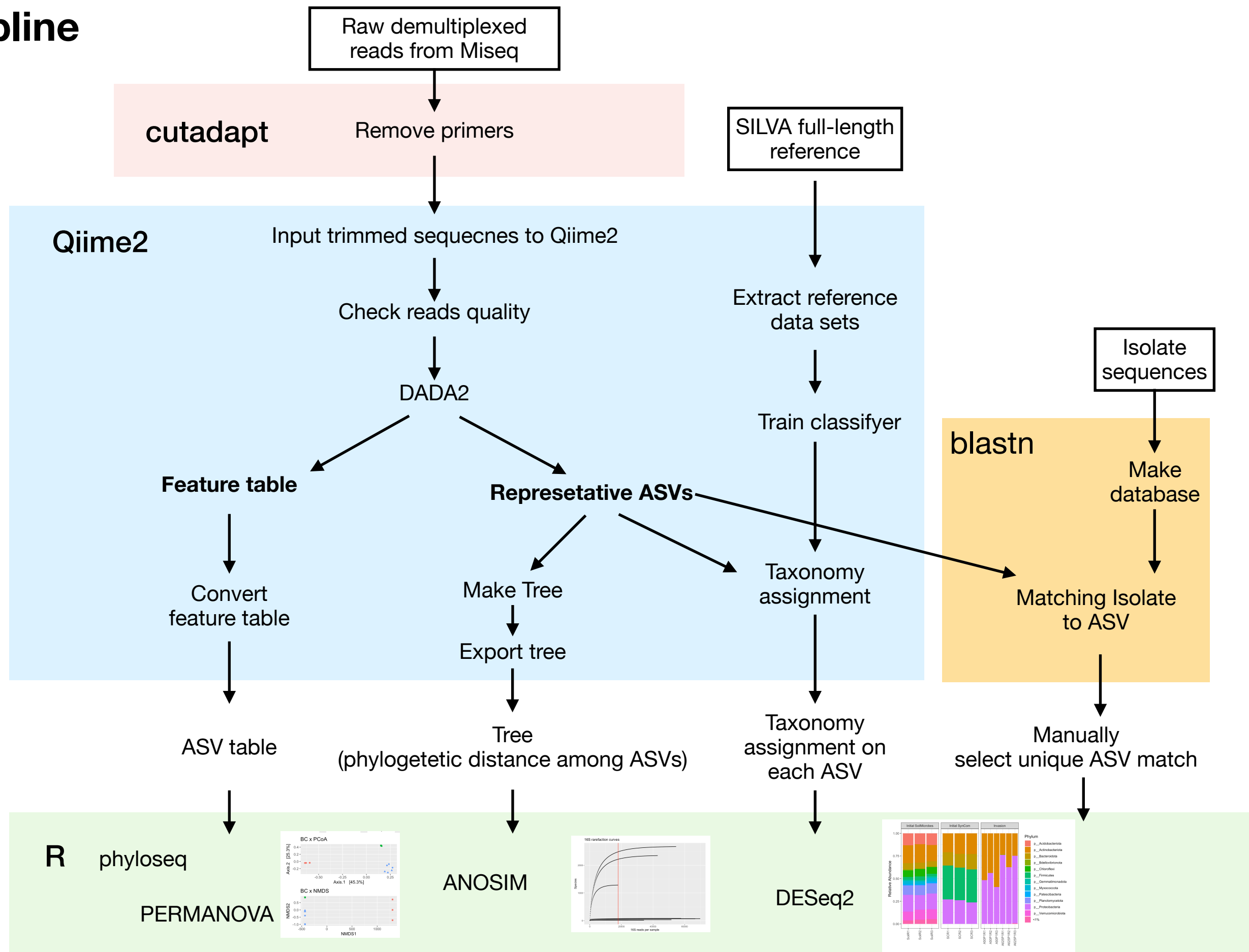


Starting from demultiplexed reads  
(Adapter and index removed)



-  A625P1R3\_S3\_L001\_R1\_001.fastq.gz
-  A625P1R3\_S3\_L001\_R2\_001.fastq.gz
-  SCR1\_S193\_L001\_R1\_001.fastq.gz
-  SCR1\_S193\_L001\_R2\_001.fastq.gz
-  SCR2\_S194\_L001\_R1\_001.fastq.gz
-  SCR2\_S194\_L001\_R2\_001.fastq.gz
-  SCR3\_S195\_L001\_R1\_001.fastq.gz
-  SCR3\_S195\_L001\_R2\_001.fastq.gz
-  SoilR1\_S189\_L001\_R1\_001.fastq.gz
-  SoilR1\_S189\_L001\_R2\_001.fastq.gz

# Pipeline



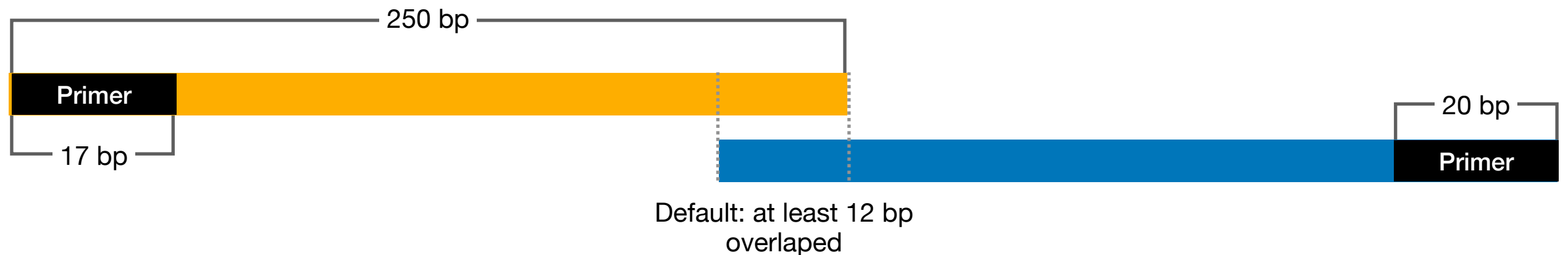
Let's open "cutadapt\_Qiime2\_DATA2.sh" and read with this flow

# Where to truncate?

## 1. Target length

## 2. Reads quality

- Our examples:
  - 341F and 806R (16S rRNA gene V3-V4 region)
  - Target length = ~465 bp
  - Miseq 250 x 2 pair-end

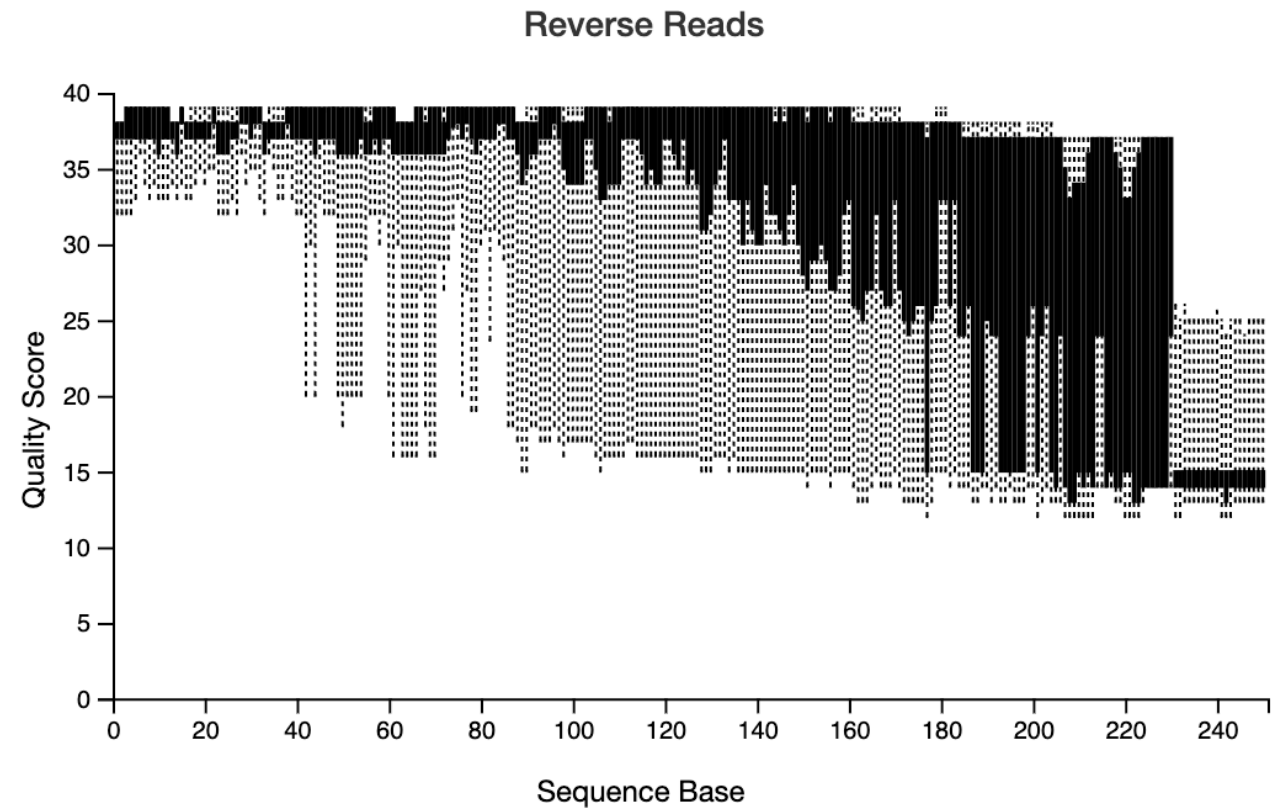
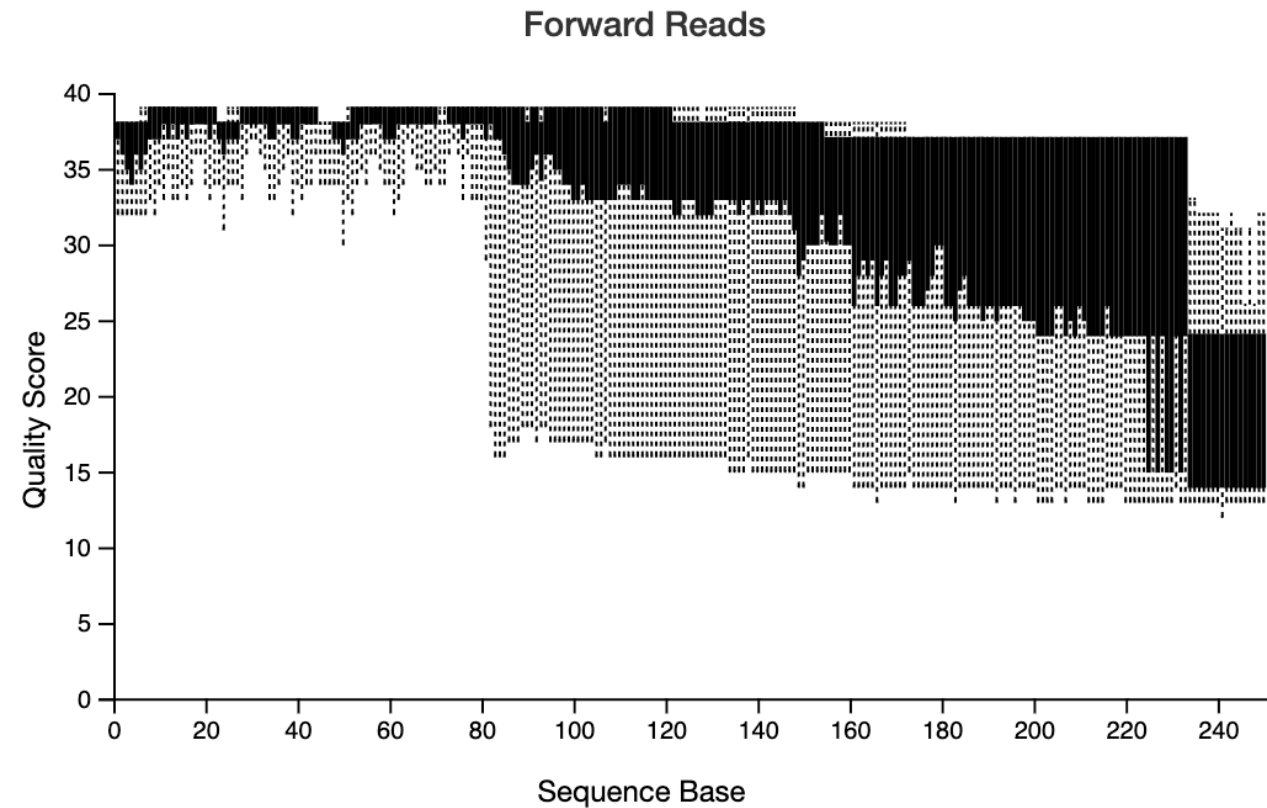


**Need at least  $465 - 17 - 20 + 12 \times 2 = 452$  bp = 226 bp for each**

**In this case: truncate at 230 bp!**

# Where to truncate?

1. Target length
2. Reads quality



I compromise for quality and let DADA2 to correct errors and/or filter them out...

This figure was from “paired-end-demux.qzv”  
Visualized by QIIME 2 View: <https://view.qiime2.org/>  
Or download and unzip it



From “denoising-stats.qza” file

	sample-id	input	filtered	percentage of input passed filter	denoised	merged	percentage of input merged	non-chimeric	percentage of input non-chimeric
	#q2:types	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric
1	A50P1R1	59553	35615	59.8	35520	35238	59.17	34153	57.35
2	A50P1R2	36925	21926	59.38	21841	21492	58.2	20760	56.22
3	A50P1R3	45169	25413	56.26	25340	24971	55.28	24241	53.67
4	A625P1R1	39757	25043	62.99	24908	24290	61.1	22984	57.81
5	A625P1R2	85747	55907	65.2	55639	54559	63.63	51571	60.14
6	A625P1R3	51963	27895	53.68	27737	27256	52.45	26063	50.16
7	SCR1	96077	80623	83.91	79832	76570	79.7	69775	72.62
8	SCR2	82850	68497	82.68	67749	64524	77.88	57641	69.57
9	SCR3	93013	76590	82.34	75884	72929	78.41	66600	71.6
10	SoilR1	84917	61263	72.14	57018	46388	54.63	43282	50.97
11	SoilR2	39229	28224	71.95	25566	19710	50.24	18122	46.2
12	SoilR3	102287	74623	72.95	70191	58578	57.27	55049	53.82
13									

Next:  
 Shorter target region or  
 Use 300 bp x 2 cycle (AVITI system)

# DADA 2: Divisive Amplicon Denoising Algorithm 2

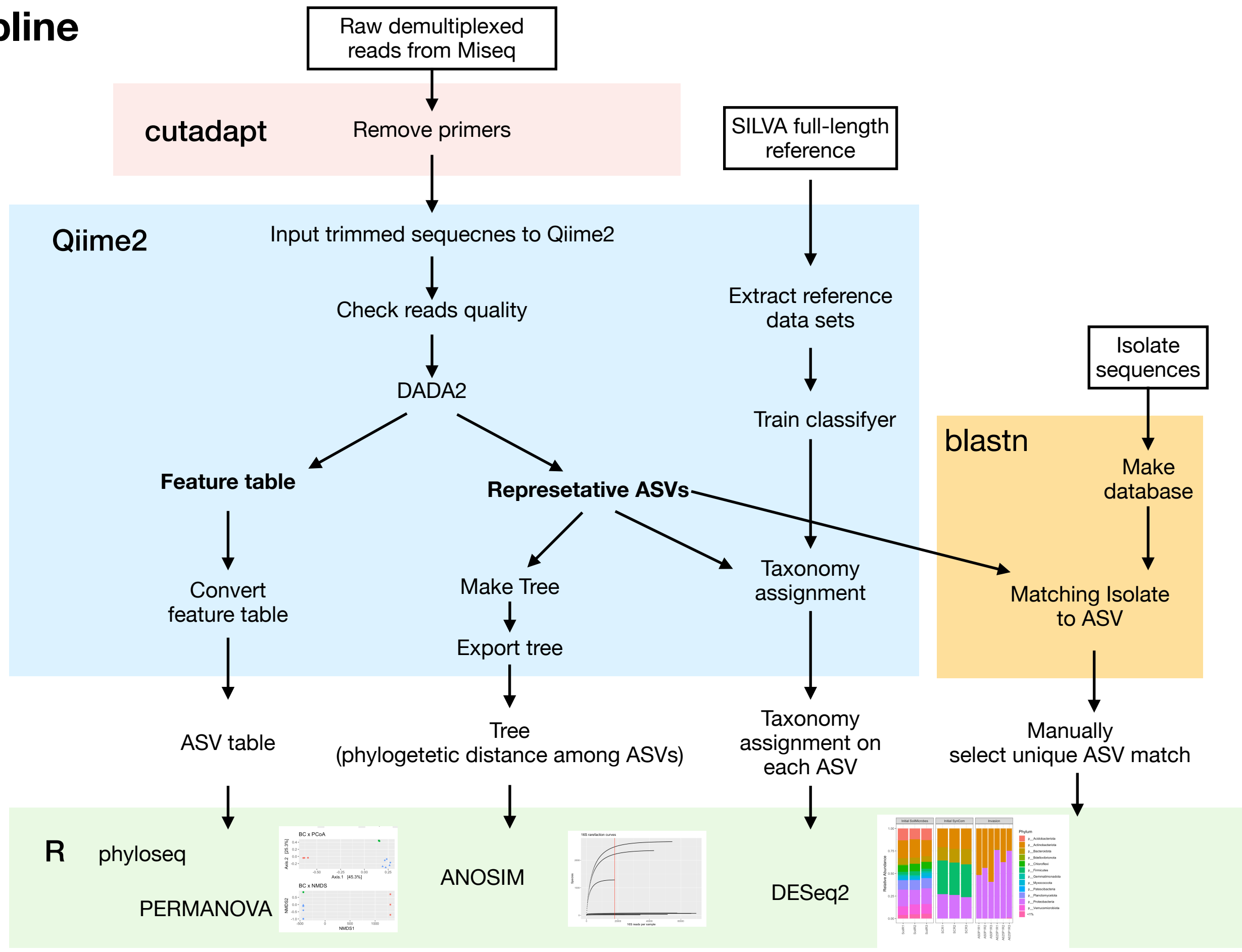
Why ASV (amplicon sequence variant) but OTU (operational taxonomic unit)?

1. Trimming and Quality Filtering Dereplication (group the same reads).
2. **Error rate modeling** (QC score + mismatch from “parent reads”).
3. **Sequence inference** (denoising).
4. Merging paired-end reads.
5. Chimera checking (compare with each parent reads, consensus/pooled mode).
6. Output generation.

## **Assumptions:**

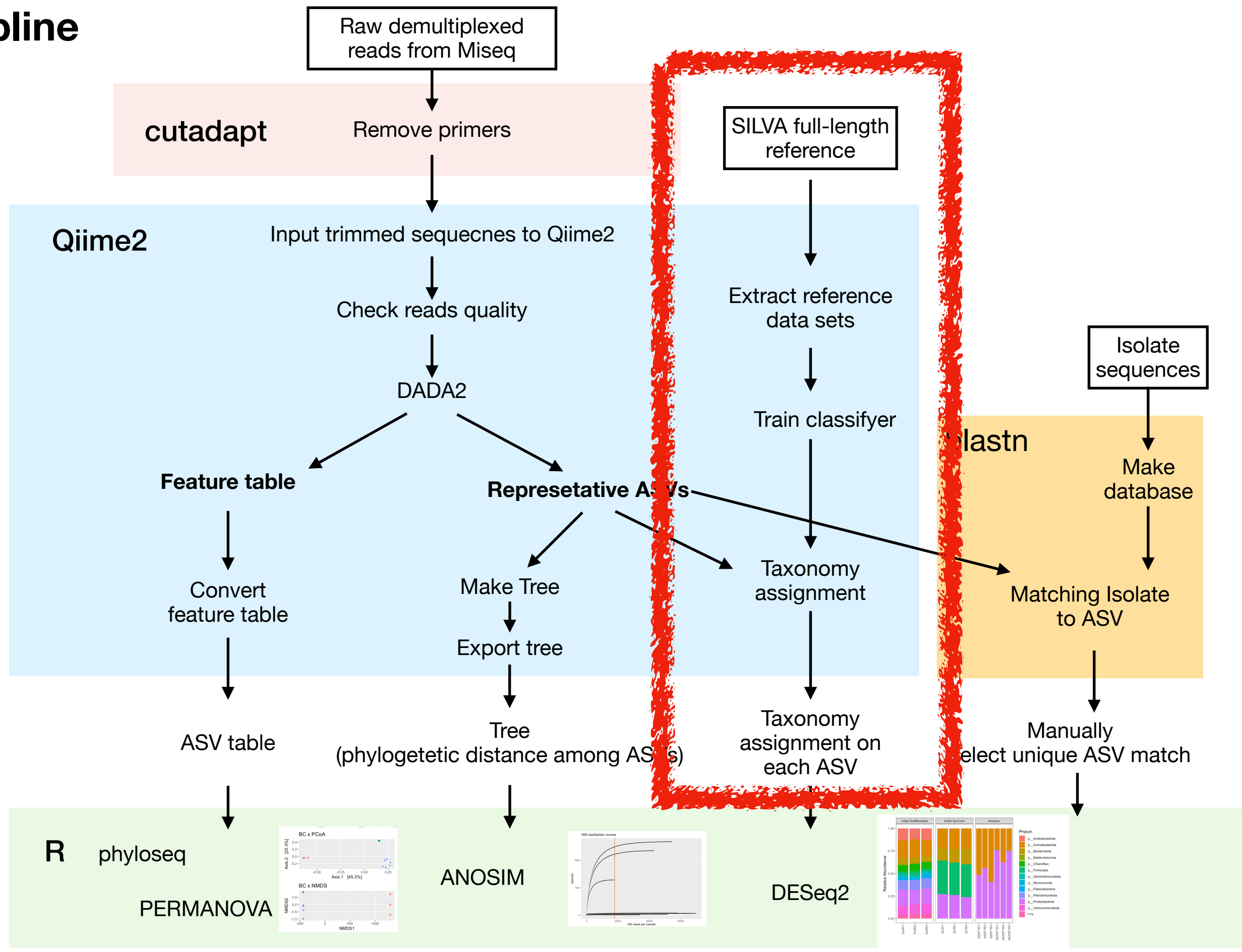
- True biological sequences are present multiple times
- Sequencing errors are reproducible and can be statistically modeled

# Pipeline



Let's open "cutadapt\_Qiime2\_DATA2.sh" and read with this flow

# Pipeline



Let's open "cutadapt\_Qiime2\_DATA2.sh" and read with this flow

# Taxonomy assignment — Train the Classifier

Download SILVA 138 sequences and taxonomy:  
<https://docs.qiime2.org/2024.10/data-resources/>

In file “classifier.sh”

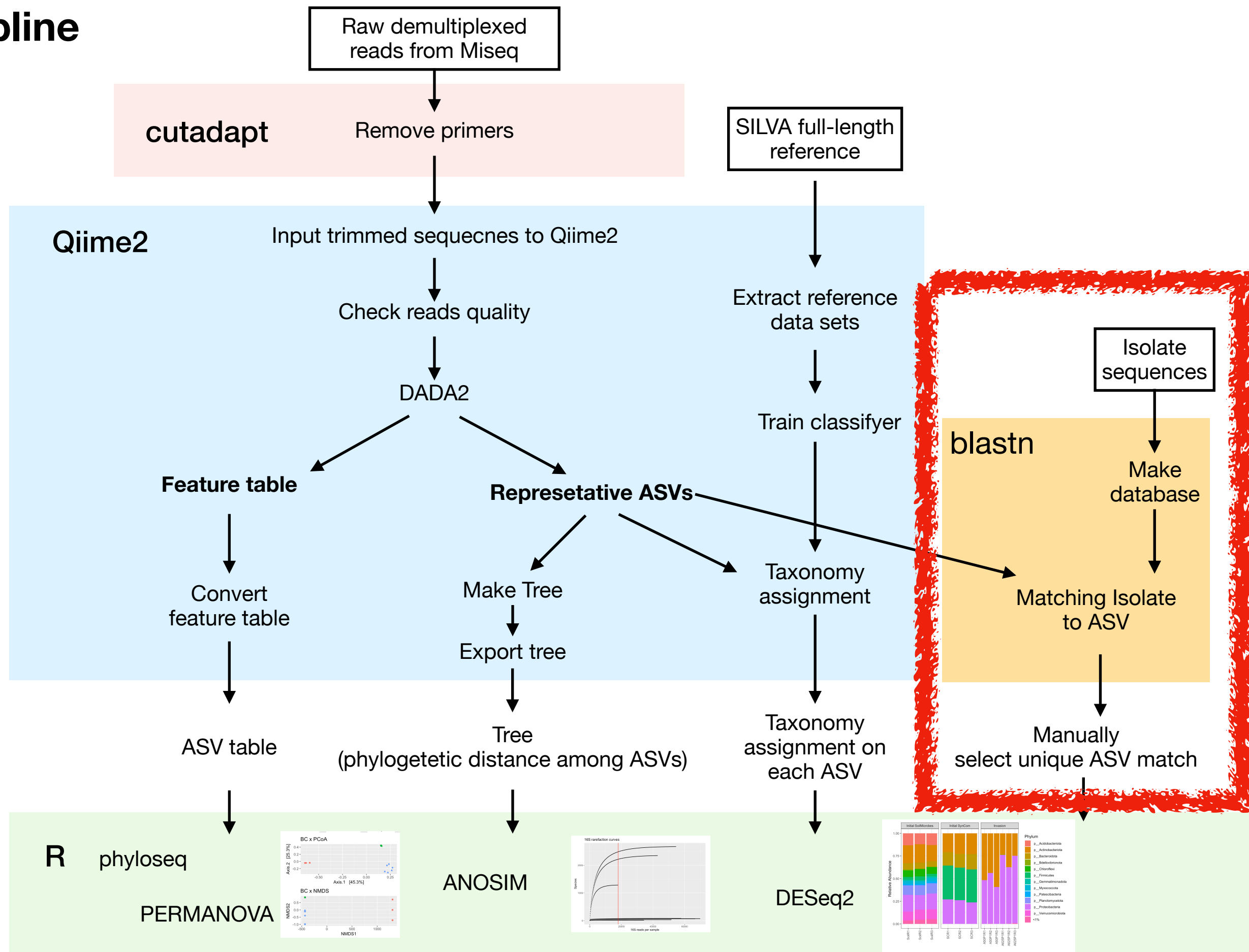
```
qiime feature-classifier extract-reads \  
  --i-sequences silva-138-99-seqs.qza \  
  --p-f-primer CCTACGGGNGGCWGCAG \  
  --p-r-primer GGACTACHVGGGTATCTAAT \  
  --p-trunc-len 470 \  
  --p-min-length 200 \  
  --p-max-length 500 \  
  --o-reads ref-seqs-341-806.qza
```

**Extract reference reads**

```
qiime feature-classifier fit-classifier-naive-bayes \  
  --i-reference-reads ref-seqs-341-806.qza \  
  --i-reference-taxonomy silva-138-99-tax.qza \  
  --o-classifier classifier-341-806.qza
```

**Train classifier**

# Pipeline

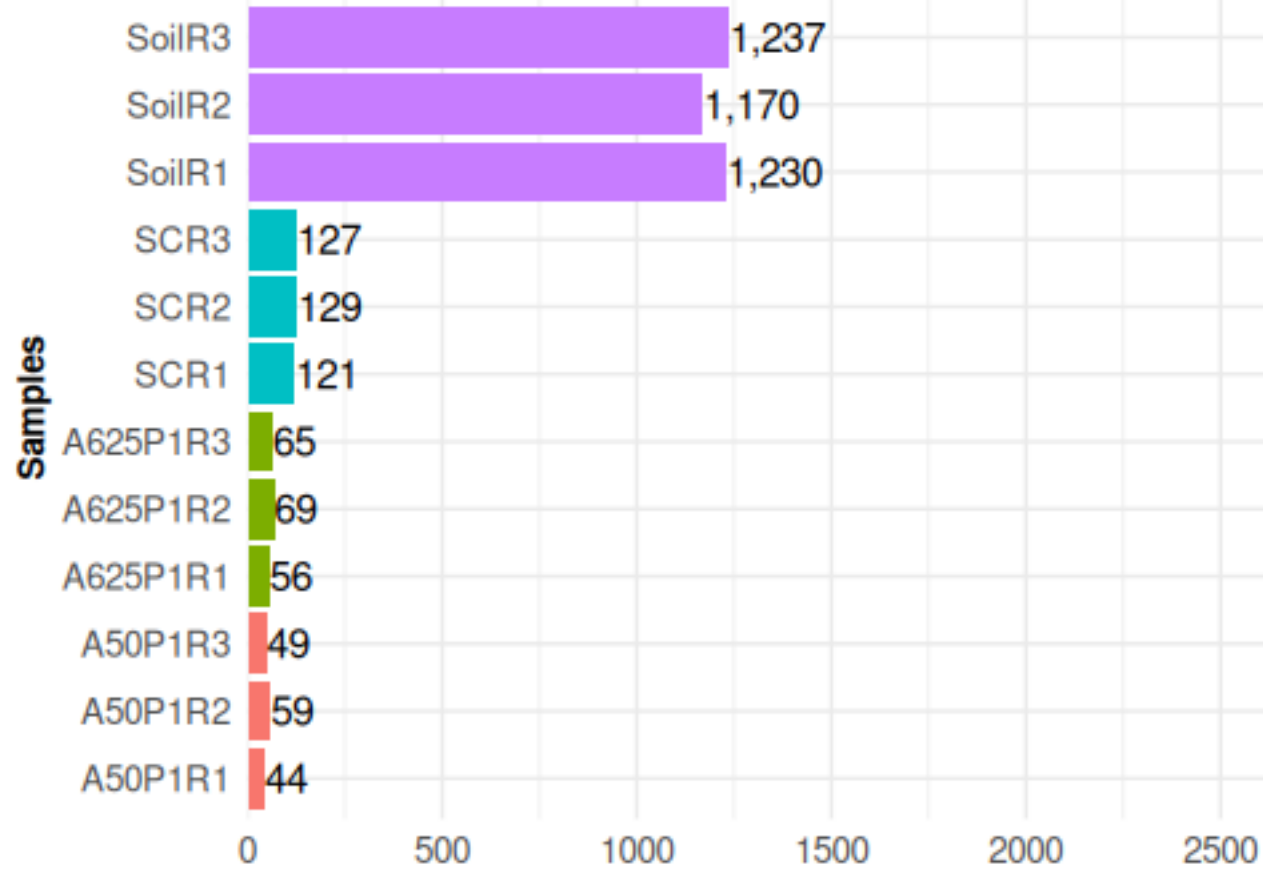


Let's open "cutadapt\_QIIME2\_DATA2.sh" and read with this flow

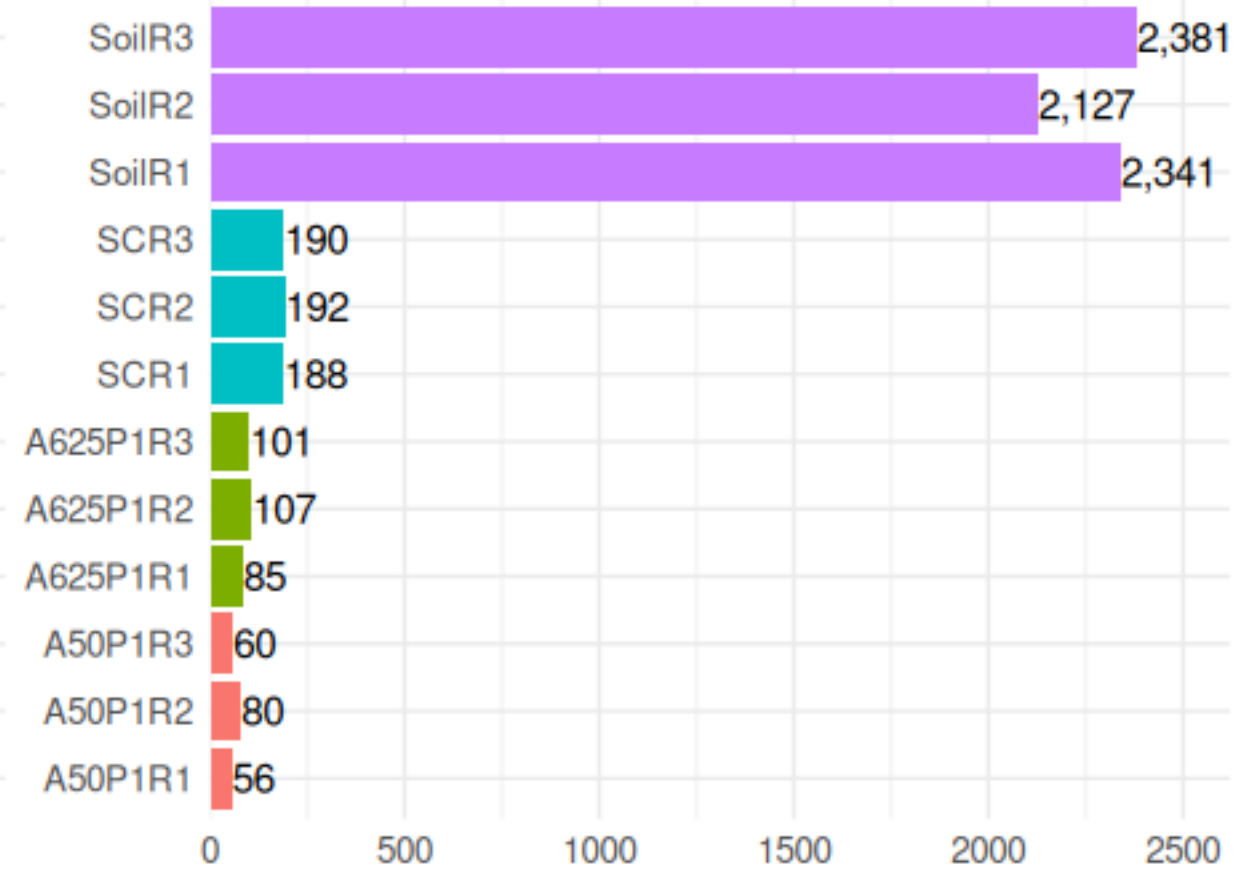
Try different pipelines give you options,  
and may save your sample!

# ASVs by Sample and Processing Strategy

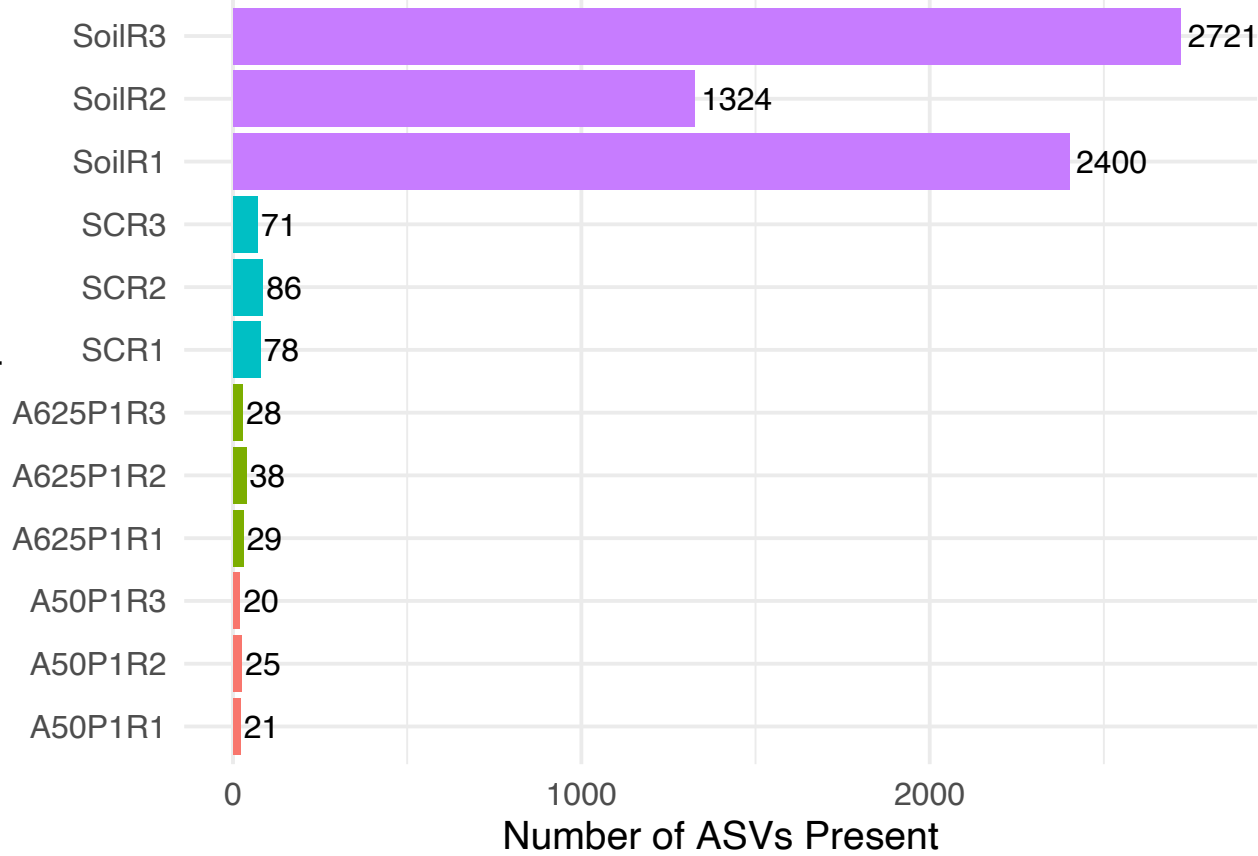
## ASVs (cutadapt)



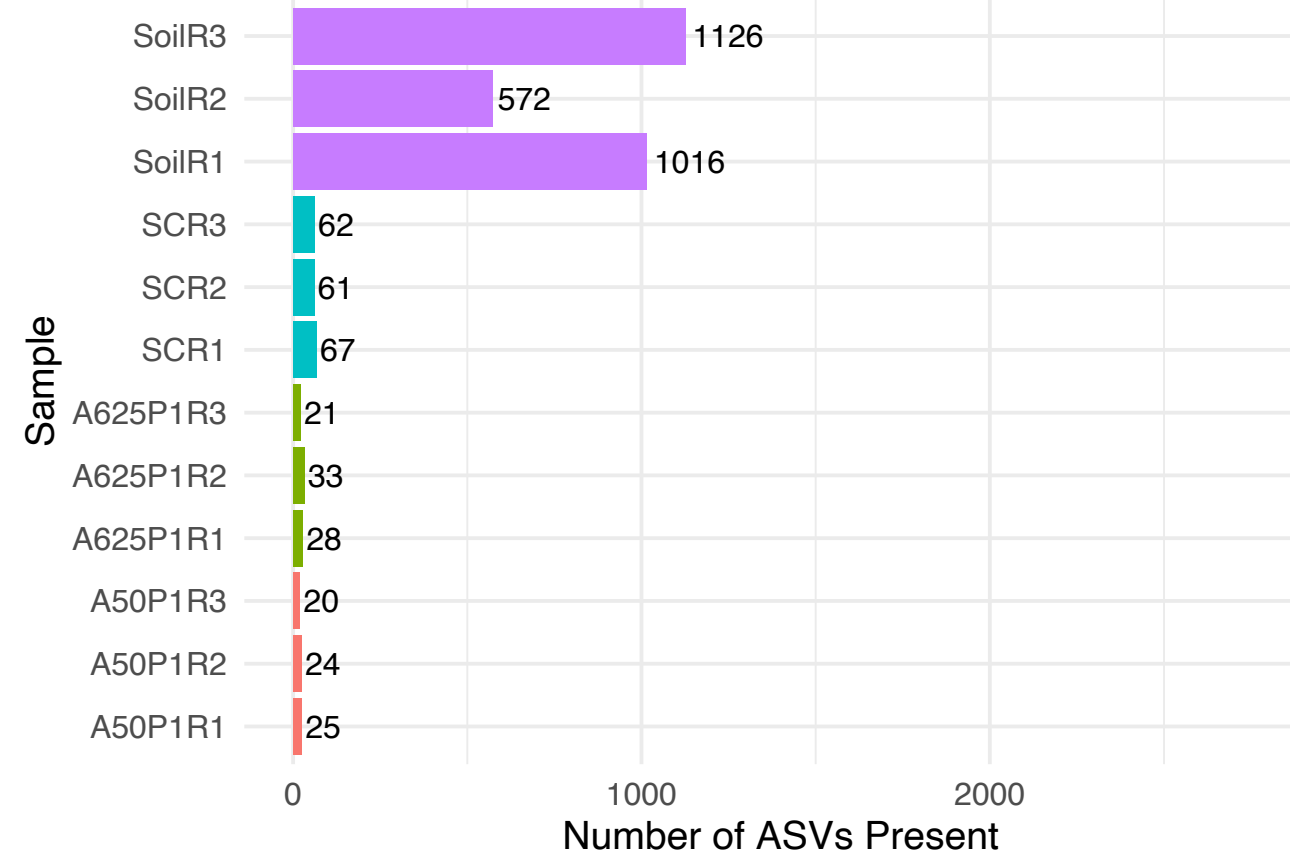
## ASVs (trimEnds)



## cutadapt + DADA2



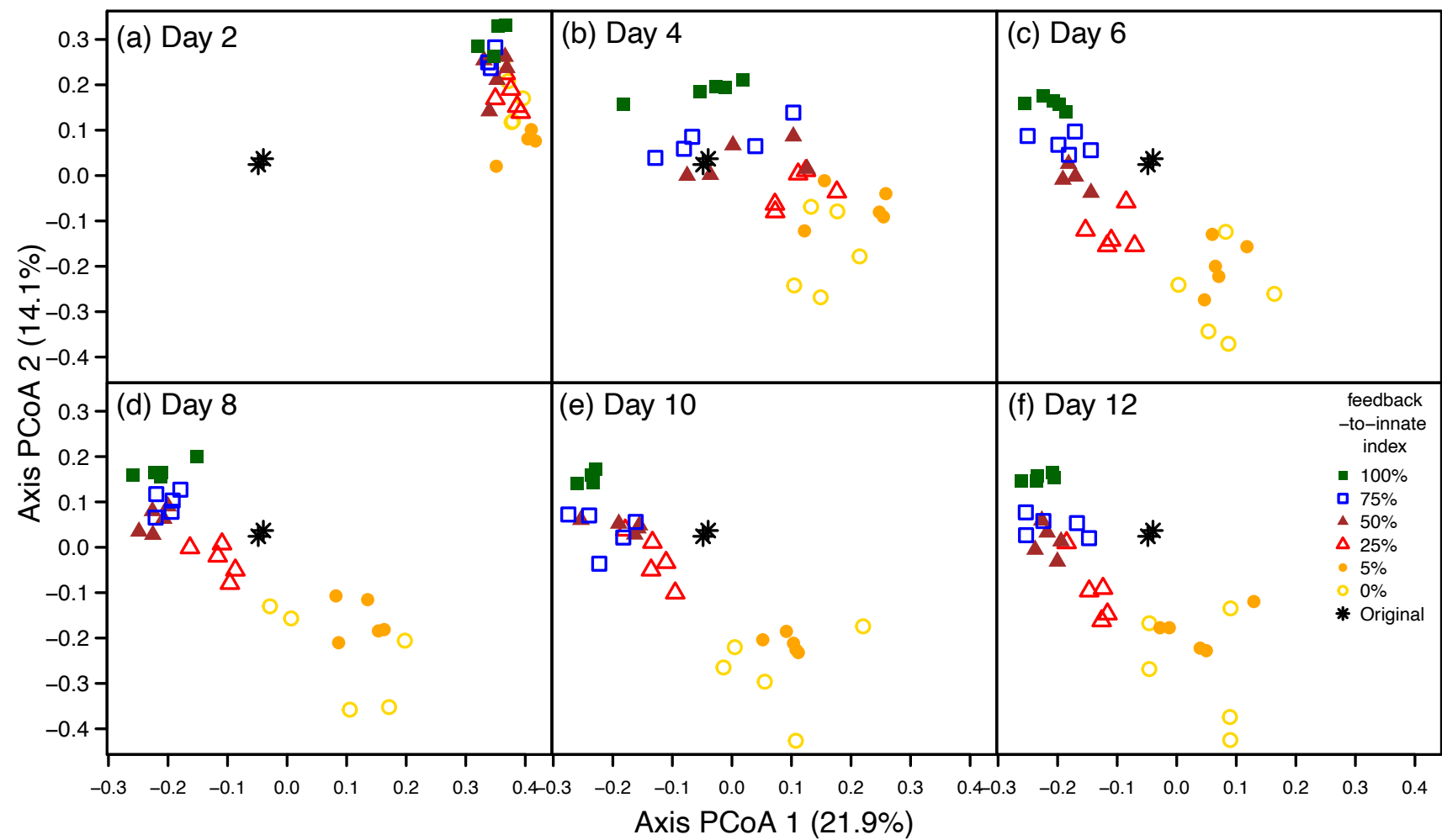
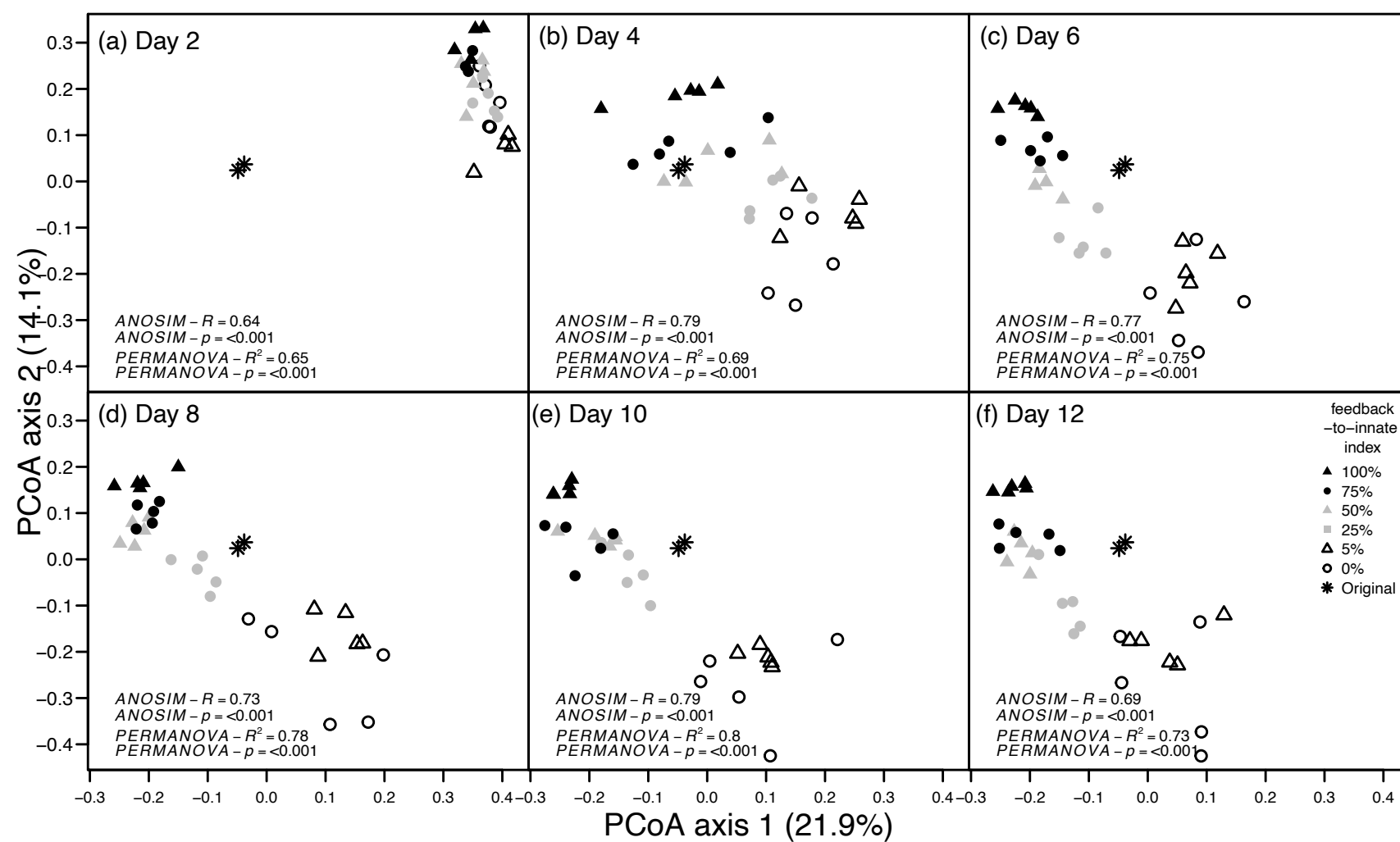
## DADA2 (same as trimEnds)



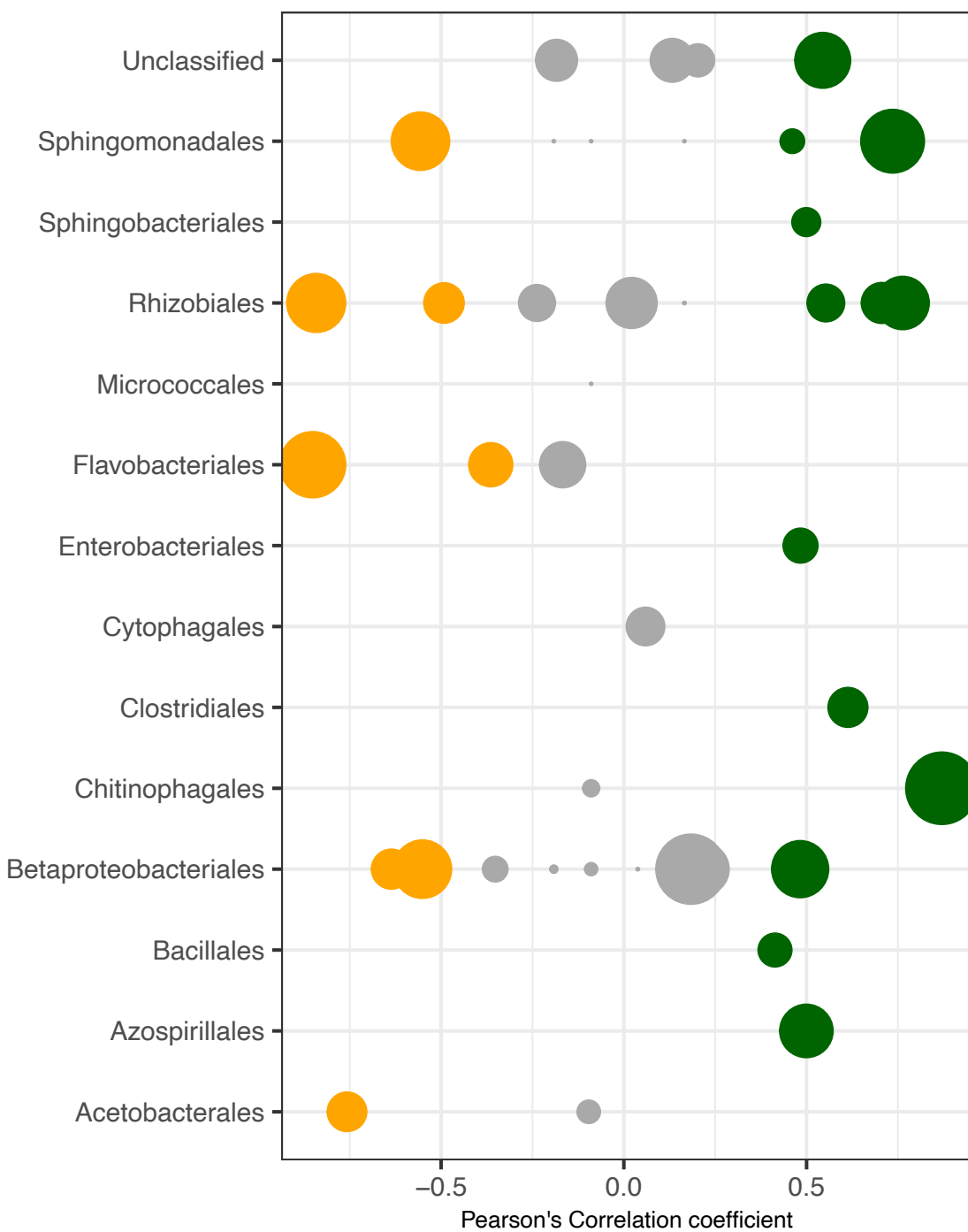


R

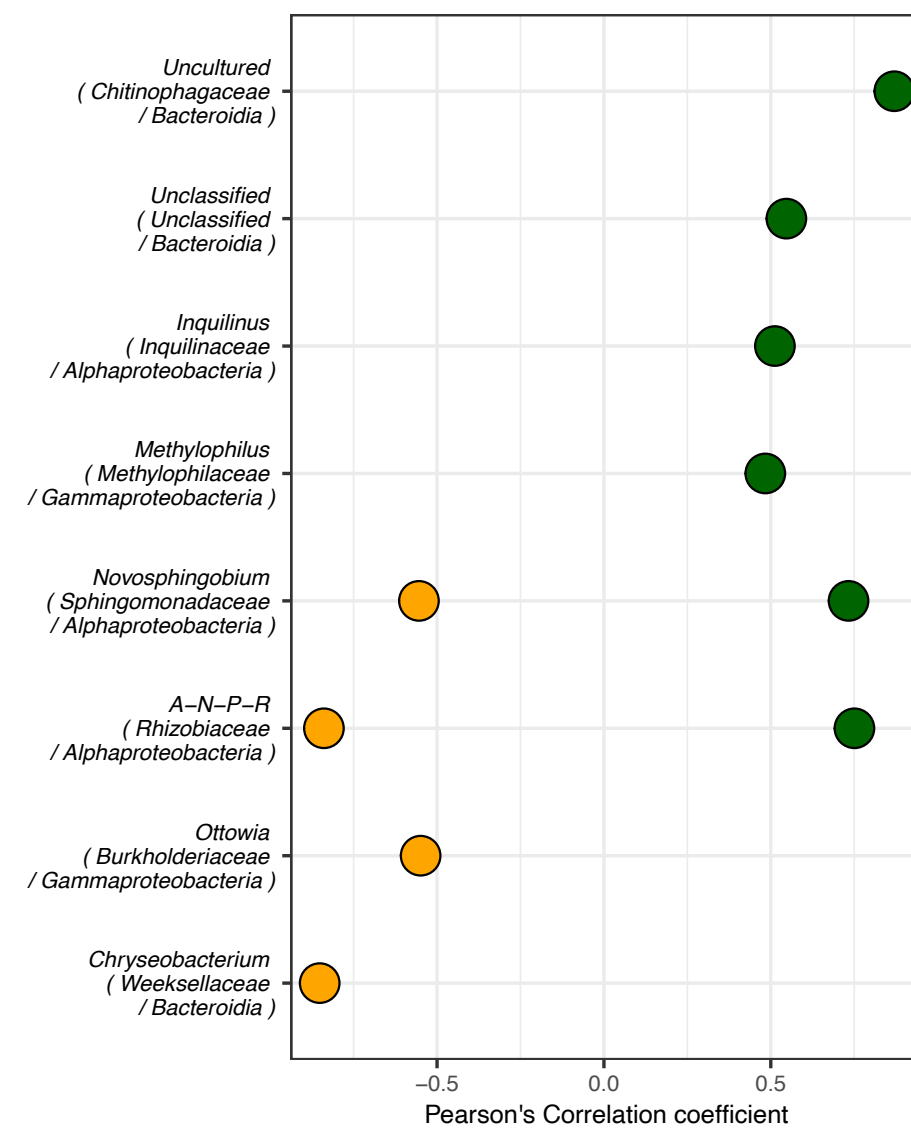
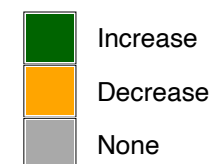
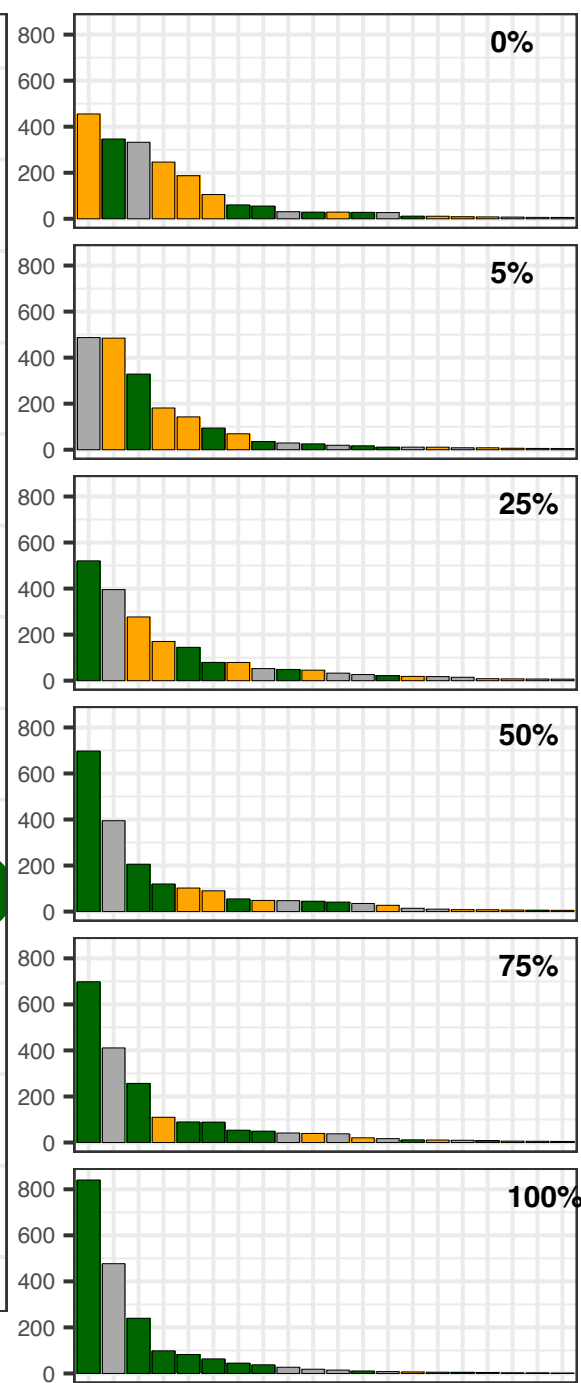
Time!



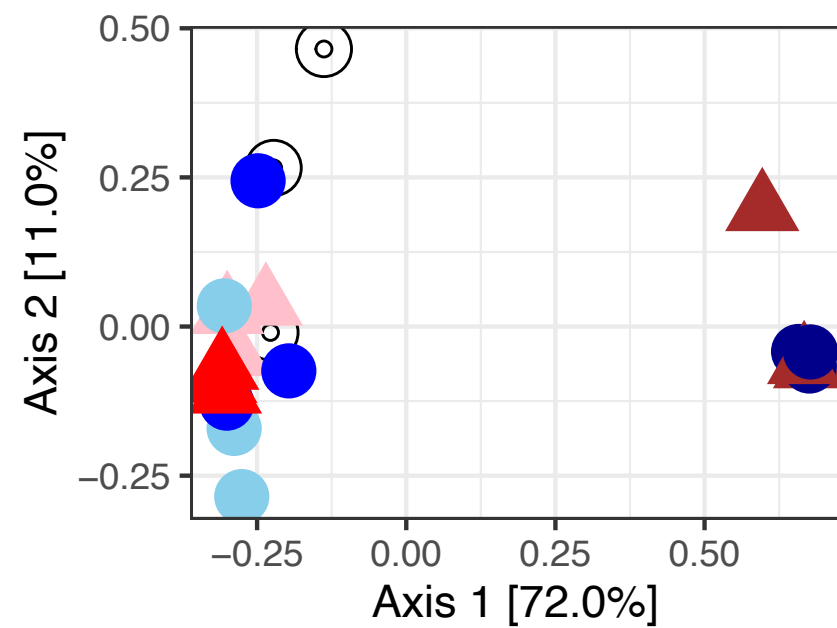
A



B



BioSTA\_2023



BioSTA\_2024

