ECOSYSTEM ECOLOGY

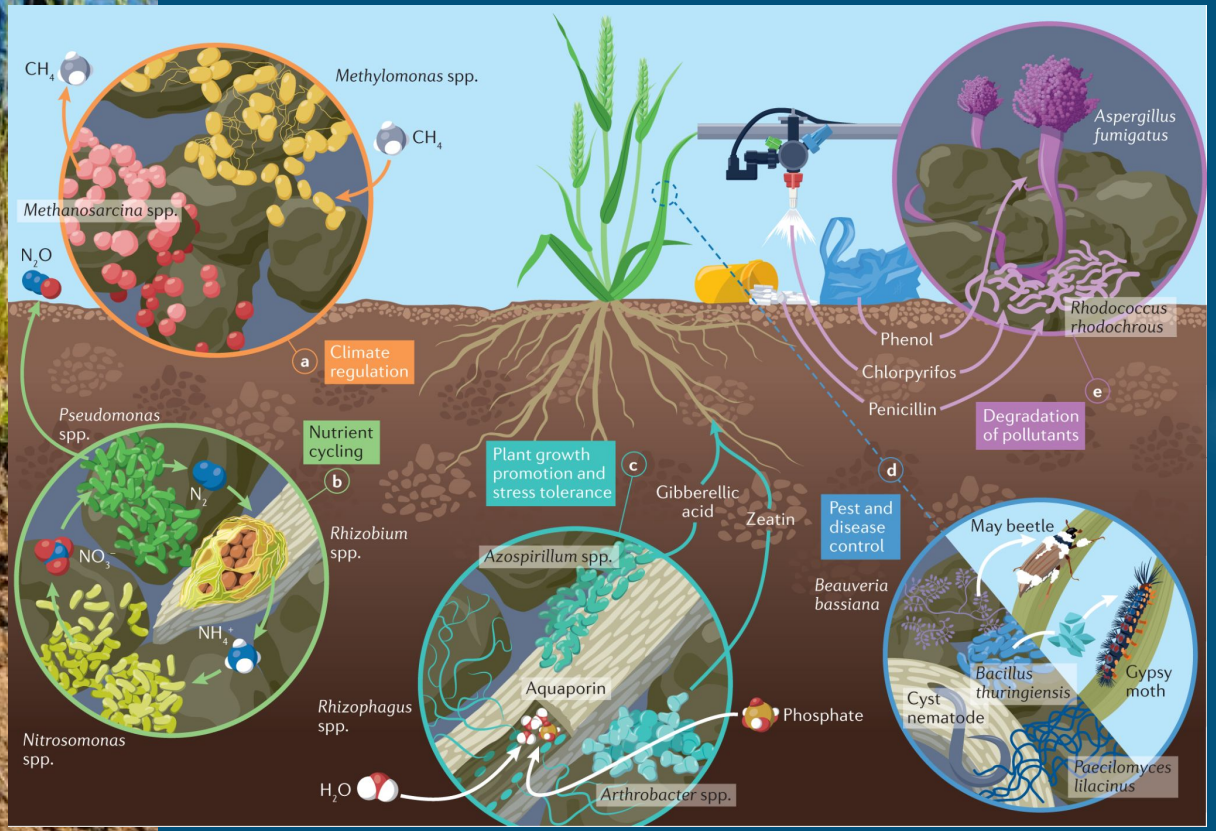
Bioinformatics Course

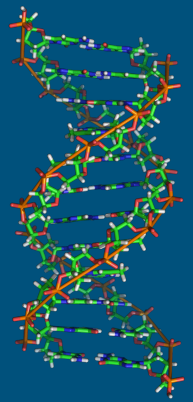
Session 1

Theory: Understanding the **metabarcoding** technology

Practice: Initial **processing** of **.fastq** reads

Ecosystem **functioning**





There is not one perfect way of doing things…

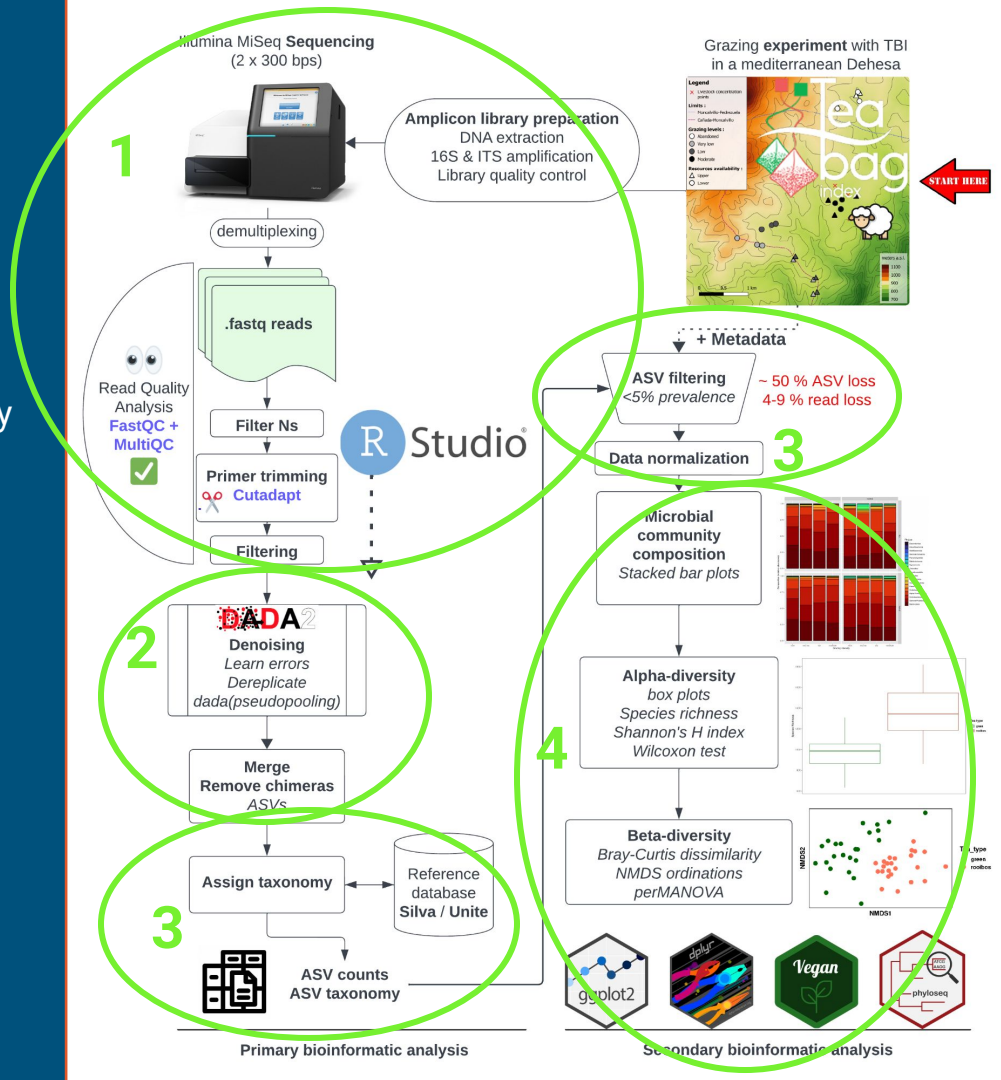
Course **plan**

Session 1
Theory: Understanding the **metabarcoding** technology
Practice: Initial processing of **.fastq** reads

Session 2
Theory: Denoising, **ASVs** vs **OTUs**
Practice: **DADA2** denoising algorithm

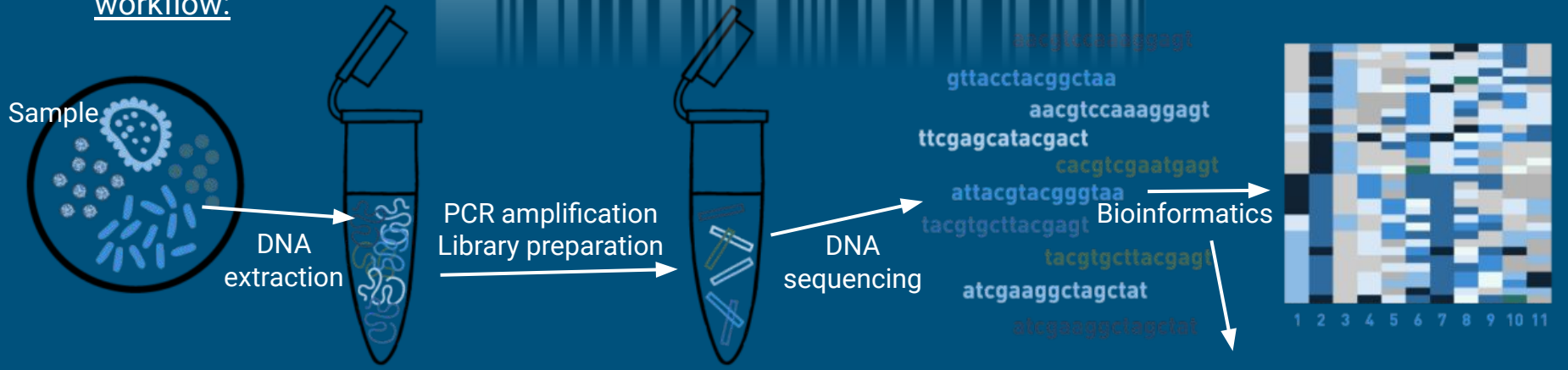
Session 3
Theory: metabarcoding **databases**
Practice: assign **taxonomy** & ASV **filtering**

Session 4
Theory: **downstream** analysis, microbial ecology
Practice:



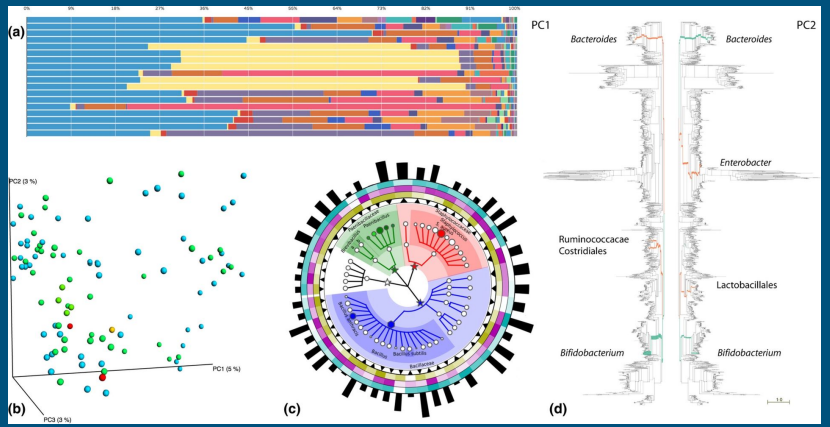
Metabarcoding

Typical & simplified workflow:

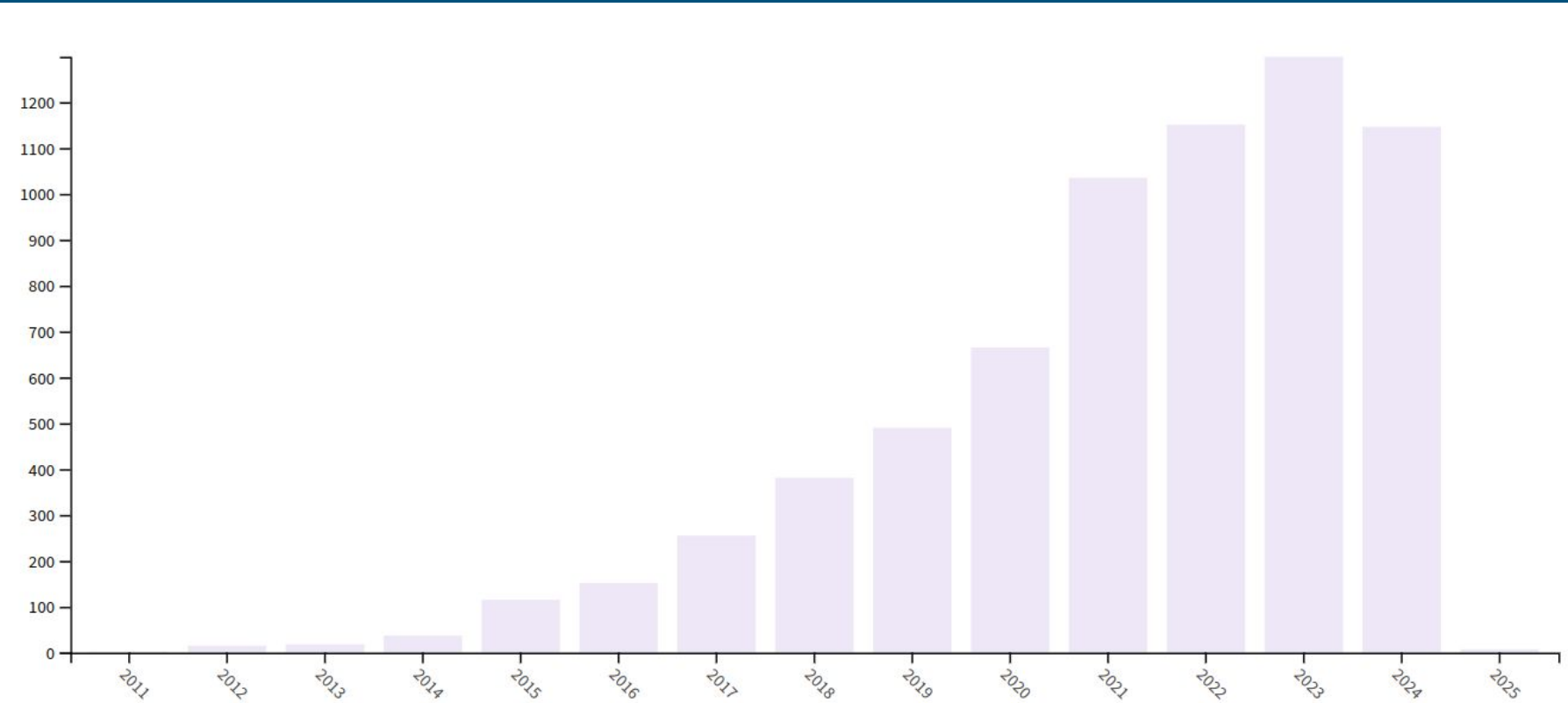


Also named:

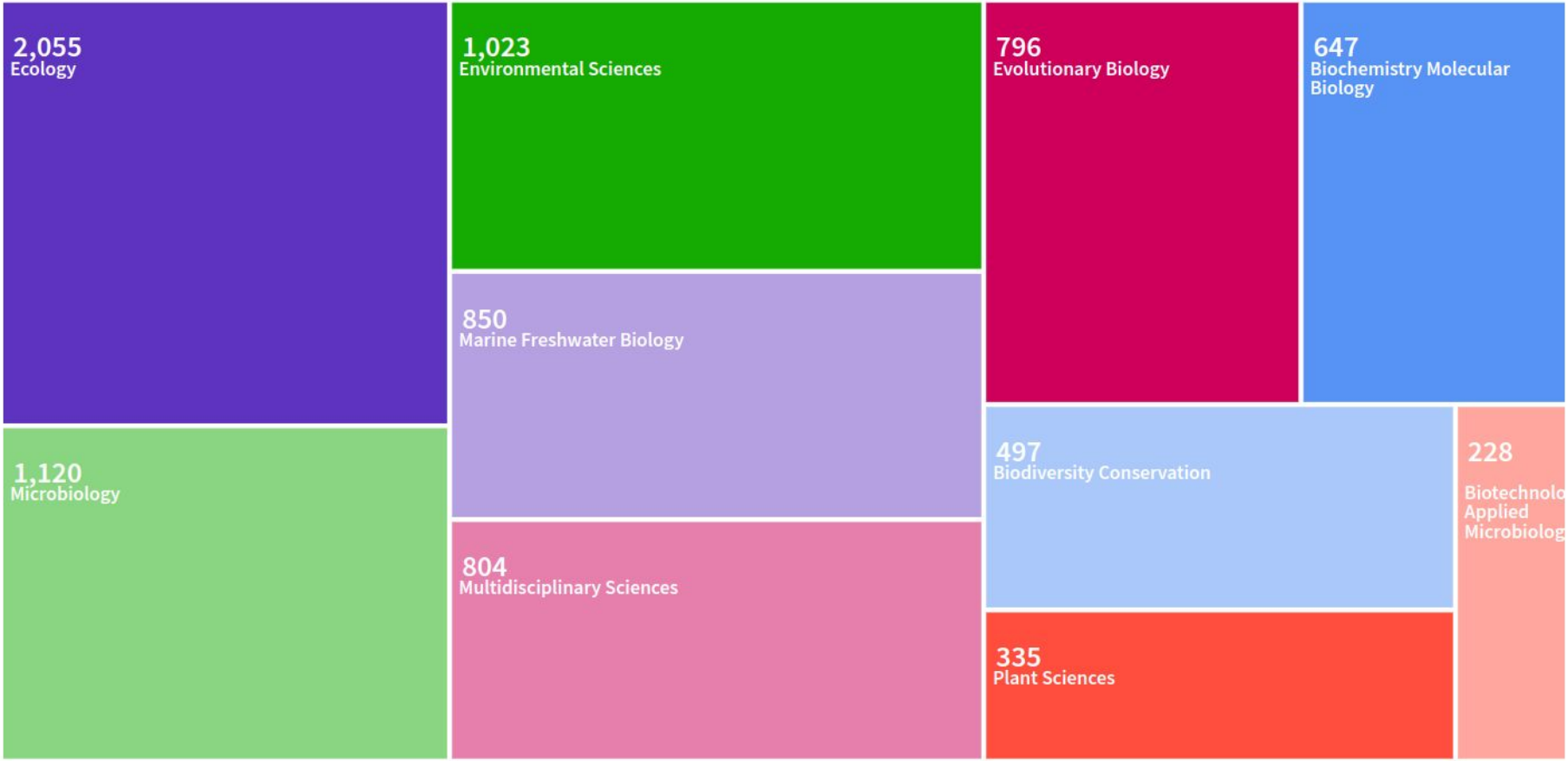
- Metataxonomy ✅
- eDNA 🤔
- Metagenomics 🤔
- Amplicon 🤔



Searching for “**metabarcoding**” in Web of Science…



Searching for “**metabarcoding**” in Web of Science…

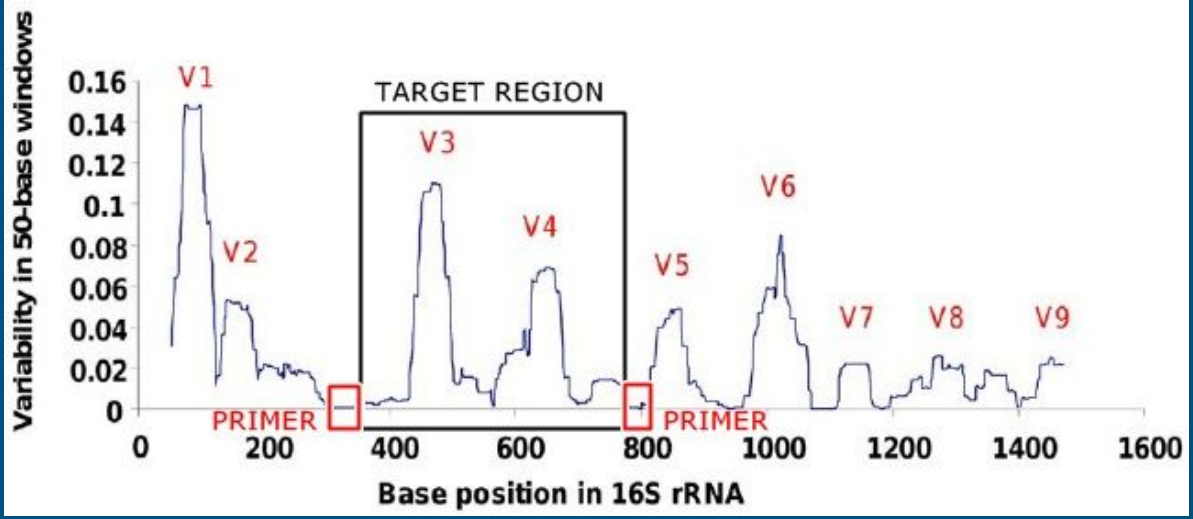


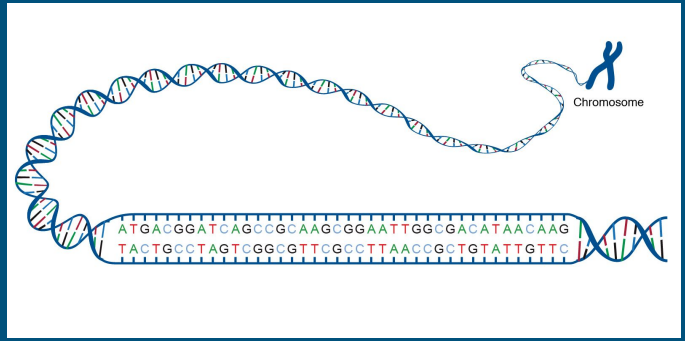
Metabarcoding

1. Barcodes

What would the **perfect** molecular **barcode** achieve?

"A good DNA barcode should be **ubiquitous**, have both **low intra-specific** and **high inter-specific variability** (high **taxonomic resolution**) and possess **conserved flanking sites** for developing universal PCR primers for wide taxonomic application. It should also, for the practical user, be reflected in large **reference databases**."





2nd generation sequencers

Platform	Use	Sequencing Technology	Amplification Type	Principle	Read Length (bp)	Limitations
454 pyrosequencing	Short read sequencing	Seq by synthesis	Emulsion PCR	Detection of pyrophosphate released during nucleotide incorporation.	400–1000	May contain deletion and insertion sequencing errors due to inefficient determination of homopolymer length.
Ion Torrent	Short read sequencing	Seq by synthesis	Emulsion PCR	Ion semiconductor sequencing principle detecting H^+ ion generated during nucleotide incorporation.	200–400	When homopolymer sequences are sequenced, it may lead to loss in signal strength.
Illumina	Short read sequencing	Seq by synthesis	Bridge PCR	Solid-phase sequencing on immobilized surface leveraging clonal array formation using proprietary reversible terminator technology for rapid and accurate large-scale sequencing using single labeled dNTPs, which is added to the nucleic acid chain.	36–300	In case of sample overloading, the sequencing may result in overcrowding or overlapping signals, thus spiking the error rate up to 1%.
SOLiD	Short read sequencing	Seq by ligation	Emulsion PCR	An enzymatic method of sequencing using DNA ligase. 8-Mer probes with a hydroxyl group at 3′ end and a fluorescent tag (unique to each base A, T, G, C) at 5′ end are used in ligation reaction.	75	This platform displays substitution errors and may also under-represent GC-rich regions. Their short reads also limit their wider applications.

3rd generation sequencers

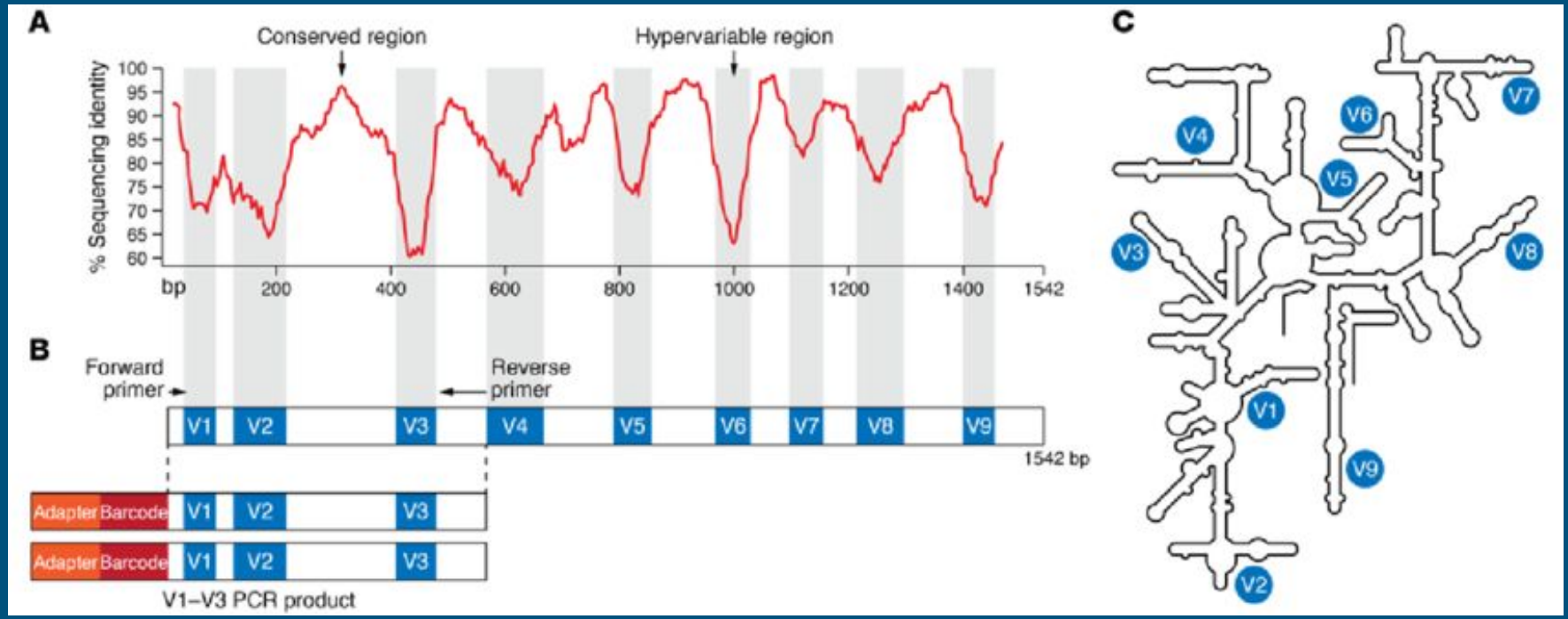
PacBio Single-molecule real-time sequencing (SMRT) technology	Long-read sequencing	Seq by synthesis	Without PCR	The SMRT sequencing employs SMRT Cell, housing numerous small wells known as zero-mode waveguides (ZMWs). Individual DNA molecules are immobilized within these wells, emitting light as the polymerase incorporates each nucleotide, allowing real-time measurement of nucleotide incorporation	average 10,000–25,000	The higher cost compared to other sequencing platforms.
Nanopore DNA sequencing	Long-read sequencing	Sequence detection through electrical impedance	Without PCR	The method relies on the linearization of DNA or RNA molecules and their capability to move through a biological pore called “nanopores”, which are eight nanometers wide. Electrophoretic mobility allows the passage of linear nucleic acid strand, which in turn is capable of generating a current signal.	average 10,000–30,000	The error rate can spike up to 15%, especially with low-complexity sequences. Compared to short-read sequencers, it has a lower read accuracy.

Metabarcoding

1. Barcodes

Prokaryota (Bacteria & Archaea)

16S rRNA

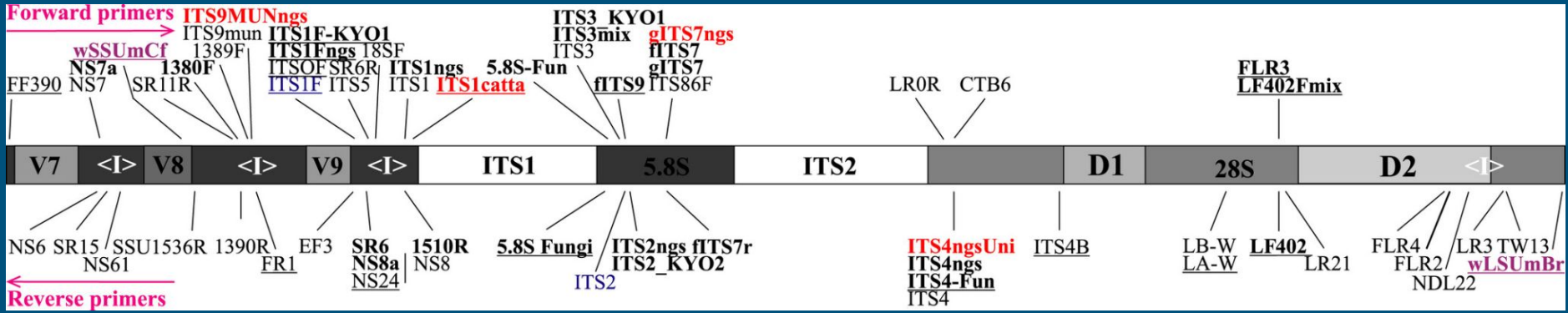


DNA sequence coding for the **16S** subunit of the **ribosomal RNA** (rRNA)

Metabarcoding

1. Barcodes

Fungi | ITS



MOLECULAR ECOLOGY

INVITED REVIEW | Free Access

Best practices in metabarcoding of fungi: From experimental design to results

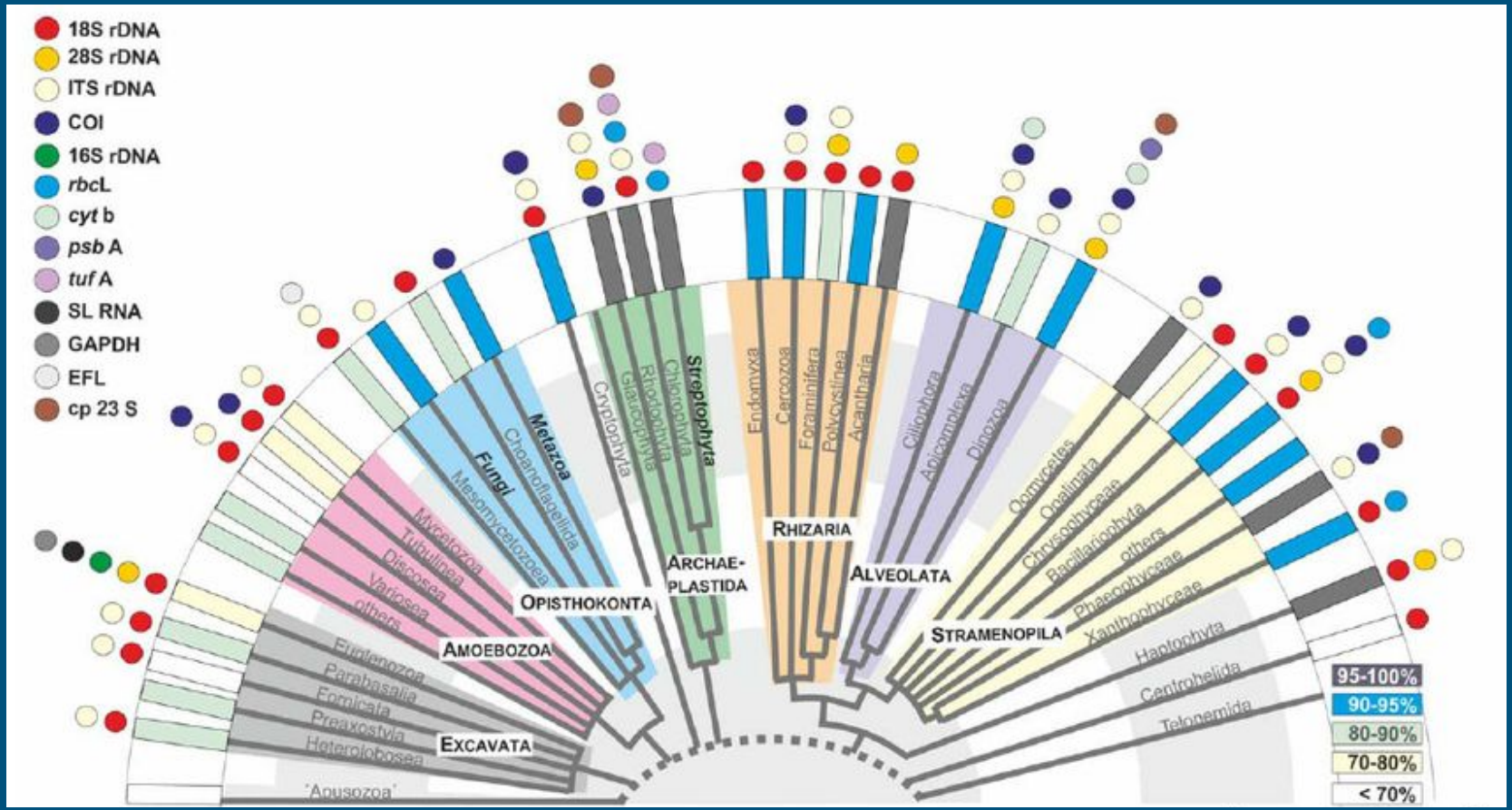
Leho Tedersoo, Mohammad Bahram, Lucie Zinger, R. Henrik Nilsson, Peter G. Kennedy, Teng Yang, Sten Anslan, Vladimir Mikryukov

First published: 08 April 2022 | https://doi.org/10.1111/mec.16460 | Citations: 51

- **Unsuited** for certain fungi because lack of existing ITS region (***Microsporidia***) or poorly conserved flanking sites for primers (***Tulasnellaceae***).
- ITS region **length variability** (50-1500)
- **ITS** copies, multinuclear hyphae,

Metabarcoding
1. Barcodes

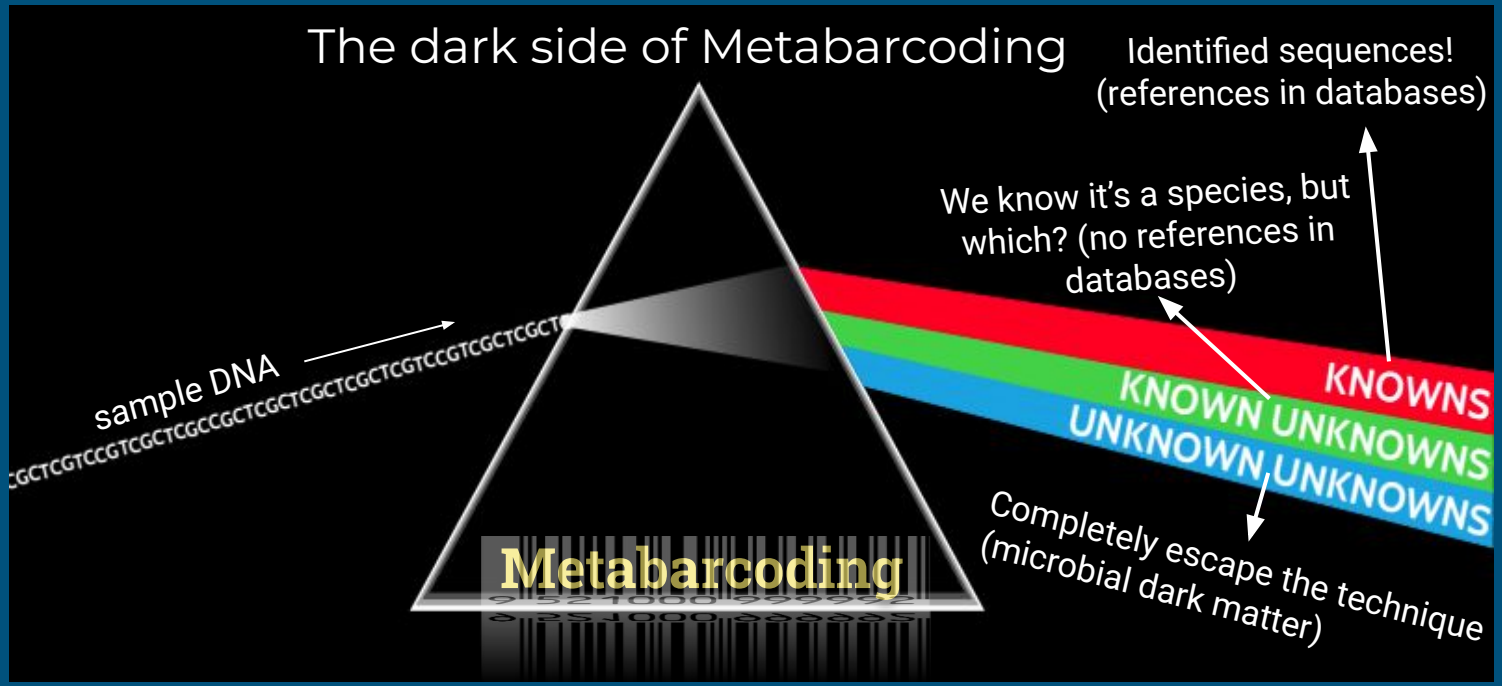
Protists 18S



Metabarcoding

1. Barcodes

Does a **perfect barcode** exist?



Here comes… The **BIAS** police!

(You don't want this guy to be your reviewer 😱)

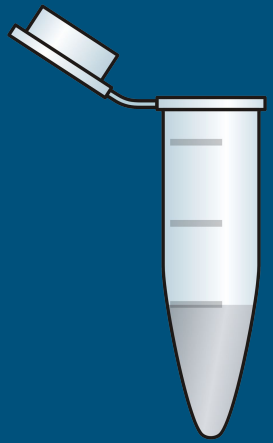
Limitations:

Microbial Dark Matter.

There is no barcode that works for everything. Some taxa will always be left out.

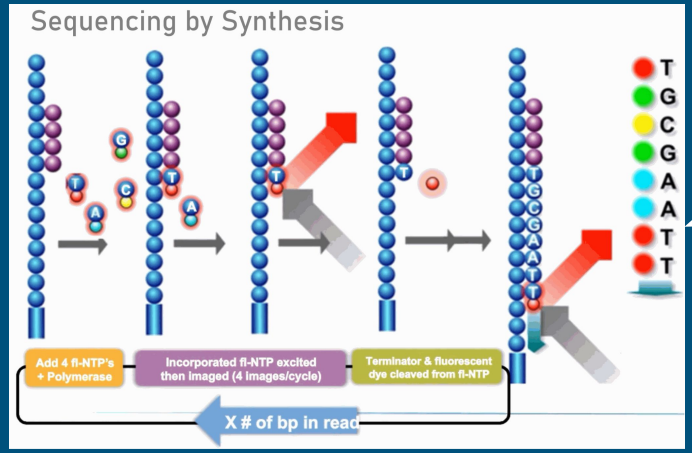
Metabarcoding

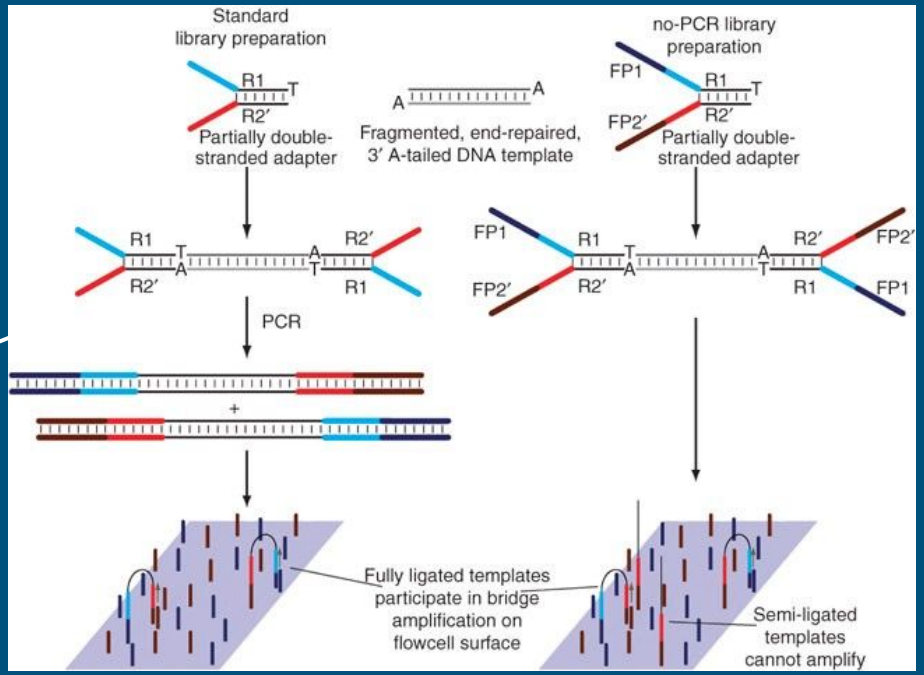
2. DNA extraction

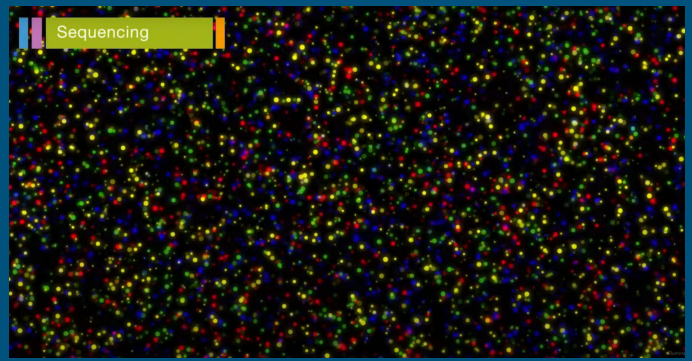


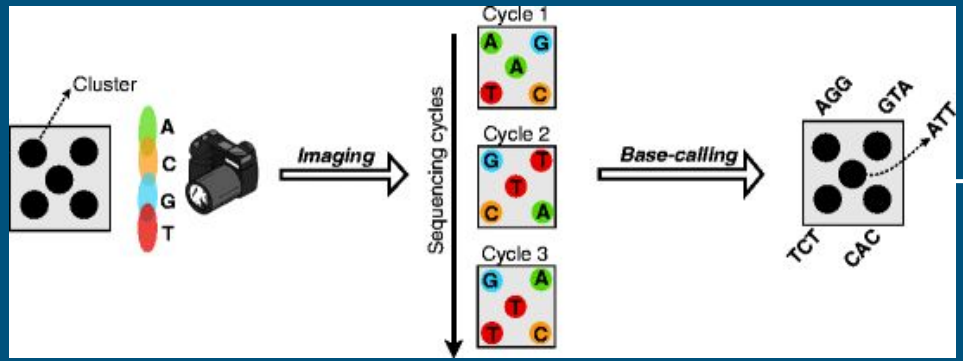
Metabarcoding
3. DNA sequencing

Illumina Sequencing









.fastq

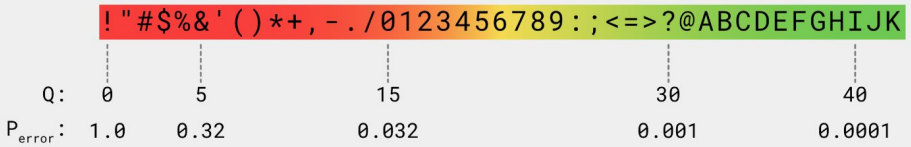
Metabarcoding

4. The DNA sequence read - .fastq file format

FASTQ file sample:

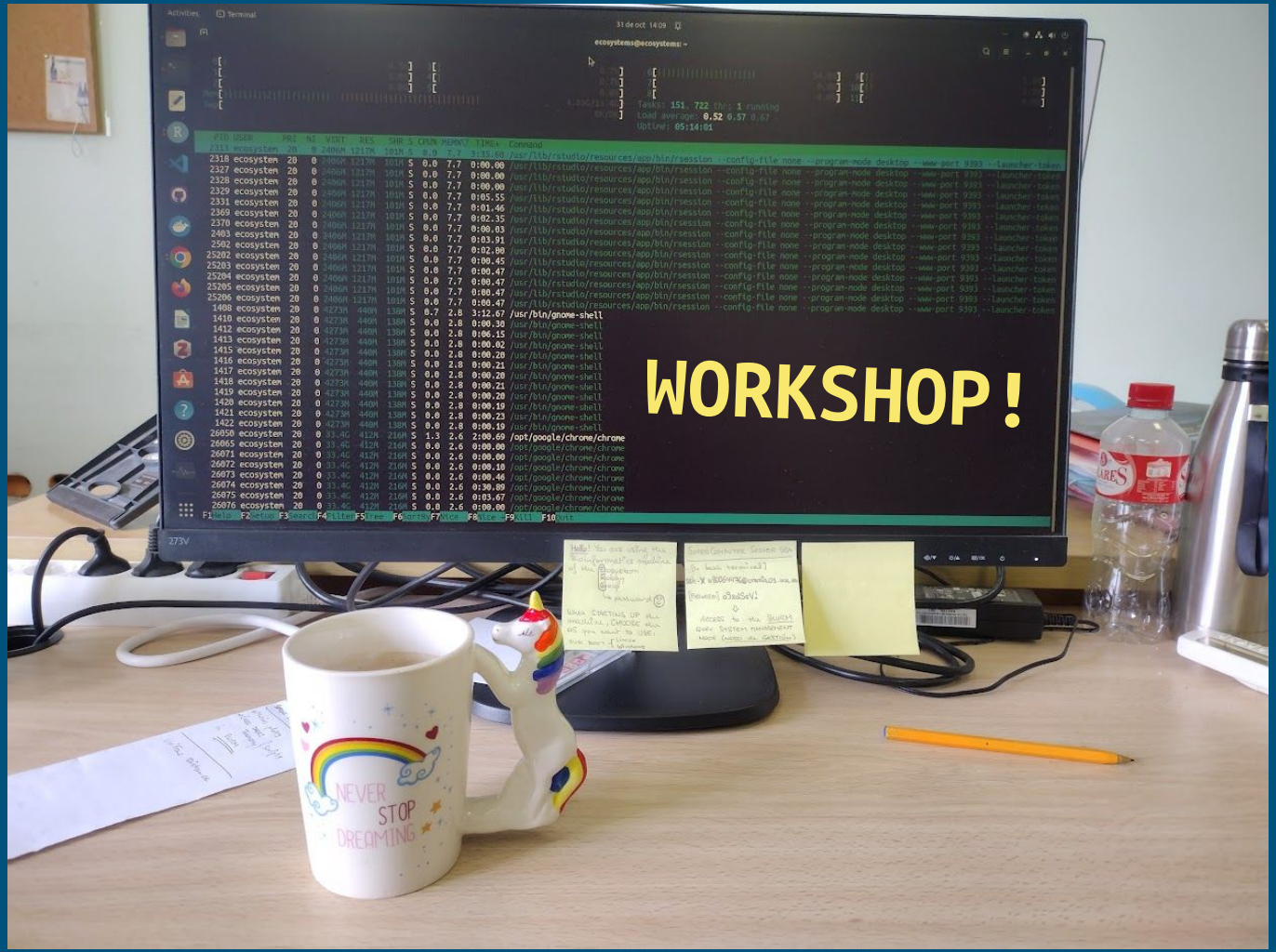
```
@SRR6407486.1 1 length=100
CCTCGTCTACAGCGACAACGTCCAGACCCGCGAACGGGTGATGCGGGCCCTGGGCAAACGGTTGCACCCGGATCTGCCCGATTTGACCTACGTCGAAGTG
+SRR6407486.1 1 length=100
BBBBBFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF<FFFFFFFFFFFFFFFFBFFFFFFFFFFFFFFFFFFFBFFFFFFFFFFF7FFFF<FF
```

Quality scores as ASCII characters:



$$Q = -10\log_{10} P_{error}$$

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


WORKSHOP!
NEVER STOP DREAMING

List of tools & commands used:

bash

```
conda activate metatax
```
Activates the conda environment (metatax contains fastqc, multiqc and cutadapt)

```
fastqc -o [OUTPUT PATH] *fastq.gz
```
Executes fastqc program on any file ended in fastq.gz (in the working directory)

```
multiqc --interactive *
```
Multiqc takes in .fastqc files and compiles multiple samples QC into one .html (working directory must contain .fastqc files)

Key steps in the Rscript: code/01_Trimming_and_Filtering.R

```
# PRIMERS [CHANGE ME!]
cat("Forward and Reverse PRIMERS: \n")
FWD <- "CCTACGGGNGGCWGCAG"; print(FWD)
REV <- "GACTACHVGGGTATCTAATCC"; print(REV)
cat("\n")
```

Specify primer sequence

	Forward	Complement	Reverse	RevComp
FWD.ForwardReads	73803	54	54	54
FWD.ReverseReads	56	53	53	58
REV.ForwardReads	58	54	54	194
REV.ReverseReads	76624	53	53	53

Check for primers in the reads

Key steps in the Rscript: code/01_Trimming_and_Filtering.R

```
for(i in seq_along(fnFs)) {
  cmd_output <- system2(cutadapt,
          args = c(R1.flags, # fwd primer & reverse-complement of rev primer
                   R2.flags, # rev primer & reverse-complement of fwd primer
                   "-n", 4, # -n number of primers to search for and trim in a read,
                            # before going on with next read
                   "-o", fnFs.cut[i],
                   "-p", fnRs.cut[i], # output file's PATH
                   fnFs[i],
                   fnRs[i], #input files
                   "--minimum-length", 10,
                   "--max-n", 0, # [IMPORTANT] Remove indeterminations (Ns)
                                 # Ns accumulate at reads' 5' and 3' ends, right
                                 # where primers are located. This makes primer
                                 # trimming difficult to achieve
                   "--cores", 6), # nº of computing cores
          stdout = TRUE,
          stderr = TRUE)
  # Append command output to summary file
  cat(cmd_output, file = cutadapt_summary, append = TRUE, sep = "\n\n")
}
```

Execute cutadapt

Key steps in the Rscript: code/01_Trimming_and_Filtering.R

```
                  Forward Complement Reverse RevComp
FWD.ForwardReads        0          0       0       0
FWD.ReverseReads        2          0       0       0
REV.ForwardReads        3          0       0       0
REV.ReverseReads        0          0       0       0
> print("Yay! Data should be clean of primers!"); cat("\n")
[1] "Yay! Data should be clean of primers!"
```

Sanity Check: there should be almost no primers remaining in the “cutadapted” data

```
out <- filterAndTrim(fnFs.cut, filtFs,
                     fnRs.cut, filtRs,
                     maxN = 0, # a MUST for DADA2: NO indeterminations (Ns)
                     maxEE = c(4, 6),
                     truncQ = 2,
                     minLen = 100,
                     trimLeft = 5, # remove 10 first bps (low Quality in QA)
                     truncLen = c(267, 232), # [CHANGE ME according to QA
                                             # & length of amplicon region]
                     rm.phix = TRUE,
                     compress = TRUE,
                     multithread = 6, # bool or int with number of threads
                     matchIDs = TRUE,
                     verbose = TRUE
)
```

filterAndTrim() command

This command executes a filtering process based on read quality. Modify the parameters according to the fastqc + multiqc analysis