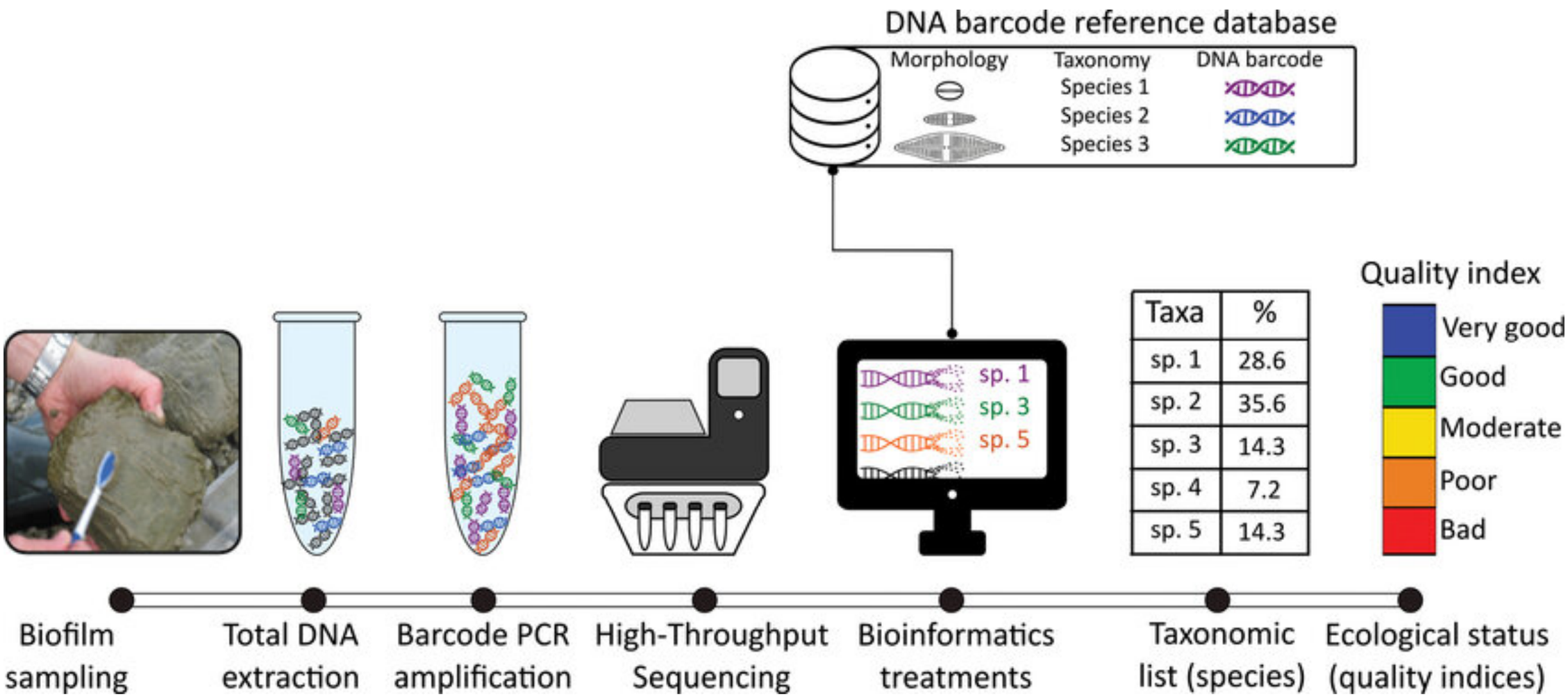
BIOLAWEB

# Bioinformatic analysis of metabarcoding data with DADA2

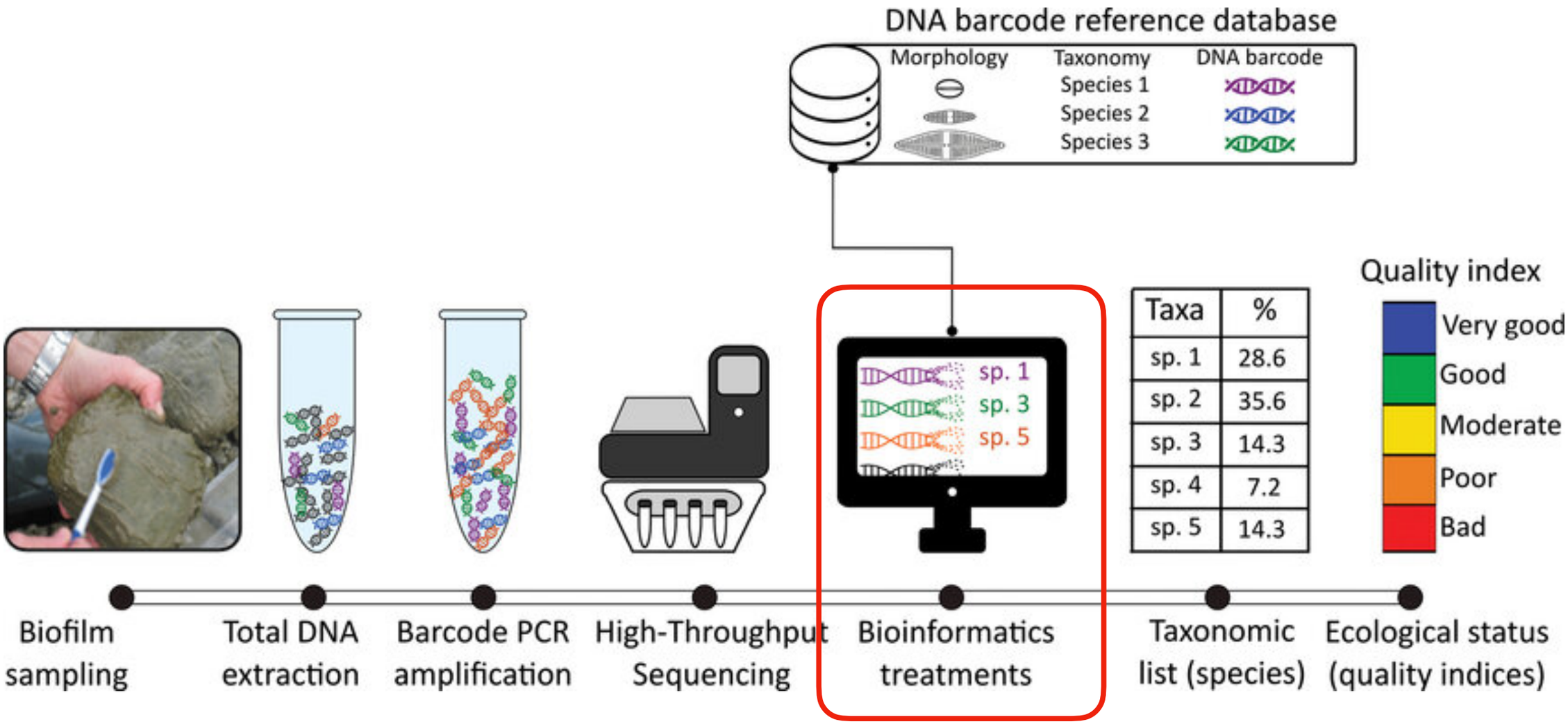
***Theorical part***

Clarisse Lemonnier

# Importance of the bioinformatic steps

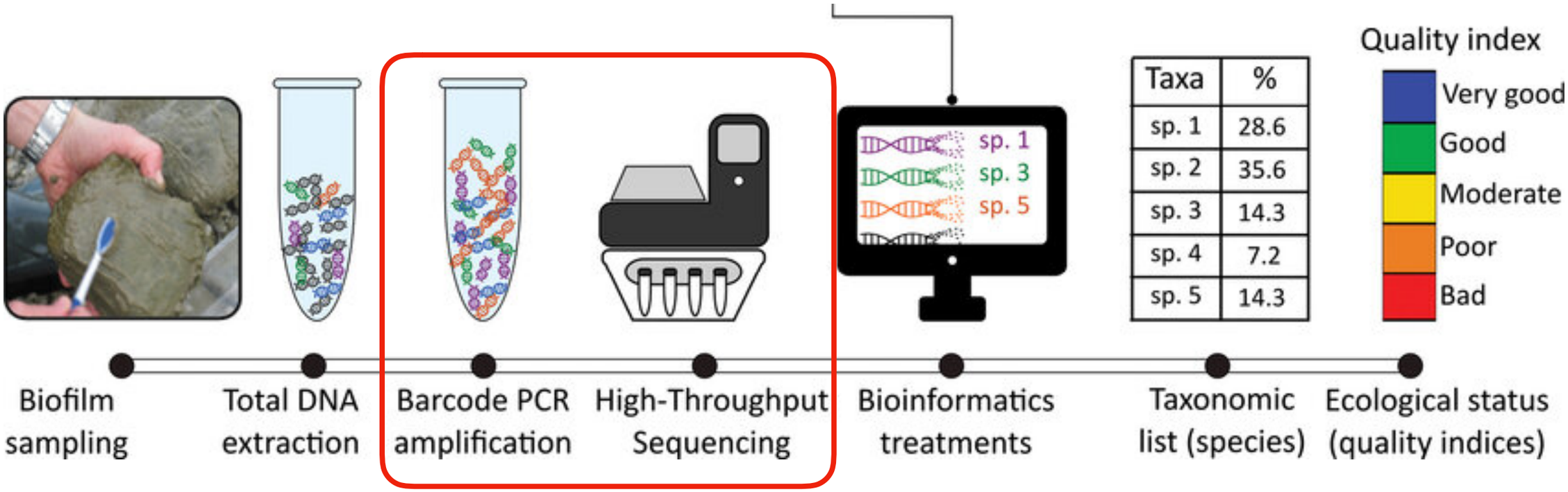


# Importance of the bioinformatic steps



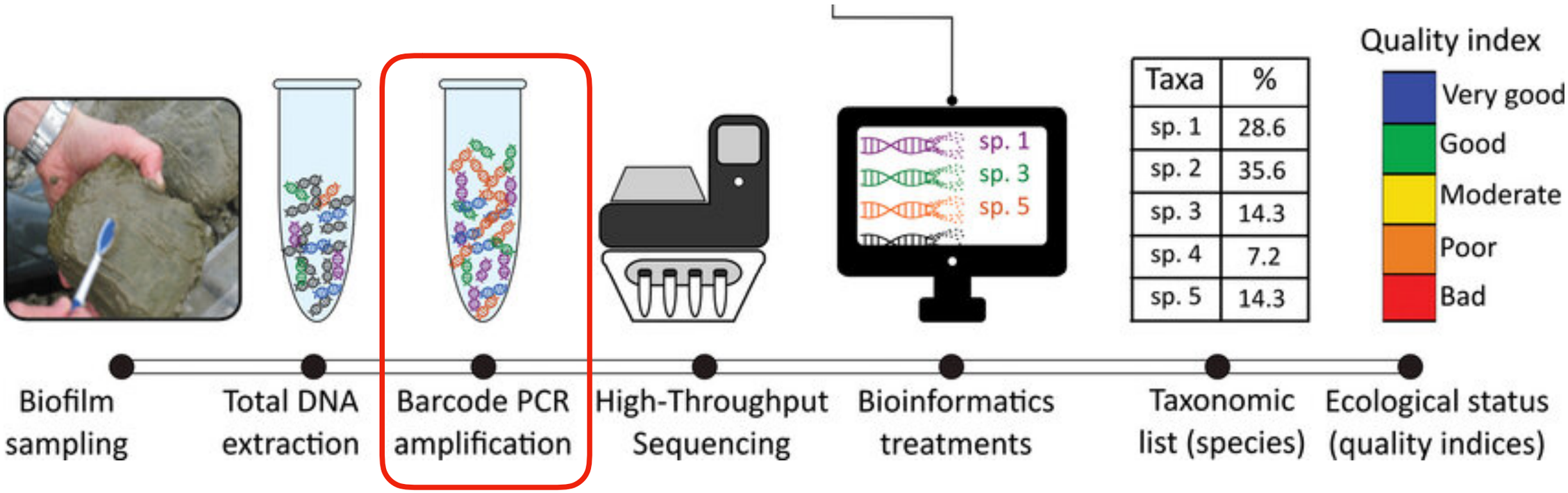

**It is this step that you will produce an ASV/OTU table that will be at the beginning of taxonomic identification, ecological analysis and quality indices calculation**

# Some reminders on the metabarcoding steps :




**It is important to know what's happening during these steps to understand all the bioinformatic pipeline**

# Some reminders on the metabarcoding steps :




## 1. PCR

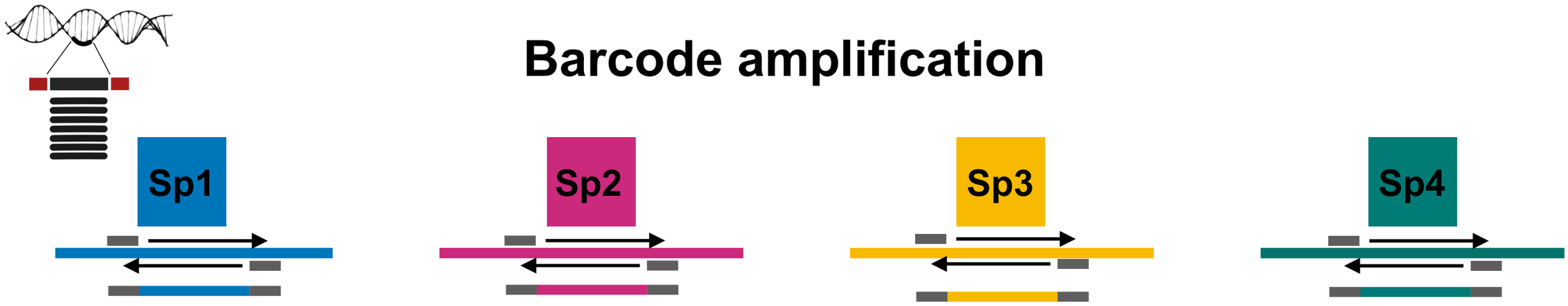
# Let’s do metabarcoding in a sample with 4 species

Sp1

Sp2

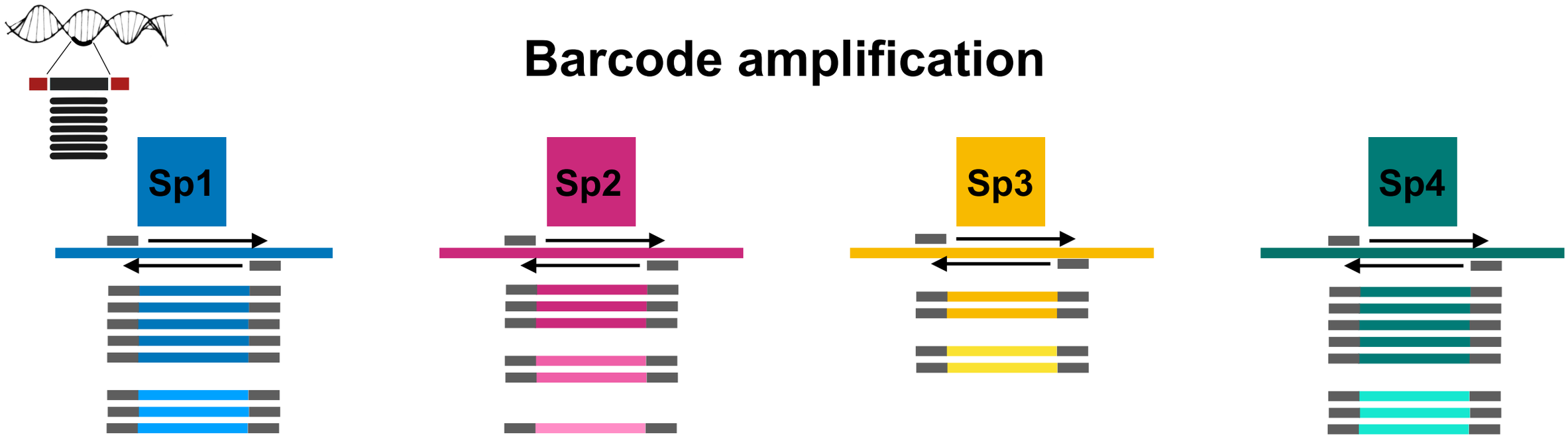
Sp3

Sp4




*Primers are added*

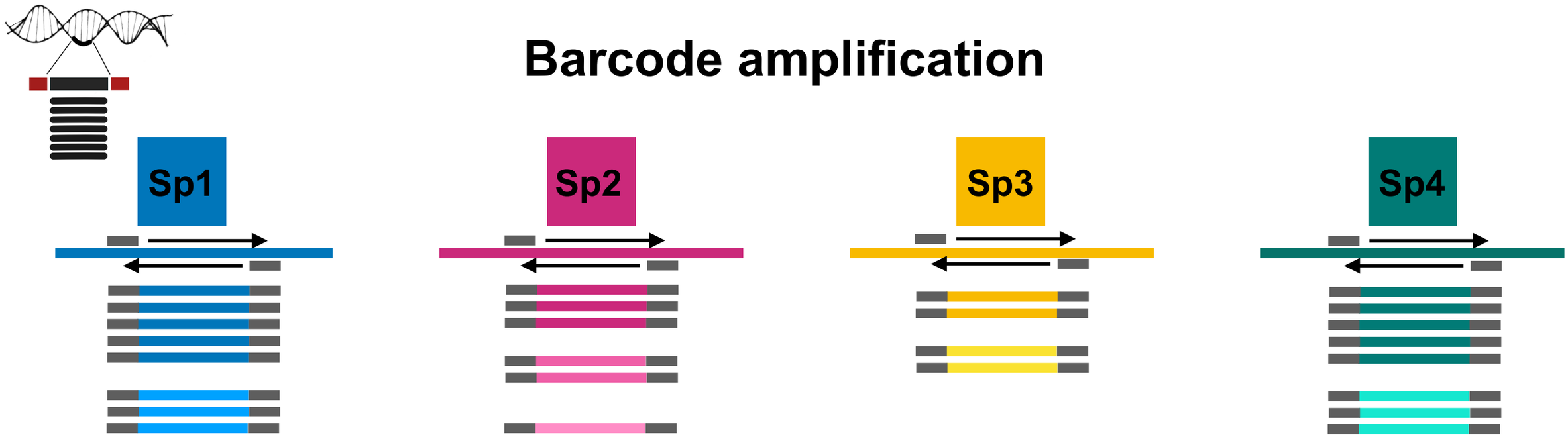
# Barcode amplification




*Primers are added*

*The DNA polymerase can make mistakes and incorporate the wrong base creating slightly different sequences*

# Barcode amplification



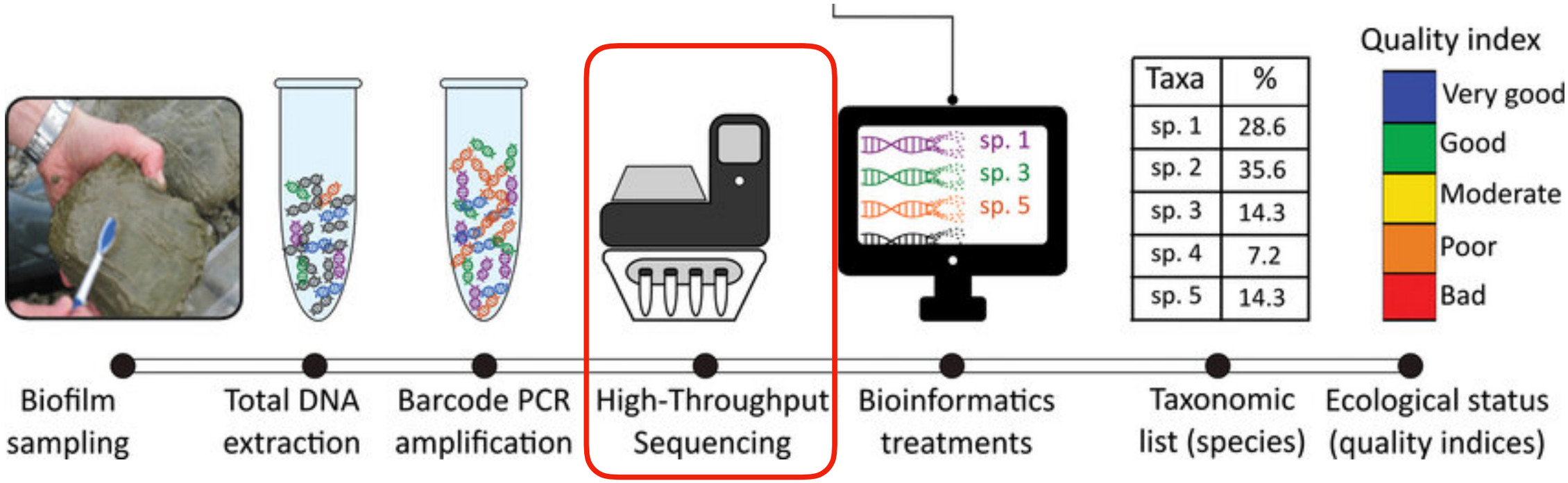

*Primers are added*

*The DNA polymerase can make mistakes and incorporate the wrong base creating slightly different sequences*

*Chimera are created*

**Vocabulary : at this step you have <u>amplicons</u> of the targeted barcode**

# Some reminders on the metabarcoding steps :




## 2. Sequencing

Note : I will be only speaking of Illumina MiSeq sequencing

# 1. Multiplexing


*A MiSeq run can produce*
*~10 millions reads*

*Sample 1*


*… but 100 000 reads in a sample is often enough*

# 1. Multiplexing


*Index (e.g ATTCG)*

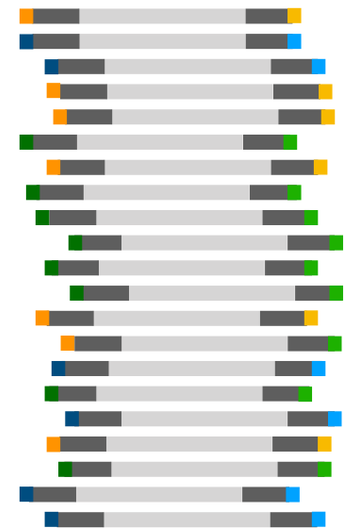
*Index (e.g CGAAG)*

*Sample 1*

*Sample 2*

*Sample 3*

***…Up to 96 samples***



***Demultiplexing is usually done by the sequencing platform***

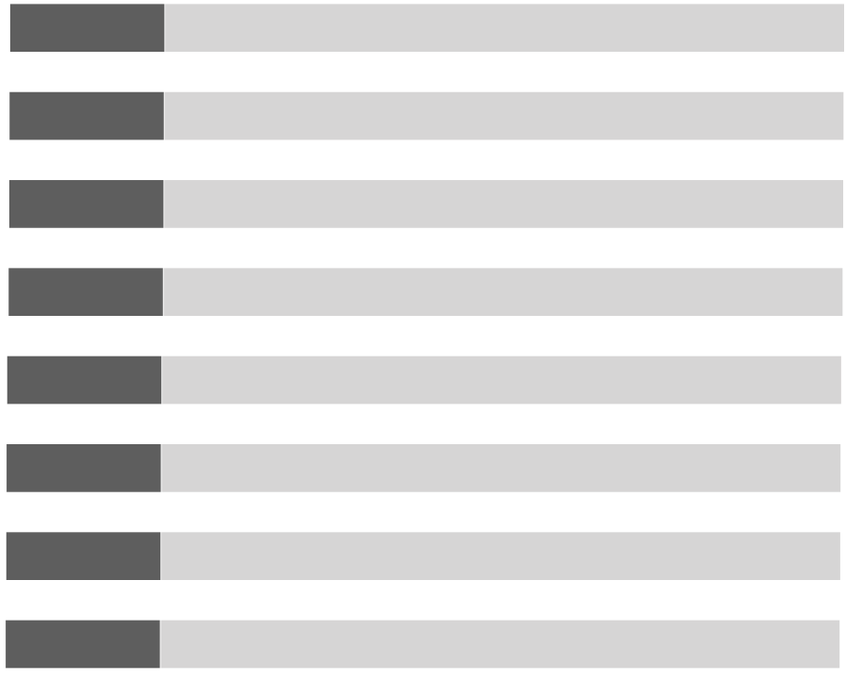
# 2. Paired-end sequencing


*Illumina technology : 300bp max (often 250bp)*

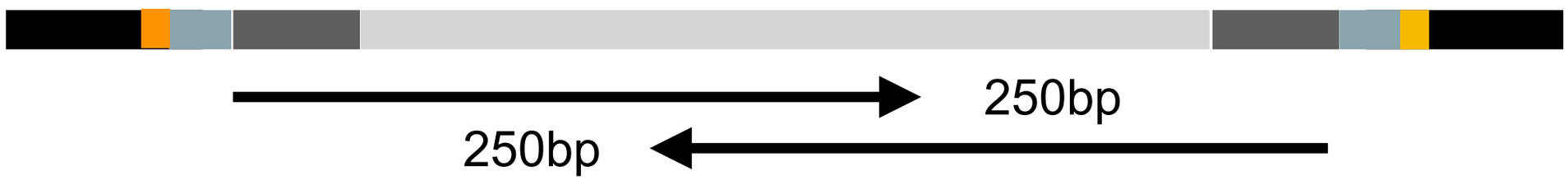
***23S phytoplankton barcode + primers = ~400bp***

# 2. Paired-end sequencing

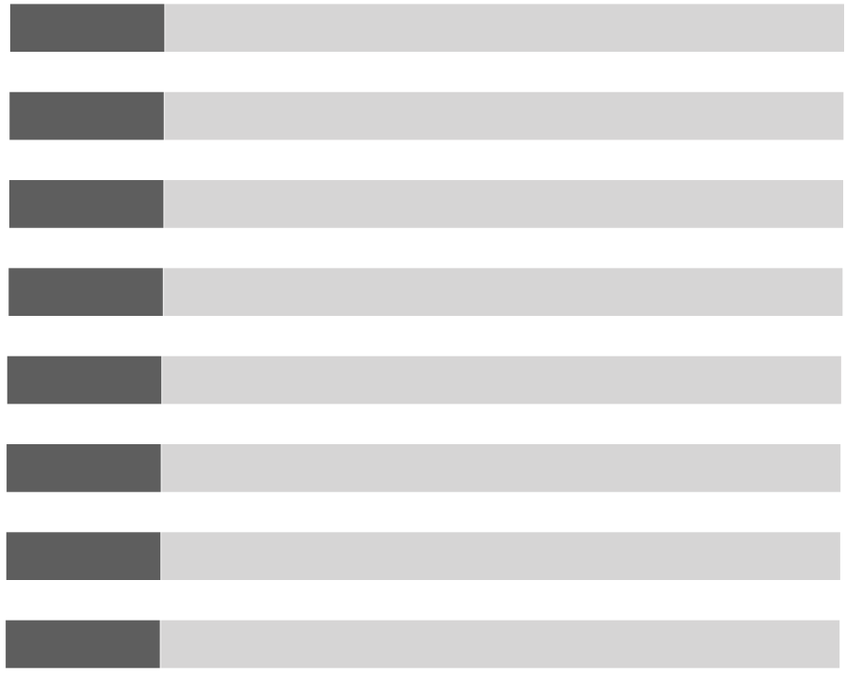

## Read 1/Forward



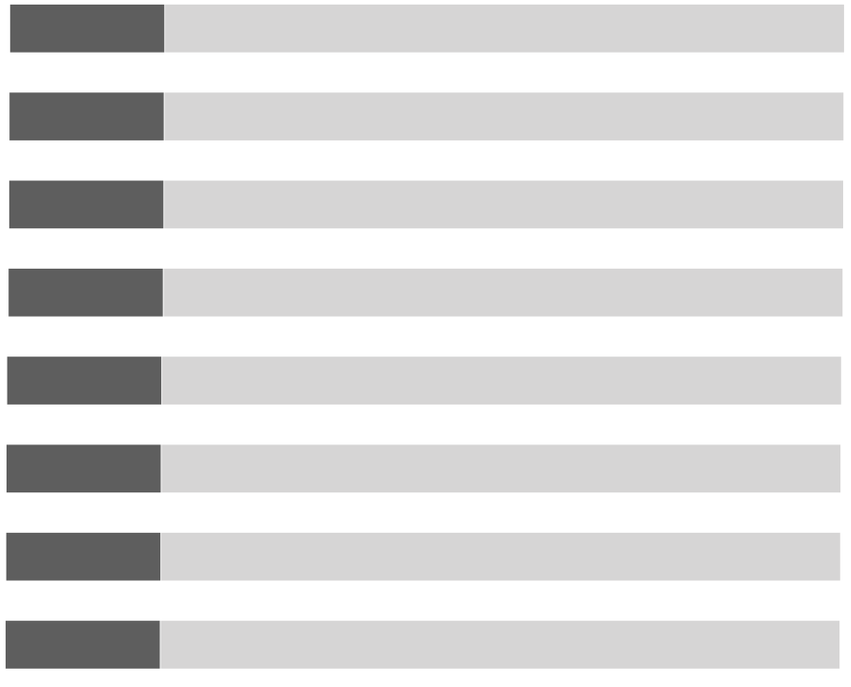
# 2. Paired-end sequencing




**Read 1/Forward**

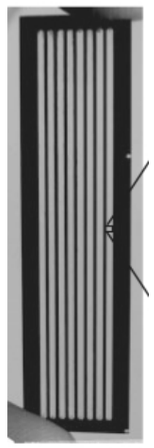


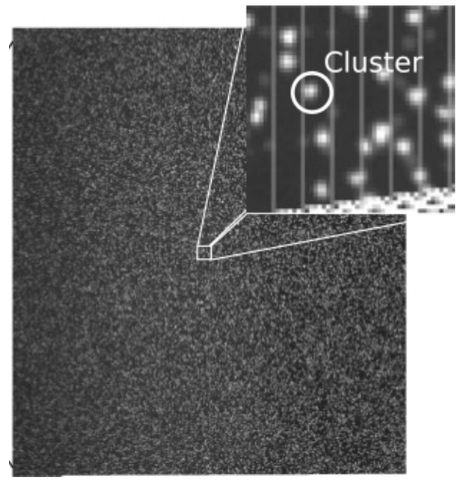
**Read 2/Reverse**



**At the end of a run, you have 2 fastq files for each sample**

# Fastq file




X Y

Defline

**@M01028:94:000000000-AHT10:1:1101:15604:1824 1:N:0:10**
ATGCTCCAGCAGCTGCGGTAATACGGAGGATGCAAGCGTTATCCGGATTTATTGGGTT
TAAAGGGTCCGCAGGCGGATCAATAAGTCAGTGGTGAAAGCCCATAGCTCAACTATG
GAACTGCCATTGAAACTGTTGATCTTGAGTCTAGTTGACGTAGGCGGAATGTGACTT
GTAGCGGTGAAATGCATAGATATGTCACAGAACACCAATTGCGAAGGCAGCTTACGT
AACTCGACTGACGCTCTTGGAC
**+**
3AABAFFFFFFFGGGGGGEGGGHHHGEGG2FFB3BDEEGG2F5FECBEEHDFGGFFEF
G3AFFG?GCCE0>EE?FG/>EE3FF3DG44FGHEHGBGE3GGHGF2?GGFF?
GFCGFBHHFHFHFBF@2F2
FHGBHHD1GHGGH1CHFF11GFBD1CFCAD1<@CCC0CC0;0<CGF0C-
@GA.0;;0C0;;FFCCFBB9090;/;9/BE..;FB09-9-:....FF;B../;.//9.9..//9.9..9B///
**@M01028:94:000000000-AHT10:1:1101:12447:1845 1:N:0:10**
**…**

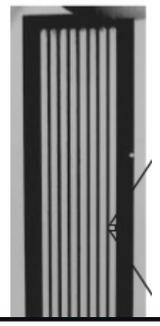
Sequence

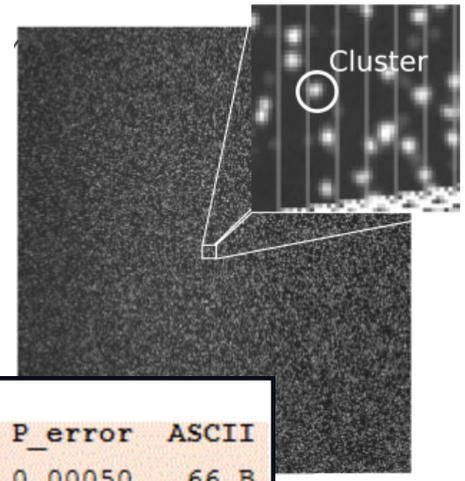
Quality score

**x50 000 - 100 000**

# Fastq file






ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

| Q | P_error | ASCII | Q | P_error | ASCII | Q | P_error | ASCII | Q | P_error | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.00000 | 33 ! | 11 | 0.07943 | 44 , | 22 | 0.00631 | 55 7 | 33 | 0.00050 | 66 B |
| 1 | 0.79433 | 34 " | 12 | 0.06310 | 45 - | 23 | 0.00501 | 56 8 | 34 | 0.00040 | 67 C |
| 2 | 0.63096 | 35 # | 13 | 0.05012 | 46 . | 24 | 0.00398 | 57 9 | 35 | 0.00032 | 68 D |
| 3 | 0.50119 | 36 $ | 14 | 0.03981 | 47 / | 25 | 0.00316 | 58 : | 36 | 0.00025 | 69 E |
| 4 | 0.39811 | 37 % | 15 | 0.03162 | 48 0 | 26 | 0.00251 | 59 ; | 37 | 0.00020 | 70 F |
| 5 | 0.31623 | 38 & | 16 | 0.02512 | 49 1 | 27 | 0.00200 | 60 < | 38 | 0.00016 | 71 G |
| 6 | 0.25119 | 39 ' | 17 | 0.01995 | 50 2 | 28 | 0.00158 | 61 = | 39 | 0.00013 | 72 H |
| 7 | 0.19953 | 40 ( | 18 | 0.01585 | 51 3 | 29 | 0.00126 | 62 > | 40 | 0.00010 | 73 I |
| 8 | 0.15849 | 41 ) | 19 | 0.01259 | 52 4 | 30 | 0.00100 | 63 ? | 41 | 0.00008 | 74 J |
| 9 | 0.12589 | 42 * | 20 | 0.01000 | 53 5 | 31 | 0.00079 | 64 @ | 42 | 0.00006 | 75 K |
| 10 | 0.10000 | 43 + | 21 | 0.00794 | 54 6 | 32 | 0.00063 | 65 A | | | |

Defline

GTAGCGGTGAAATGCATAGATATGTCACAGAACACCAATTGCGAAGGCAGCTTACGT
AACTCGACTGACGCTCTTGGAC

Sequence

**+**

3AABAFFFFFFFGGGGGGEGGGHHHGEGG2FFB3BDEEGG2F5FECBEEHDFGGFFEF
G3AFFG?GCCE0>EE?FG/>EE3FF3DG44FGHEHGBGE3GGHGF2?GGFF?
GFCGFBHHFHFHFBF@2F2
FHGBHHD1GHGGH1CHFF11GFBD1CFCAD1<@CCC0CC0;0<CGF0C-
@GA.0;;0C0;;FFCCFBB9090;/;9/BE..;FB09-9-:....FF;B../;.//9.9..//9.9..9B///

Quality score

**@M01028:94:000000000-AHT10:1:1101:12447:1845 1:N:0:10**

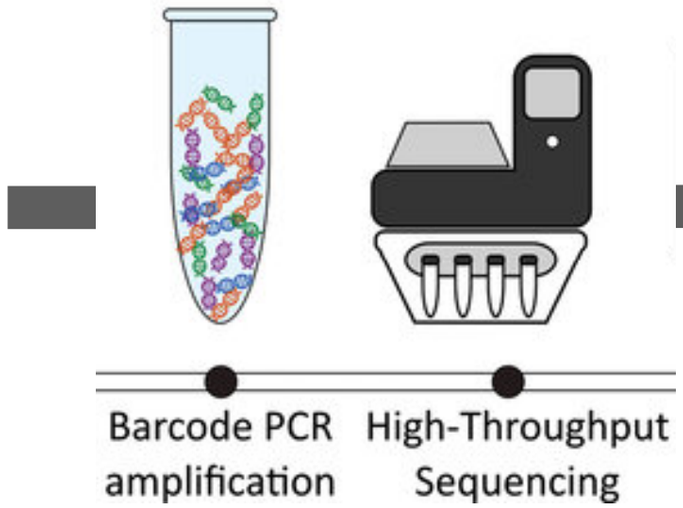
**…**

**x50 000 - 100 000**

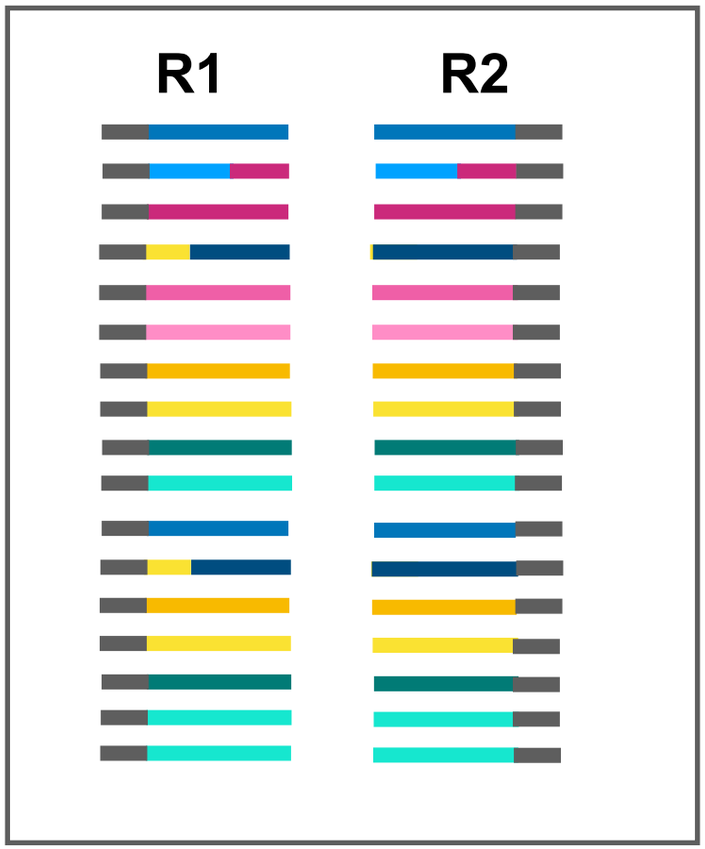
# In summary

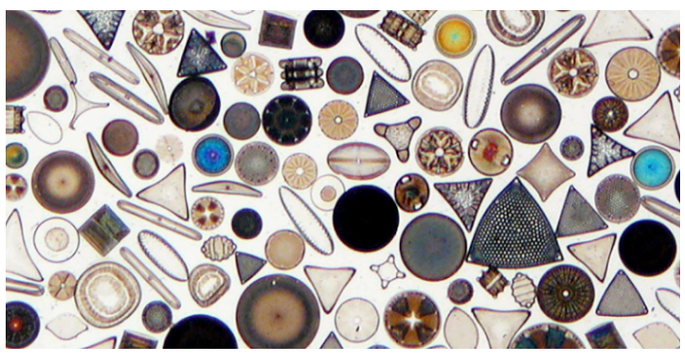
**You started with this…**

**Sample 1**




**…and now your dealing with this !**

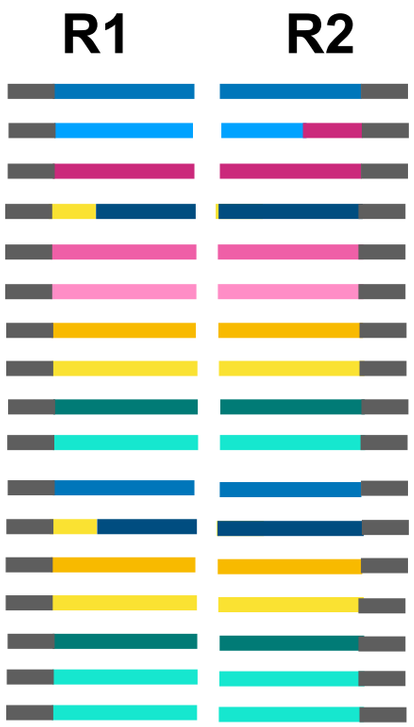


**<u>A lot of noise is added during the PCR and sequencing steps</u>**

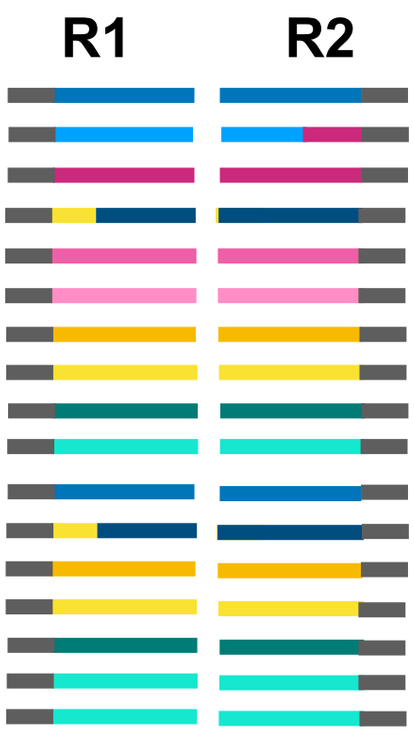
# BIOINFORMATIC PIPELINE - the theory

# First objective of bioinformatic pipeline : having full sequences of good quality




Raw

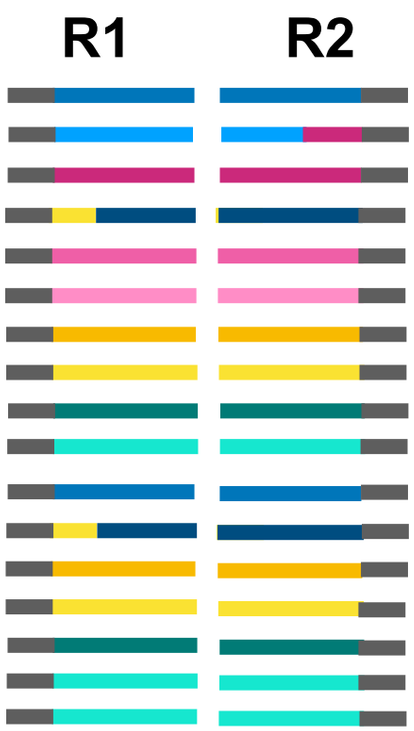
# First objective of bioinformatic pipeline : having full sequences of good quality




Raw

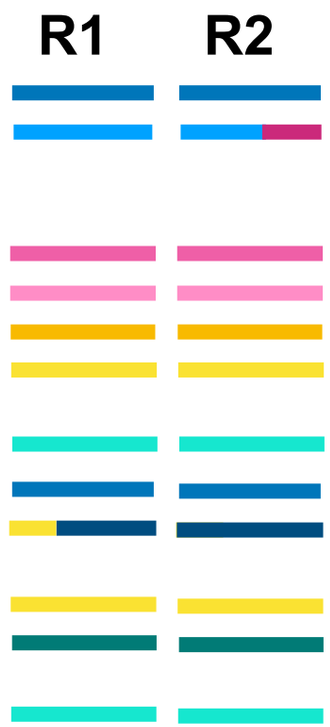

Remove primers

# First objective of bioinformatic pipeline : having full sequences of good quality




Raw


Remove primers

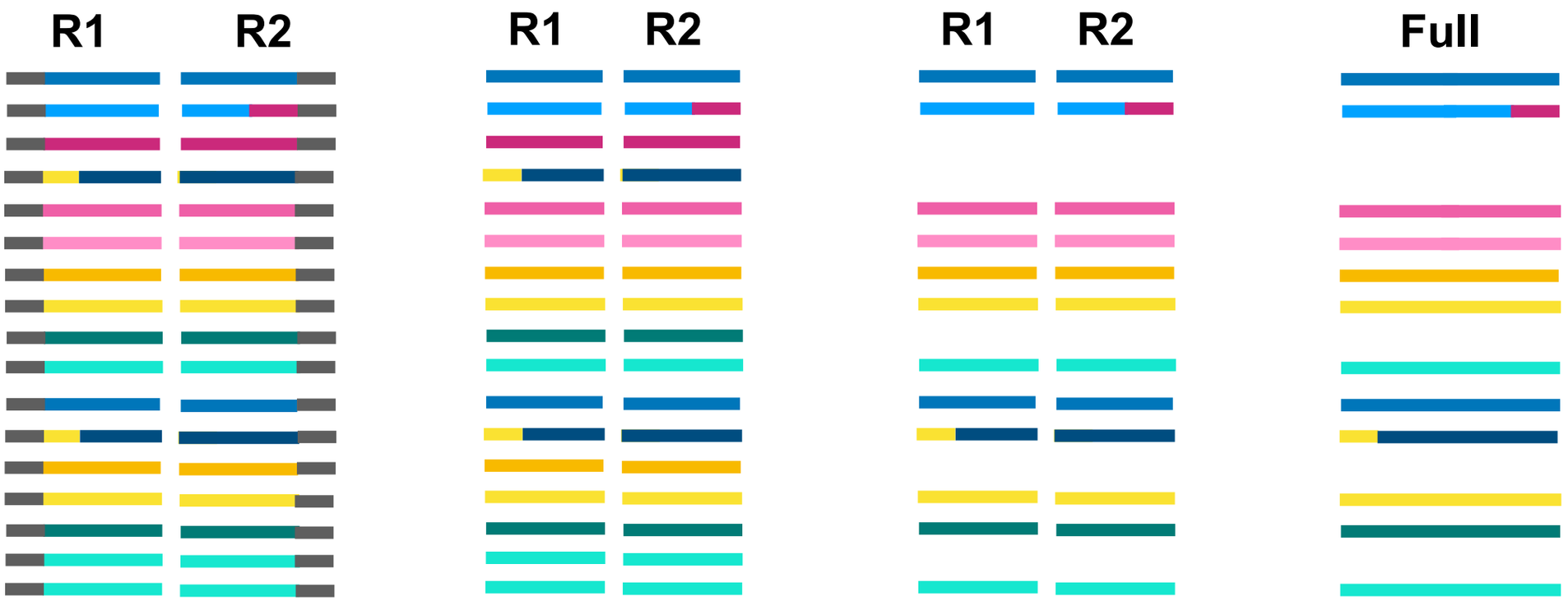



Remove sequences of bad quality

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

| Q | P_error | ASCII | Q | P_error | ASCII | Q | P_error | ASCII | Q | P_error | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.00000 | 33 ! | 11 | 0.07943 | 44 , | 22 | 0.00631 | 55 7 | 33 | 0.00050 | 66 B |
| 1 | 0.79433 | 34 " | 12 | 0.06310 | 45 - | 23 | 0.00501 | 56 8 | 34 | 0.00040 | 67 C |
| 2 | 0.63096 | 35 # | 13 | 0.05012 | 46 . | 24 | 0.00398 | 57 9 | 35 | 0.00032 | 68 D |
| 3 | 0.50119 | 36 $ | 14 | 0.03981 | 47 / | 25 | 0.00316 | 58 : | 36 | 0.00025 | 69 E |
| 4 | 0.39811 | 37 % | 15 | 0.03162 | 48 0 | 26 | 0.00251 | 59 ; | 37 | 0.00020 | 70 F |
| 5 | 0.31623 | 38 & | 16 | 0.02512 | 49 1 | 27 | 0.00200 | 60 < | 38 | 0.00016 | 71 G |
| 6 | 0.25119 | 39 ' | 17 | 0.01995 | 50 2 | 28 | 0.00158 | 61 = | 39 | 0.00013 | 72 H |
| 7 | 0.19953 | 40 ( | 18 | 0.01585 | 51 3 | 29 | 0.00126 | 62 > | 40 | 0.00010 | 73 I |
| 8 | 0.15849 | 41 ) | 19 | 0.01259 | 52 4 | 30 | 0.00100 | 63 ? | 41 | 0.00008 | 74 J |
| 9 | 0.12589 | 42 * | 20 | 0.01000 | 53 5 | 31 | 0.00079 | 64 @ | 42 | 0.00006 | 75 K |
| 10 | 0.10000 | 43 + | 21 | 0.00794 | 54 6 | 32 | 0.00063 | 65 A | | | |

# First objective of bioinformatic pipeline : having full sequences of good quality



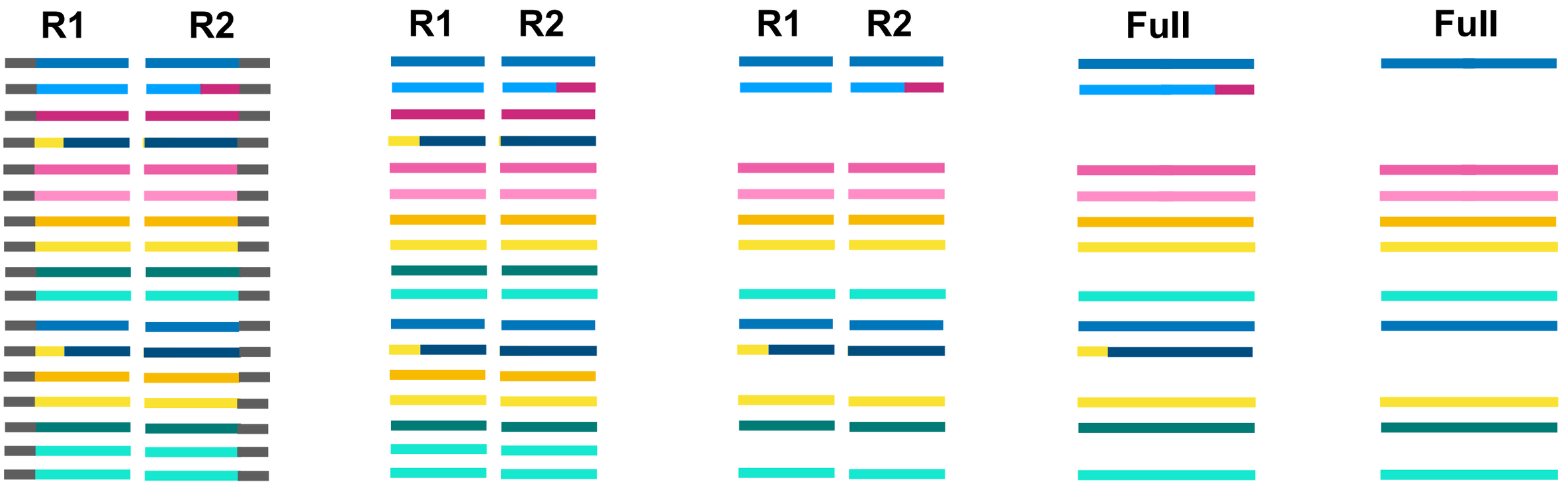

Raw

Remove primers

Remove sequences of bad quality

Merge

# First objective of bioinformatic pipeline : having full sequences of good quality




Raw

Remove primers

Remove sequences of bad quality

Merge

Remove chimera

# Second objective of bioinformatic pipeline : Make sense of these sequences in a biological/ ecological way

**Full**

**Good quality sequences**

There are still PCR or sequencing errors

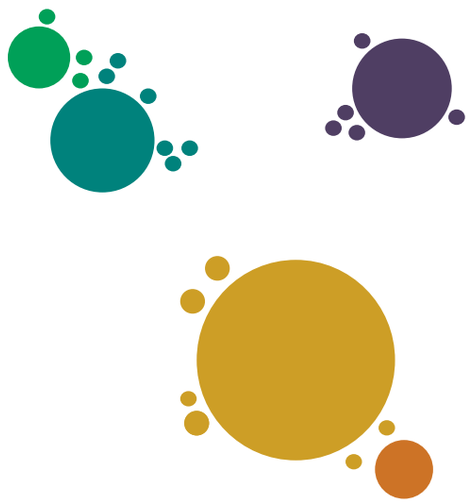
**<u>2 main strategies are used to deal with sequencing errors:</u>**

**clustering or denoising**

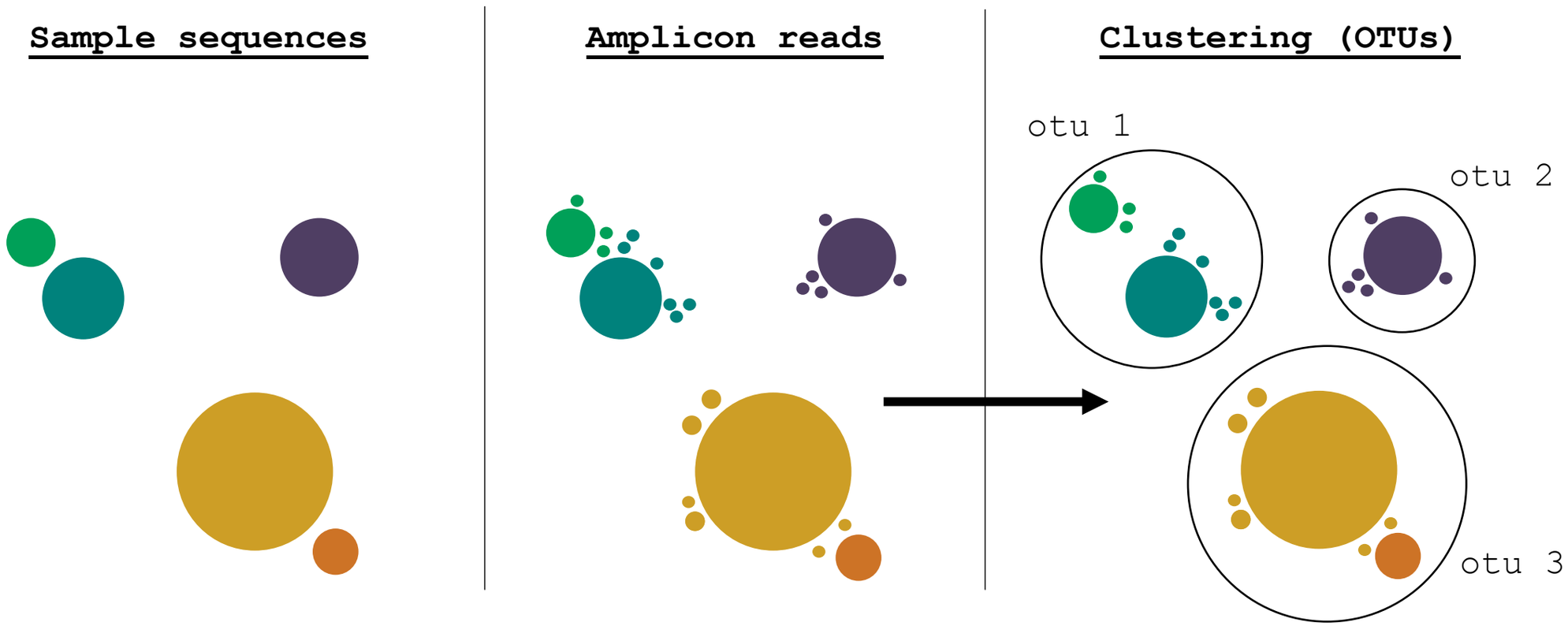
# 1/ Clustering

**Sample sequences**

**Amplicon reads**



# 1/ Clustering




The first method used: cluster sequences based on their similarity, with a user-defined threshold (often 97%)

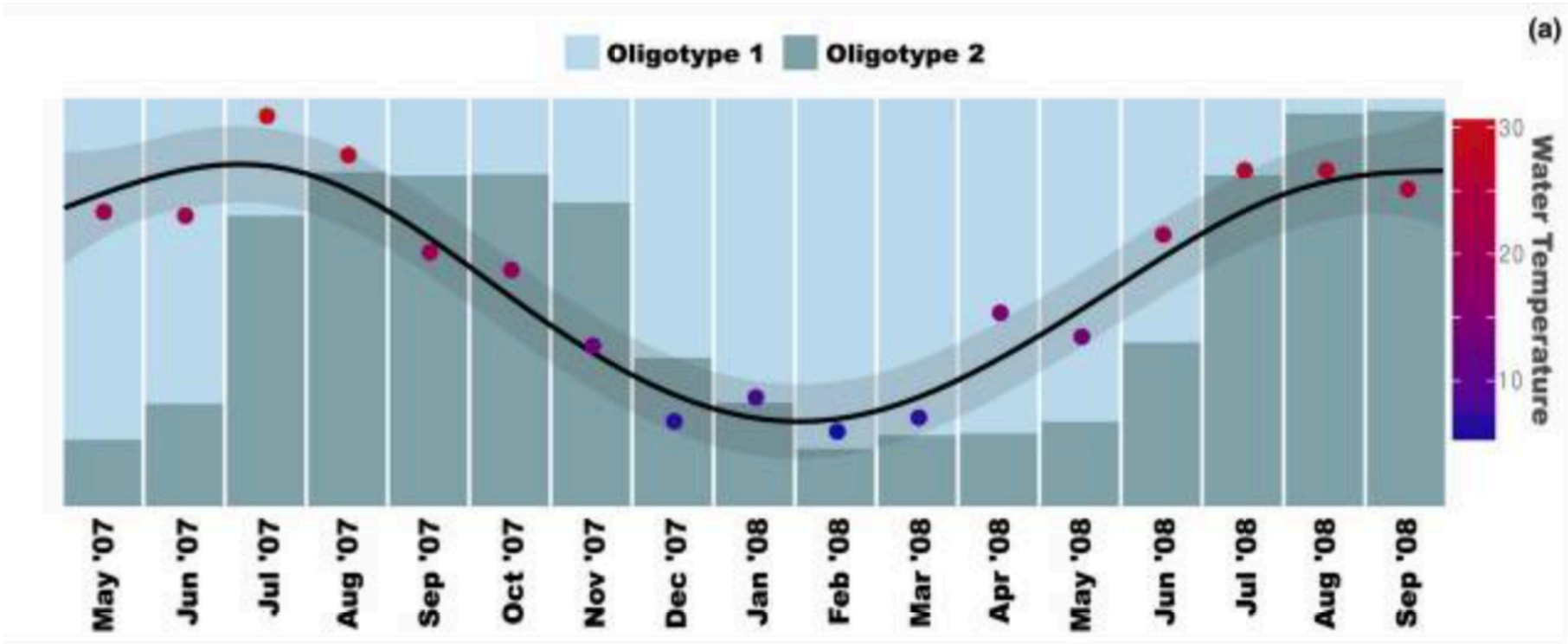
Scientists at the beginning wanted to be close to taxonomic concepts as well as getting rid of sequencing errors. This 97% threshold was thought to correspond to the species level.

Clustered sequences are then called Operational Taxonomic Units (or OTU)

# 1/ Clustering

However, defining an arbitrary threshold has some limits

Very similar sequences can have distinct ecological patterns

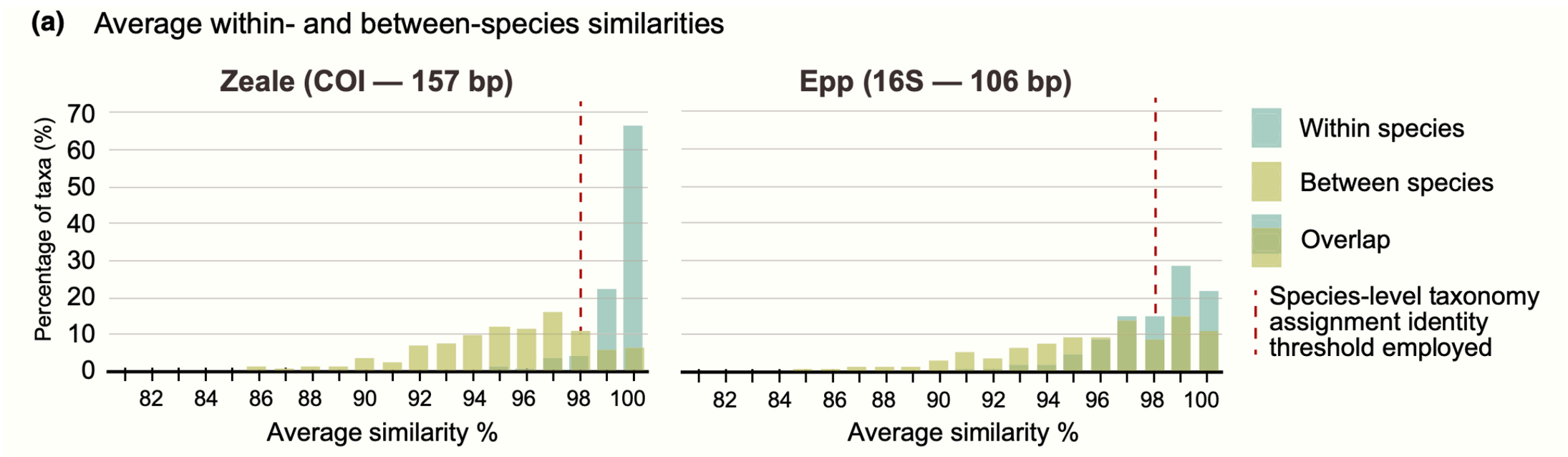



Here the two sequences of 459 nucleotides
are identical at 99.57%

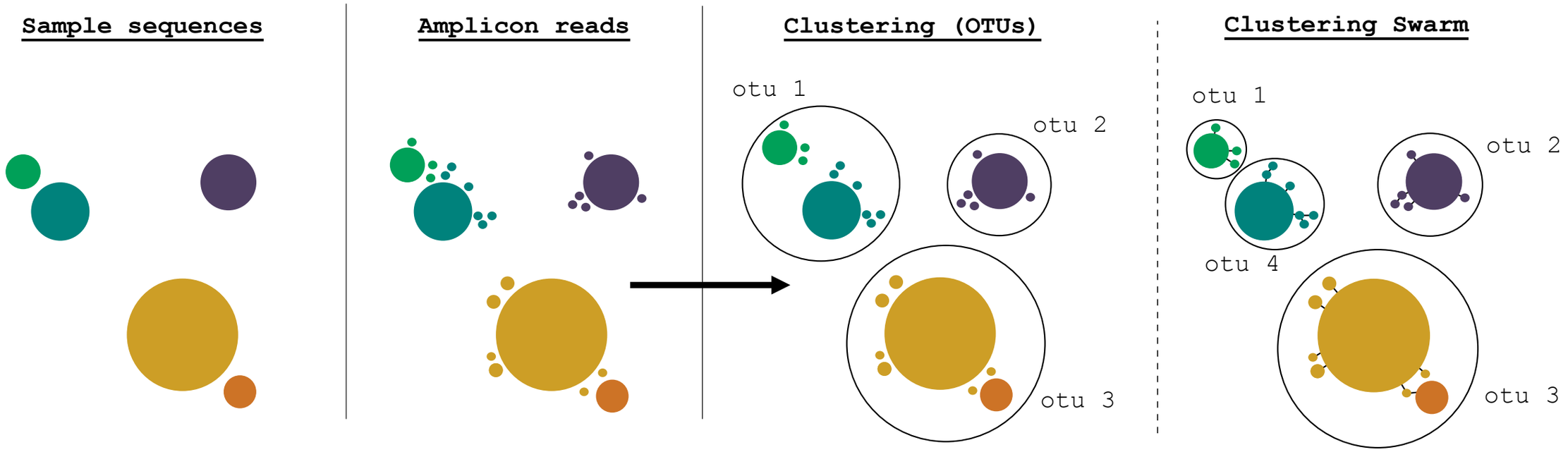
# 1/ Clustering

However, defining an arbitrary threshold has some limits

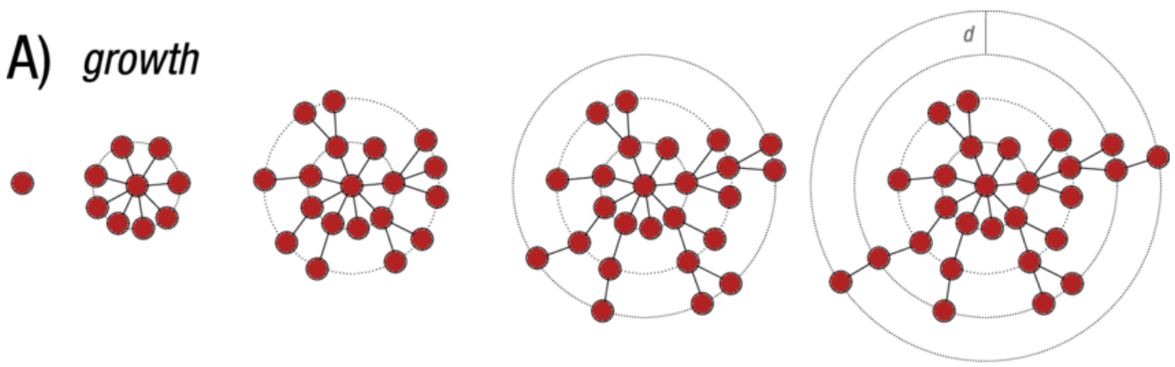
And the similarity threshold varies depending on the species

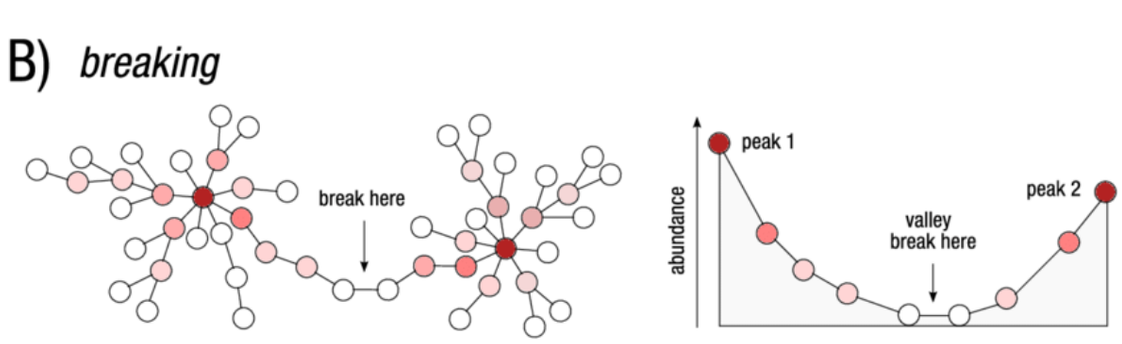


# 1/ Clustering

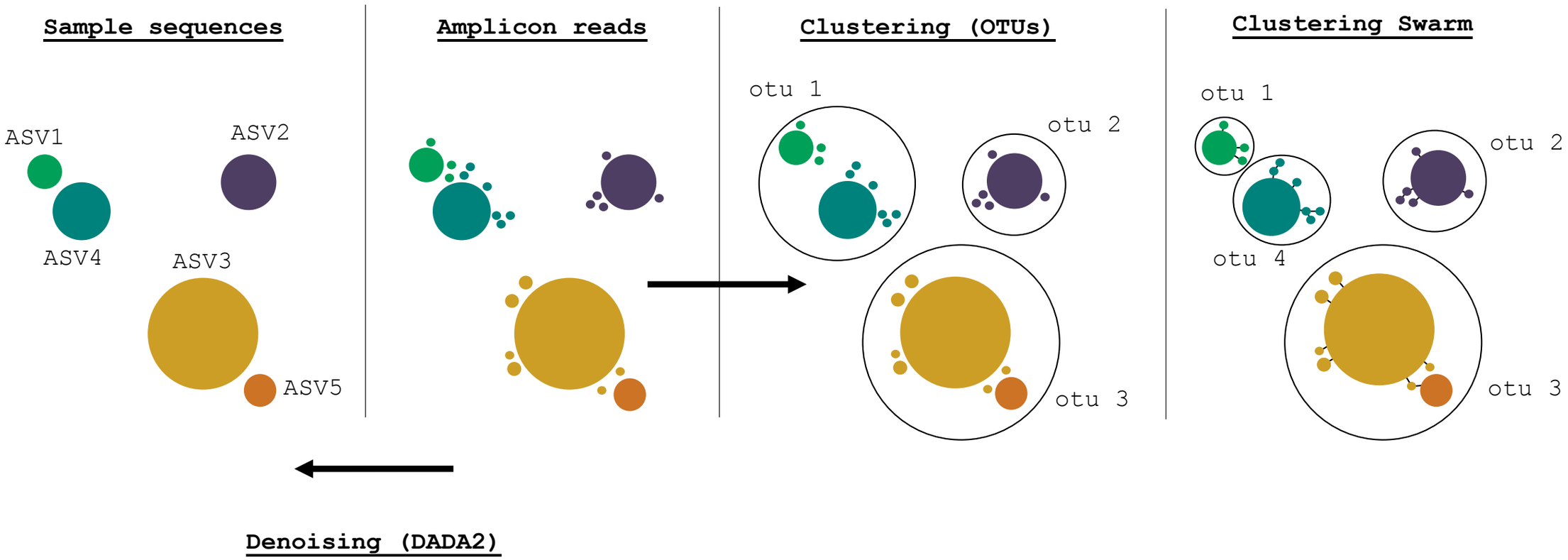



## Exemple of a clustering method using a soft-threshold : Swarm (Mahé et al., 2014)



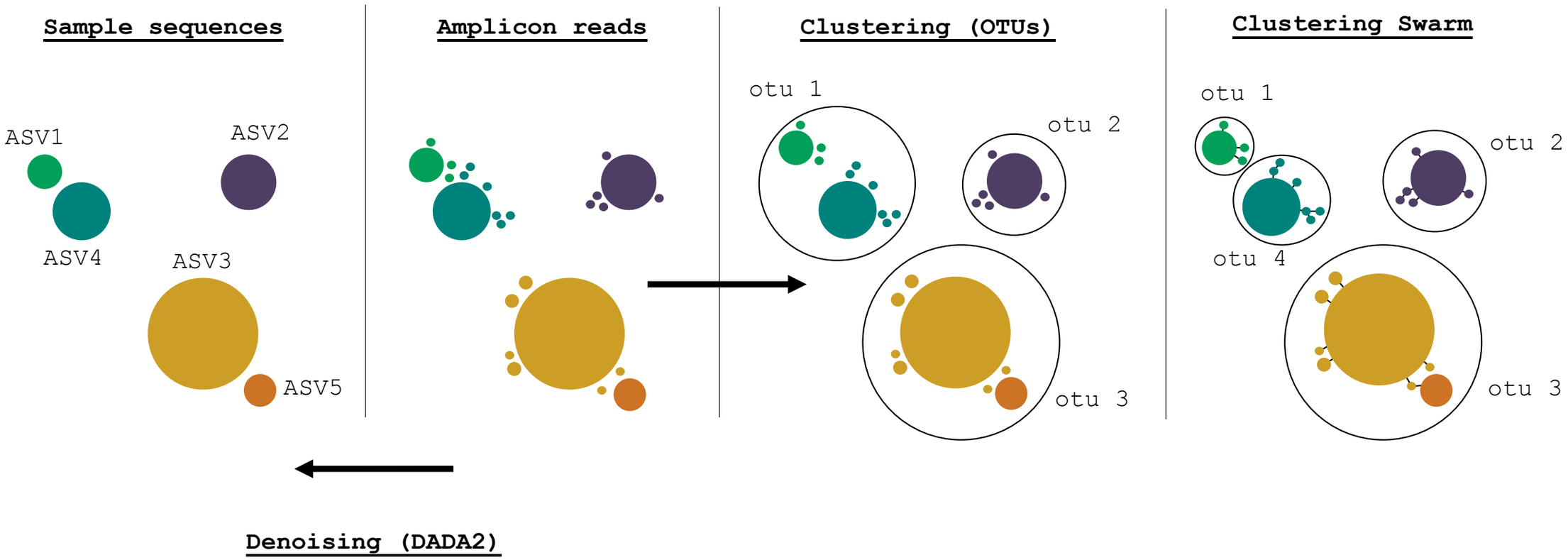
# 2/ Denoising




A more recent approach : **denoising** (e.g. DADA2, Deblur, UNOISE3)

Try to correct the sequencing errors directly to retrieve « true » sequences - or Amplicon Sequence Variants (ASV)

# 2/ Denoising




However, 2 different ASVs can represent the same species (different copies of the same gene)

# Clustering or denoising ?

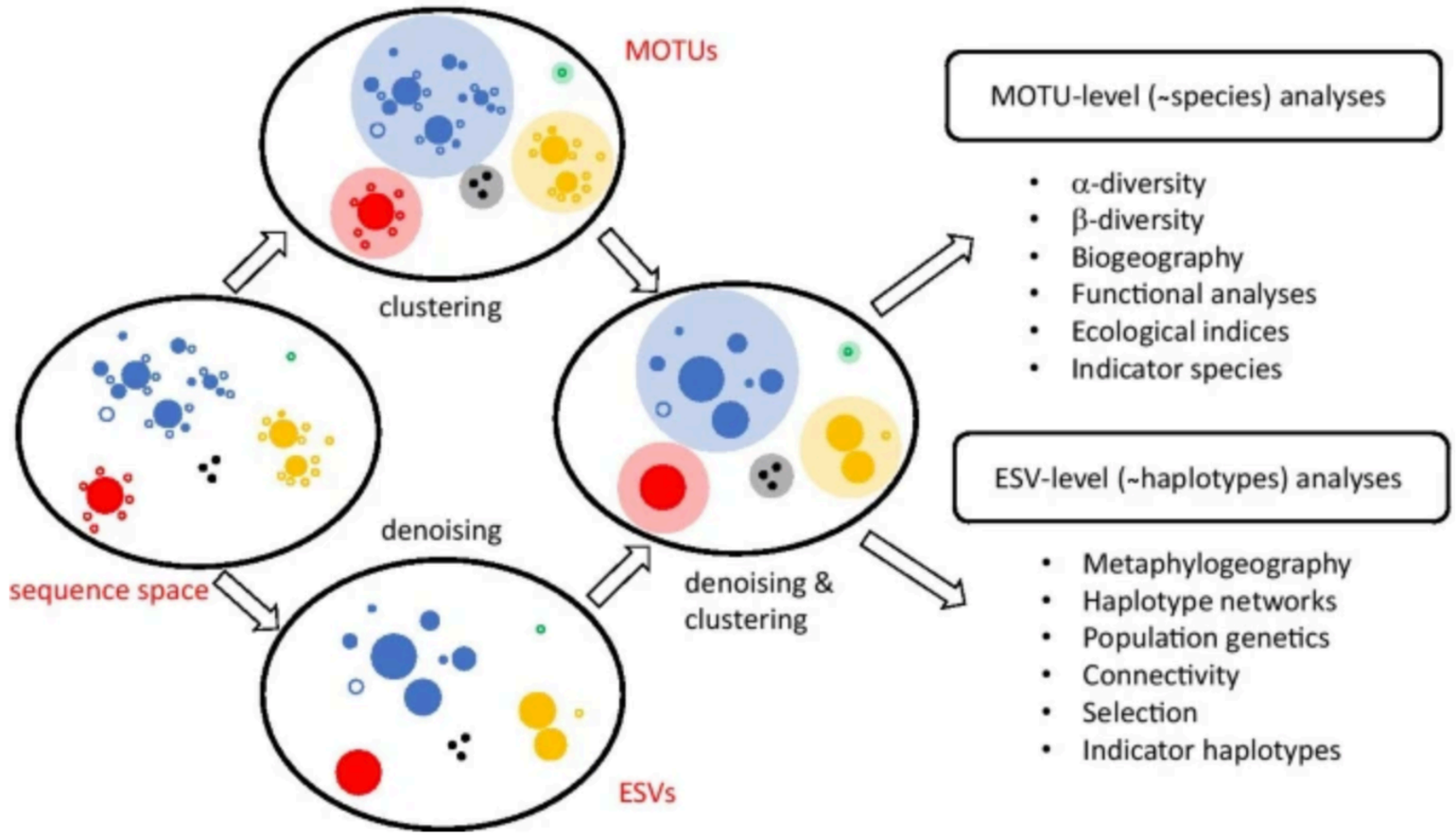
With ASVs you better discriminate ecological patterns but you can also be too efficient (the same genome can contain multiple ASVs as it can have multiple copies of the targeted genetic locus)

With OTUs you can reduce this variability but with a drawback of missing ecologically important variants.

+ reconstructing OTUs is dataset-dependent - if you analyse more samples, you need to re-run entirely the analysis which is not the case with DADA2 as you infer exact variants.

# Clustering or denoising ? It will depend on the question

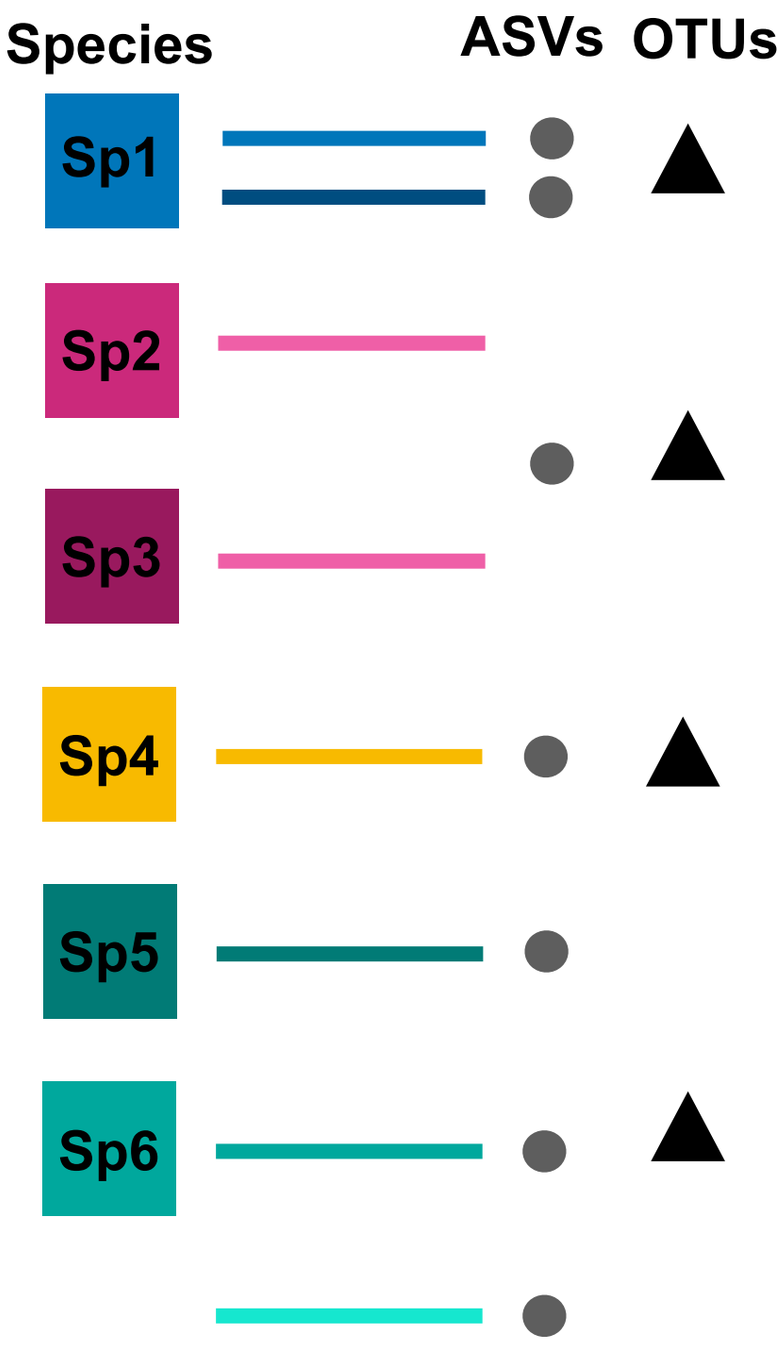
**Fig. 1**




Antich et al., 2021

**For microbial organisms (protists, phytoplankton, bacteria), denoising with DADA2 is now the most used approach.**

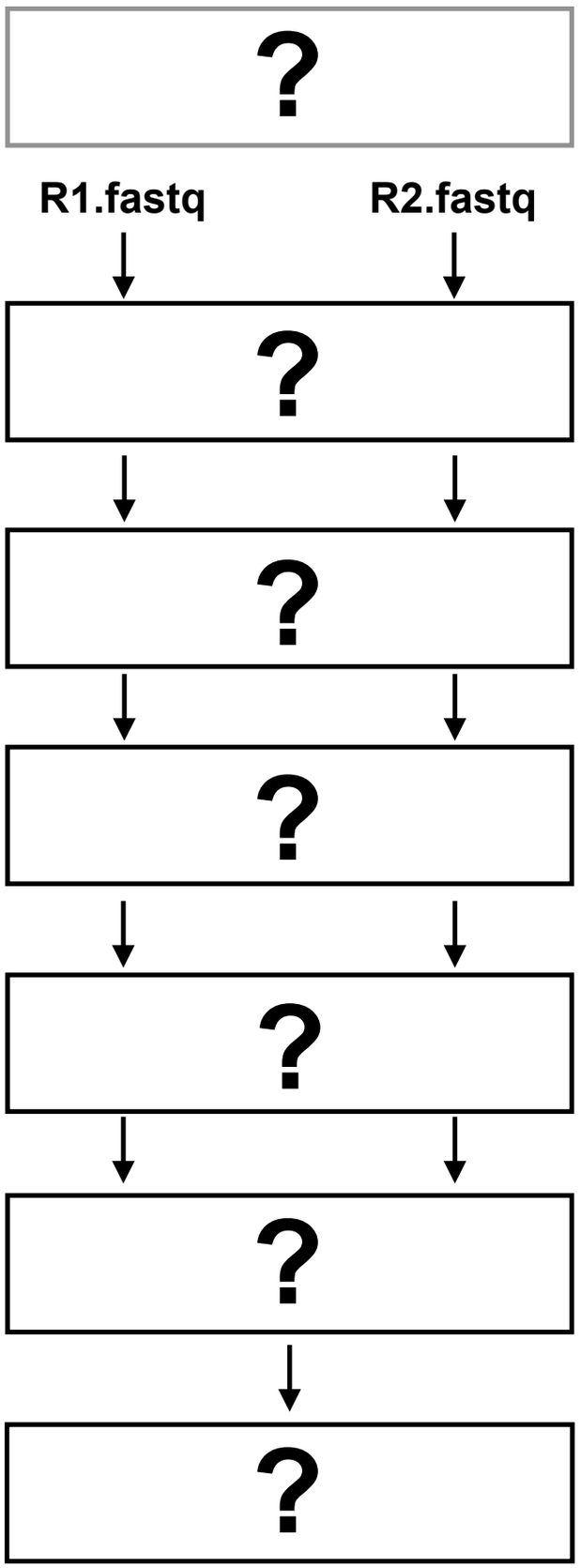
# OTUs/ASVs are not species



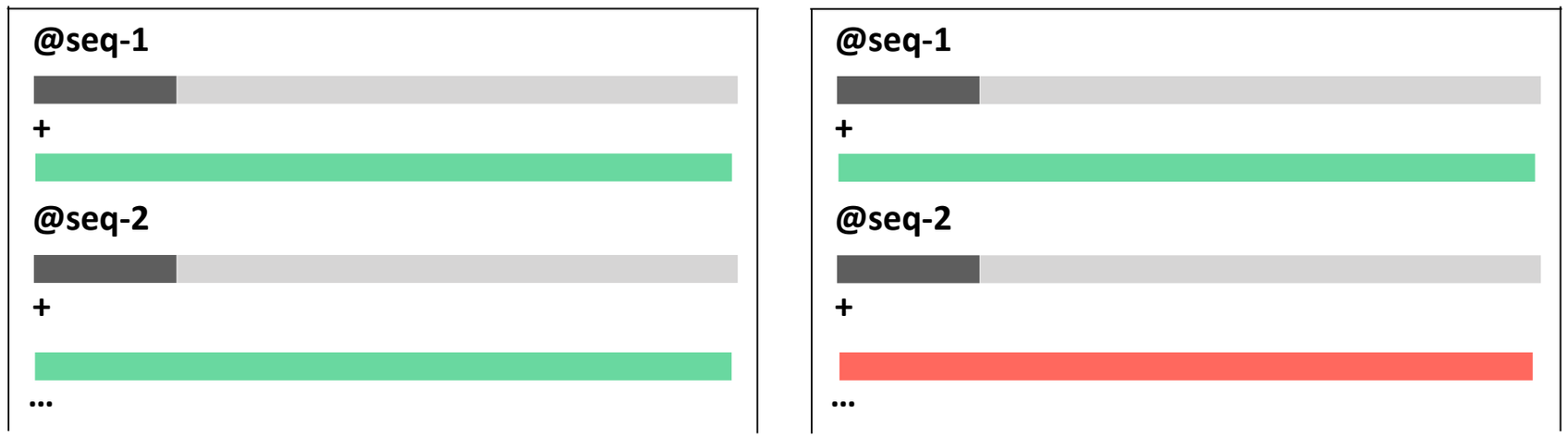

One sequence does not mean one species

Whatever approach you use you don’t have species at the end, but operational taxonomic units (OTUs) or amplicon sequences variants (ASVs).

# BIOINFORMATIC PIPELINE - the steps

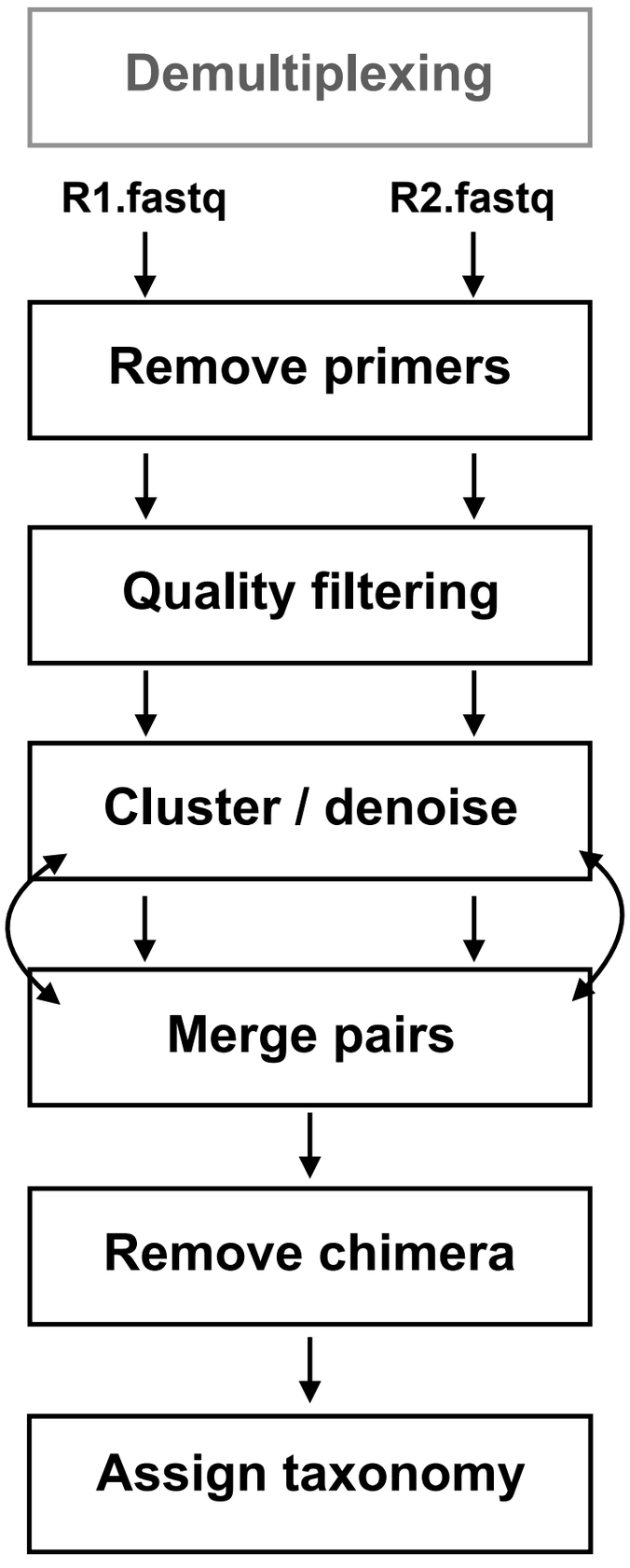



**Typical bioinformatic pipeline for metabarcoding data analysis**




**Can you guess some of the steps?**

# Bioinformatic pipeline




*Done by sequencing platform or with tools (e.g. demultiplexer)*

*Need to know their exact sequence*

*Remove sequences with low quality scores*

*To make sens of all these slightly different reads*

*To have the full barcode*

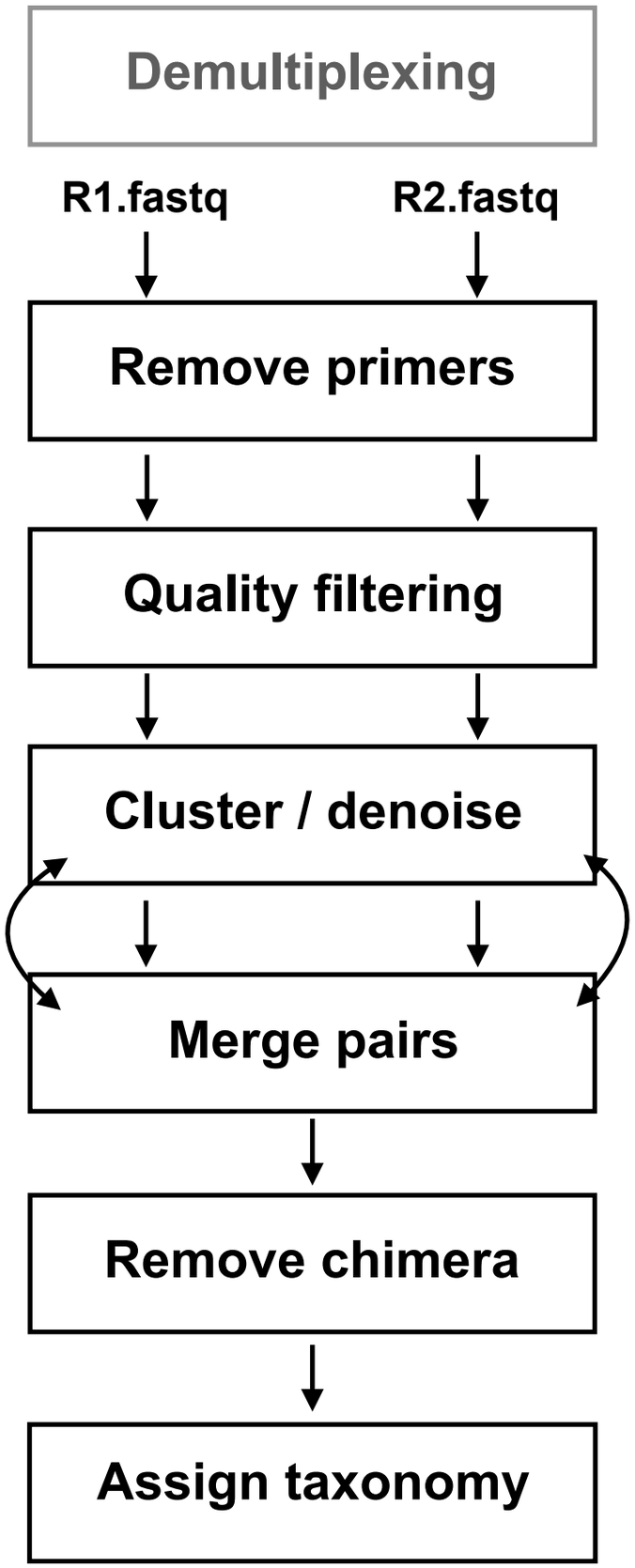
*Use an algorithm to detect chimera and then remove them*

*Using a dedicated database*

**Typical bioinformatic pipeline for metabarcoding data analysis**

# Bioinformatic pipeline

**Typical bioinformatic pipeline for metabarcoding data analysis**




*Done by sequencing platform or with tools (e.g. demultiplexer)*

*Need to know their exact sequence*

*Remove sequences with low quality scores*

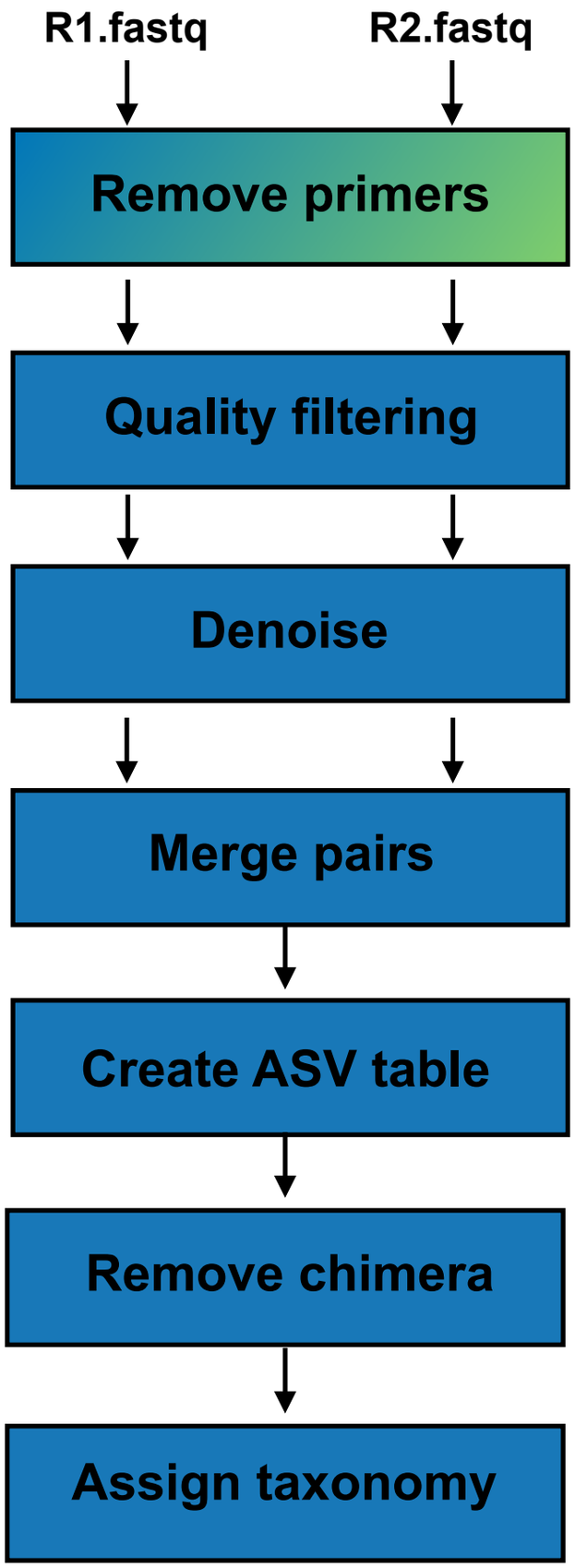
*To make sens of all these slightly different reads*

*To have the full barcode*

*Use an algorithm to detect chimera and then remove them*

*Using a dedicated database*

# DADA2 pipeline




*removePrimers()*
But cutadapt is better

*filterAndTrim()*

*learnErrors()*

*dada()*

*mergePairs()*

*makeSequenceTable()*

*removeBimeraDenovo()*

*assignTaxonomy()*

marcelm/**cutadapt**

Cutadapt removes adapter sequences from sequencing reads

ANY QUESTIONS ?

**Informations you need before running the pipeline :**

- Primers sequences / or size
- Barcode expected length
- Dedicated reference library for your barcode.

They will be essential for some command lines and also to check if all the steps went well.

Remember : it is one thing to run the pipeline, it is another one to be confident of what you have at the end !

Do not take any result as acquired. You need some checking point to be sure of what you have at the end.

# Preparation for Illumina sequencing


*Index (e.g ATTCG)*

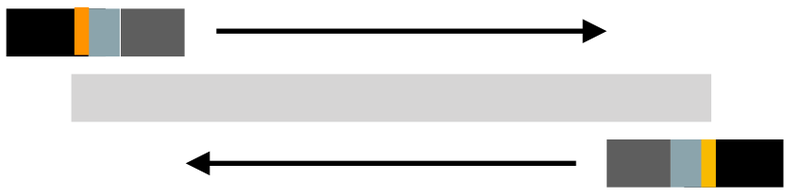
*Illumina adapters + sequencing initiations sites*

*Index (e.g CGAAG)*

**Two main way to do it :**

## 1 PCR step

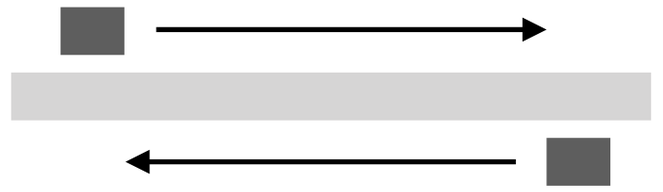
Primers already have everything



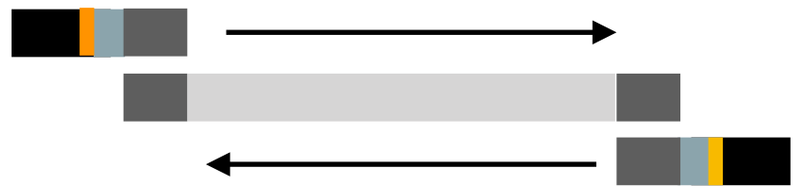
*PCR is done with a different combination of primers for each sample*

## 2 PCR steps

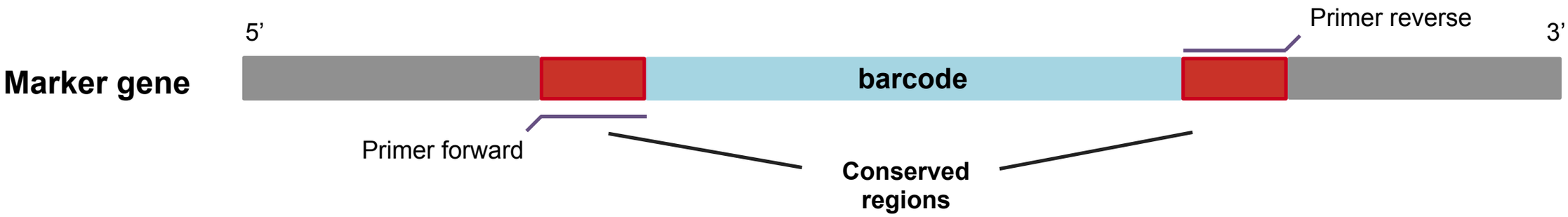
PCR1 : with « simple » primers



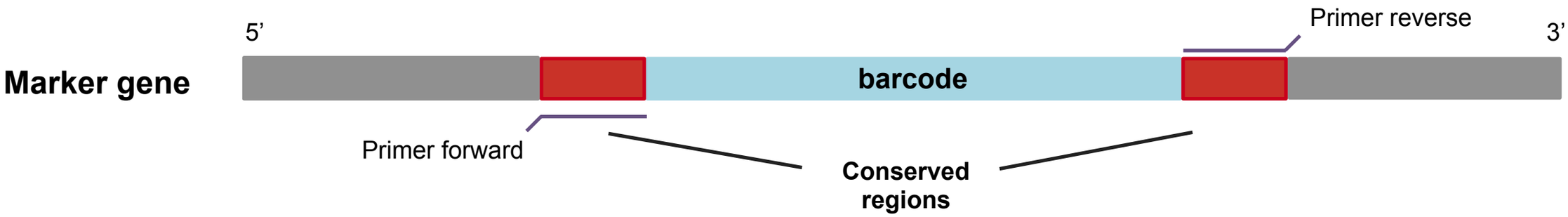
PCR2 : ligate adapters + index



# Primers : some reminders

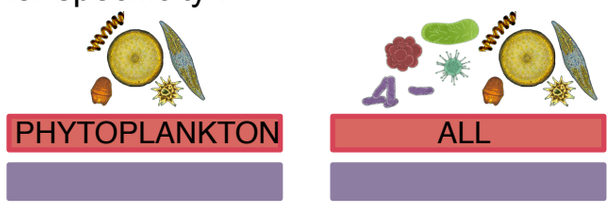


# Primers : some reminders

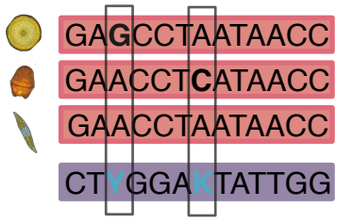



**For wide community analysis, they are often degenerated - used in multiple versions**

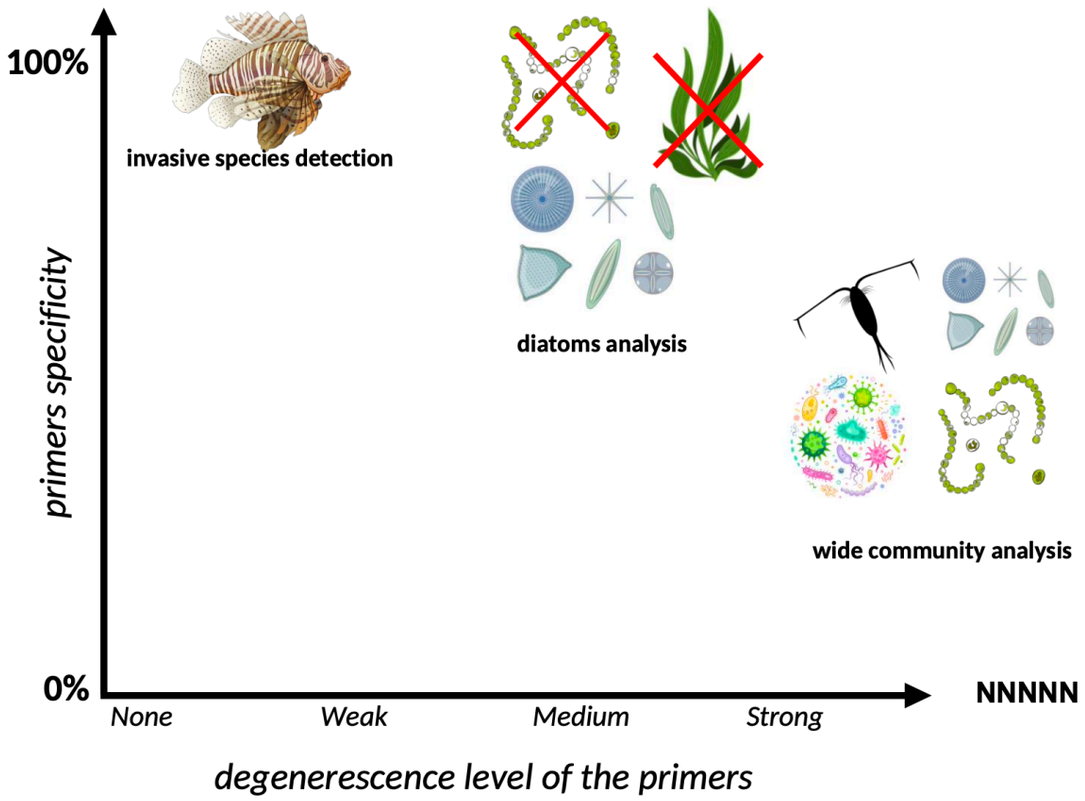
Primer specificity :



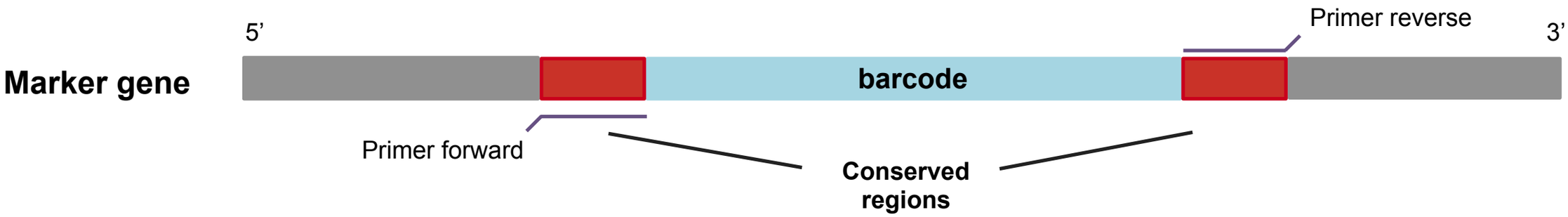

Primer degeneresccence :




| Code | Description |
|---|---|
| M | AC |
| R | AG |
| W | AT |
| S | CG |
| Y | CT |
| K | GT |
| V | ACG |
| H | ACT |
| D | AGT |
| B | CGT |
| N | ACGT |

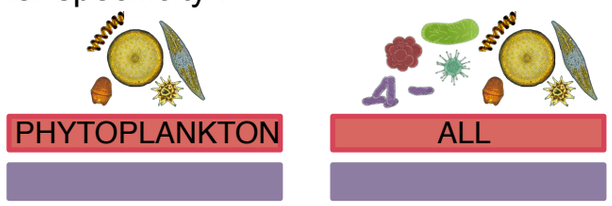


# Primers : some reminders

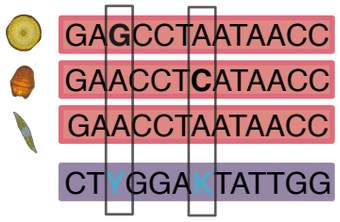



**For wide community analysis, they are often degenerated - used in multiple versions**

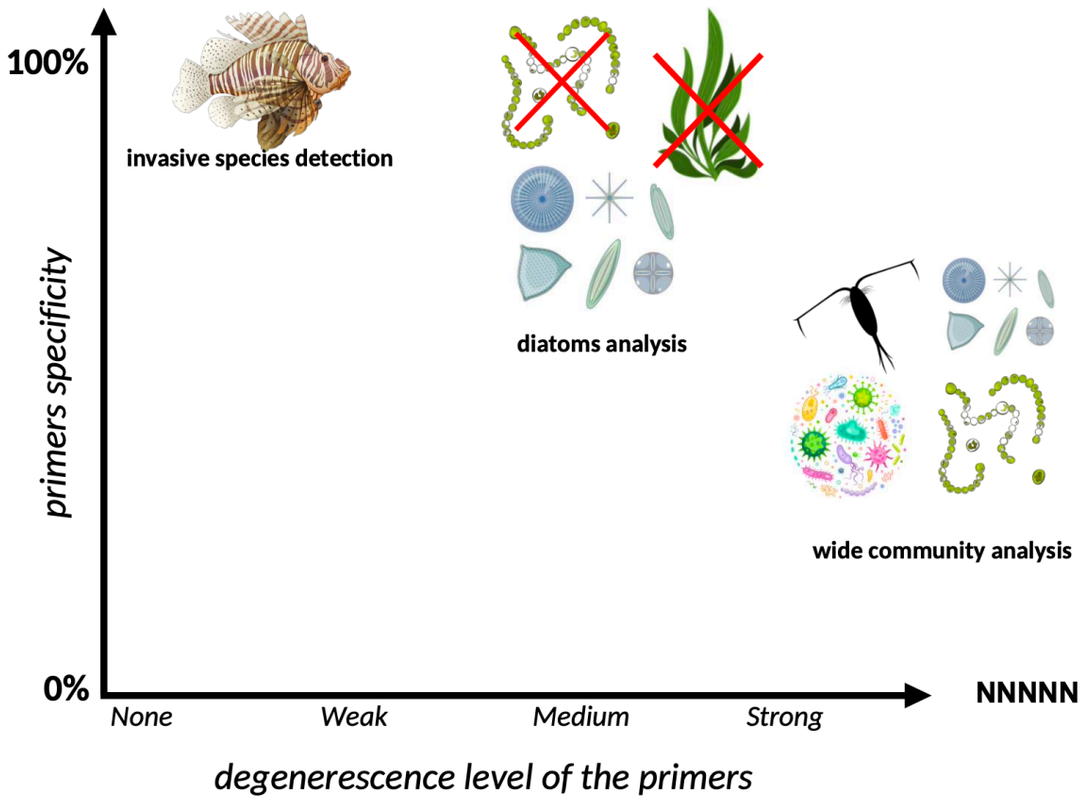
Primer specificity :



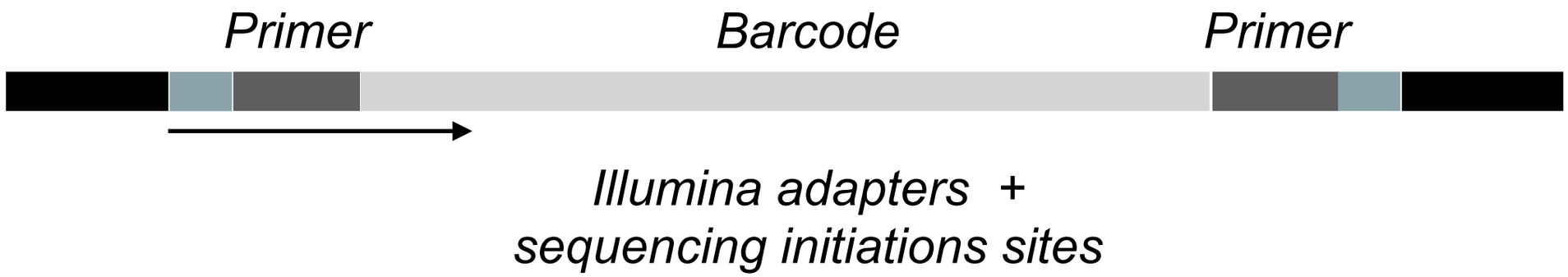

Primer degeneresccence :



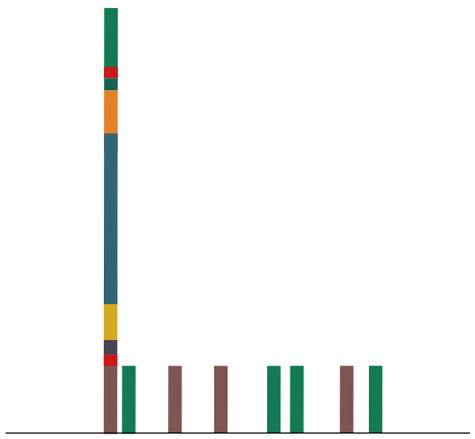

| Code | Description |
|---|---|
| M | AC |
| R | AG |
| W | AT |
| S | CG |
| Y | CT |
| K | GT |
| V | ACG |
| H | ACT |
| D | AGT |
| B | CGT |
| N | ACGT |



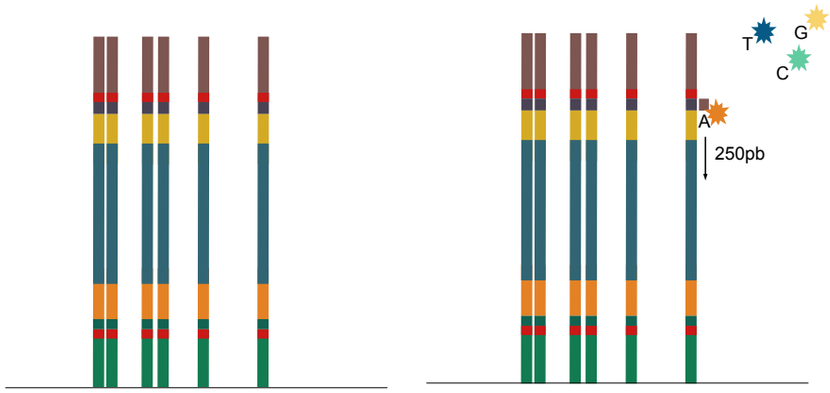
# Preparation for Illumina sequencing



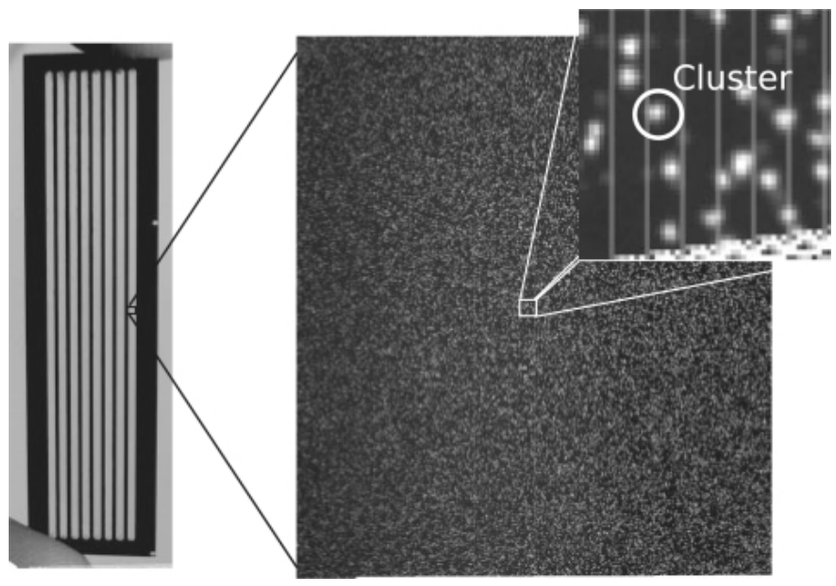

## 1. Illumina adapters and sequencing initiation sites



**Fixation in the flow cell**



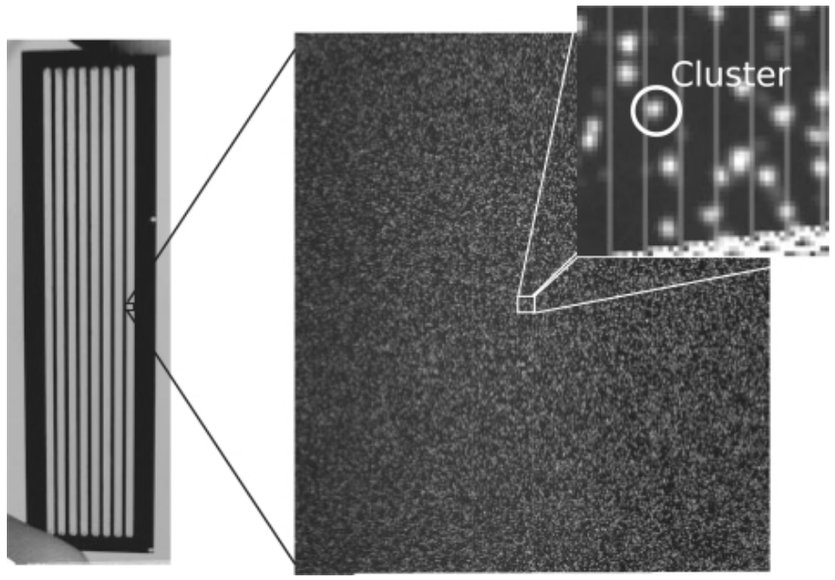

**Sequencing**



# Preparation for Illumina sequencing


## 2. Index for multiplexing

### (for MiSeq, up to 96 samples)




*~ 10 000 000 reads (per Ilumina MiSeq run)*