

# Day 1

Introduction to Bash scripting  
Decision tree  
Quality filtering WGS data  
Genome assembly



# Welcome!

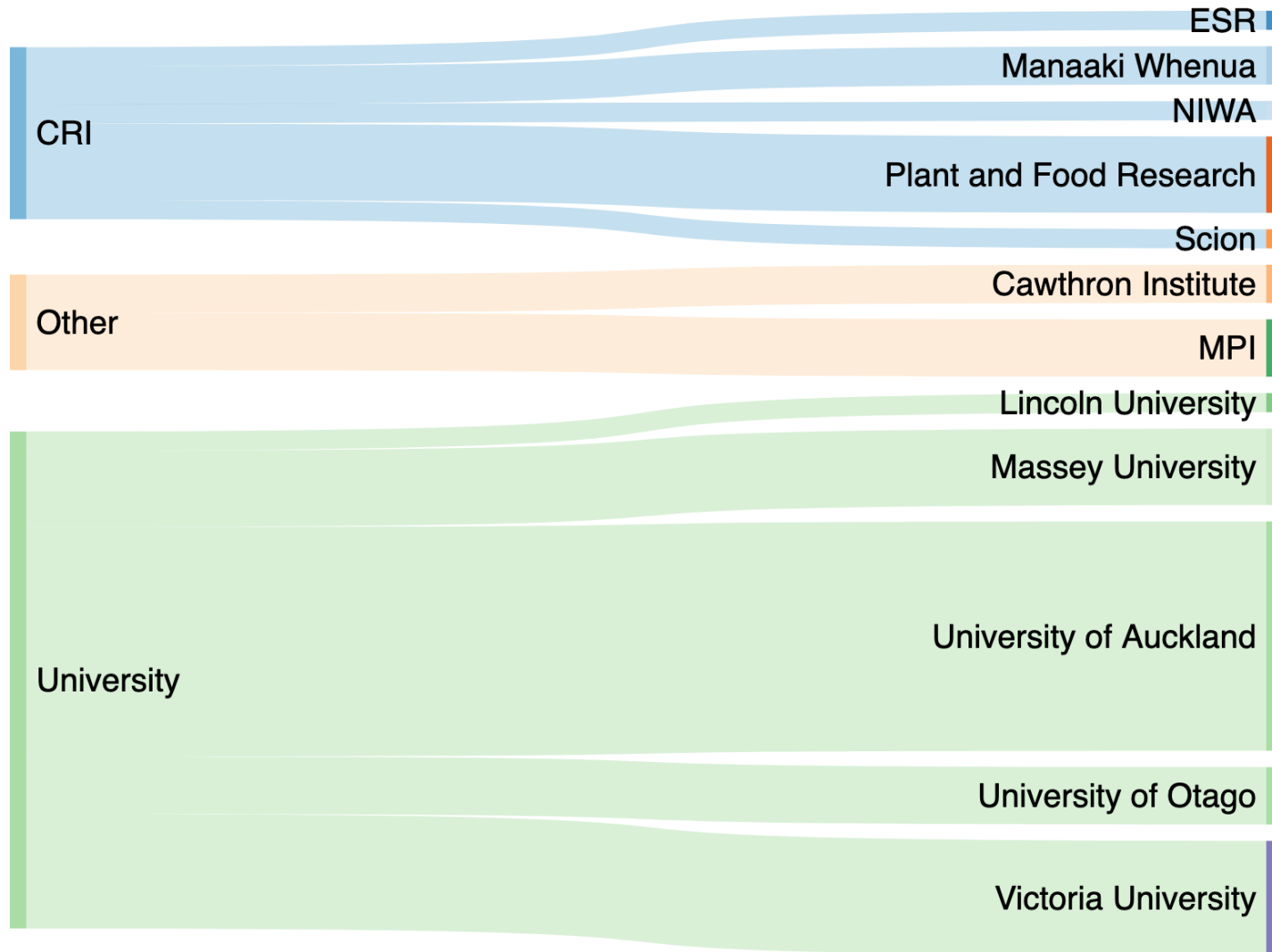
---

- **Housekeeping**
- **Overview of attendees**
  - Where are we from?
  - How experienced are we?
- **Any questions?**



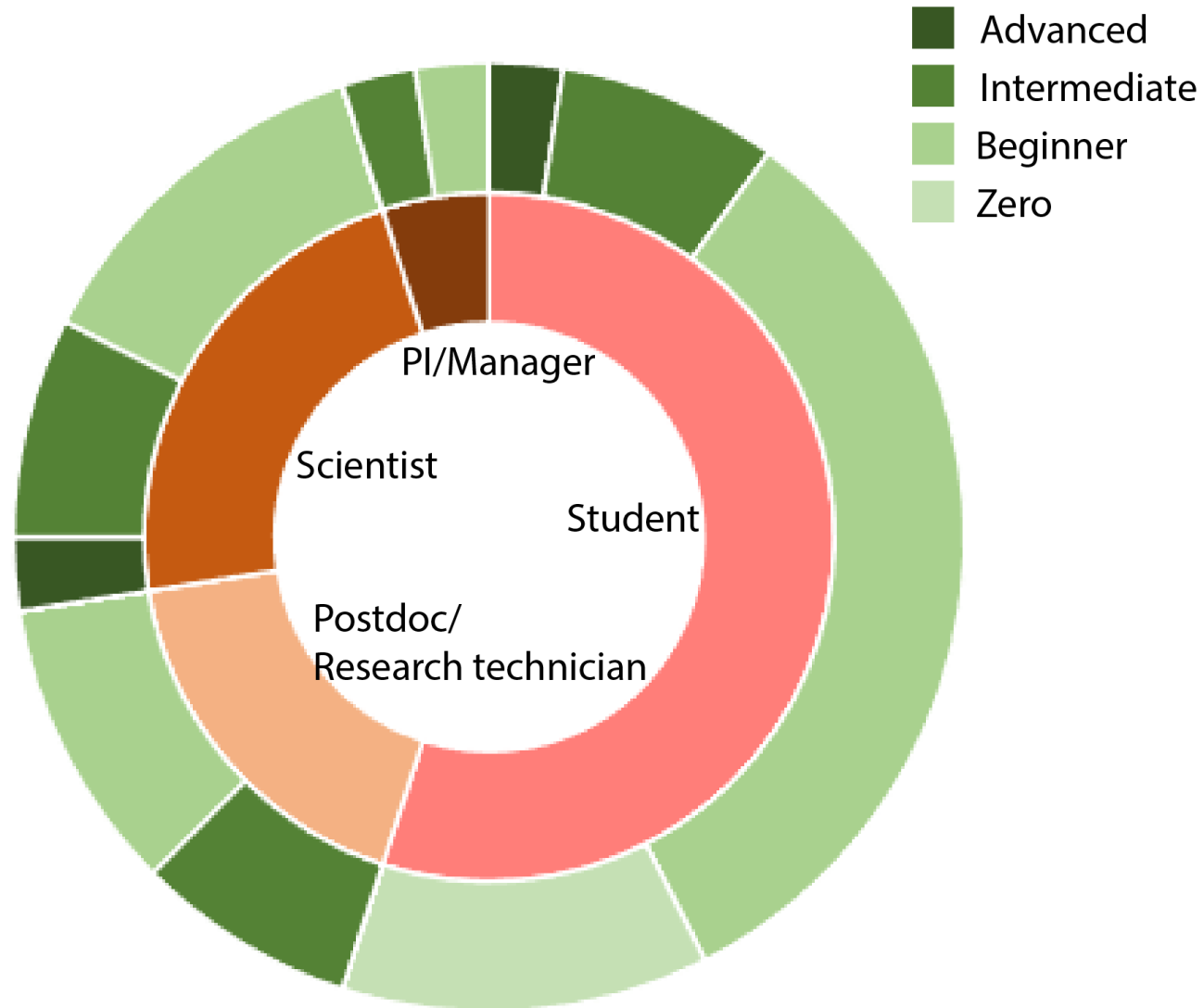
# Where are we from?

---



# How experienced are we?

---



# Genomics Aotearoa - Resources

---

## Genomics Aotearoa – GitHub repositories

<https://github.com/GenomicsAotearoa/>

- Metagenomics Summer School material
- RNA seq workshop
- Environmental metagenomics
  - Metagenomic annotation and binning
- Methods and musings
  - Bin cluster refinement
  - Genome assembly ont
  - Metagenomic ont



# Structure of workshop

## Workshop exercises

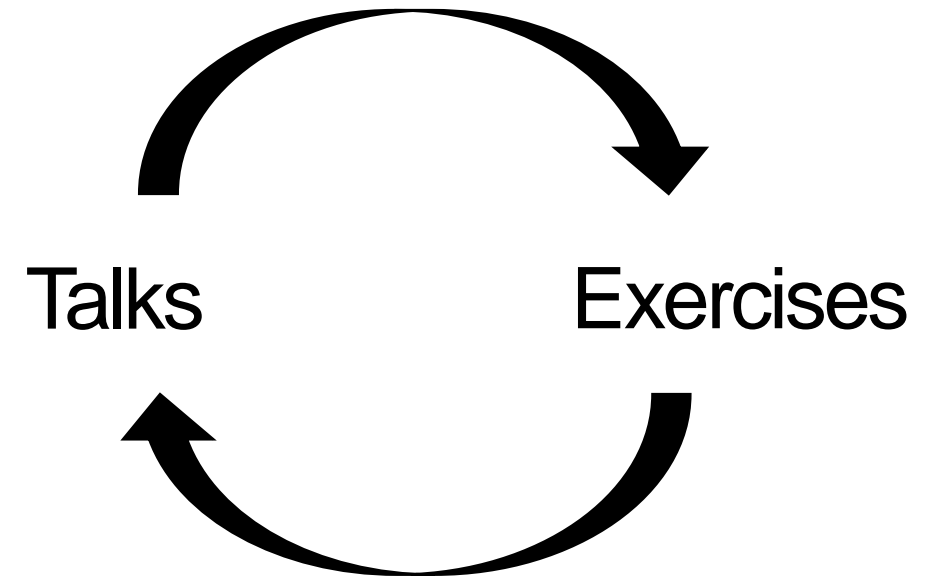
---

### Day 1

1. Bash scripting
2. Quality filtering raw reads
3. Assembly (part 1)
4. Assembly (part 2)
5. Evaluating the assembly

### Day 2

1. Binning (part 1, read mapping)
2. Binning (part 2, initial binning)
3. Binning (part 3, dereplication)
4. Bin refinement



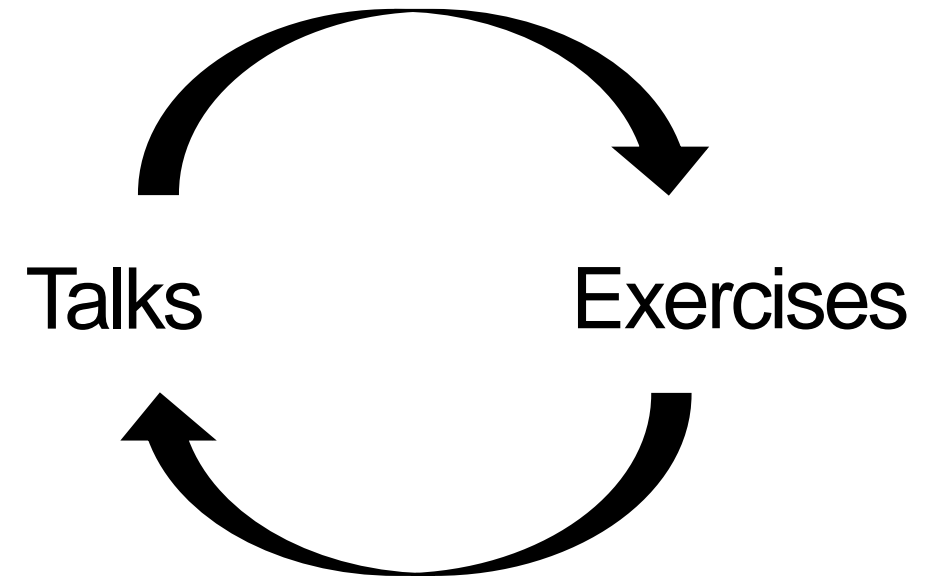
# Structure of workshop

## Day 3

1. Viruses
2. Coverage and Taxonomy
3. Gene prediction
4. Gene annotation (part 1)
5. Gene annotation (part 2)

## Day 4

1. Gene annotation (part 3)
2. Presentation of data



# Starting each session

1. Log in to the NeSI Jupyter hub via a browser
2. Also log in to NeSI via a standard terminal
3. Open the workshop exercise materials on GitHub
4. *Optional: Open a text editor such as Notepad or Word*





# Bash scripting

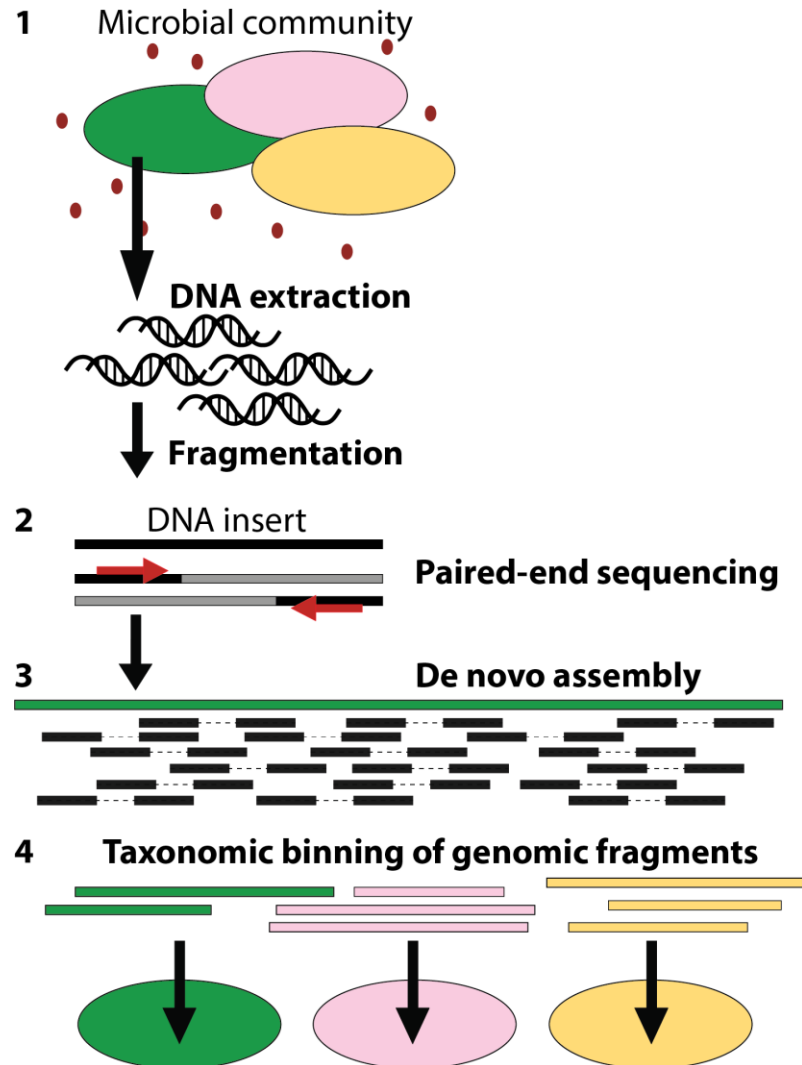


# Metagenomic decision tree(s)



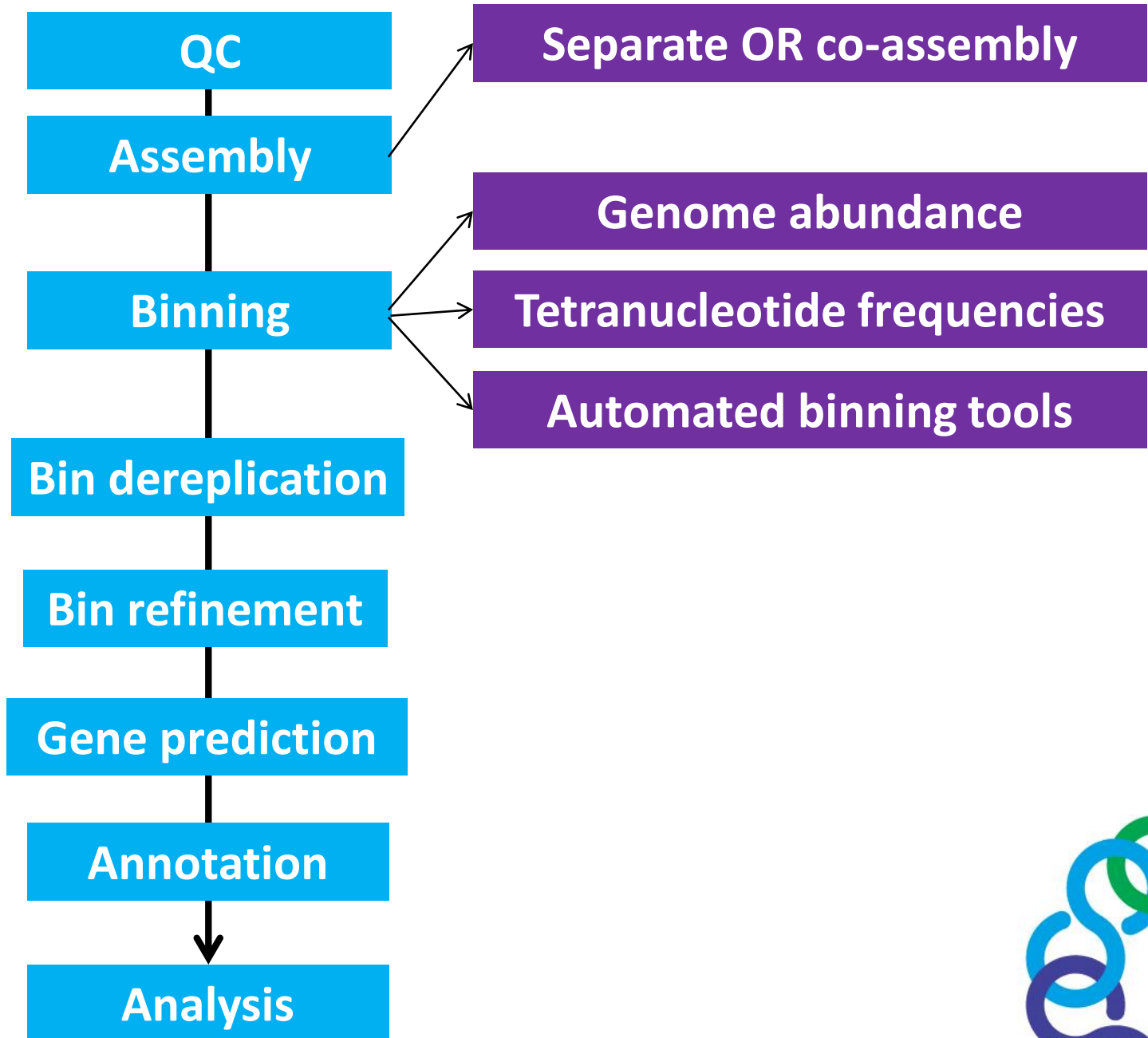
# Our goal: genome recovery

---



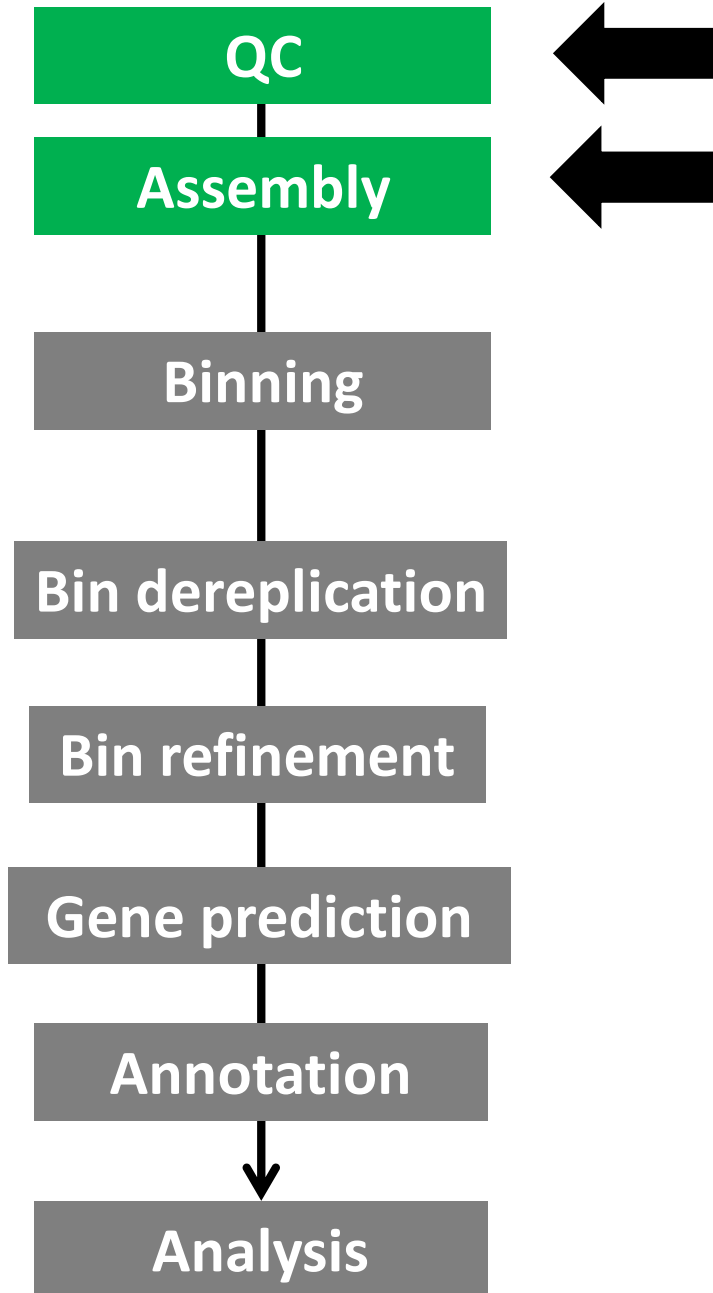
# Overview

---



# Overview

---



# Decision tree

---

- Starts with experimental design
- DNA extraction
- WGS library prep
- Amount of sequencing



Samples/\$\$\$

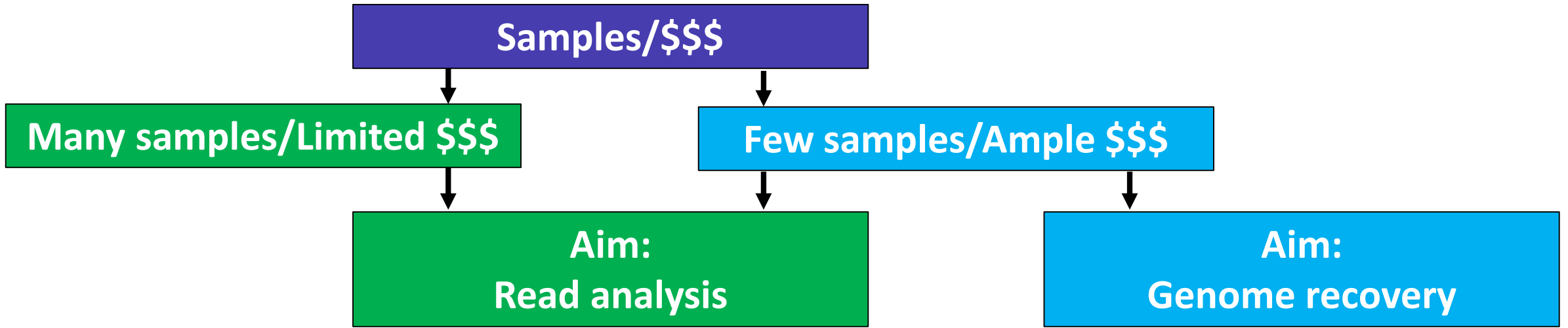


Many samples/Limited \$\$\$



Few samples/Ample \$\$\$







Samples/\$\$\$

Many samples/Limited \$\$\$

Few samples/Ample \$\$\$

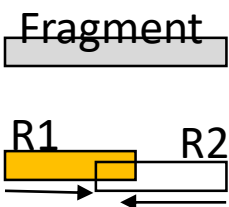
Aim:  
Read analysis

Aim:  
Genome recovery

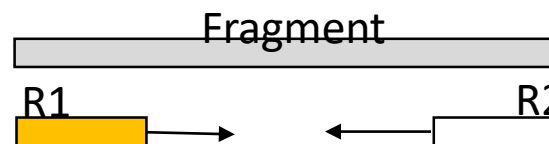
Libraries:  
Short overlapping PE inserts

Libraries:  
Longer gapped PE inserts

(e.g. 200 bp DNA fragments  
for 2x125 bp reads)



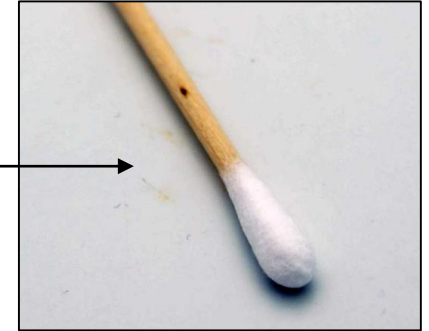
(e.g.  $\geq 550$  bp DNA fragments)



# DNA input

---

- **Very low inputs (e.g. nanograms) for Nextera library prep = enzymatic fragmentation with broad size distributions**



- **High inputs (e.g. 100s ng) for TruSeq = physical fragmentation with defined size selection**



Tends to yield sequences of larger inserts



Samples/\$\$\$

Many samples/Limited \$\$\$

Few samples/Ample \$\$\$

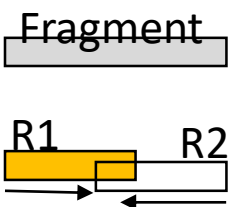
Aim:  
Read analysis

Aim:  
Genome recovery

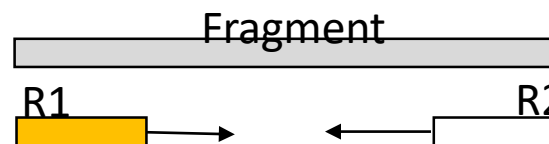
Libraries:  
Short overlapping PE inserts

Libraries:  
Longer gapped PE inserts

(e.g. 200 bp DNA fragments  
for 2x125 bp reads)



(e.g.  $\geq 550$  bp DNA fragments)



**Samples/\$\$\$**

**Many samples/Limited \$\$\$**

**Few samples/Ample \$\$\$**

**Aim:  
Read analysis**

**Aim:  
Genome recovery**

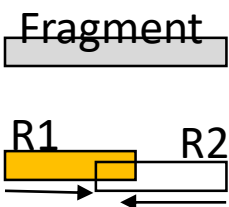
**Libraries:  
Short overlapping PE inserts**

**Libraries:  
Longer gapped PE inserts**

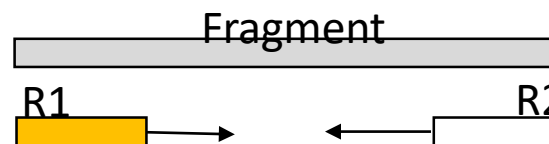
**Sequencing depth:  
Shallow (<10 Gbp)**

**Sequencing depth:  
Deep (e.g. >=10s Gbp)**

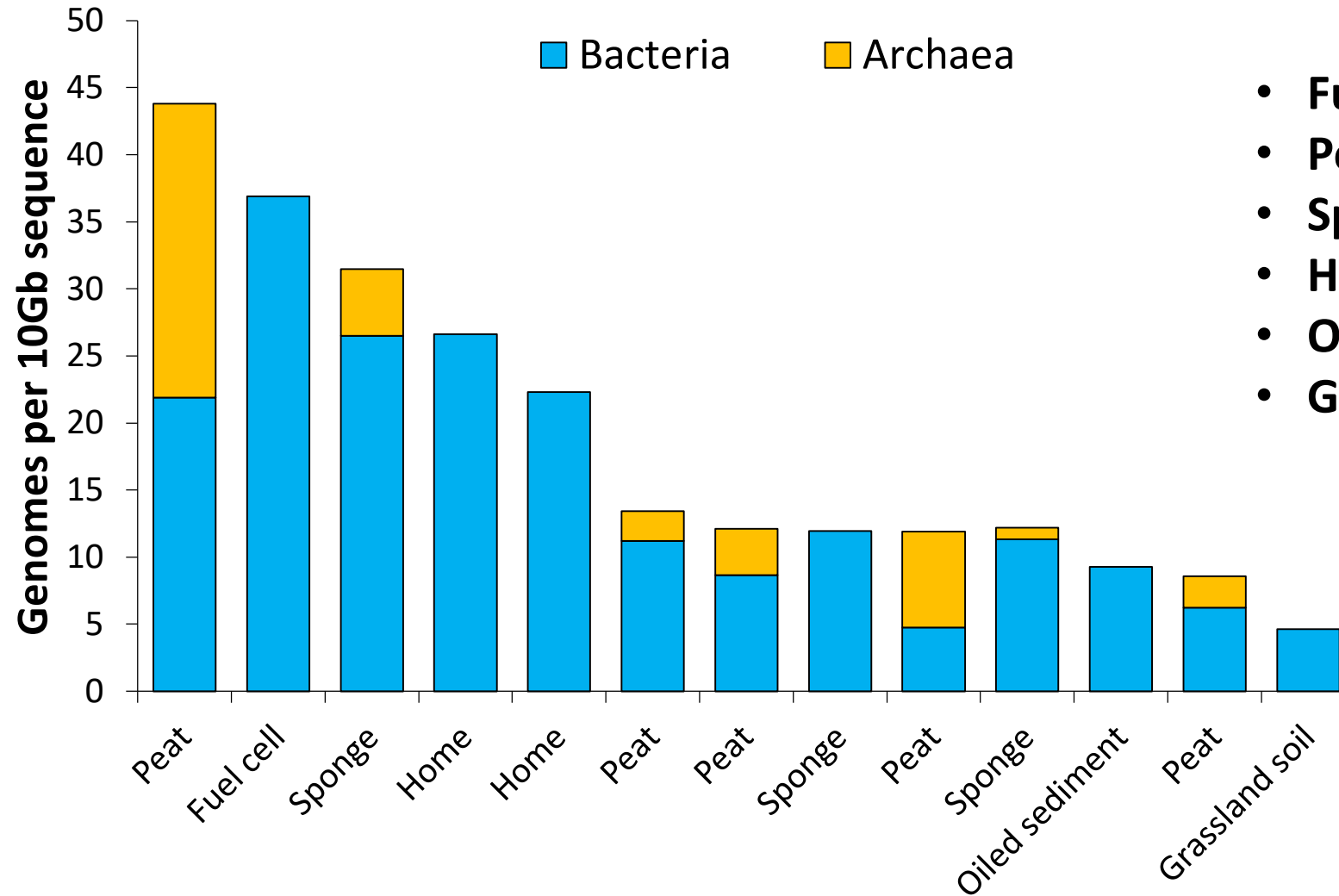
(e.g. 200 bp DNA fragments  
for 2x125 bp reads)



(e.g. >=550 bp DNA fragments)



# Genome recovery per environment

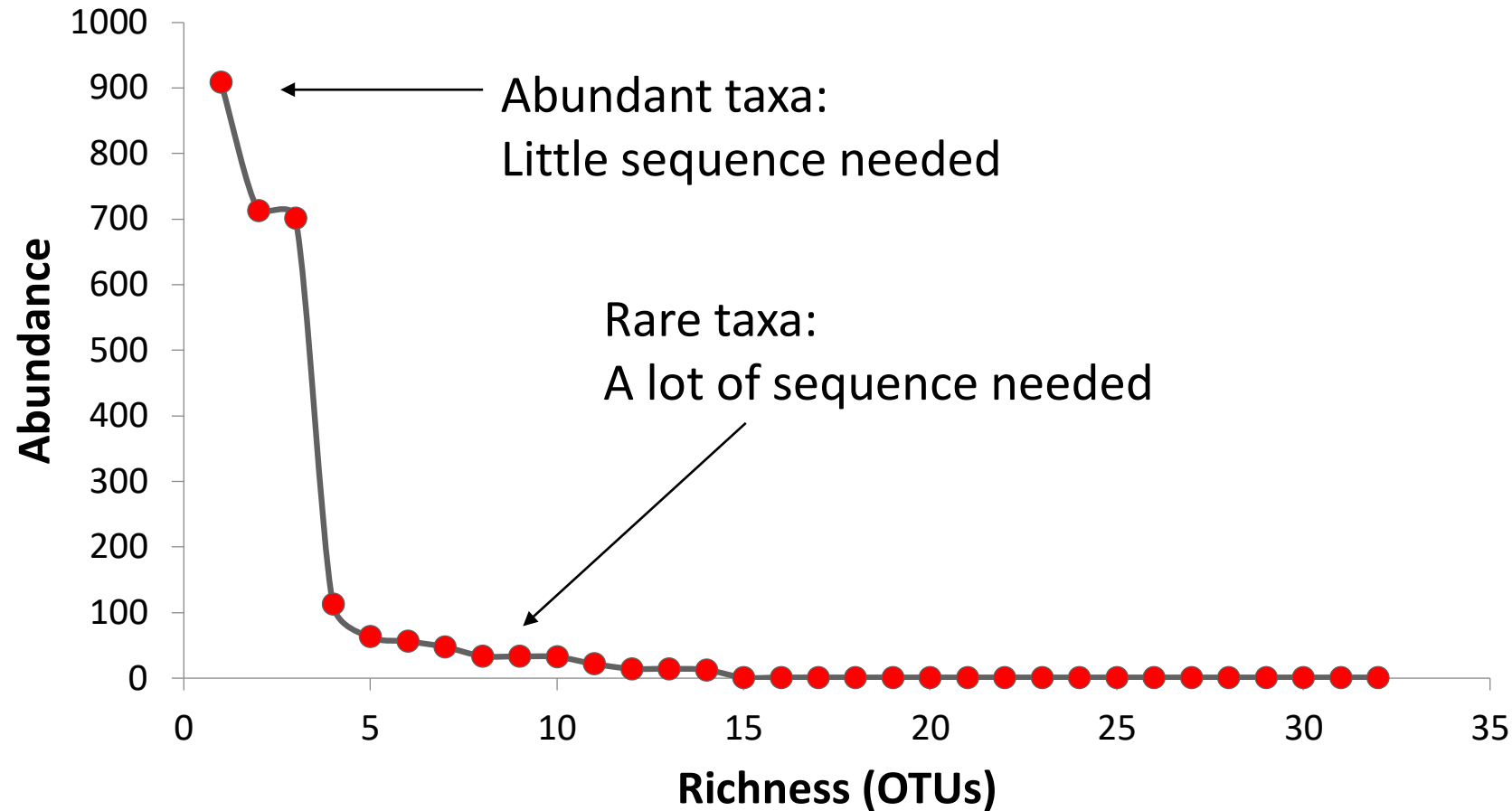


- Fuel cell microbiome
- Peat (boreal)
- Sponge microbiome
- Home microbiome
- Oiled sediment (seafloor)
- Grassland soil

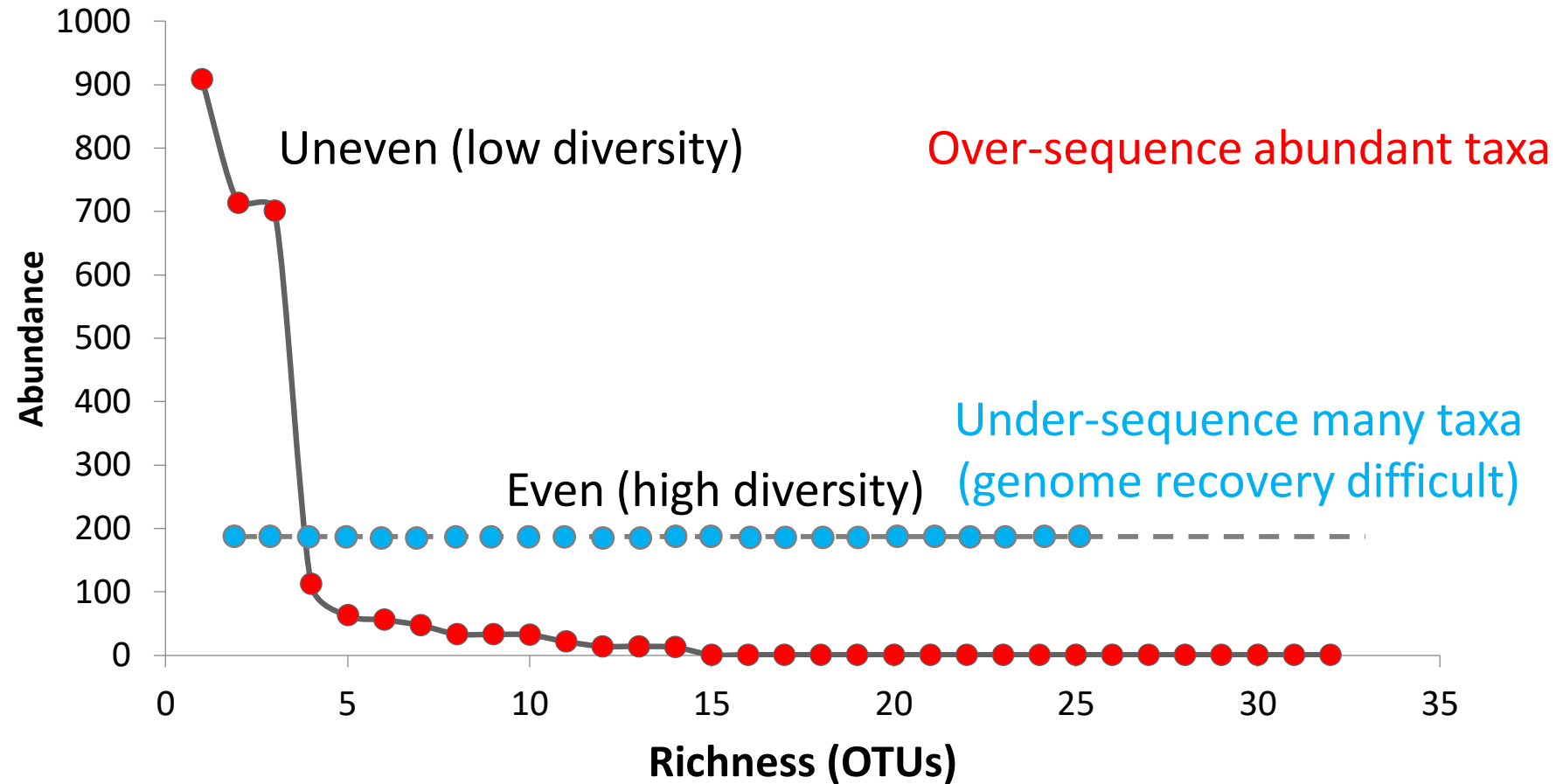


# Estimate sequencing depth

---



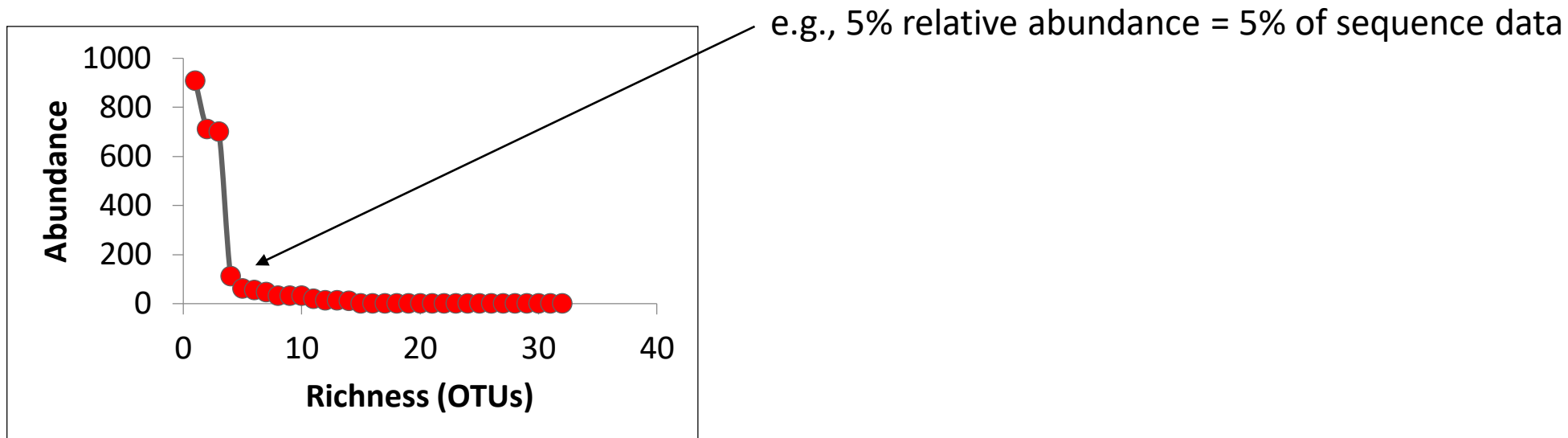
# Community structure matters



# Estimate sequencing depth

---

- Estimate generously
- Determine/guesstimate relative abundance of rarest target organism
- Determine/guesstimate the average genome size
- Factor in larger eukaryote genomes
- Decide the minimum desired coverage (e.g. 30x)





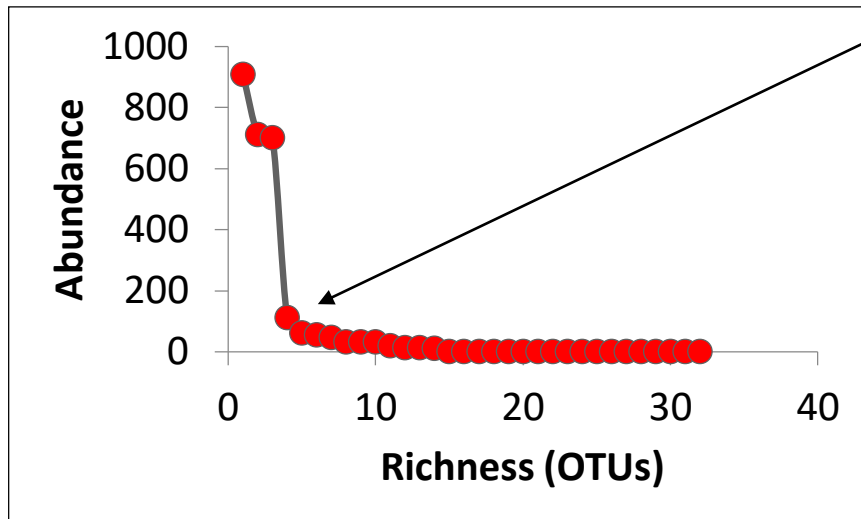
# Prokaryotic genome sizes

---



# Estimate sequencing depth

- Estimate generously
- Determine/guesstimate relative abundance of rarest target organism
- Determine/guesstimate the average genome size
- Factor in larger eukaryote genomes
- Decide the minimum desired coverage (e.g. 30x)



e.g., 5% relative abundance = 5% of sequence data

## Mock parameters:

- Bacterial genome 5 Mbp long
- 5% abundance (need 100/5 or 20x a genome at 100% abundance)
- 30x coverage

$$5 \text{ Mbp} \times 20 \times 30 = 3,000 \text{ Mbp (or 3 Gbp)}$$



# When you have so many genomes

---

You need a:

- Clear goal
- Question
- Hypothesis to test



# Quality control/filtering



# The FastQ data format

---

```
@SEQUENCE_1  
ATCGATCGATCG  
+  
4:<AIIIFI  
@SEQUENCE_2  
AATGATCCATG  
+  
IIIIIIIIII  
@SEQUENCE_3  
TGTGTGACATG  
+  
BBGBBCIFIII
```

Each sequence is represented by four lines

1. Sequence name
2. Sequence content
3. Spacer line (+, or +Sequence name)
4. Quality information



# The FastQ data format

---

- What does the quality score even mean?
  - It represents the probability of a nucleotide position being incorrectly called

$$Q = -10 \log_{10} p$$

Q	p	Prob. correct
0	1	0
10	0.1	0.9
20	0.01	0.99
30	0.001	0.999
40	0.0001	0.9999

*How each Q value is encoded varies between sequencing platforms*

Generally we work with the **Illumina 1.8+** (Phred+33) standard



# Quality filtering WGS data

---

- Remove barcode and adapter regions
- Remove low-quality regions of reads
- Identify potential problems during sequencing
  - Deciphering 'aberrant' metrics in FastQC
  - Adapter read-through



# Quality filtering WGS data

---

- Remove barcode and adapter regions
- Remove low-quality regions of reads
- Identify potential problems during sequencing
  - Deciphering 'aberrant' metrics in FastQC
  - Adapter read-through





# Task: Quality filtering

---

## Visualising data with FastQC

1. Inspecting *fastq* files
2. Identifying regions of concern

## Quality filtering with trimmomatic

1. Removing adapter sequences
2. Removing low-quality regions



# Common issues with WGS data

---

Do I need to remove adapters?



Yes.

I don't know if adapters have been removed or not



Check the per-nucleotide distributions  
You will see 100% skews if they remain.

What's the lowest Q to allow when trimming?



Assembly is a self-correcting process, so  
you can be surprisingly lenient

What if my GC skew is outside of the expected range?



FastQC is calibrated to genome data where you expect GC conservation.  
Metagenomes do not adhere to this assumption

How to interpret  
over-represented  
kmers

What do high  
sequence  
duplication levels  
mean?

Issues when  
working with low-  
input DNA libraries



# Discussion: Filtering out host DNA

---



# Assembly



# Genome assembly

---

## Overlap-Consensus-Layout (OCL) assembly



# Genome assembly

---

## Overlap-Consensus-Layout (OCL) assembly

TTGAAGAGTT

GGCTCAGATT

TTTGATCATG

AAGAGTTTGA

AACGCTGGCG

GATTGAACGC

CTCAGATTGA

TGAAGAGTTT

ACGCTGGCGC

TCATGGCTCA



# Genome assembly

---

## Overlap-Consensus-Layout (OCL) assembly

```
TTGAAGAGTTTGGCTCAGATTGAACGCTGGCGC
TTGAAGAGTT          GGCTCAGATT AACGCTGGCG
          TTTGATCATG          GATTGAACGC
      AAGAGTTTGA          CTCAGATTGAACGCTGGCGC
TGAAGAGTTT  TCATGGCTCA
```

The problem for *de novo* assembly?

$$N. comparisons = \frac{(n)(n-1)}{2} = \frac{(10)(10-1)}{2} = 45$$



# Genome assembly

---

## De Bruijn graph assembly

Break reads into shorter *k*-mers

TTGAAGAGTT  
TTGA  
TGAA  
GAAG  
AAGA  
AGAG  
GAGT  
AGTT

TTGA TGAA GAAG AAGA AGAG GAGT AGTT





# Genome assembly

---

## De Bruijn graph assembly

Identify sequences of shared  $k$ -mers

TTGAAGAGTT

AAGAGTTTGA  
AAGA  
AGAG  
GAGT  
AGTT  
GTTT  
TTTG  
TTGA

TTGA TGAA GAAG AAGA AGAG GAGT AGTT GTTT TTTG TTGA



# Genome assembly

---

## De Bruijn graph assembly

Identify sequences of shared  $k$ -mers

TTGAAGAGTT

AAGAGTTTGA

AAGA  
AGAG  
GAGT  
AGTT  
GTTT  
TTTG  
TTGA

TTGA TGAA GAAG AAGA AGAG GAGT AGTT GTTT TTTG TTGA  
TTGAAGAGTTTGA



# Genome assembly

---

## De Bruijn graph assembly

Problem #1 –  $k$ -mers are short?

TTGAAGAGTTTGTATCATGGCTCAGATTGAACGCTGGCGC  
TTG TTG TTG TGG  
TGA TGA GGC GGC  
GAA GAA  
TCA TCA CGC CGC



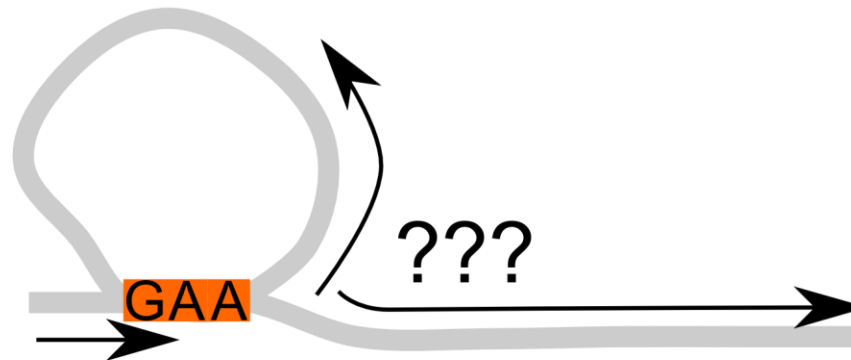
# Genome assembly

---

## De Bruijn graph assembly

Problem #1 –  $k$ -mers are short?

TTGAAGAGTTTGATCATGGCTCAGATTGAACGCTGGCGC



# Genome assembly

---

## De Bruijn graph assembly

Problem #2 –  $k$ -mers are long?

TTGAAGAGTT  
TTGAAGAG  
TGAAGAGT  
GAAGAGTT

AAGAGTTTGA  
AAGAGTTT  
AGAGTTTG  
GAGTTTGA

TTGAAGAG TGAAGAGT GAAGAGTT

AAGAGTTT AGAGTTTG GAGTTTGA



# De Bruijn graph assembly

---

## We want a range of *k*-mer sizes

- Short *k*-mers yield higher coverage
- Long *k*-mers assemble longer contigs (jump repeat regions)

## Other considerations for picking *k*-mer sizes

- Size cannot be longer than read length
- Always pick odd *k*-mer sizes
- The more sizes you use, the longer assembly will take

K-mers	N. contigs	Longest contig	N50 >2kbp	L50 >2kbp
21, 33, 55	4,239,806	660,812	6,782	12,906
43, 55, 77, 99, 121	2,519,669	1,022,083	7,990	12,673
21, 43, 55, 77, 99, 121	3,388,682	1,022,083	7,789	13,327



# Which assembler is best?

---

There are three good options

- SPAdes
- IDBA-UD
- MegaHIT



# Which assembler is best?

---

There are three good options

- SPAdes
- IDBA-UD
- MegaHIT

*In conclusion, it can be said that the choice of assembler should depend on the data at hand and on the exact research question asked. Generally, the best assembly is performed by multi k-mer assemblers such as metaSPAdes, Megahit and IDBA-UD. If micro diversity is not a major issue, and the primary research goal is to bin and reconstruct representative bacterial genomes from a given environment, metaSPAdes should clearly be the assembler of choice. This assembler yields the best contig size statistics while capturing a high degree of community diversity, even at high complexity and low read coverage. If micro diversity is however an issue, or if the degree of captured diversity is far more important than contig lengths, then IDBA-UD or Megahit should be preferred.*

Vollmers et al. 2017 ([https://doi.org/ 10.1371/journal.pone.0169662](https://doi.org/10.1371/journal.pone.0169662))





# Which assembler is best?

---

## There are three good options

- SPAdes
- IDBA-UD
- MegaHIT

*In conclusion, it can be said that the choice of assembler should depend on the data at hand and on the exact research question asked. Generally, the best assembly is performed by multi k-mer assemblers such as metaSPAdes, Megahit and IDBA-UD. If micro diversity is not a major issue, and the primary research goal is to bin and reconstruct representative bacterial genomes from a given environment, metaSPAdes should clearly be the assembler of choice. This assembler yields the best contig size statistics while capturing a high degree of community diversity, even at high complexity and low read coverage. If micro diversity is however an issue, or if the degree of captured diversity is far more important than contig lengths, then IDBA-UD or Megahit should be preferred.*

Vollmers et al. 2017 ([https://doi.org/ 10.1371/journal.pone.0169662](https://doi.org/10.1371/journal.pone.0169662))



# What are some key considerations?

---

## Biological

1. What is your hypothesis?
2. What do you want from the data?

## Computational and resource

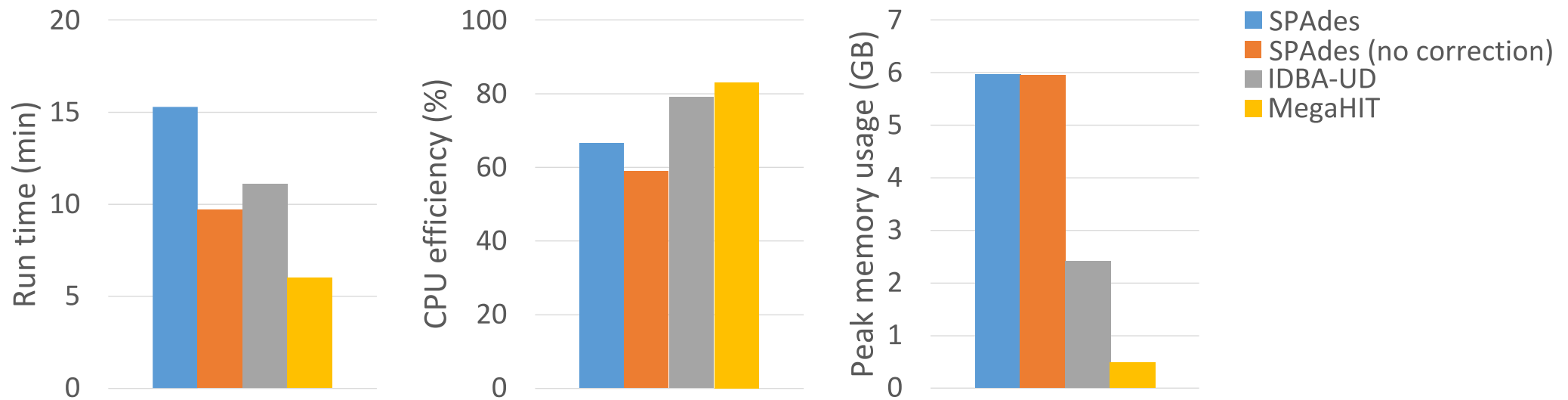
1. How much data do you have?
2. What are your computational resources?
3. What are your time resources?



# Genome assembly

---

What are some key considerations?



# Too much data?

---

- Consider testing sub-samples when coverage is very high, e.g. 100s or 1000s
- Example: abundant groundwater genome at 2000x coverage in full dataset
- Empirical testing of subsample sizes identified assembly sweet spot

~200x coverage



~2000x coverage



# Task: Assembly

---

## **Preparing data for assembly**

1. Learn to prepare input files for SPAdes and IDBA-UD
2. Configure the basic parameters for assembly

## **Perform assembly (assemblies)**

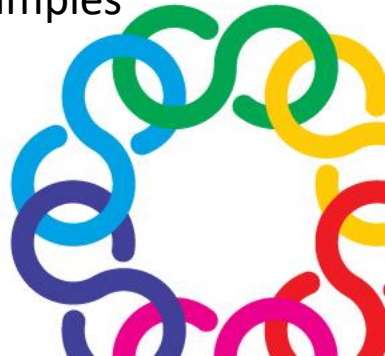
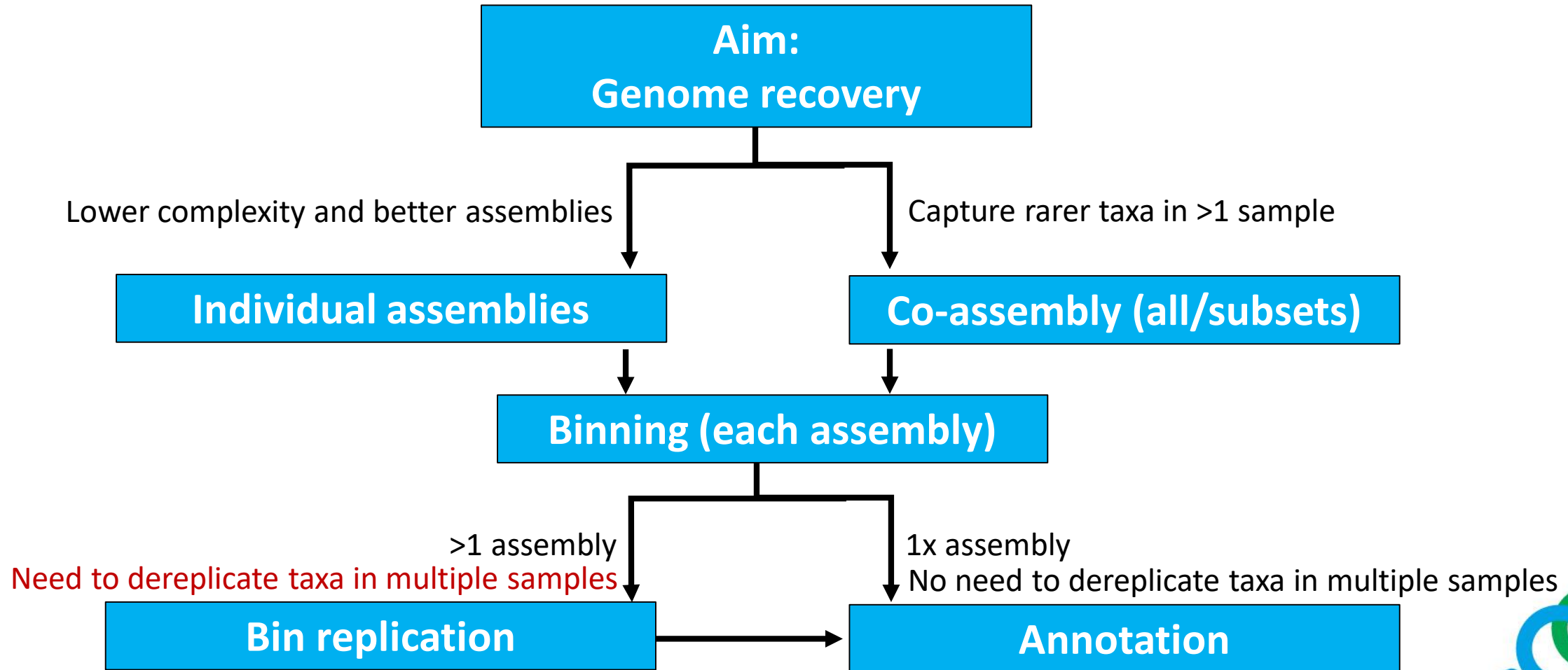
1. Prepare an assembly job to run under slurm
2. Submit several jobs with varying parameters



# **Future considerations and Assembly evaluation**



# Future considerations

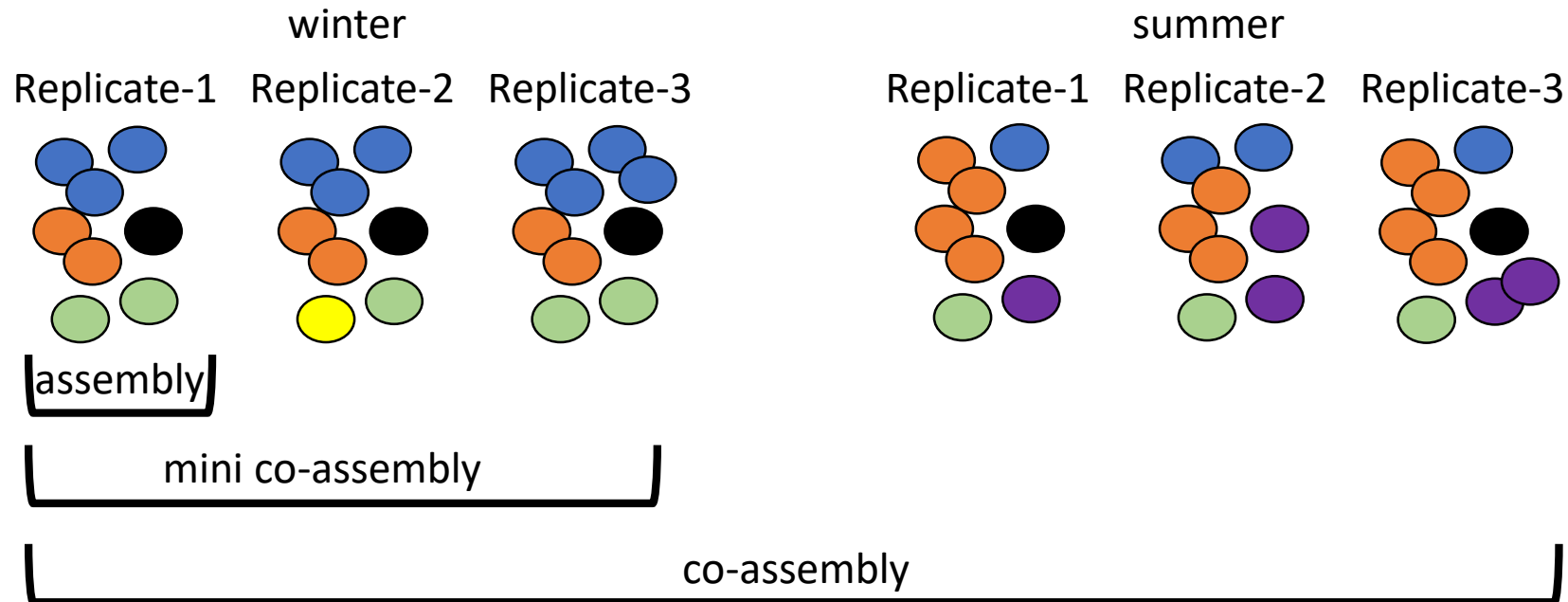


# Future considerations

---

## Assembly options:

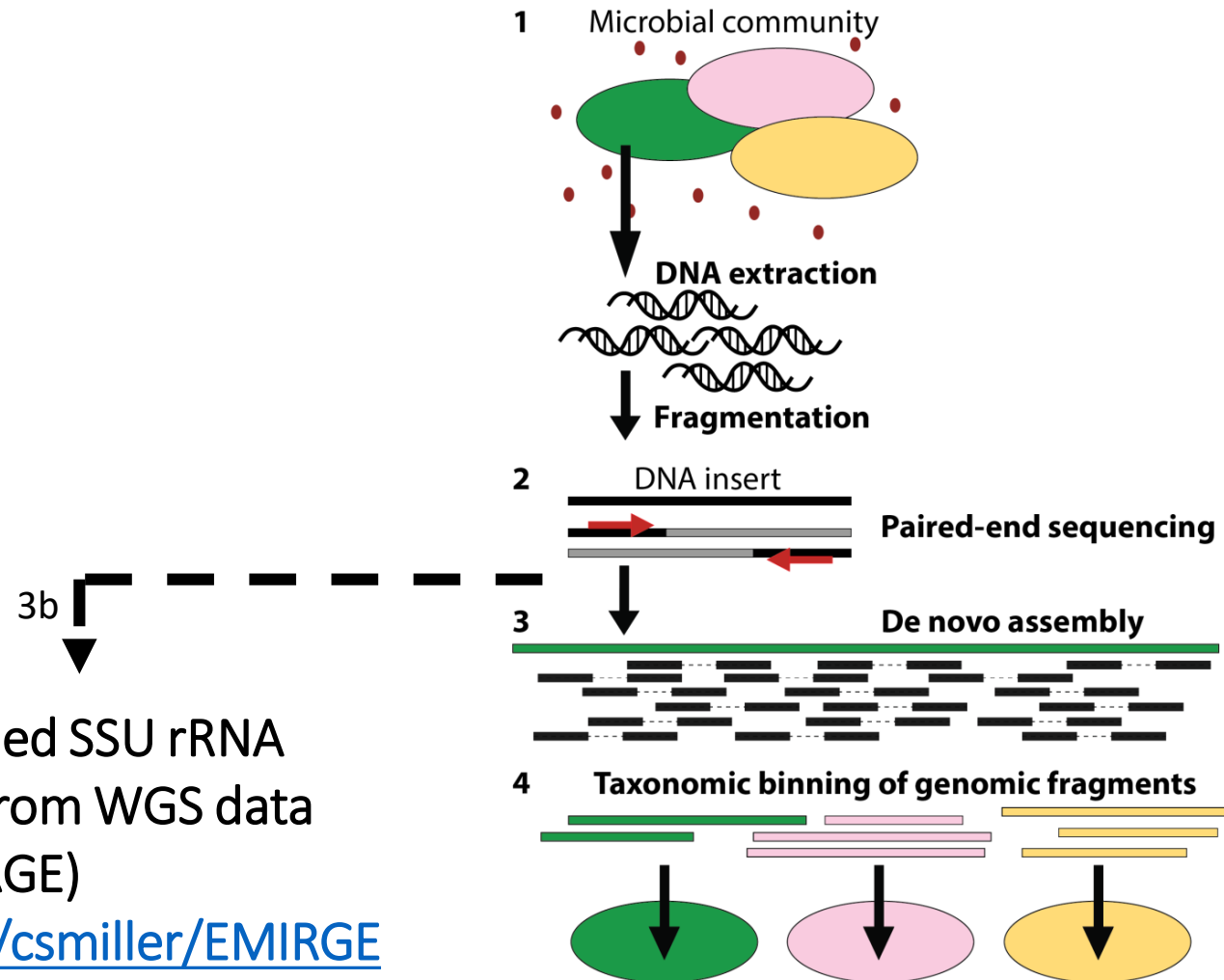
- **Assembly each community separately**
- **Combine reads and assembly all together (co-assembly)**
- **Combine only reads from the same season (mini co-assemblies)**





# Future considerations: rRNA genes

---

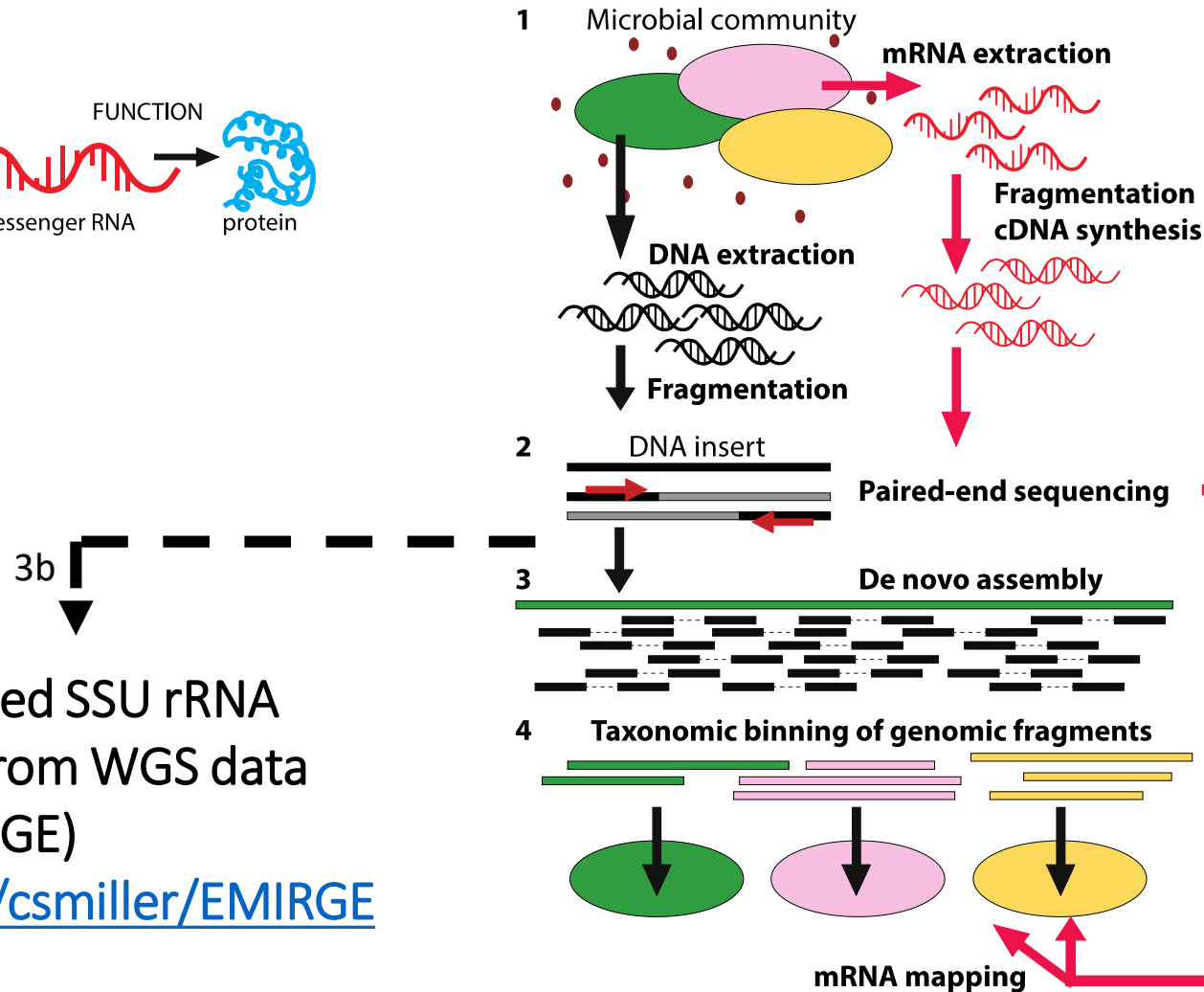
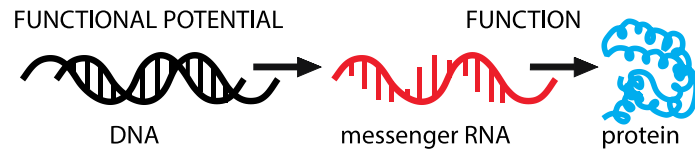


Reference-guided SSU rRNA  
reconstruction from WGS data  
(EMIRGE)

<https://github.com/csmiller/EMIRGE>



# Future considerations: mRNA



Reference-guided SSU rRNA  
reconstruction from WGS data  
(EMIRGE)

<https://github.com/csmiller/EMIRGE>

Metatranscriptomics



# Assembly evaluation

---

Parameters to use in evaluation:

- Total length of contigs (= amount assembled)
- Total length of contigs usable (e.g. >1,000 bp)
- Number of contigs (less is more)
- N50 (minimum contig length at 50% of the total genome length)
- Length distribution of contigs
- Recovery of particular genomes (determined at later stage)



# Task: Assembly evaluation

---

[Go to Github MSS webpage](#)

**Task:**

- Run evaluation tool/script in `ex5_evaluating_assemblies`

