genomics aotearoa

Metagenomics Summer School 2023

Day 1

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

WiFi

Wifi Name: **UoA-Guest-WiFI**

Username: **mgss2023@wifi.com**

Password: **2P15Pr6Z**

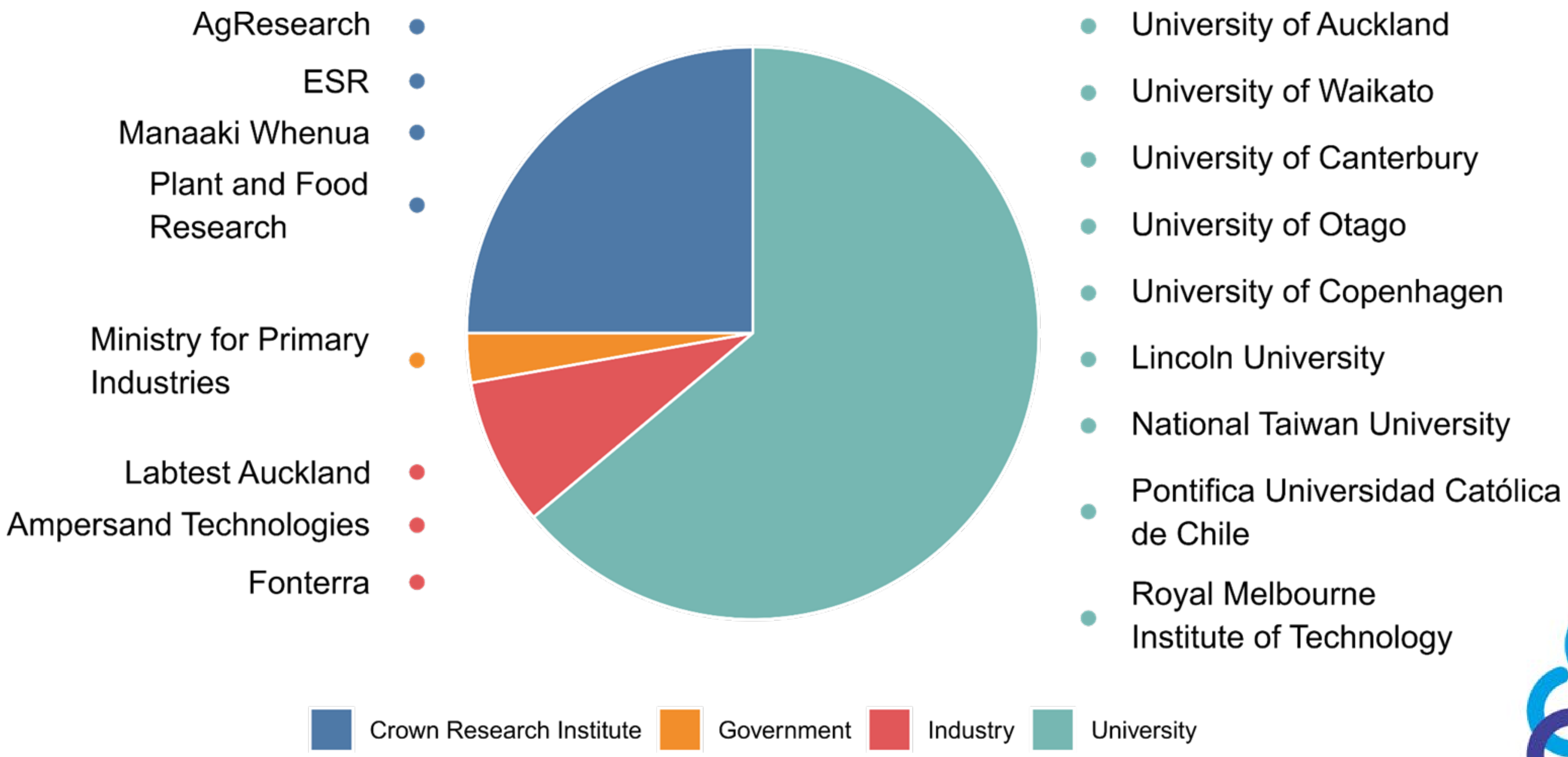
Shared working doc

https://tinyurl.com/mgss2023doc

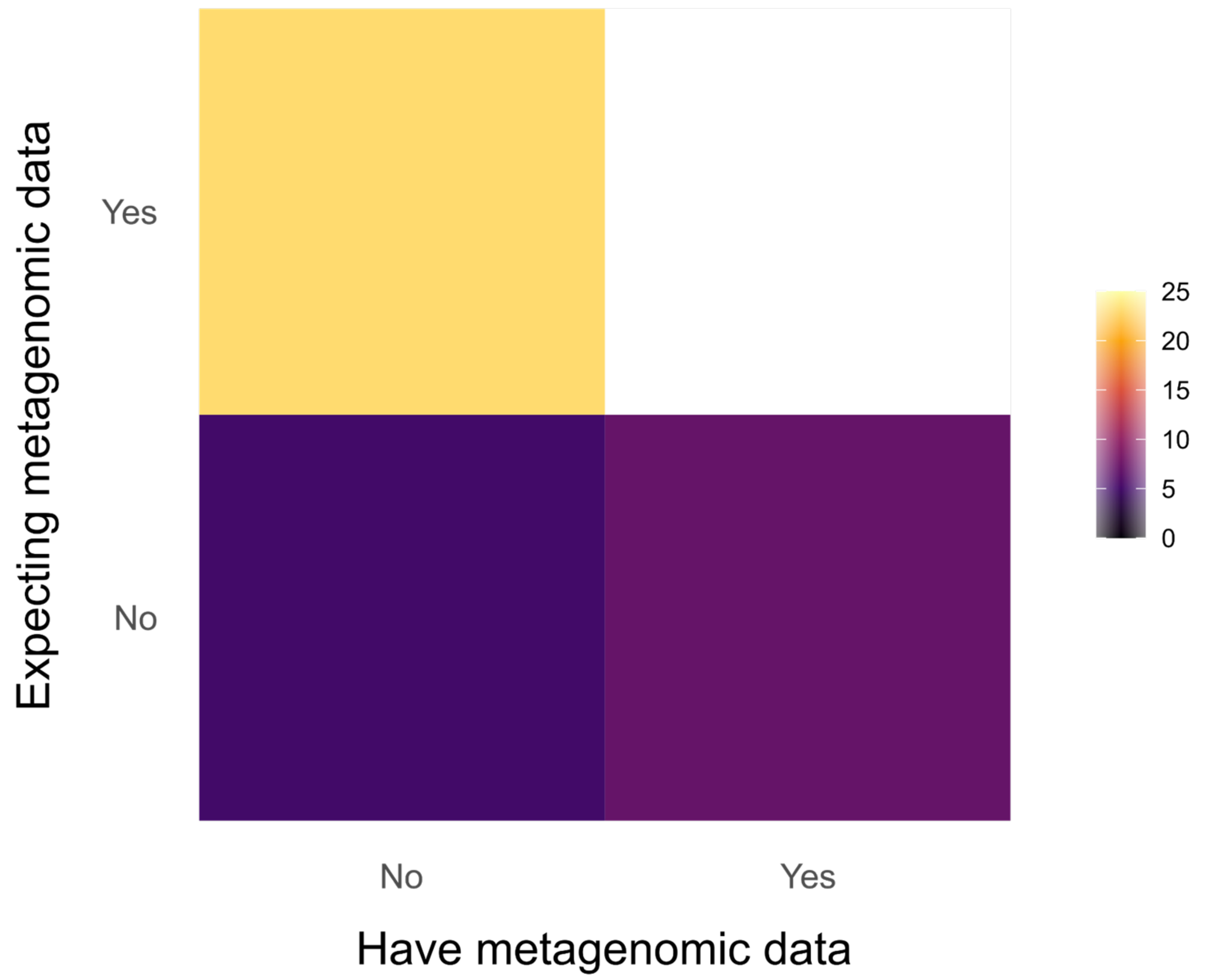
Welcome!

- **Housekeeping**
- **Etherpad for collaborative Q&A/comments**
    - https://tinyurl.com/mgss2023doc
- **Overview of attendees**
- **Any questions?**

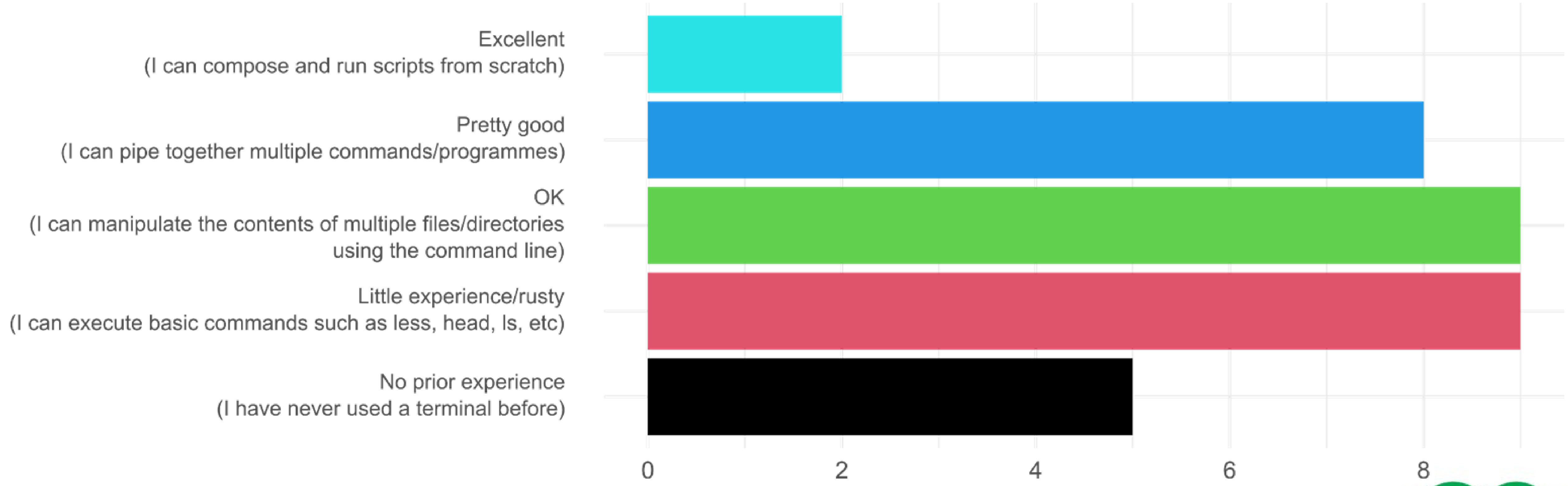
Where are we from?



Do we have data?



Command line experience?



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

Starting each session

1. Log in to the NeSI Jupyter hub via a browser

1. Open the workshop exercise materials on GitHub

1. Optional: Open a (plain text) text editor for taking notes

Bash scripting

Task: Bash scripting

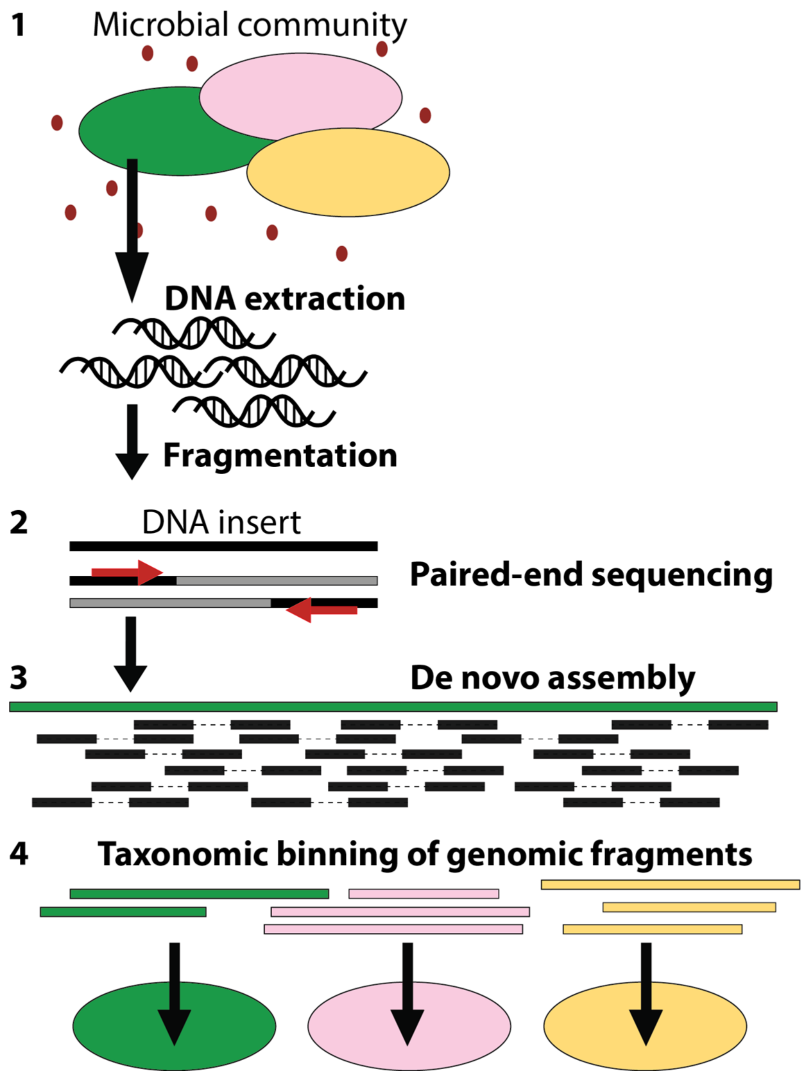
Go to Github MGSS webpage

Tasks:

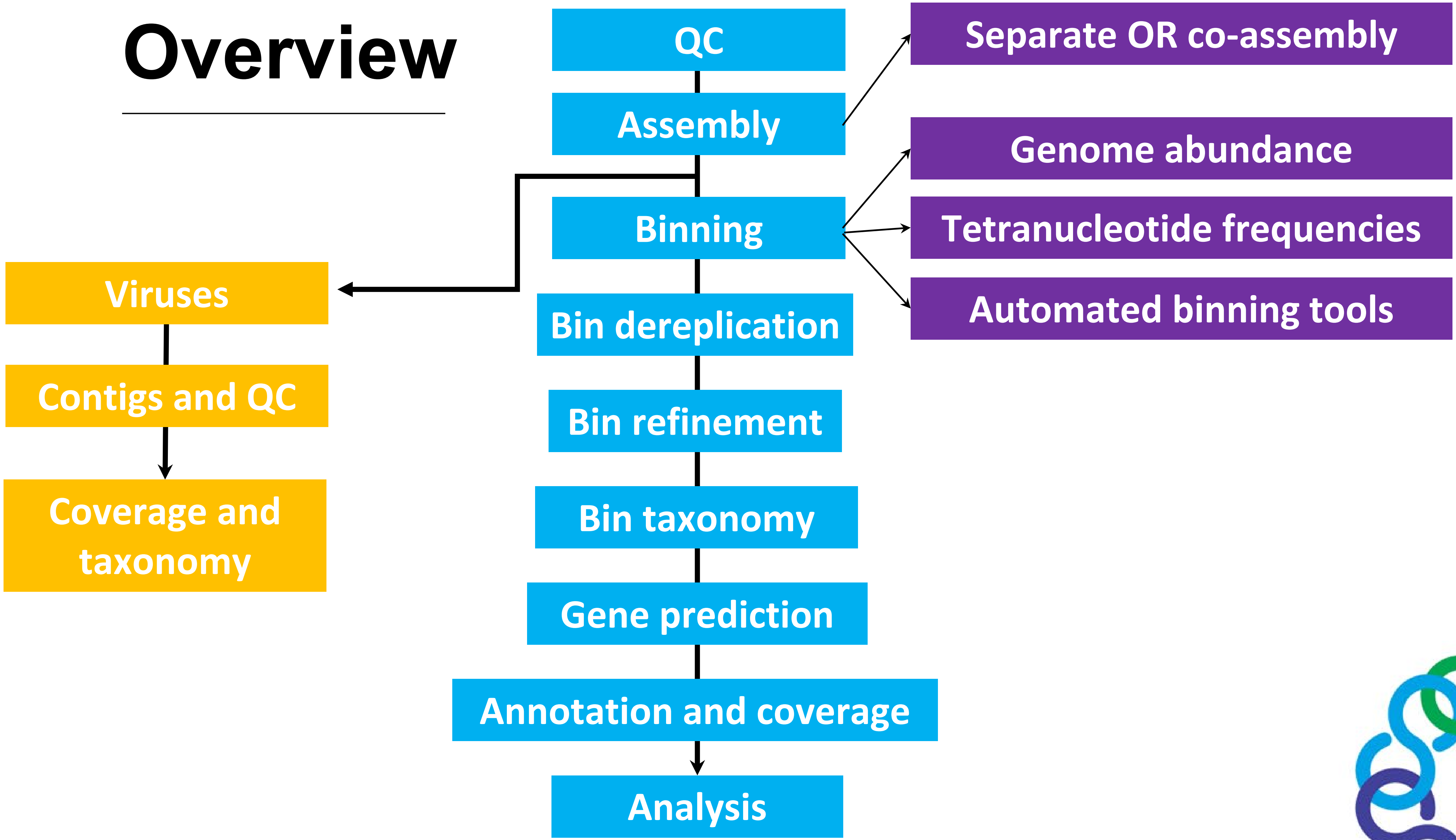
- **Introduction to shell**
- **Introduction to HPC & HPC job**

Metagenomic decision tree(s)

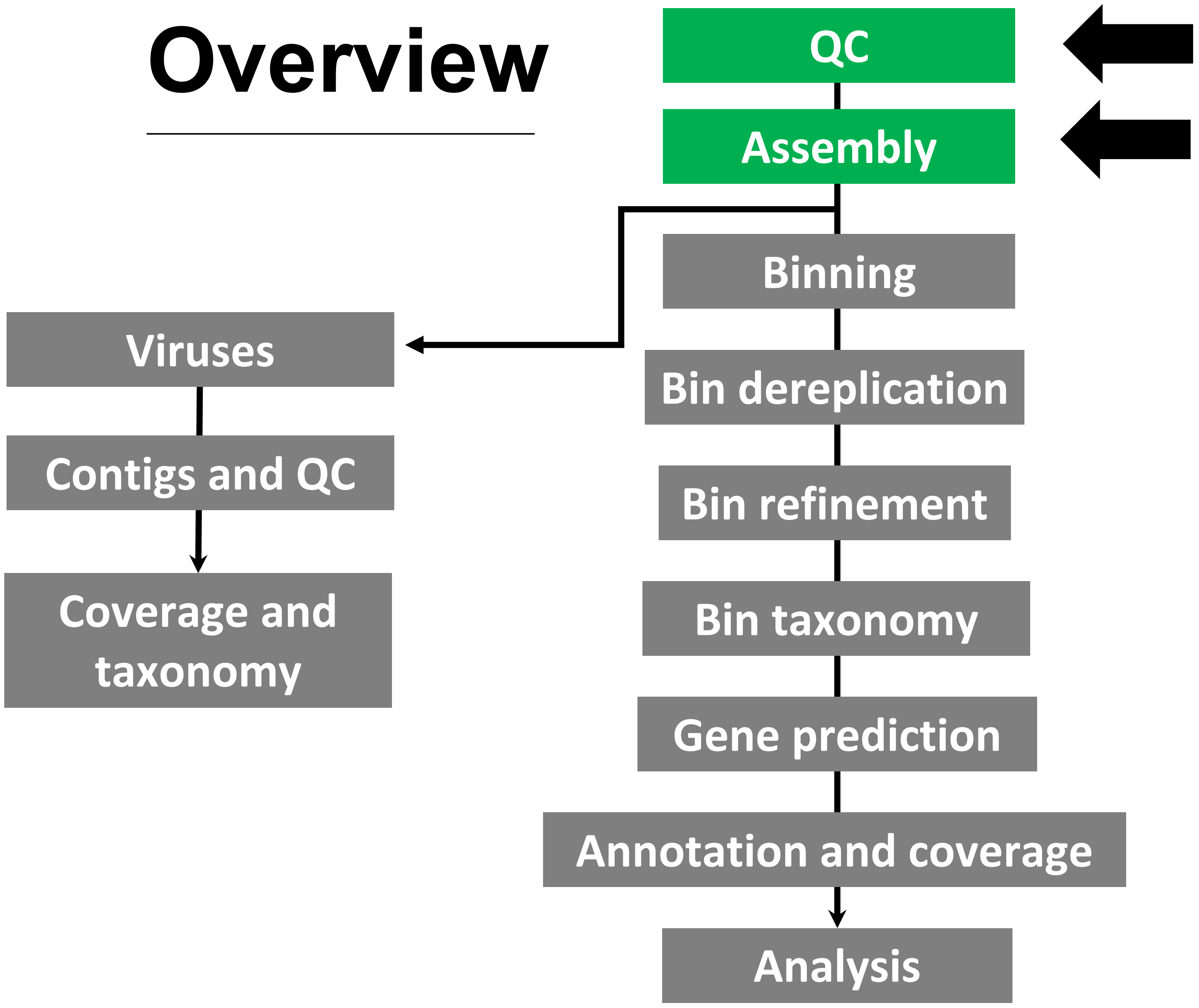
Our goal: genome recovery



Overview


QC
Assembly
Separate OR co-assembly
Genome abundance
Binning
Tetranucleotide frequencies
Automated binning tools
Viruses
Contigs and QC
Coverage and taxonomy
Bin dereplication
Bin refinement
Bin taxonomy
Gene prediction
Annotation and coverage
Analysis

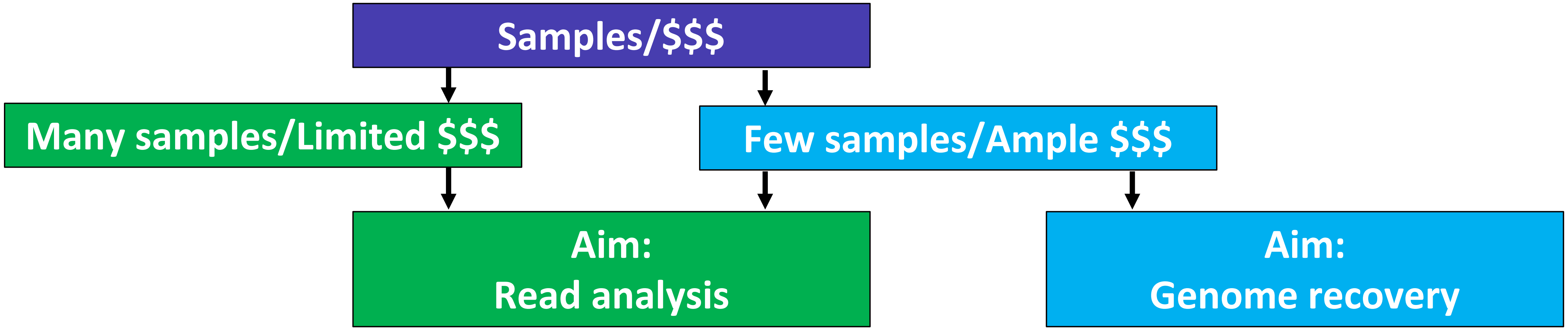
Overview

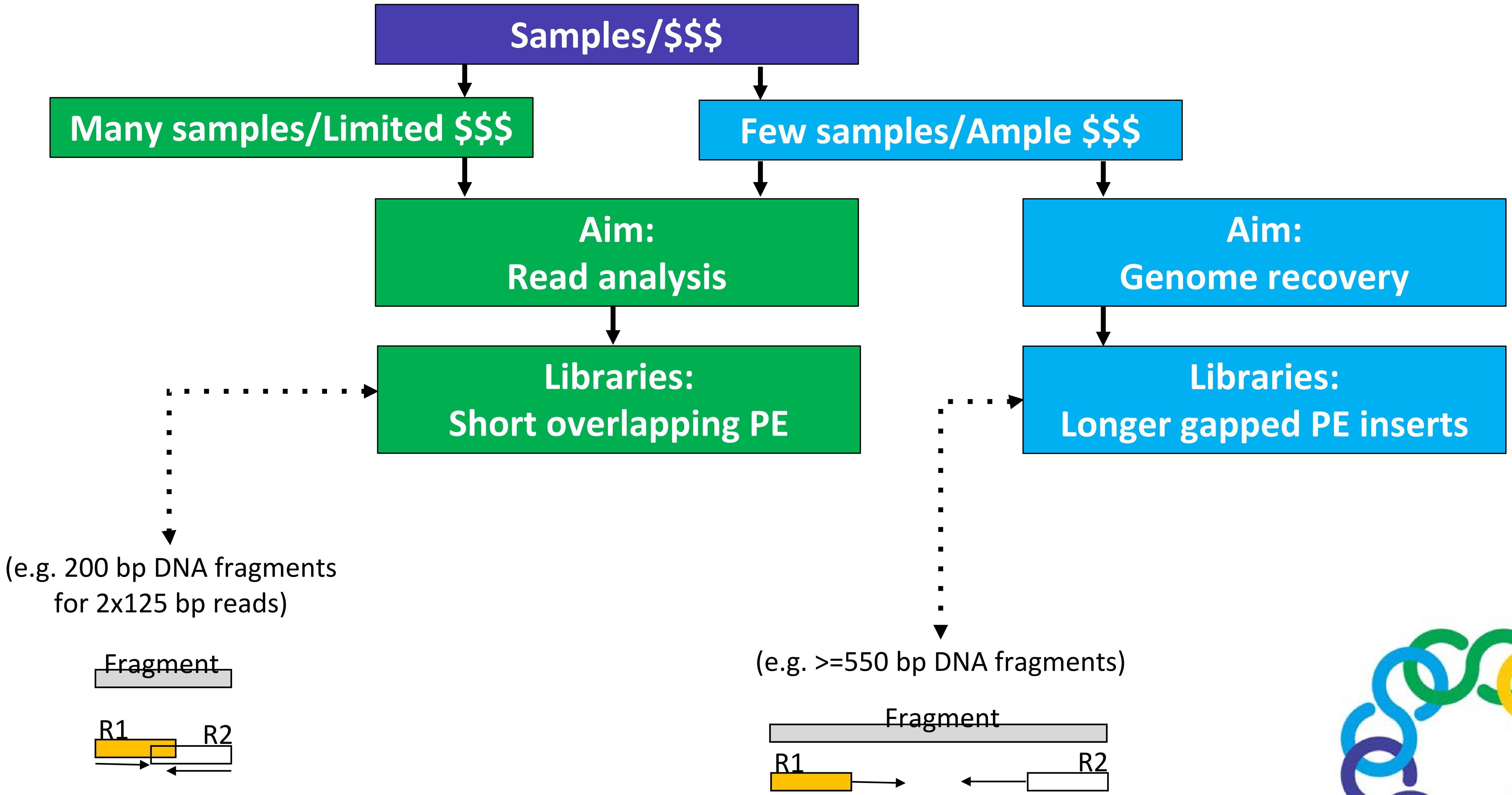

QC
Assembly
Binning
Bin dereplication
Bin refinement
Bin taxonomy
Gene prediction
Annotation and coverage
Analysis
Viruses
Contigs and QC
Coverage and taxonomy

Decision tree

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

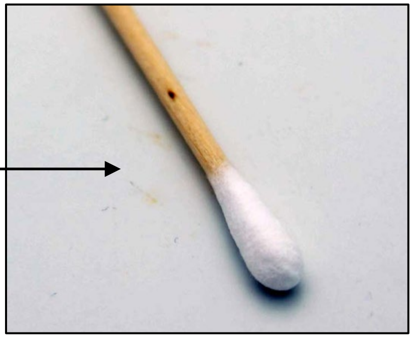
$$\begin{tikzcd}[column sep=small] & \text{Samples/\$\$\$} \arrow[dl] \arrow[dr] & \\ \text{Many samples/Limited \$\$\$} & & \text{Few samples/Ample \$\$\$} \end{tikzcd}$$


Samples/$$$
Many samples/Limited $$$
Few samples/Ample $$$
Aim:
Read analysis
Aim:
Genome recovery


Samples/$$$
Many samples/Limited $$$
Few samples/Ample $$$
Aim:
Read analysis
Aim:
Genome recovery
Libraries:
Short overlapping PE
Libraries:
Longer gapped PE inserts
(e.g. 200 bp DNA fragments
for 2x125 bp reads)
Fragment
R1
R2
(e.g. >=550 bp DNA fragments)
Fragment
R1
R2

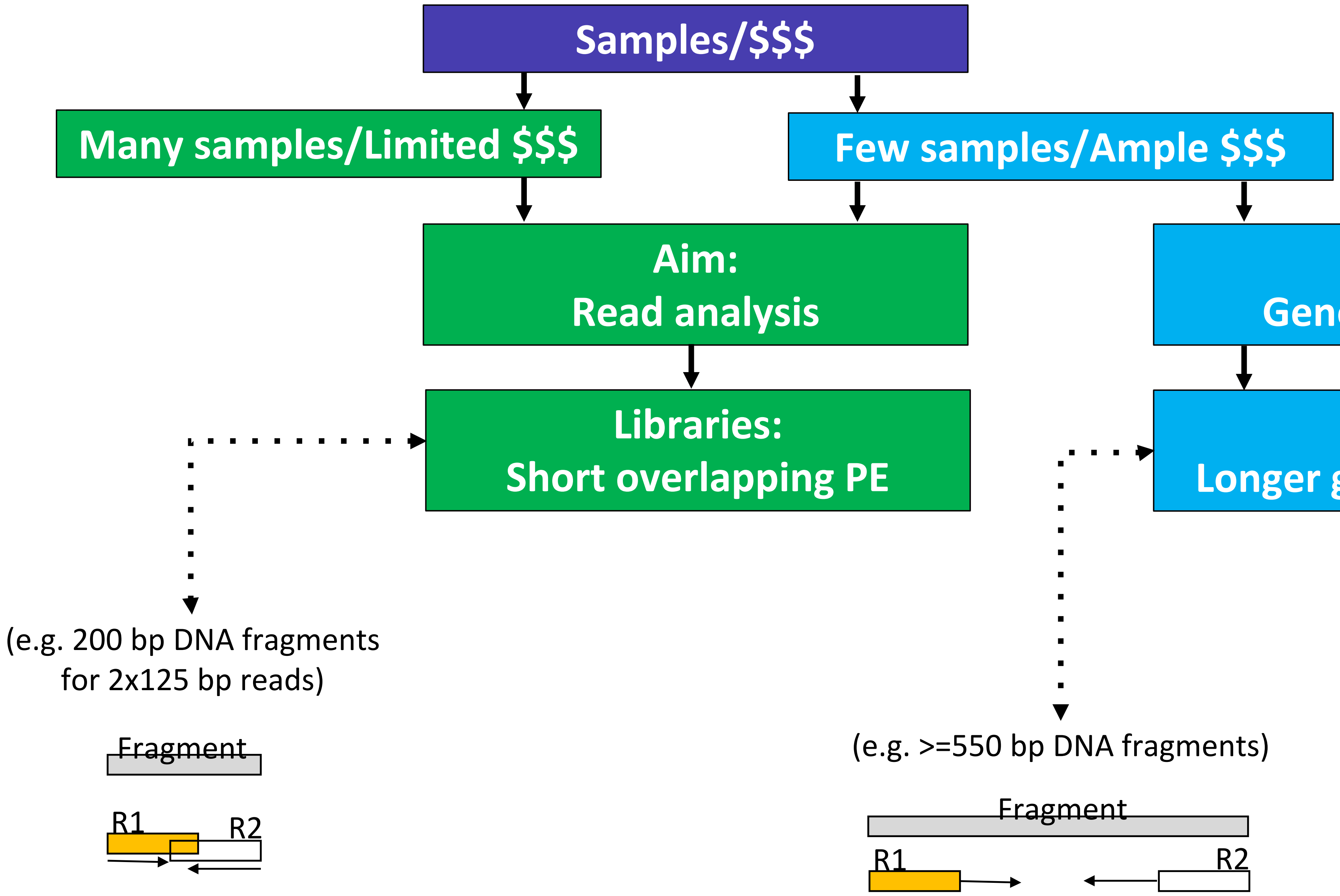
DNA input

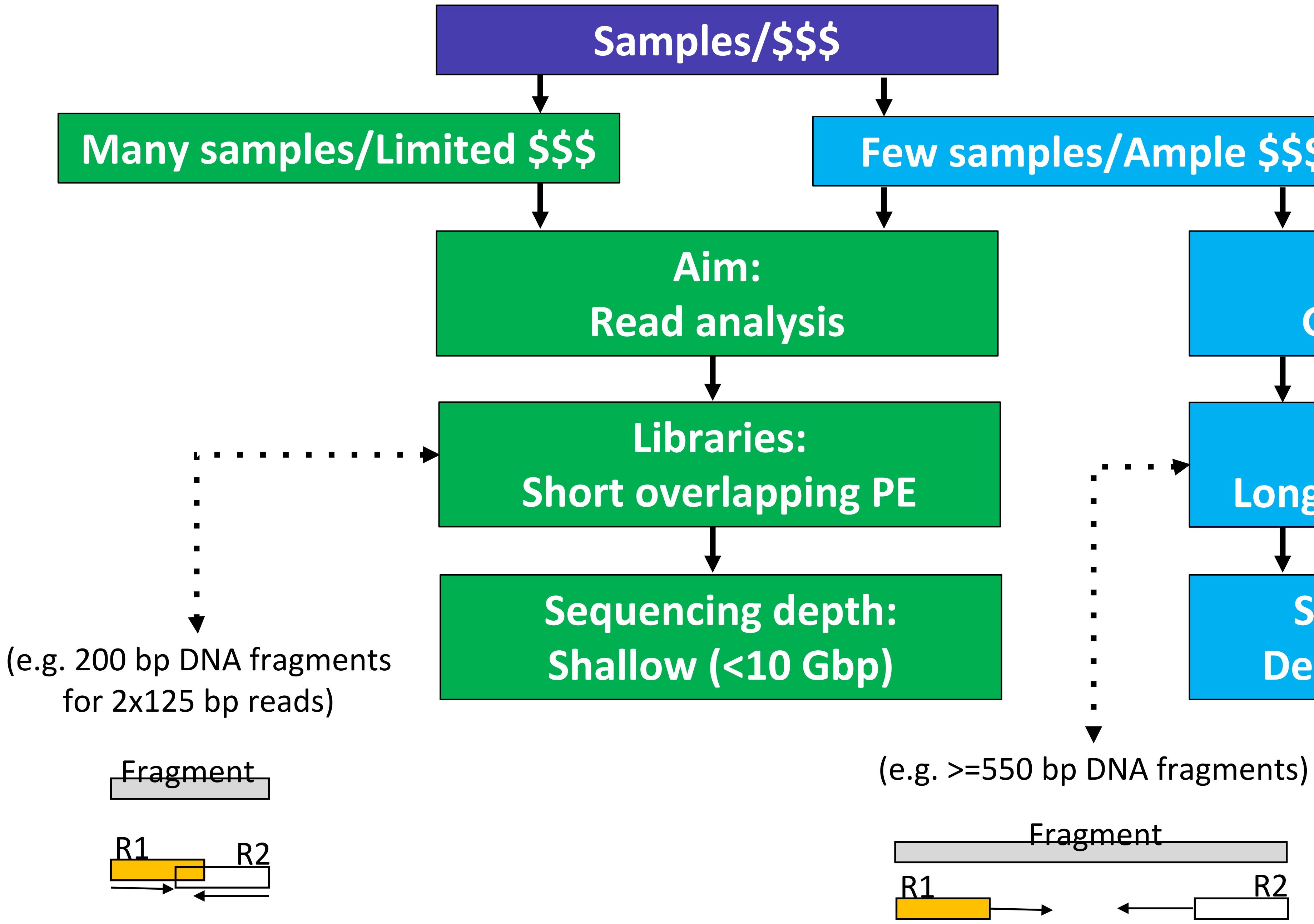
- **Very low inputs (e.g. nanograms) for Nextera library prep = enzymatic fragmentation with broad size distributions**



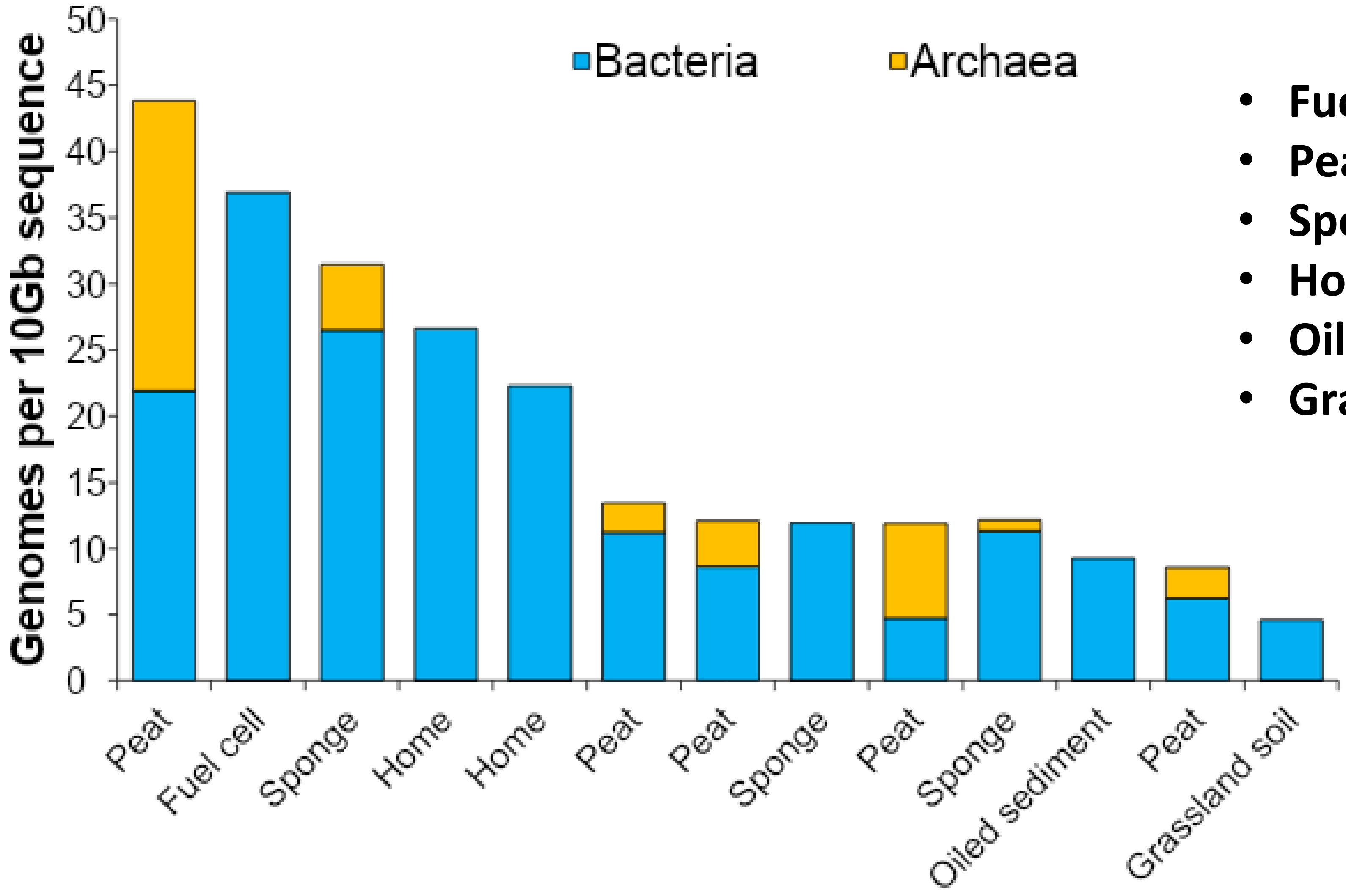
- **High inputs (e.g. 100s ng) for TruSeq = physical fragmentation with defined size selection**

Tends to yield sequences of larger inserts


Samples/$$$
Many samples/Limited $$$
Few samples/Ample $$$
Aim:
Read analysis
Aim:
Genome recovery
Libraries:
Short overlapping PE
Libraries:
Longer gapped PE inserts
(e.g. 200 bp DNA fragments
for 2x125 bp reads)
Fragment
R1
R2
(e.g. >=550 bp DNA fragments)
Fragment
R1
R2

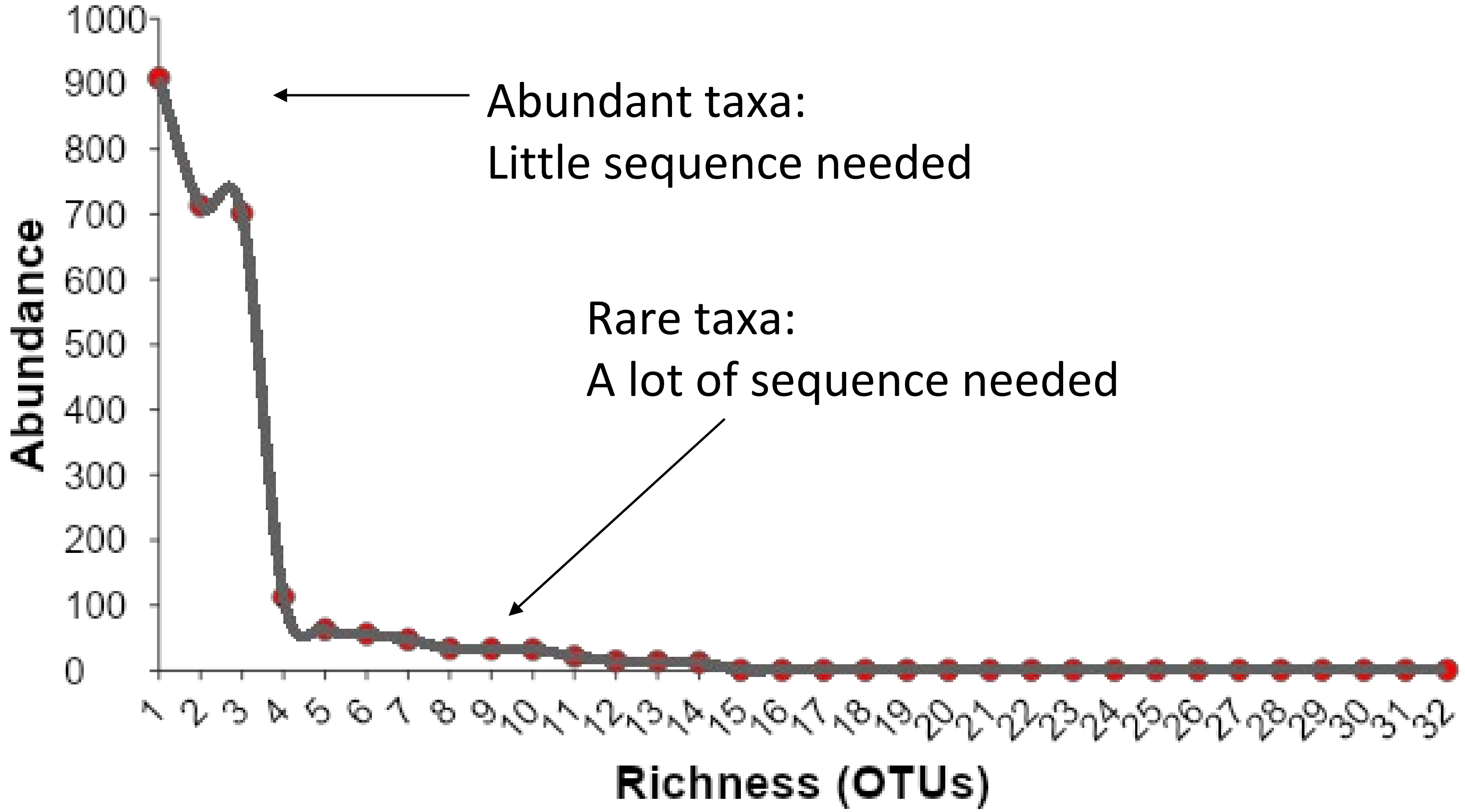

Samples/$$$
Many samples/Limited $$$
Few samples/Ample $$$
Aim:
Read analysis
Aim:
Genome recovery
Libraries:
Short overlapping PE
Libraries:
Longer gapped PE inserts
Sequencing depth:
Shallow (<10 Gbp)
Sequencing depth:
Deep (e.g. >=10s Gbp)
(e.g. 200 bp DNA fragments
for 2x125 bp reads)
Fragment
R1
R2
(e.g. >=550 bp DNA fragments)
Fragment
R1
R2

Genome recovery per environment



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

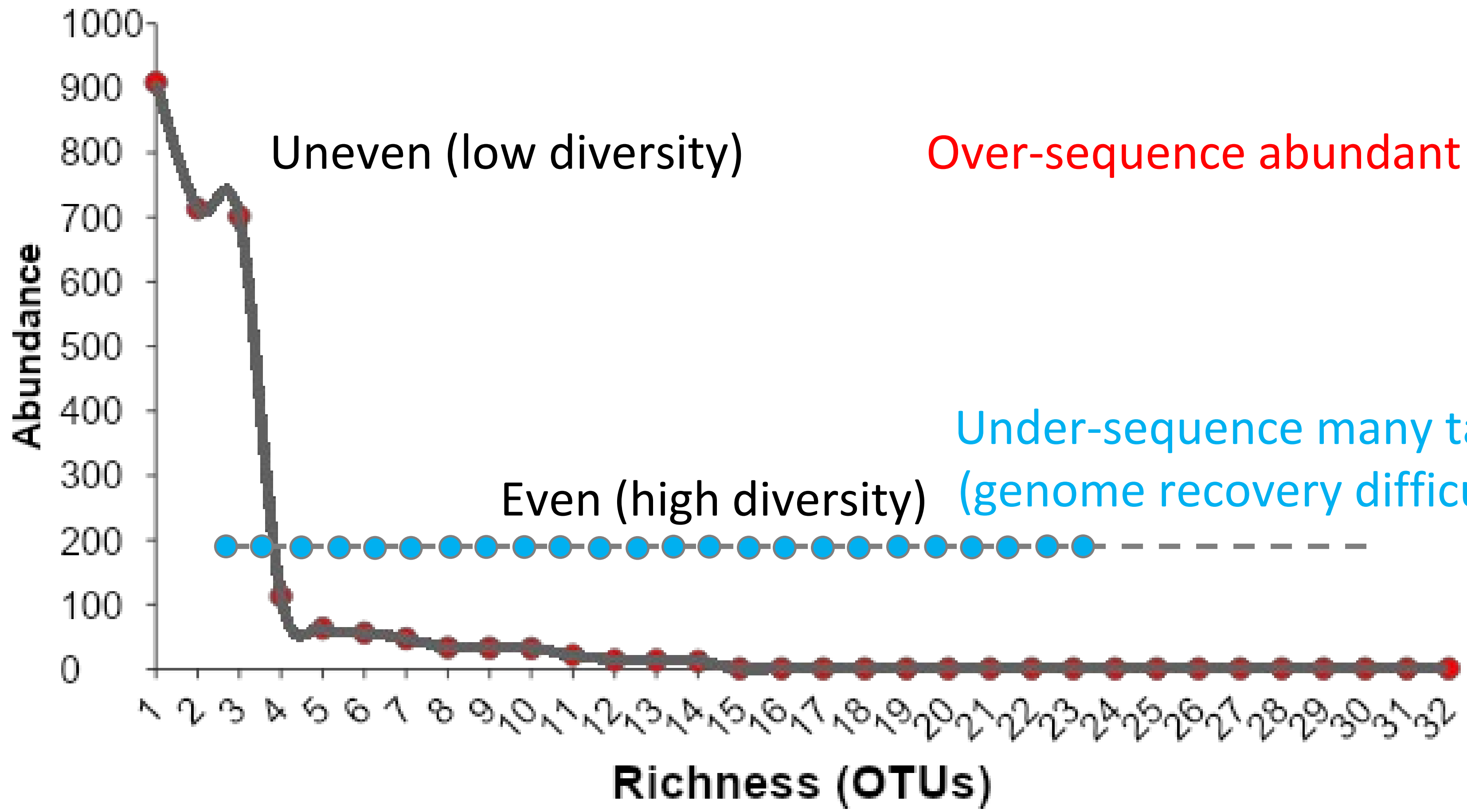
Estimate sequencing depth



Abundant taxa:
Little sequence needed

Rare taxa:
A lot of sequence needed

Community structure matters

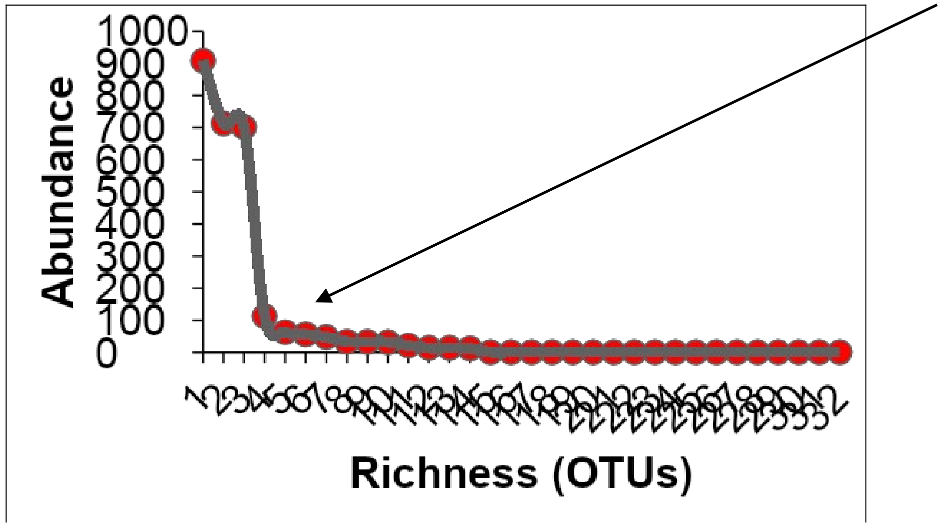


Over-sequence abundant taxa

Under-sequence many taxa (genome recovery difficult)

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

Prokaryotic genome sizes

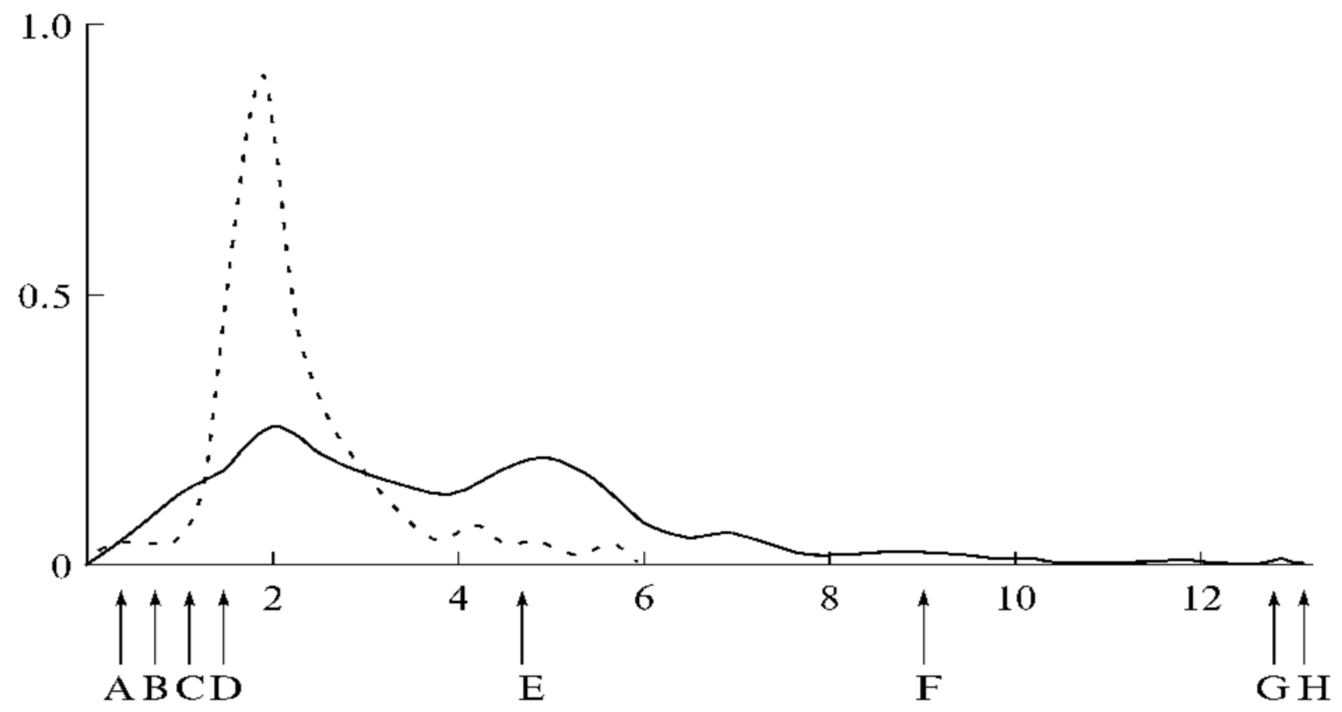
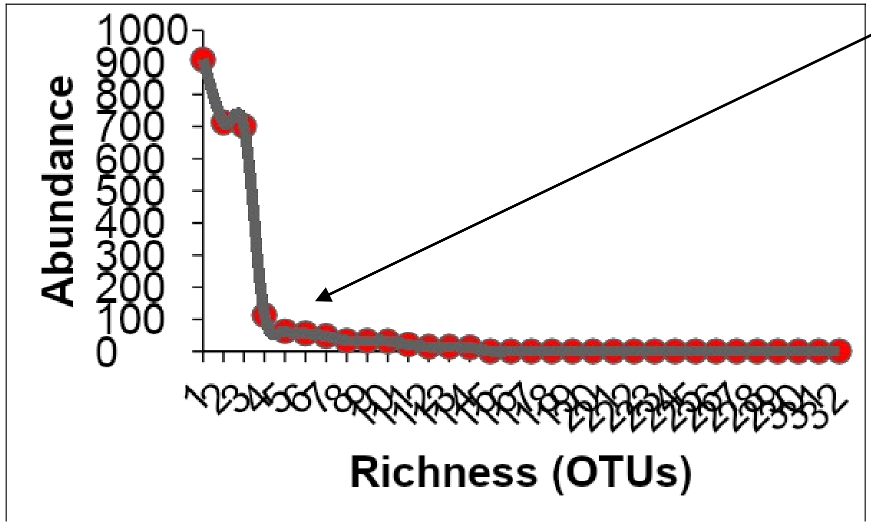


Fig. 1. Ranges of bacterial and archaeal genome sizes. Abscissa shows genome size, Mbp; ordinate shows number of genomes; solid line indicates bacterial genomes; dashed line indicates archaeal genomes; A, *C. ruddii* genome; B, *N. equitans* genome; C, minimal size for free-living microorganisms; D, major peak for genome sizes of bacterial and archaeal genomes; E, minor peak for bacterial genomes; F, *Nostoc punctiforme* genome; G, *Sorangium cellulosum* genome; and H, Van Nimwegen limit.

(Smirnov 2010 Molecular Genetics, Microbiology and Virology)

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)**
- **30x coverage**

5 Mbp x 20 x 30 = 3,000 Mbp (or 3 Gbp)

When you have so many genomes

You need a:

- Clear goal
- Question
- Hypothesis to test

Q&A

Approaches to metagenomics analyses, e.g.

- **Short read vs long read sequencing**
- **Assembled genomes vs unbinned reads/contigs**

Q&A

Approaches to metagenomics analyses, e.g.

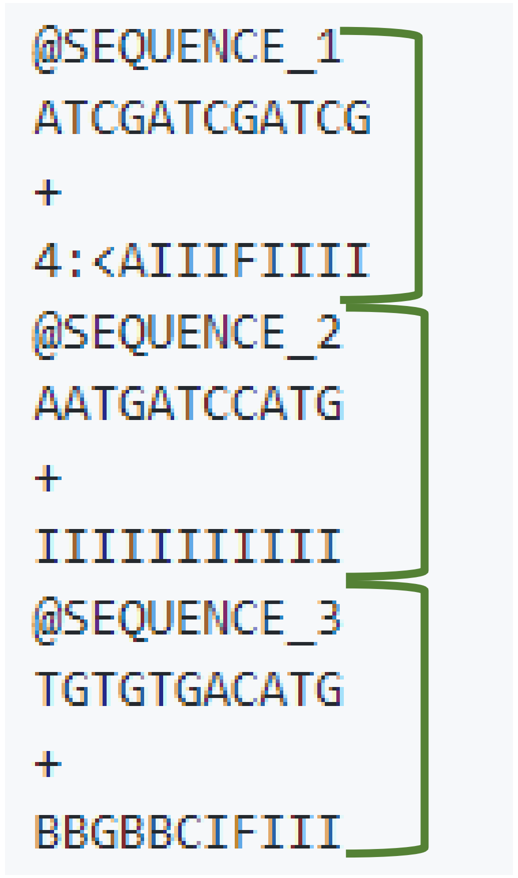
- **Short read vs long read sequencing**
- **Assembled genomes vs unbinned reads/contigs**

Mini-project

- Denitrification (Nitrate or nitrite to nitrogen)
- Ammonia oxidation (Ammonia to nitrite or nitrate)
- Anammox (Ammonia and nitrite to nitrogen)
- Sulfur oxidation (SOX pathway, thiosulfate to sulfate)
- Sulfur reduction (DSR pathway, sulfate to sulfide)
- Photosynthetic carbon fixation
- Non-photosynthetic carbon fixation (Reverse TCA or Wood-Ljungdahl)
- Non-polar flagella expression due to a chromosomal deletion
- Plasmid-encoded antibiotic resistance
- Aerobic (versus anaerobic) metabolism

Quality control/filtering raw reads

The FastQ data format



```
@SEQUENCE_1
ATCGATCGATCG
+
4:<AIIIFIIII
@SEQUENCE_2
AATGATCCATG
+
IIIIIIIIIII
@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

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

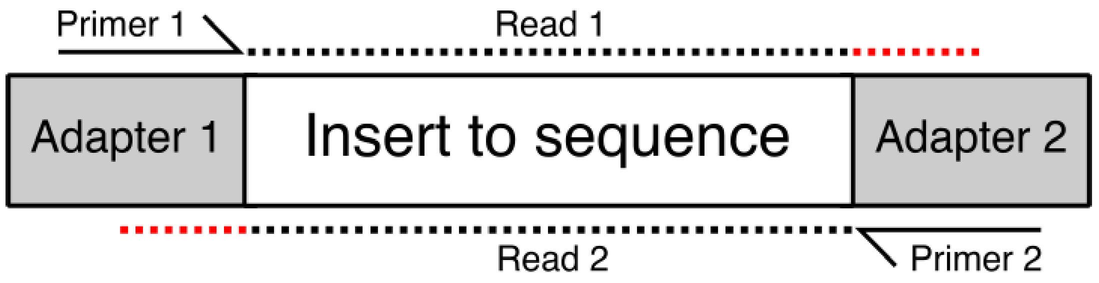
```
(33):  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
```

Quality filtering WGS data

- Remove barcode and adapter regions
- Remove low-quality regions of reads
- Identify potential problems that occurred during sequencing
    - Deciphering ‘aberrant’ metrics in FastQC
        - e.g. Adapter read-through
        - e.g. Rapid drop off in sequence quality

Quality filtering WGS data

- Remove barcode and adapter regions
- Remove low-quality regions of reads
- Identify potential problems that occurred during sequencing
  - Deciphering ‘aberrant’ metrics in FastQC
    - e.g. Adapter read-through
    - e.g. Rapid drop off in sequence quality



Task: Quality filtering

Go to Github MGSS webpage

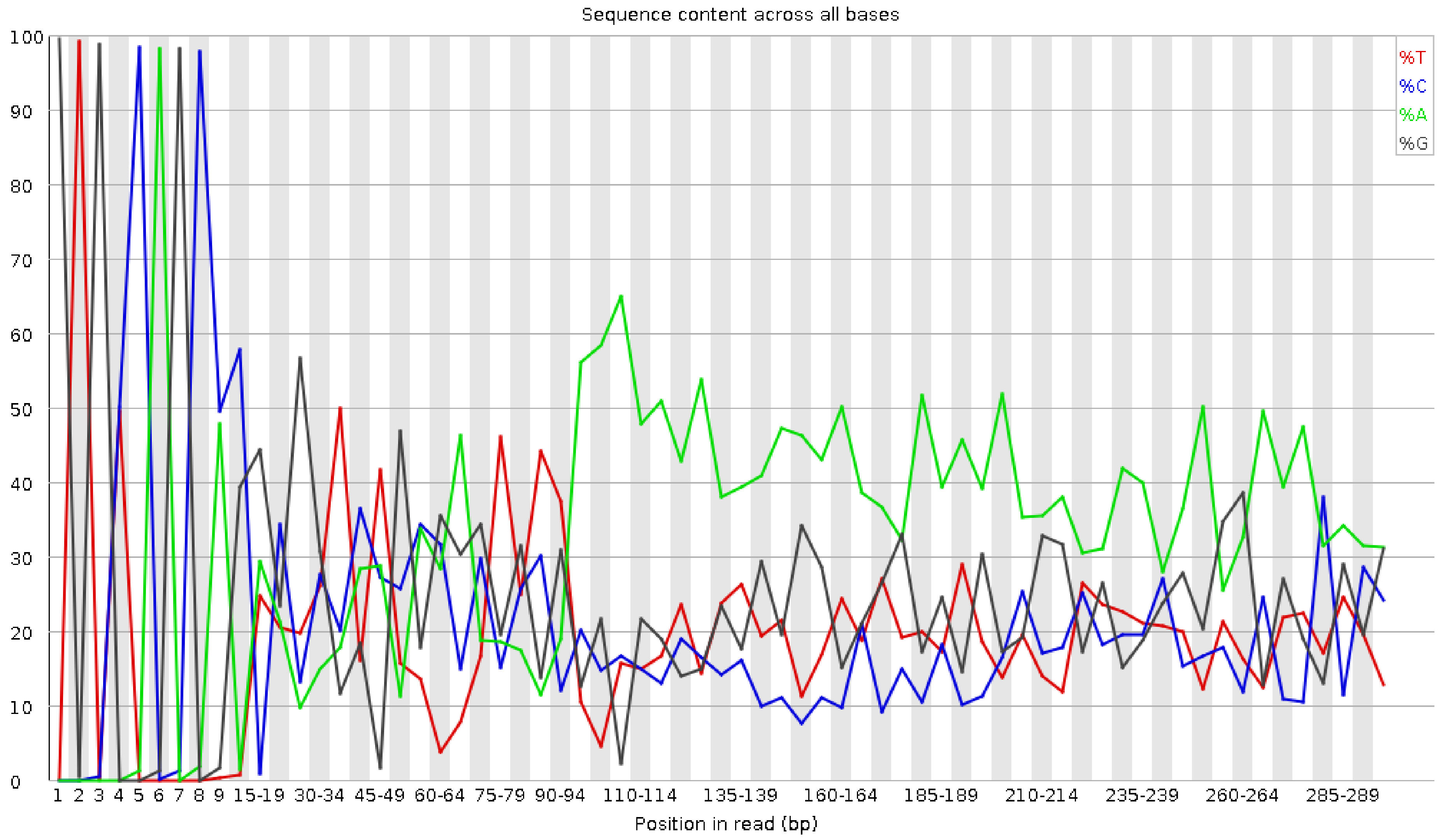
Tasks:

- **Visualisation with *FastQC***
  - Inspecting FASTQ files
  - Identifying regions of concern
- **Read trimming and adapter removal with Trimmomatic**
  - Removing adapter sequences
  - Removing low-quality regions
- **Diagnosing poor libraries**
- **(Optional) Filtering out host DNA**

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.

Common issues with WGS data



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.

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/

Filtering out host DNA

Metagenome data derived from microbial communities associated with a host should ideally be filtered to remove any reads originating from host DNA. This may improve the quality and efficiency of downstream data processing

Important for submission to databases e.g. NCBI

- Ethics for human host DNA
- Taonga species in Aotearoa

Task: Quality filtering

Go to Github MGSS webpage

Tasks:

✓ • **Visualisation with *FastQC***
  - Inspecting FASTQ files
  - Identifying regions of concern

✓ • **Read trimming and adapter removal with Trimmomatic**
  - Removing adapter sequences
  - Removing low-quality regions

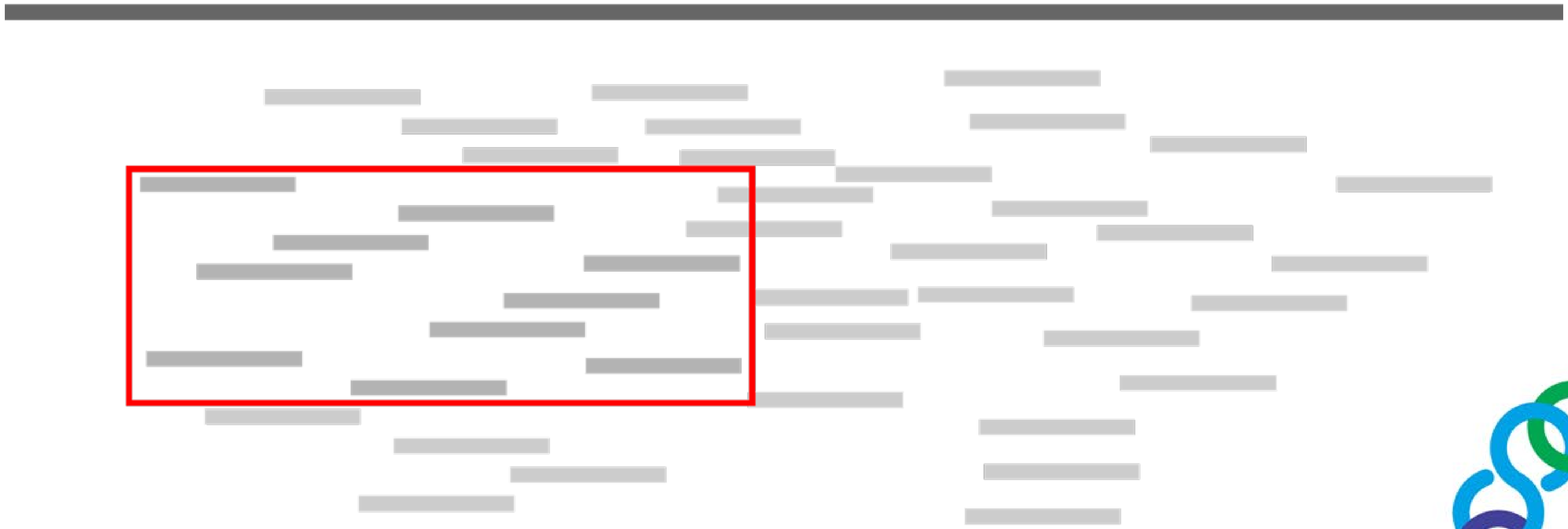
✓ • **Diagnosing poor libraries**

• **(Optional) Filtering out host DNA**

Assembly

Genome assembly

Overlap-Consensus-Layout (OCL) assembly



Genome assembly

Overlap-Consensus-Layout (OCL) assembly

TTGAAGAGTT

GGCTCAGATT

TTTGATCATG

AACGCTGGCG

AAGAGTTTGA

GATTGAACGC

CTCAGATTGA

TGAAGAGTTT

ACGCTGGCGC

TCATGGCTCA

Genome assembly

Overlap-Consensus-Layout (OCL) assembly

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

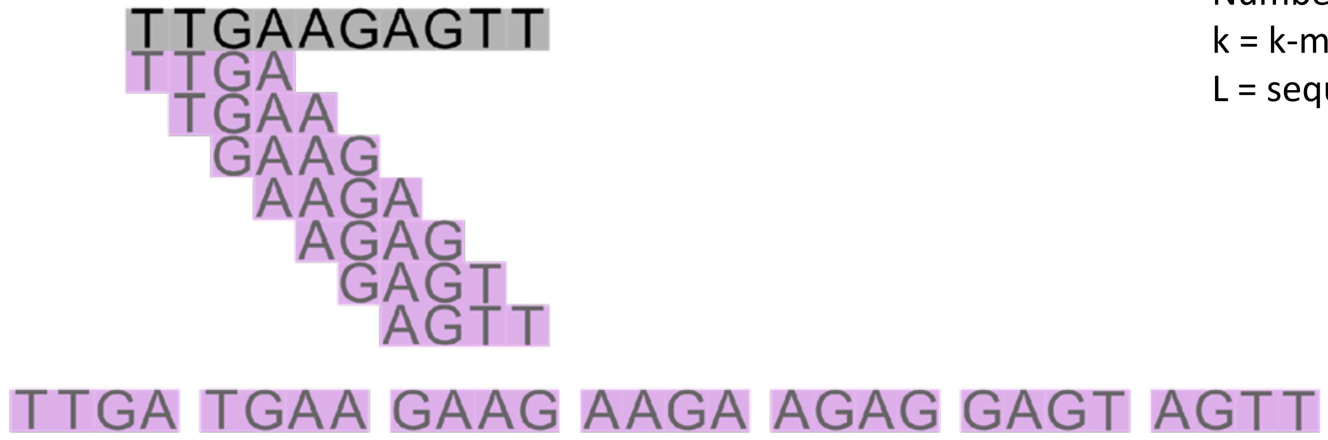
The problem for *de novo* assembly?

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

Genome assembly

De Bruijn graph assembly

Break reads into shorter *k*-mers

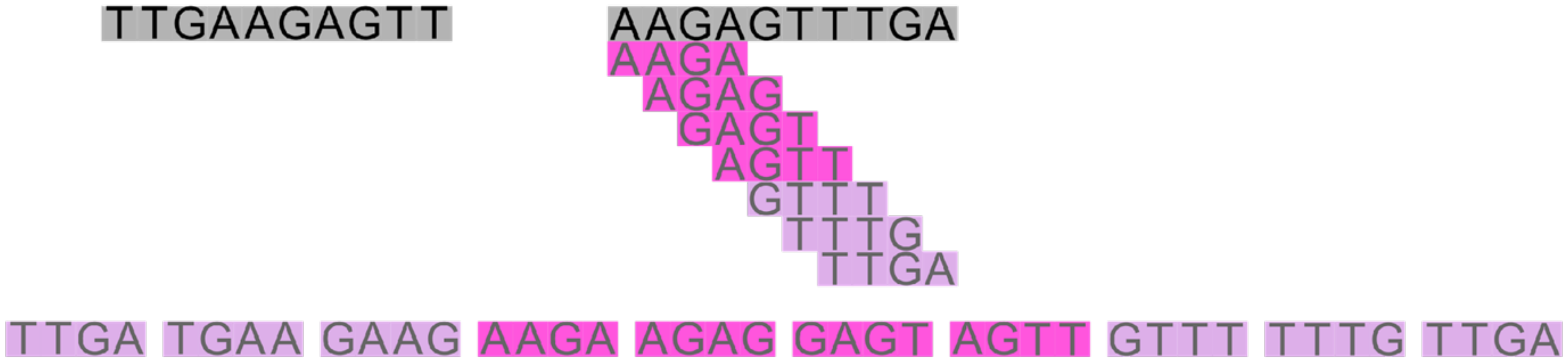


Number kmers per sequence = (L – k) + 1
k = k-mer length
L = sequence length

Genome assembly

De Bruijn graph assembly

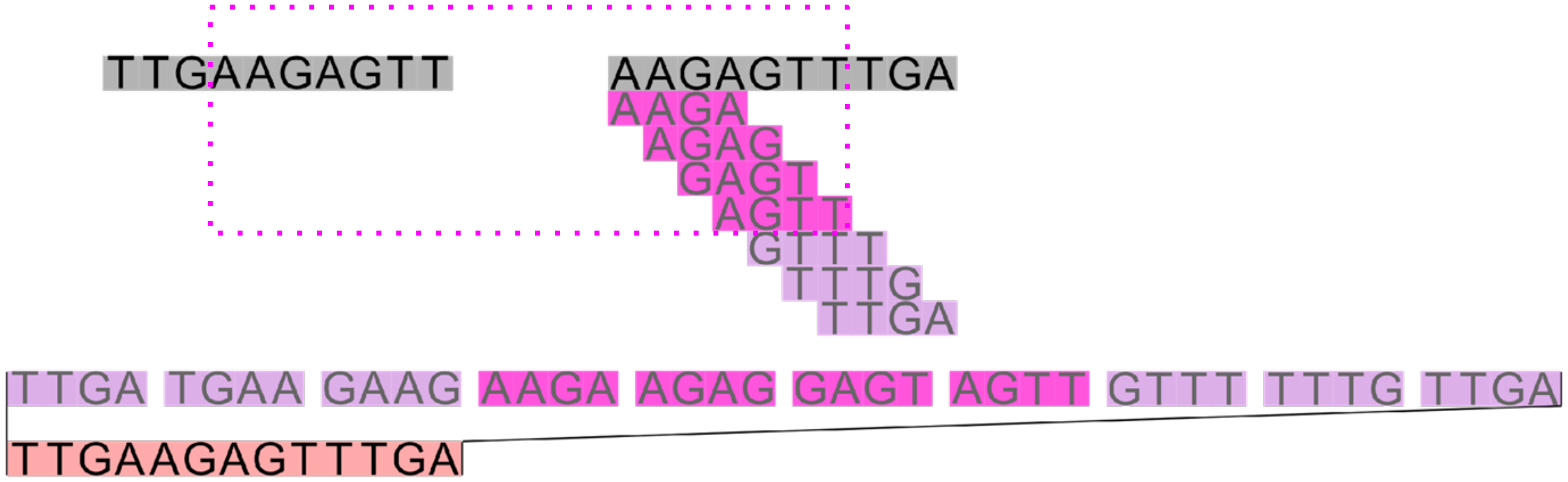
Identify sequences of shared *k*-mers



Genome assembly

De Bruijn graph assembly

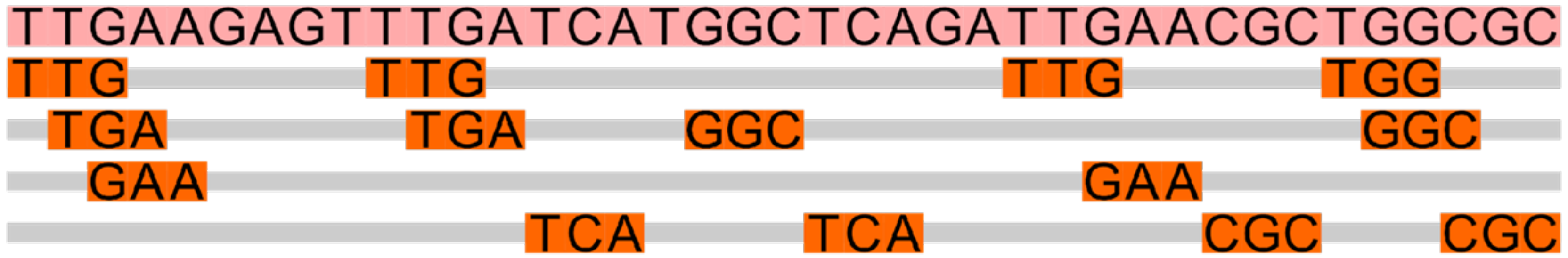
Identify sequences of shared *k*-mers



Genome assembly

De Bruijn graph assembly

Problem #1 – *k*-mers are short?

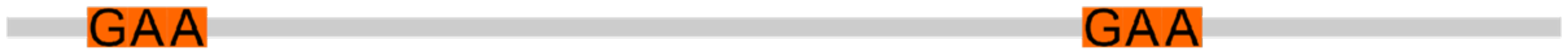


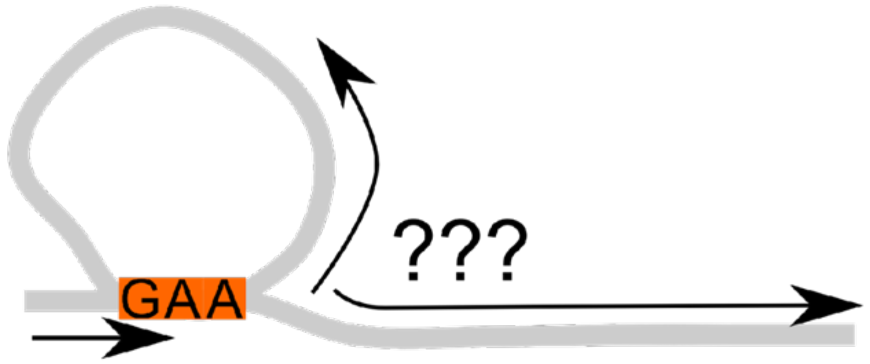
Genome assembly

De Bruijn graph assembly

Problem #1 – *k*-mers are short?

TTGAAGAGTTTGATCATGGCTCAGATTGAACGCTGGCGC





Genome assembly

De Bruijn graph assembly

Problem #2 – *k*-mers are long?

```
TTGAAGAGTT        AAGAGTTTGA
TTGAAGAG          AAGAGTTT
 TGAAGAGT          AGAGTTTG
  GAAGAGTT          GAGTTTGA
```

```
TTGAAGAG  TGAAGAGT  GAAGAGTT

    AAGAGTTT  AGAGTTTG  GAGTTTGA
```

De Bruijn graph assembly

We want a range of *k*-mer sizes

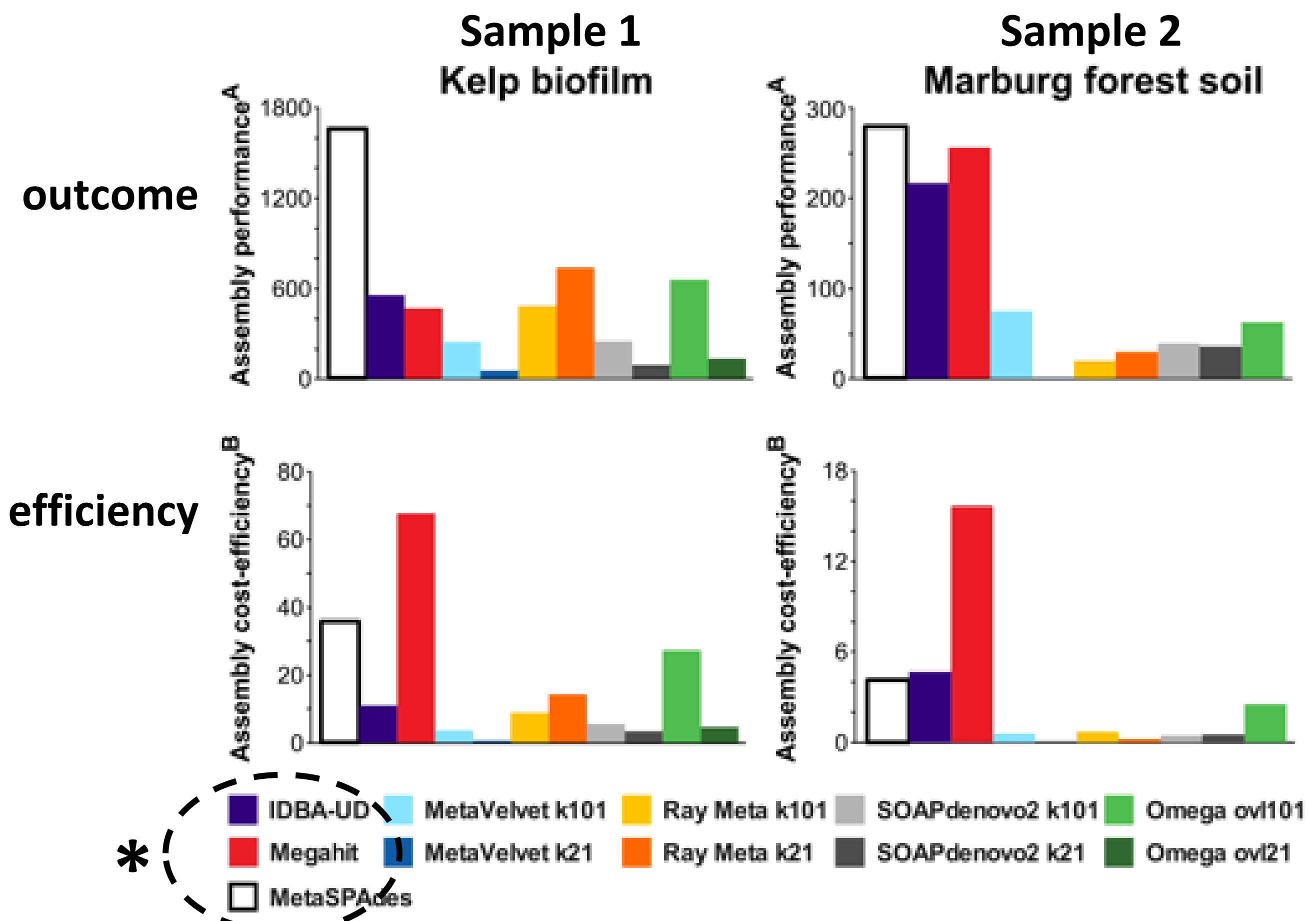
- Short *k*-mers yield higher coverage
- Long *k*-mers assemble longer contigs (jump repeat regions)

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?



[A] Assembly performance = Read mapping rate [%] x N50 [kb]

[B] Cost-efficiency = Assembly performance / (RAM usage [Gb] + runtime [h] per cpu)

(Vollmers et al., 2017, PLOS ONE)

Outcomes vary by dataset.

Assembly optimization generally requires empirically testing:

- Assemblers
- Parameters

Which assembler is best?

There are three good options

- SPAdes
- MegaHIT
- IDBA-UD

Which assembler is best?

There are three good options

- SPAdes
- MegaHIT
- IDBA-UD

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 mico 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)

Which assembler is best?

There are three good options

- SPAdes
- MegaHIT
- IDBA-UD

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 mico 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)

What are some key considerations?

Biological

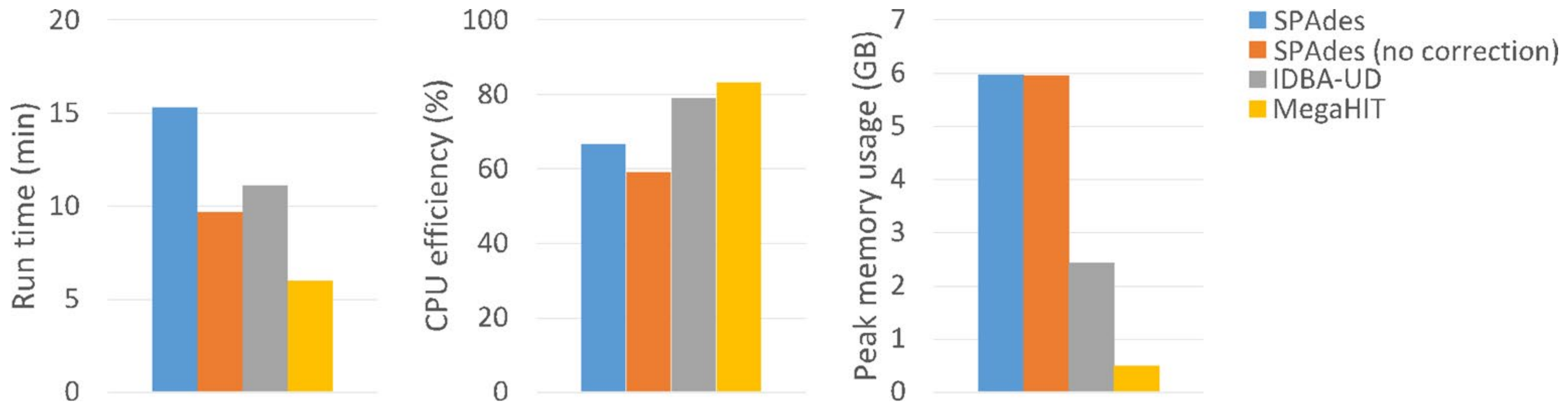
1. What is your hypothesis?
2. What do you want from the data?

Computational and resource

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

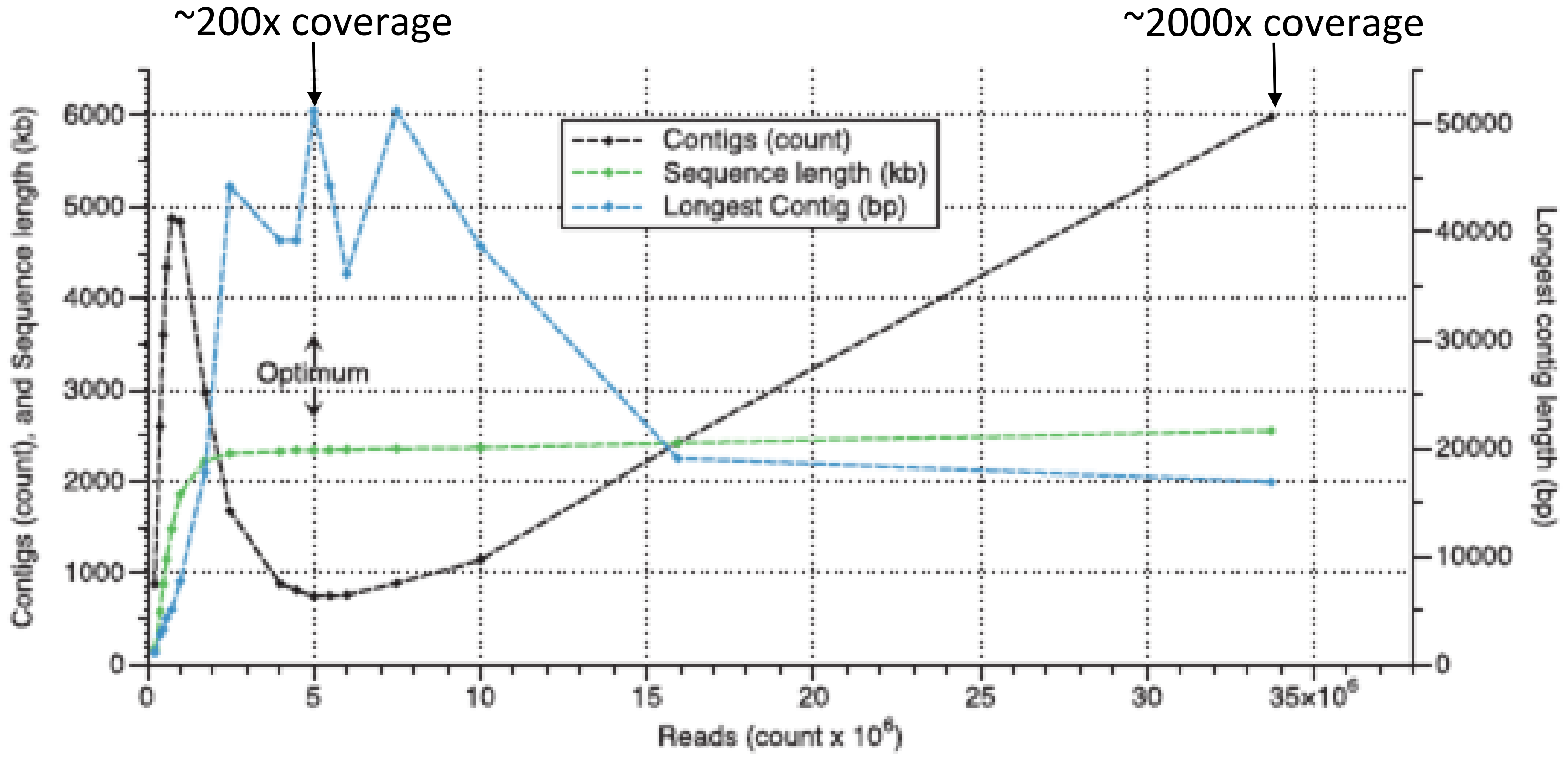
Genome assembly

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



(Fig. S1, Handley et al., 2014, Environ. Microbiol.)

Task: Assembly

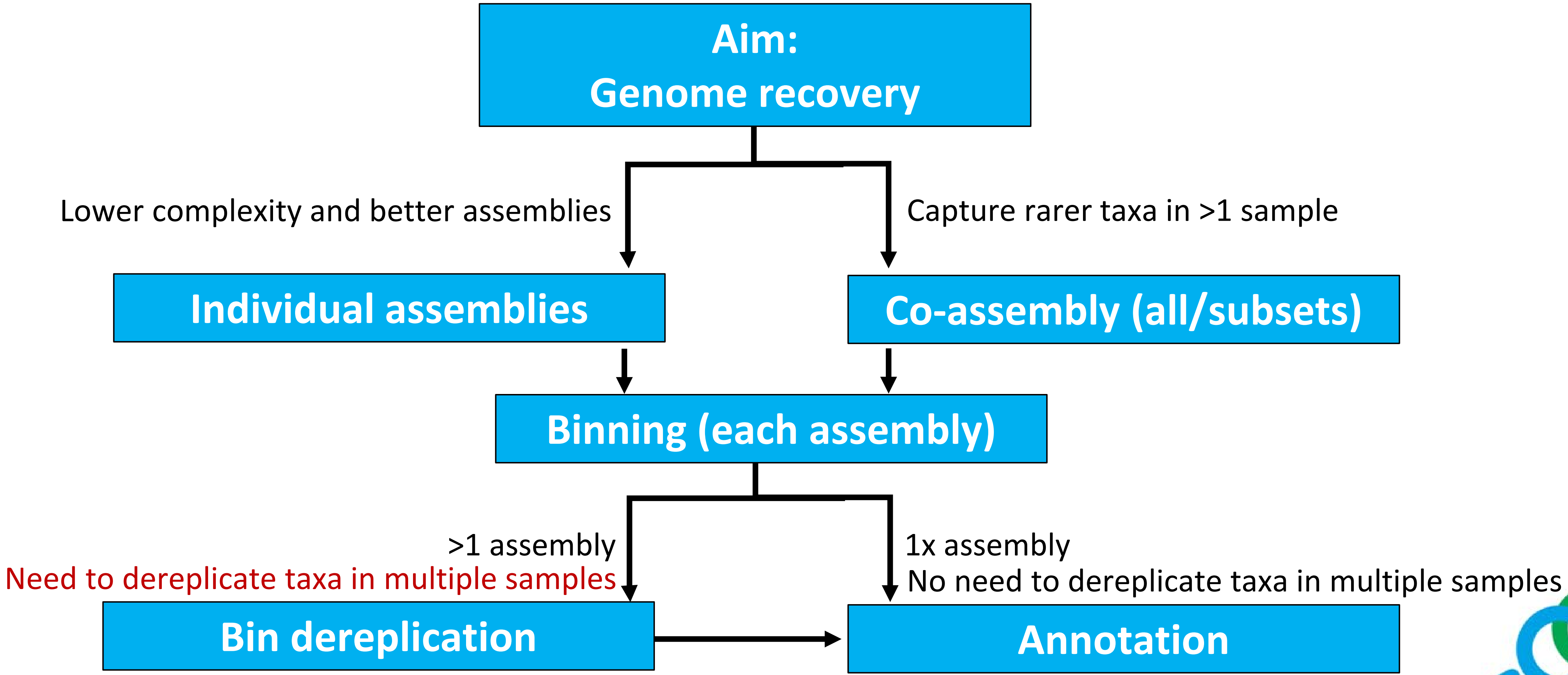
Go to Github MGSS webpage

Tasks:

- **Preparing data for assembly (Run IDBA_UD assembly)**
- **Exploring assembler options**
  - Configure the basic parameters for assembly
- **Submitting jobs to NeSI via slurm**
  - Prepare an assembly job to run under slurm
- **Run SPAdes and IDBA_UD assembly**
- **(Optional) Submitting variant assemblies to NeSI**

Future considerations and Assembly evaluation

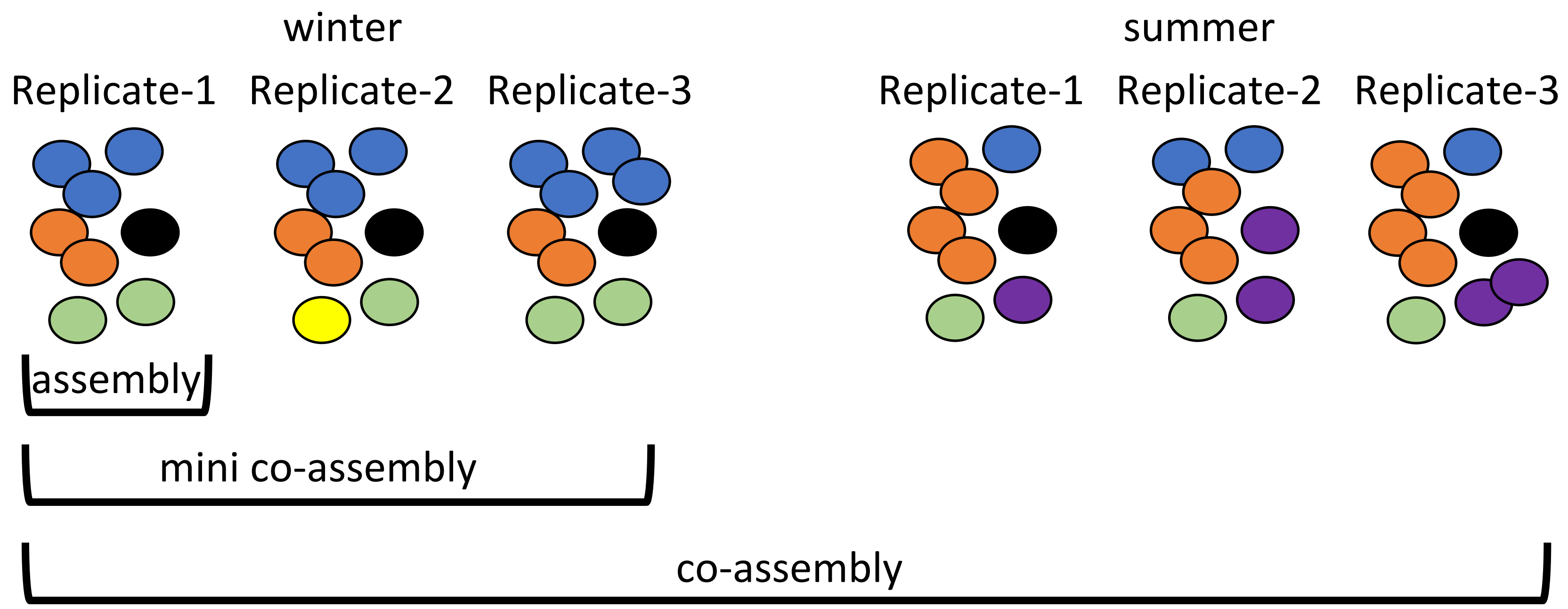
Future considerations


Aim:
Genome recovery
Lower complexity and better assemblies
Capture rarer taxa in >1 sample
Individual assemblies
Co-assembly (all/subsets)
Binning (each assembly)
>1 assembly
Need to dereplicate taxa in multiple samples
1x assembly
No need to dereplicate taxa in multiple samples
Bin dereplication
Annotation

Future considerations

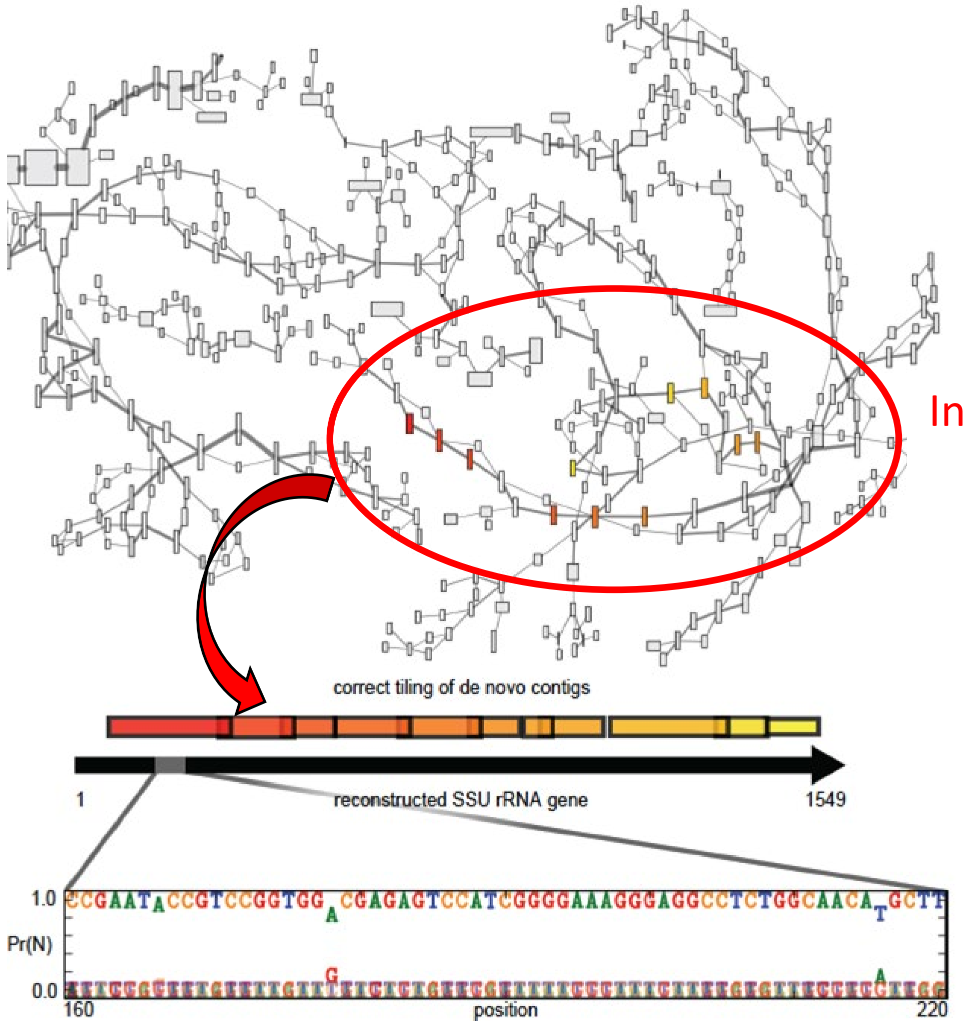
Assembly options:

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



Future considerations: rRNA genes

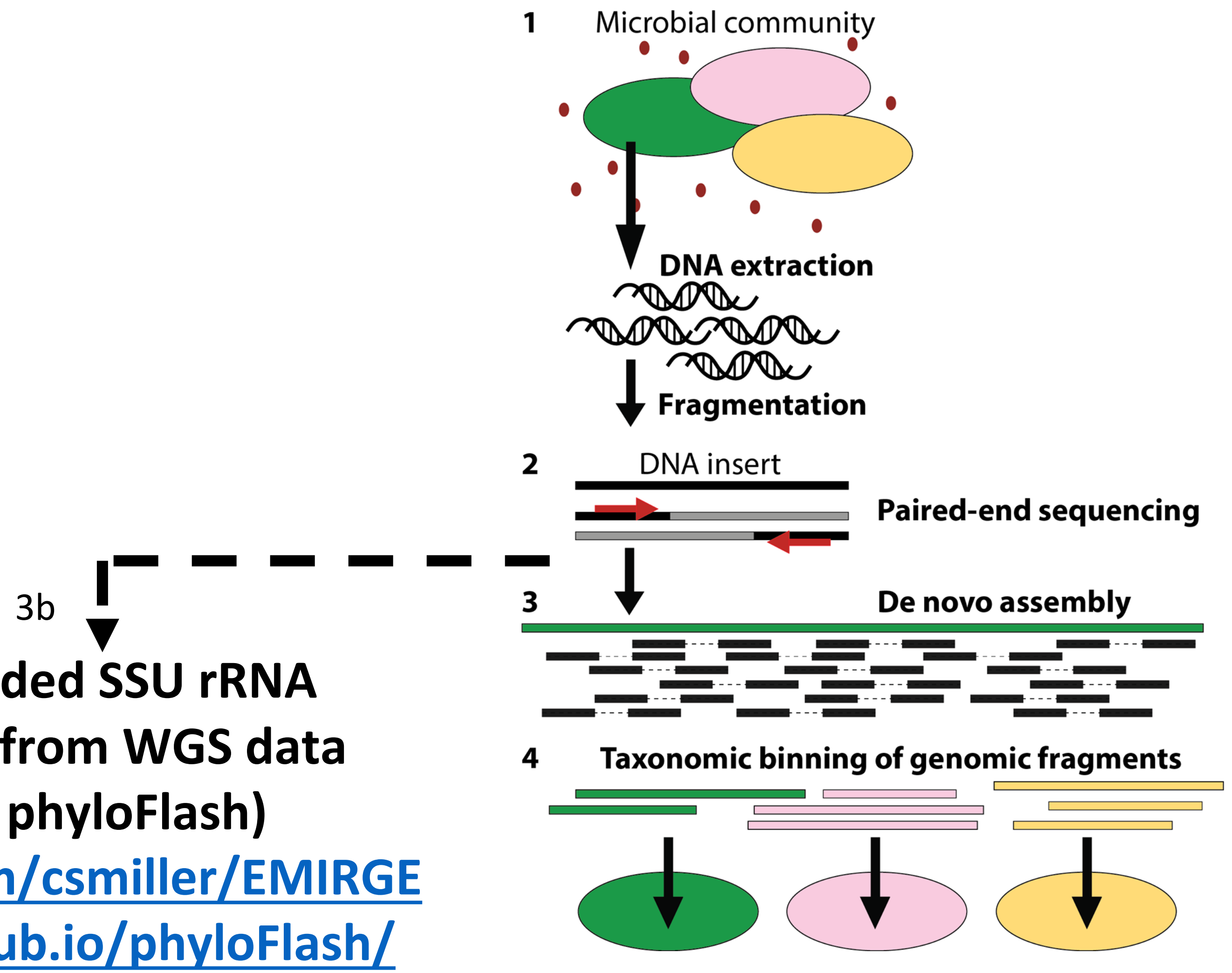
SSU rRNA reference guided and iterative assembly


correct tiling of de novo contigs
1
reconstructed SSU rRNA gene
1549
1.0
Pr(N)
0.0
160
position
220

Incorrect in *de novo* assembly

(Miller et al., 2011, Genome Biology)

Future considerations: rRNA genes

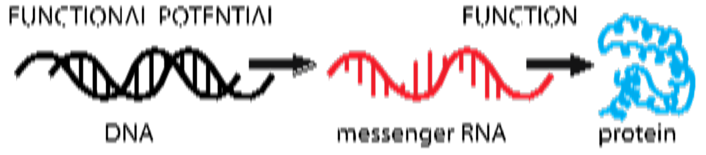


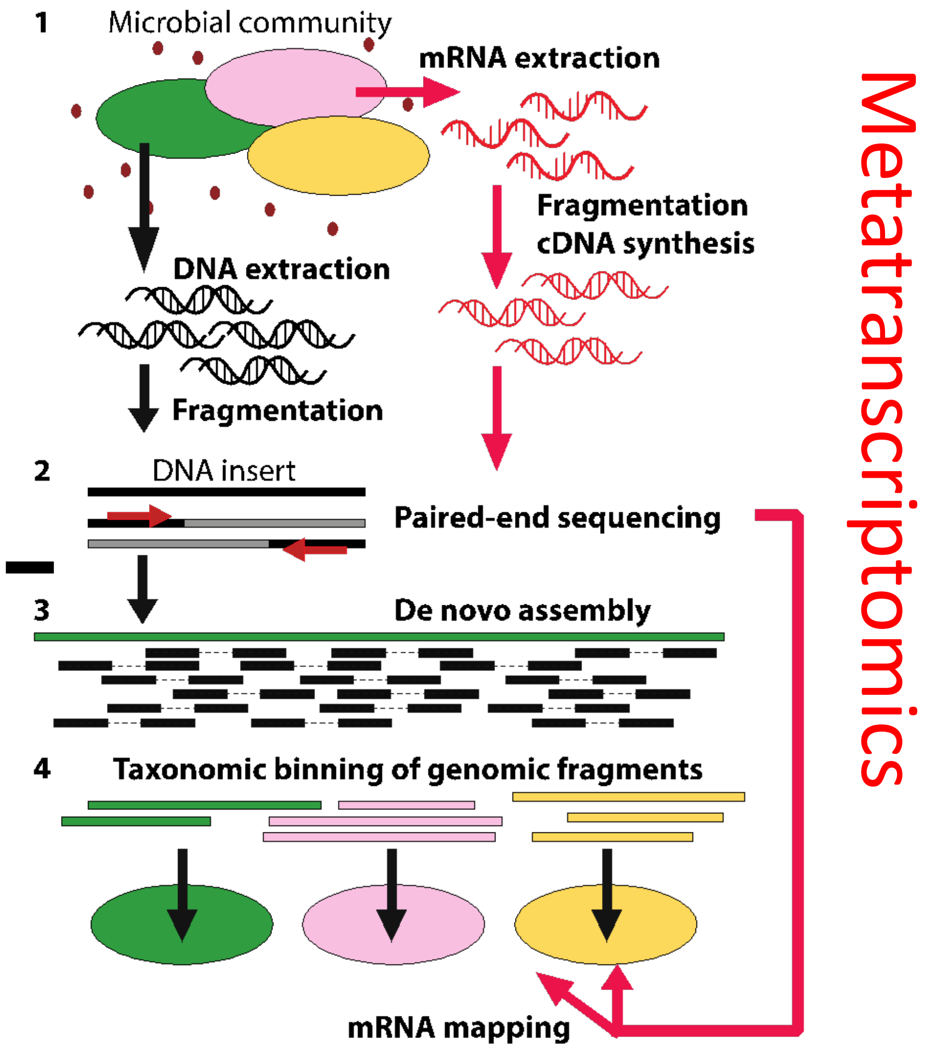
Reference-guided SSU rRNA reconstruction from WGS data (EMIRGE or phyloFlash)

https://github.com/csmiller/EMIRGE

https://hrgv.github.io/phyloFlash/

Future considerations: mRNA





3b

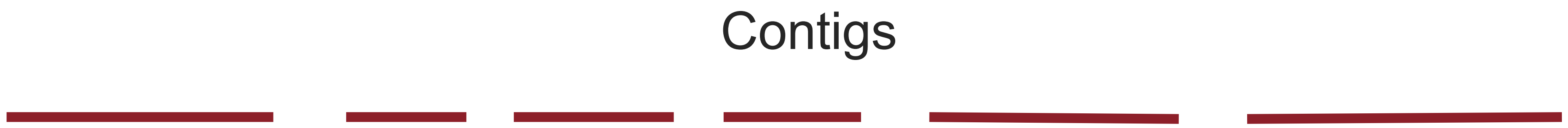
Reference-guided SSU rRNA reconstruction from WGS data (EMIRGE or phyloFLASH)

https://github.com/csmiller/EMIRGE

https://hrgv.github.io/phyloFlash/

Genome assembly

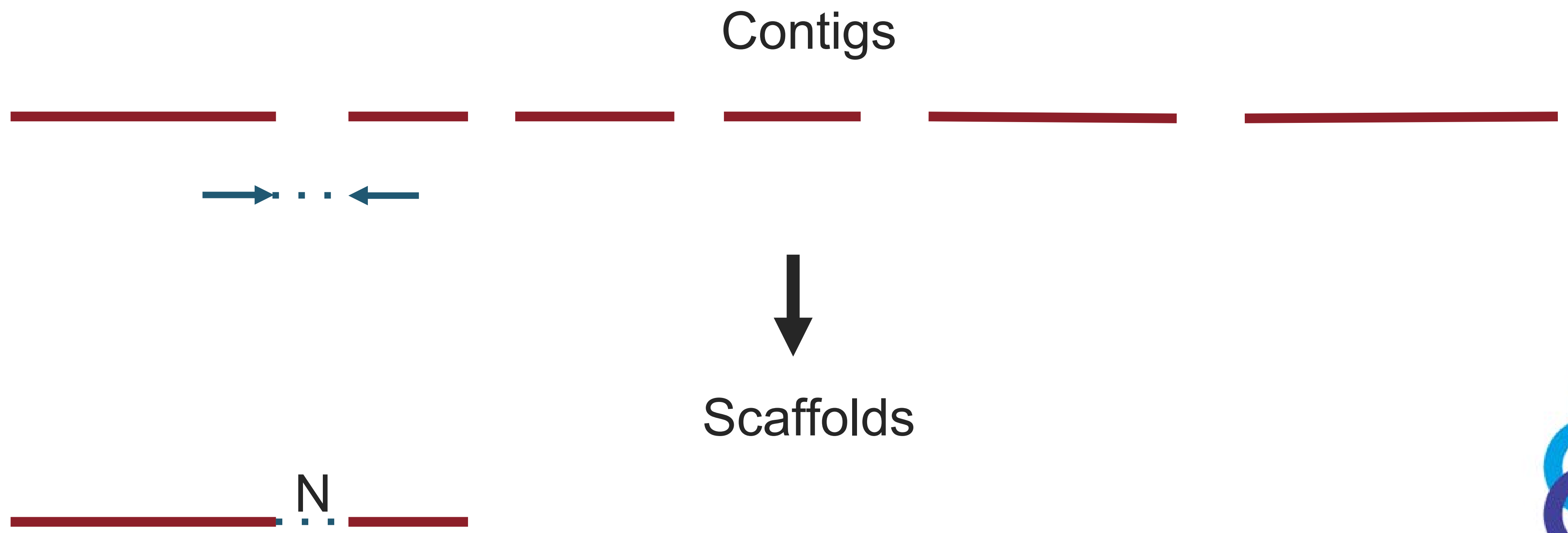
Contigs vs Scaffolds



Genome assembly

Contigs vs Scaffolds

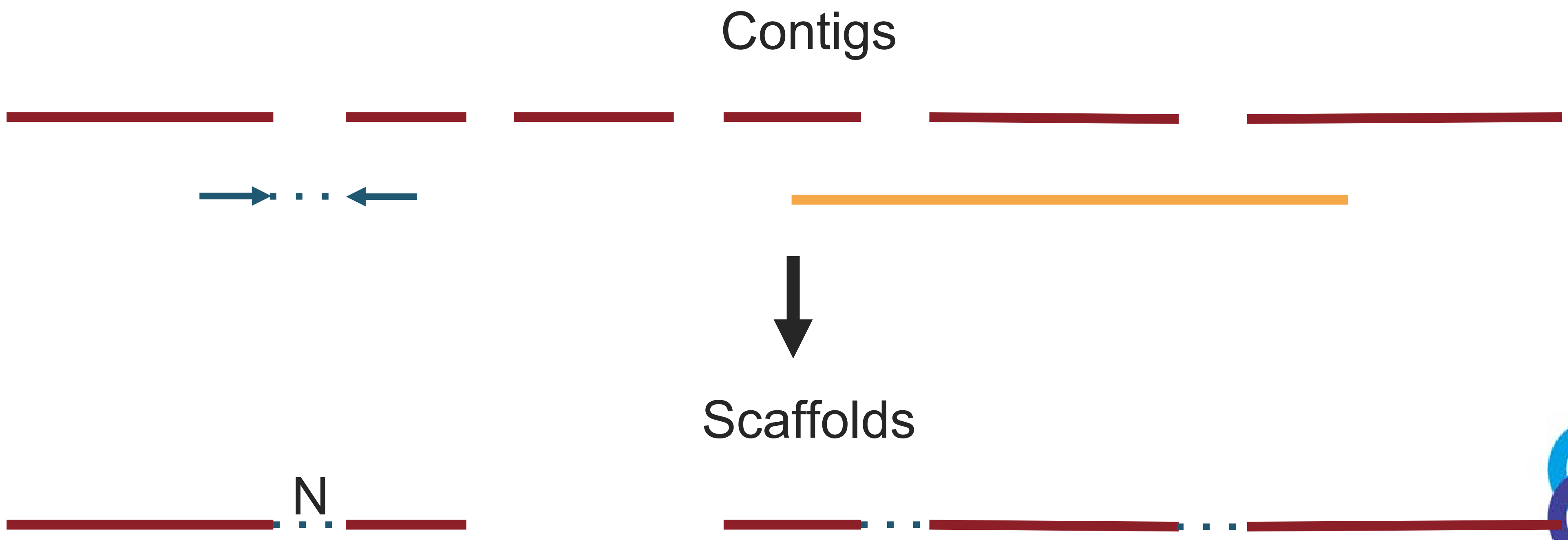
- Overlapping insert



Genome assembly

Contigs vs Scaffolds

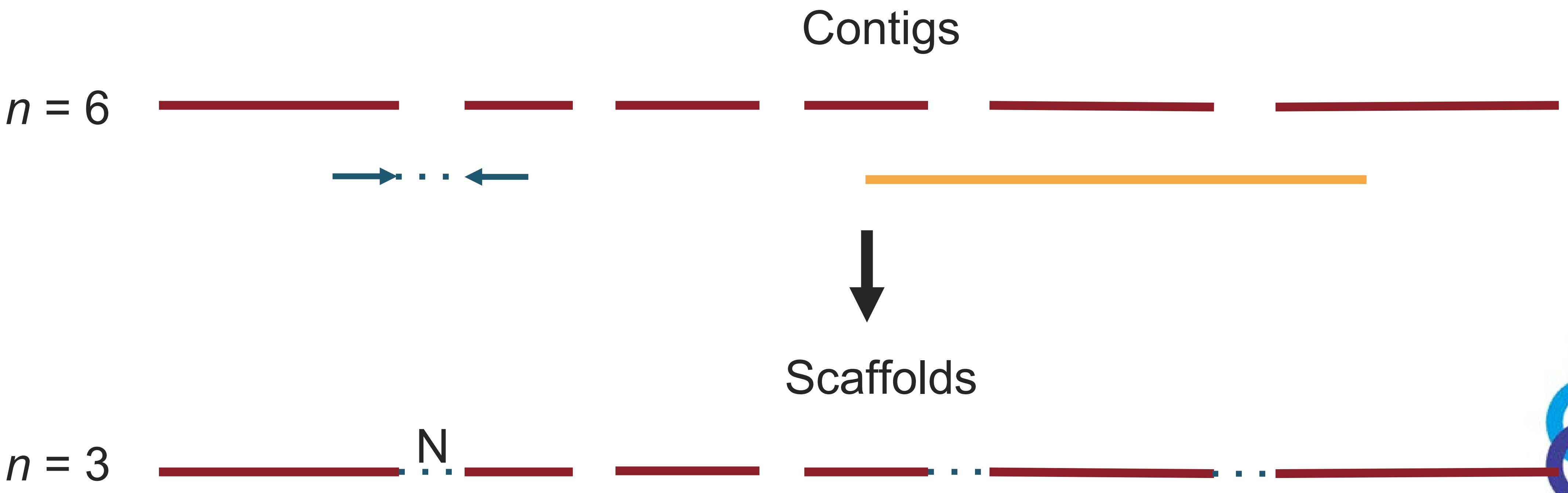
- Overlapping insert
- Long read sequencing (hybrid assembly)



Genome assembly

Contigs vs Scaffolds

- Overlapping insert
- Long read sequencing (hybrid assembly)



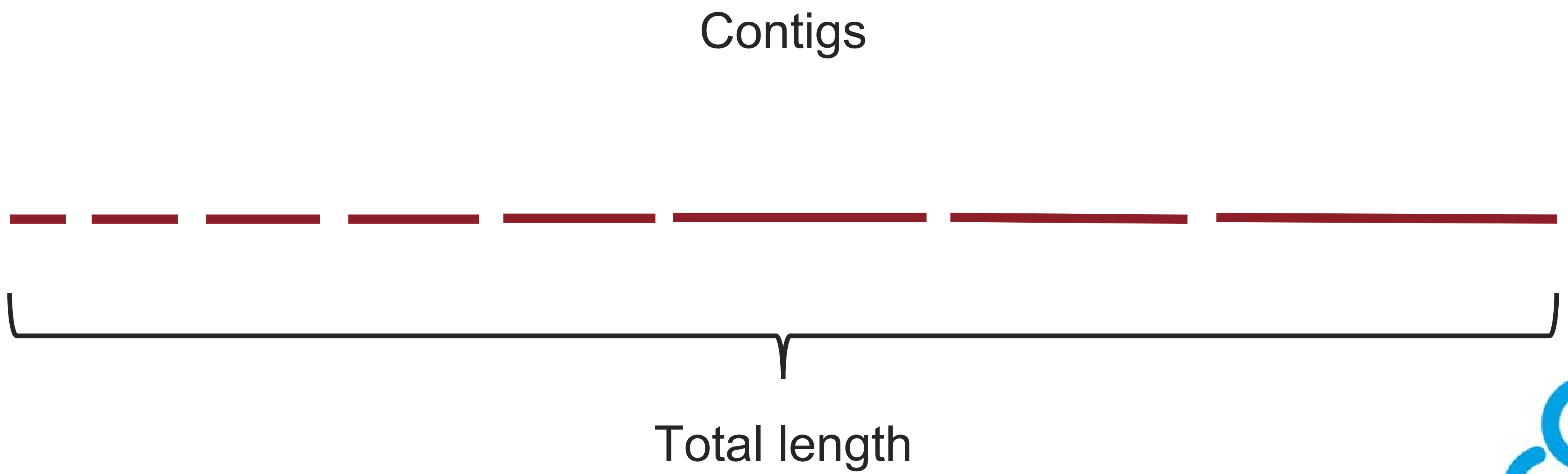
Assembly evaluation

Parameters to use in evaluation:

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

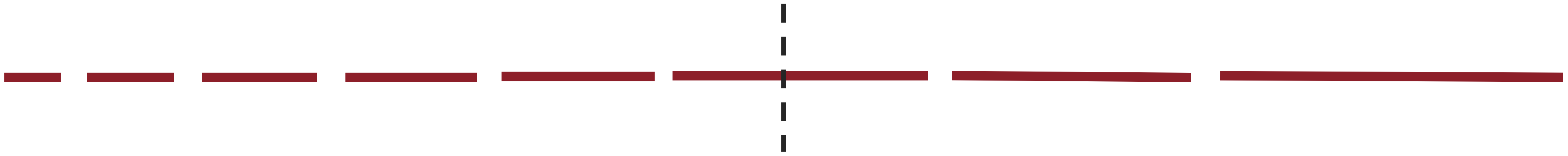
Assembly evaluation

N50 vs L50



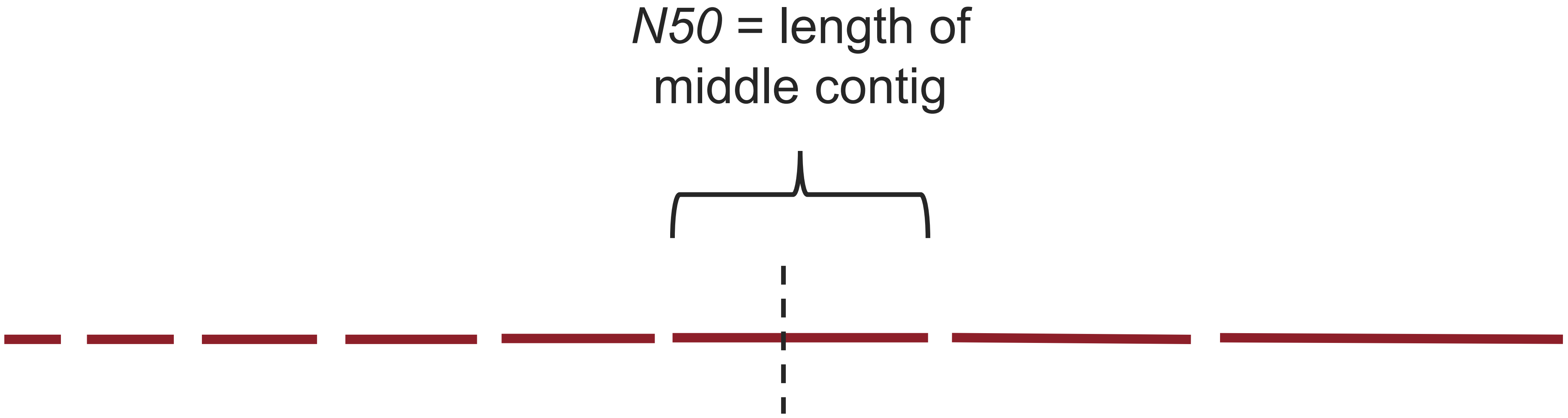
Assembly evaluation

N50 vs L50



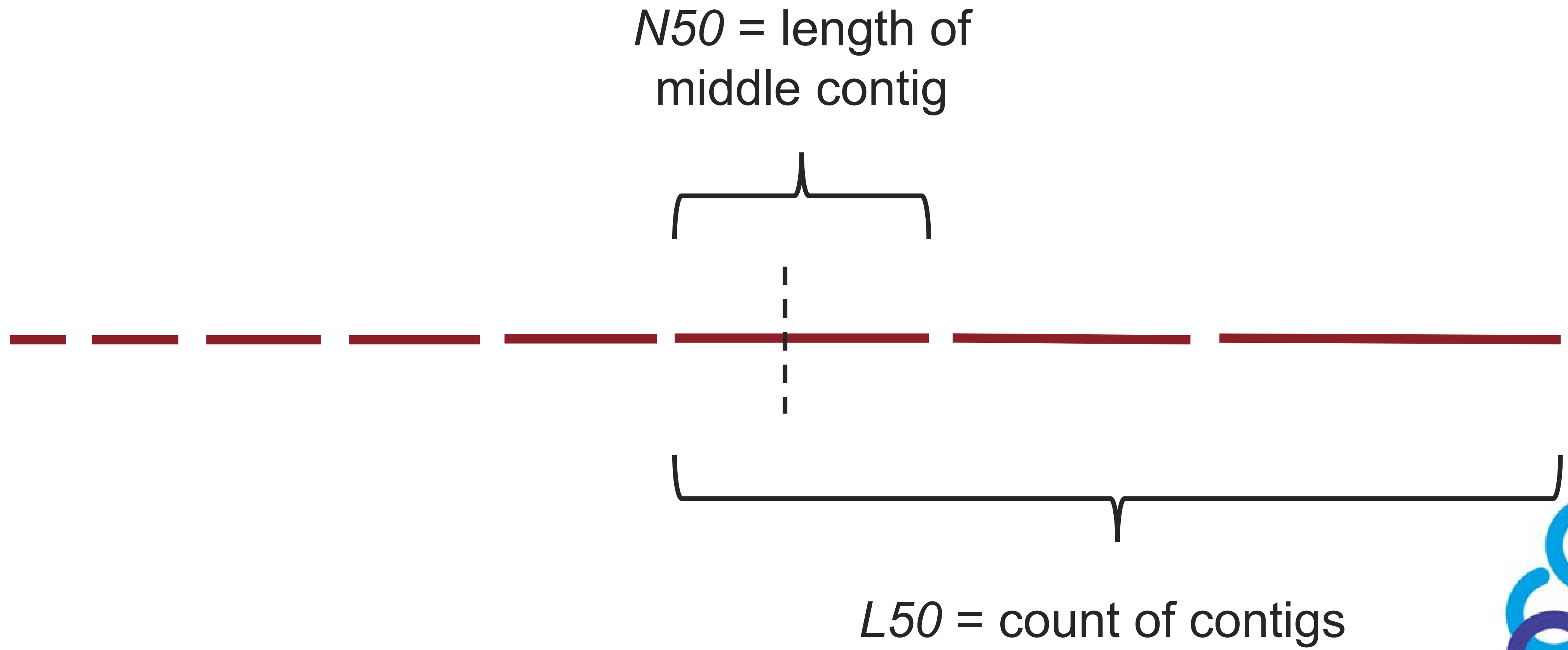
Assembly evaluation

N50 vs L50

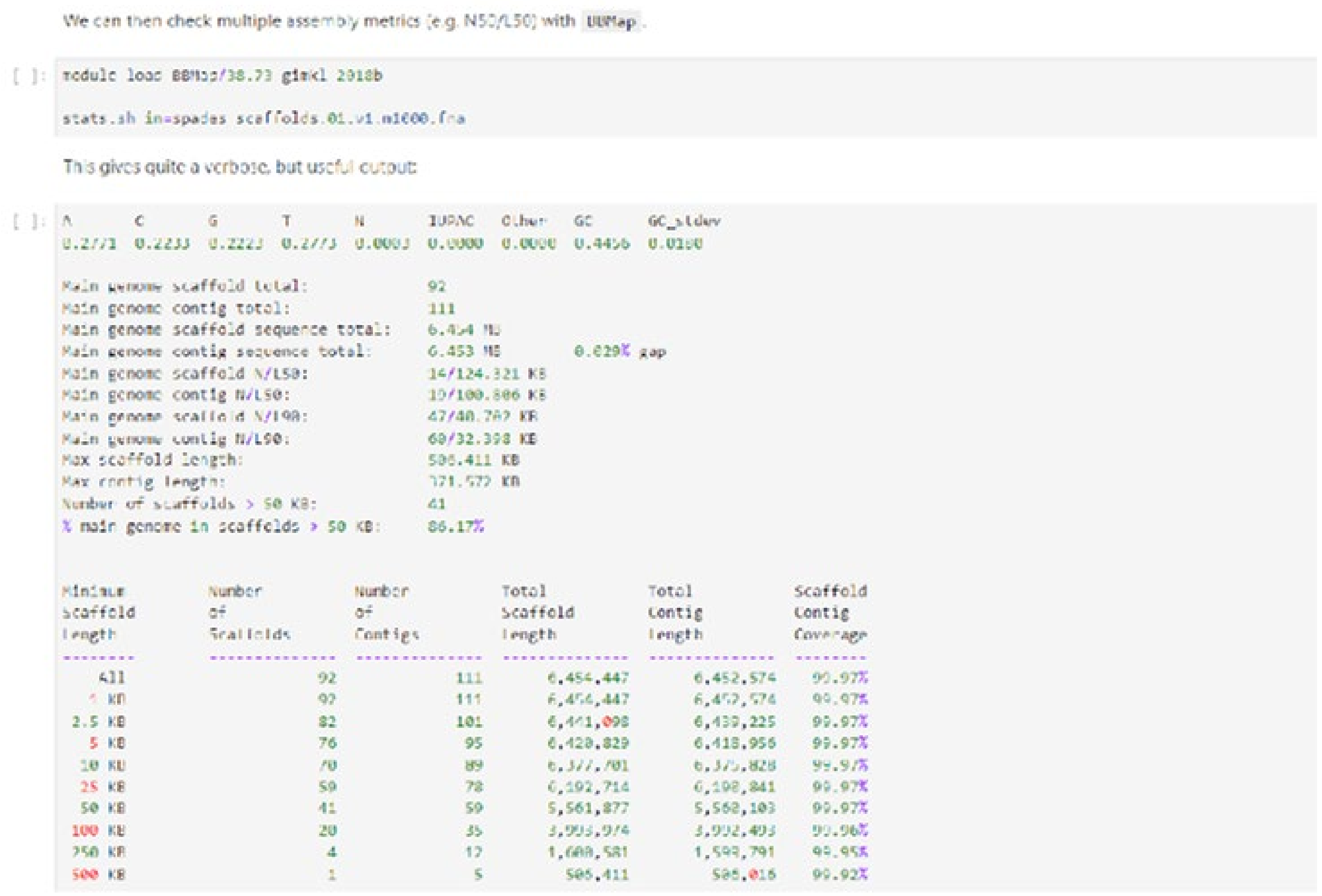


Assembly evaluation

N50 vs L50



Assembly evaluation



We can then check multiple assembly metrics (e.g. N50/L50) with `BBMap`.

```
[ ]: module load BBMap/38.73 gimkl 2018b

     stats.sh in=spades scaffolds.01.v1.m1000.fna
```

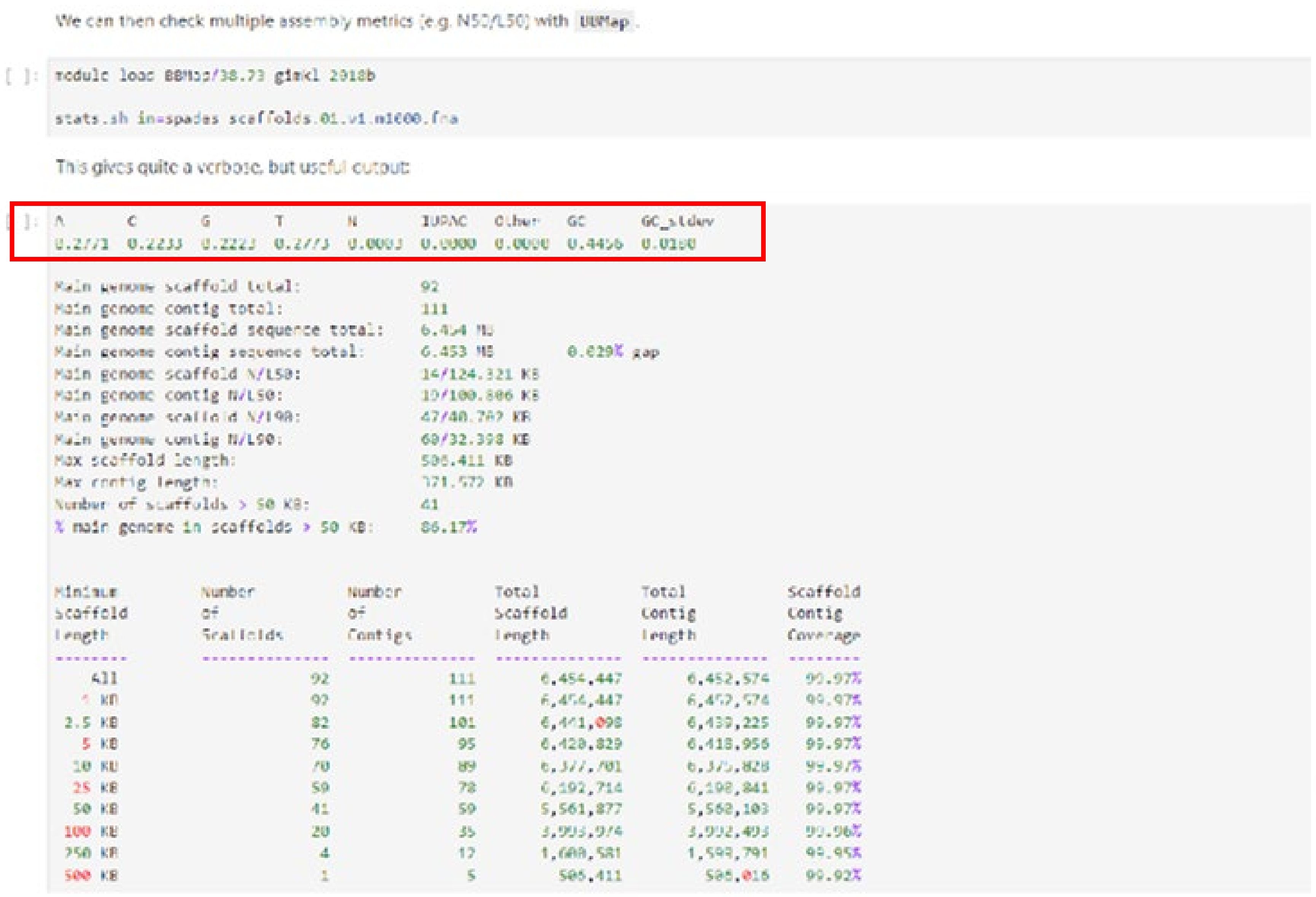
This gives quite a verbose, but useful output:

```
[ ]: A       C       G       T       N       IUPAC   Other   GC      GC_stdev
     0.2771  0.2233  0.2223  0.2773  0.0003  0.0000  0.0000  0.4456  0.0180

     Main genome scaffold total:             92
     Main genome contig total:               111
     Main genome scaffold sequence total:    6.454 MB
     Main genome contig sequence total:      6.453 MB        0.029% gap
     Main genome scaffold N/L50:             14/124.321 KB
     Main genome contig N/L50:               19/100.806 KB
     Main genome scaffold N/L90:             47/40.762 KB
     Main genome contig N/L90:               60/32.398 KB
     Max scaffold length:                    506.411 KB
     Max contig length:                      371.572 KB
     Number of scaffolds > 50 KB:            41
     % main genome in scaffolds > 50 KB:     86.17%


     Minimum         Number          Number          Total           Total           Scaffold
     Scaffold        of              of              Scaffold        Contig          Contig
     Length          Scaffolds       Contigs         Length          Length          Coverage
     --------        --------------  --------------  --------------  --------------  --------
         All                     92             111       6,454,447       6,452,574    99.97%
        1 KB                     92             111       6,454,447       6,452,574    99.97%
      2.5 KB                     82             101       6,441,098       6,439,225    99.97%
        5 KB                     76              95       6,420,829       6,418,956    99.97%
       10 KB                     70              89       6,377,701       6,375,828    99.97%
       25 KB                     59              78       6,192,714       6,190,841    99.97%
       50 KB                     41              59       5,561,877       5,560,103    99.97%
      100 KB                     20              35       3,995,974       3,992,493    99.96%
      250 KB                      4              12       1,600,581       1,599,791    99.95%
      500 KB                      1               5         506,411         506,016    99.92%
```

Assembly evaluation



We can then check multiple assembly metrics (e.g. N50/L50) with `BBMap`.

```
[ ]: module load BBMap/38.73 gimkl 2018b

     stats.sh in=spades scaffolds.01.v1.m1000.fna
```

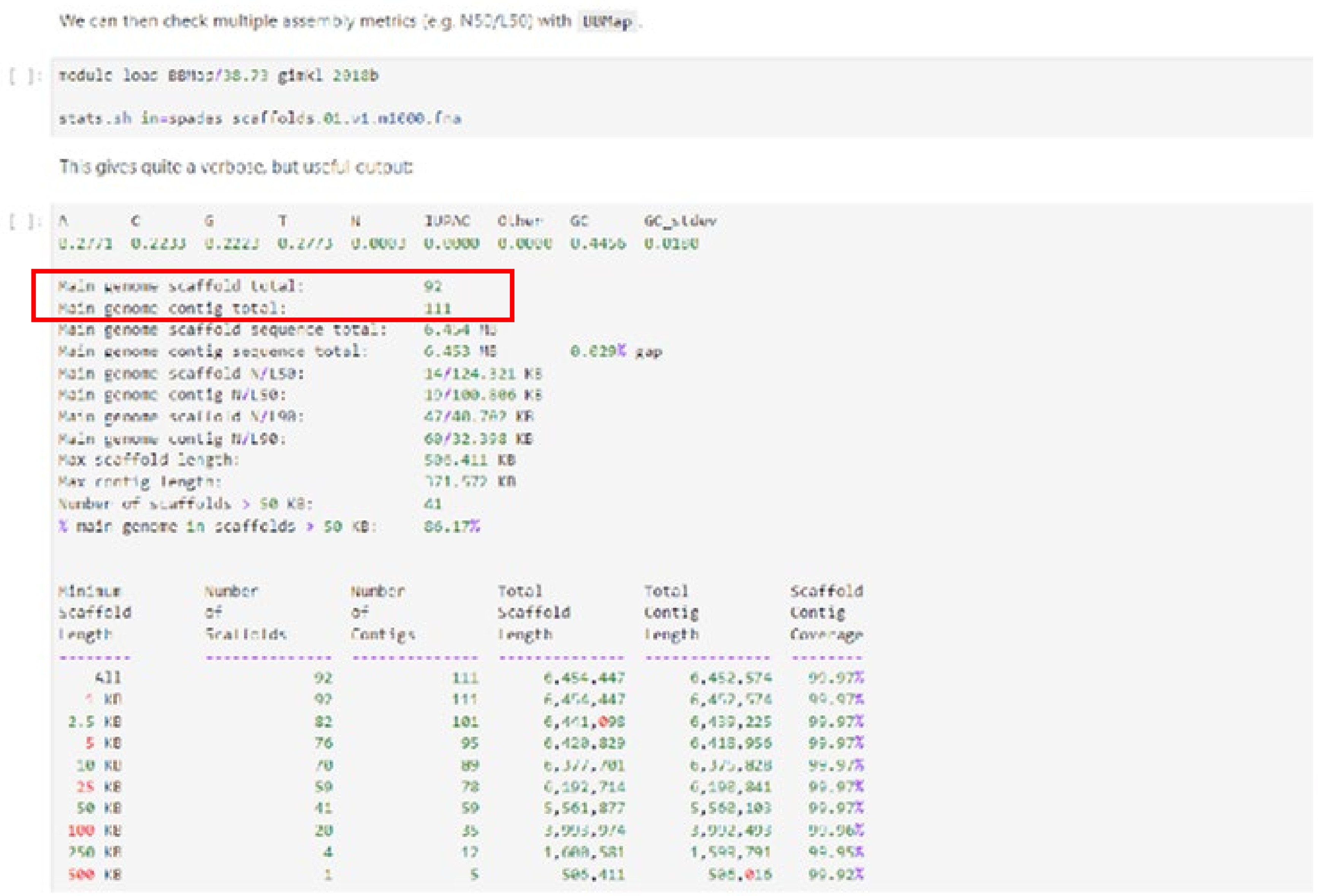
This gives quite a verbose, but useful output:

```
[ ]: A       C       G       T       N       IUPAC   Other   GC      GC_stdev
     0.2771  0.2233  0.2223  0.2773  0.0003  0.0000  0.0000  0.4456  0.0180

     Main genome scaffold total:             92
     Main genome contig total:               111
     Main genome scaffold sequence total:    6.454 MB
     Main genome contig sequence total:      6.453 MB        0.029% gap
     Main genome scaffold N/L50:             14/124.321 KB
     Main genome contig N/L50:               19/100.806 KB
     Main genome scaffold N/L90:             47/40.762 KB
     Main genome contig N/L90:               60/32.398 KB
     Max scaffold length:                    506.411 KB
     Max contig length:                      371.572 KB
     Number of scaffolds > 50 KB:            41
     % main genome in scaffolds > 50 KB:     86.17%


     Minimum         Number          Number          Total           Total           Scaffold
     Scaffold        of              of              Scaffold        Contig          Contig
     Length          Scaffolds       Contigs         Length          Length          Coverage
     --------        --------------  --------------  --------------  --------------  --------
         All                     92             111       6,454,447       6,452,574    99.97%
        1 KB                     92             111       6,454,447       6,452,574    99.97%
      2.5 KB                     82             101       6,441,098       6,439,225    99.97%
        5 KB                     76              95       6,420,829       6,418,956    99.97%
       10 KB                     70              89       6,377,701       6,375,828    99.97%
       25 KB                     59              78       6,192,714       6,190,841    99.97%
       50 KB                     41              59       5,561,877       5,560,103    99.97%
      100 KB                     20              35       3,993,974       3,992,403    99.96%
      250 KB                      4              12       1,600,581       1,599,791    99.95%
      500 KB                      1               5         506,411         506,016    99.92%
```

Assembly evaluation



We can then check multiple assembly metrics (e.g. N50/L50) with `BBMap`.

```
[ ]: module load BBMap/38.73 gimkl 2018b

     stats.sh in=spades_scaffolds.01.v1.m1000.fna
```

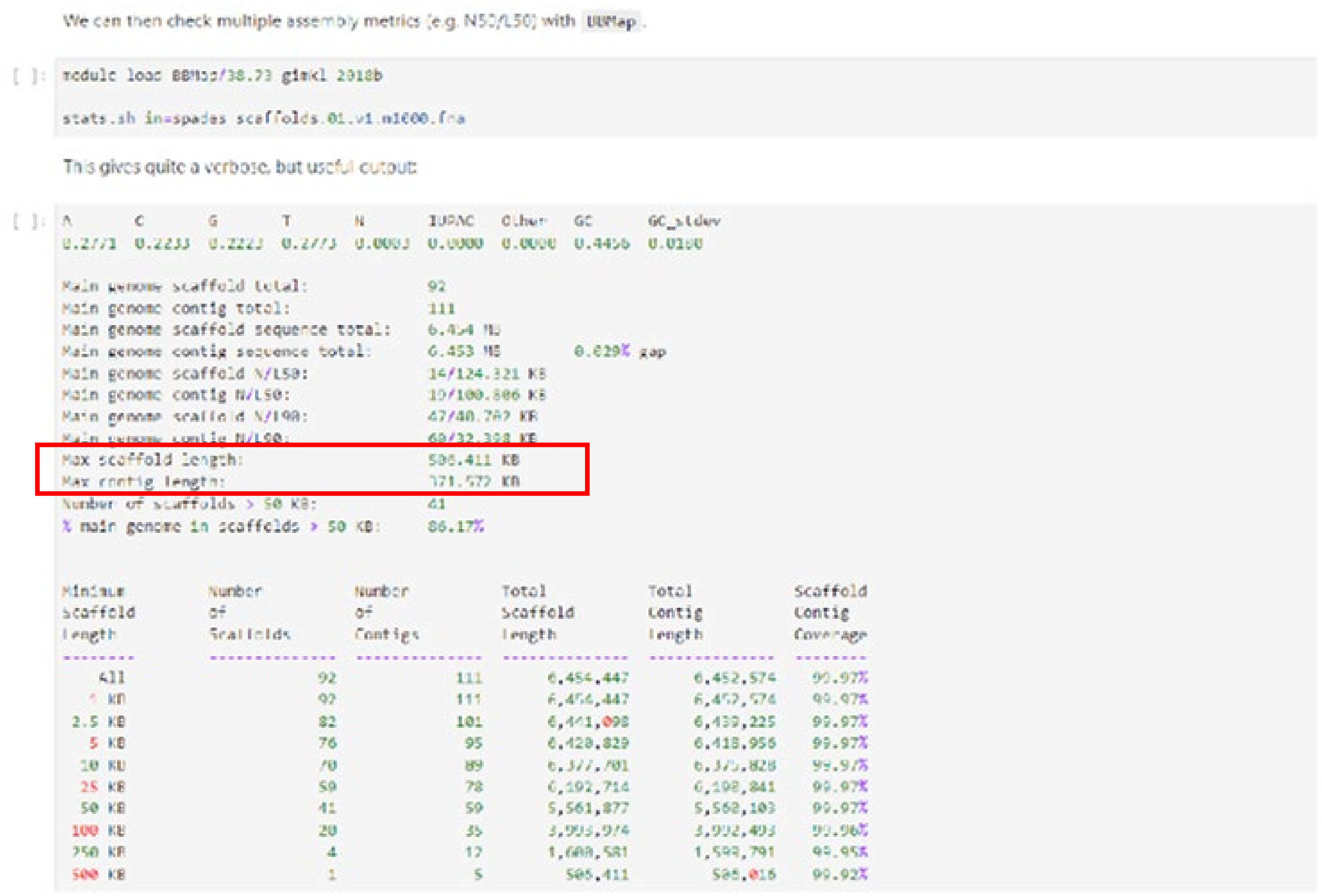
This gives quite a verbose, but useful output:

```
[ ]: A       C       G       T       N       IUPAC   Other   GC      GC_stdev
     0.2771  0.2233  0.2223  0.2773  0.0003  0.0000  0.0000  0.4456  0.0180

     Main genome scaffold total:             92
     Main genome contig total:               111
     Main genome scaffold sequence total:    6.454 MB
     Main genome contig sequence total:      6.453 MB        0.029% gap
     Main genome scaffold N/L50:             14/124.321 KB
     Main genome contig N/L50:               19/100.806 KB
     Main genome scaffold N/L90:             47/40.742 KB
     Main genome contig N/L90:               60/32.398 KB
     Max scaffold length:                    506.411 KB
     Max contig length:                      371.572 KB
     Number of scaffolds > 50 KB:            41
     % main genome in scaffolds > 50 KB:     86.17%


     Minimum     Number          Number          Total           Total           Scaffold
     Scaffold    of              of              Scaffold        Contig          Contig
     Length      Scaffolds       Contigs         Length          Length          Coverage
     --------    --------------  --------------  --------------  --------------  --------
         All                 92             111       6,454,447       6,452,574    99.97%
        1 KB                 92             111       6,454,447       6,452,574    99.97%
      2.5 KB                 82             101       6,441,098       6,439,225    99.97%
        5 KB                 76              95       6,420,829       6,418,956    99.97%
       10 KB                 70              89       6,377,701       6,375,828    99.97%
       25 KB                 59              78       6,192,714       6,190,841    99.97%
       50 KB                 41              59       5,561,877       5,560,103    99.97%
      100 KB                 20              35       3,995,974       3,992,493    99.96%
      250 KB                  4              12       1,600,581       1,599,791    99.95%
      500 KB                  1               5         506,411         506,018    99.92%
```

Assembly evaluation



We can then check multiple assembly metrics (e.g. N50/L50) with `BBMap`.

```
[ ]: module load BBMap/38.73 gimkl 2018b

     stats.sh in=spades scaffolds.01.v1.m1000.fna
```

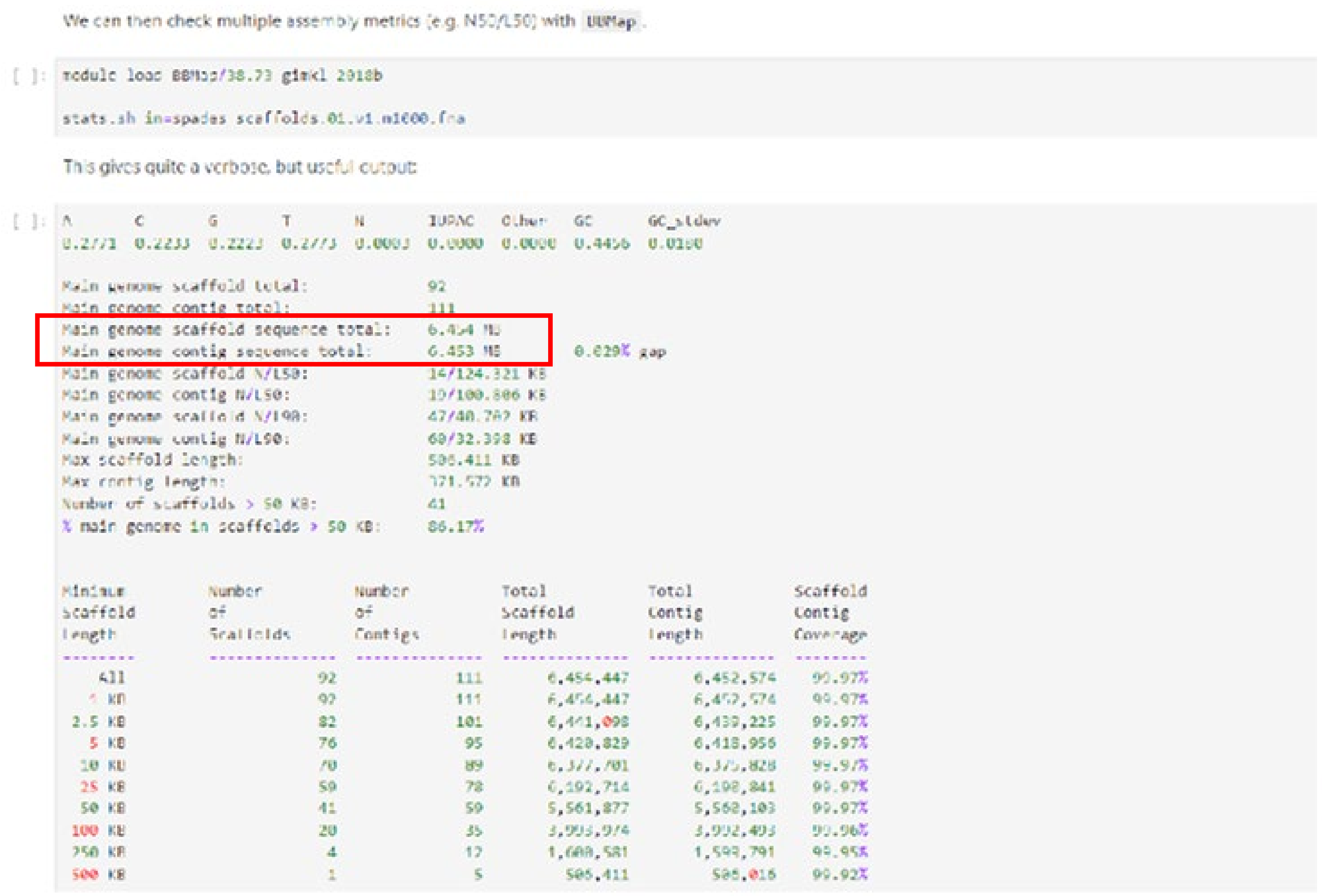
This gives quite a verbose, but useful output:

```
[ ]: A       C       G       T       N       IUPAC   Other   GC      GC_stdev
     0.2771  0.2233  0.2223  0.2773  0.0003  0.0000  0.0000  0.4456  0.0180

     Main genome scaffold total:             92
     Main genome contig total:               111
     Main genome scaffold sequence total:    6.454 MB
     Main genome contig sequence total:      6.453 MB        0.029% gap
     Main genome scaffold N/L50:             14/124.321 KB
     Main genome contig N/L50:               19/100.806 KB
     Main genome scaffold N/L90:             47/40.762 KB
     Main genome contig N/L90:               60/32.398 KB
     Max scaffold length:                    506.411 KB
     Max contig length:                      371.572 KB
     Number of scaffolds > 50 KB:            41
     % main genome in scaffolds > 50 KB:     86.17%


     Minimum         Number          Number          Total           Total           Scaffold
     Scaffold        of              of              Scaffold        Contig          Contig
     Length          Scaffolds       Contigs         Length          Length          Coverage
     --------        --------------  --------------  --------------  --------------  --------
         All                     92             111       6,454,447       6,452,574   99.97%
        1 KB                     92             111       6,454,447       6,452,574   99.97%
      2.5 KB                     82             101       6,441,098       6,439,225   99.97%
        5 KB                     76              95       6,420,829       6,418,956   99.97%
       10 KB                     70              89       6,377,701       6,375,828   99.97%
       25 KB                     59              78       6,192,714       6,190,841   99.97%
       50 KB                     41              59       5,561,877       5,560,103   99.97%
      100 KB                     20              35       3,993,974       3,992,403   99.96%
      250 KB                      4              12       1,600,581       1,599,791   99.95%
      500 KB                      1               5         506,411         506,016   99.92%
```

Assembly evaluation



We can then check multiple assembly metrics (e.g. N50/L50) with `BBMap`.

```
[ ]: module load BBMap/38.73 gimkl 2018b

     stats.sh in=spades_scaffolds.01.v1.m1000.fna
```

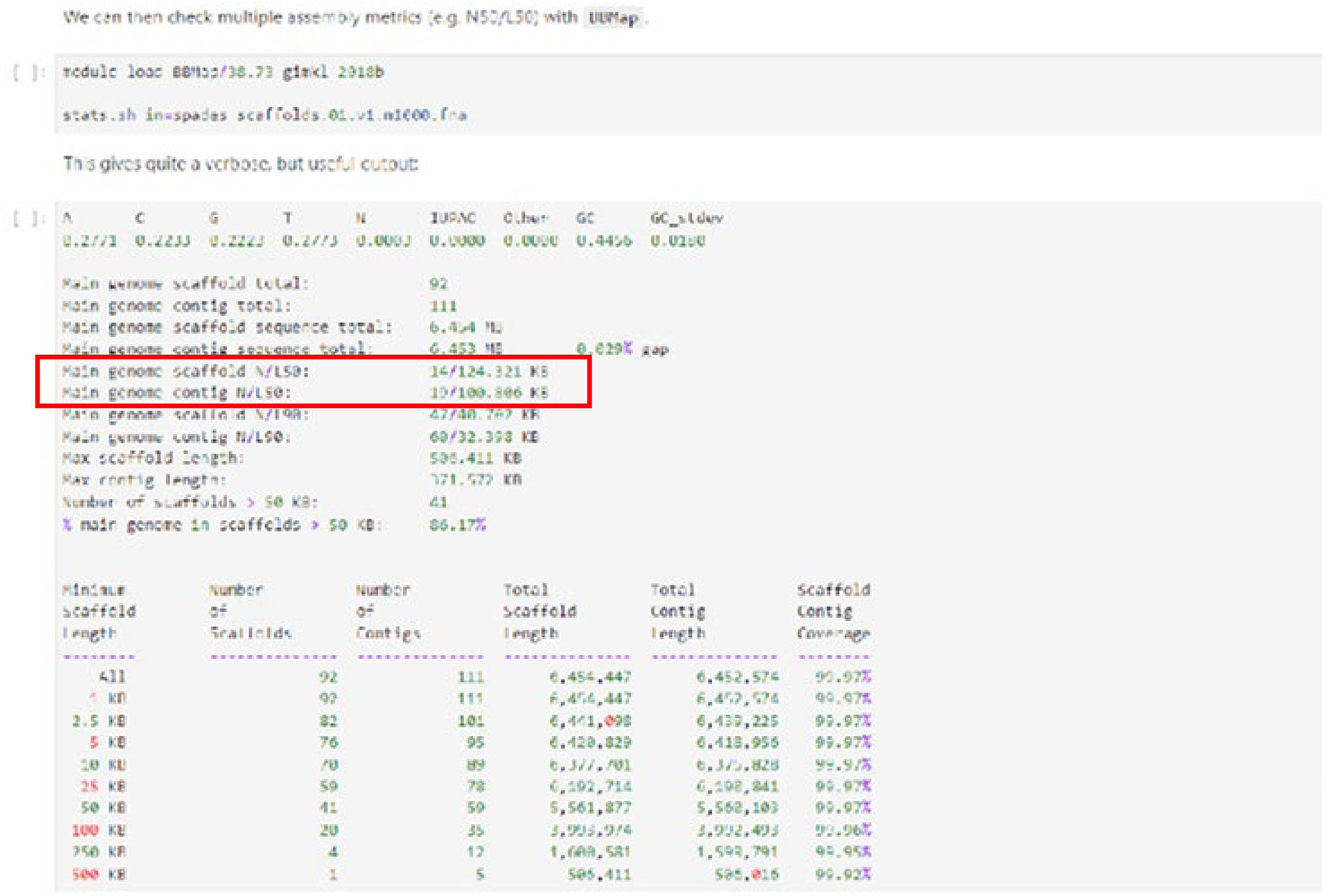
This gives quite a verbose, but useful output:

```
[ ]: A       C       G       T       N       IUPAC   Other   GC      GC_stdev
     0.2771  0.2233  0.2223  0.2773  0.0003  0.0000  0.0000  0.4456  0.0180

     Main genome scaffold total:            92
     Main genome contig total:              111
     Main genome scaffold sequence total:   6.454 MB
     Main genome contig sequence total:     6.453 MB        0.029% gap
     Main genome scaffold N/L50:            14/124.321 KB
     Main genome contig N/L50:              19/100.806 KB
     Main genome scaffold N/L90:            47/40.762 KB
     Main genome contig N/L90:              60/32.398 KB
     Max scaffold length:                   506.411 KB
     Max contig length:                     371.572 KB
     Number of scaffolds > 50 KB:           41
     % main genome in scaffolds > 50 KB:    86.17%


     Minimum       Number          Number          Total           Total           Scaffold
     Scaffold      of              of              Scaffold        Contig          Contig
     Length        Scaffolds       Contigs         Length          Length          Coverage
     --------      --------------  --------------  --------------  --------------  --------
         All                   92             111       6,454,447       6,452,574    99.97%
        1 KB                   92             111       6,454,447       6,452,574    99.97%
      2.5 KB                   82             101       6,441,098       6,439,225    99.97%
        5 KB                   76              95       6,420,829       6,418,956    99.97%
       10 KB                   70              89       6,377,701       6,375,828    99.97%
       25 KB                   59              78       6,192,714       6,190,841    99.97%
       50 KB                   41              59       5,561,877       5,560,103    99.97%
      100 KB                   20              35       3,993,974       3,992,403    99.96%
      250 KB                    4              12       1,600,581       1,599,791    99.95%
      500 KB                    1               5         506,411         506,016    99.92%
```

Assembly evaluation



We can then check multiple assembly metrics (e.g. N50/L50) with `BBMap`.

```
[ ]: module load BBMap/38.73 gimkl 2018b

     stats.sh in=spades scaffolds.01.v1.m1000.fna
```

This gives quite a verbose, but useful output:

```
[ ]: A       C       G       T       N       IUPAC   Other   GC      GC_stdev
     0.2771  0.2233  0.2223  0.2773  0.0003  0.0000  0.0000  0.4456  0.0180

     Main genome scaffold total:             92
     Main genome contig total:               111
     Main genome scaffold sequence total:    6.454 MB
     Main genome contig sequence total:      6.453 MB        0.029% gap
     Main genome scaffold N/L50:             14/124.321 KB
     Main genome contig N/L50:               19/100.806 KB
     Main genome scaffold N/L90:             47/40.767 KB
     Main genome contig N/L90:               60/32.398 KB
     Max scaffold length:                    506.411 KB
     Max contig length:                      371.572 KB
     Number of scaffolds > 50 KB:            41
     % main genome in scaffolds > 50 KB:     86.17%


     Minimum         Number          Number          Total           Total           Scaffold
     Scaffold        of              of              Scaffold        Contig          Contig
     Length          Scaffolds       Contigs         Length          Length          Coverage
     --------        --------------  --------------  --------------  --------------  --------
         All                     92             111       6,454,447       6,452,574    99.97%
        1 KB                     92             111       6,454,447       6,452,574    99.97%
      2.5 KB                     82             101       6,441,098       6,439,225    99.97%
        5 KB                     76              95       6,420,829       6,418,956    99.97%
       10 KB                     70              89       6,377,701       6,375,828    99.97%
       25 KB                     59              78       6,192,714       6,190,841    99.97%
       50 KB                     41              59       5,561,877       5,560,103    99.97%
      100 KB                     20              35       3,993,974       3,992,403    99.96%
      250 KB                      4              12       1,600,581       1,599,791    99.95%
      500 KB                      1               5         506,411         506,016    99.92%
```

Task: Assembly evaluation

Go to Github MGSS webpage

Tasks:

- **Assembly evaluation**
- **Short contig removal**