Whole Genome Sequencing and Bioinformatics SeqAfrica Training

4-7th March 2025
CHSU, Lilongwe

Marco van Zwetselaar

Niamh Lacy-Roberts

Day 2

The Fleming Fund Regional Grants

UKaid from the British people

DTU

SeqAfrica

The Fleming Fund
Regional Grants

ONT QC

The Fleming Fund
Regional Grants

QC for ONT data

- Before beginning any post sequencing analyses, is an important to perform QC to understand if your data meets certain requirements and matches your expectations from the sequencing run.

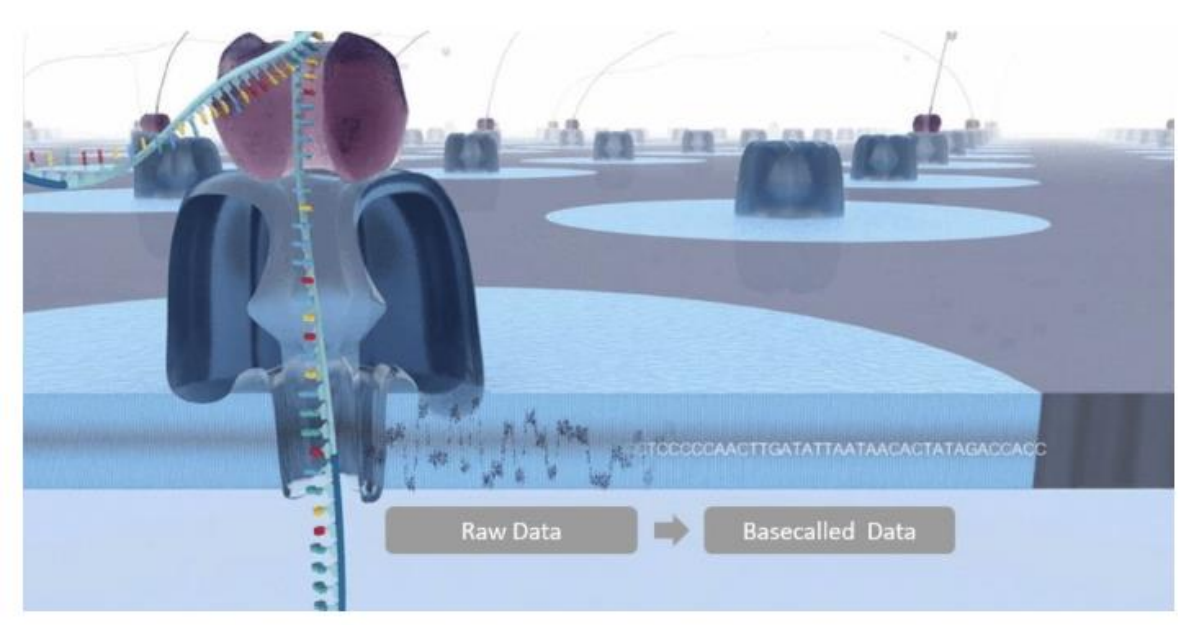
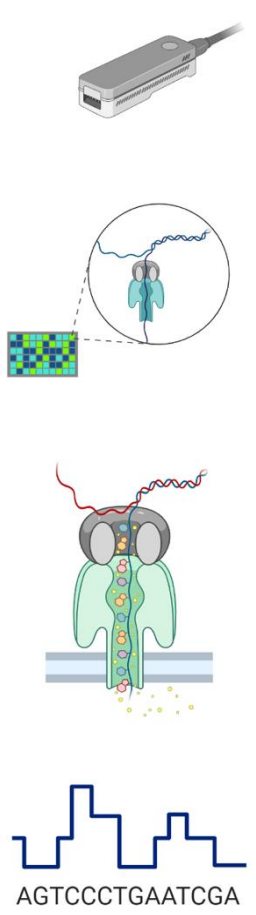


Image accessed from https://nanoporetech.com/platform/technology/basecalling

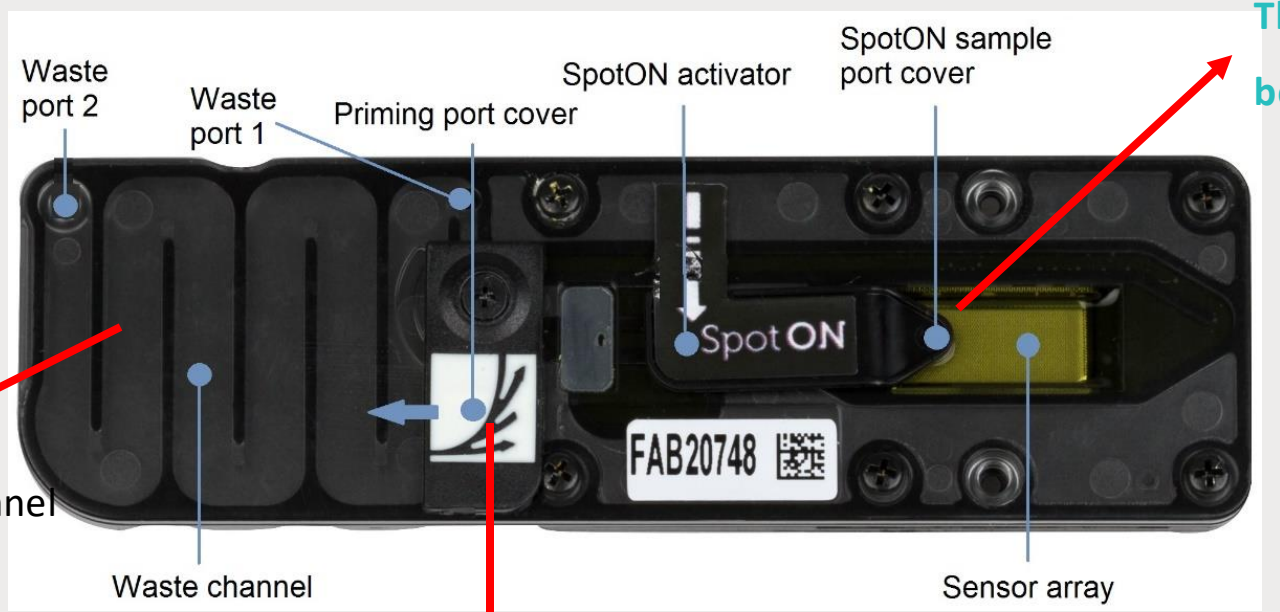
- Nanopore sequencing, can suffer from errors (basecalling errors, contamination, low-quality reads etc).
- We can use QC to identify and mitigate errors ☺
- While live basecalling, **MinKNOW** provides real-time feedback, such as read quality, read length and N50. This information is presented in an **interactive run report** and can be exported during and after the sequencing run.
- You can run **EPI2ME workflows** (e.g bacterial genomes) on the cloud or locally on your computer.

Why do errors occur?



- The pattern obtained from the nanopore needs to be interpreted.
  - Interpretation is based on machine learning models.
  - Signal varies depending on neighboring nucleotides, condition of pore (contamination, bubbles) and DNA speed variation (temperature issues).
- DNA string may slip in the pore during translocation.
- Short reads (<500 bp) tend to have worse quality and are often noise.
  - Longer reads provide better coverage but amplify homopolymer and repetitive regions issues.
- Stretches of homopolymers are difficult to call.
  - Repeated nucleotides (e.g., “TTTTT”) generate similar current disruptions.
  - Hard to distinguish between exact base counts (leading to insertions or deletions).
- Newer chemistry and updated basecalling models improves significantly on accuracy.

Bubbles in the flowcell



The SpotON port cover must be closed !!!!

Bubbles in the waste channel do not matter.

Pipette back via priming port

The Fleming Fund
Regional Grants

Examples of errors

Substitution Errors

- Incorrectly identifying one base as another (e.g., A → G).
    - Signal noise from current disruptions as DNA passes through the nanopore.
    - Inaccurate basecalling algorithms.

Affects gene sequences, especially for detecting SNPs and AMR mutations.

Insertion Errors

- Extra bases are called that do not exist in the actual sequence.
    - Signal misinterpretation due to homopolymer regions.
    - "AAAA" may be misread as "AAAAA."

Assembly errors, especially in repetitive or homopolymeric regions.

Deletion Errors

- Bases in the sequence are missed (e.g., a "C" is skipped).
    - Weak signal-to-noise ratio during strand passage.
    - Homopolymer stretches cause difficulty for nanopores.

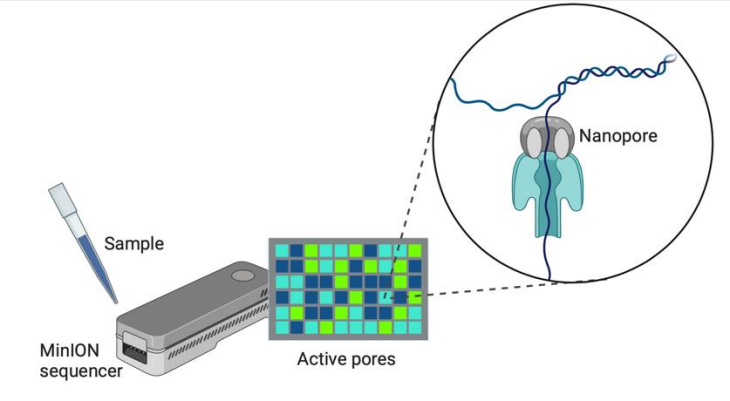
Loss of genetic information; problematic for gene annotation and AMR detection.

The Fleming Fund
Regional Grants

Errors in downstream analysis

Downstream Analysis	Impact of Errors
Genome Assembly	Fragmented assemblies, reduced N50, inaccurate contigs.
Variant Calling	Misidentification of SNPs, insertions, or deletions.
AMR Detection	False positives or false negatives in AMR gene predictions.
Phylogenetics	Errors propagate into phylogenetic trees, misleading clusters.

ONT sequencing output



Created in Biorender.com

Base calling

Fastq files containing 4000 reads (default)

```
@SRR1770413.1 1/1
CACCCGGCATCAGGTGCGGTACTTTTGCGCCTCCCAGCCGGACCGGCCCTGCGGCGTAATA
CCAGCCTCACATCCCTCGCTGCCTGCGTATCCAGCTCACTCTCCCTGGTTGCCGCCTACAT
GCTCCCTCCCGCTGTTCCACCCCTTTGCACCCCCCCTCTGCCCCTCCTGCTCGCCAGCCCC
+
CCCCCECFCEFC@8F8C77B7BFEFD,C+@@@BCB##########################
##############################################################
##############################################################
```

Also get POD5 files, QC report…

The Fleming Fund
Regional Grants

Read quality – Q scores

$$Q = -10 \cdot \log_{10}(P)$$

or in terms of probability

$$P = 10^{-\frac{Q}{10}}$$

Where

P = probability of incorrect base call

Q = Phred quality score

- The Phred quality score is a logarithmic score based on the probability that the base call (nucleotide) is incorrect

- Q10 = 1/10 risk of incorrect base
- Q20 = 1/100 risk of incorrect base
- Q30 = 1/1000 risk of incorrect base

- This means that in a sequence of 100 bp at Q20, there will most likely be at least 1 error.

Phred quality score	Probability of incorrect base call	Probability of being correct
10	0.1	90%
20	0.01	99%
30	0.001	99.9%

Read quality, read length and N50

- bp = base pair
- kb (= kbp) = kilo–base-pair = 1,000 bp
- Mb (= Mbp) = mega–base-pair = 1,000,000 bp
- Gb (= Gbp) = giga–base-pair = 1,000,000,000 bp

- Nanopore technology routinely generates sequencing reads that are tens of kilobases in length

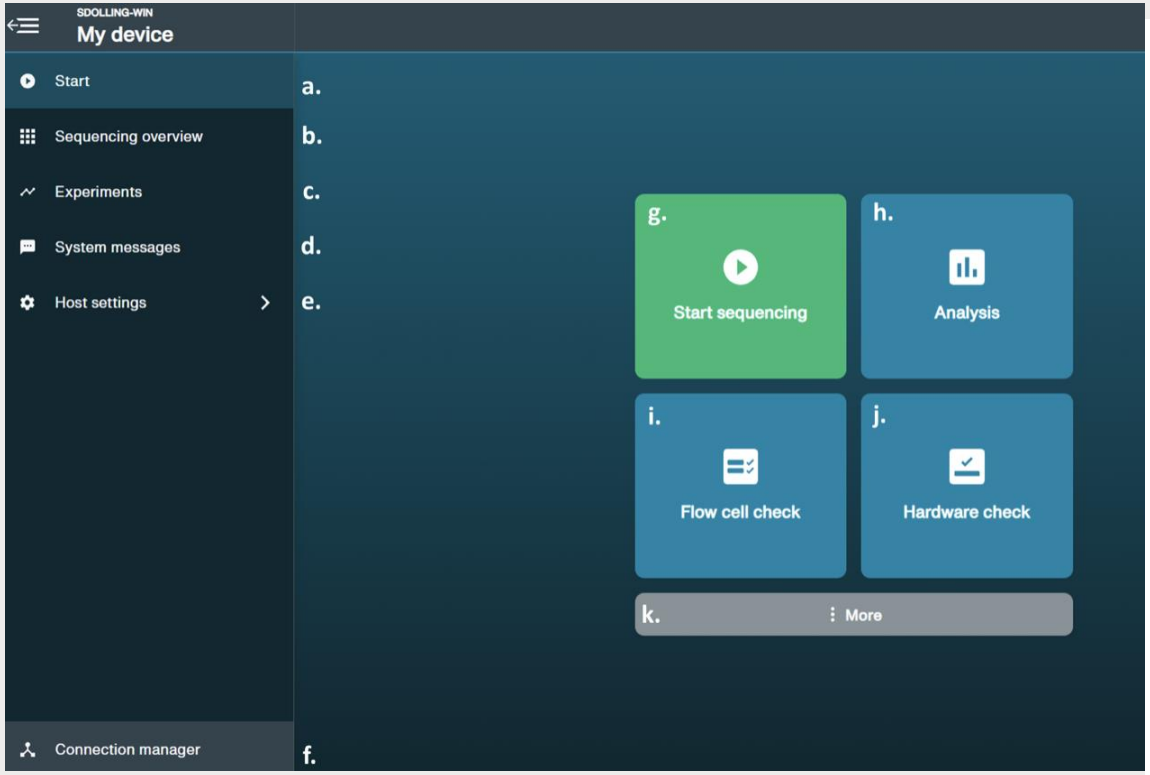
- The longest DNA fragment sequenced to date using nanopore technology is 4.2 Mb, which was achieved using the Ultra-Long DNA Sequencing Kit.

- N50 — the length at which half of the nucleotides in the fastq/assembly belong in reads/contigs of this length or longer.

- Default Q score is at least 10, this can specified when setrting up you sequencing.

The Fleming Fund
Regional Grants

MinKNOW Interface



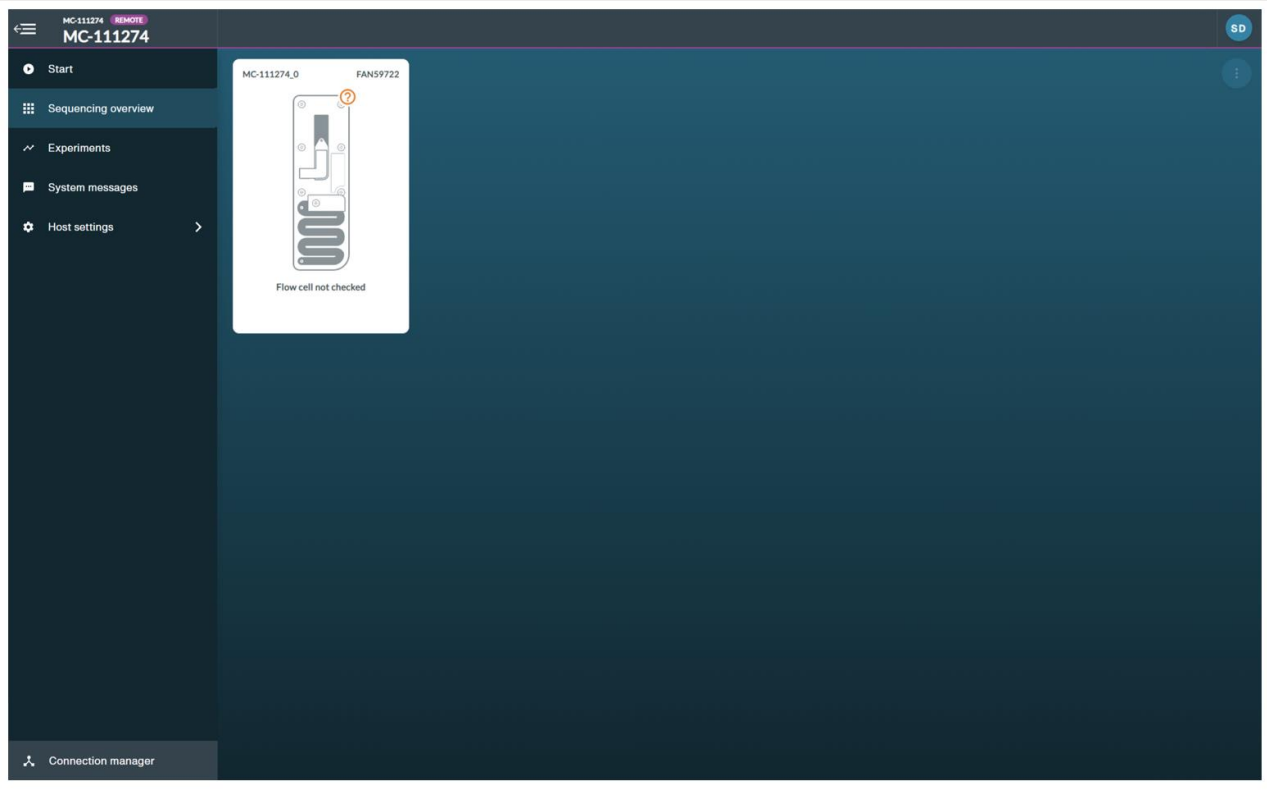
Homepage overview

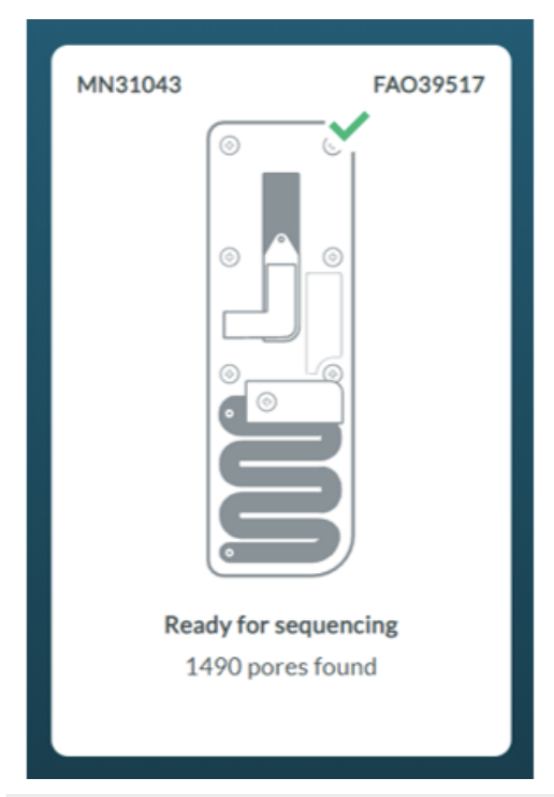
The MinKNOW Homepage enables the user to navigate to:

a. **Start** homepage
b. **Sequencing Overview** of connected flow cells
c. Recent and current **Experiments**
d. **System Messages**
e. **Host Settings**
f. **Connection Manager** to connect with other available devices
g. **Start Sequencing** experiment
h. Post-run **Analysis**
i. **Flow Cell Check**
j. **Hardware Check**
k. **More** includes option to generate .mmi from .fasta file or to import a sample sheet
l. **Guest/initials** to logout

Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

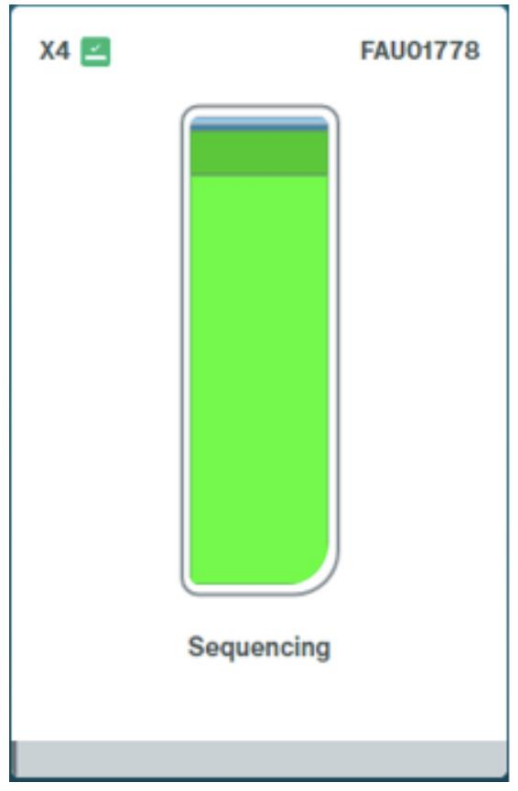
Flow Cell Check

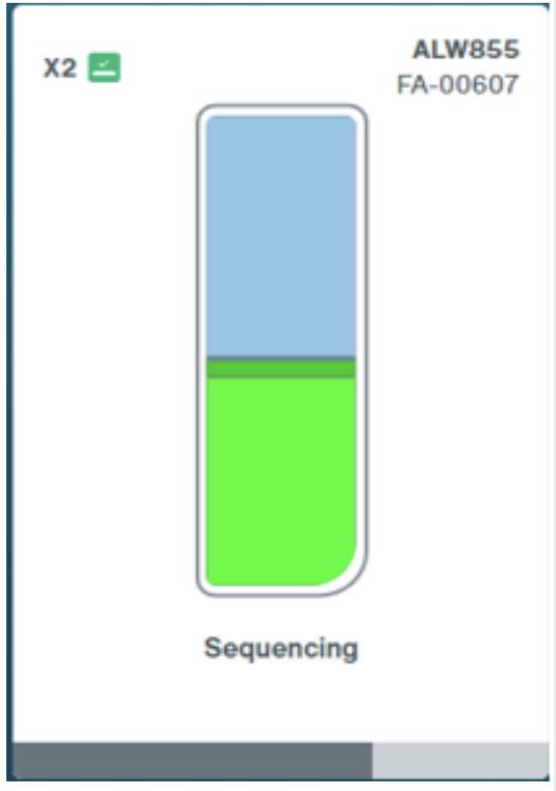




Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

Flow cell health

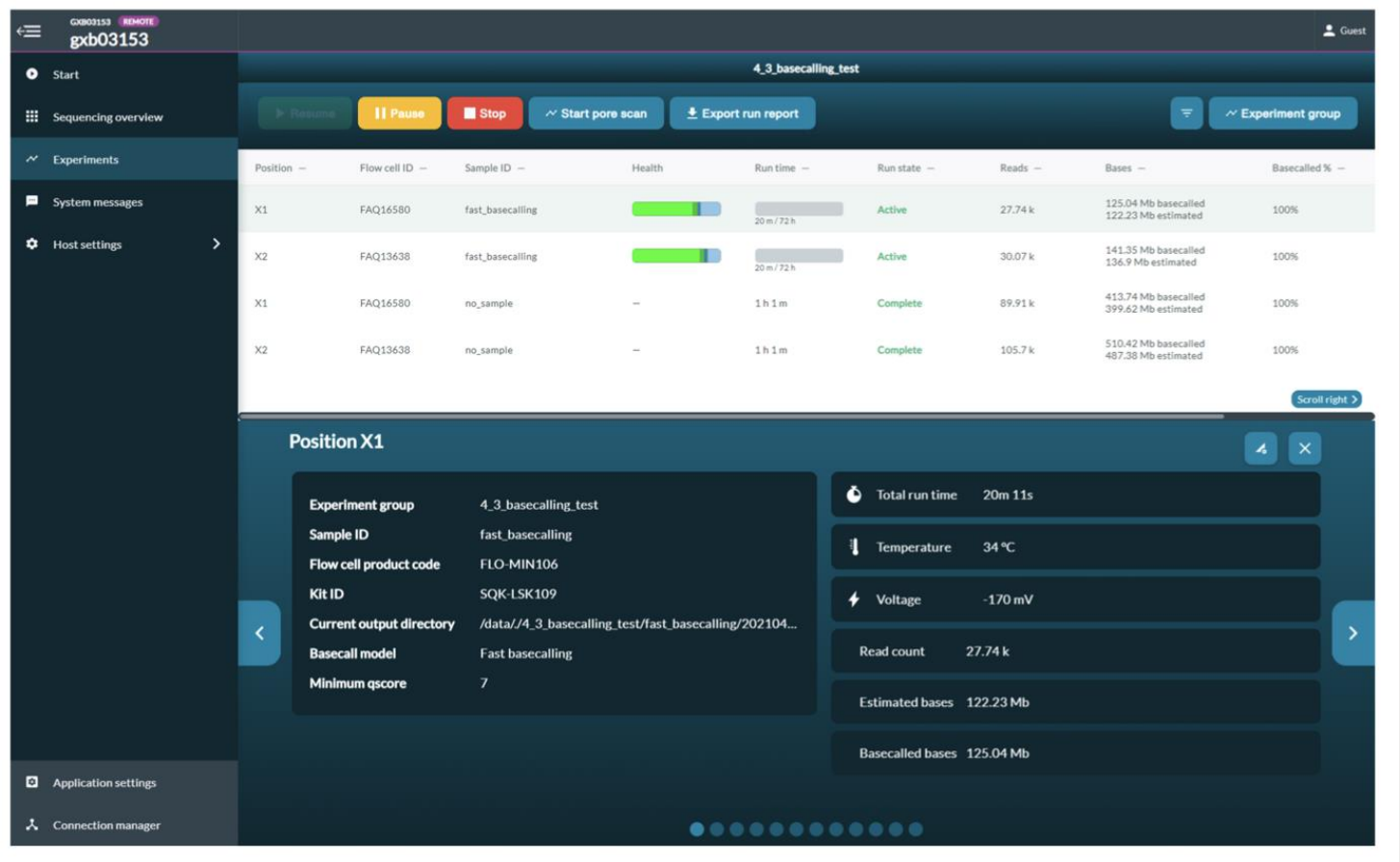




- During a sequencing experiment, the Sequencing Overview page shows a flow cell icon with coloured bars.
- The bars represent the combined health of all pores in a flow cell

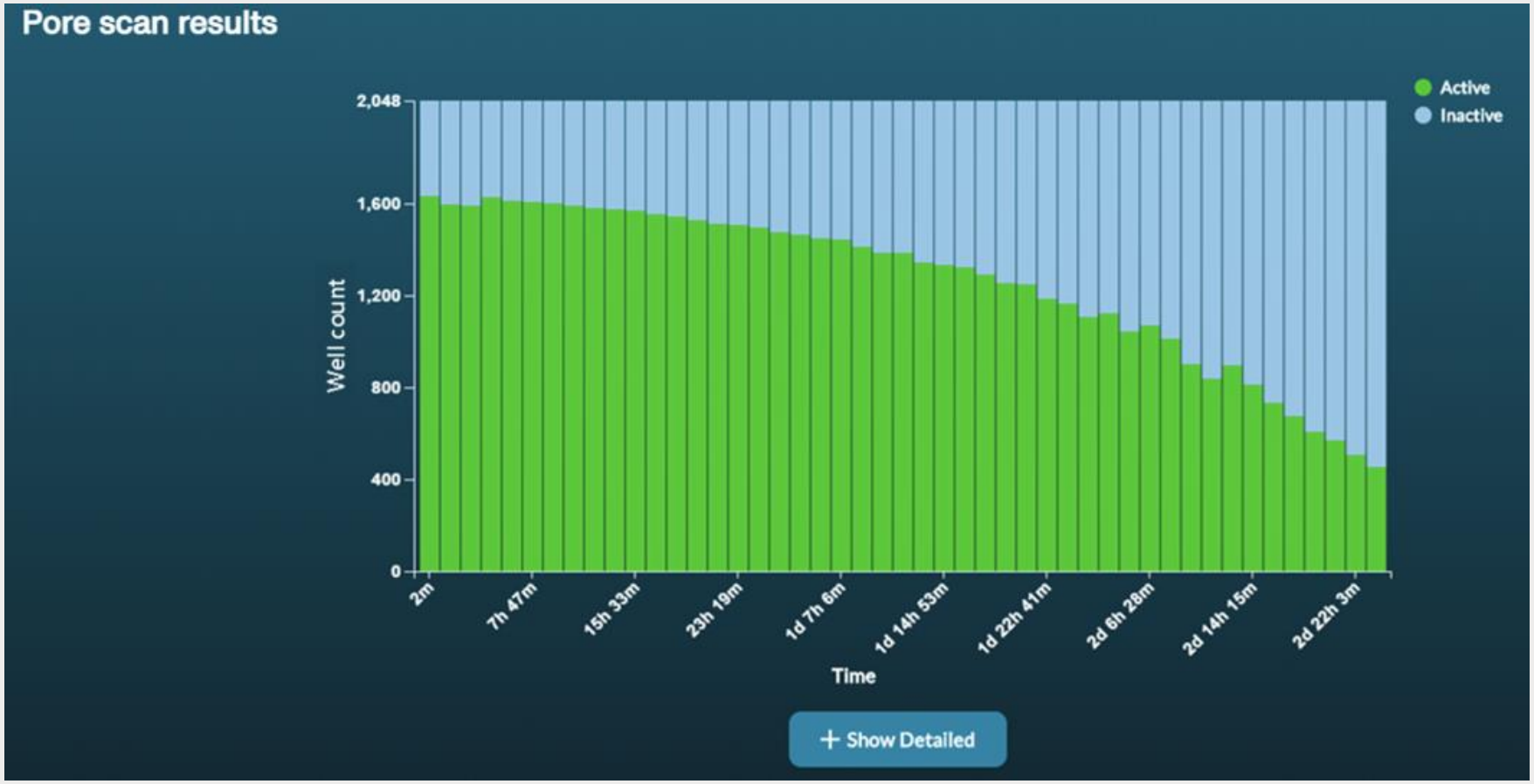
Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

Experiment Summary Information



- Minknow will basecall and demultiplex live
- Real time information on flow cell health and sequencing

Pore Scan



Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

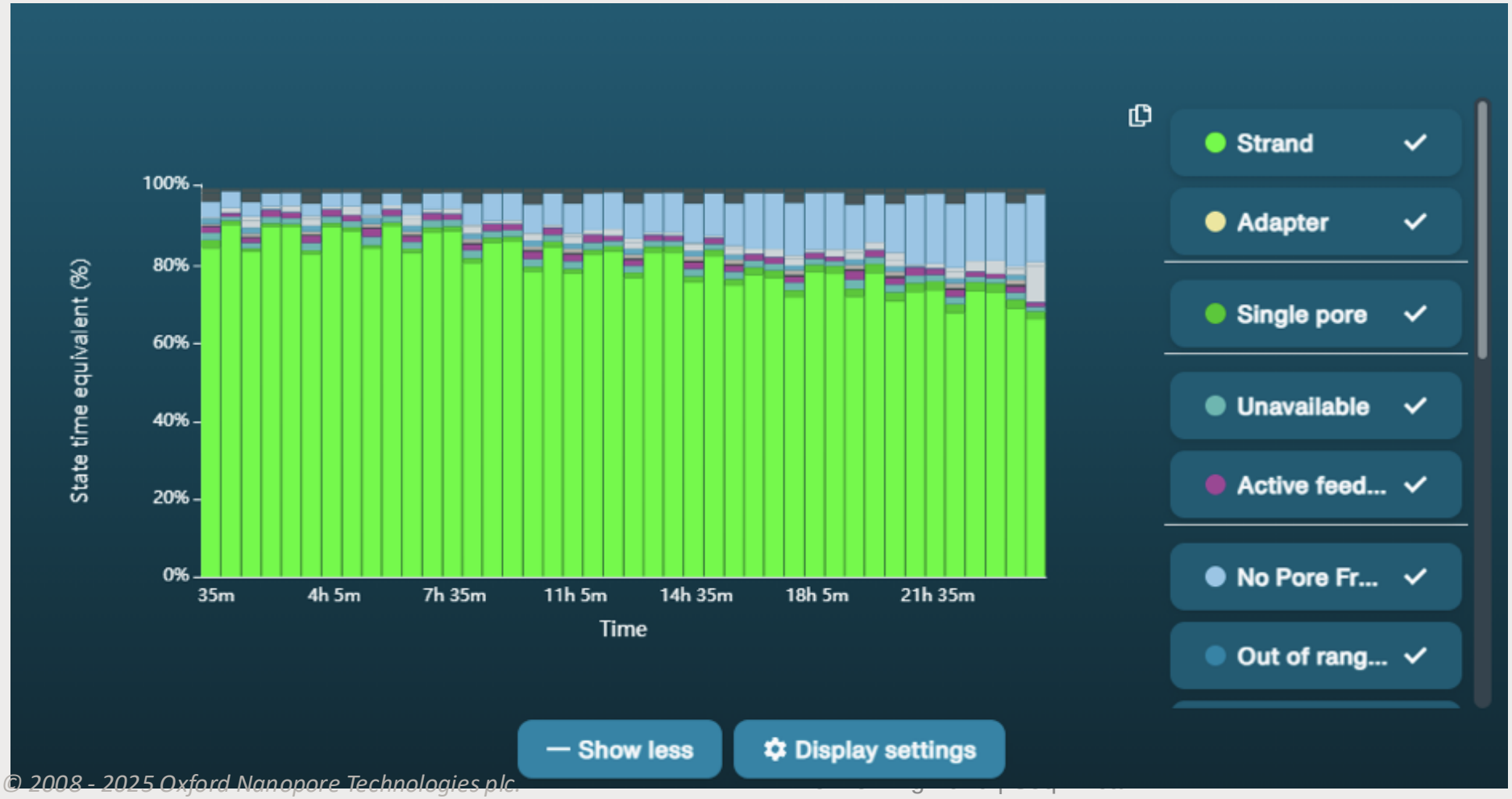
The Fleming Fund
Regional Grants

Pore Occupancy

Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

The Fleming Fund
Regional Grants

Good library

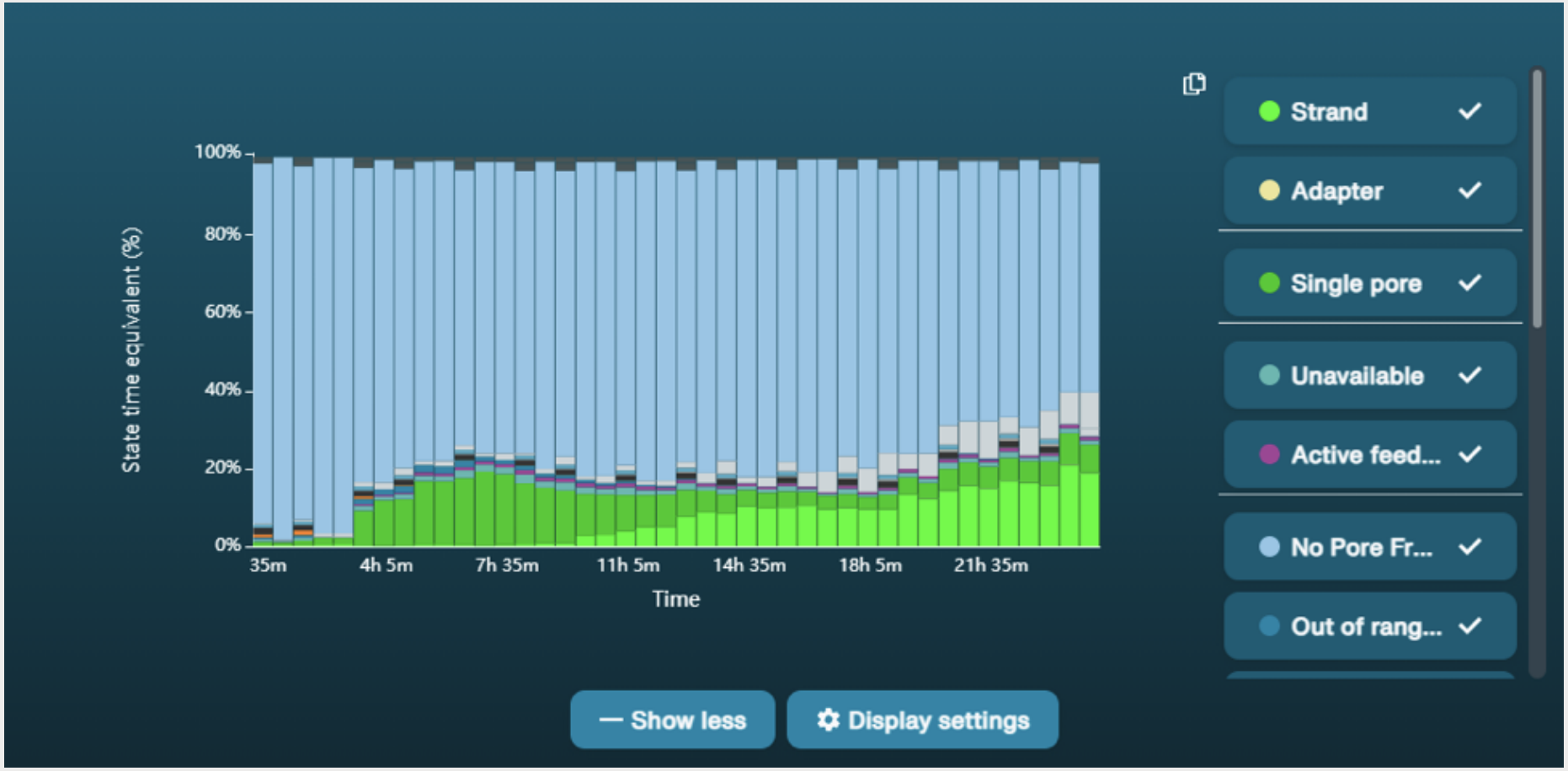


Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

Channel Blocking

Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

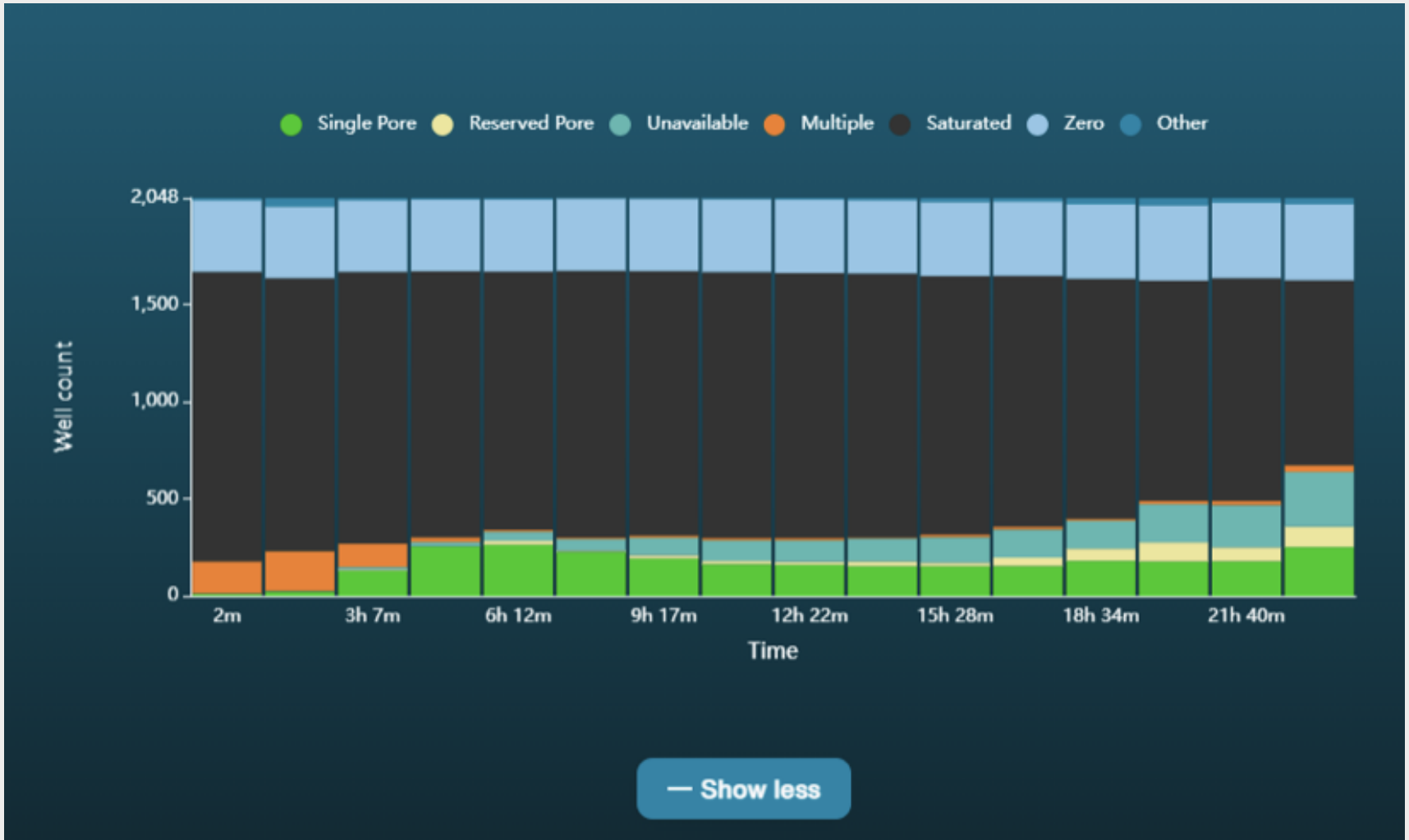
Osmotic Imbalance



Picture belongs to oxford nanopore (https://store.nanoporetech.com/eu/rapid-barcoding-kit-1.html)

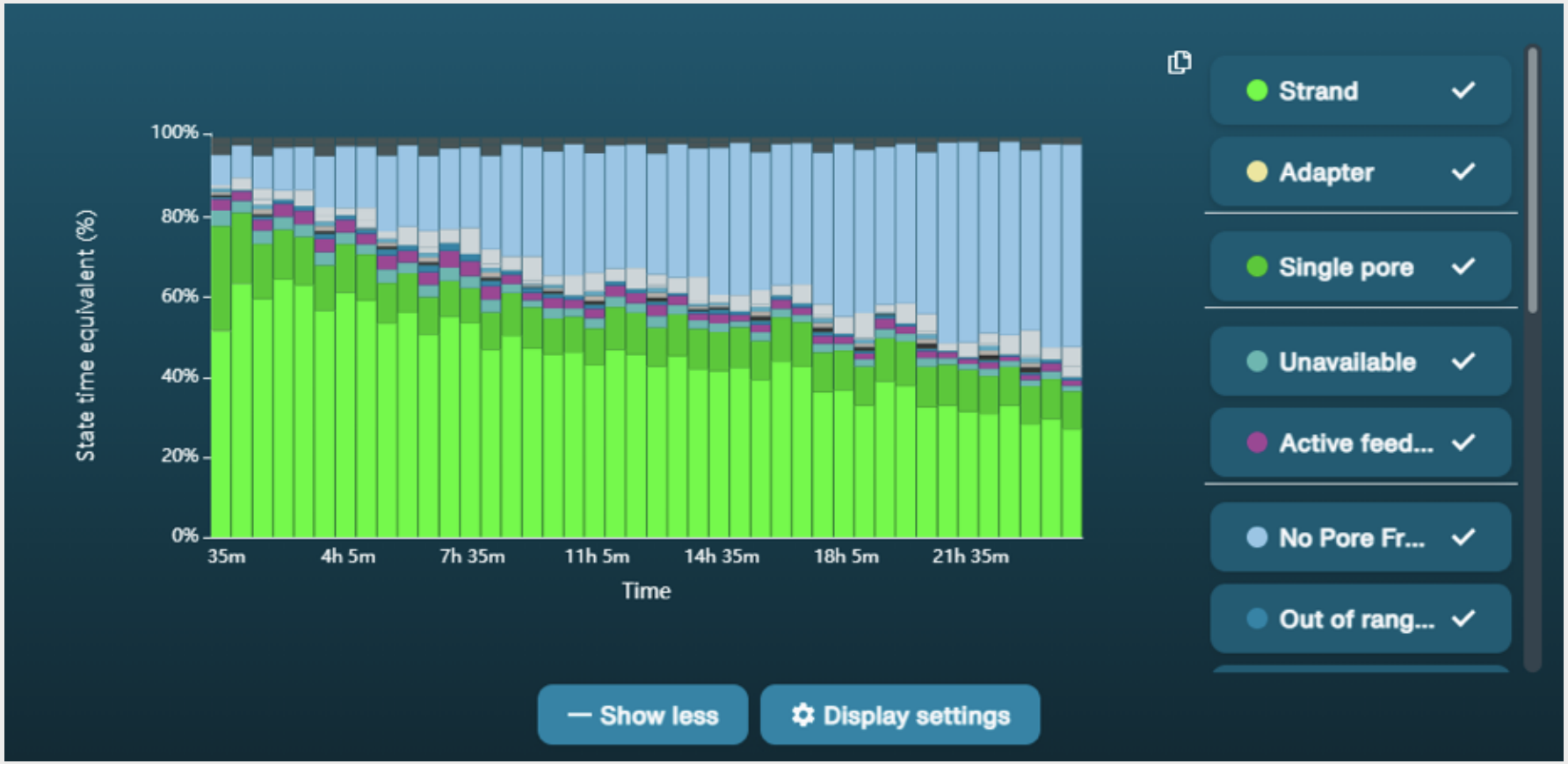
The Fleming Fund
Regional Grants

Osmotic Imbalance – channel scan



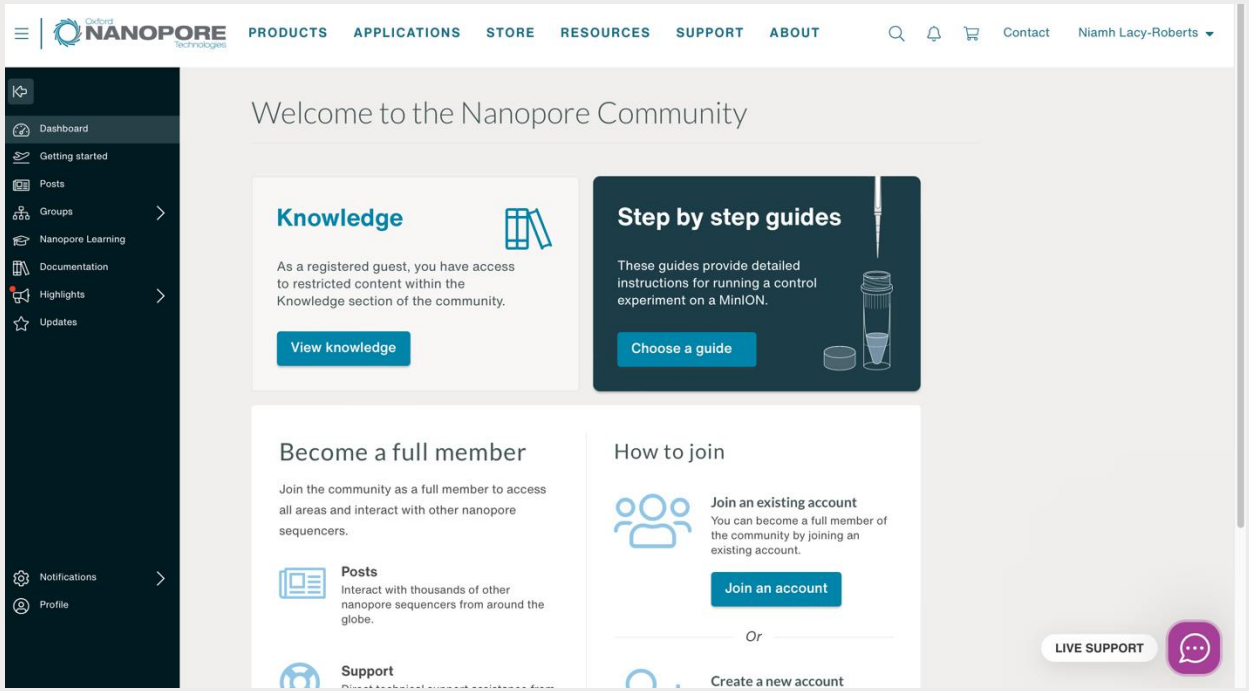
The Fleming Fund
Regional Grants

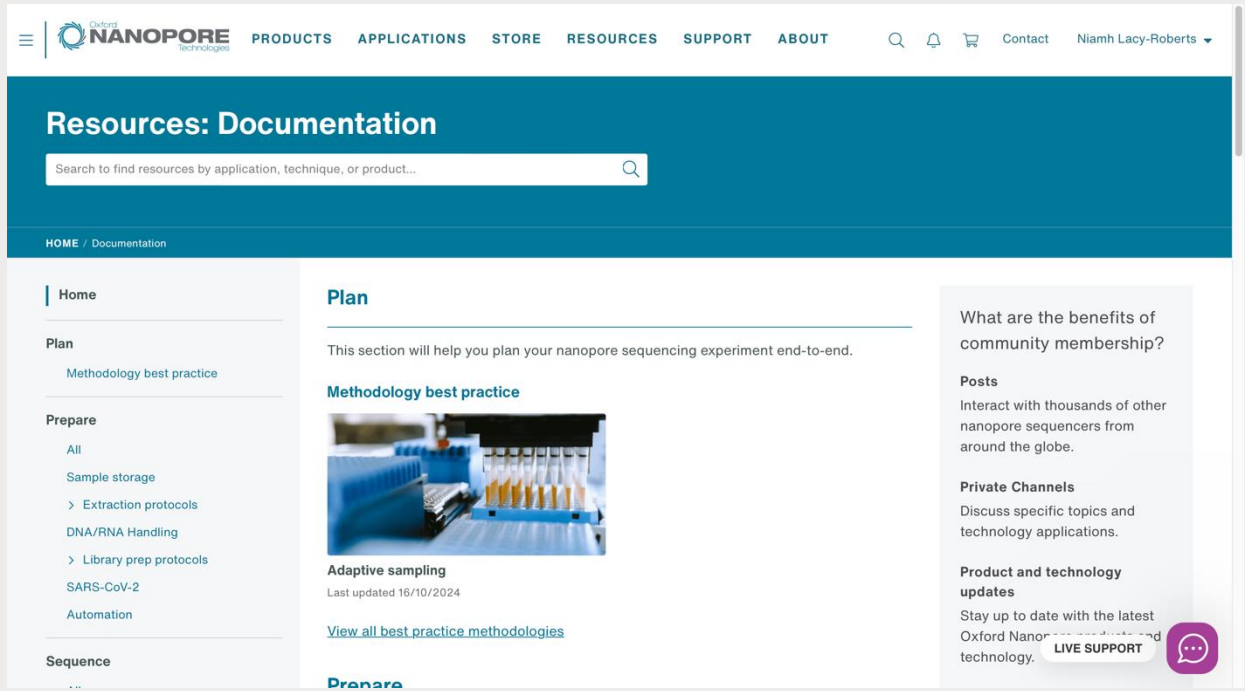
Low Pore Occupancy



The Fleming Fund
Regional Grants

Use ONT website and community to troubleshoot





Let’s look at some QC reports

- Report 1
- Report 2
- Report 3

The Fleming Fund
Regional Grants

Trimming and filtering

- We can employ bioinformatic tools to trim and filter our sequences:
  - **Trimming** focuses on removing unwanted sequences such as adapters, barcodes, or low-quality bases from the ends of reads.
  - **Filtering** focuses on removing entire reads based on predefined criteria, such as low Q-scores, overly short reads, or other contaminants.
- Some examples of command-line tools:
  - Nanoflit
  - Flitlong
  - Porechop (no longer maintained)
  - Fastplong
  - SeqKit
  - NanoPack

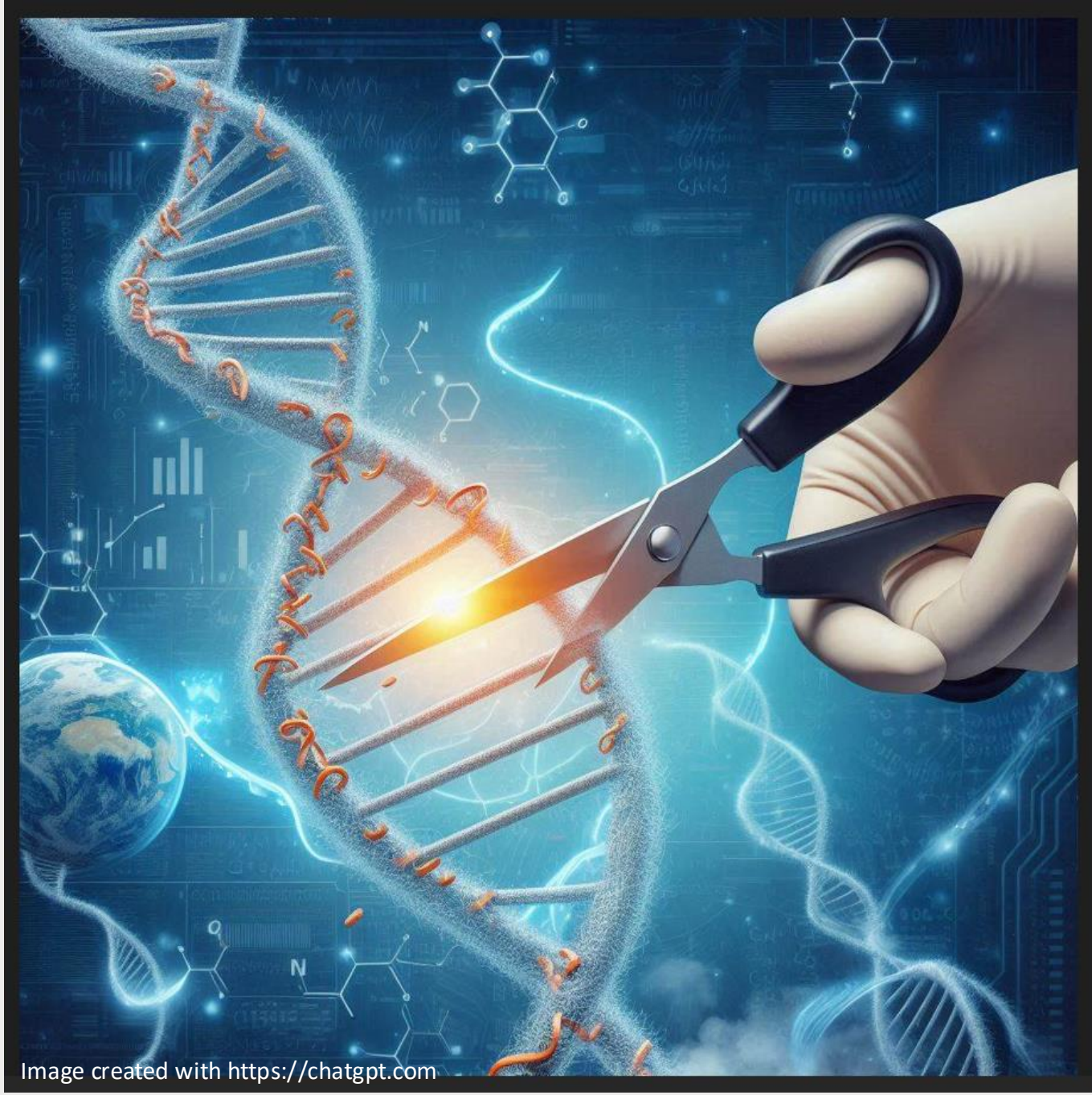


Image created with https://chatgpt.com

Flitlong

- Filters reads based on length and quality
  - Can prioritize specific reads.
  - Can downsample to desired coverage.

- Here is an example of running it on the command line:

```
filtlong --min_length 1000 --keep_percent 95 input.fastq.gz | gzip > long.fastq.gz
```

- This will remove any reads shorter than 1 kbp and also exclude the worst 5% of reads.

- Filtlong considers shorter reads ‘bad’ and longer reads ‘good’ so more aggressive filtering will leave you with few reads on the short end of the spectrum. For most of the genome this is probably a good thing, but it can be disastrous for small plasmids.

- For example, if you have a big read set that you’ve aggressively filtered with Filtlong, you might be left with no reads smaller than 10 kbp. If that genome has a small plasmid 4 kbp in size, it will now be gone from the read set!

Visualization and QC Reporting Tools for ONT Data

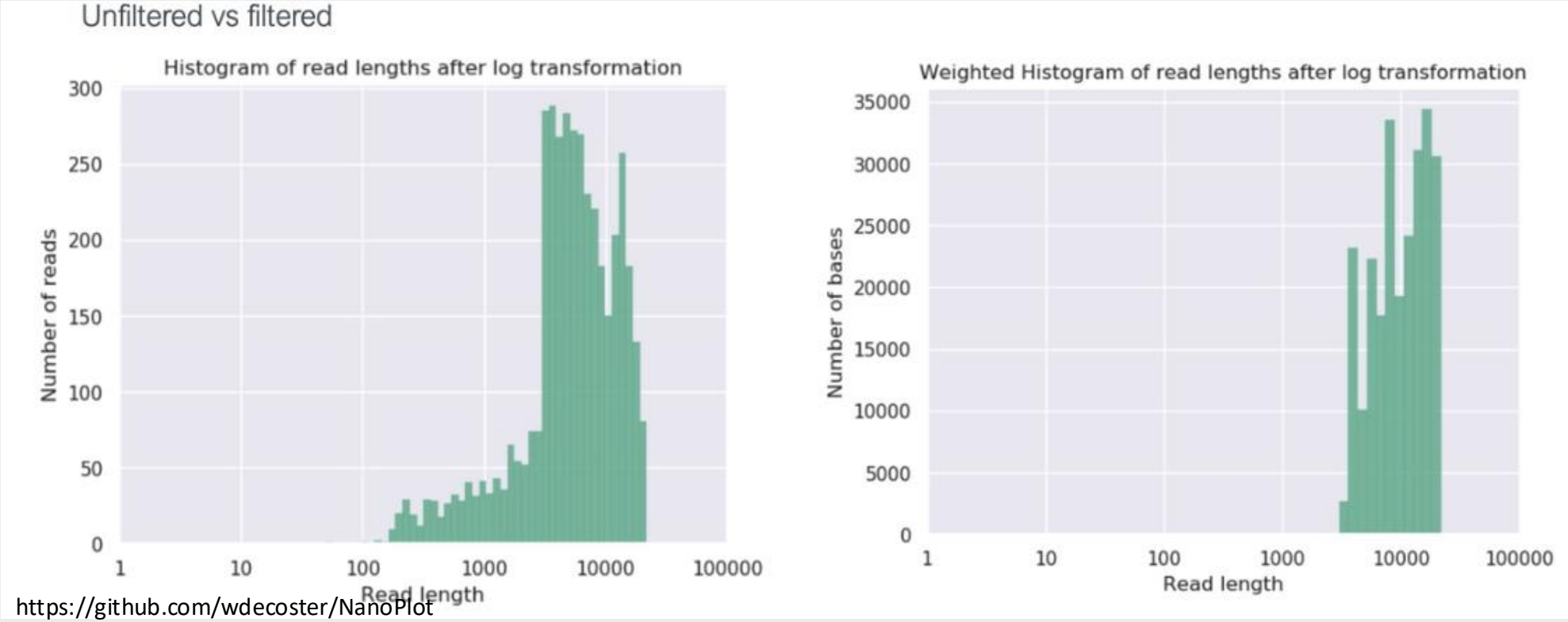
- After trimming and filtering ONT sequencing data, it is essential to visualize and report the quality of the cleaned reads to ensure the data is suitable for downstream analyses like genome assembly or AMR detection.

- Some examples of command line tools:
  - NanoStat (no longer maintained), superseded by CRAMINO
  - NanoPlot
  - pycoQC (no longer maintained)
  - nanoQC
  - nanoq

The Fleming Fund
Regional Grants

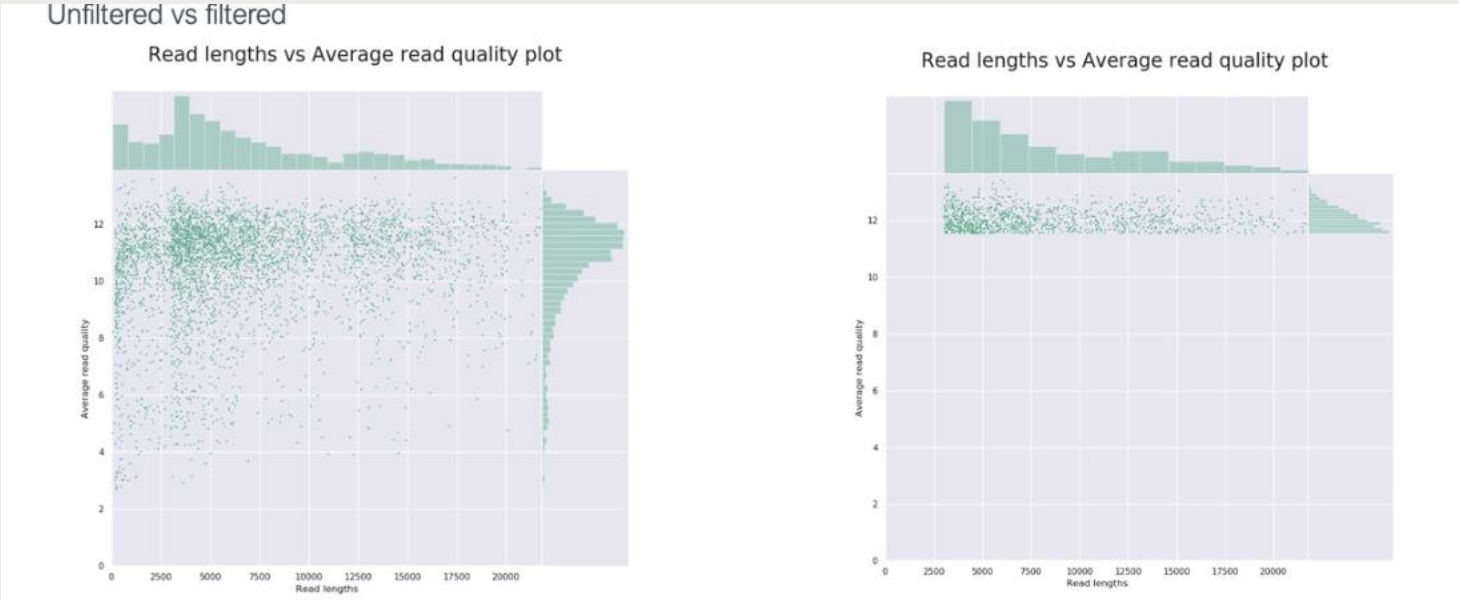
Nanoplot

- Generates and plots metrics from FASTQ files
- Outputs statistical summery, plots and html summary file



https://github.com/wdecoster/NanoPlot

Nanoplot



https://github.com/wdecoster/NanoPlot

The Fleming Fund
Regional Grants

NanoPlot

Summary statistics: unfiltered vs filtered

```
General summary:
Mean read length:                6,891.0
Mean read quality:                  10.5
Median read length:              5,400.0
Median read quality:                11.0
Number of reads:                 4,100.0
Read length N50:                10,208.0
Total bases:               28,253,135.0
Number, percentage and megabases of reads above quality cutoffs
>Q5:      4010 (97.8%) 27.9Mb
>Q7:      3834 (93.5%) 27.1Mb
>Q10:     3014 (73.5%) 21.9Mb
>Q12:     640 (15.6%) 5.0Mb
>Q15:     0 (0.0%) 0.0Mb
Top 5 highest mean basecall quality scores and their read lengths
1:        13.9 (403)
2:        13.6 (17389)
3:        13.6 (13346)
4:        13.6 (2028)
5:        13.5 (1068)
Top 5 longest reads and their mean basecall quality score
1:        21834 (9.3)
2:        21795 (11.9)
3:        21794 (11.9)
4:        21773 (12.5)
5:        21674 (11.9)
```

Unfiltered data

```
General summary:
Mean read length:                8,677.3
Mean read quality:                  12.1
Median read length:              7,124.0
Median read quality:                12.0
Number of reads:                 1,154.0
Read length N50:                11,623.0
Total bases:               10,013,547.0
Number, percentage and megabases of reads above quality cutoffs
>Q5:      1154 (100.0%) 10.0Mb
>Q7:      1154 (100.0%) 10.0Mb
>Q10:     1154 (100.0%) 10.0Mb
>Q12:     565 (49.0%) 4.9Mb
>Q15:     0 (0.0%) 0.0Mb
Top 5 highest mean basecall quality scores and their read lengths
1:        13.6 (17389)
2:        13.6 (13346)
3:        13.4 (5936)
4:        13.3 (6092)
5:        13.3 (5501)
Top 5 longest reads and their mean basecall quality score
1:        21795 (11.9)
2:        21794 (11.9)
3:        21773 (12.5)
4:        21674 (11.9)
5:        21591 (12.2)
```

Filtered data

https://github.com/wdecoster/NanoPlot

The Fleming Fund
Regional Grants

Let’s take a break ☺

The Fleming Fund
Regional Grants

Overview of assembly approaches

The Fleming Fund
Regional Grants

From fastq to fasta

```
@SRR1928200.1 HWI-ST1106:418:D1H56ACXX:2:1207:10978:124033/1
TGCCGAGTGATATCGCTGACGTCATCCTTGAGGGTGAAGTTCAGGTCGTCGAGCAACTCGGCAACGAAACTCAAATCCATATCCAGATCCCTTCCATTCG
+
@@CFFDFBFFHHHJJJIJIJIGGIIJJJGIIHIFBGHIHHHJJIIFGHIGJJJHHHHFFFCCDDDDDDDDDCCCC;:@CDDDDEDDCDDDCDDDC>CDD>
```

```
>ENA|LR822054|LR822054.1 Citrobacter werkmanii isolate BB1479 genome assembly, plasmid: pCW-CTX-M-15A_
CGTCAGCTTTCCAGTCGACGGCTGATTGAAGTCGGGAATAGCGTCCTTGAAAAGAAGAAC
TTCATTCGAGTTCATCGTGTGGATCCCCCAGTTTTATTGTTATTTTCCGGGTATCTTGGA
ATGCCCAGTCCGGGCGAATGTATCACGGTGATTTTTATTGATCATGAGAAATAGGGGTCA
TTTAGTCCCCATTTATCGGGTATTGGTTTTTATTTGTACTAAATCAATACGTTATTTCAG
AGATGAATCGGATAAATGTCGTTGACATCAAATTTTTGATCTGCTGCCAGTGTGGACAAA
AAATGAATACCGATCACCTATTTTTGAGATTTGTTACGTATGATTATGTTTTTATTTGAT
GTTTTCATTAGCACAGCAGATGTTGATAATTAAGTTCCTTTCCCCTTCCAATCCCACCGT
TATTCCCTTTGAACACCACCAGCTACCAGGCTAACCCCACCGACAGCCCTTCAGAGCTCA
CTTTTTTCCCTCTCAACCCCACCGGGGCAGGTCTTCAGAGCTTACCAGCTGCGGGTTTGC
GGGAGCGGGGATCTTTTTGGTTCTATTTGGTCTTAATCTGGATCGATCTGTTGATCTACC
```

The Fleming Fund
Regional Grants

Types of Assembly Tools

User-friendly platforms

- **Platforms** designed with **graphical interfaces** (accessible by a web browser) with **pipelines** already **developed**, making bioinformatics accessible to non-experts
- Ideal for **quick** analyses on small datasets or **limited computational resources**

BV-BRC Bacterial and Viral Bioinformatics Resource Center | Galaxy | KBase PREDICTIVE BIOLOGY

Command Line Tools

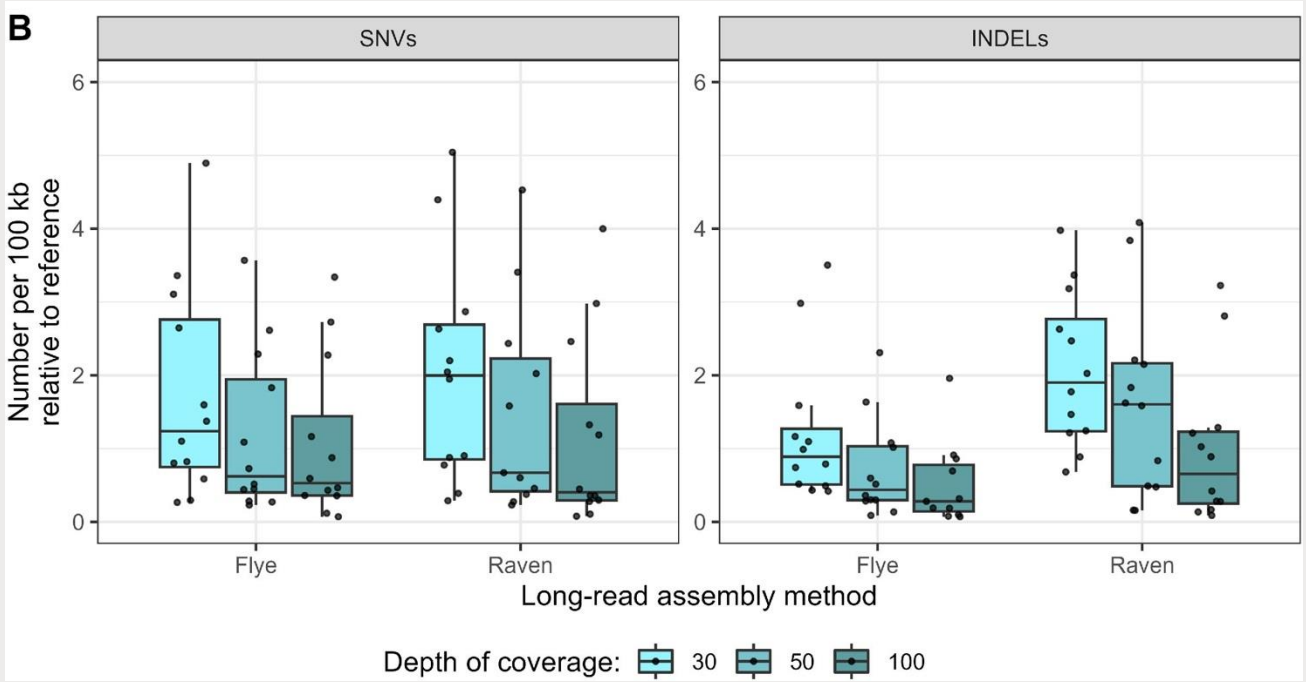
- Tools that are **executed** using text-based **commands** in a terminal
- **Require installation** and **configuration** on a local computer or server
- Recommended for **large-scale projects**, **up-to-date tools**, or **unique workflows**

Trycycler | Unicycler

The choice between both tools depends on expertise, project scale, and the need for customization (flexibility)

What depth of coverage is best for long-read-only (ONT) genome assembly?

100 × **<u>ONT</u>** coverage is the ideal for long-read-only assemblies → **<u>Assembly tool dependent</u>** (some tools tools show improved accuracy with higher read depth)



Lerminiaux et al. (2024). Can J Micr;70(5):178-189

What depth of coverage is best for hybrid (Illumina and ONT) genome assembly?

ONT read depth of **30 ×** is sufficient to achieve high-quality genome assemblies, provided that at least **50 ×** **Illumina** data is also available → **Assembly tool dependent**

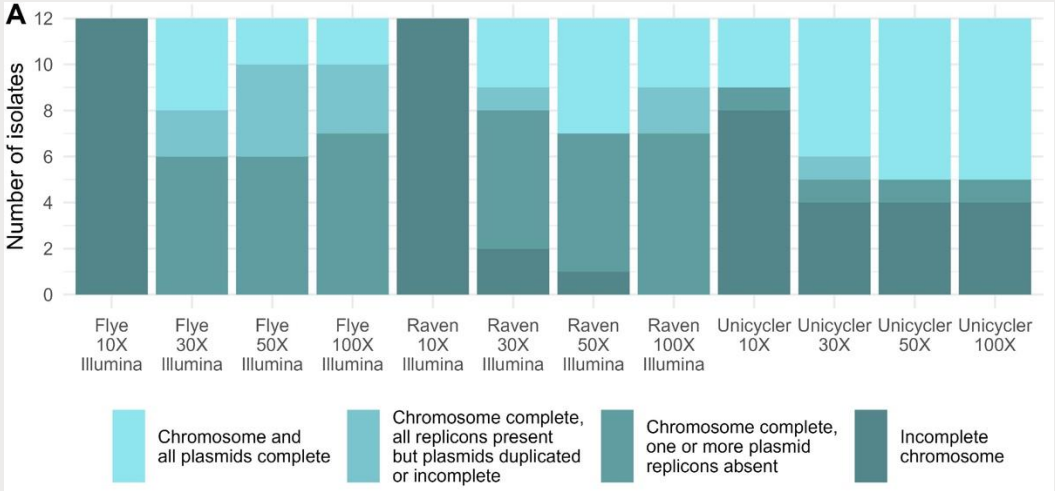


Fig. Hybrid assembly completeness relative to the reference genomes. (A) Completeness across all replicons

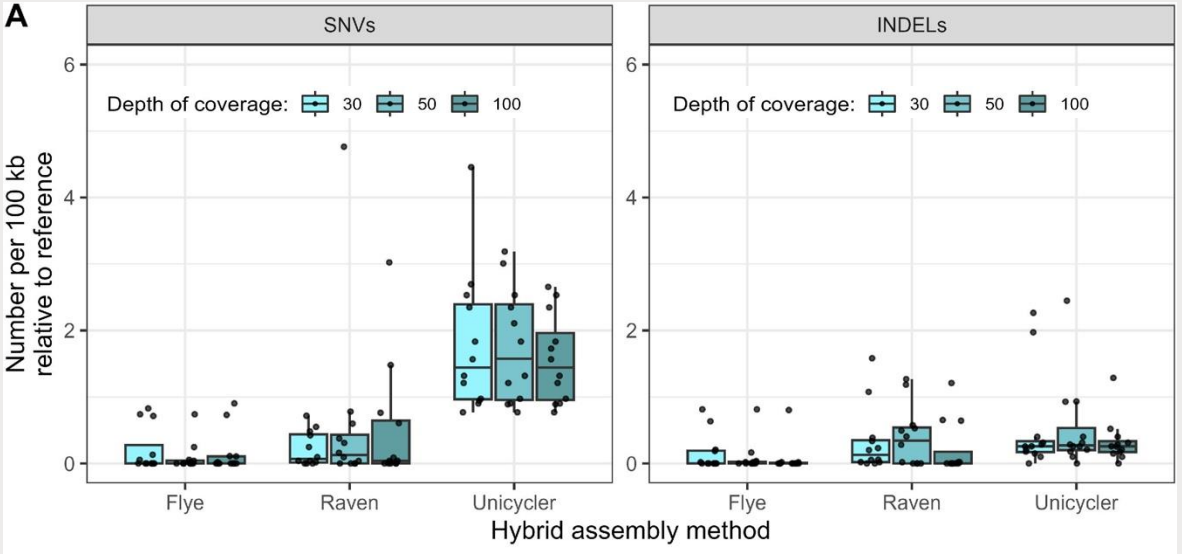


Fig. Comparison of single nucleotide variants (SNVs) and small insertions/deletions < 60 bases (INDELs) in (A) hybrid assembly methods at three different average read depth of coverage to the reference genomes. Lines in the center of boxplots represent the mean

Lerminiaux et al. (2024). Can J Micr;70(5):178-189

How close are we to using ONT data to produce Illumina-quality assemblies?

Polishing long-read-only assemblies with Illumina short-read data significantly reduces the number of single nucleotide variants (SNVs) and small insertions/deletions (INDELs) → **Illumina short-read data is still valuable for high accuracy genomes**

Table 2. Mean count of single nucleotide variants (SNVs) and small (<60 bp) insertions/deletions (INDELs) per 100 kb in hybrid and long-read-only assemblies from the RBKv14 dataset.

Assembler	Depth of coverage	Hybrid: SNVs per 100 kB	Long-read: SNVs per 100 kB	Hybrid: INDELs per 100 kB	Long-read: INDELs per 100 kB
Flye	10×	1.75	27.9	1.16	36.7
Flye	30×	0.21	1.74	0.16	1.23
Flye	50×	0.09	1.19	0.09	0.72
Flye	100×	0.09	1.11	0.08	0.51
Raven	10×	10.7	32.8	4.28	25.6
Raven	30×	0.22	2.03	0.33	2.08
Raven	50×	0.59	1.91	0.40	1.62
Raven	100×	0.50	1.16	0.21	0.99

Lerminiaux et al. (2024). Can J Micr;70(5):178-189

A note on Polishing

- Long-read polishing is the process of correcting errors in long-read sequences to improve assembly accuracy.
- Addresses issues like indels, mismatches, and sequencing artifacts.
- Can significantly improve sequence accuracy, and since long reads can span most repeats, long-read polishing can make repeats just as accurate as non-repetitive sequences.

Types of Polishing:

- **Self-Polishing**:
  - Uses the long reads themselves to correct errors.
  - Tools: Racon, Medaka.
- **Short-Read Polishing**:
  - Incorporates accurate short reads (e.g., Illumina) for error correction.
  - Tools: Pilon, POLCA.

Assembly Tools for Short-read Data

Tools like **SPAdes** and **SKESA** are leaders for short-read **bacterial** genome assembly

Used sometimes after assembly to **correct base-level errors** and improve accuracy

Assembly Tools

- **SPAdes** (Unicycler and Shovill are tools based on SPAdes)
- **SKESA**
- Velvet
- Abyss
- SOAPdenovo2
- MEGAHIT

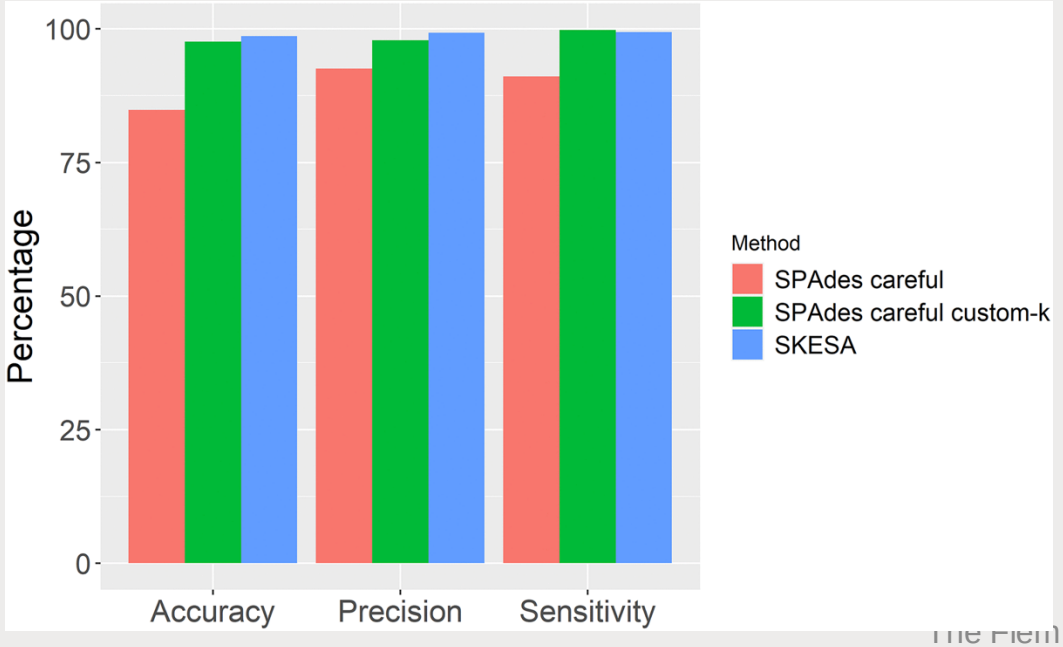
Polishing Tools

- **NextPolish** (supports short- and long-read data)
- **Polypolish**
- **Pypolca** (Python-based implementation of POLCA)
- Pilon
- HyPo

Prjibelski et al. (2020). Curr Protoc Bioinformatics. 70(1):e102; Souvorov et al. (2018). Genome Biol. 19(1):153

Assembly Tools for Short-read Data

Optimization of assembly tools – even the same tool (e.g., SPAdes with --careful and k-mer tuning) – can significantly **improve the accuracy of the assembly** and consequently of all downstream analysis



Overall, **SKESA** (inclusion of --allow_snps) and **SPAdes careful custom-k** (-k 33,55,77,99,121) performed the best concerning both accuracy, precision, and sensitivity

Fig. Performance metrics for the three methods. Accuracy, precision, and sensitivity of *Staphylococcus spa* type determination are shown for SPAdes careful, SPAdes careful custom-k, and SKESA.

Mollerup et al. (2022). Microbiol Spectr. 10(6):e0218922

The Fleming Fund
Regional Grants

Assembly Tools for Long-read Data

Long-read data **assemblers**, are frequently used with **polishing tools**, and finishing utilities (e.g., Circlator - “circularize” bacterial genomes from draft assemblies)

Assembly Tools

- **Flye** (popular for bacteria)
- **Canu**
- Dragonflye
- Raven
- SMARTdenovo
- Miniasm
- **Trycycler** (consensus tool)

Polishing Tools

- **Medaka** (uses a Machine Learning model trained on ONT data)
- **NextPolish**
- Racon (consensus polishing)
- FMLRC2 (long-read error correction using high-quality short-read data)

Kolmogorov et al. (2019). Nat Biotechnol. 37(5):540-546; Koren et al. (2017). Genome Res. 27(5):722-736; Wick et al. (2021). Genome Biol. 22(1):266; Luan et al. (2024). BMC genomics. 25(1):679

Polishing Tools for Long-read Data

Benchmarking short and long read polishing tools for nanopore assemblies: achieving near-perfect genomes for outbreak isolates

BMC

Tu Luan[1†], Seth Commichaux[2*†], Maria Hoffmann[3], Victor Jayeola[3], Jae Hee Jang[3], Mihai Pop[1], Hugh Rand[3] and Yan Luo[3]

Among the tools evaluated, **Medaka** was identified as a **more accurate and efficient** long-read polisher compared to **Racon**

Be **cautious** of any major changes made by **polishers**, as their goal is to correct large-scale errors, but some tools may **introduce new errors** while fixing others

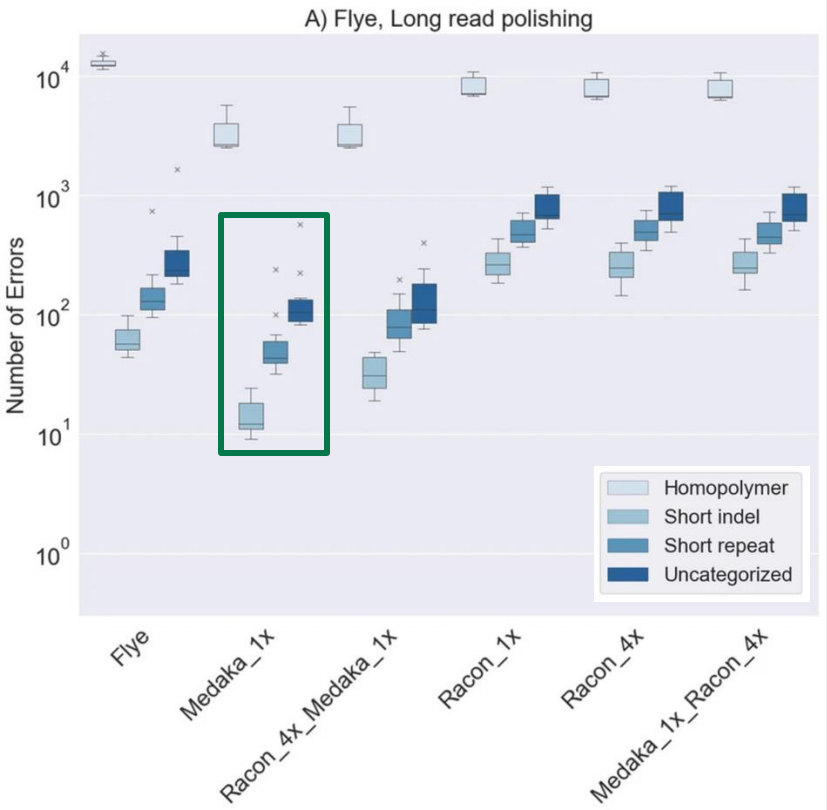


Fig. The genomic features associated with the errors in the long-read polished Flye

Luan et al. (2024). BMC genomics. 25(1):679

Assembly Tools for Long-read Data

Ryan Wick's bioinformatics blog About

ONT-only accuracy with R10.4.1

5 May 2023 • Ryan Wick

Assembly tools used in 5 different bacterial species were ranked from best to worst: **Trycycler** (more time consuming), Canu and Flye

This table shows the error count (top) and qscore (bottom) for each assembly:

Genome	Flye	Canu	Trycycler	Trycycler +Medaka	Main cause of errors
Salmonella enterica	52 Q49.7	35 Q51.4	13 Q55.7	9 Q57.3	homopolymers
Vibrio parahaemolyticus	149 Q45.4	91 Q47.5	52 Q50.0	81 Q48.0	unknown methylation?
Escherichia coli	332 Q42.0	223 Q43.7	171 Q44.8	171 Q44.8	M1.EcoMI methylation
Campylobacter jejuni	1004 Q32.5	1113 Q32.0	508 Q35.4	578 Q34.9	CtsM methylation
Listeria monocytogenes	12 Q53.9	7 Q56.2	0 Q∞	0 Q∞	n/a

<u>Flye</u> frequently duplicated small plasmids (between 4.1 and 9.3 kb) or was missing small plasmid replicons altogether (between 1.4 and 5.2 kb)

Lerminiaux et al. (2024). Can J Micr;70(5):178-189; https://rrwick.github.io/2023/05/05/ont-only-accuracy-with-r10.4.1.html

The Fleming Fund
Regional Grants

Let’s take a break ☺

The Fleming Fund
Regional Grants

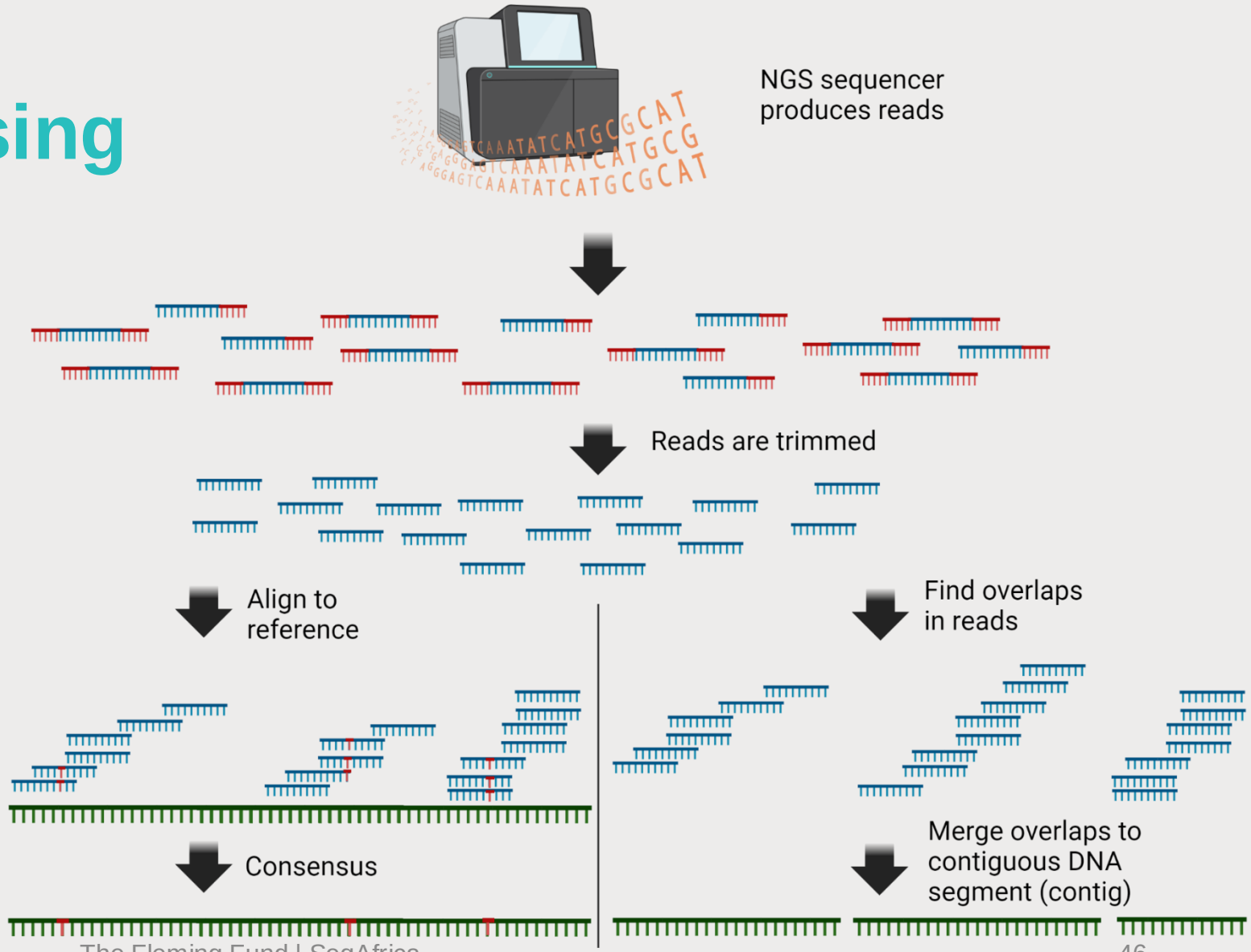
Assembly theory

The Fleming Fund
Regional Grants

Created with BioRender.com

NGS data processing

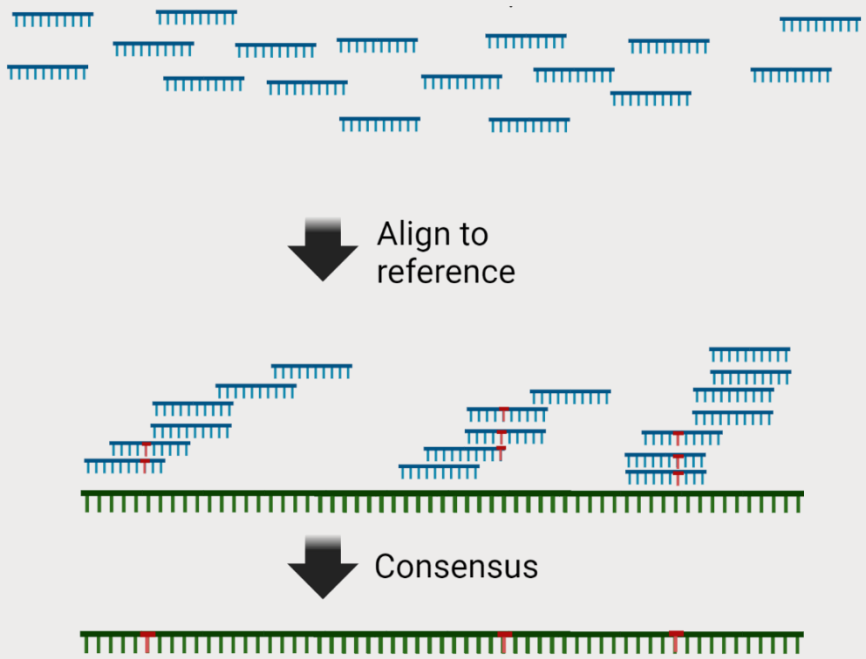
- Raw reads are produced by the sequencing platform
- Trimming - poor sequences are removed from the raw reads, leaving high confidence trimmed reads
- QC – visualize metrics
- Assembly - we can then apply two standard approaches:
  - Mapping to reference
  - De novo assembly



Mapping to reference

- Reads are aligned to a reference genome using alignment tools such as
  - Burris Wheelers Aligner (short reads) GitHub - lh3/bwa: Burrow-Wheeler Aligner for short-read alignment (see minimap2 for long-read alignment).
  - Minimap2 (long reads) GitHub - lh3/minimap2: A versatile pairwise aligner for genomic and spliced nucleotide sequences.

- Depend on availability of high quality "closely" related strain.



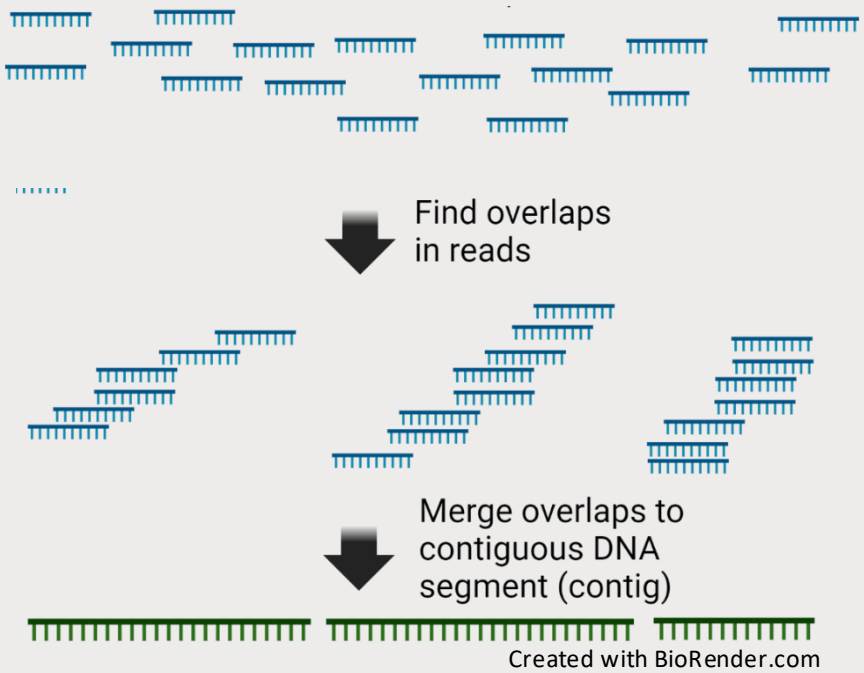
Created with BioRender.com

De novo assembly

- Reference independent assembly of reads, for an "Unbiased" reconstruction of the genome.

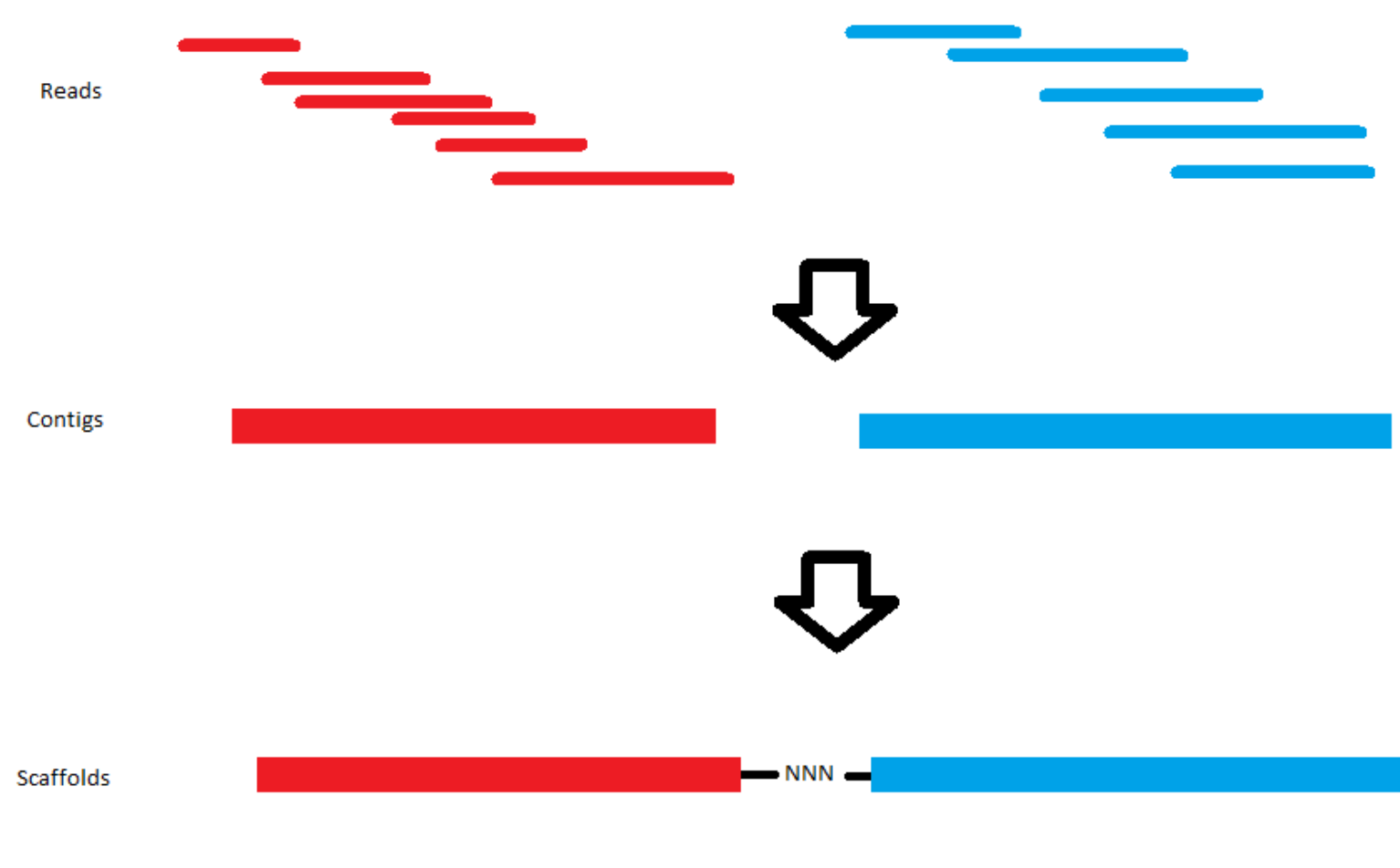
- For short read technologies, repeated segments are an issue.

- For Long-read technologies these issues are less pronounced.
  - You will typically have reads that are longer than the longest repeat in the genome.
  - E.g. if your genome's longest repeat is 6 kbp and your reads have an N50 length of 10 kbp, assembly should proceed well. But keep in mind that some bacterial genomes do have very long repeats (e.g. 100 kbp) and complete assembly in such cases will require ultra-long reads.



De novo assembly

- Many programs can do assembly, they differentiate by how precisely they can construct the assembly, how fast and how computationally heavy their workload
  - SPAdes
  - Flye
  - Canu
  - Raven
  - Trycycler
  - Unicycler
- The assembly should not contain unknown bases (N), e.g. we usually work with the contigs, and not the scaffolds



Flye

- Flye is an overall strong performer. Its main downside is that you'll need a bit more computational resources than you would for other assemblers. 32 GB of RAM and 1 hour should be sufficient for most read sets.
- You can run Flye like this

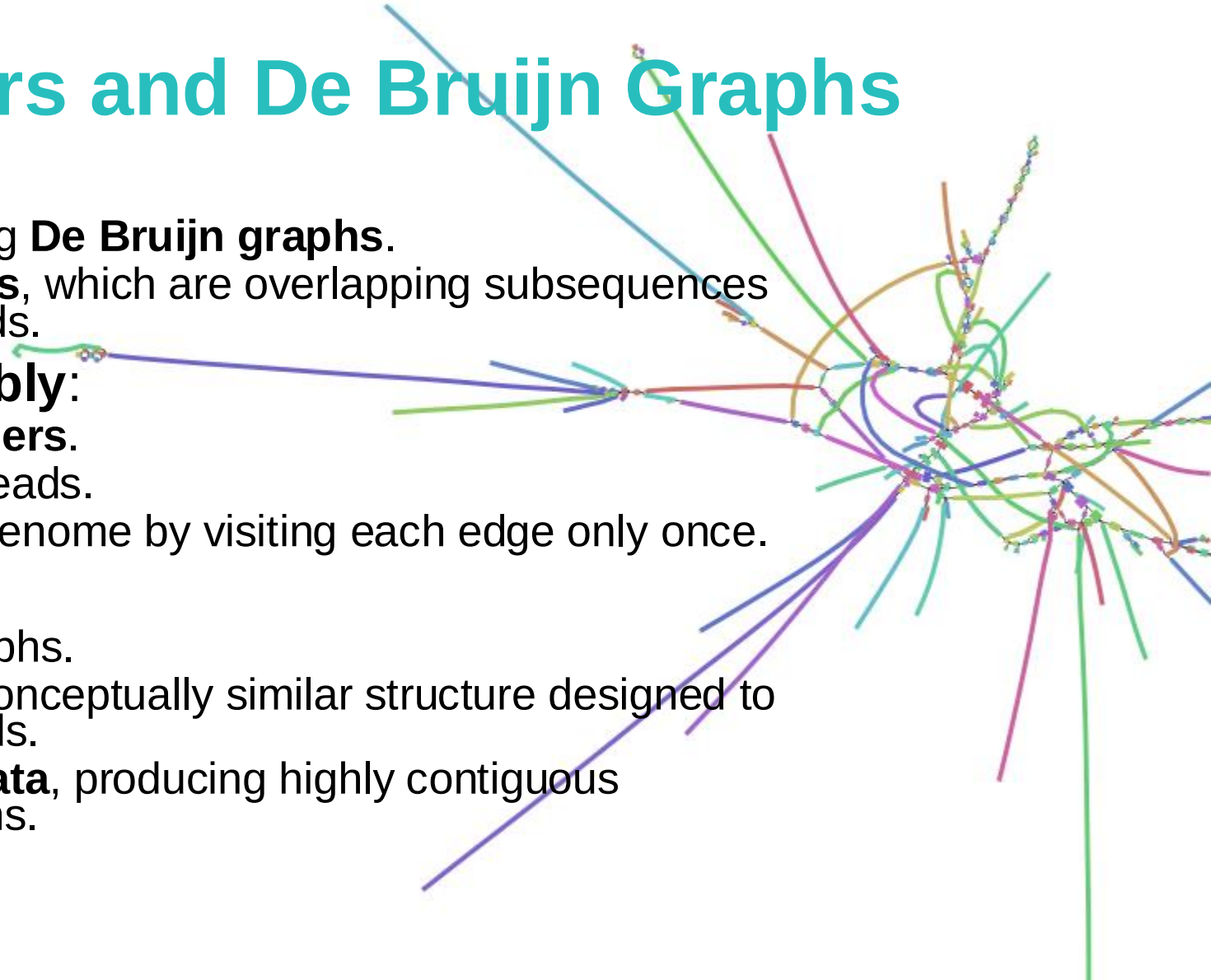
```
flye -o flye_assembly --plasmids --threads 16 --nano-raw long.fastq.gz
```

- Flye's --plasmids option enables a nice feature which tries to recover small plasmids in the genome.
- However, it has a nasty habit of sometimes doubling small plasmids in a single contig. E.g. if your genome has a 4 kbp plasmid, Flye might create an 8 kbp contig with two whole copies of the plasmid sequence. Something to keep an eye out for!

The Fleming Fund
Regional Grants

Nanopore Assemblers and De Bruijn Graphs

- **SPAdes**:
  - Constructs **de novo assemblies** using **De Bruijn graphs**.
  - De Bruijn graphs are built from **K-mers**, which are overlapping subsequences of length K derived from the input reads.
- **Steps in De Bruijn Graph Assembly**:
  - Split sequences into overlapping **K-mers**.
  - Connect identical K-mers across all reads.
  - Traverse the graph, assembling the genome by visiting each edge only once.
- **Flye**:
  - Does not use traditional De Bruijn graphs.
  - Instead, employs a **repeat graph**, a conceptually similar structure designed to resolve repeats in high-error long reads.
  - Optimized for **long-read nanopore data**, producing highly contiguous assemblies even with repetitive regions.



K-mers

- Worked example of a 4-mer:

Reads		Resulting 4-mers
ATGCGTGAC	→	

K-mers

- Worked example of a 4-mer:
  - First 4-mer consists of the 4 first nucleotides.

Reads		Resulting 4-mers
ATGCGTGAC	→	ATGC

K-mers

- Worked example of a 4-mer:
  - First 4-mer consists of the 4 first nucleotides.
  - Slide 1 nucleotide down the sequence.

Reads		Resulting 4-mers
ATGCGTGAC	→	ATGC
		TGCG

K-mers

- Worked example of a 4-mer:
  - First 4-mer consists of the 4 first nucleotides.
  - Slide 1 nucleotide down the sequence.
  - Repeat for rest of sequence.

Reads		Resulting 4-mers
ATGCGTGAC	→	ATGC
		TGCG
		GCGT

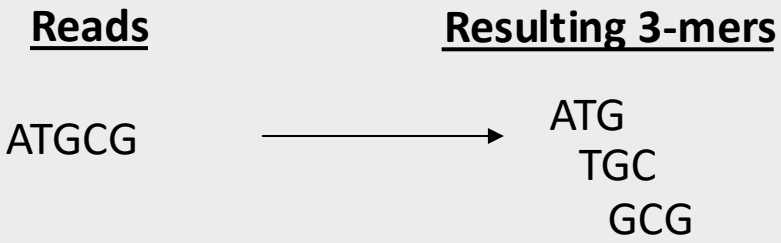
K-mers

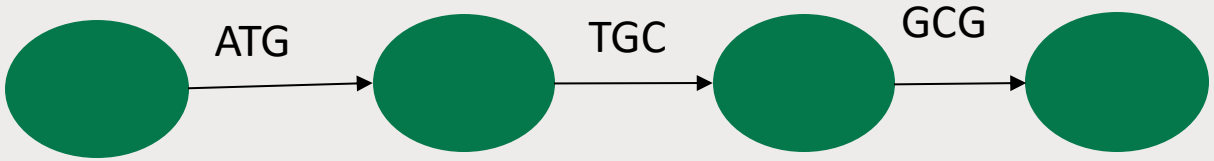
- Worked example of a 4-mer:
  - First 4-mer consists of the 4 first nucleotides.
  - Slide 1 nucleotide down the sequence.
  - Repeat for rest of sequence.

Reads	Resulting 4-mers
ATGCGTGAC	ATGC
	TGCG
	GCGT
	CGTG
	GTGA
	TGAC

De novo assembly using de Bruijn graphs

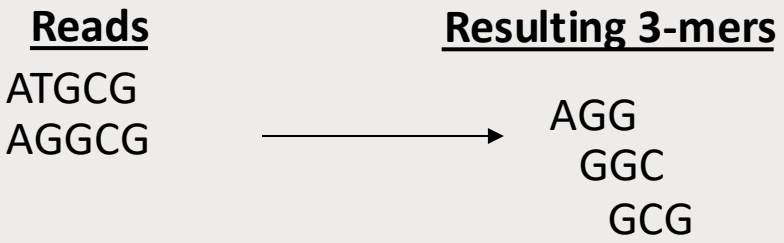
- Example:
  - Reads of 5 bp is split into K-mers of length 3 (3-mers)
  - De Brujn graph constructed with 3-mers as edges

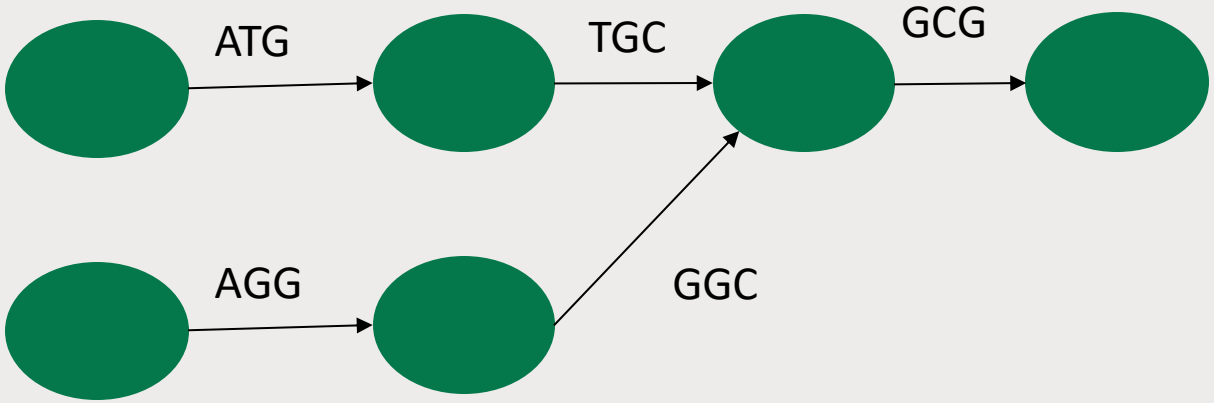




De novo assembly using de Bruijn graphs

- Example:
  - Reads of 5 bp is split into K-mers of length 3 (3-mers)
  - De Brujn graph constructed with 3-mers as edges
  - Process repeated for new read





De novo assembly using de Bruijn graphs

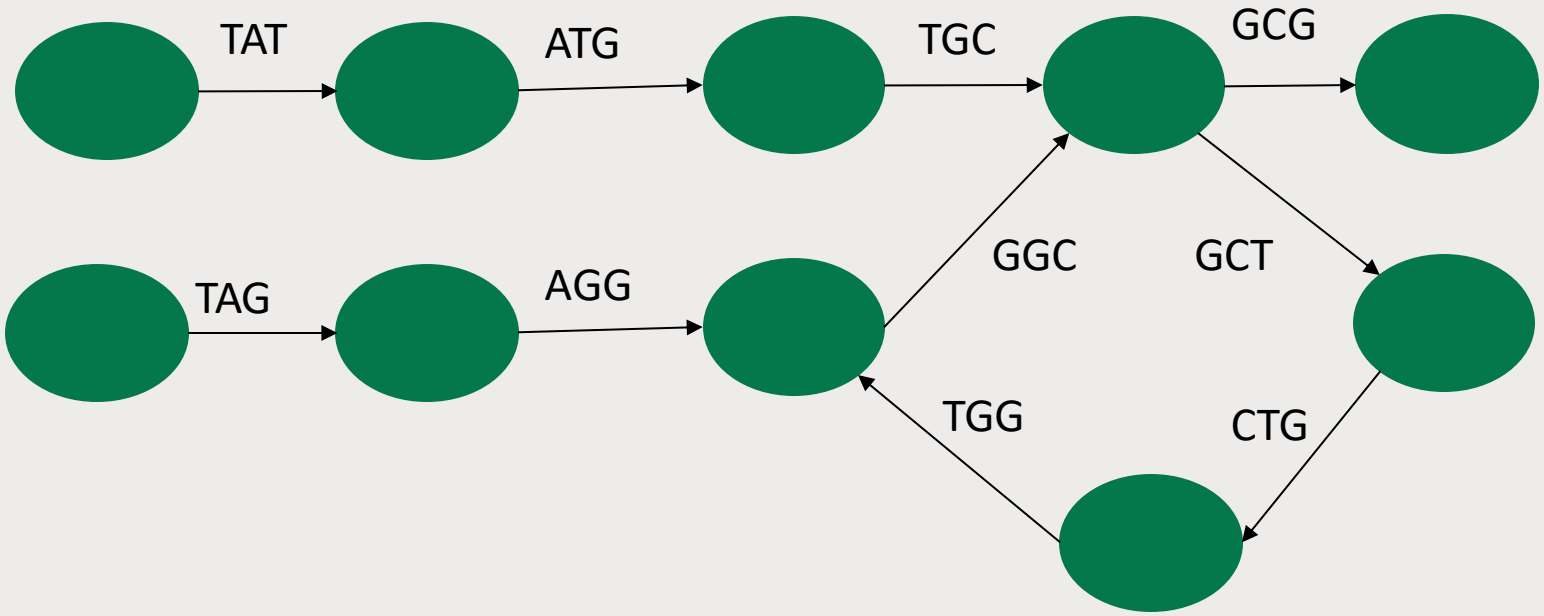
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG



>contig1

De novo assembly using de Bruijn graphs

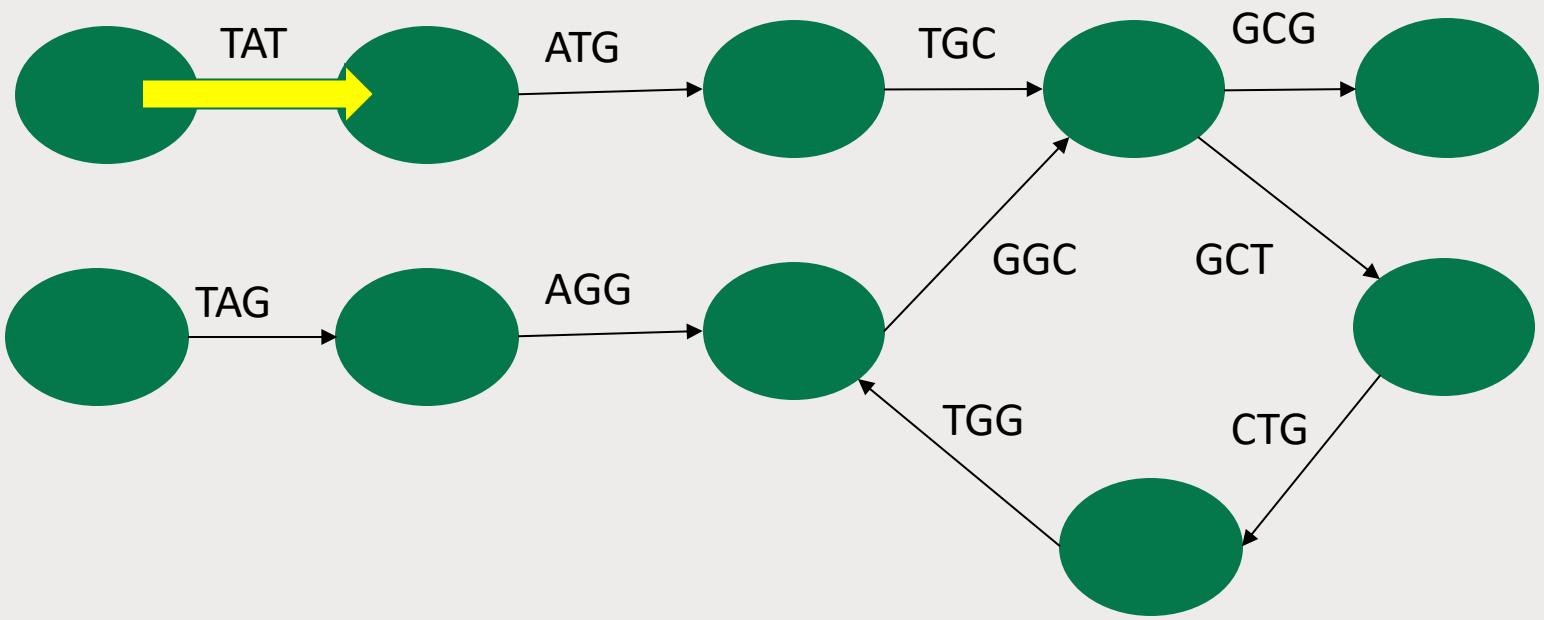
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TAT
```

De novo assembly using de Bruijn graphs

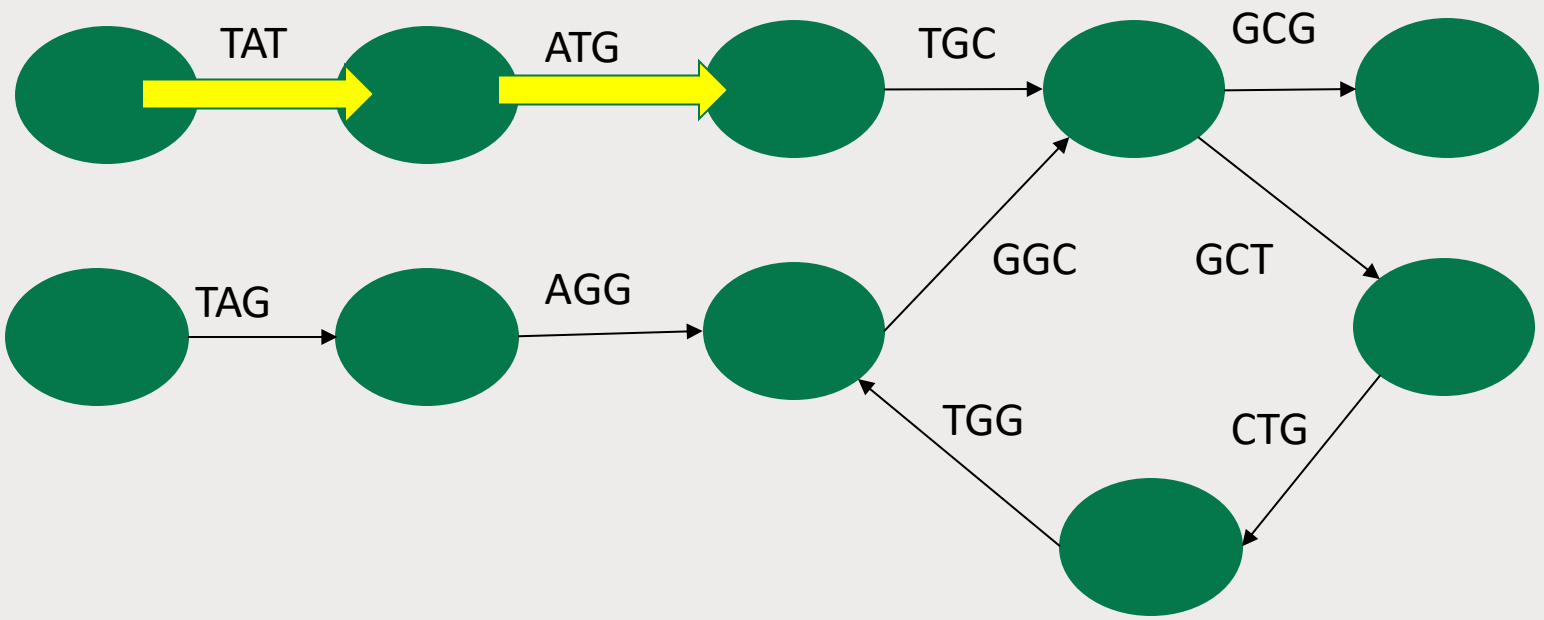
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TATG
```

De novo assembly using de Bruijn graphs

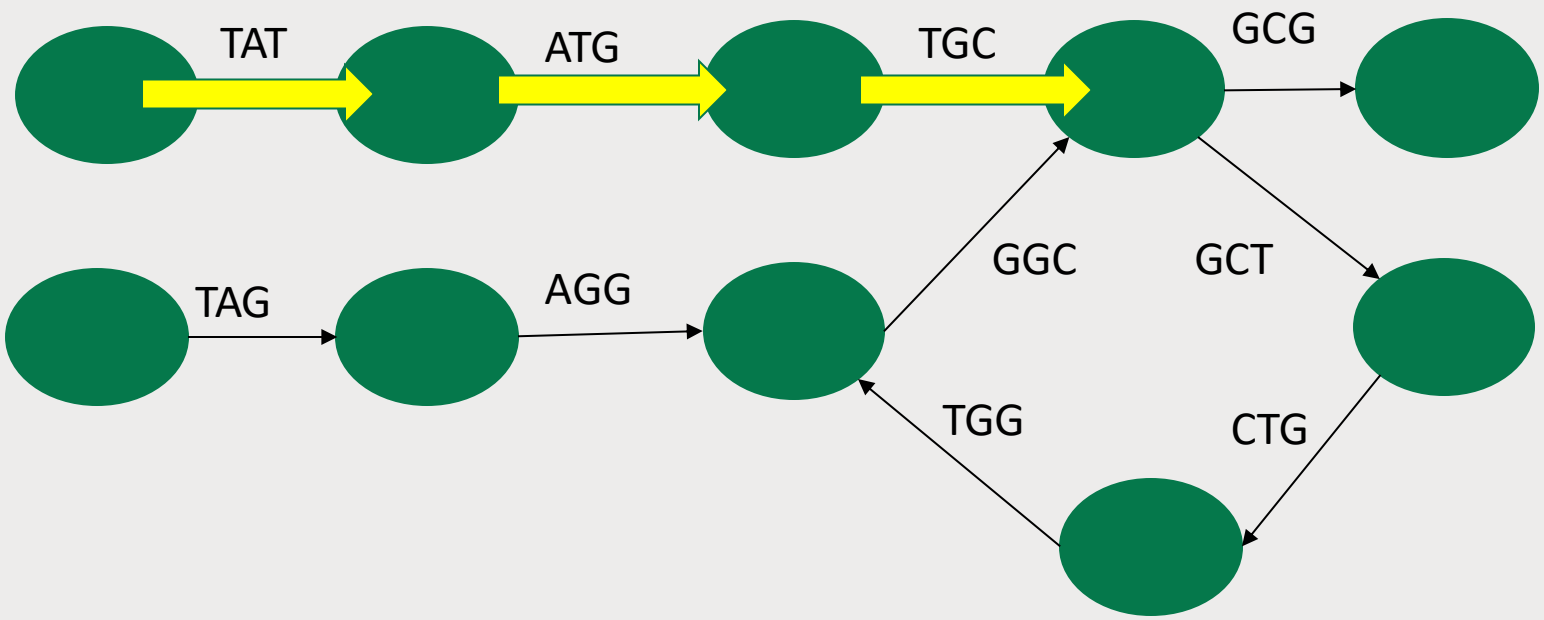
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TATGC
```

De novo assembly using de Bruijn graphs

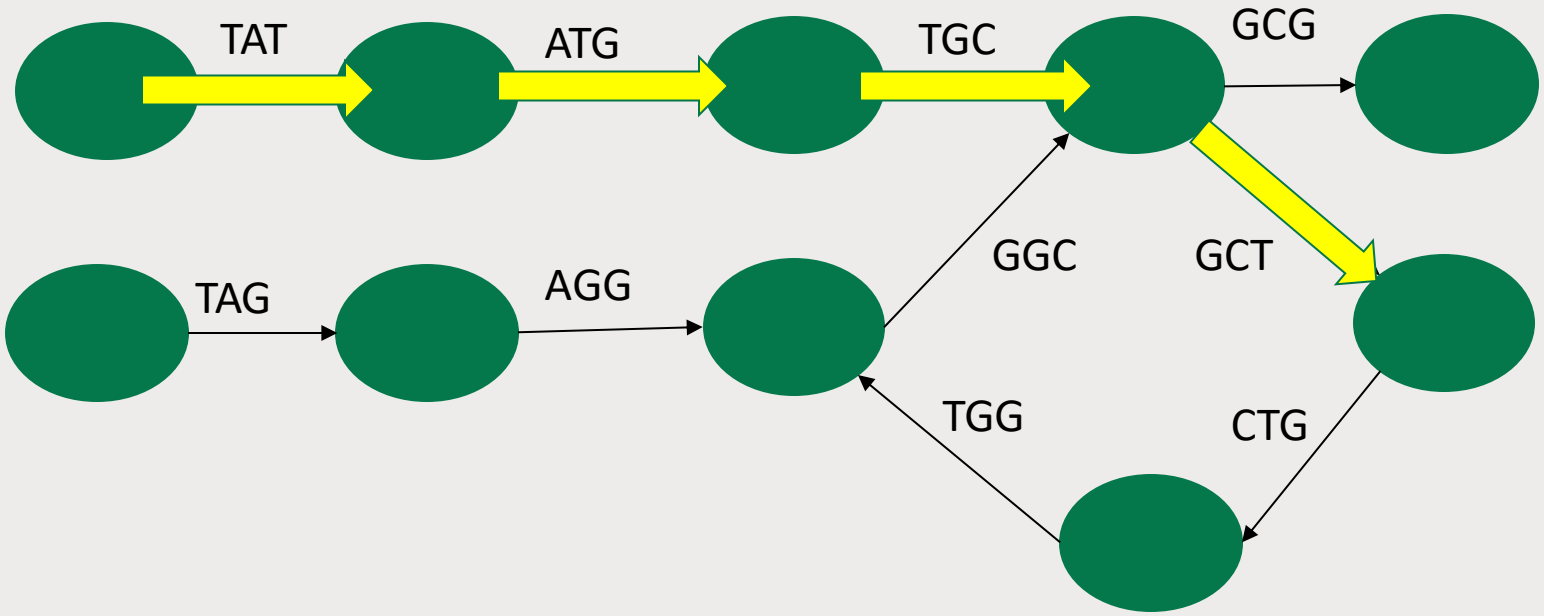
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TATGCT
```

De novo assembly using de Bruijn graphs

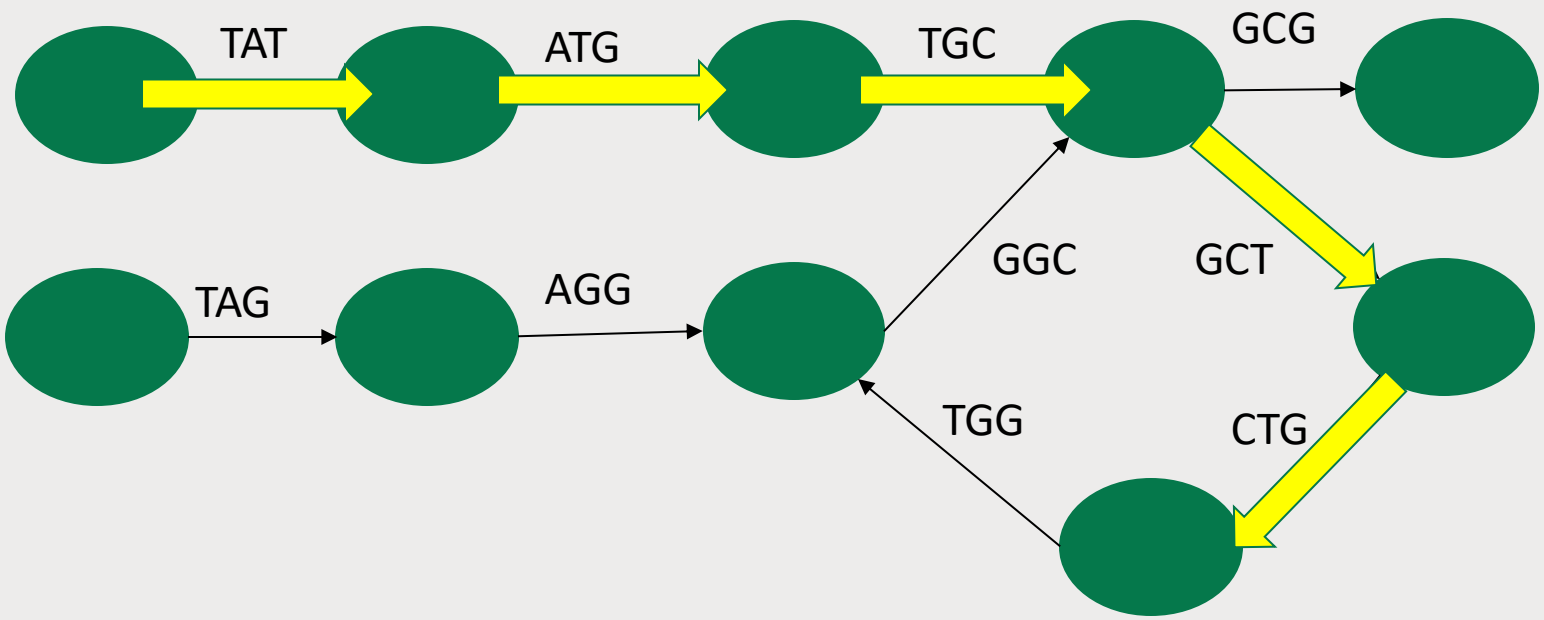
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TATGCTG
```

De novo assembly using de Bruijn graphs

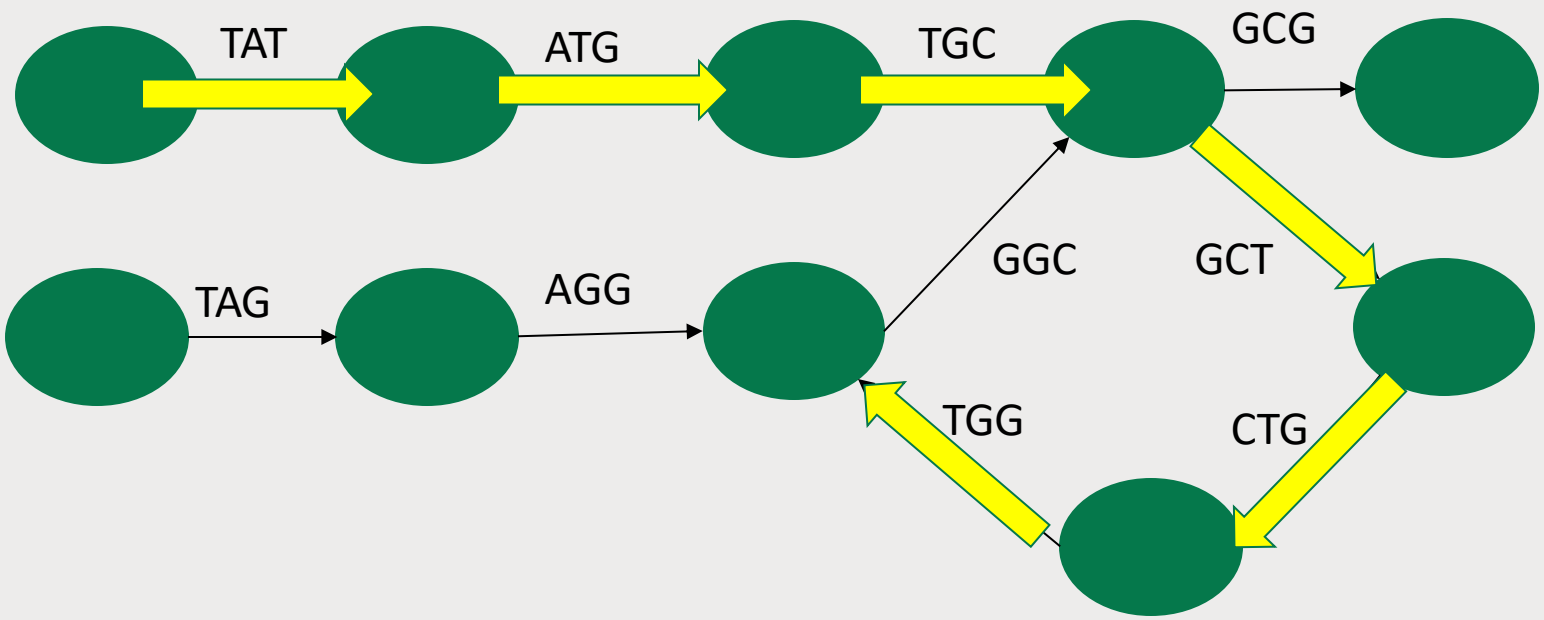
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TATGCTGG
```

De novo assembly using de Bruijn graphs

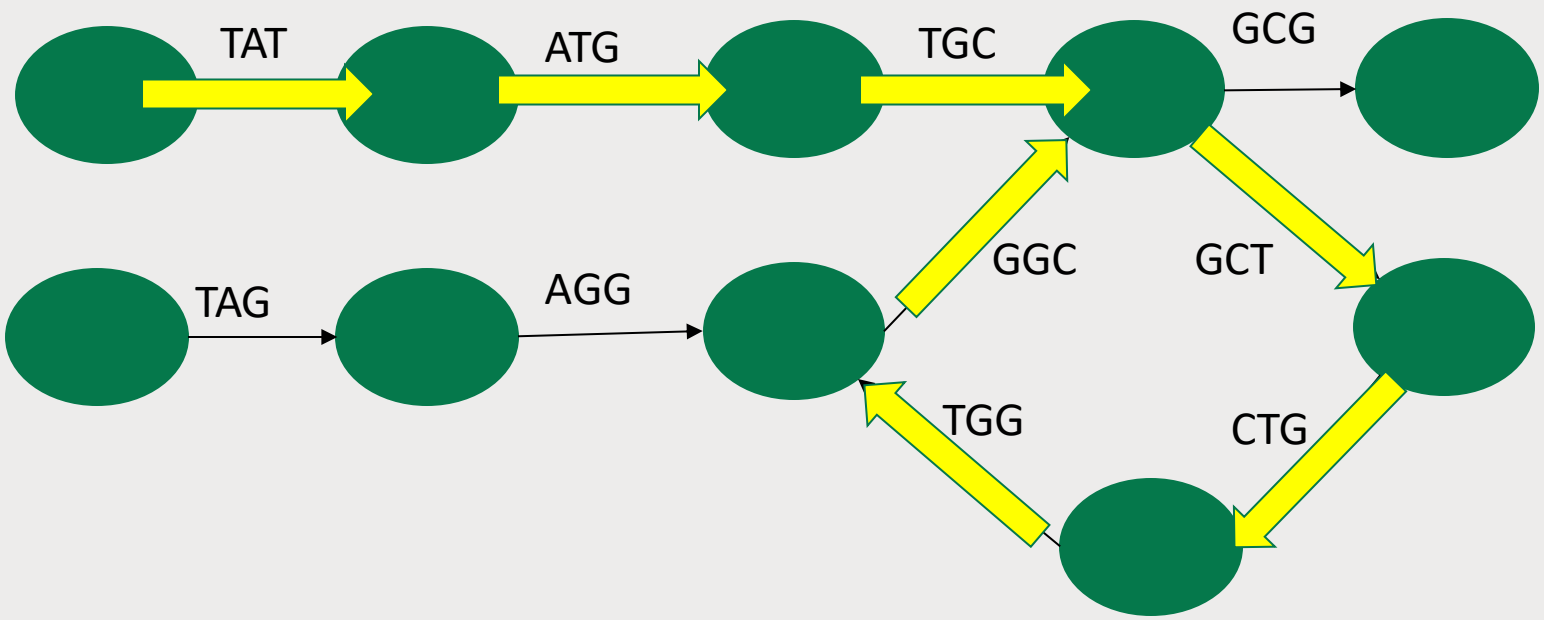
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG




```
>contig1
TATGCTGGC
```

De novo assembly using de Bruijn graphs

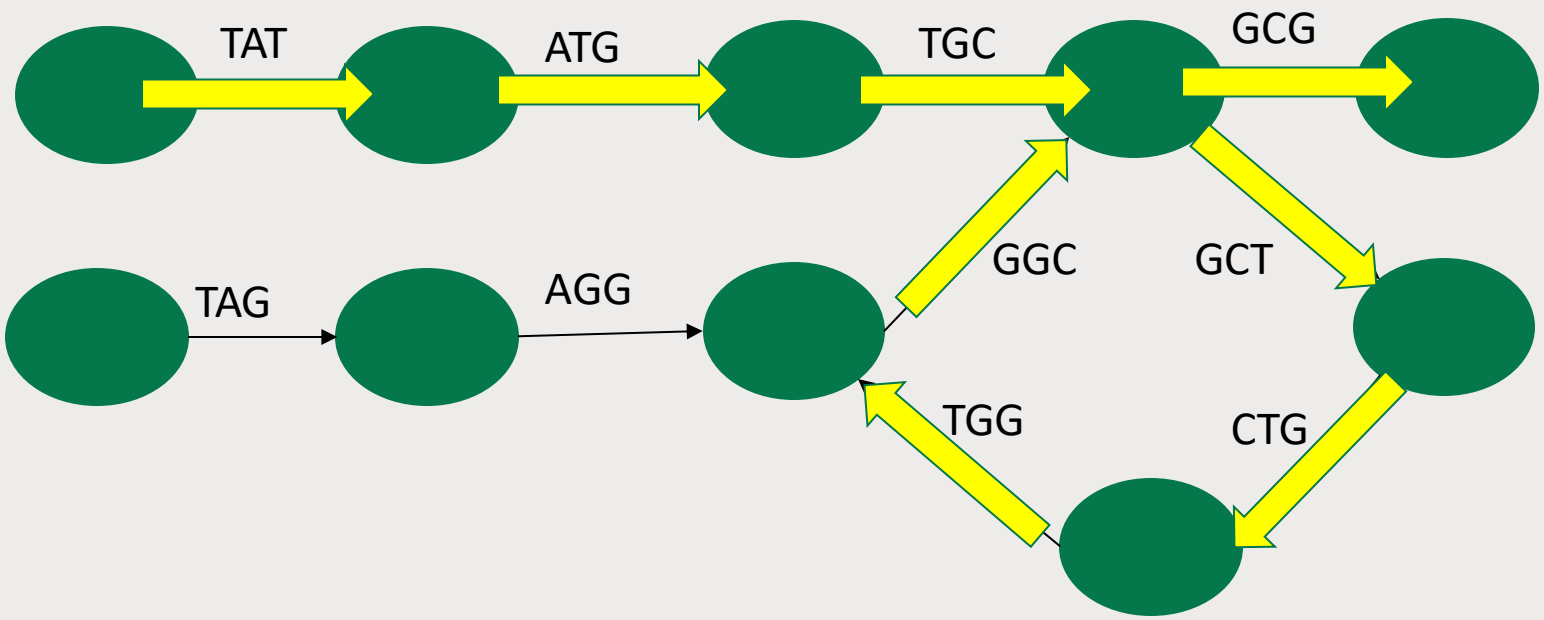
- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

Reads

ATGCG
AGGCG
TATGCG
TAGGCG

Resulting 3-mers

TAG
AGG
GGC
GCG

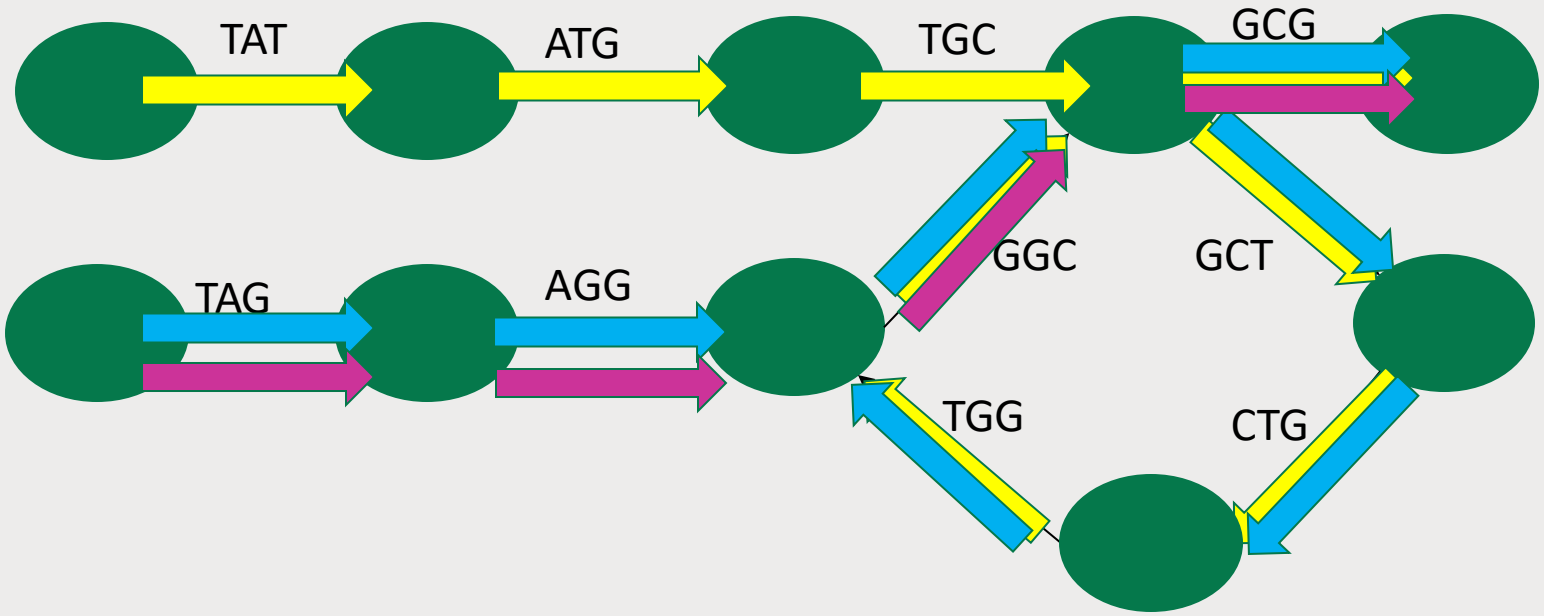



```
>contig1
TATGCTGGCG
```

De novo assembly using de Bruijn graphs

- When all reads have been processed your complete graph is resolved to get contigs
- Different assemblers may vary in how the resolve graphs

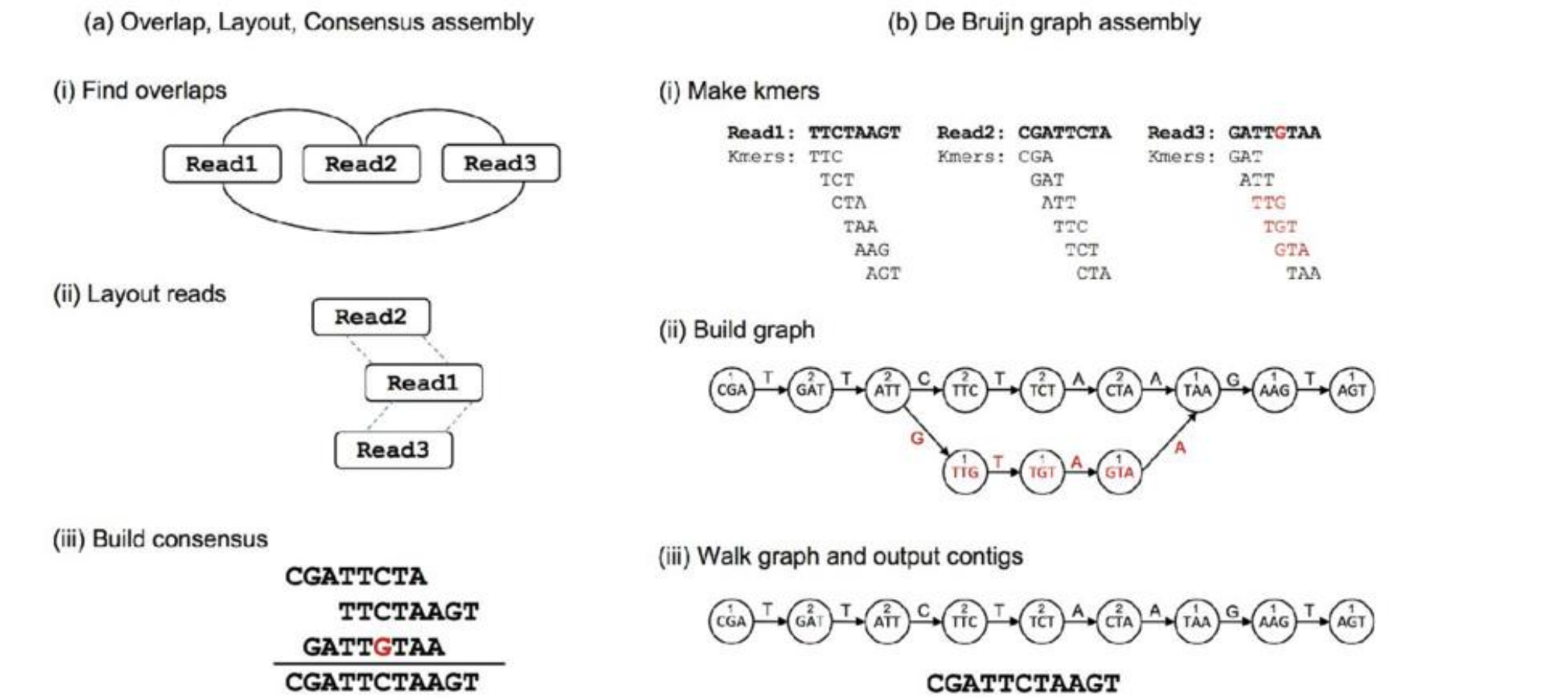
Reads	Resulting 3-mers
ATGCG	TAG
AGGCG	AGG
TATGCG	GGC
TAGGCG	GCG



?

```
>contig1
TATGCTGGCG
>contig2
TAGGCTGGCG
>contig2
TAGGCG
```

Overlap-Layout-Consensus (OLC) vs De Bruijn Graphs



New approaches for assembly of short-read metagenomic data - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Two-different-approaches-to-genome-assembly-a-in-Overlap-Layout-Consensus_fig1_328816579 [accessed 18 Dec 2024]

The Fleming Fund
Regional Grants

Let’s take a break ☺

The Fleming Fund
Regional Grants

Hybrid Assembly

The Fleming Fund
Regional Grants

Hybrid Assembly Approaches

Used when researchers want to leverage the strengths of both short-read and long-read sequencing data to produce a more complete and accurate genome assembly

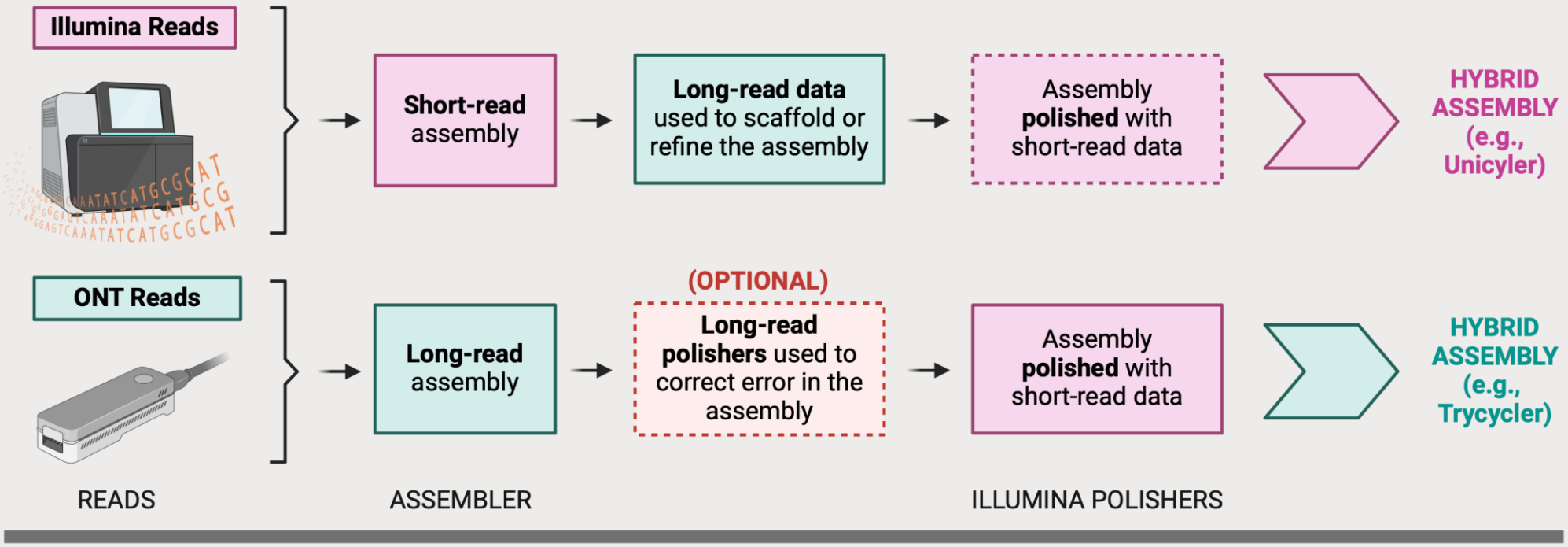


Image Created with BioRender

The Fleming Fund
Regional Grants

Assembly Tools for Hybrid Data

Most **long-read assembly tools** (e.g., Flye, Raven) can be integrated into hybrid assembly workflows, where initial assemblies are polished with long-read tools followed by further polishing with short-read data

Assembly Tools

- **Unicycler**
- **Hybracter**
- **Trycycler** (consensus tool)
- Dragonflye
- HybridSPAdes
- MaSuRCA

Do we still need Illumina sequencing data? Evaluating Oxford Nanopore Technologies R10.4.1 flow cells and the Rapid v14 library prep kit for Gram negative bacteria whole genome assemblies

Nicole Lerminiaux, Ken Fakharuddin, Michael R. Mulvey, and Laura Mataseje

When **compared to Flye and Raven** → **Unicycler** produced the most accurate assemblies, closely resembling reference genomes, with fewer issues such as missing or duplicated plasmids

Wick et al. (2017). PLoS Comput Biol. 13(6):e1005595; Bouras et al. (2024). Microb Genom. 10(5):001244; Wick et al. (2021). Genome Biol. 22(1):266

Assembly Tools for Hybrid Data

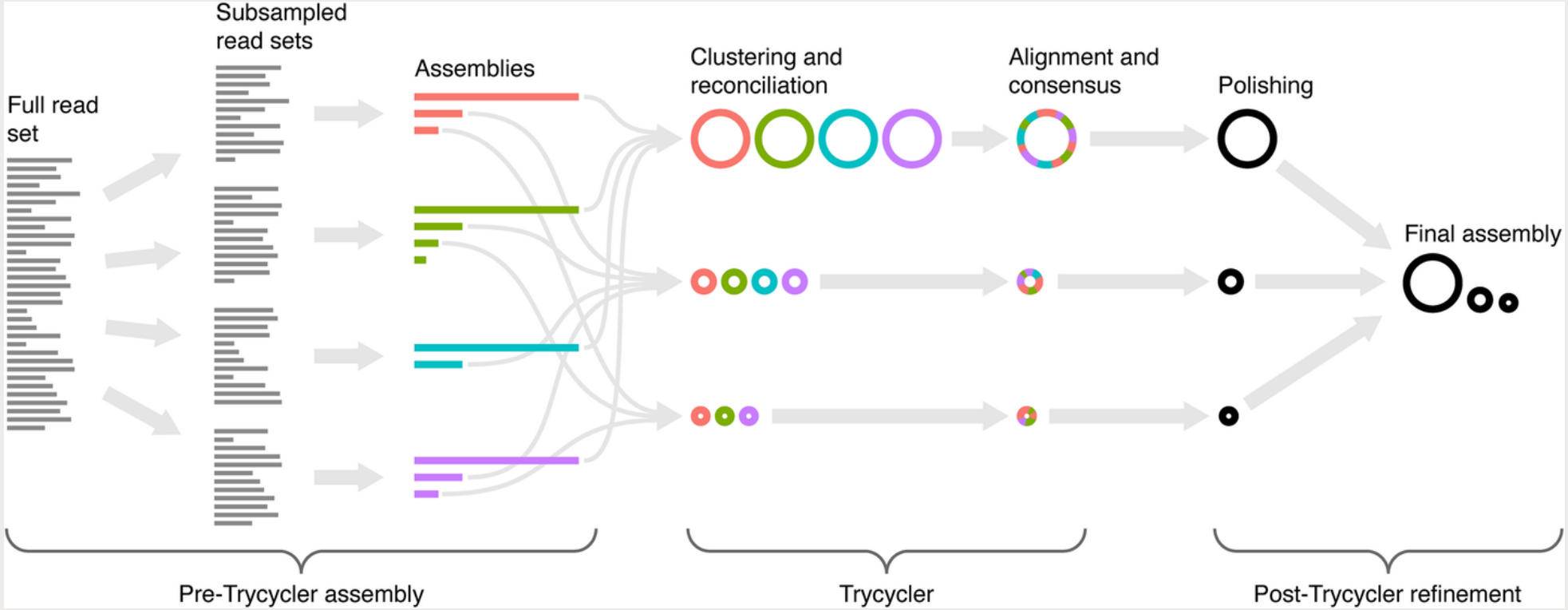
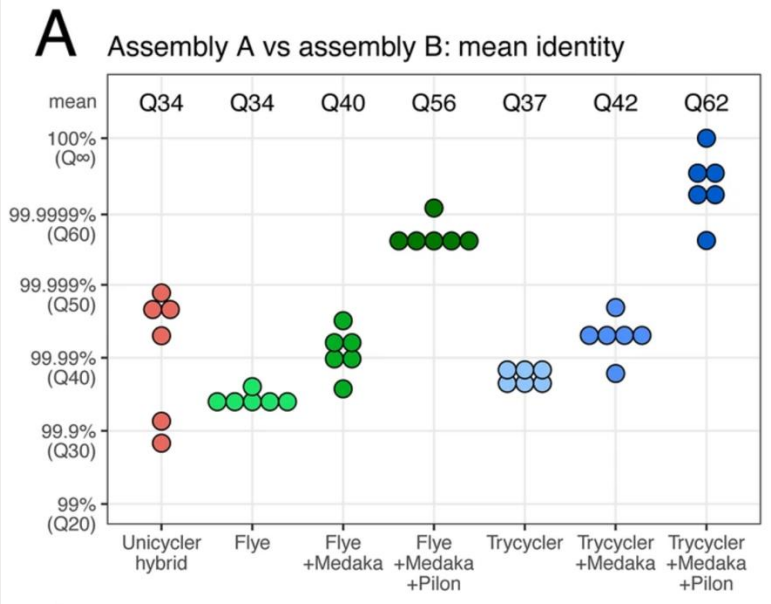


Fig. Overview of the Trycycler long-read assembly pipeline. Before Trycycler is run, the user must generate multiple complete assemblies of the same genome, e.g., by assembling different subsets of the original long-read set. Trycycler then clusters contigs from different assemblies and produces a consensus contig for each cluster. These consensus contigs can then be polished (e.g., with Medaka) and combined into a final high-quality long-read-only assembly

Assembly Tools for Hybrid Data

For bacterial genomes, a **Trycycler+Medaka(optional)+Pilon** approach can deliver assemblies which are very close to this goal: approximately one error per 2 Mbp, equivalent to two errors in an *E. coli* genome



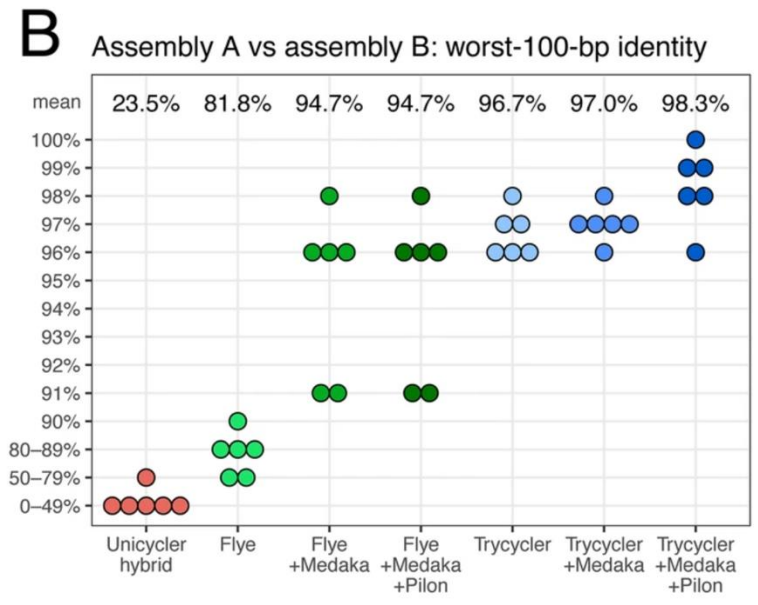


Fig. Results for the real-read tests. For six genomes, we produced two independent hybrid read sets from the same DNA extraction. For each genome and each assembly approach, we aligned the two independently assembled chromosomes to each other to determine the mean assembly identity **(A)** and the worst identity in a 100-bp sliding window **(B)**.

Wick et al. (2021). Genome Biol. 22(1):266; https://github.com/rrwick/Trycycler

The Fleming Fund
Regional Grants

Assembly Tools for Hybrid Data

Aims to provide a comprehensive tutorial based on Trycycler for achieving **error-free bacterial genome assemblies** by integrating Oxford Nanopore Technologies (ONT) long-read sequencing with Illumina short-read sequencing

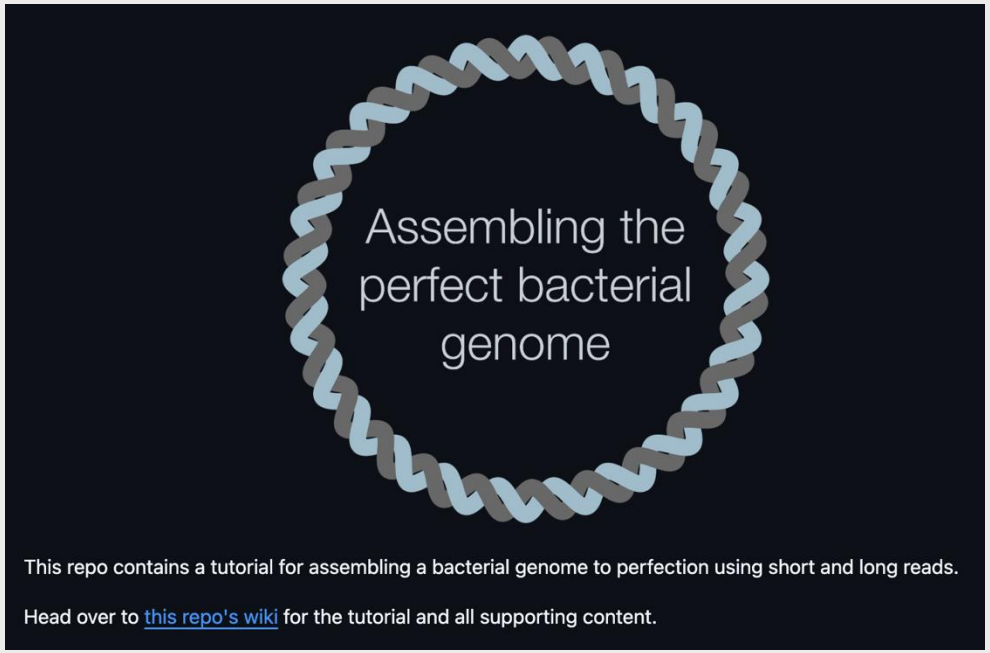
PLOS COMPUTATIONAL BIOLOGY

EDUCATION

Assembling the perfect bacterial genome using Oxford Nanopore and Illumina sequencing

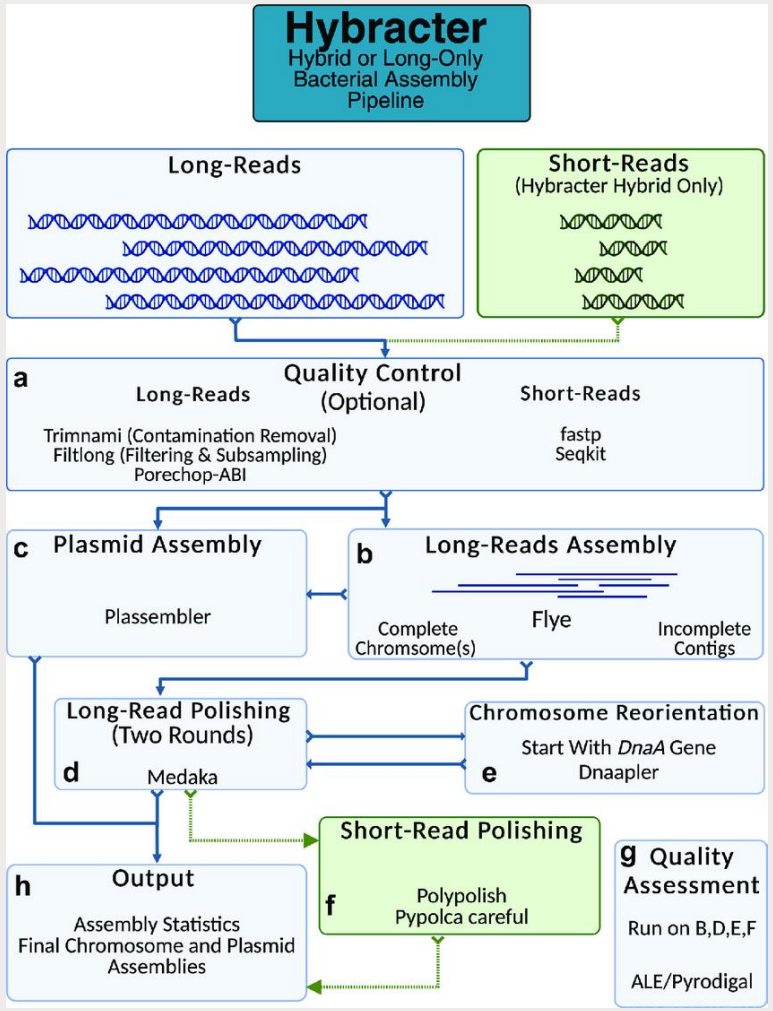
Ryan R. Wick[1]*, Louise M. Judd[2], Kathryn E. Holt[1,3]

1 Department of Infectious Diseases, Central Clinical School, Monash University, Melbourne, Australia, **2** Department of Microbiology and Immunology, University of Melbourne at the Peter Doherty Institute for Infection and Immunity, Melbourne, Australia, **3** Department of Infection Biology, London School of Hygiene & Tropical Medicine, London, United Kingdom



Wick et al. (2023). PLoS Comput Biol. 19(3):e1010905; https://github.com/rrwick/Perfect-bacterial-genome-tutorial

Assembly Tools for Hybrid Data



Hybracter builds on the principles of Trycycler but incorporates short-read data polishing directly into the workflow → **minimal manual intervention**

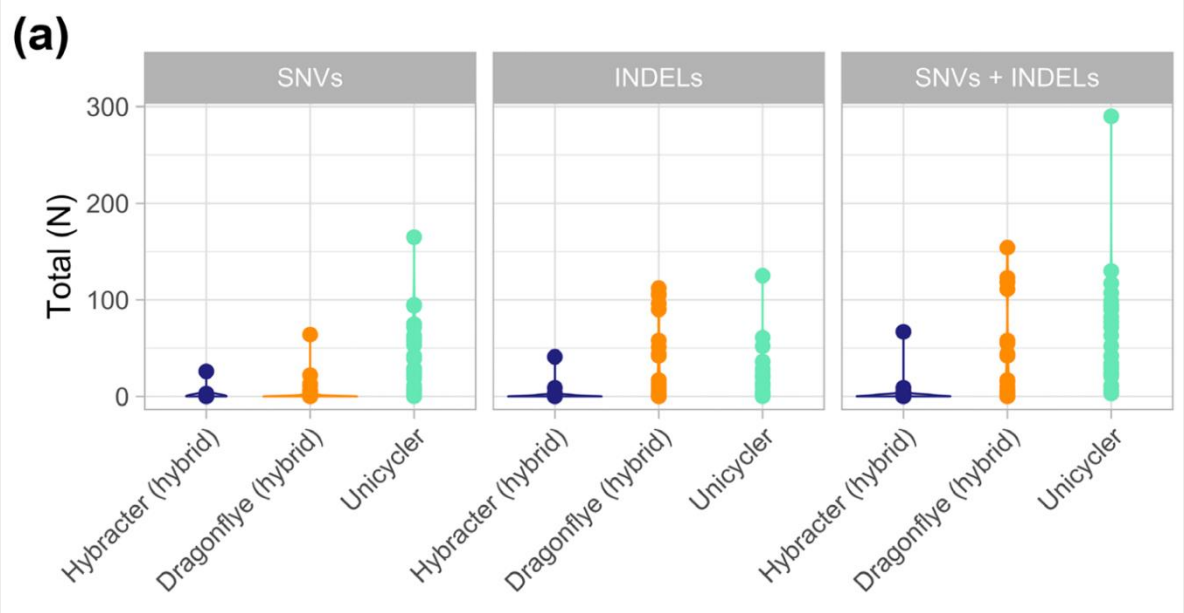


Fig. Comparison of the counts of SNVs, small (<60 bp) INDELs for the hybrid tools benchmarked

Bouras et al. (2024). Microb Genom. 10(5):001244; https://github.com/gbouras13/hybracter

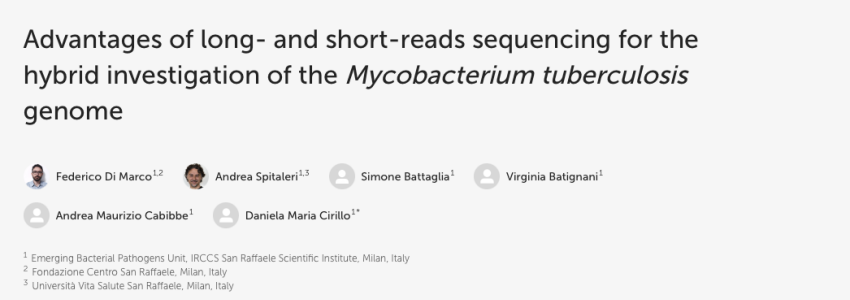
The Fleming Fund
Regional Grants

Short vs Long vs Hybrid sequencing - What's in the literature?

- Advantages of long- and short-reads sequencing for the hybrid investigation of the Mycobacterium tuberculosis genome (Feb, 2023)

- **Background**: *Mycobacterium tuberculosis* (MTB) genome contains ~10% PE/PPE family genes, characterized by GC-rich repetitive regions.
- **Challenge**: Short-read sequencing (SRS) struggles with accurately mapping these repetitive regions, leading to incomplete assemblies.
- **Objective**: Evaluate and compare the effectiveness of SRS, long-read sequencing (LRS), and hybrid sequencing (HYBR) in analysing the MTB genome.

Di Marco F, Spitaleri A, Battaglia S, Batignani V, Cabibbe AM, Cirillo DM. Advantages of long- and short-reads sequencing for the hybrid investigation of the *Mycobacterium tuberculosis* genome. Front Microbiol. 2023 Feb 2;14:1104456. doi: 10.3389/fmicb.2023.1104456. PMID: 36819039; PMCID: PMC9932330.


Advantages of long- and short-reads sequencing for the hybrid investigation of the *Mycobacterium tuberculosis* genome

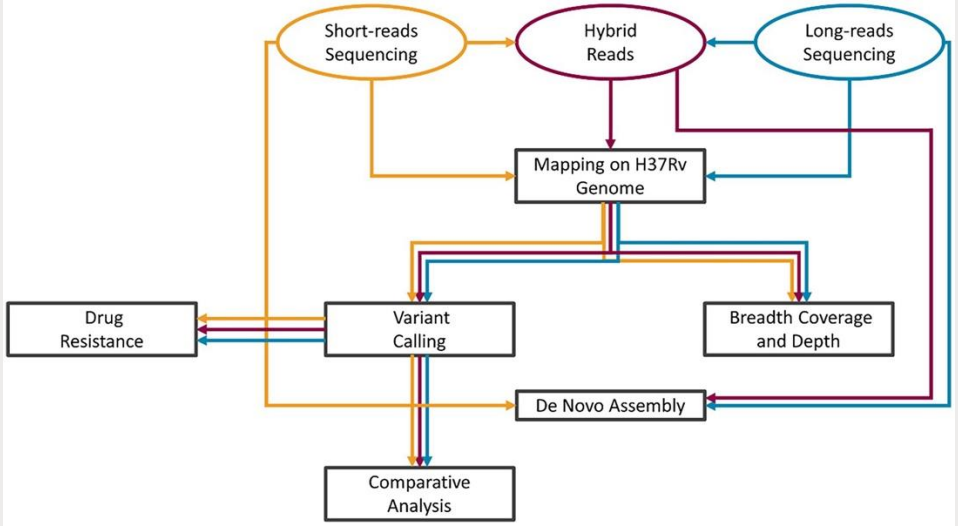
Federico Di Marco[1,2] Andrea Spitaleri[1,3] Simone Battaglia[1] Virginia Batignani[1]
Andrea Maurizio Cabibbe[1] Daniela Maria Cirillo[1*]

[1] Emerging Bacterial Pathogens Unit, IRCCS San Raffaele Scientific Institute, Milan, Italy
[2] Fondazione Centro San Raffaele, Milan, Italy
[3] Università Vita Salute San Raffaele, Milan, Italy

Methods

- Sample Set: 13 clinical MTB isolates.
- Sequencing Techniques:
  - SRS: High-accuracy short reads.
  - LRS: Longer reads capable of spanning repetitive regions.
  - HYBR: Combination of SRS and LRS, with long reads corrected using short reads.

- Analytical Focus:
  - Genome coverage estimation.
  - Variant calling and cluster analysis.
  - Drug resistance detection.
  - De novo assembly evaluation.

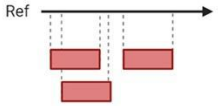
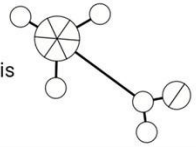


Di Marco F, Spitaleri A, Battaglia S, Batignani V, Cabibbe AM, Cirillo DM. Advantages of long- and short-reads sequencing for the hybrid investigation of the *Mycobacterium tuberculosis* genome. Front Microbiol. 2023 Feb 2;14:1104456. doi: 10.3389/fmicb.2023.1104456. PMID: 36819039; PMCID: PMC9932330.

The Fleming Fund
Regional Grants

Results

- **Genome Coverage**:
  - HYBR provided superior coverage, especially in GC-rich PE/PPE regions.
- **Variant Calling**:
  - HYBR approach enhanced the accuracy of single nucleotide polymorphism (SNP) detection.
- **Drug Resistance Detection**:
  - All three methods identified known resistance mutations, but HYBR offered higher confidence levels.
- **De Novo Assembly**:
  - HYBR assemblies were more contiguous and accurate, effectively resolving repetitive regions.

	SRS	LRS	HYBR
Genome Coverage 	! Low coverage in repetitive regions (41/169)	✓	✓
Variant Calling Comparative Analysis 	✓	! High-coverage to overcome error rate	✓
Drug Resistance 	✓	✓	✓
De Novo Assembly 	✗ High number of contigs Low NG50	✓	✓

Di Marco F, Spitaleri A, Battaglia S, Batignani V, Cabibbe AM, Cirillo DM. Advantages of long- and short-reads sequencing for the hybrid investigation of the *Mycobacterium tuberculosis* genome. Front Microbiol. 2023 Feb 2;14:1104456. doi: 10.3389/fmicb.2023.1104456. PMID: 36819039; PMCID: PMC9932330.

Conclusions

- Advantages of Hybrid Sequencing:
  - Combines the accuracy of SRS with the extended reach of LRS.
  - Delivers comprehensive genome assemblies, crucial for understanding MTB's genetic structure.
  - Improves detection of variants and drug resistance markers, aiding in better clinical decision-making.

- Recommendation: Implementing hybrid sequencing approaches is highly beneficial for the genomic investigation of MTB and potentially other organisms with complex genomes.

Should I use Unicycler or Trycycler to assemble my bacterial genome?

- If you have lots of long reads (~100× depth or more), use Trycycler+polishing. If you have sparse long reads (~25× or less), use Unicycler. If your long-read depth falls between those values, it might be worth trying both approaches.
- Unicycler works best when the short-read set is very good (deep and complete coverage) which yields a nice short-read assembly graph for scaffolding. Conversely, when Unicycler fails, it's usually due to problems with the short-read assembly graph.
- The Trycycler+polishing approach is much less dependent on the quality of the short-read set. However, Trycycler requires a deep long-read set while Unicycler does not.
- Occasions where small misassemblies occur within short-read contigs in Unicycler (made by SPAdes). This usually happens in repetitive regions of the genome. Since Unicycler builds its final assembly by scaffolding the short-read contigs, any misassemblies they contain will persist in the final assembly. Trycycler seems to do much better in such regions.

A note on Unicycler

- Unicycler was built in a different time (2016) when Oxford Nanopore read sets could be quite shallow, so it was necessary to rely more on short-read sets.

- Since then, improvements in Oxford Nanopore yield have largely fixed that problem.

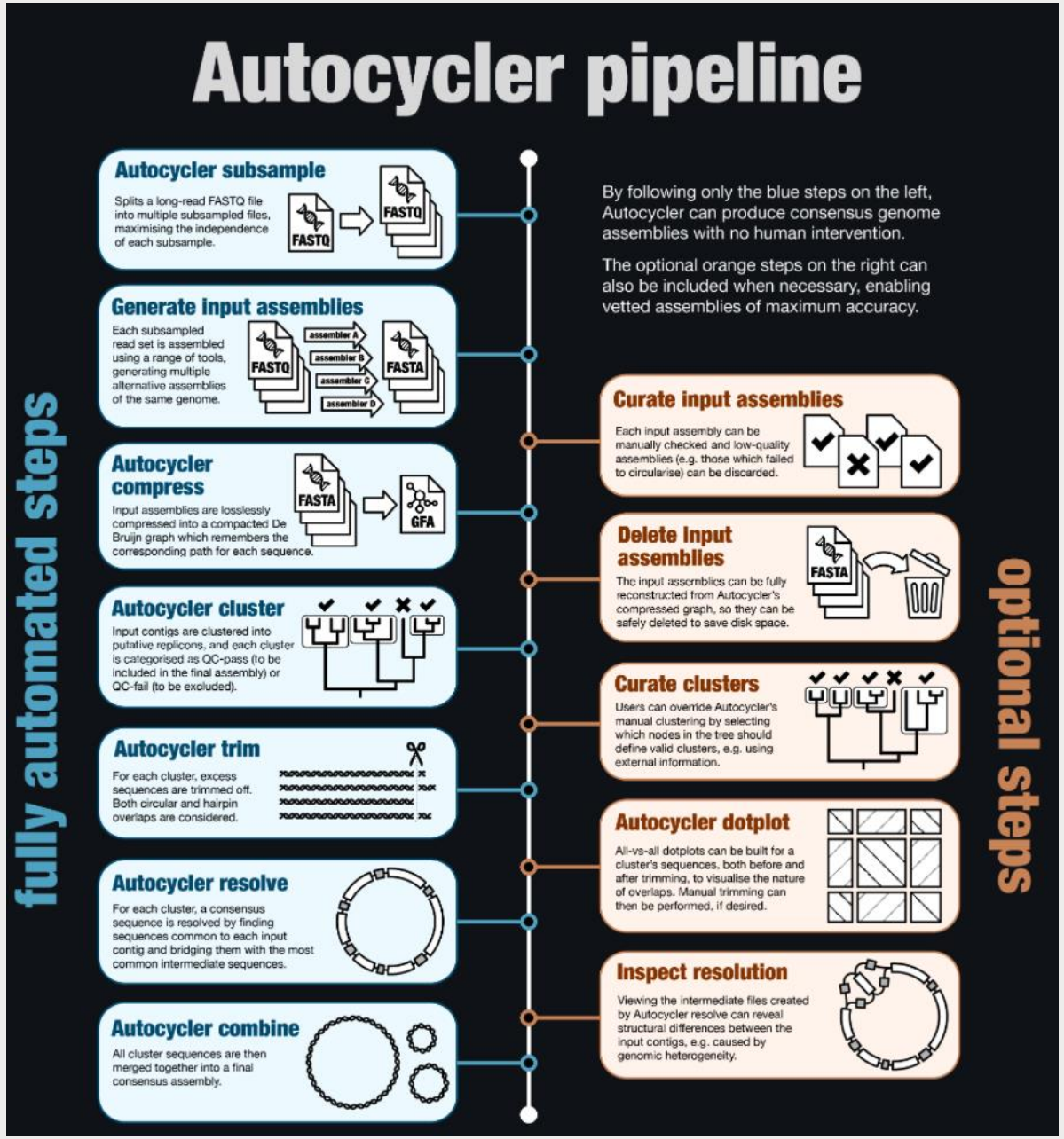
- SoTrycycler+polishing is probably the best way to do a hybrid bacterial genome assembly, with Unicycler as a fall-back option for cases where your short-read set is good but your long-read set is weak.

The Fleming Fund
Regional Grants

Autocycler (re-written Trycycler)

- Autocycler was released end of 2024
- A complete rewrite of Trycycler designed for improved performance and automation.

https://github.com/rrwick/Autocycler



The Fleming Fund
Regional Grants

Let’s take a break ☺

The Fleming Fund
Regional Grants

Assembly QC

Why Assembly QC is Important?

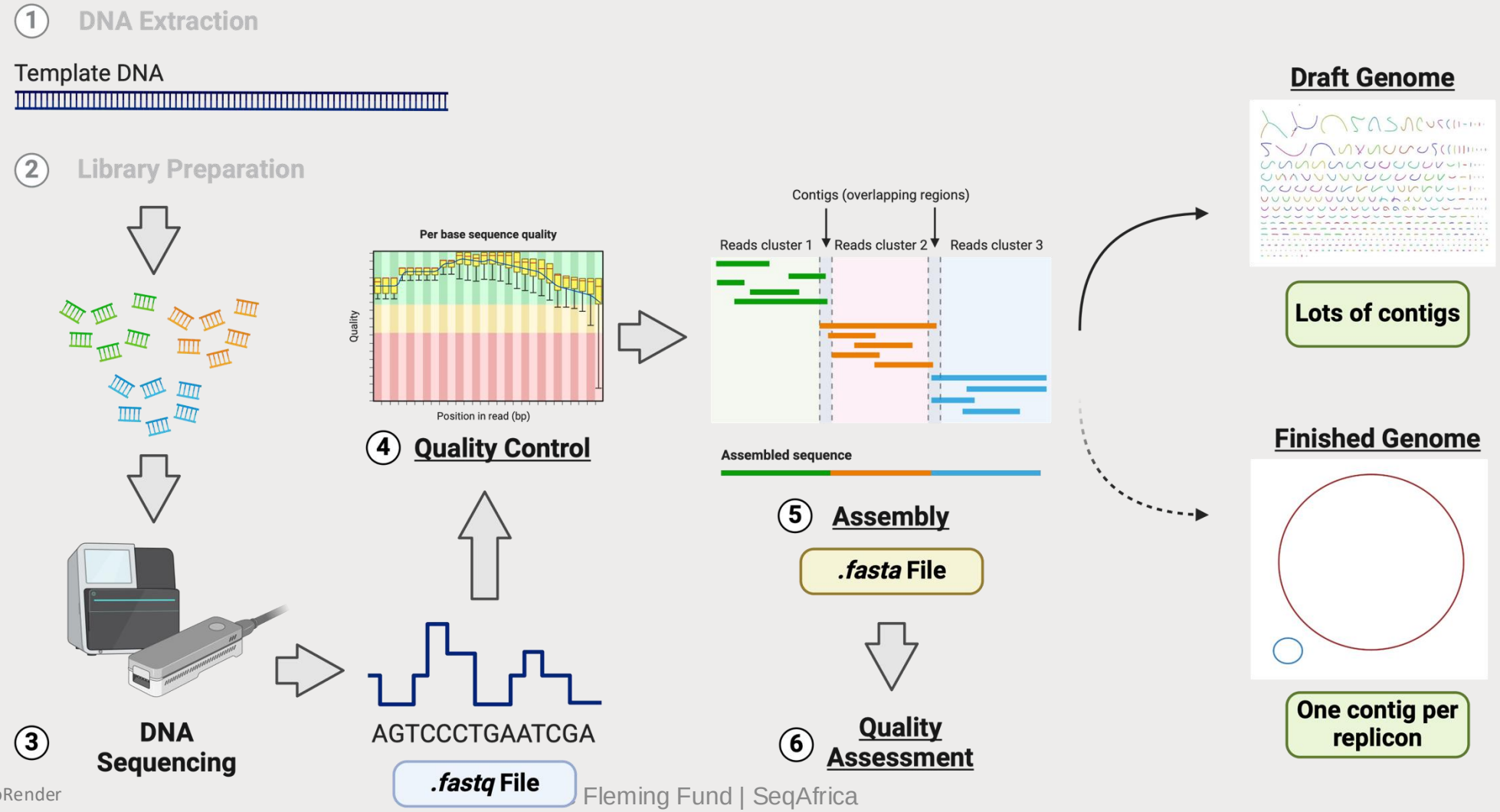


Image Created with BioRender

The Fleming Fund
Regional Grants

Overview of Assembly QC

Extent to which a genome is assembled into **long, uninterrupted sequences**

Contiguity

Completeness

Correctness

Proportion of the **original genome represented** by the **assembly**

BUSCO

from QC to gene prediction and phylogenomics

(estimates genome completeness – based on core genes)

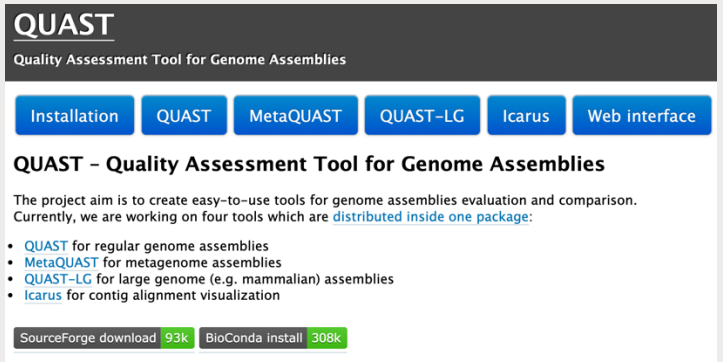
https://busco.ezlab.org/

+

CheckM

(estimates genome completeness and contamination – core genes)

https://ecogenomics.github.io/CheckM/



Proportion of the assembly that is **free from mistakes** (e.g., using **polishing tools**):

- Check for self consistency
- Align all the reads back to contigs
- Look for inconsistencies

The Fleming Fund
Regional Grants

QUAST – Quality Assessment Tool

Provides comprehensive **metrics** and visual reports to assess the **accuracy**, **completeness**, and **contiguity** of assemblies → suitable for *de novo* **short-read**, **long-read**, and **hybrid** assemblies

```
# Install QUAST

# At the end of the run an html report will be provided with the results

$ conda install -c bioconda quast

# Run QUAST in your assembly FASTA files

$ quast.py -o quast_results ~/assembly/my_hybrid_assembly.fasta

# Run QUAST in your assembly FASTA files but provide a reference genome

$ quast.py -r ~/tutorial/raw_data/reference.fasta -g ~/tutorial/raw_data/annotation.gff -o quast_results ~/assembly/my_hybrid_assembly.fasta
```

BV-BRC Bacterial and Viral Bioinformatics Resource Center
Galaxy
KBase PREDICTIVE BIOLOGY

The Fleming Fund
Regional Grants

QUAST – Quality Assessment Tool

All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

Aligned to "Isolate_3_Unicycler" | 5 339 413 bp | 3 fragments | 51.89% G+C

Worst Median Best

☑ Show heatmap

Genome statistics	Isolate_3_SPADES	Isolate_3_SKESA	Isolate_3_Flye	Isolate_3_Hybracter
Genome fraction (%)	98.634	98.224	100	100
Duplication ratio	1	1	1	1
Largest alignment	418 481	386 345	4 972 900	4 972 906
Total aligned length	5 267 928	5 244 491	5 337 245	5 339 411
NGA50	238 829	223 407	4 972 900	4 972 906
LGA50	9	9	1	1
Misassemblies				
# misassemblies	1	0	3	0
Misassembled contigs length	55 374	0	366 501	0
Mismatches				
# mismatches per 100 kbp	0.23	0.51	1.72	0.97
# indels per 100 kbp	0.04	0.02	0.28	0.07
# N's per 100 kbp	0	0	0	0
Statistics without reference				
# contigs	73	70	2	3
Largest contig	418 481	386 345	4 972 900	4 972 906
Total length	5 268 036	5 246 420	5 339 401	5 339 411
Total length (>= 1000 bp)	5 256 843	5 241 952	5 339 401	5 339 411
Total length (>= 10000 bp)	5 178 984	5 152 802	5 339 401	5 339 411
Total length (>= 50000 bp)	4 909 047	4 695 688	5 339 401	5 339 411

Genome Fraction (%): percentage of the reference genome that is covered by the assembled contigs → related to the **Total Aligned Length**

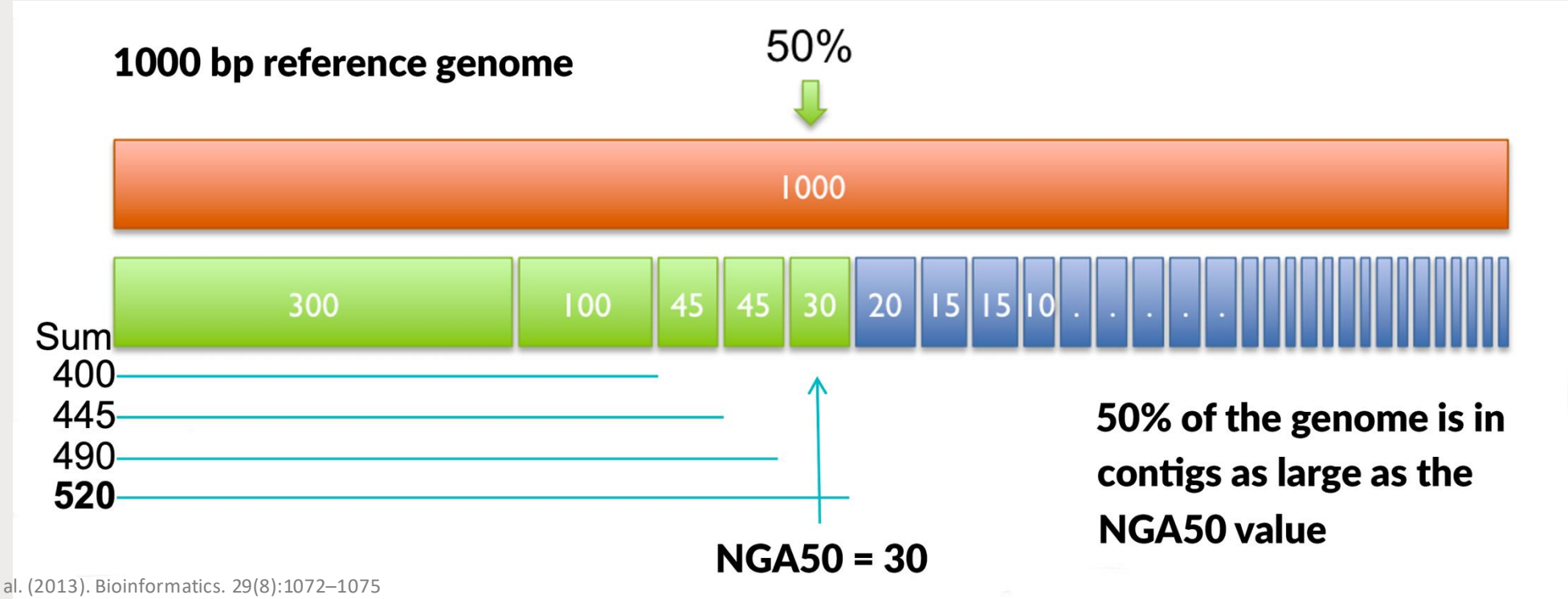
Gurevich et al. (2013). Bioinformatics. 29(8):1072–1075

The Fleming Fund
Regional Grants

QUAST – Quality Assessment Tool

NGA50: the length of the contig at which 50% of the reference genome is covered by contigs of that length or larger → **higher NGA50, better assembly continuity**

LGA50: the smallest number of aligned contigs that together constitute 50% of the reference genome size → **smaller LGA50, better assembly continuity**



Gurevich et al. (2013). Bioinformatics. 29(8):1072–1075

The Fleming Fund
Regional Grants

QUAST – Quality Assessment Tool

All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

Aligned to "Isolate_3_Unicycler" | 5 339 413 bp | 3 fragments | 51.89% G+C

Worst Median Best

☑ Show heatmap

Genome statistics	Isolate_3_SPADES	Isolate_3_SKESA	Isolate_3_Flye	Isolate_3_Hybracter
Genome fraction (%)	98.634	98.224	100	100
Duplication ratio	1	1	1	1
Largest alignment	418 481	386 345	4 972 900	4 972 906
Total aligned length	5 267 928	5 244 491	5 337 245	5 339 411
NGA50	238 829	223 407	4 972 900	4 972 906
LGA50	9	9	1	1
Misassemblies				
# misassemblies	1	0	3	0
Misassembled contigs length	55 374	0	366 501	0
Mismatches				
# mismatches per 100 kbp	0.23	0.51	1.72	0.97
# indels per 100 kbp	0.04	0.02	0.28	0.07
# N's per 100 kbp	0	0	0	0
Statistics without reference				
# contigs	73	70	2	3
Largest contig	418 481	386 345	4 972 900	4 972 906
Total length	5 268 036	5 246 420	5 339 401	5 339 411
Total length (>= 1000 bp)	5 256 843	5 241 952	5 339 401	5 339 411
Total length (>= 10000 bp)	5 178 984	5 152 802	5 339 401	5 339 411
Total length (>= 50000 bp)	4 909 047	4 695 688	5 339 401	5 339 411

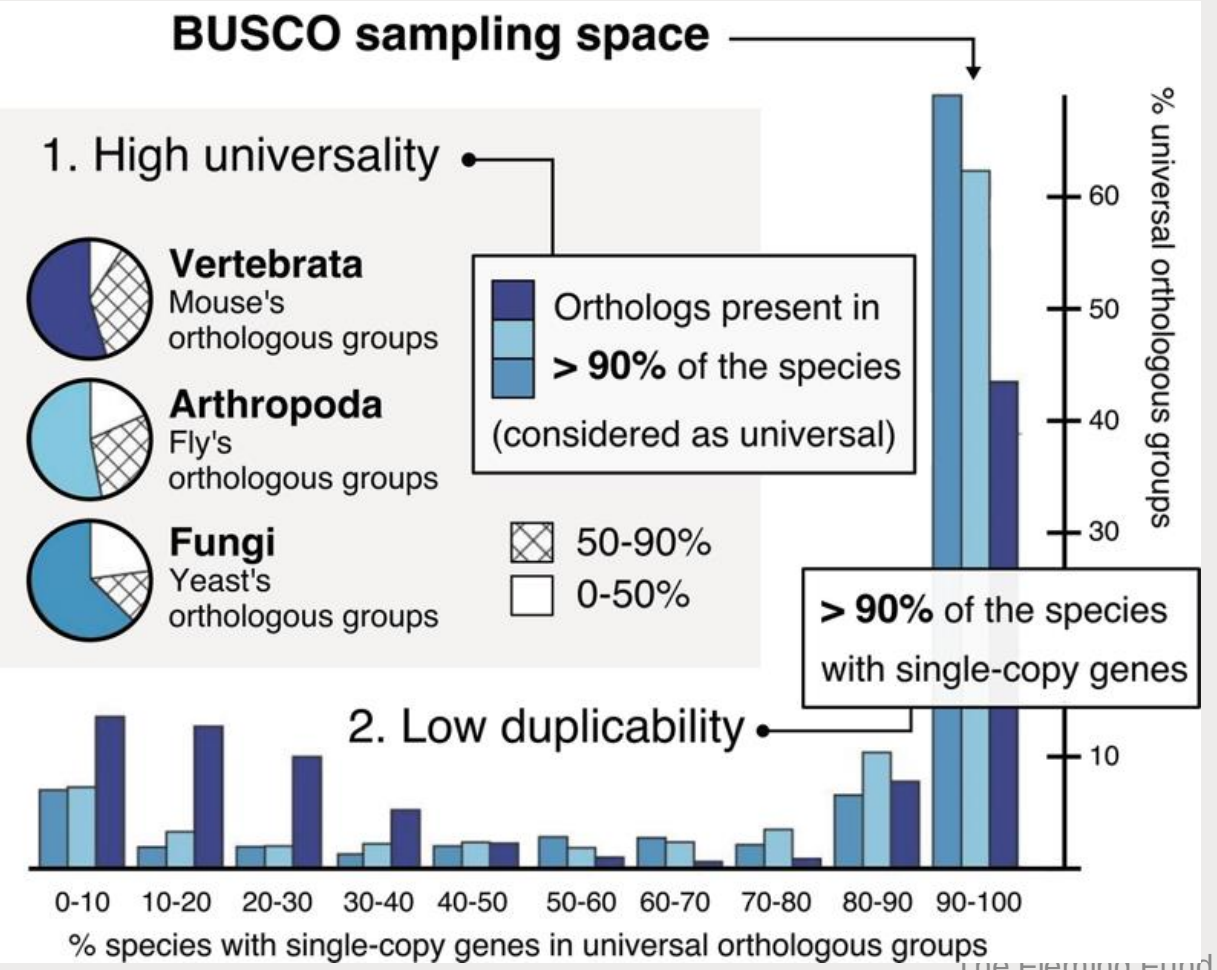
Mismatches and Indels: average number of base mismatches or insertions/deletions per 100.000 bp → **lower number** suggest **higher accuracy**

Contigs: total number of contigs in the assembly → **fewer contigs** generally suggest **better contiguity**

Often, contigs below a certain threshold are not counted (e.g., 200 bp or 500 bp)

Gurevich et al. (2013). Bioinformatics. 29(8):1072–1075

BUSCO



- Evaluate the completeness of genome assemblies by determining how many highly conserved, single-copy orthologous genes are present in the assembly
- Provides a quantitative measure of genome quality by checking for the **presence**, **completeness**, and **duplication** of the core genes

Manni et al. (2021). Curr Protoc. 1(12):e323

The Fleming Fund
Regional Grants

BUSCO

```
# Install BUSCO

$ conda install -c conda-forge -c bioconda busco=5.8.2

# Check all available datasets

$ busco --list-datasets

# Run QUAST in your assembly FASTA files

# -m: Mode (genome, proteins, or transcriptome)

$ busco -i ~/assembly/my_hybrid_assembly.fasta -l bacteria_odb12 -o busco_results -m geno
```

Galaxy

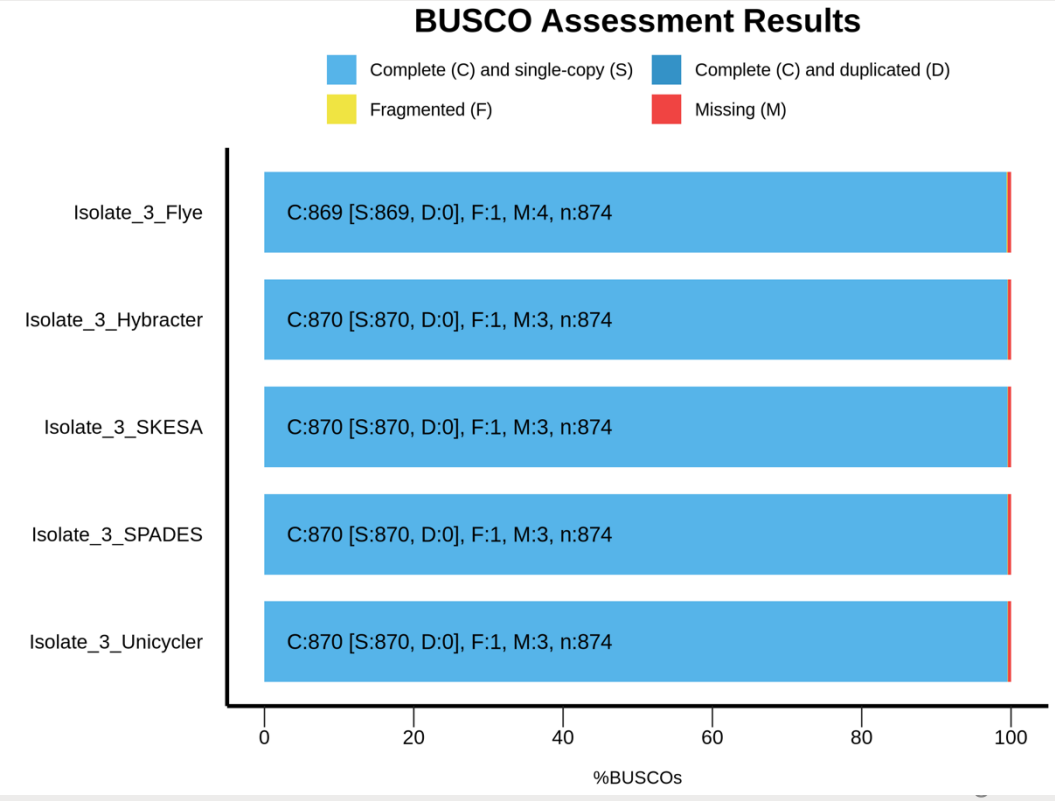
KBase
PREDICTIVE BIOLOGY

BUSCO relies on **taxonomy-** and **lineage-specific datasets** of universal **single-copy orthologs**. It provides precomputed datasets of orthologs for specific taxonomic groups (e.g., **bacteria_odb12**) or lineages (e.g., **enterobacterales_odb12**)

Manni et al. (2021). Curr Protoc. 1(12):e323

BUSCO

Uses tools like HMMER and BLAST to search for orthologs in the genome assembly, identifying their **presence**, **length**, and **completeness**



Orthologous genes are classified into the **following categories**:

Complete (C): Present and full-length

Duplicated (D): Present more than once, indicating **duplication**

Fragmented (F): **Partial** genes present, likely due to assembly gaps

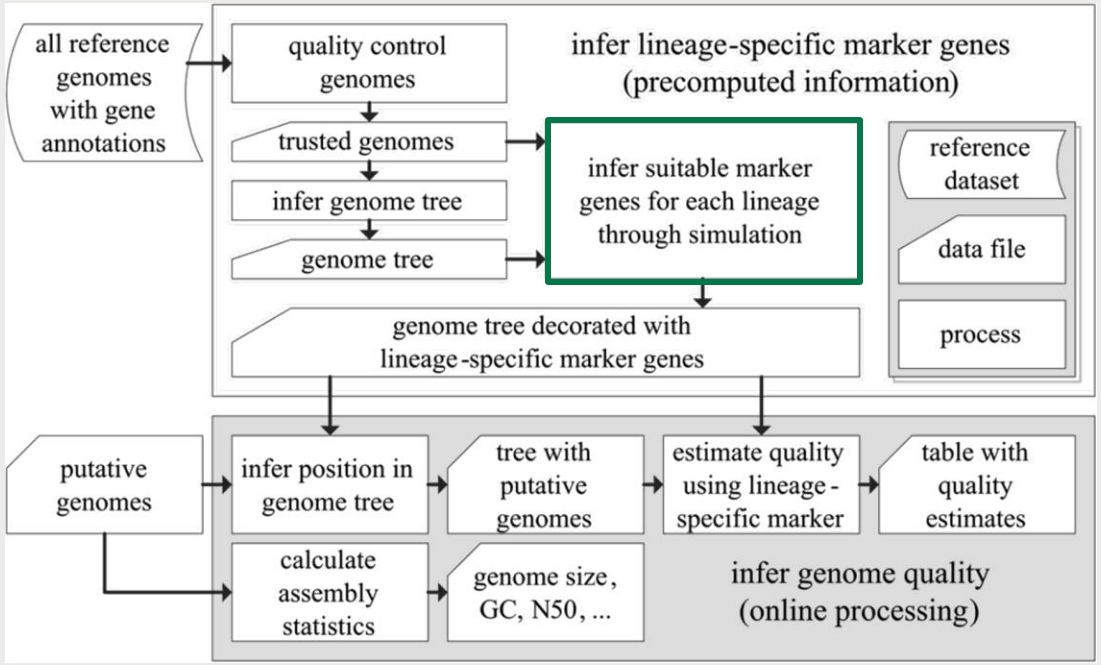
Missing (M): Genes **not found**, indicating incomplete regions

Manni et al. (2021). Curr Protoc. 1(12):e323

The Fleming Fund
Regional Grants

CheckM

Designed to assess the quality (**completeness**, **contamination**, and **heterogeneity**) of genome assemblies based on the presence of lineage-specific markers (present in ≥97% of genomes), with a focus on microbial (bacterial and archaeal) and metagenomic assemblies



New **CheckM2** release

Uses a Machine learning (ML) model trained on curated microbial genomes to quickly predict completeness and contamination

Parks et al. (2015). Genome Res. 25(7):1043-55; https://github.com/Ecogenomics/CheckM/; https://github.com/chklovski/CheckM2

The Fleming Fund
Regional Grants

CheckM

```
# Install CheckM

$ conda install -c bioconda checkm-genome

# Download the database

$ wget https://data.ace.uq.edu.au/public/CheckM_databases/checkm_data_2015_01_16.tar.gz

# Run CheckM in your assembly FASTA files

$ checkm lineage_wf -t 20 -f results_checkm.txt --tab_table -x fasta /home/Databases/CheckM_DB/bins <output folder>
```

Galaxy

KBase
PREDICTIVE BIOLOGY

It can automatically **infer the lineage** and evaluate quality of the genome **without requiring pre-selection** of a dataset

Parks et al. (2015). Genome Res. 25(7):1043-55; https://github.com/Ecogenomics/CheckM/; https://github.com/chklovski/CheckM2

The Fleming Fund
Regional Grants

CheckM

A high-quality genome typically has **>90% completeness** and **<5% contamination**

Ideally it should be "0"

Bin Id	Marker lineage	# genomes	# markers	# marker sets	0	1	2	3	4	5	Completeness	Contamination	Strain heterogeneity
Isolate_3_Flye	f__Enterobacteriaceae (UID5139)	119	1169	340	0	1166	2	1	0	0	100.00	0.15	0.00
Isolate_3_Hybracter	f__Enterobacteriaceae (UID5139)	119	1169	340	0	1166	2	1	0	0	100.00	0.15	0.00
Isolate_3_SKESA	f__Enterobacteriaceae (UID5139)	119	1169	340	0	1166	2	1	0	0	100.00	0.15	0.00
Isolate_3_SPADES	f__Enterobacteriaceae (UID5139)	119	1169	340	0	1166	2	1	0	0	100.00	0.15	0.00
Isolate_3_Unicycler	f__Enterobacteriaceae (UID5139)	119	1169	340	0	1166	2	1	0	0	100.00	0.15	0.00

- **Completeness (%)** - percentage of genome that is complete
- **Contamination (%)** - is always calculated based on the number of markers that are detected more than 1 time

- **Strain heterogeneity (SH)** – measures the **sequence variation** (genetic variability) between **duplicated** marker genes in a genome assembly (SH = 0 – identical sequences; SH > 0 – different sequences).

Parks et al. (2015). Genome Res. 25(7):1043-55; https://github.com/Ecogenomics/CheckM/; https://github.com/chklovski/CheckM2

The Fleming Fund
Regional Grants

Let’s take a break ☺

The Fleming Fund
Regional Grants

Thank you

UKaid from the British people

MOTT MACDONALD

DTU

This programme is being funded by the UK Department of Health and Social Care.
The views expressed do not necessarily reflect the UK Government's official policies.