ESCMID

MANAGING INFECTIONS
PROMOTING SCIENCE

ESCMID Online
Education Course

How to reconstruct mobile genetic elements using short and/or long read sequencing

Moving beyond single species outbreaks: the role of mobile genetic elements

Session 2:	**How to accurately reconstruct mobile genetic elements using bioinformatics**
Chairs:	*Andrew Page, Lee Katz, Nabil-Fareed Alikhan (Micro Binfies)*
13:30 – 16:00	**How to reconstruct mobile genetic elements using short and/or long read sequencing**
16:00 – 17:00	**Electronic break-out rooms**
17:00	**End of day**

Micro Binfie Podcast

https://soundcloud.com/microbinfie/

@microbinfie

A podcast about microbial genomics and bioinformatics

This session's agenda

13:30 Overview &
Long and short read de novo genome assembly

14:15 Bioinformatics tools for MGE

14:45 Pangenomes and MGE

A quick word about phylogenetic trees

15:30 Plasmids in a Public Health Space

16:00 Q&A panel discussion

17:00 End of Day

Dr Andrew Page
Head of Informatics
@andrewjpage

Dr Lee Katz
Senior Bioinformatician
@lskatz

Dr Nabil-Fareed Alikhan
Deputy Head of Bioinformatics
@happy_khan

Dr Hattie Webb
Research Scientist
@hattie_webb

Dr Kaitlin Tagg
Research Scientist
@_ktagg

This session's structure

Goal: Retrieve my MGE of interest

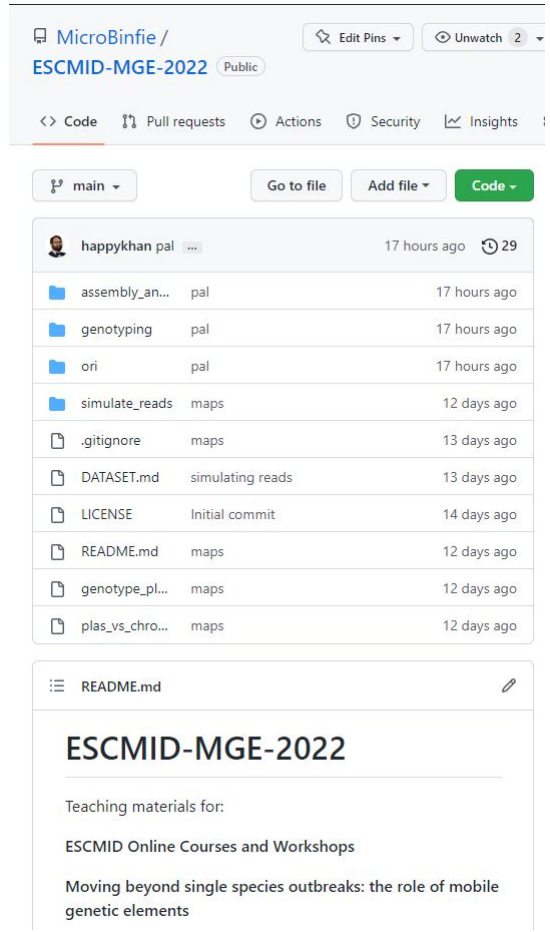
Locate existing elements via genotypes

Locate new elements via pangenome

Construct a good genome assembly

INPUT: Sequenced long/short reads

Session handout (github)



https://github.com/MicroBinfie/ESCMID-MGE-2022

SCAN ME

https://tinyurl.com/2022mge

ESCMID EUROPEAN SOCIETY OF CLINICAL MICROBIOLOGY AND INFECTIOUS DISEASES

A warning

- You can not be in expert in three hours.
- We aim to provide a general workflow which can inspire you.

Some tools we missed:

- Other genome assemblers
- PanACoTA (Rocha)
- PIRATE (Feil)

An admission

- We aim to provide worked examples to help you get started but we cannot be comprehensive
- We chose simpler accessible tools over more complex cutting edge ones.

Session handout (github)

https://github.com/MicroBinfie/ESCMID-MGE-2022

https://tinyurl.com/2022mge

SCAN ME

ESCMID EUROPEAN SOCIETY OF CLINICAL MICROBIOLOGY AND INFECTIOUS DISEASES

Quadram Institute
Science Health Food Innovation

Long and short read *de novo* genome assembly

Dr Nabil-Fareed Alikhan

Quadram Institute
Science Health Food Innovation

Deputy Head of Bioinformatics

@happy_khan

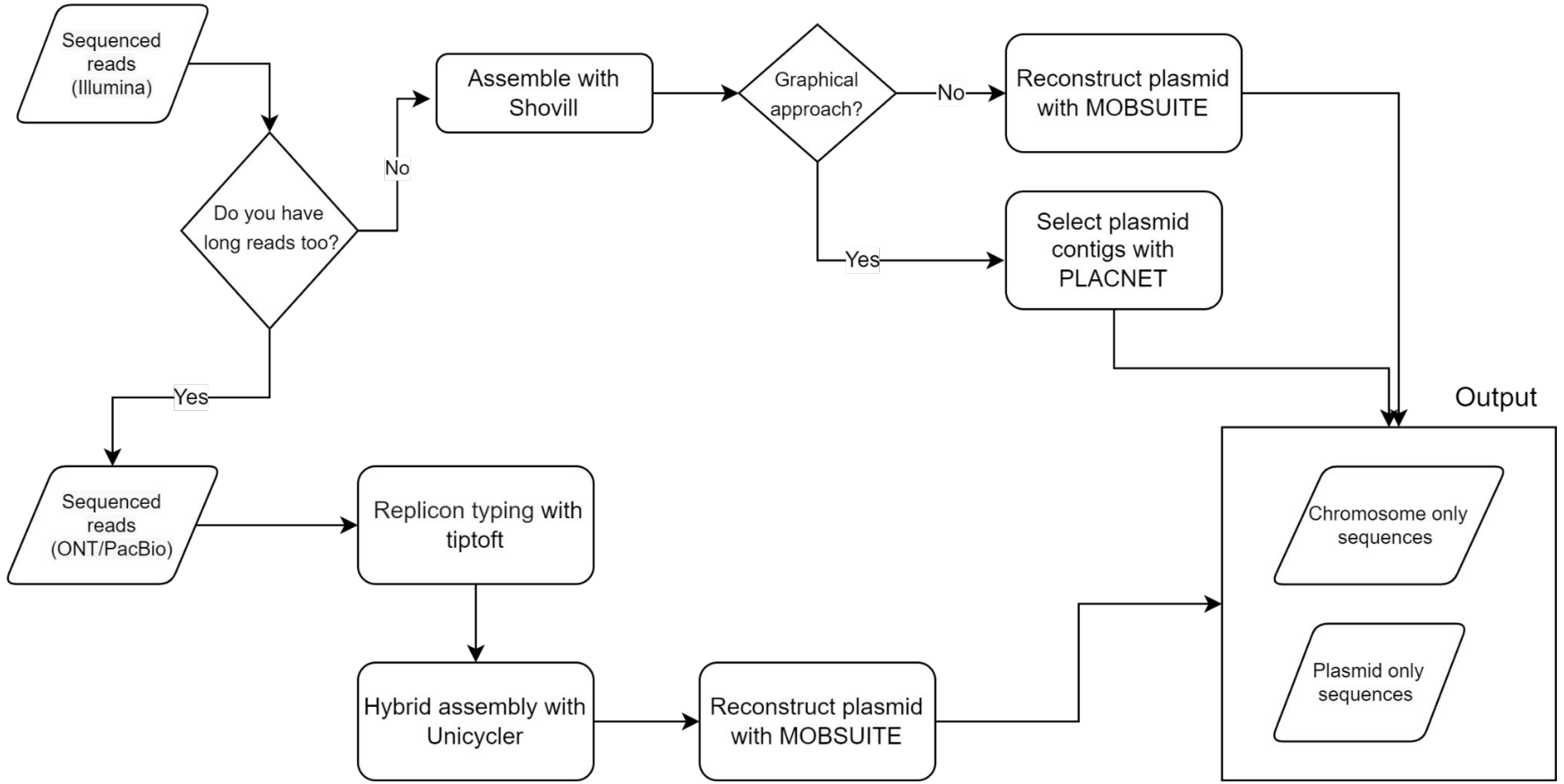
nabil-fareed.alikhan@quadram.ac.uk

Important to understand which genomic regions are plasmid vs chromosome

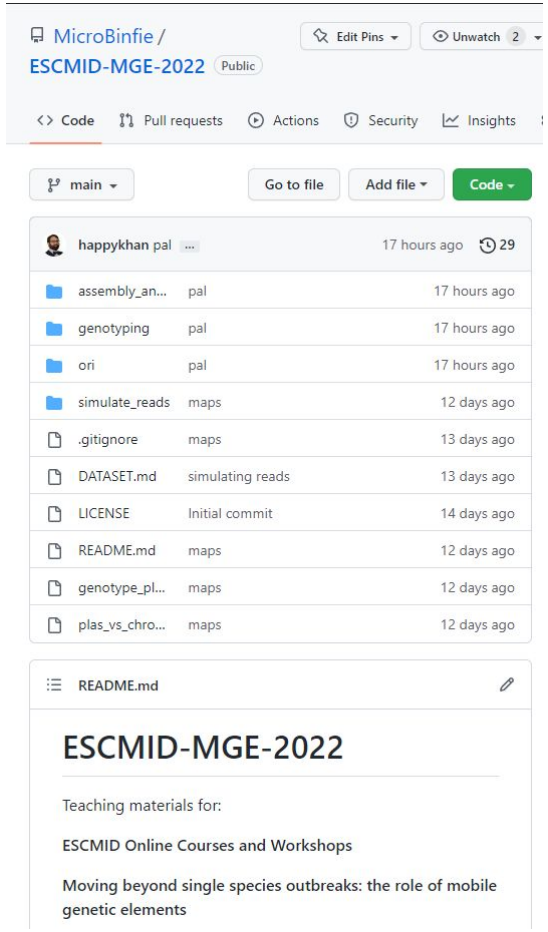
- Sanity check
- How is gene of interest mobilised
- Context for pangenome for novel MGE.
- Starts complex analysis (ICE/Plasmid)

Identifying plasmid sequence in genome assemblies

Through this process you should be able to separate your assembled contigs into plasmid and chromosome.



Again: Session handout (github)

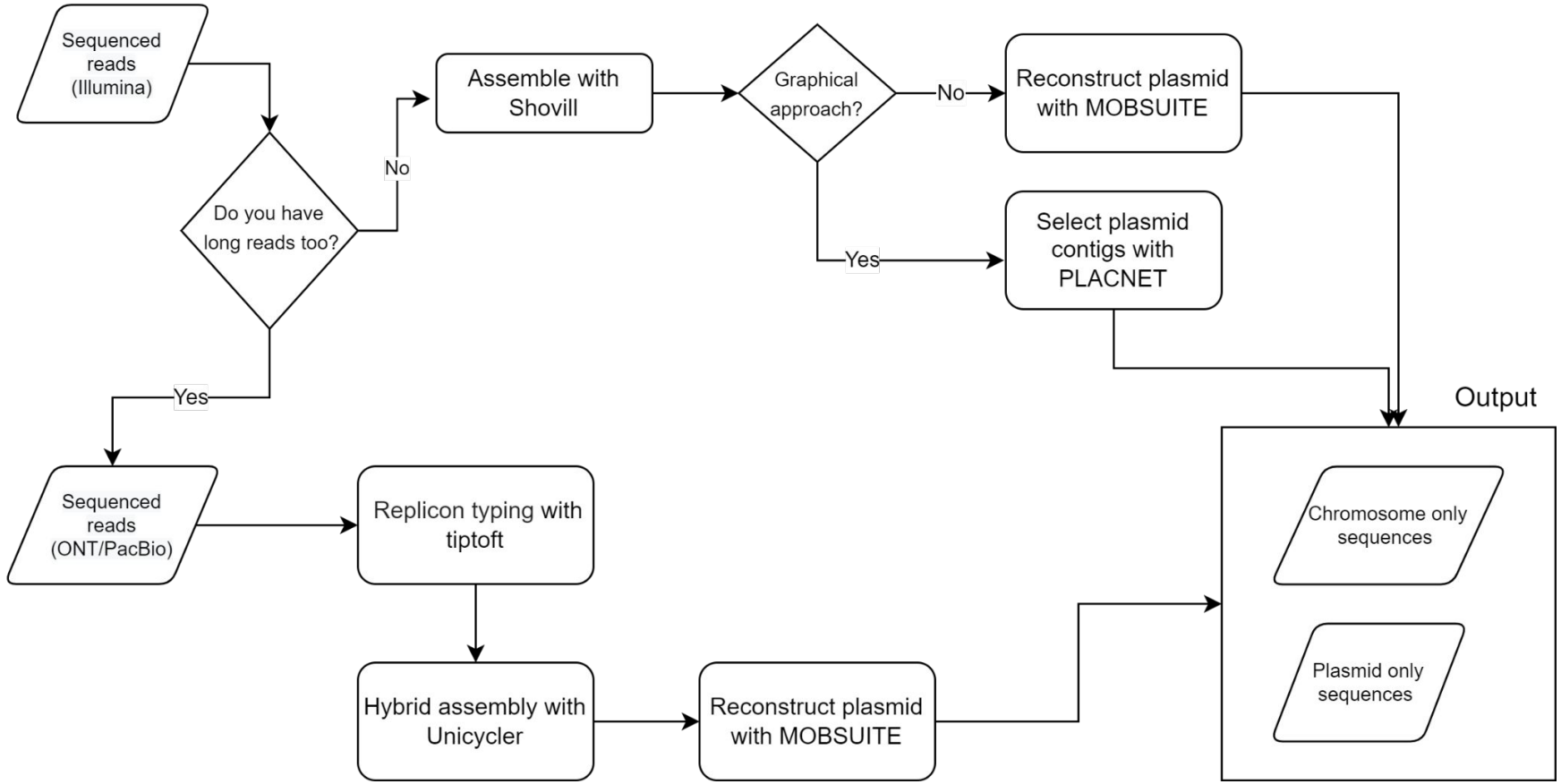


https://github.com/MicroBinfie/ESCMID-MGE-2022

https://tinyurl.com/2022mge

Identifying plasmid sequence in genome assemblies

Through this process you should be able to separate your assembled contigs into plasmid and chromosome.



Agenda: Work through this flowchart, and compare long vs short reads

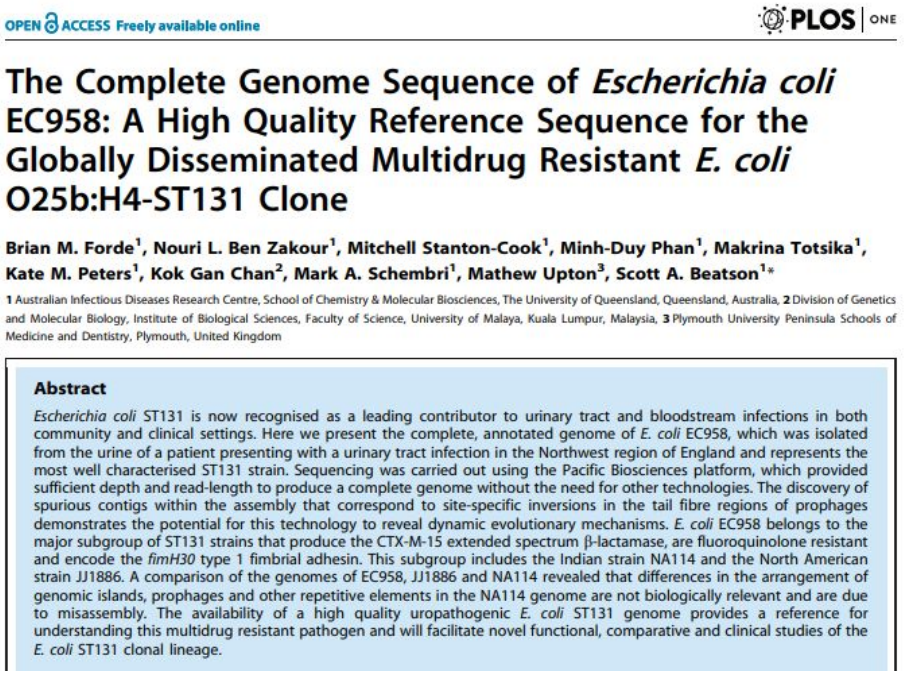
Example data

In the handouts and during this talk, worked examples use an *E. coli* ST 131 (EC958)

- Multidrug resistant, genes encoded on a plasmid
- Interesting genome islands
- Multiple prophage

Simulated long (Oxford nanopore) and short read (illumina paired-end)

Walkthrough of how to generate simulate data, see:
Handout > simulate_reads > simulate_reads.ipynb



OPEN ACCESS Freely available online PLOS ONE

The Complete Genome Sequence of *Escherichia coli* EC958: A High Quality Reference Sequence for the Globally Disseminated Multidrug Resistant *E. coli* O25b:H4-ST131 Clone

Brian M. Forde[1], Nouri L. Ben Zakour[1], Mitchell Stanton-Cook[1], Minh-Duy Phan[1], Makrina Totsika[1], Kate M. Peters[1], Kok Gan Chan[2], Mark A. Schembri[1], Mathew Upton[3], Scott A. Beatson[1]*

1 Australian Infectious Diseases Research Centre, School of Chemistry & Molecular Biosciences, The University of Queensland, Queensland, Australia, **2** Division of Genetics and Molecular Biology, Institute of Biological Sciences, Faculty of Science, University of Malaya, Kuala Lumpur, Malaysia, **3** Plymouth University Peninsula Schools of Medicine and Dentistry, Plymouth, United Kingdom

Abstract

Escherichia coli ST131 is now recognised as a leading contributor to urinary tract and bloodstream infections in both community and clinical settings. Here we present the complete, annotated genome of *E. coli* EC958, which was isolated from the urine of a patient presenting with a urinary tract infection in the Northwest region of England and represents the most well characterised ST131 strain. Sequencing was carried out using the Pacific Biosciences platform, which provided sufficient depth and read-length to produce a complete genome without the need for other technologies. The discovery of spurious contigs within the assembly that correspond to site-specific inversions in the tail fibre regions of prophages demonstrates the potential for this technology to reveal dynamic evolutionary mechanisms. *E. coli* EC958 belongs to the major subgroup of ST131 strains that produce the CTX-M-15 extended spectrum β-lactamase, are fluoroquinolone resistant and encode the *fimH30* type 1 fimbrial adhesin. This subgroup includes the Indian strain NA114 and the North American strain JJ1886. A comparison of the genomes of EC958, JJ1886 and NA114 revealed that differences in the arrangement of genomic islands, prophages and other repetitive elements in the NA114 genome are not biologically relevant and are due to misassembly. The availability of a high quality uropathogenic *E. coli* ST131 genome provides a reference for understanding this multidrug resistant pathogen and will facilitate novel functional, comparative and clinical studies of the *E. coli* ST131 clonal lineage.

https://doi.org/10.1371/journal.pone.0104400

Handout link: https//tinyurl.com/2022mge

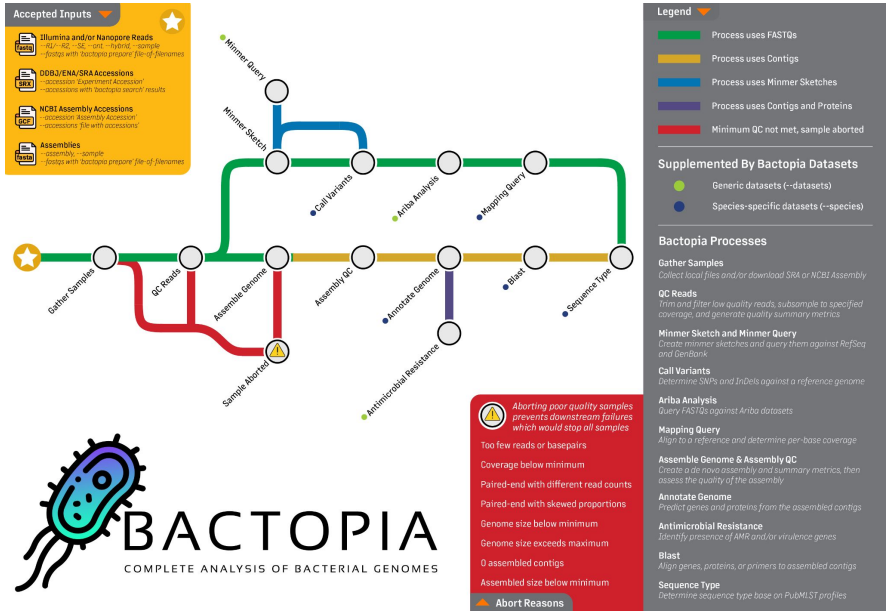
Working with bioinformatics software

A quick aside, how should I organise my projects?

- Use conda for package management (https://docs.conda.io/en/latest/)
- Jupyter notebooks for exploring data and plotting figures (https://jupyter.org/)

Processing large datasets

- Use workflow languages (nextflow)
- Bactopia is a good all-included workflow to start

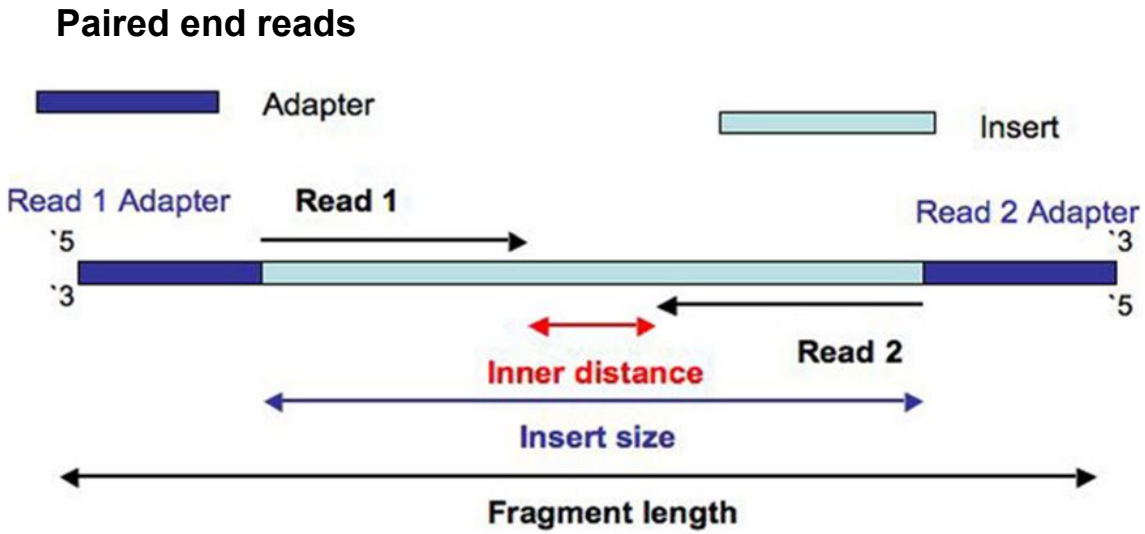


https://bactopia.github.io/

Handout link: https//tinyurl.com/2022mge

What is a sequencer doing?

- Shear target DNA
- Prepare a sequencing library
- Read with sequencing machine
- Machines have a maximum read length
  - Illumina: hundreds of bp.
  - Long read: Kilo to Megabases



https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

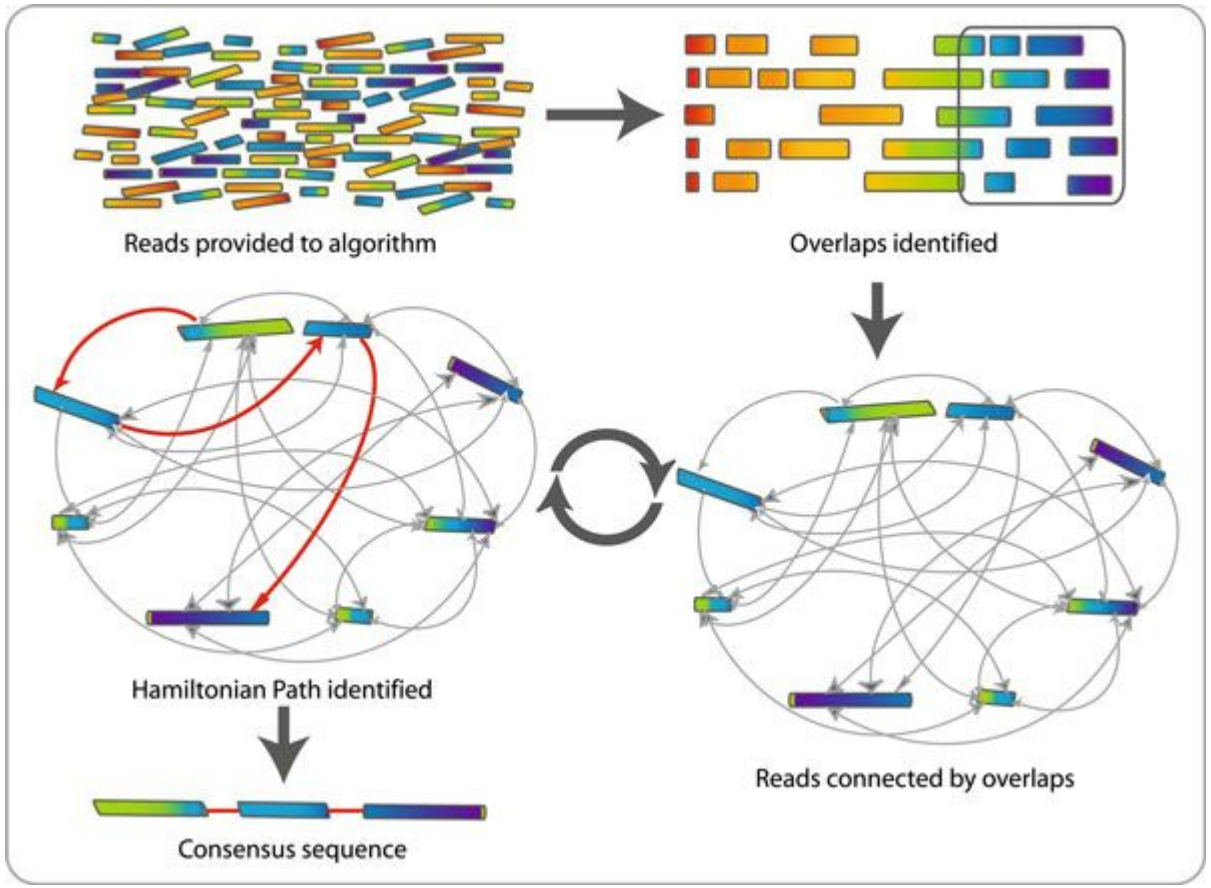
What is a genome assembler doing?

Genome assemblers assume the following about sequenced reads:

- Reads are resolved into nucleotide bases (ATGC & ambiguous base calls)
- Reads are randomly distributed across the target DNA, and
- Reads are represent an oversampling of the target DNA, such that individual reads repeatedly overlap

Genome assemblers calculate overlaps between reads and (usually) represent as a graph/network. Then “walk” the graph to determine the original sequence.

See Torsten Seemann’s slides on de novo genome assembly: https://tinyurl.com/torstaseembler



http://dx.doi.org/10.1007/s12575-009-9004-1

https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

How genome assemblers fail perfection

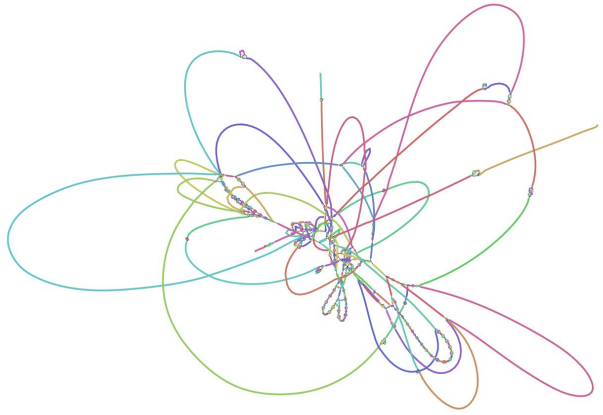
We need to pay attention as genome assemblers struggle with repetitive elements, which link to MGE

In theory, Genome assembly software with perfect reads of good length will reconstruct the genome verbatim

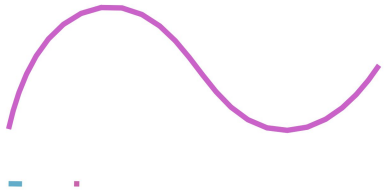
Sequencing platform errors (causes error downstream):

- Struggle with GC rich and/or AT rich DNA.
- Have lower read quality towards the end of reads (5', 3' or both ends)
- Have difficulty reading homopolymers (e.g. AAAAA or TTTTTTT) accurately
- **Have read lengths that does not span repeated sequences in the genome**

Fragmented genome assembly



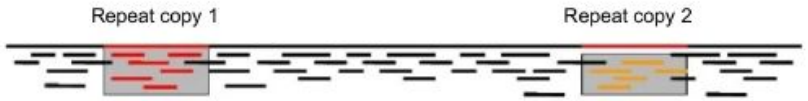
Perfect

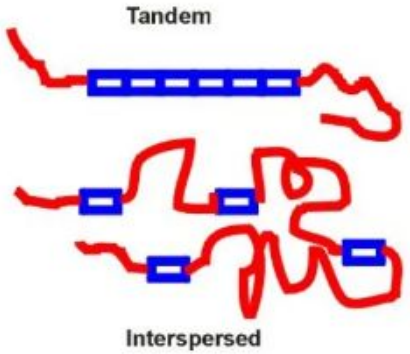


https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

Repeats and read length

- Repeats: A segment of DNA that occurs more than once in the genome
- Read length must span the repeat (remember CRISPR)

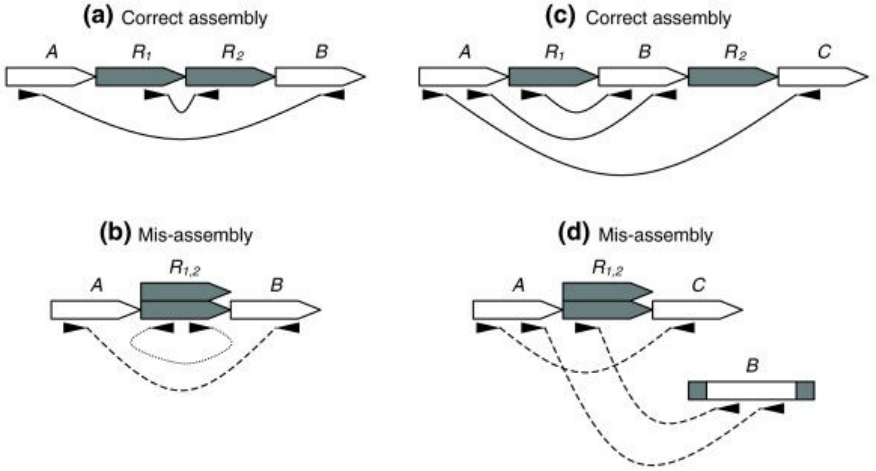




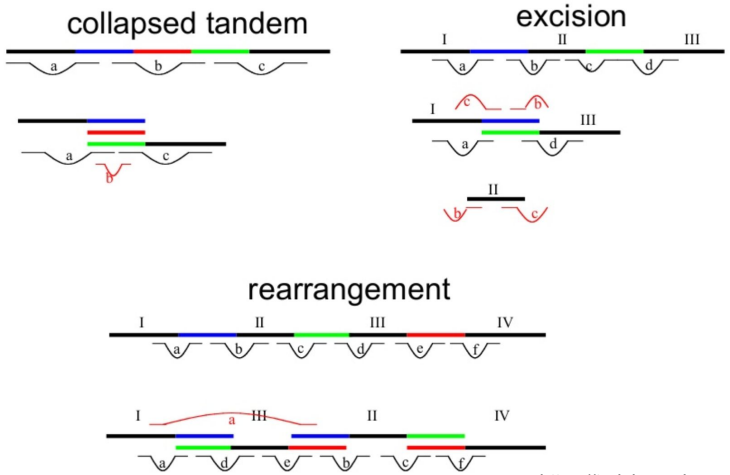
How to span repeats

- Long reads (ONT, Pacbio)
- Long reads (Capillary)
- Optical mapping
- Hi-C
- Or just don't!

Outcomes in your final contigs



https://doi.org/10.1186%2Fgb-2008-9-3-r55

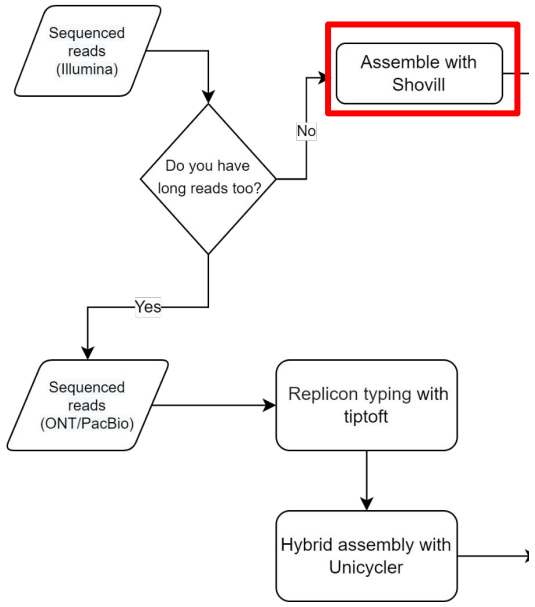


https://training.galaxyproject.org/training-material/topics/assembly/tutorials/get-started-genome-assembly/slides-plain.html

https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

Short read assembly with Shovill

Let's talk about the impact of read length (long read vs short read) on genome assembly.
In doing so I will also show you how to run the software and assess the output.



We can run Shovill with:

```
shovill --R1 seq/MGE-2022_1.fastq.gz  --R2  seq/MGE-2022_2.fastq.gz --outdir ass/shovill --force
```

Output

10-seqtk.tab	nput_dataset.yaml	spades.fasta
20-kmc.log	kmc.kmc_pre	spades.log
40-lighter.log	kmc.kmc_suf	
50-flash.log	params.txt	
contigs.fa	shovill.corrections	
contigs.gfa	**shovill.log**	

SAVE THIS!

- **contigs.fa:** The final assembly you should use
- **contigs.gfa:** Assembly graph (spades)
- **shovill.log:** Full log file for bug reporting

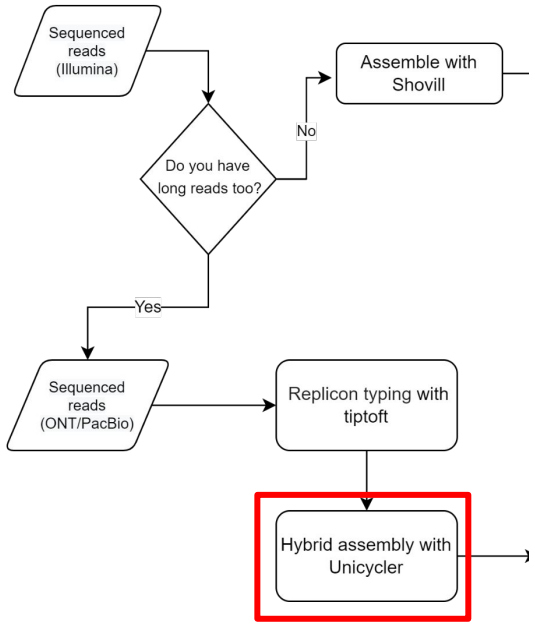
Why Shovill?

- Pipeline wrapped around SPAdes (and other) assemblers
- Have yet to encounter major mis-assemblies
- Easy install
- Includes quality control (read trimming)

Hybrid (long + short) read assembly with Unicycler

How to run:

```
unicycler -1 seq/MGE-2022_1.fastq.gz  -2 seq/MGE-2022_2.fastq.gz -l seq/MGE-2022_ONT.fastq.gz -o ass/unicycler
```



Output

```
001_spades_graph_k027.gfa      002_depth_filter.gfa
001_spades_graph_k053.gfa      003_overlaps_removed.gfa
001_spades_graph_k071.gfa      004_long_read_assembly.gfa
001_spades_graph_k087.gfa      005_bridges_applied.gfa
001_spades_graph_k099.gfa      006_final_clean.gfa
001_spades_graph_k111.gfa      assembly.fasta
001_spades_graph_k119.gfa      assembly.gfa
001_spades_graph_k127.gfa      unicycler.log
```

SAVE THIS!

- **assembly.fasta:** The final assembly you should use
- **assembly.gfa:** Assembly graph
- **unicycler.log:** Full log file for bug reporting

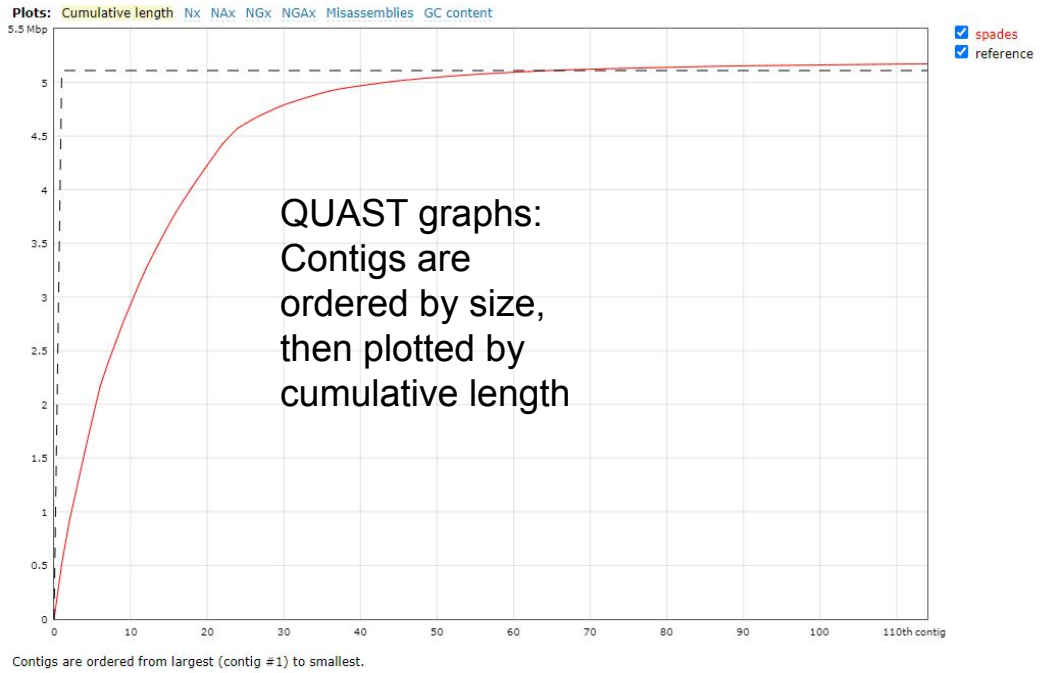
Why Unicycler?

- All in one package - Good hybrid assembly
- Can use short read only and bridge gaps with some guesswork
- A lot of options on how aggressive you want it to join contigs
- Easy install
- Checks for circularisation

Assessing genome assembly quality

Contiguity

- Less contigs, Longer contigs (see QUAST plot below)
- N50, avg. contig length, number of contigs etc.



Completeness

- Compare to reference genome (Mauve; later)
- Assume a genome should have single copy essential genes
  - MLST intact?
  - BUSCO panel
  - CheckM panel

https://github.com/tseemann/mlst

https://busco.ezlab.org/

https://ecogenomics.github.io/CheckM

Correctness

- Assembly free from errors
  - Mis-joins
  - Collapsed repeats
  - Duplication artefacts
  - False SNPs, InDels
- Compare to reference genome
- Map original reads back to assembled contigs
- Structural rearrangement tools - Socru

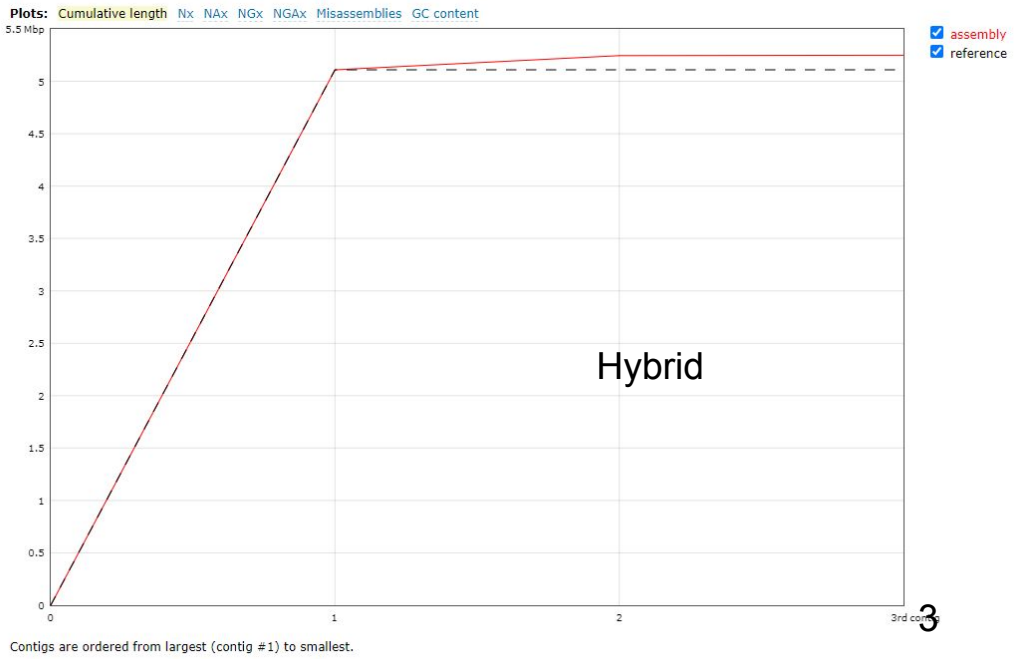
https://github.com/quadram-institute-bioscience/socru

Check for contamination too! Kraken/Bracken

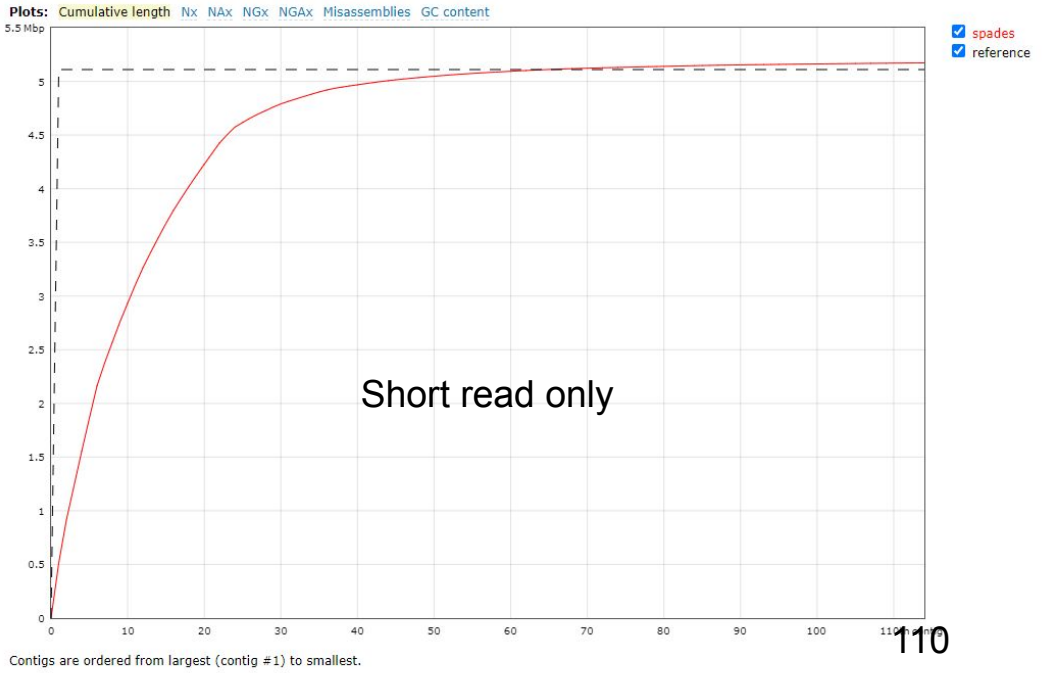
https://ccb.jhu.edu/software/bracken/

Comparing our assemblies - Contiguity

Metric	Hybrid assembly	Short read only assembly
Contigs #	3	240
total assembled length	5249059	5204624
Reference genome (chr) recovery	99.997%	98.629%



3

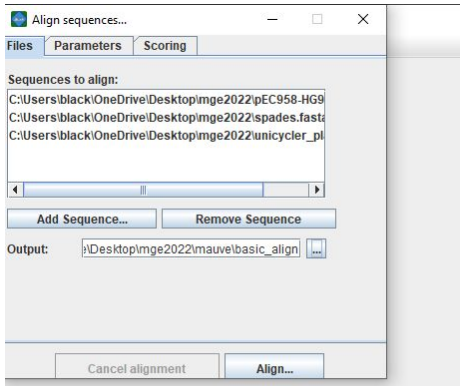


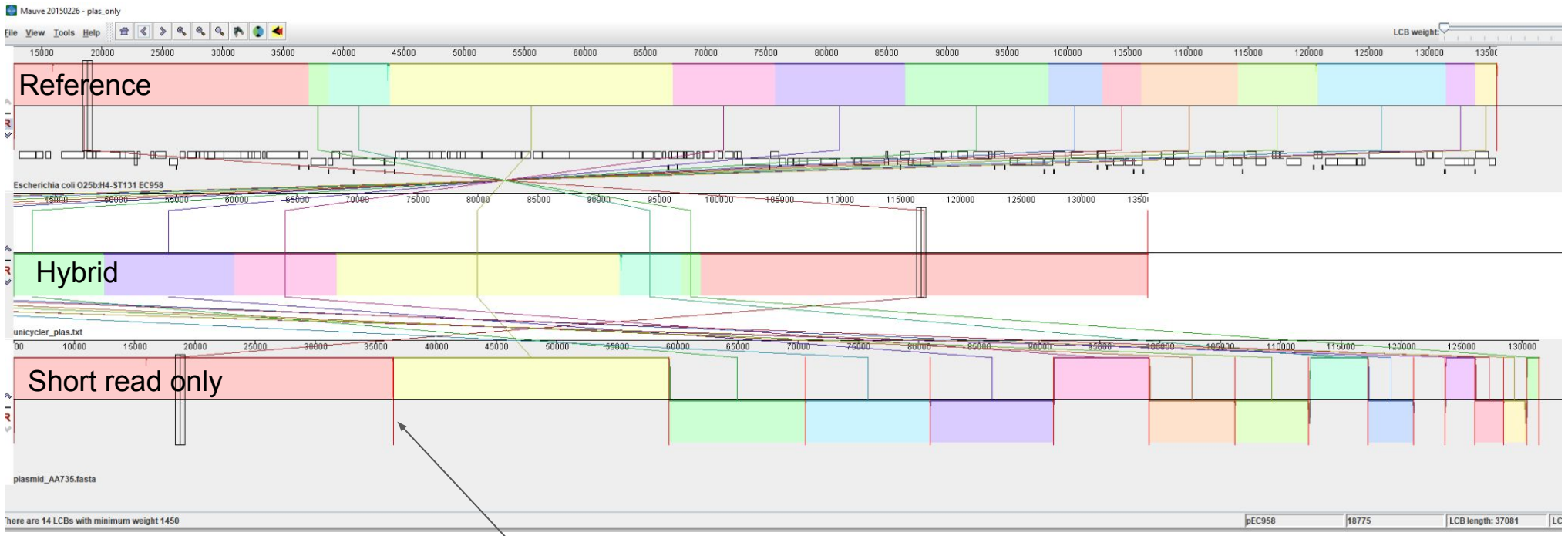
110

https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

Comparing our example data visually with Mauve

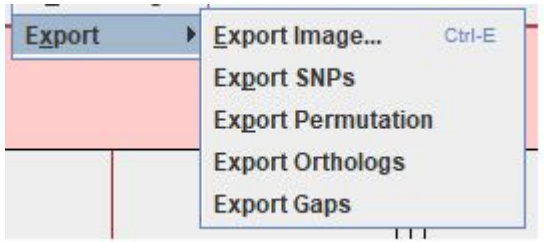
Input





Contig break

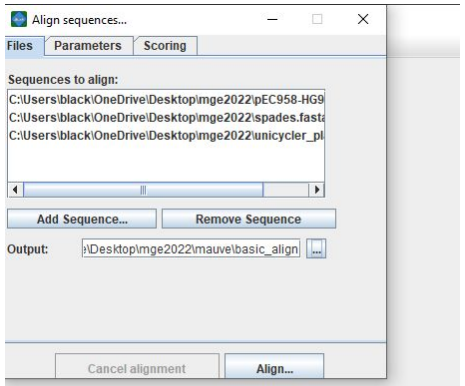
Extra output:

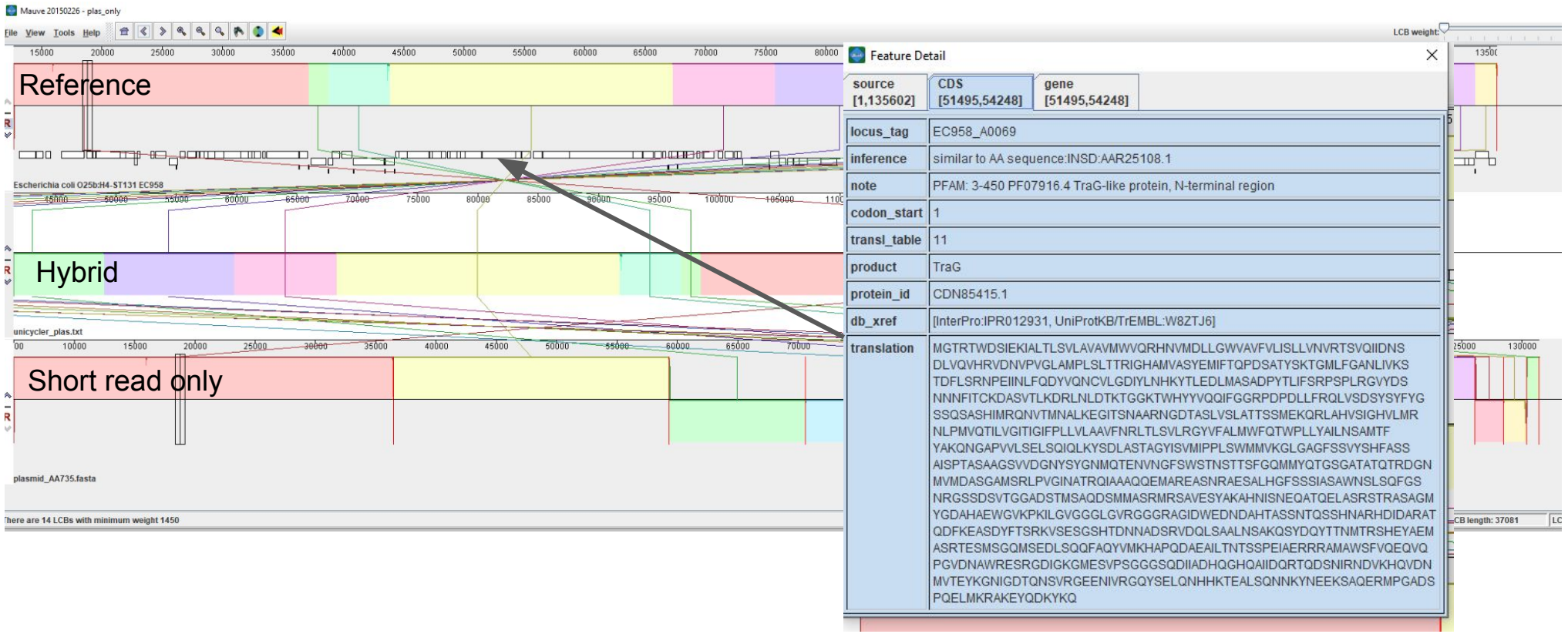


https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

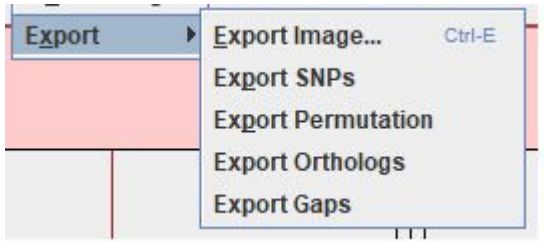
Comparing our example data visually with Mauve

Input





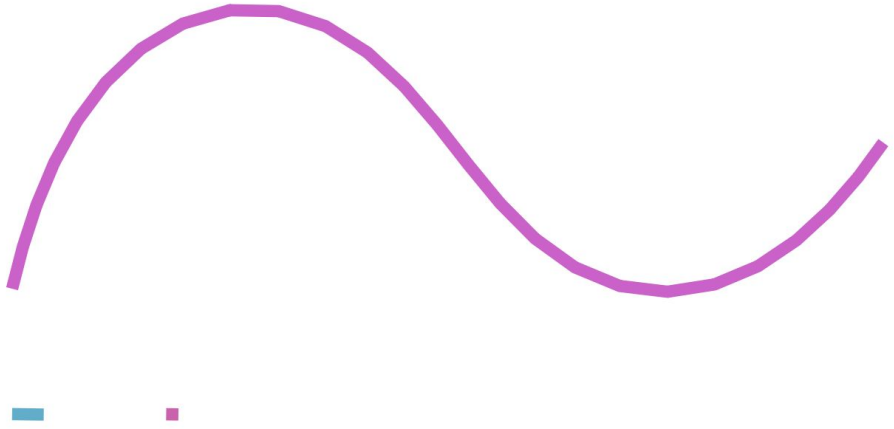
Extra output:



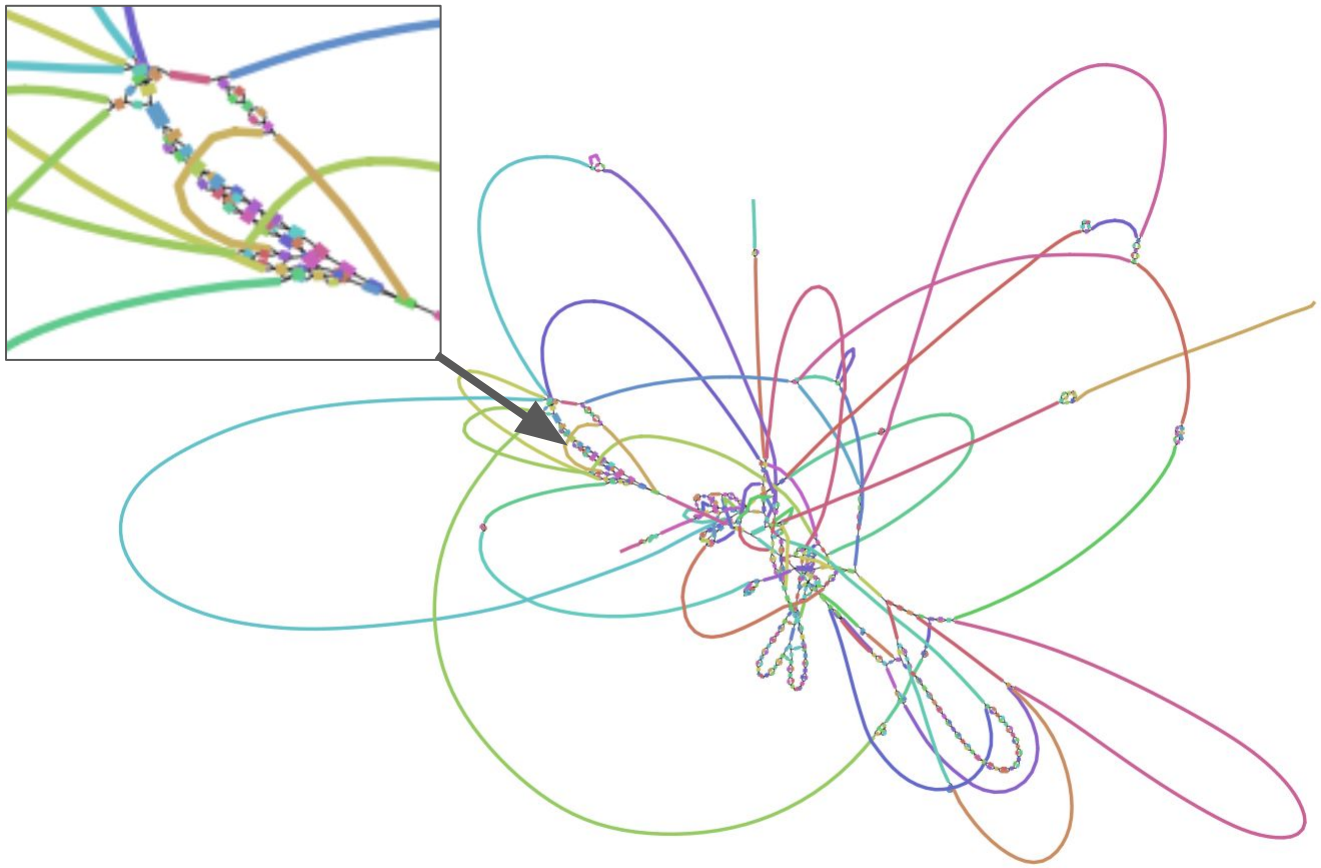
https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

Exploring genome graphs with Bandage

https://rrwick.github.io/Bandage/



Hybrid



Short read only

https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

Exploring genome graphs with Placnet

For each genome, upload the Illumina read files in the slots below, one for each fastq gzip-compressed read file.

As soon as your data is uploaded, we will provide you a link to the online processing tool. The data is expected to be ready in a few hours (you can refresh the link and check) and will remain accessible for one week.

Id of the job: MGEExample

Read file 1 (.gz) Choose File MGE-2022_1.fastq.gz

Read file 2 (.gz) Choose File MGE-2022_2.fastq.gz

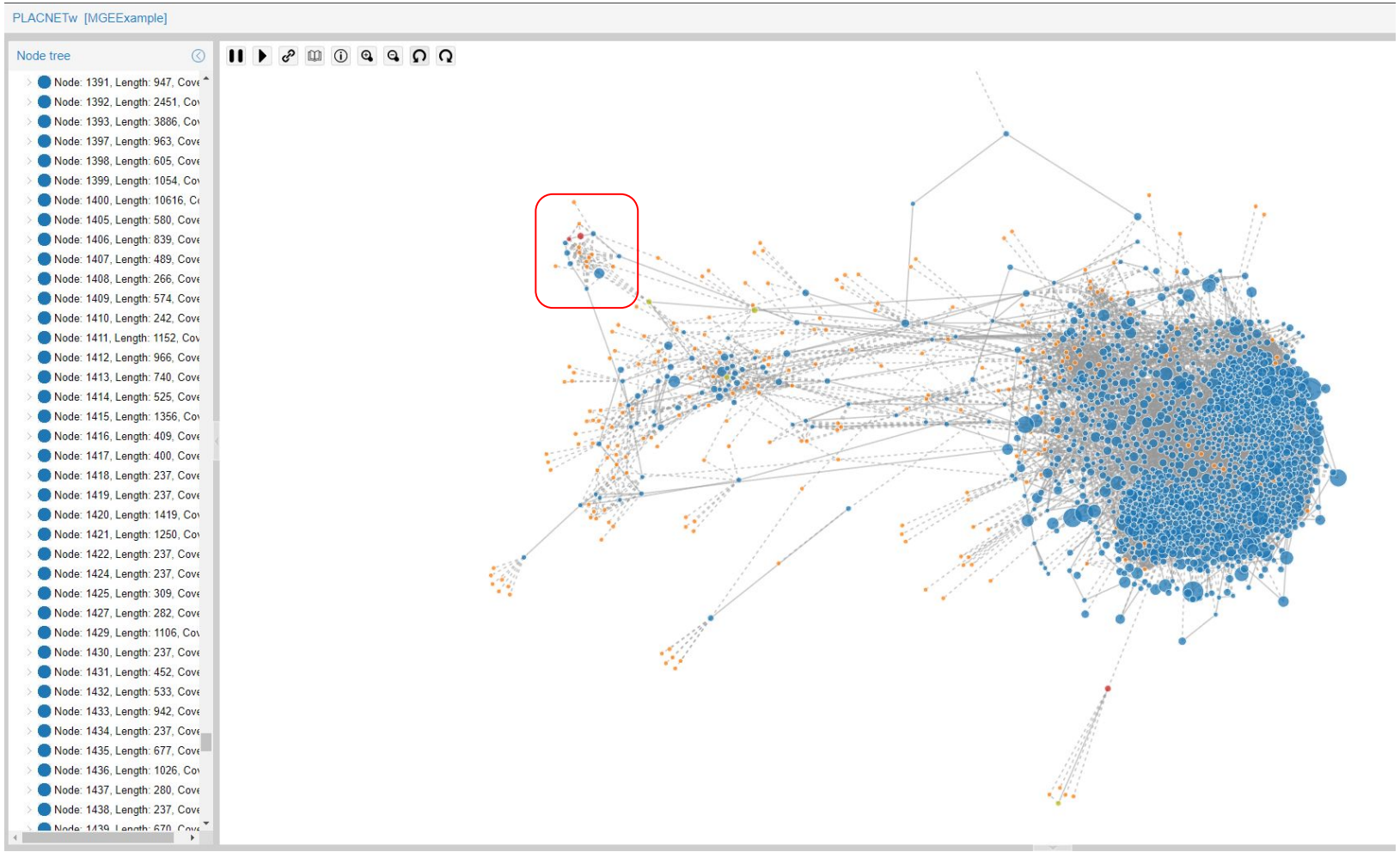
Upload

10%

Please cite as: Luis Vielva, Maria de Toro, Val F Lanza, Fernando de la Cruz, PLACNETw: a web-based tool for plasmid reconstruction from bacterial genomes, Bioinformatics, Volume 33, Issue 23, 01 December 2017, Pages 3796-3798, https://doi.org/10.1093/bioinformatics/btx462

Bibtex Ris Enw

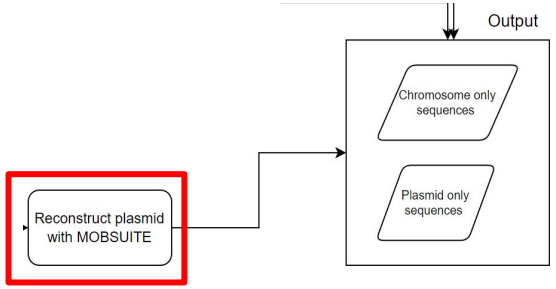
Tutorial video Source code Examples



https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/assembly_annotation/comparing_long_short_assem.ipynb

Plasmid reconstruction with MOB-Suite

Want a GUI web based option? - see COPLA
https://castillo.dicom.unican.es/copla/



What does it do?

- Reference database approach
- Identifying contigs of plasmid origin
- Aggregates the plasmid contigs into groups based on an internal clustering scheme.

How to run

We can run it on our sample long read assembly as follows:

```
mob_recon -u  --infile ../ori/sample_unicycler_assembly.fasta  --outdir mobrecon_unicycler
```

Output - Everything this program produces is useful

- **contig_report.txt** This file describes the assignment of the contig to chromosome or a particular plasmid grouping
- **mge.report.txt** Blast HSP of detected MGE's/repetitive elements with contextual information
- **chromosome.fasta** Fasta file of all contigs found to belong to the chromosome
- **plasmid_(X).fasta** Each plasmid group is written to an individual fasta file which contains the assigned contigs
- **mobtyper_results** Aggregate MOB-typer report files for all identified plasmid

Closely related plasmids!

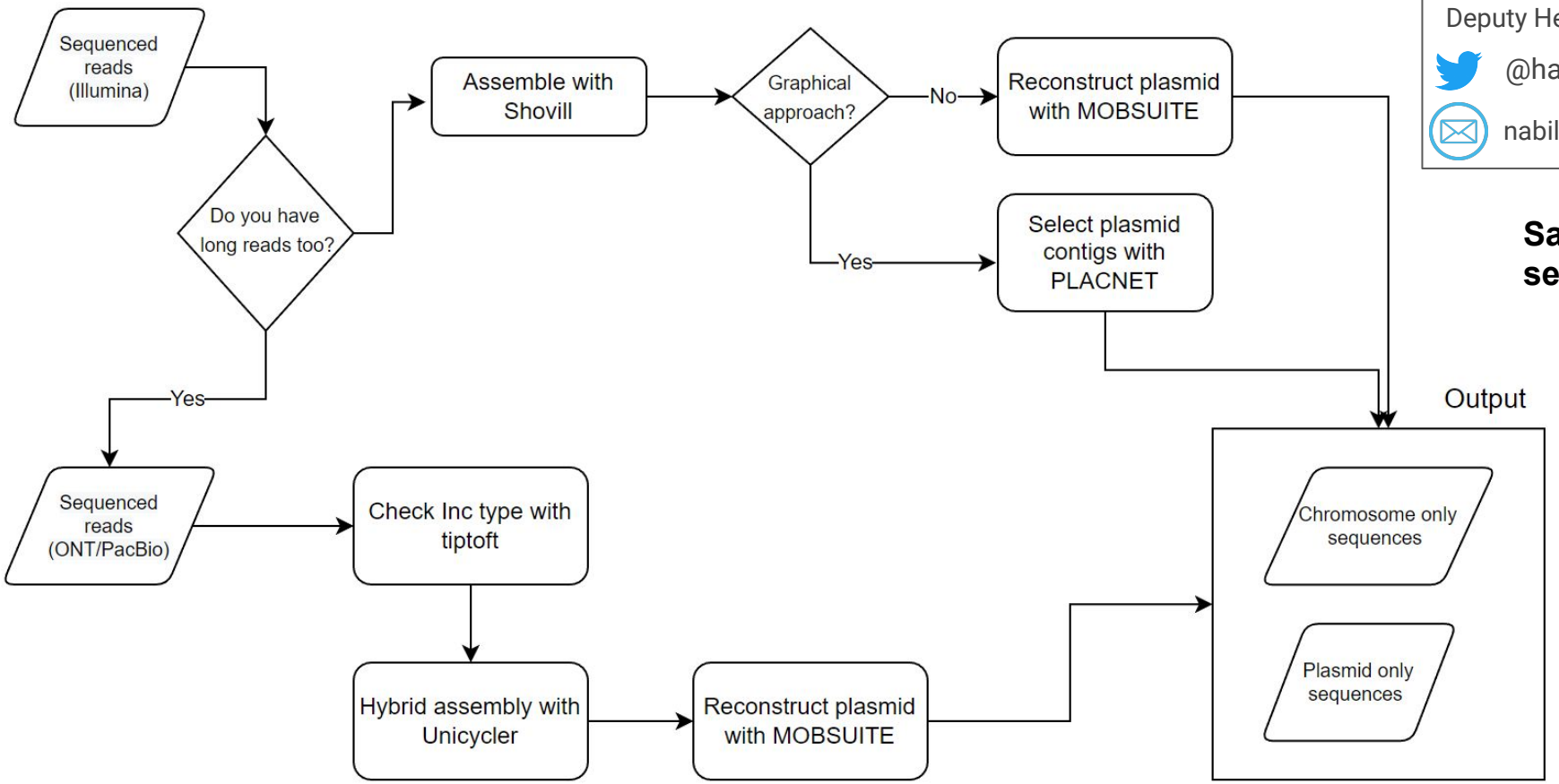
molecule_type	contig_id	size	rep_type(s)	relaxase_type(s)	mpf_type
chromosome	1 length=5109618 depth=1.00x	5109618	-	MOBP	-
plasmid	2 length=135479 depth=1.00x	135479	IncFIA,IncFIC,IncFII	MOBF	MPF_F,MPF_F,MPF_F,M
plasmid	3 length=3962 depth=1.02x	3962	rep_cluster_1778	MOBQ	-

https://github.com/MicroBinfie/ESCMID-MGE-2022/blob/main/genotyping/plasmid_reconstruction_mobsuite.ipynb

Summing up

Identifying plasmid sequence in genome assemblies

Through this process you should be able to separate your assembled contigs into plasmid and chromosome.



Dr Nabil-Fareed Alikhan

Quadram Institute Science • Health • Food • Innovation

Deputy Head of Bioinformatics

@happy_khan

nabil-fareed.alikhan@quadram.ac.uk

Save questions for the end of the session, or @ me on twitter.

This session's structure

Goal: Retrieve my MGE of interest

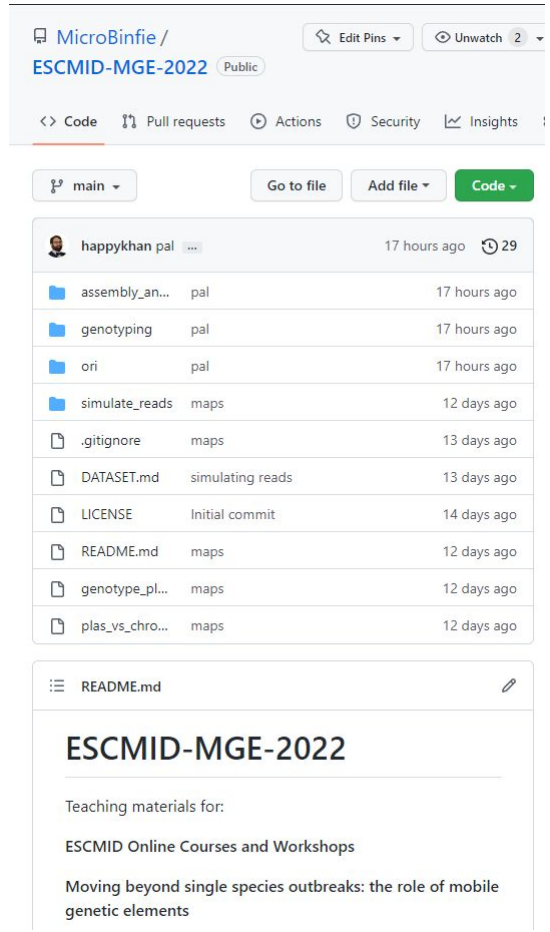

Locate existing elements via genotypes

Locate new elements via pangenome

Construct a good genome assembly

INPUT: Sequenced long/short reads

Session handout (github)



https://github.com/MicroBinfie/ESCMID-MGE-2022

SCAN ME

https://tinyurl.com/2022mge

ESCMID EUROPEAN SOCIETY OF CLINICAL MICROBIOLOGY AND INFECTIOUS DISEASES