Hybrid genome assembly - nanopore and illumina (1hr)

Anticipated workshop duration is **1 hour**.

For queries relating to this workshop, contact Melbourne Bioinformatics (bioinformatics-training@unimelb.edu.au).

Overview

Topic

- [x] Genomics
- [ ] Transcriptomics
- [ ] Proteomics
- [ ] Metabolomics
- [ ] Statistics and visualisation
- [ ] Structural Modelling
- [x] Basic skills

Skill level

- [x] Beginner
- [ ] Intermediate
- [ ] Advanced

This workshop is designed for participants with no command line knowledge. A web-based platform called Galaxy will be used to run our analysis.

How do long- and short-read assembly methods differ?

Description

Assemble a genome!
Learn how to create and assess genome assemblies using the powerful combination of nanopore and illumina reads

This tutorial explores how long and short read data can be combined to produce a high-quality 'finished' bacterial genome sequence. Termed 'hybrid assembly', we will use read data produced from two different

sequencing platforms, Illumina (short read) and Oxford Nanopore Technologies (long read), to reconstruct a bacterial genome sequence.

In this tutorial we will perform '*de novo* assembly'. De novo assembly is the process of assembling a genome from scratch using only the sequenced reads as input - no reference genome is used. This approach is common practise when working with microorganisms, and has seen increasing use for eukaryotes (including humans) in recent times.

Using short read data (Illumina) alone for *de novo* assembly will produce a complete genome, but often in many pieces (commonly called a 'draft genome'). For the genome to be assembled into a single chromosome (plus a sequence for each plasmid), reads would need to be longer than the longest repeated element on the genome (usually ~7,000 base pairs, Note: Illumina reads are 350 base maximum). Draft bacterial genome sequences are cheap to produce (less than AUD$60) and useful (>300,000 draft *Salmonella enterica* genome sequences published at NCBI https://www.ncbi.nlm.nih.gov/pathogens/organisms/), but sometimes you need a high-quality 'finished' bacterial genome sequence. There are <1,000 are 'finished' or 'closed' *Salmonella enterica* genome sequences.

In these cases, long reads can be used together with short reads to produce a high-quality assembly. Nanopore long reads (commonly >40,000 bases) can fully span repeats, and reveal how all the genome fragments should be arranged. Long reads currently have higher error rate than short reads, so the combination of technologies is particularly powerful. Long reads provide information on the genome structure, and short reads provide high base-level accuracy.

Combining read data from the long and short read sequencing platforms allows the production of a complete genome sequence with very few sequence errors, but the cost of the read data is about AUD$ 1,000 to produce the sequence. Understanably, we usually produce a draft genome sequence with very few sequence errors using the Illumina sequencing platform.

Nanopore sequencing technology is rapidly improving, expect the cost difference to reduce!!

Data: Nanopore reads, Illlumina reads, bacterial organism (*Bacillus subtilis*) reference genome
Tools: Unicycler, Quast, BUSCO
Pipeline: FastQC, MultiQC, NanoPlot, Unicycler, Quast, BUSCO

Learning Objectives

At the end of this introductory workshop, you will:

- Understand how Nanopore and Illumina reads can be used together to produce a high quality assembly
- Be familiar with genome assembly and polishing programs
- Learn how to assess the quality of a genome assembly, regardless of whether a reference genome is present or absent
- Be able to assemble an unknown, previously undocumented genome to high-quality using Nanopore and Illumina reads.

Requirements and preparation

Attendees are required to bring their own laptop computers.

All data and tools are available on usegalaxy.org.au. You will need a computer to connect to and use their platform. Before the tutorial, navigate to https://usegalaxy.org.au/ and use your email to create an account. Click "Login or register" in the top navigation bar of galaxy to do this.

Preparing your laptop prior to starting this workshop

- No additional software needs to be installed for this workshop.

Required Data

- No additional data needs to be downloaded for this workshop.

Author Information

Written by: Grace Hall
Melbourne Bioinformatics, The University of Melbourne

Created/Reviewed: March 2020

Background

How do we produce the genomic DNA for a bacterial isolate?

Traditional *in vitro* culture techniques are important. Take a sample (e.g. a swab specimen from an infected sore) and streak a 'loopful' on to solid growth medium that suppoprts the growth of the bacteria. **Technology from the time of Louis Pasteur!**

Mixtures of bacterial types can be sequenced e.g. prepare genomic DNA from environmental samples containing bacteria - water, soil, faecal samples etc. (Whole Metagenome Sequencing)

One colony contains 10^7 – 10^8 cells. The genomic DNA extracted from one colony is enough for Illumina sequencing. Larger amounts of genomic DNA are required for Nanopore sequencing.

Shotgun sequencing - Illumina Sequencing Library

Genomic DNA is prepared for sequencing by fragmenting/shearing: multiple copies of Chromosome + plasmid --> ~500 bp fragments

Note: Nanopore sequencing - there is usually no need to shear the genomic DNA **specialist methods are used to minimise shearing during DNA preparation**. For Nanopore sequencing the longer the DNA fragments the better!

Section 1: Read set summaries and QC

In this section we will import and perform quality control (QC) on our data.

Today we will use 4 pieces of data - 2 short read sets, 1 long read set, and a reference genome to compare our assembly with.

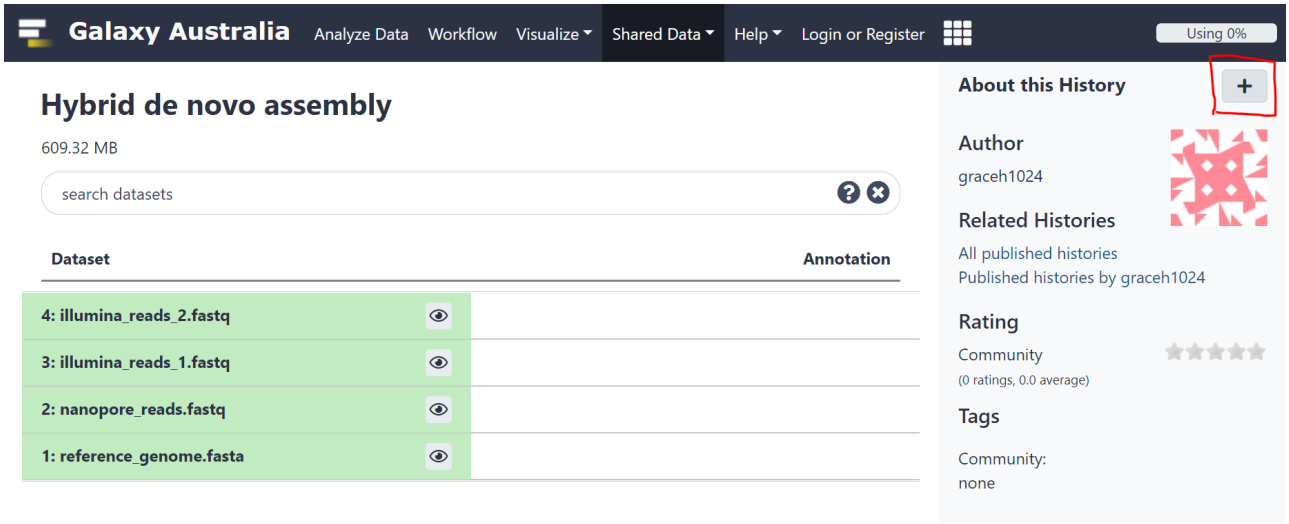
Getting the data

1. **Make sure you have an instance of Galaxy ready to go.**

- Navigate to the Galaxy Australia server and sign in if you have an account.

2. **Copy an existing history**

   - The data you will need is available in an existing Galaxy history:
   - https://usegalaxy.org.au/u/graceh1024/h/hybrid-de-novo-assembly---1hr
   - Import the history '+' icon at the top right of the page.



3. **Look at the history you imported**

   - There are 4 files - Nanopore reads, two sets of Illumina reads, and a reference genome for the organism we will assemble.
   - Our Illumina data are paired-end reads. Two separate files will be present, with only '1' and '2' being different in their filenames.
   - We will use the reference_genome.fasta to assess the quality of our assembly

Read set summaries

Often, it is prudent to first assess the quality of our read sets. For the short reads, we are concerned with base quality, sequence duplication, and presence of adapter sequences. For nanopore, we want to know about the length and quality distribution of reads, as these may both be highly variable.

FastQC creates summary reports for short read data. We will use this tool twice - once for each Illumina read set. We can then use a tool called MultiQC to combine these reports for easy viewing.

For Nanopore data, NanoPlot is a great option. It creates plots which aim to summarise the length and quality distribution of long read sets.

Depending on these summaries, we may choose to perform a QC step to remove any poor quality reads before proceeding.

Run FastQC on each short read set

- Find FastQC in the tools panel. It is listed as '**FastQC** Read Quality reports'

- Parameters:
    - ***Short read data from your current history:*** illumina_reads_1.fastq
    - Leave all else default and execute the program.
- Run FastQC again as above, but change ***Short read data from your current history:*** to 'illumina_reads_2.fastq'.
- Rename the FastQC RawData output for illumina_reads_1.fastq to 'FastQC reads 1 - RawData'
- Rename the FastQC RawData output for illumina_reads_2.fastq to 'FastQC reads 2 - RawData'

FastQC produces two outputs - 'RawData', and 'Webpage'. Typically, the webpage is for human viewing, and the RawData can be given to other programs, such as MultiQC.

At this stage, we have two FastQC outputs - one for each short read set. We will now combine them into a single output using MultiQC for easier interpretation.

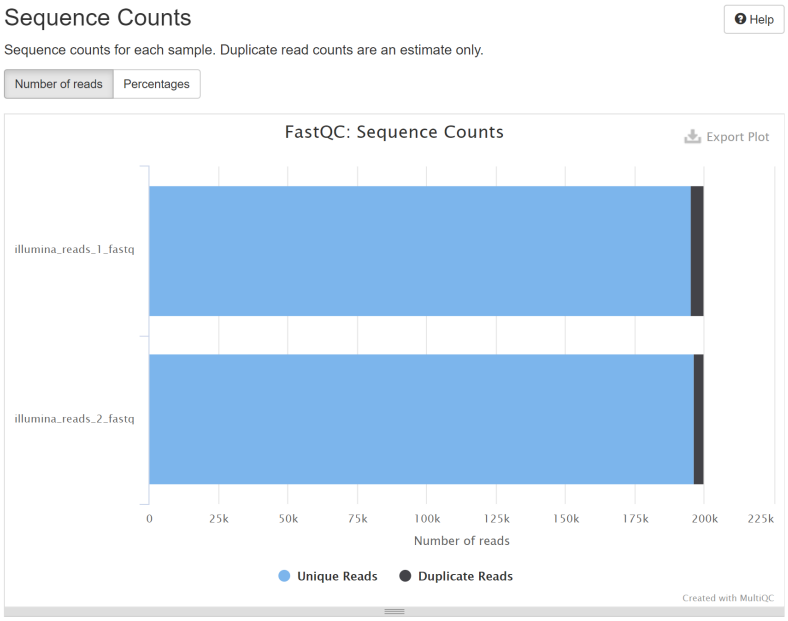
Run MultiQC

- Find MultiQC in the tools panel. It is listed as '**MultiQC** aggregate results from bioinformatics analyses into a single report'
- Parameters:
    - ***Which tool was used generate logs?*** FastQC
    - ***FastQC output***
        - ***Type of FastQC output?*** RawData
            - ***FastQC ouptut*** FastQC reads 1 - RawData
    - click '+ Insert FastQC output'
    - ***FastQC output***
        - ***Type of FastQC output?*** RawData
            - ***FastQC ouptut*** FastQC reads 2 - RawData
- Leave all else default and execute the program.

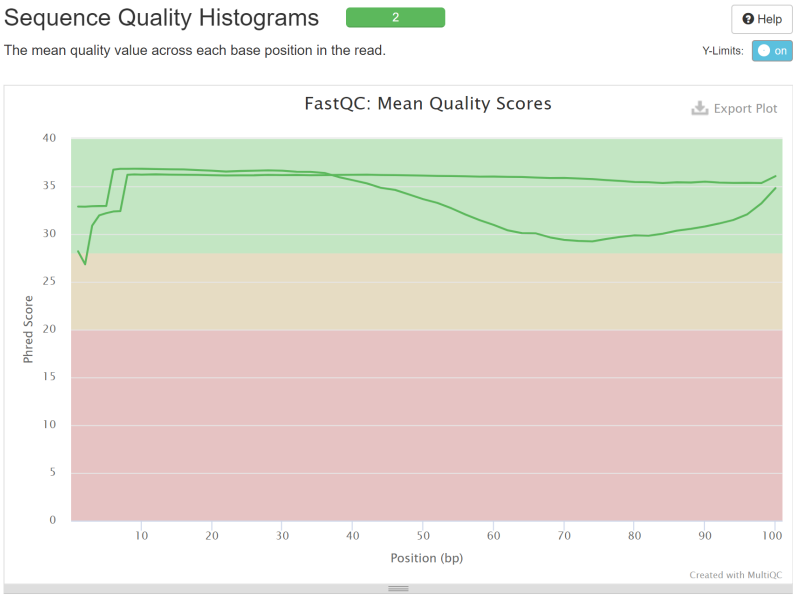
MultiQC also produces two outputs - 'Stats' and 'Webpage'. Inspect the Webpage output by clicking the eye icon on this history item.

MultiQC produces a number of plots, but we are only interested in two today: the 'Sequence Counts' and 'Sequence Quality Histograms' plots.

The 'Sequence Count' displays multiple pieces of information. Firstly, we can see that both read sets have the same total number of reads (sanity check - the number should be identical otherwise some reads don't have a mate!). Additionally, 'Duplicate Reads' are a fraction of the 'Unique Reads'. High levels of sequence duplication can be caused by many factors, but since this is whole genome sequencing (WGS) data, we expect most reads to be unique.



Scrolling down, the 'Sequence Quality Histograms' plot informs us that read 1 for each mate pair is generally lower quality than read 2. This said, they are both high quality and will be fine for our purpose.



Our Illumina reads seem to be reasonable quality. We will now inspect the Nanopore reads.

Run NanoPlot

- Find MultiQC in the tools panel. It is listed as '**NanoPlot** Plotting suite for Oxford Nanopore sequencing data and alignments'
- Parameters:
    - ***Type of the file(s) to work on***
        - ***files*** nanopore_reads.fastq
- Leave all else default and execute the program.

NanoPlot produces 5 outputs, but we are only interested in the 'HTML report' output. View this file by clicking the eye icon on this history item.

The NanoPlot HTML report includes a table, followed by a number of plots. The table provides a summary of the read set. The main statistics we will look at are ***Median read length***, ***Median read quality***, and ***Number of reads***.

NanoPlot report

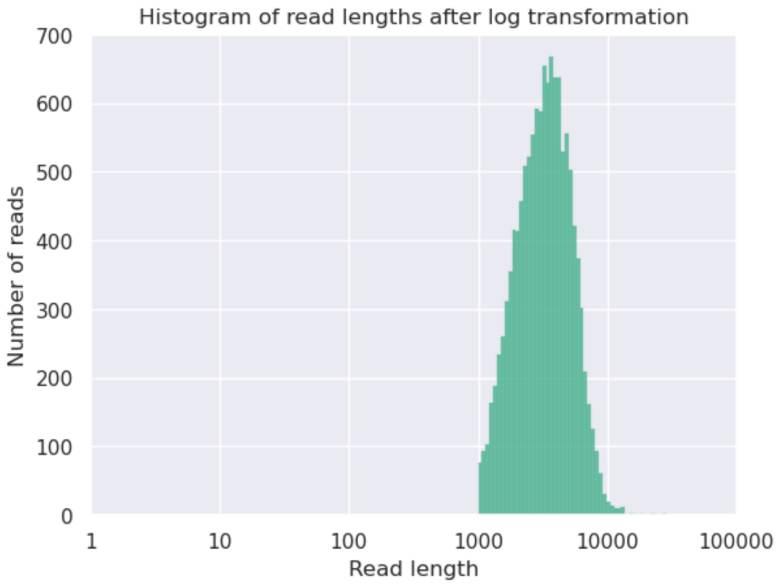
Summary statistics

feature	
General summary	
Mean read length	3,586.5
Mean read quality	11.5
Median read length	3,261.0
Median read quality	11.7
Number of reads	12,500.0
Read length N50	4,143.0
Total bases	44,831,169.0

Our median read length (3261 bp) is short for Nanopore data, but the median quality is good (11.7), equating to a per base accuracy of approximately 93%.

Below the summary table, we see plots relating to read length and quality distributions. Below is the log transformed read length histogram:

Histogram of read lengths after log transformation



The vast majority of our reads sit between 1 - 10 kbp in length. Looking at this histogram closely, we see that the histogram is abruptly cut off at 1000 bp - this read set has previously been filtered to remove reads less than 1000 base pairs, resulting in this effect!

Our read sets appear to be good enough to proceed without performing further QC. Next, we will create an assembly from these reads using Unicycler.

Section 2: Hybrid de-novo assembly

In this section you will use a tool called 'Unicycler' to create a draft genome assembly from Illumina and Nanopore reads.

Unicycler

Unicycler performs assembly using multiple steps. Illumina reads are used first to create an assembly graph using a program called **SPAdes**, then Nanopore reads are used to disentangle problems in the graph. The Nanopore reads serve to bridge Illumina contigs, and to reveal how the contigs are arranged sequentially in the genome. After the assembly is created, a program called **Pilon** uses the Illumina reads to perform a final round of error-correction before the final assembly is output.

For more information on Unicycler, see this link: https://github.com/rrwick/Unicycler

Run Unicycler

- Find Unicycler in the tools panel. It is listed as 'Create assemblies with Unicycler'
- Run Unicycler using the Nanopore and Illumina read sets.
- Parameters:
    - ***Paired or Single end data?*** - Paired
    - ***Select first set of reads*** - illumina_reads_1.fastq
    - ***Select second set of reads*** - illumina_reads_2.fastq
    - ***Select long reads*** - nanopore_reads.fastq
      (if nanopore_reads.fastq does not appear in the dropdown, its datatype needs to be changed - click then pencil icon next to nanopore_reads.fastq in the history panel -> 'Datatypes' tab -> 'New Type' - fastqsanger)
    - ***SPAdes options***
        - ***Number of k-mer steps to use in SPAdes assembly*** 5
    - ***Rotation options***
        - ***Do not rotate completed replicons to start at a standard gene.*** 'Yes'
    - ***Pilon options***
        - ***Do not use Pilon to polish the final assembly.*** 'Yes'
    - Leave all else default and execute the program.
    - Rename the 'Final Assembly' output of Unicycler to 'Unicycler_assembly.fasta'

Unicycler will output two files - a Final Assembly, and a Final Assembly Graph. We are interested in the Final Assembly.

For the sake of time, we have disabled some quality settings in Unicycler. We reduced the number of k-mer steps from 10 to 5 in SPAdes assembly (overall worse assembly), have disabled assembly polishing (improves base-level accuracy), and have chosen not to set the start of each replicon to a standard position. Given more time, we would leave these default and not disable any of these features.

We now have an assembly which we can use for all kinds of downstream analysis. That said, we currently have no idea of the overall quality of this assembly. Additionally, we disabled features which improve the assembly quality due to time constraints, so this is particularly important in our case.

Many factors may impact assembly quality, including the following:

- Read depth / amount of sequence data
- Read quality
- The repetitiveness of the sequenced genome
- Its ploidy (number of copies of each chromosome - humans are diploid: 2 copies)

We need to therefore assess the quality of the assembly we have created. We will use two tools - **Quast**, and **BUSCO** to do this.

Section Questions

Why did we select 'Paired' for our Illumina reads in the Unicycler tool?

▶ Answer (click to reveal)

Does Unicycler begin by using the Long or Short reads?

▶ Answer (click to reveal)

How does Unicycler use long reads to improve its assembly graph?

▶ Answer (click to reveal)

Section 3: Assessing assembly quality

Once we have created the assembly, we need to assess its quality. In this section we will use Quast and BUSCO to perform this assessment. We will also perform BUSCO analysis on the supplied reference genome itself, to record a baseline for our theoretical best BUSCO report.

Quast

A reference genome for the organism we sequenced has been supplied - listed as reference_genome.fasta in the history. This reference genome was assembled using a much higher sequencing depth, and can be used as 'ground truth' to assess our assembly against. The 'Quast' tool will perform this comparison.

- Search for the Quast tool in the tools panel.
- Parameters:
    - ***Contigs/scaffolds file*** Unicycler_assembly.fasta
    - ***Use a reference genome?*** Yes
    - ***Reference genome*** reference_genome.fasta

- Leave all else default and click 'execute'

We are mainly interested in one of the outputs - the HTML report

Open the report. It may look something like this:

Genome statistics	**Unicycler_assembly**
Genome fraction (%)	98.337
Duplication ratio	1
Largest alignment	1 011 894
Total aligned length	3 977 902
NGA50	571 142
LGA50	3
Misassemblies	
# misassemblies	2
Misassembled contigs length	993 585
Mismatches	
# mismatches per 100 kbp	10.31
# indels per 100 kbp	19.1
# N's per 100 kbp	0
Statistics without reference	
# contigs	13
Largest contig	1 011 951
Total length	3 978 601
Total length (>= 1000 bp)	3 978 601

Note the Genome fraction (%), # mismatches per 100 kbp, # indels per 100 kbp and # contigs information.

We seem to have good coverage and not too many contigs, but our error rate is quite high.

In this case we were able to use a reference genome to assess assembly quality, but this is not always the case. When our sample organism is unknown, we need another method to assess assembly quality. BUSCO analysis is one way to do this.

BUSCO

BUSCO analysis uses the presence, absence, or fragmentation of key genes in an assembly to determine is quality.

BUSCO genes are specifically selected for each taxonomic clade, and represent a group of genes which each organism in the clade is expected to possess. At higher clades, 'housekeeping genes' are the only members, while at more refined taxa such as order or family, lineage-specific genes can also be used.

We will use BUSCO to assess our Unicycler draft assembly. While sometimes we will not have a reference genome for comparison, we may be able to find out roughly where it sits in the tree of life using certain methods. In this tutorial, we will suspect that our organism is within the 'Bacillales' *order*.

- Search for the Quast tool in the tools panel.
- Parameters:
    - ***Sequences to analyse*** Unicycler_assembly.fasta
    - ***Lineage*** Bacillales
- Leave all else default and execute the program.

After the program has run, look at the 'short summary' output. It may look something like this:

```
***** Results: *****

C:99.1%[S:98.9%,D:0.2%],F:0.7%,M:0.2%,n:450
446     Complete BUSCOs (C)
445     Complete and single-copy BUSCOs (S)
1       Complete and duplicated BUSCOs (D)
3       Fragmented BUSCOs (F)
1       Missing BUSCOs (M)
450     Total BUSCO groups searched
```

The 'full table' is also useful. It gives a detailed list of the genes we are searching for, and information about whether they would missing, fragmented, or complete in our assembly.

# Busco id	Status	Sequence	Gene Start	Gene End	Score	Length	OrthoDB url	Description
359at1385	Complete	10_52	35379	38960	1999.8	1004	https://www.orthodb.org/v10?query=359at1385	DNA-directed RNA polymerase subu
362at1385	Complete	2_802	758503	762867	1978.2	1101	https://www.orthodb.org/v10?query=362at1385	DNA polymerase III PolC-type
434at1385	Complete	10_51	31718	35317	1827.9	965	https://www.orthodb.org/v10?query=434at1385	DNA-directed RNA polymerase subu
1064at1385	Complete	2_693	653917	657132	1552.3	877	https://www.orthodb.org/v10?query=1064at1385	carbamoyl phosphate synthase large
1366at1385	Complete	11_29	27943	31476	1305.1	826	https://www.orthodb.org/v10?query=1366at1385	Transcription-repair-coupling factor
2297at1385	Complete	1_451	448025	450550	1227.7	725	https://www.orthodb.org/v10?query=2297at1385	Protein translocase subunit SecA
2753at1385	Complete	10_72	54877	57309	1352.1	707	https://www.orthodb.org/v10?query=2753at1385	ATP-dependent Clp protease ATP-bi
3174at1385	Complete	3_416	381574	384216	1057.5	673	https://www.orthodb.org/v10?query=3174at1385	DNA polymerase I
3201at1385	Complete	2_962	962623	965046	1152.8	676	https://www.orthodb.org/v10?query=3201at1385	DNA topoisomerase IV subunit A
3378at1385	Complete	3_430	399659	403006	1033.0	842	https://www.orthodb.org/v10?query=3378at1385	DNA polymerase III subunit alpha
3440at1385	Complete	9_1	301	2766	1407.2	767	https://www.orthodb.org/v10?query=3440at1385	DNA gyrase subunit A

It seems that most expected genes are missing or fragmented in our assembly. It is likely that the frequent errors in our draft assembly are causing this result. We should be able improve our assembly with the Illumina reads available and correct some of these errors.

This process involves two steps. We will first align the Illumina reads to our draft assembly, then supply the mapping information to Pilon which will use this alignment information to error-correct our assembly.

Section Questions

How does Quast inform on assembly quality?

▶ Answer (click to reveal)

How does BUSCO inform on assembly quality?

▶ Answer (click to reveal)

Conclusion

Today we have learned one workflow for performing hybrid de novo genome assembly. The combination of long- and short-read technology is clearly powerful, represented by our ability to create a good assembly with only approximately 20x coverage (87Mb) of Nanopore, and 25x coverage of Illumina reads (100Mb).

To further improve our assembly, extra read data may provide most benefit. For bacterial genomes, we often prefer 100x or greater coverage, with which we may achieve a high per-base accuracy assembly with the correct number of contigs (number of replicons in the genome).

The development of new purpose-built tools for hybrid de novo assembly like Unicycler have improved the quality of assemblies we can produce. These tools are of great importance and while they already produce great results, they will continue to improve over time.

Additional reading

Links to additional recommended reading and suggestions for related tutorials.
Pilon: https://github.com/broadinstitute/pilon/wiki/Methods-of-Operation
Unicycler: https://github.com/rrwick/Unicycler
Quast: https://academic.oup.com/bioinformatics/article/29/8/1072/228832
BUSCO analysis: https://academic.oup.com/bioinformatics/article/31/19/3210/211866