Research Institute for Medicines

UNIVERSIDADE DE LISBOA

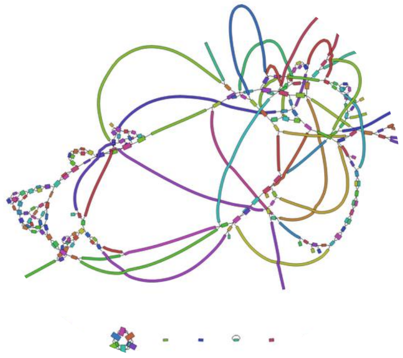
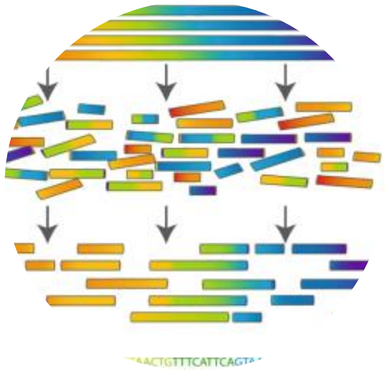
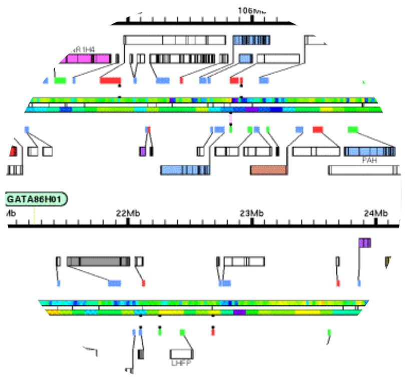
FACULDADE DE FARMÁCIA Universidade de Lisboa

PMB2022

PATHOGEN MULTIOMICS AND BIOINFORMATICS

LISBON 2022

Module 2: *De novo* Assembly

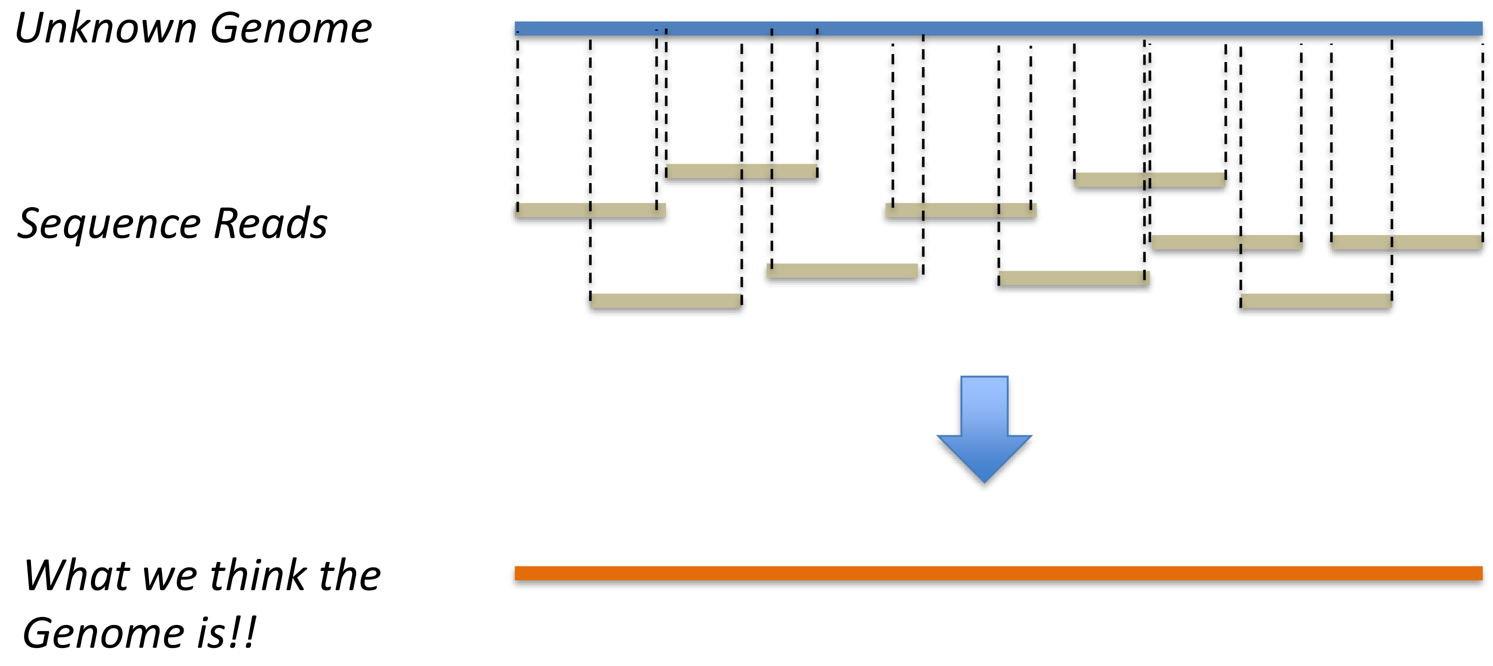




João Perdigão

De novo Assembly

De novo assembly – genome assembly without a reference genome, i.e., starting *de novo* from sequence data

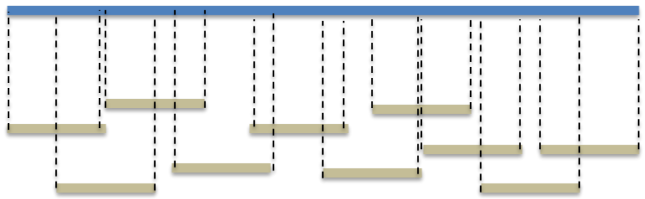
De novo assembly ≠ Reference assembly



De novo Assembly

*de novo a*ssembly attempts to reconstruct genomes by exploring read overlapping and contiguity

Problems and Challenges:



- Large volume of sequence reads
- Sequencing errors
- Genomic repeat patterns/regions and homopolymers
- Uneven coverage/sequencing

But, what is the definition of an assembly?

Best set of sequences that can approximate the sequenced genetic material

Implications?

imed Research Institute for Medicines

Why to assemble?

Objective/Purpose of the Assembly:

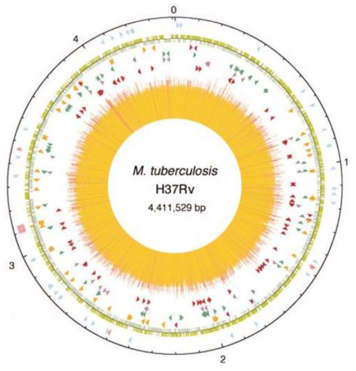
Obtain a reference genome

↓

PacBio

↓

Finished Genome Assembly



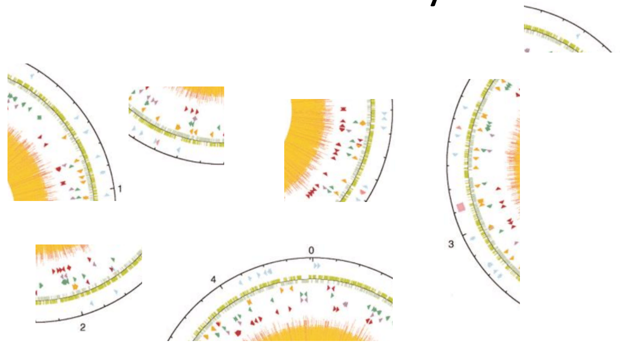
Gene content, insertions, deletions

↓

Illumina

↓

Draft Genome Assembly



←

Manual closure

Mate-pair sequencing

Assembly Methods

Three major methods for assembly:

i) overlap-and-extend

Finds read overlap and extends – suffix of a read is equal to the prefix of another read with a length that meets a defined threshold.

Software: SSAKE, VCAKE and SHARCGS

ii) string graph

Construction of string graph where each read is a vertex with edges connecting overlapping nodes.

Software: Edena and BOA

Problems: high memory comsumption and sequencing errors

Assembly Methods

Three major methods for assembly:

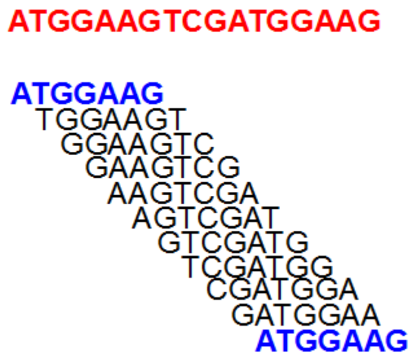
iii) de Bruijn Graph

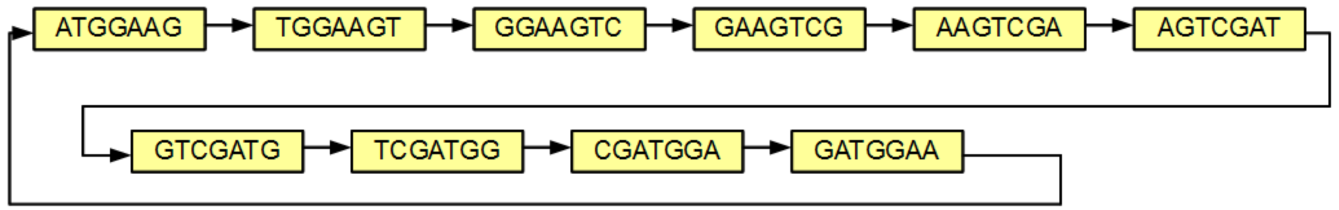
Each vertex represents a substring of length *k* (*k*-mer) in a read. Edges connect vertexes if these are consecutive vertexes, i.e., the last *k*-1 nucleotides in k-mer *u* are the same as the as the first *k*-1 nucleotides of k-mer *v.*

Software: Velvet, SOAPdenovo, SPAdes

Most widely used approaches.

Objective: represent every possible *k*-mer present in the genome!





Assembly Methods

Three major methods for assembly:

iii) de Bruijn Graph

Connecting the nodes:

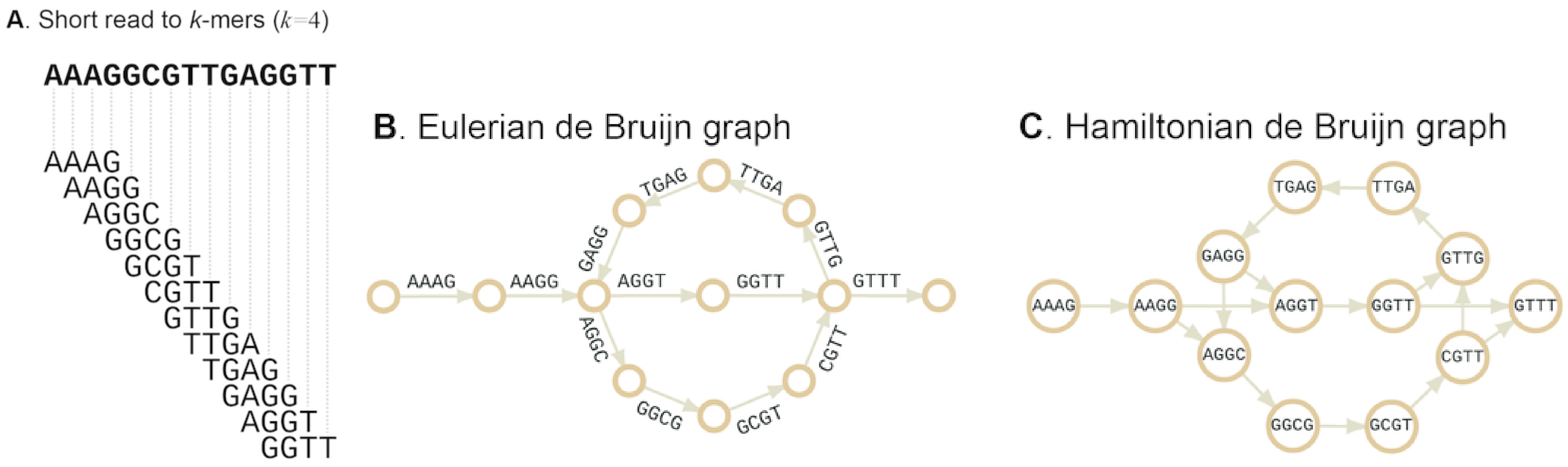
•*Hamiltonian path*: visits **every vertex in the graph** (exactly once, because it is a path)

•*Eulerian trail*: visits **every edge in the graph** exactly once (because it is a trail, vertices may well be crossed more than once.)

William Hamilton

Leonhard Euler

Nicolaas de Bruijn



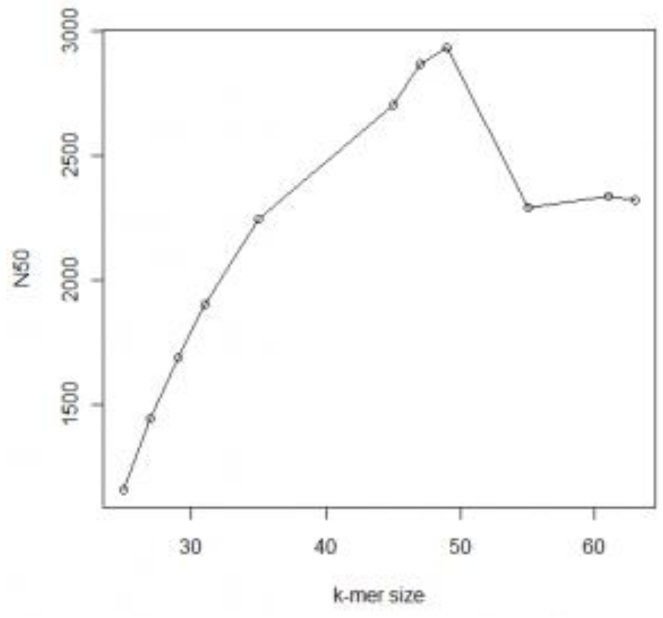
Assembly Methods

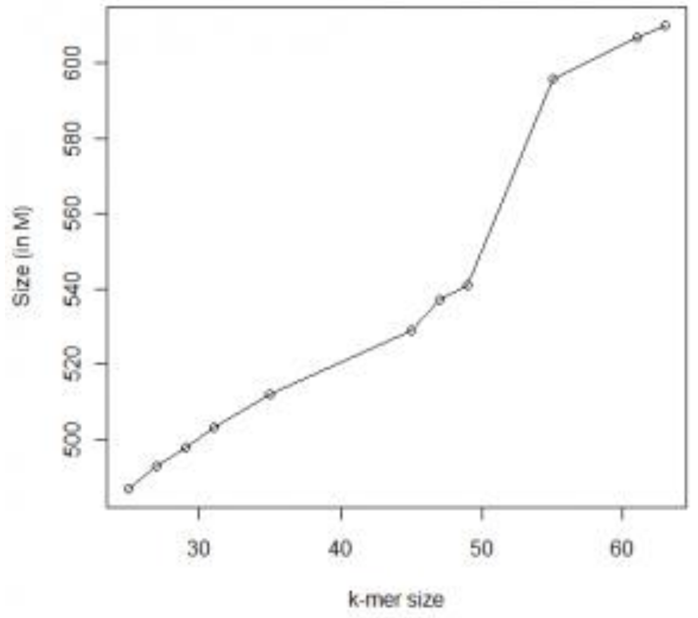
Three major methods for assembly:

iii) de Bruijn Graph

Choice of the k-mer length: always below the read-length of your data!

The k-mer length should be an odd number: avoid palindromes – an odd-sized k-mer cannot form palindromes when reverse-complemented!





https://homolog.us/blogs/genome/2012/10/10/multi-kmer-de-bruijn-graphs/

De Bruijn Graphs - Exercise

Let's consider the following reads:

ATGCTA

GCTAGC

TAGCAC

CTAGCA

TGC GCT CTA

ATG

TAG

CAC

AGC

GCA

1 List all 3bp k-mers

2 Establish links (edges) between k-mers diffreing by k-1 nucleotides

3 Visit all nodes and use the minimal path length

De Bruijn Graphs - Exercise

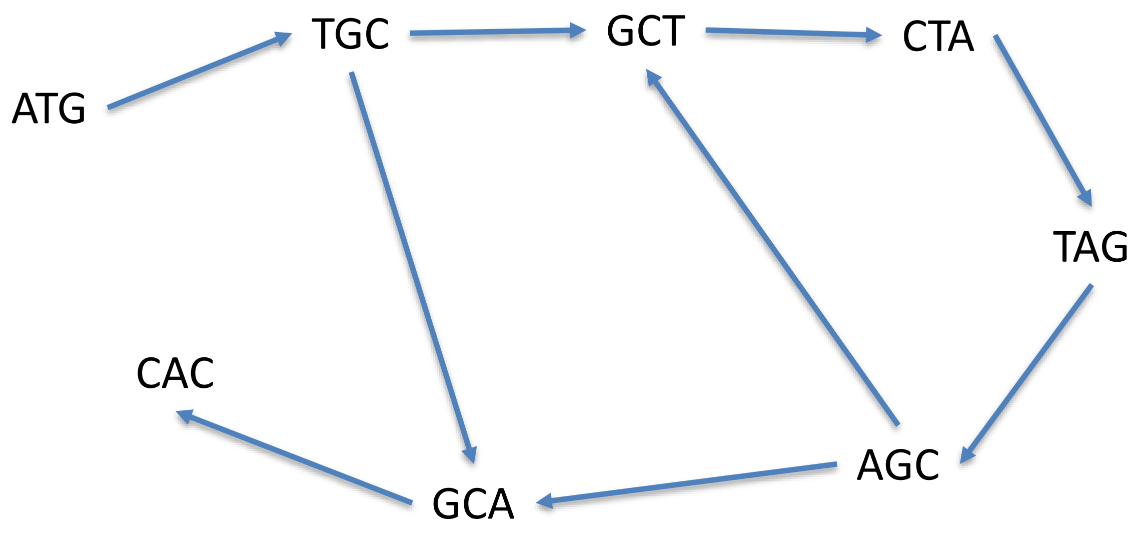
Let's consider the following reads:

ATGCTA

GCTAGC

TAGCAC

CTAGCA



1 List all 3bp k-mers

2 Establish links (edges) between k-mers diffreing by k-1 nucleotides

3 Visit all nodes only once

De Bruijn Graphs - Exercise

Let's consider the following reads:

ATGCTA

GCTAGC

TAGCAC

CTAGCA

$$\begin{tikzcd}
 & TGC \arrow[rr] \arrow[ddr] & & GCT \arrow[r] & CTA \arrow[dr] & \\
ATG \arrow[ur] & & & & & TAG \arrow[dl] \\
 & CAC & GCA \arrow[l] & & AGC \arrow[ll] \arrow[uul] &
\end{tikzcd}$$

ATGCTAGCAC

1 List all 3bp k-mers

2 Establish links (edges) between k-mers diffreing by k-1 nucleotides

3 Visit all nodes only once

De Bruijn Graphs - Exercise

Let’s consider the following reads:

ATGCTA

GCTAGC

TAGCAC

CTAGCA

The Eulerian Path

$$\begin{tikzcd} AT \arrow[r, "ATG"] & TG \arrow[r, "TGC"] & GC \arrow[r, "GCT"] & CT \arrow[r, "CTA"] & TA \arrow[r, "TAG"] & AG \arrow[r, "AGC"] & GC \arrow[r, "GCA"] & CA \arrow[r, "CAC"] & AC \end{tikzcd}$$

1 List all 3bp k-mers

2 Define nodes between edges

3 Visit all edges only once

De Bruijn Graphs - Exercise

Let’s consider the following reads:

ATGCTA

GCTAGC

TAGCAC

CTAGCA

The Eulerian Path

$$\begin{tikzcd}
AT \arrow[r, "ATG"] & TG \arrow[r, "TGC"] & GC \arrow[r, "GCT"] & CT \arrow[r, "CTA"] & TA \arrow[r, "TAG"] & AG \arrow[r, "AGC"] & GC \arrow[r, "GCA"] & CA \arrow[r, "CAC"] & AC
\end{tikzcd}$$

1 List all 3bp k-mers

2 Define nodes between edges

3 Visit all edges only once

De Bruijn Graphs - Exercise

Let’s consider the following reads:

ATGCTA

GCTAGC

TAGCAC

CTAGCA

The Eulerian Path

$$\begin{tikzcd}
AT \arrow[r, "ATG"] & TG \arrow[r, "TGC"] & GC \arrow[r, "GCT"] & CT \arrow[r, "CTA"] & TA \arrow[r, "TAG"] & AG \arrow[r, "AGC"] & GC \arrow[r, "GCA"] & CA \arrow[r, "CAC"] & AC
\end{tikzcd}$$

$$\begin{tikzcd}
AT \arrow[r, "ATG"] & TG \arrow[r, "TGC"] & GC \arrow[r, "GCT"] \arrow[dr, "GCA"] & CT \arrow[r, "CTA"] & TA \arrow[r, "TAG"] & AG \arrow[lll, bend right, "AGC"'] \\
 & & & CA \arrow[r, "CAC"] & AC &
\end{tikzcd}$$

ATGCTAGCAC

1 List all 3bp k-mers

2 Define nodes between edges

3 Visit all edges only once

Evaluating and Comparing Assemblies

Metrics to Evaluate and Compare Assemblies:

Evaluating an assembly can be reference-free or comparing to a reference genome**!**

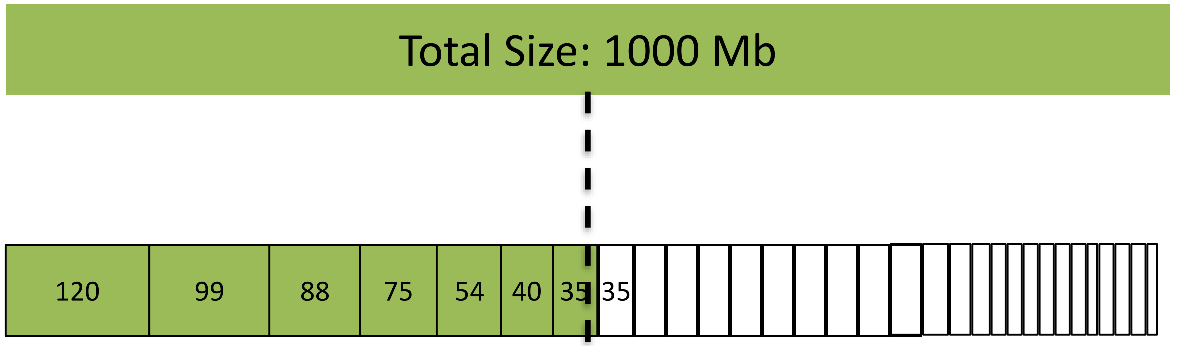
Purpose:

1. Assess the individual quality of an assembly;
2. Compare assemblers.

Metrics commonly used:
- Number of contigs/scaffolds
- Total length of the assembly
- Length of the largest contig/scaffold
- Percentage of gaps in scaffolds ('N')
- N50/NG50 of contigs/scaffolds
- Number of predicted genes
- Number of core single-copy genes

Software:
QUAST – allows comparison against reference
BUSCO – evaluates the presence of core single-copy orthologous genes

N50 and L50 and other metrics



N50 – shortest contig length spanning the midpoint of the assembly length (after sorting from largest to smallest contig);

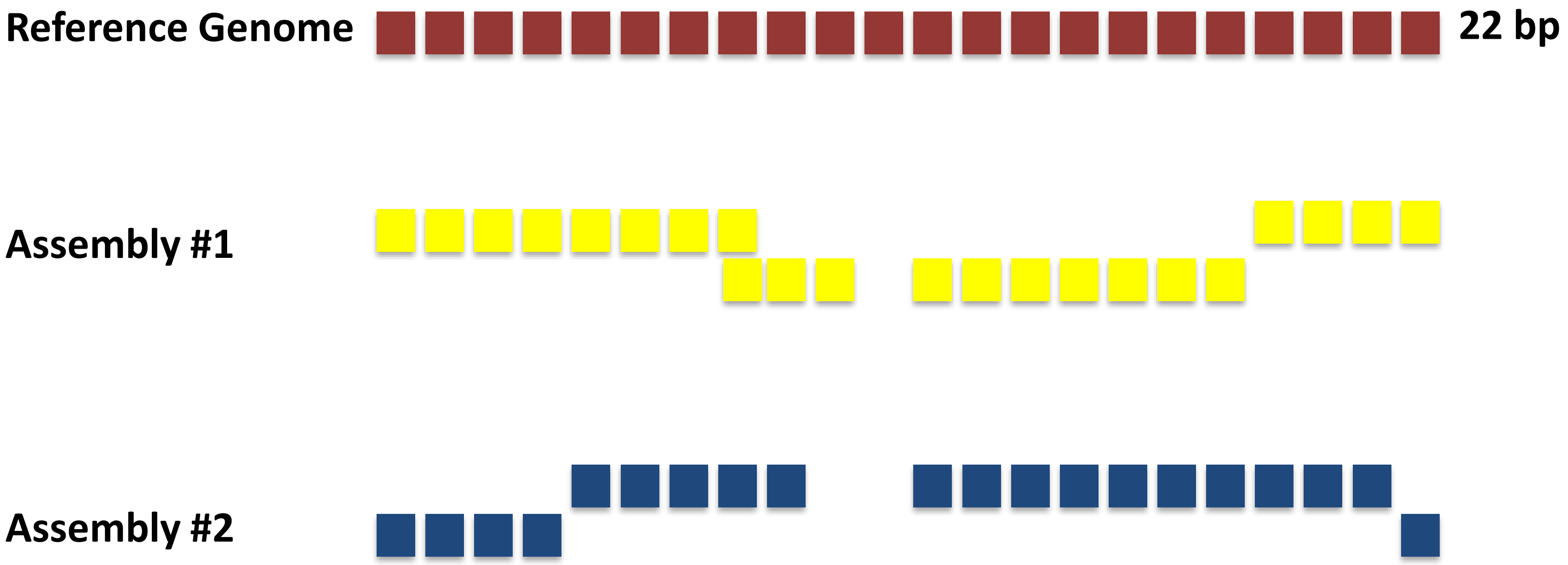
NG50 – shortest contig length spanning the midpoint of the estimated genome size (after sorting from largest to smallest contig);

L50 – number of contigs necessary to span the midpoint of the assembly length (after sorting from largest to smallest contig)

LG50 – number of contigd necessary to span the midpoint of the estimated genome size (after sorting from largest to smallest contig)

N50 and L50 in this exemple?

Evaluating and Comparing Assemblies

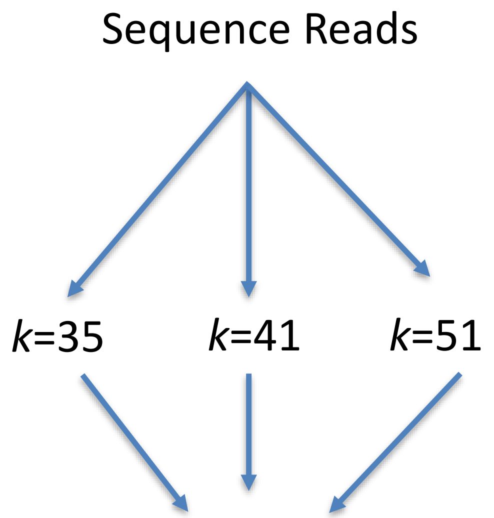


Calculate:	Assembly #1	Assembly #2
N50	**7**	**10**
NG50	7	**5**
Coverage	**95.4%**	**90.9%**
Total Length of Assembly	**22**	**20**

Multi *k-mer* assembly – the way forward

Multi k-mer asssembly:

Multi-k-mer stategies always provide better results than single k-mer



Merge Assemblies -> Velvet+cd-hit+minimus2
Iterative Assembly removing assembled reads – IDBA
Multi-kmer de Bruijn graph - **SPAdes**

SPAdes 3.13.0

Assembly pipelines

Unicycler

As input, Unicycler takes one of the following:
•Illumina reads from a bacterial isolate (ideally paired-end, but unpaired works too)
•A set of long reads (either PacBio or Nanopore) from a bacterial isolate (uncorrected long reads are fine, though corrected long reads should work too)
•Illumina reads and long reads from the same isolate (best case)

Unicycler does:
- ***Short-read assembly***
- ***Long-read assembly***
- ***Hybrid assembly***

Reasons to use Unicycler:
•It circularises replicons without the need for a separate tool like Circlator.
•It handles plasmid-rich genomes.
•It can use long reads of any depth and quality in hybrid assembly. 10x or more may be required to complete a genome, but Unicycler can make nearly-complete genomes with far fewer long reads.
•It produces an assembly *graph* in addition to a contigs FASTA file, viewable in Bandage.
•It has very low misassembly rates.
•It can cope with very repetitive genomes, such as *Shigella*.
•It's easy to use: runs with just one command and usually doesn't require tinkering with parameters.

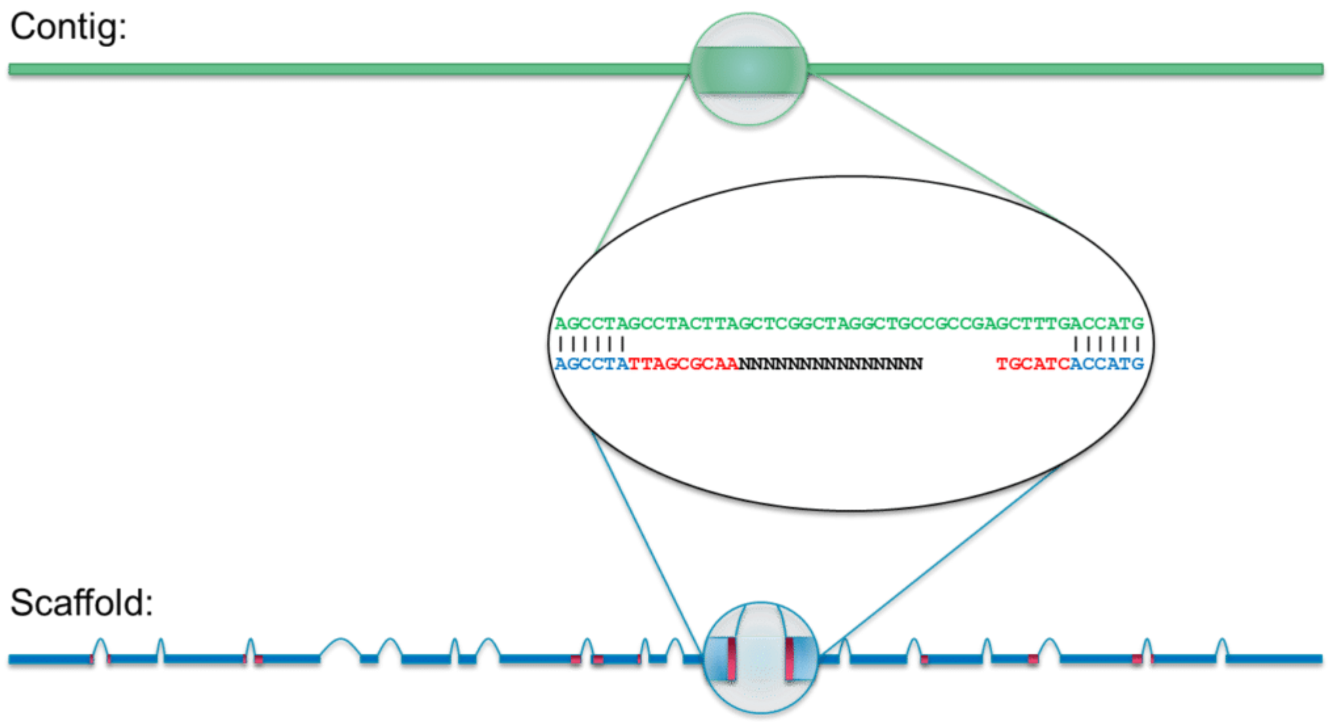
Reasons to not use Unicycler:
•You're assembling a eukaryotic genome or a metagenome (Unicycler is designed exclusively for bacterial isolates).
•Your Illumina reads and long reads are from different isolates (Unicycler struggles with sample heterogeneity).
•You're impatient (Unicycler is thorough but not especially fast).

https://github.com/rrwick/Unicycler

Scaffolding

Contigs are continuous stretches of sequence containing only A, C, G, or T bases without gaps.

*Scaffolds are created by chaining contigs together **using additional information** about the relative position and orientation of the contigs in the genome*


Contig:
AGCCTAGCCTACTTAGCTCGGCTAGGCTGCCGCCGAGCTTTGACCATG
AGCCTATTAGCGCAANNNNNNNNNNNNNNN
TGCATCACCATG
Scaffold:

ABACAS

MAUVE

ARCS

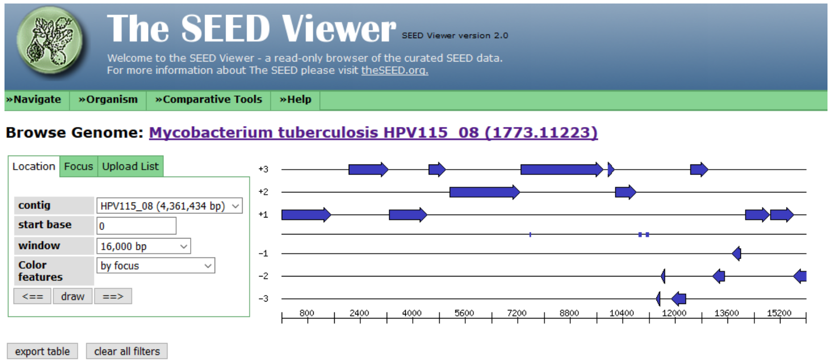
imed Research Institute for Medicines

Genome Annotation

Prokka

RAST

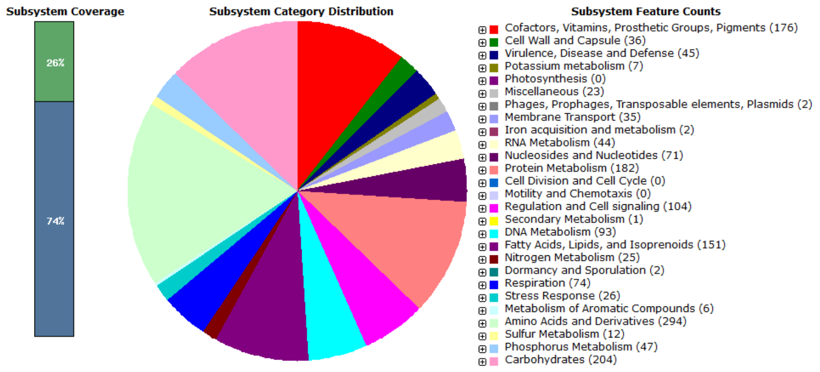
NCBI Prokaryotic Annotation Pipeline


The SEED Viewer SEED Viewer version 2.0

Welcome to the SEED Viewer - a read-only browser of the curated SEED data.
For more information about The SEED please visit theSEED.org.

Browse Genome: Mycobacterium tuberculosis HPV115_08 (1773.11223)

contig	HPV115_08 (4,361,434 bp)
start base	0
window	16,000 bp
Color features	by focus


Subsystem Coverage
Subsystem Category Distribution
Subsystem Feature Counts
Cofactors, Vitamins, Prosthetic Groups, Pigments (176)
Cell Wall and Capsule (36)
Virulence, Disease and Defense (45)
Potassium metabolism (7)
Photosynthesis (0)
Miscellaneous (23)
Phages, Prophages, Transposable elements, Plasmids (2)
Membrane Transport (35)
Iron acquisition and metabolism (2)
RNA Metabolism (44)
Nucleosides and Nucleotides (71)
Protein Metabolism (182)
Cell Division and Cell Cycle (0)
Motility and Chemotaxis (0)
Regulation and Cell signaling (104)
Secondary Metabolism (1)
DNA Metabolism (93)
Fatty Acids, Lipids, and Isoprenoids (151)
Nitrogen Metabolism (25)
Dormancy and Sporulation (2)
Respiration (74)
Stress Response (26)
Metabolism of Aromatic Compounds (6)
Amino Acids and Derivatives (294)
Sulfur Metabolism (12)
Phosphorus Metabolism (47)
Carbohydrates (204)