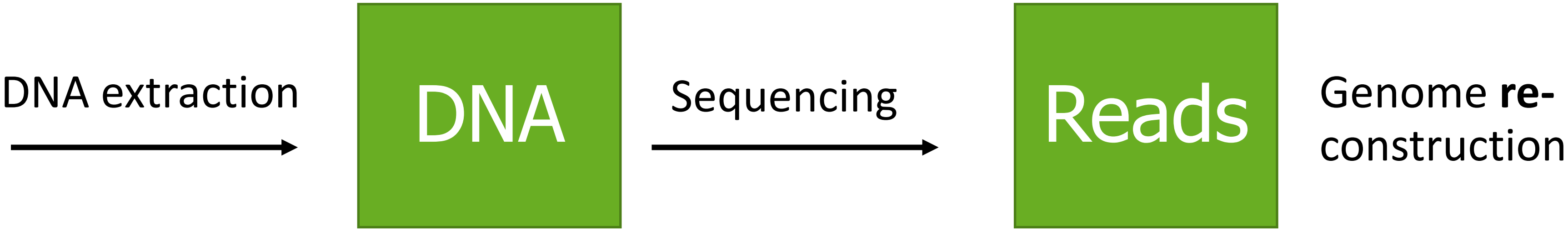
EUROPEAN CENTRE FOR DISEASE PREVENTION AND CONTROL

Sequencing and assembly

Intro to genome assembly strategies

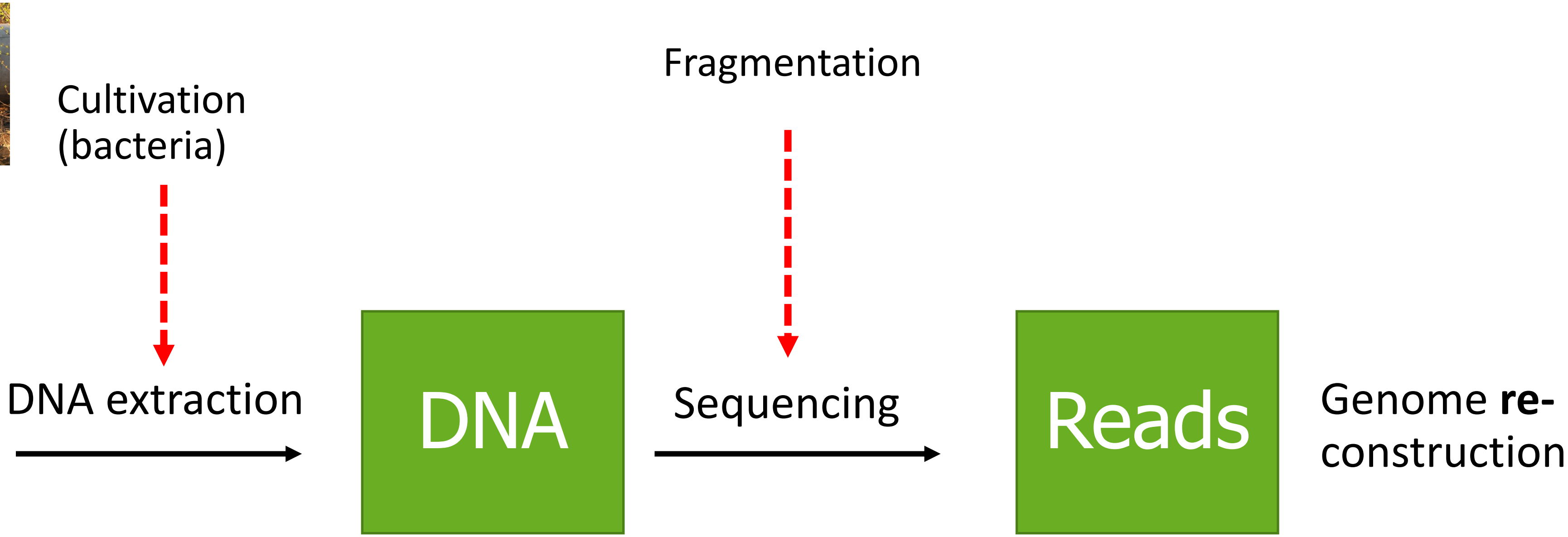
How to sequence a genome



DNA extraction → DNA → Sequencing → Reads → Genome **re-**construction

Images: giantmicrobes.com, https://nccid.ca/wastewater-surveillance-for-covid-19/

How to sequence a genome



Cultivation (bacteria)

DNA extraction

DNA

Fragmentation

Sequencing

Reads

Genome **re-**construction

Images: giantmicrobes.com, https://nccid.ca/wastewater-surveillance-for-covid-19/

The assembly problem

TAGCC ATGTT

AGCCG GCCGG GTTTA

TGTTT

TTAGC

TTTAG

The assembly problem

TAGCC

AGCCG

GCCGG

TTTAG
GTTTA
TTAGC

ATGTT
TGTTT

The assembly problem

ATGTTTAGCCGG

ATGTT

TGTTT

GTTTA

TTTAG

TTAGC

TAGCC

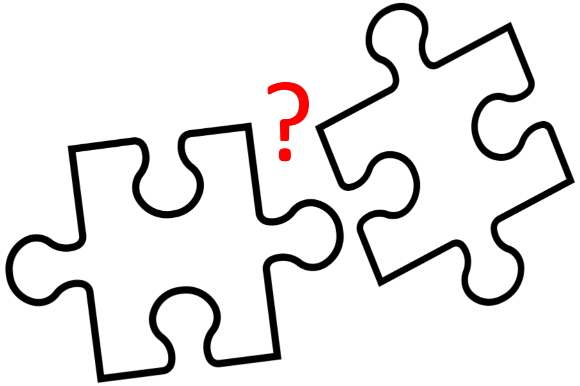
AGCCG

GCCGG

The assembly problem

read-length and base-calling quality matters

PHASE TWO : INTERPRETATION
I THINK I FOUND A CORNER PIECE.
3 BILLION PIECES
GENOME
The Star-Ledger

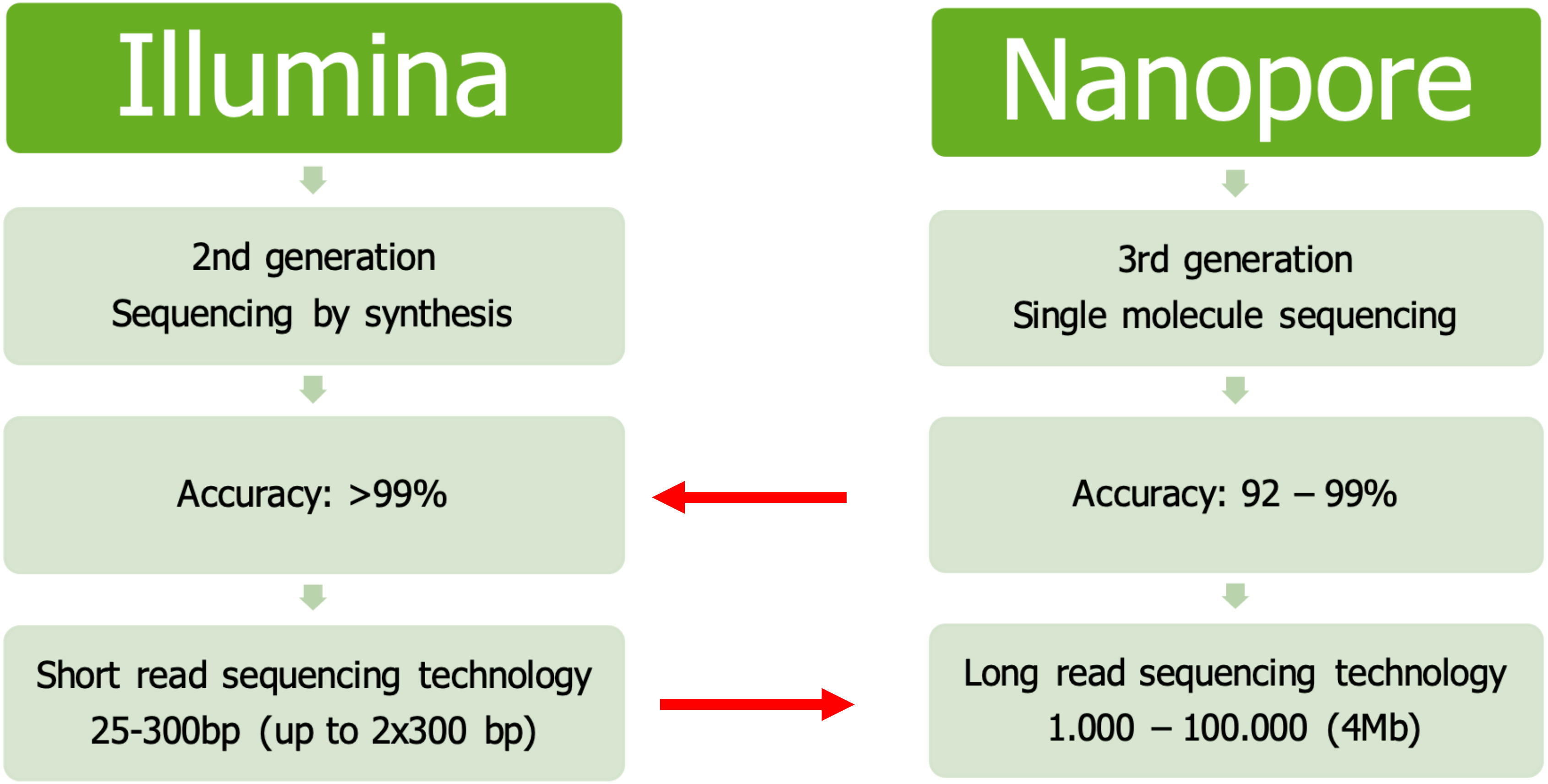


Read-length:
Jigsaw puzzles with many small pieces are hard!

Read-accuracy:
Like a jigsaw puzzle with somewhat misshapen pieces

https://twitter.com/FLGenomics/status/520611003751735296

There is a trade-off between read-length and quality



Illumina

2nd generation
Sequencing by synthesis

Accuracy: >99%

Short read sequencing technology
25-300bp (up to 2x300 bp)

Nanopore

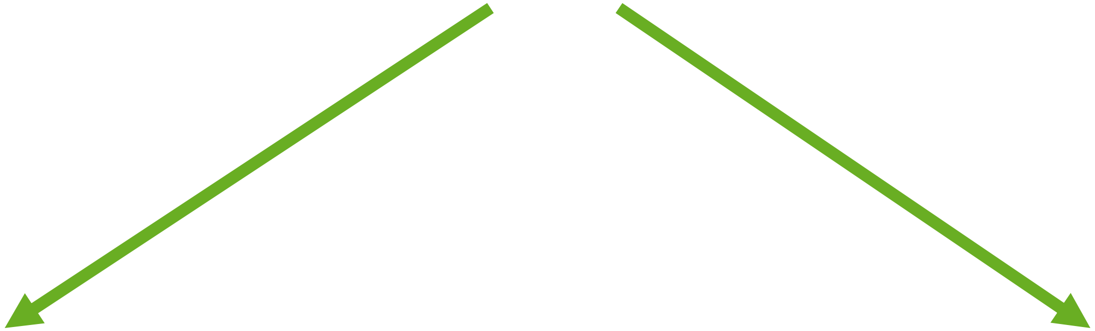
3rd generation
Single molecule sequencing

Accuracy: 92 – 99%

Long read sequencing technology
1.000 – 100.000 (4Mb)

Assembly algorithms

two main classes



OLC: Overlap layout consensus

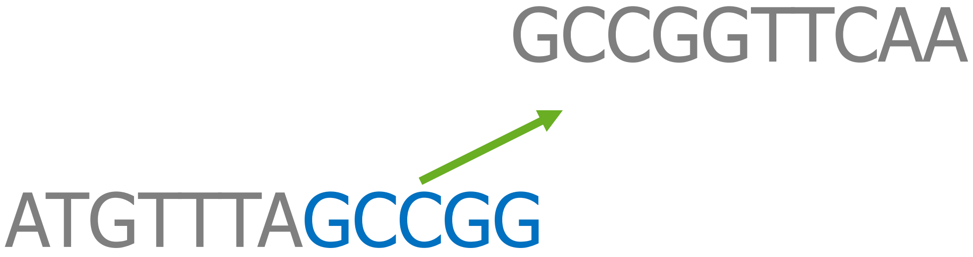
- Assembly graph made from overlaps between reads

De-Bruijn graphs

- Assembly graph made from shared k-mers of reads

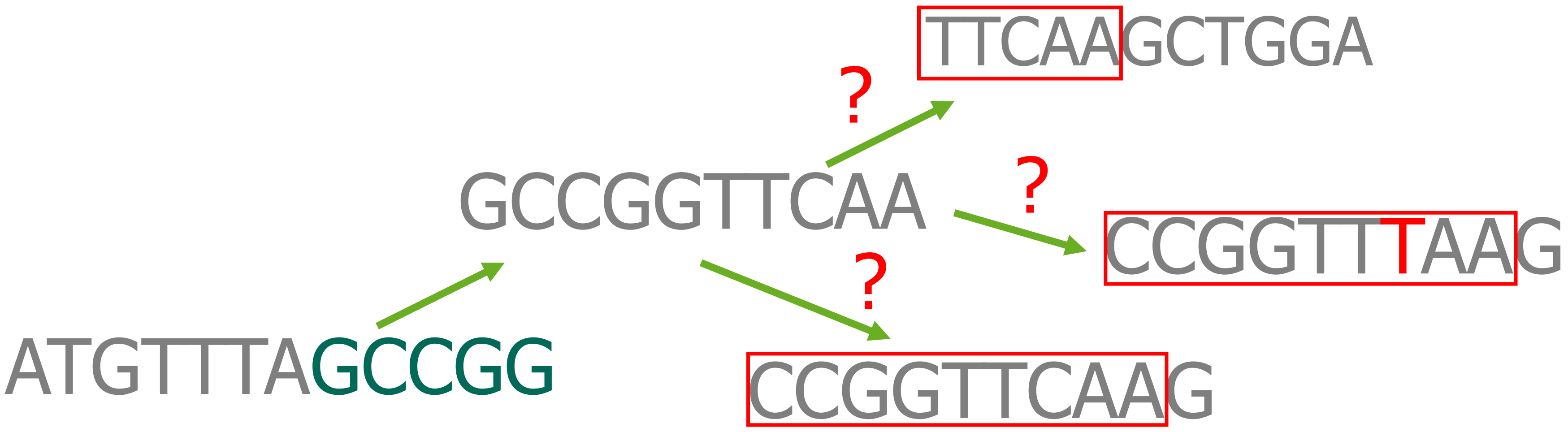
OLC: Overlap-layout-consensus

- Intuitive. First assemblers used this approach.
- Looking for overlaps: if suffix of one read has significant similarity with prefix of another, the two reads are connected in the assembly graph.

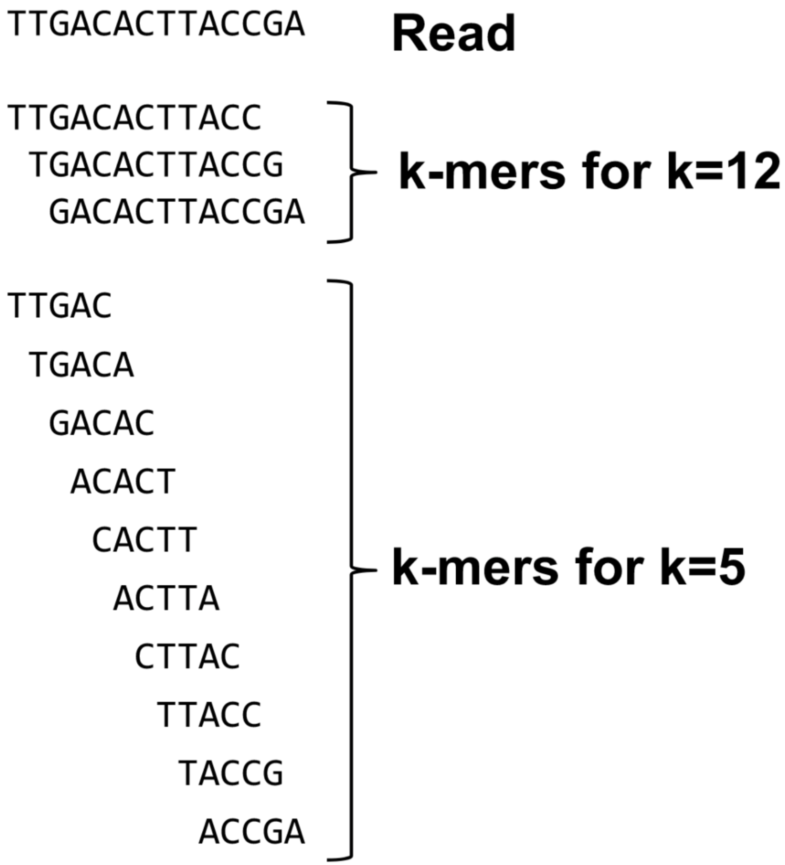


"OLC": Overlap-layout-consensus

- Intuitive. First assemblers used this approach
- Looking for overlaps: if suffix of one read has "significant similarity" with prefix of another, the two reads are connected in the assembly graph.



De bruijn graphs



TTGACACTTACCGA — Read

TTGACACTTACC
TGACACTTACCG
GACACTTACCGA

k-mers for k=12

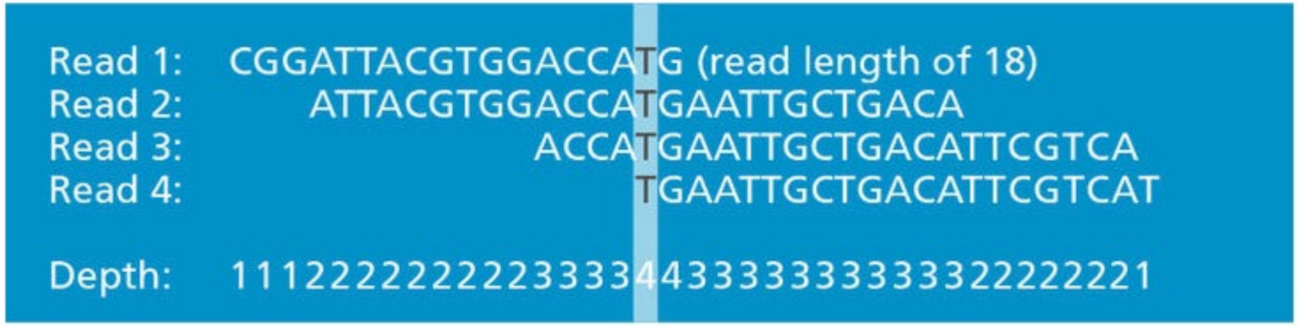
TTGAC
TGACA
GACAC
ACACT
CACTT
ACTTA
CTTAC
TTACC
TACCG
ACCGA

k-mers for k=5

- de Bruijn graphs utilize **k-mers** of reads to construct the assembly graph
- Once the k-mers have been constructed, reads having a k-mer in common can be found very quickly **(“Hashing”)**

training.galaxyproject.org, geeksforgeeks.org/what-is-hashing

De bruijn graphs:
Coverage matters



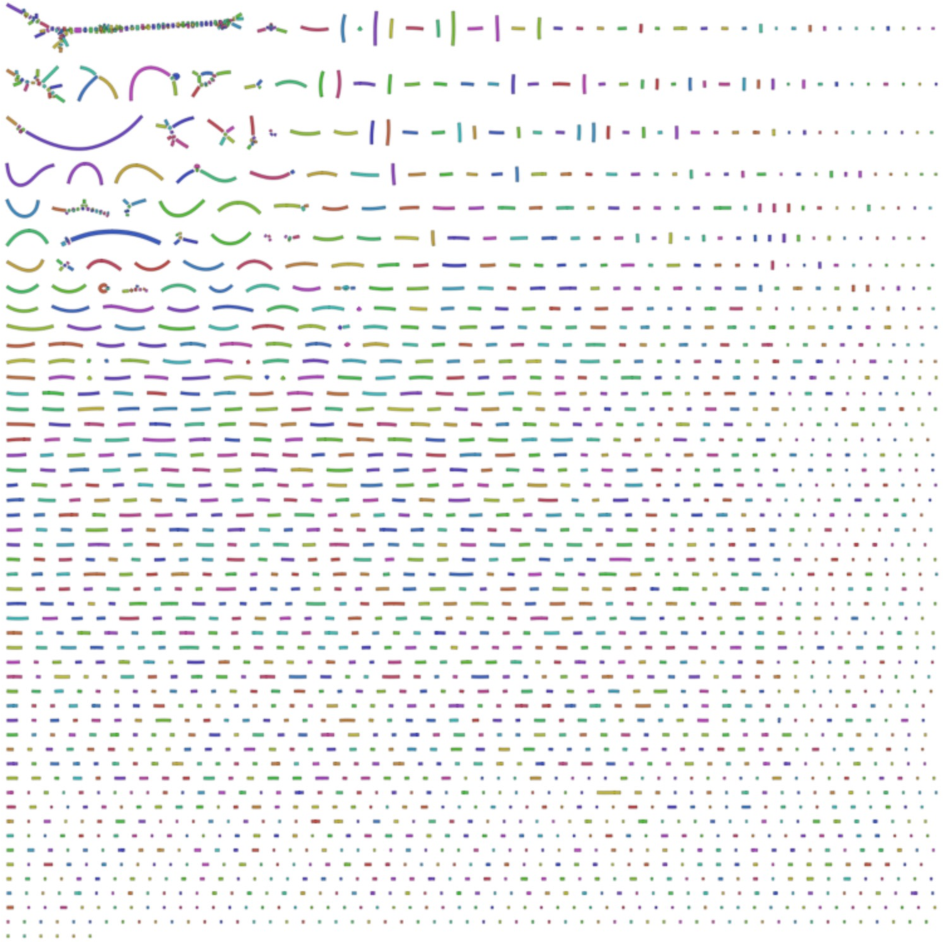
```
Read 1:   CGGATTACGTGGACCATG (read length of 18)
Read 2:      ATTACGTGGACCATGAATTGCTGACA
Read 3:               ACCATGAATTGCTGACATTCGTCA
Read 4:                   TGAATTGCTGACATTCGTCAT

Depth:    1112222222223333443333333333322222221
```

- de bruijn graphs require **exact** k-mer matches
- More coverage => more error-free reads
- The frequency of k-mers can be counted and used to annotate the graph (and to clean it up)

Which k-mer size to use?

- If the k-mer is too **large**, the reads wont get connected in the graph

Salmonella genome assembled with Illumina **100bp** (single-end) reads, and **kmer-size 91**

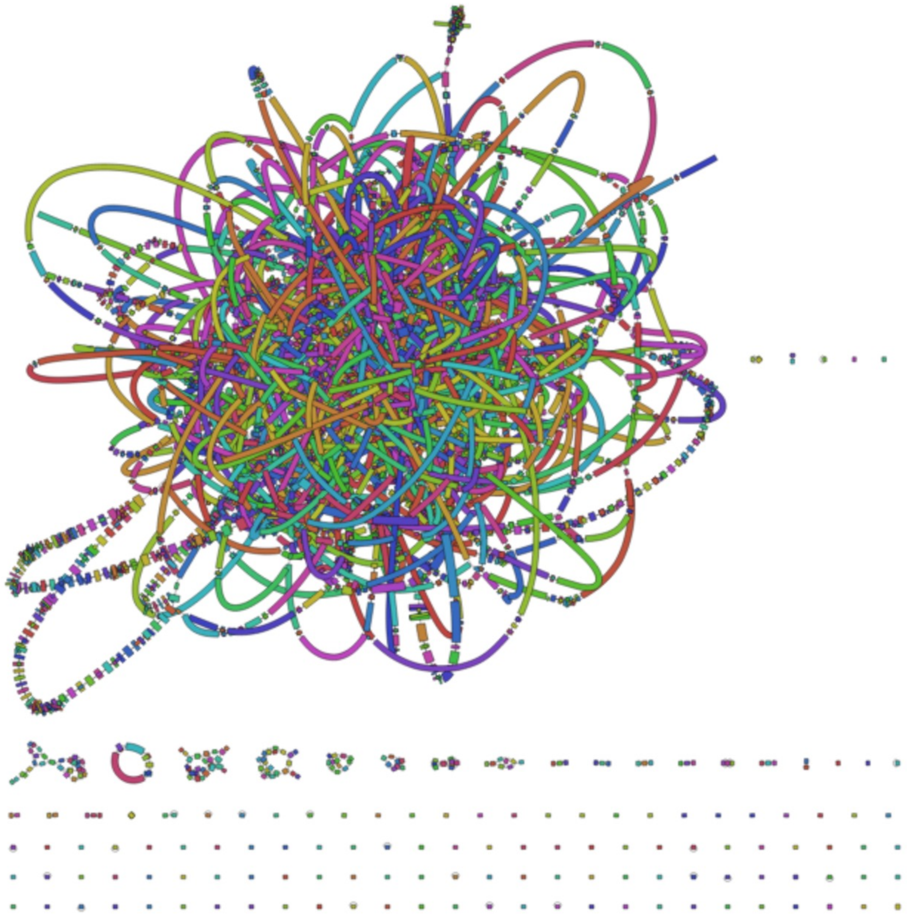


https://github.com/rrwick/Bandage/wiki/Effect-of-kmer-size

Which kmer-size to use?

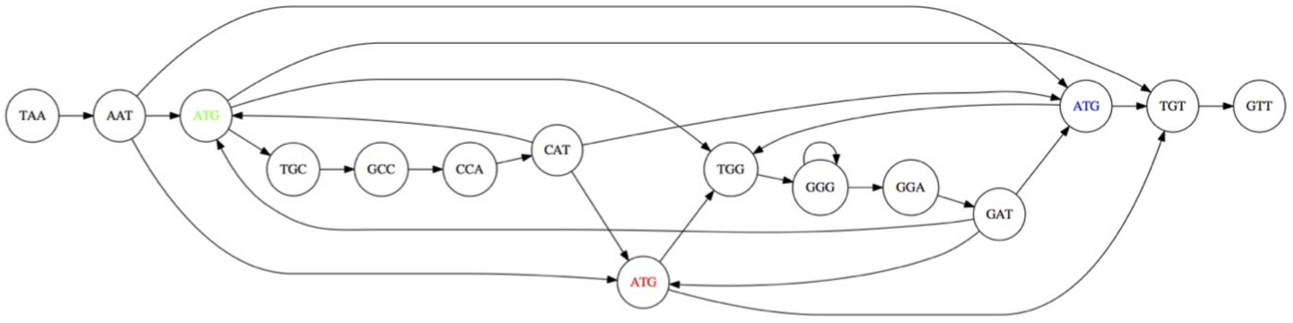
- If the k-mer is too small, all the reads start connecting to each other in the graph (aka “the hairball”)

Salmonella genome assembled with Illumina **100bp** (single-end) reads, and **kmer-size 51**



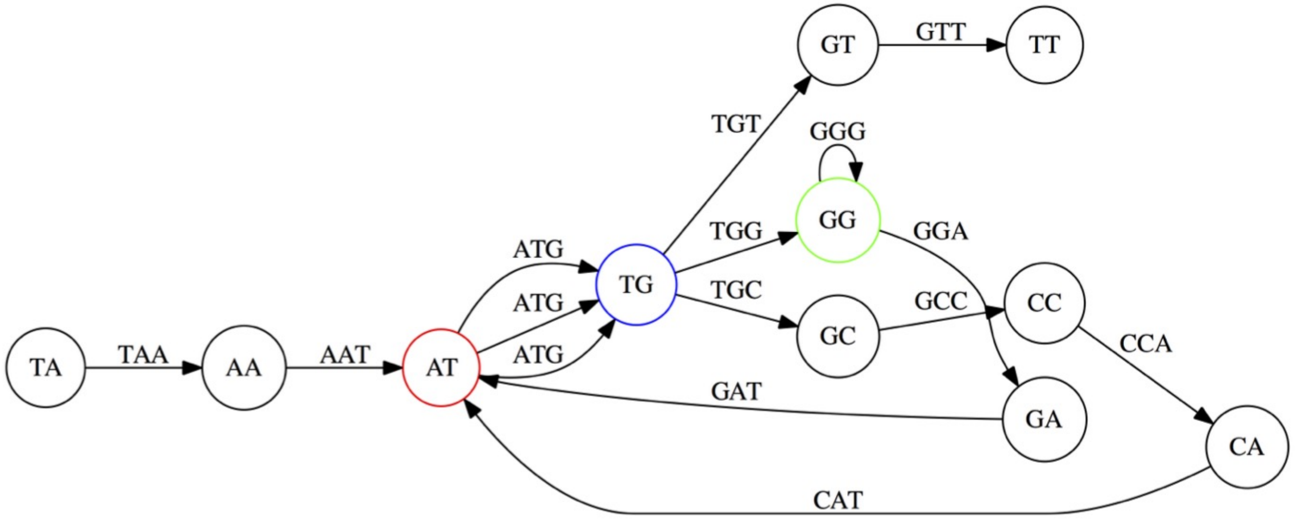
https://github.com/rrwick/Bandage/wiki/Effect-of-kmer-size

Overlap graph



Nodes are reads, edges are overlaps between reads

de Bruijn Graph - same data



Nodes are overlaps, edges are reads

Assembly algorithm comparison

OLC: Overlap-layout-consensus

- Intuitive. First assemblers used this approach
- Looking for overlaps: if suffix of one read has significant similarity with prefix of another, the two reads are connected in the assembly graph.

✓ **Works well with long reads and small amounts of data.**

% **Computationally expensive (many comparisons to make)**

De Bruijn graph

- Counter-intuitive. Yet, the most widely used genome assembly algorithm
- Uses k-mers of reads to generate the assembly graph (“hashing”)

✓ **Works with with short reads and large amounts of data**

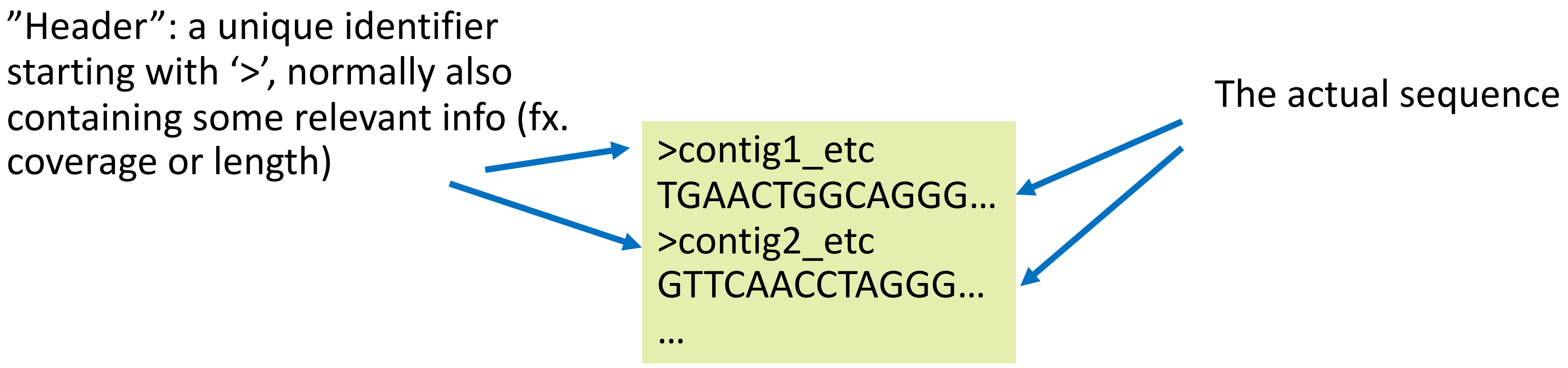
✓ **Computationally efficient**

% **Sensitive to low coverage and sequencing errors**

Genome assembly outcome

contig: Contiguous representation of a genomic region

- The genome assembler will generate a **fasta file**, with the assembled sequence(s)



”Header”: a unique identifier starting with ‘>’, normally also containing some relevant info (fx. coverage or length)

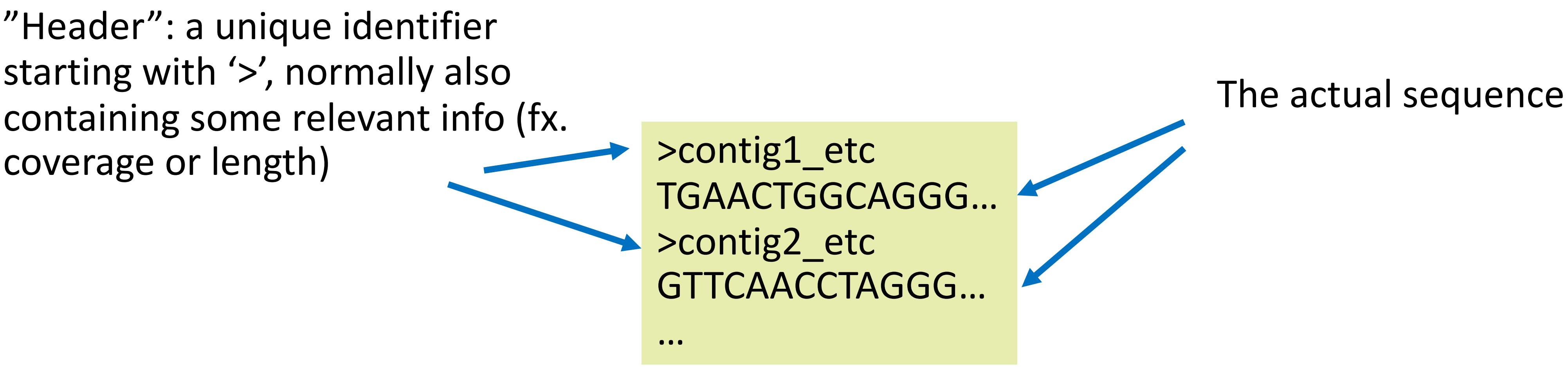
```
>contig1_etc
TGAACTGGCAGGG…
>contig2_etc
GTTCAACCTAGGG…
…
```

The actual sequence

Genome assembly outcome

contig: Contiguous representation of a genomic region

- The genome assembler will generate a **fasta file**, with the assembled sequence(s)



”Header”: a unique identifier starting with ‘>’, normally also containing some relevant info (fx. coverage or length)

The actual sequence

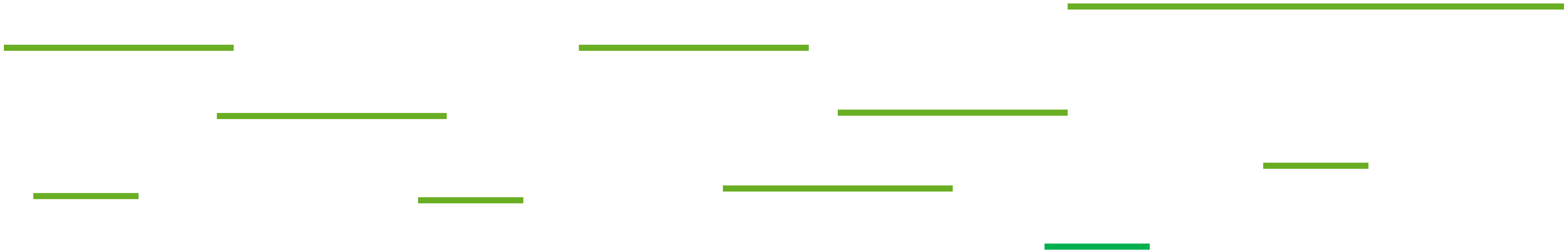
```
>contig1_etc
TGAACTGGCAGGG…
>contig2_etc
GTTCAACCTAGGG…
…
```

- Most assemblers will also generate several additional files, fx.:
  - A record of the settings you used
  - Intermediate result-files
  - Other stats to help you evaluate the quality of your assembly

Genome assembly outcome

Desired result: **1 contig**

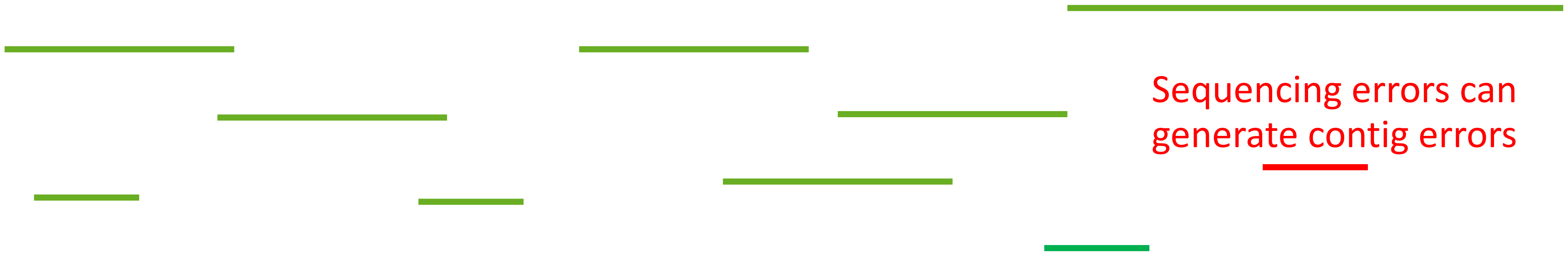
More likely result: **collection of contigs**



Genome assembly outcome

Desired result: **1 contig**

More likely result: **collection of contigs**



Sequencing errors can generate contig errors

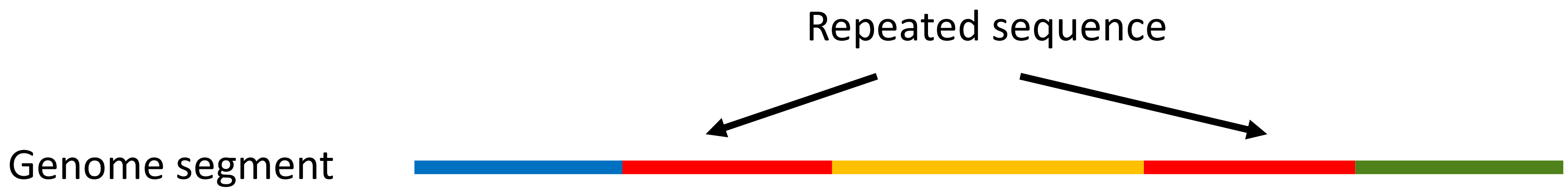
Why not one contig?

- Jigsaw puzzles with lots of blue sky are hard.
- Imagine if you have identical pieces..

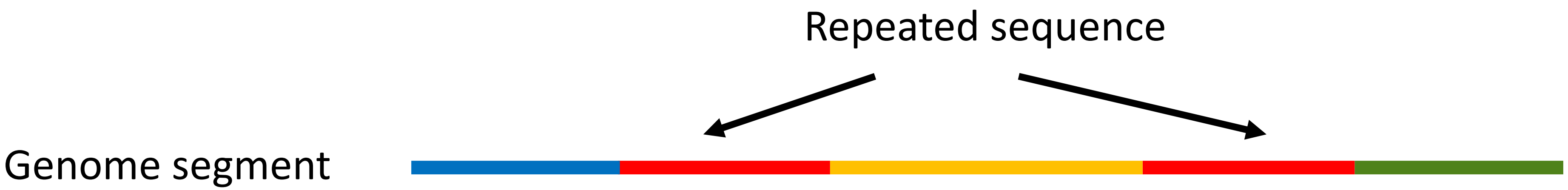
=> **repeats!**

Private photo: Kirsten Ellegaard

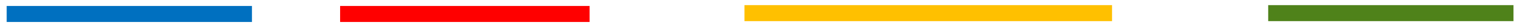
Why are repeats a problem for assembly?



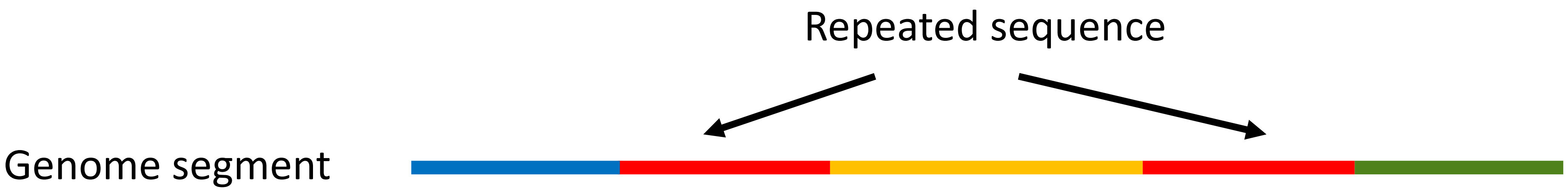
Why are repeats a problem for assembly?



What it looks like in our contig file:



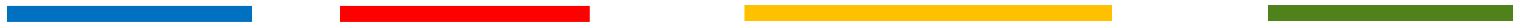
Why are repeats a problem for assembly?



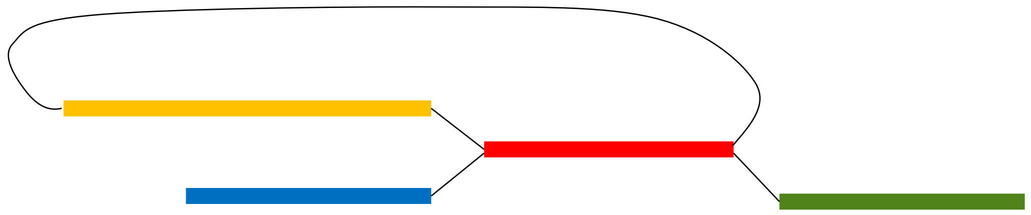
Repeated sequence

Genome segment

What it looks like in our contig file:

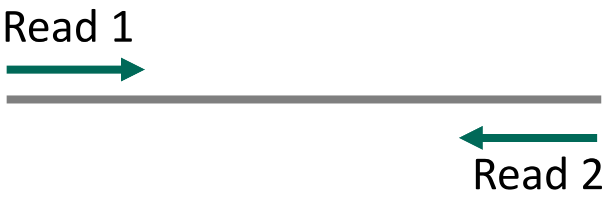


What we know from the gfa files and paths files:



Adding long-distance information can greatly facilitate assembly

Paired-end reads

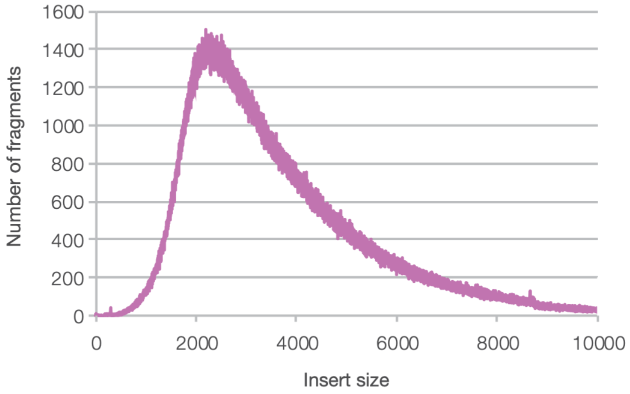


- Each end of a DNA fragment is sequenced
- The reads are known to come from the same DNA fragment, and the approximate fragment size is known
  - Typically 300-500bp

=> Longer contigs

Adding long-distance information can greatly facilitate assembly

Long-insert paired end reads (mate pair)

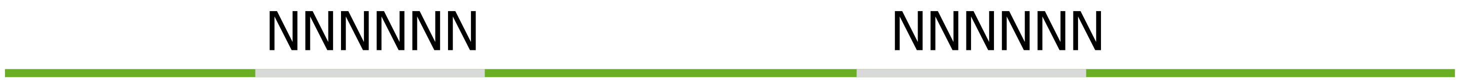


- Mate-pair sequencing can provide even longer distance information
- Several other molecular tricks have been developed
  - Fx. Hi-C, optical maps..
  - Generally, more expensive (and more challenging wetlab procedure)

=> nanopore data

Illumina.com (technote_nextera_matepair_data_processing.pdf)

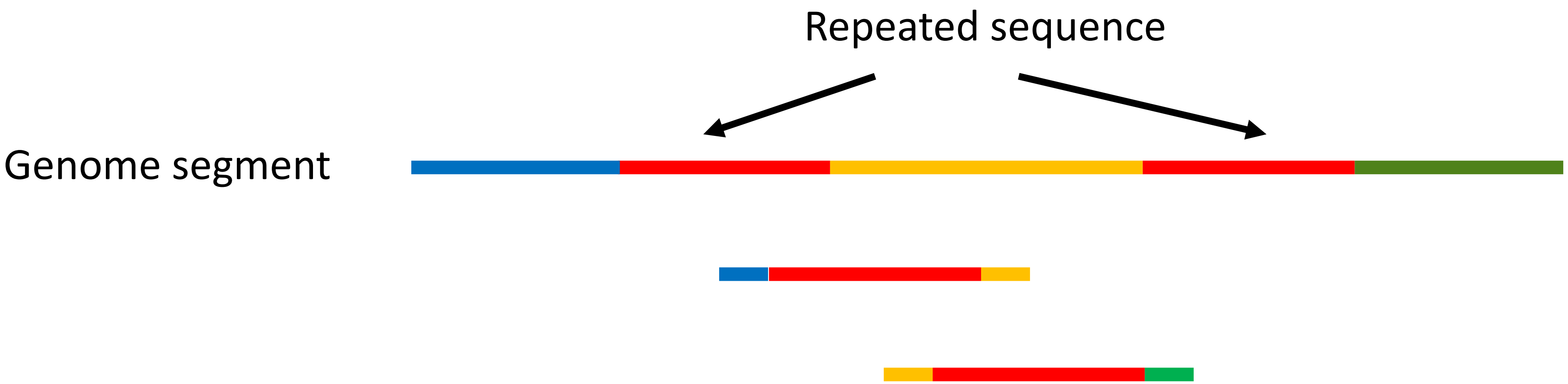
Adding long-distance information can greatly facilitate assembly



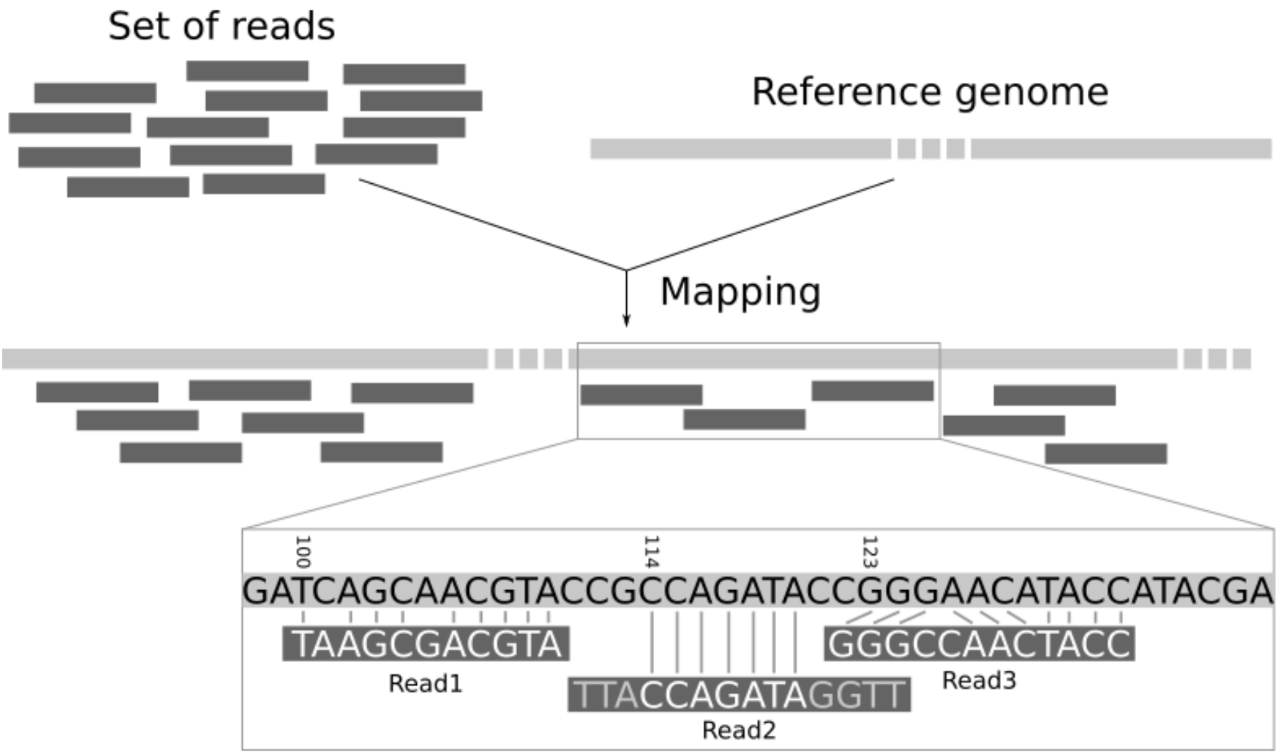
Scaffold: a set of contigs that have been ordered and oriented using long-distance information

Illumina.com (technote_nextera_matepair_data_processing.pdf)

Reads longer than repeats -> problem solved



Reference-based genome assembly

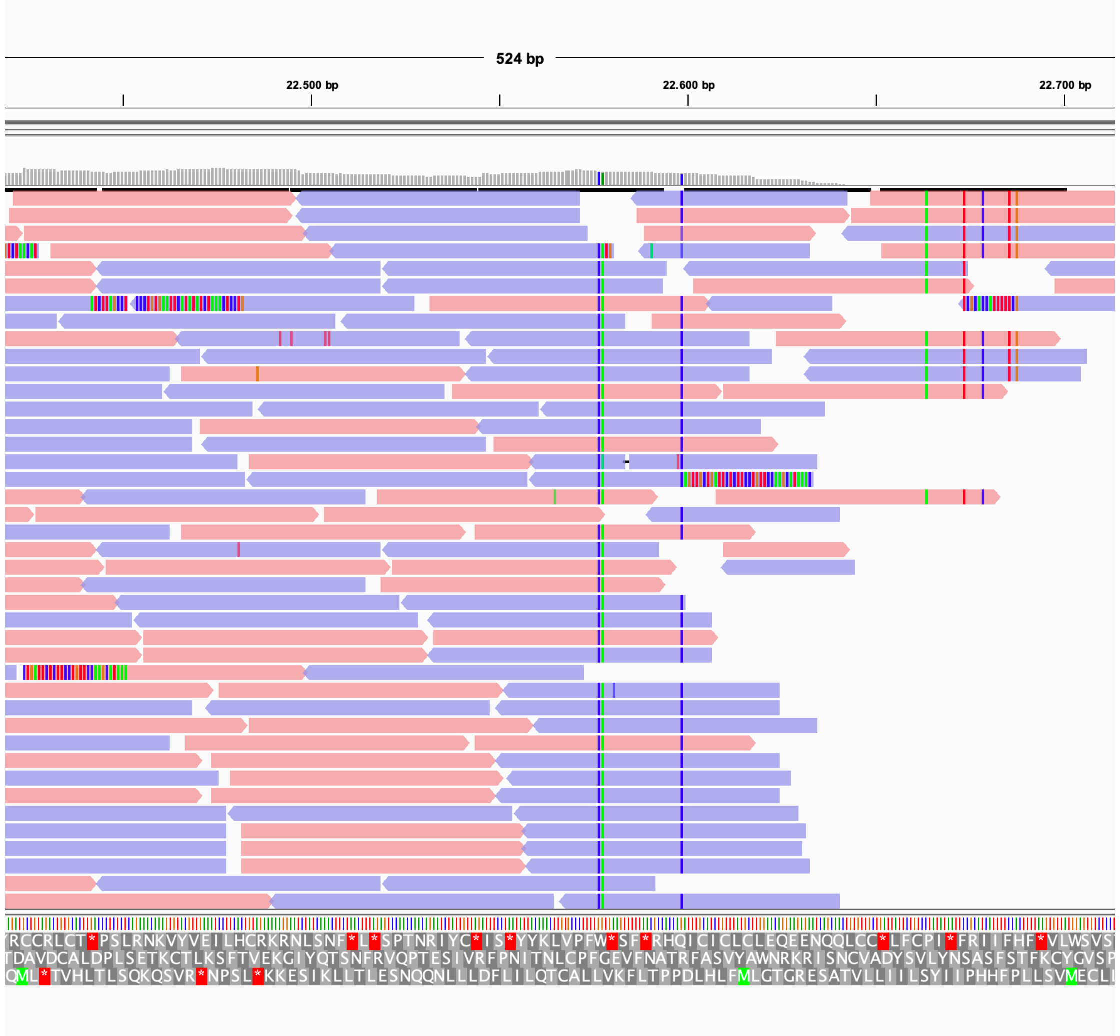


If a reference genome is available, a genome can be contructed:

1. Map reads
2. Determine differences
3. Generate genome

Read-mapping is much easier than assembly!

Minor variations can be reliably identified with reference-based genome assembly

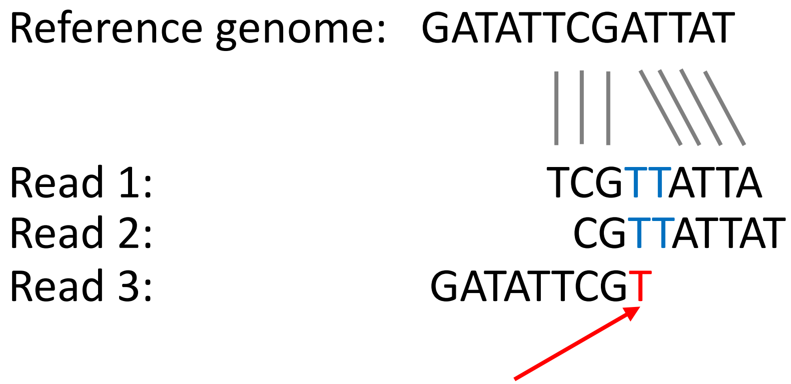

524 bp
22.500 bp
22.600 bp
22.700 bp
RCCRLCT*PSLRNKVYVEILHCRKRNLSNF*L*SPTNRIYC*IS*YYKLVPFW*SF*RHQICICLCLEQEENQQLCC*LFCPI*FRIIFHF*VLWSVS
TDAVDCALDPLSETKCTLKSFTVEKGIYQTSNFRVQPTESIVRFPNITNLCPFGEVFNATRFASVYAWNRKRISNCVADYSVLYNSASFSTFKCYGVSP
QML*TVHLTLSQKQSVR*NPSL*KKESIKLLTLESNQQNLLLDFLILQTCALLVKFLTPPDLHLFMLGTGRESATVLLIILSYIIPHHFPLLSVMECLL

- ✓ SNVs (single-nucleotide variants) are straightforward
- ✓ Higher sensitivity than for genome assembly
- ✓ Non-clonal infections can potentially be determined
- ✓ Contamination from un-related organisms is less of a problem (unless it’s a lot)

Snapshot from IGVviewer, SARS-Cov-2 data

Insertions/deletions are more challenging to call correctly with a reference genome

- Small insertions/deletions can be called
  - They must be small enough that they can be contained within the sequenced reads (alignment on both sides of variant)
  - Depends on read-length (and software/thresholds)



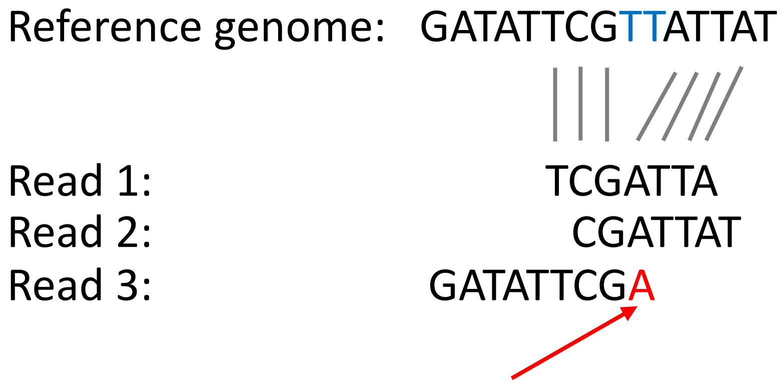
Reference genome: GATATTCGATTAT

Read 1: TCGTTATTA

Read 2: CGTTATTAT

Read 3: GATATTCGT

Treated as SNV (A to T)



Reference genome: GATATTCGTTATTAT

Read 1: TCGATTA

Read 2: CGATTAT

Read 3: GATATTCGA

Treated as SNV (T to A)

Pros and cons of reference-based genome assembly

- Very efficient when there is a close match between your reads and your reference genome (small variations are easily called)
- Tolerates low coverage better than de novo assembly
- Contamination from unrelated organisms can easily be ignored

Pros and cons of reference-based genome assembly

➢ Very efficient when there is a close match between your reads and your reference genome (small variations are easily called)

➢ Tolerates low coverage better than de novo assembly

➢ Contamination from unrelated organisms can easily be ignored

% Quality drops rapidly with genetic distance

% Insertions/deletions will likely cause problems, due to misaligned reads

% You can only assemble genes that are present in your reference genome

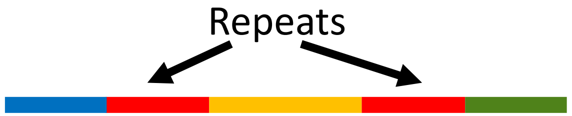
What does a good genome assembly look like?

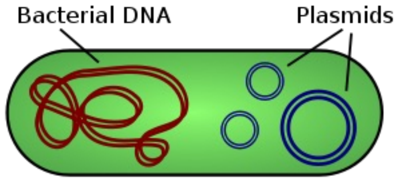
Depends!

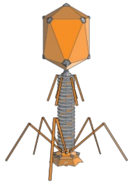
- Adjust your expectations according to read-length and base-calling quality

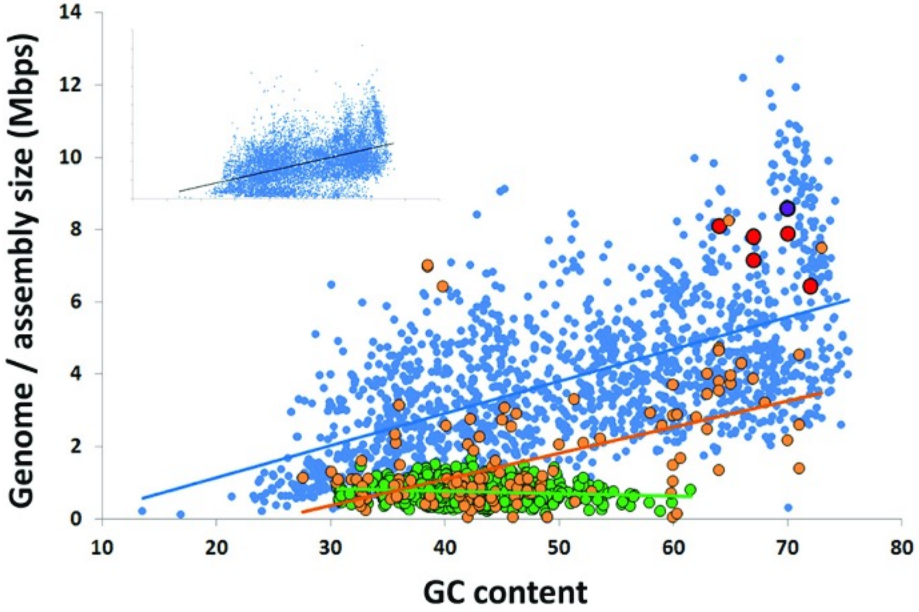
What does a good genome assembly look like?

Some pathogens are harder to sequence than others









https://en.wikipedia.org/wiki
Origin of an alternative genetic code in the extremely small and GC-rich genome of a bacterial symbiont, McCutcheon, J. and McDonald, B., 2009, PLoS Genetics, 5(7):e1000565. CC (https://creativecommons.org/licenses/by/4.0/).

Some pathogens are harder to sequence than others

Wet-lab challenges

- DNA extraction can be difficult for some organisms
- Is the target amenable to cultivation (and thus **isolation**)?
- Alternatively, can we enrich for our target in the sample?
  - Filtering, differential centrifugation, binding of target
  - For small genomes (i.e. viruses), enrichment by PCR may be possible

What if one cant culture or enrich?
Metagenomic sequencing

Metagenomics is an important up-and-coming method, also in public health

- Immuno-compromised patients infected with unusual pathogens
- Diseases where the aetiology is unclear (fx. meningitis)
- Emerging pathogens (The first SARS-CoV-2 genome was sequenced by meta-transcriptomic sequencing of BALF sample from patient)

Genomics is the foundation for metagenomics, many similar principles apply

What if one cant culture or enrich?

Metagenomic sequencing challenges

In generel, much more coverage is required

- Depletion of human DNA
- Contamination is almost unavoidable (include controls)

For *de-novo* metagenome assembly:

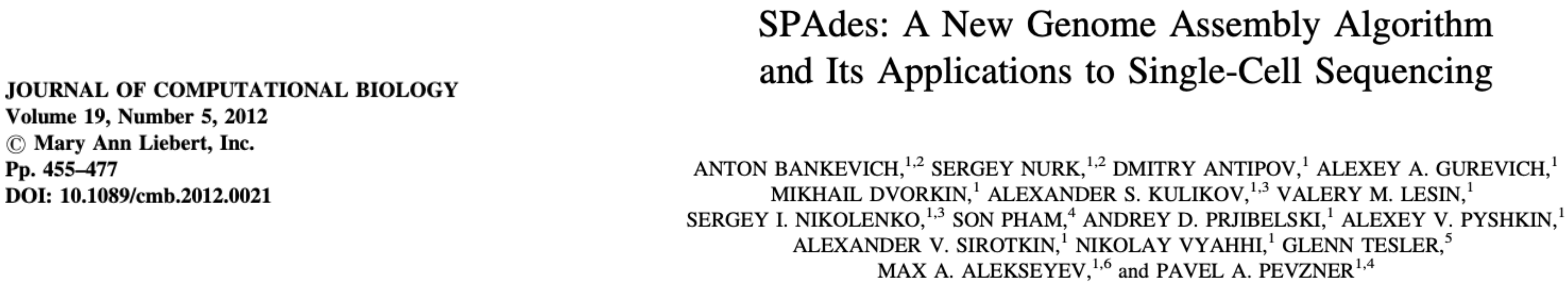
- Multiple closely related targets will cause issues (resembling sequencing errors and repeats)
- Contigs must be binned (assigned to organism bins). Not trivial.

As for genomics, using a database of reference genomes will greatly increase sensitivity

- Consider depth and breadth of coverage, and alignment identity

Practical intro: Short-read genome assembly

About SPAdes



JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 19, Number 5, 2012
© Mary Ann Liebert, Inc.
Pp. 455–477
DOI: 10.1089/cmb.2012.0021

SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing

ANTON BANKEVICH,[1,2] SERGEY NURK,[1,2] DMITRY ANTIPOV,[1] ALEXEY A. GUREVICH,[1] MIKHAIL DVORKIN,[1] ALEXANDER S. KULIKOV,[1,3] VALERY M. LESIN,[1] SERGEY I. NIKOLENKO,[1,3] SON PHAM,[4] ANDREY D. PRJIBELSKI,[1] ALEXEY V. PYSHKIN,[1] ALEXANDER V. SIROTKIN,[1] NIKOLAY VYAHHI,[1] GLENN TESLER,[5] MAX A. ALEKSEYEV,[1,6] and PAVEL A. PEVZNER[1,4]

- First published in 2012
- Continously updated
- One of the most widely used genome assemblers for short-read sequencing data

Why did SPAdes get so popular?

- Multi-sized de bruijn graph
  - Variable coverage

Why did SPAdes get so popular?

- Multi-sized de bruijn graph
  - Variable coverage

- Better use of paired-end reads
  - Variable insert-size

Why did SPAdes get so popular?

- Multi-sized de bruijn graph
    - Variable coverage

- Better use of paired-end reads
    - Variable insert-size

- Better read error correction
    - Probabilistic (BayesHammer)

Why did SPAdes get so popular?

- Multi-sized de bruijn graph
  - Variable coverage

- Better use of paired-end reads
  - Variable insert-size

- Better read error correction
  - Probabilistic (BayesHammer)

- Good computational performance

Why did SPAdes get so popular?

- ➢ Multi-sized de bruijn graph
  - Variable coverage
- ➢ Better use of paired-end reads
  - Variable insert-size
- ➢ Better read error correction
  - Probabilistic (BayesHammer)
- ➢ Good computational performance
- ➢ Ease of installation and usage

Acknowledgements

The creation of this training material was commissioned by ECDC to Statens Serum Institut (SSI) with the direct involvement of Kirsten Ellegaard