

Genome assembly

Bioinformatics Applications (PLPTH813)

Sanzhen Liu

4/18/2017

Outline

- Genome assembly: concept
- Assembly algorithms: OLC/De Bruijn graph
- Error correction (Kmer counts)
- Assembly strategy to cope with the repeat problem (mate-pair reads, long reads)
- Assembly evaluation (N50, comparison to a reference genome)

Whole genome shotgun (WGS) sequencing

chromosome



sequencing
reads

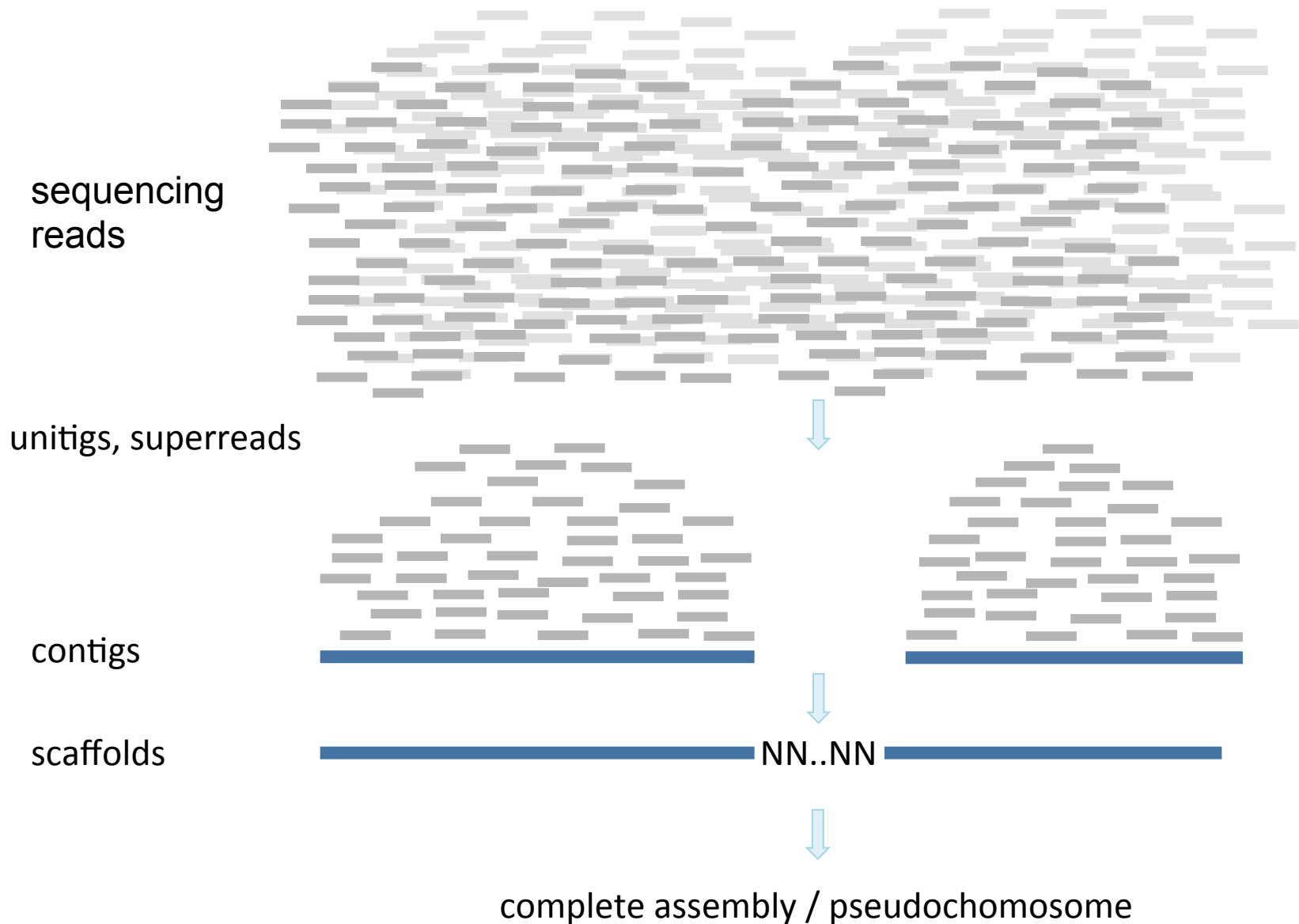


Complexity of genome assembly



Algorithm can solve 10,000
Piece jigsaw in 24 Hours

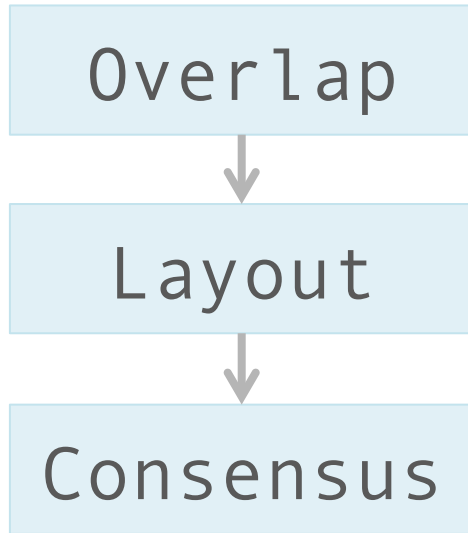
Genome assembly



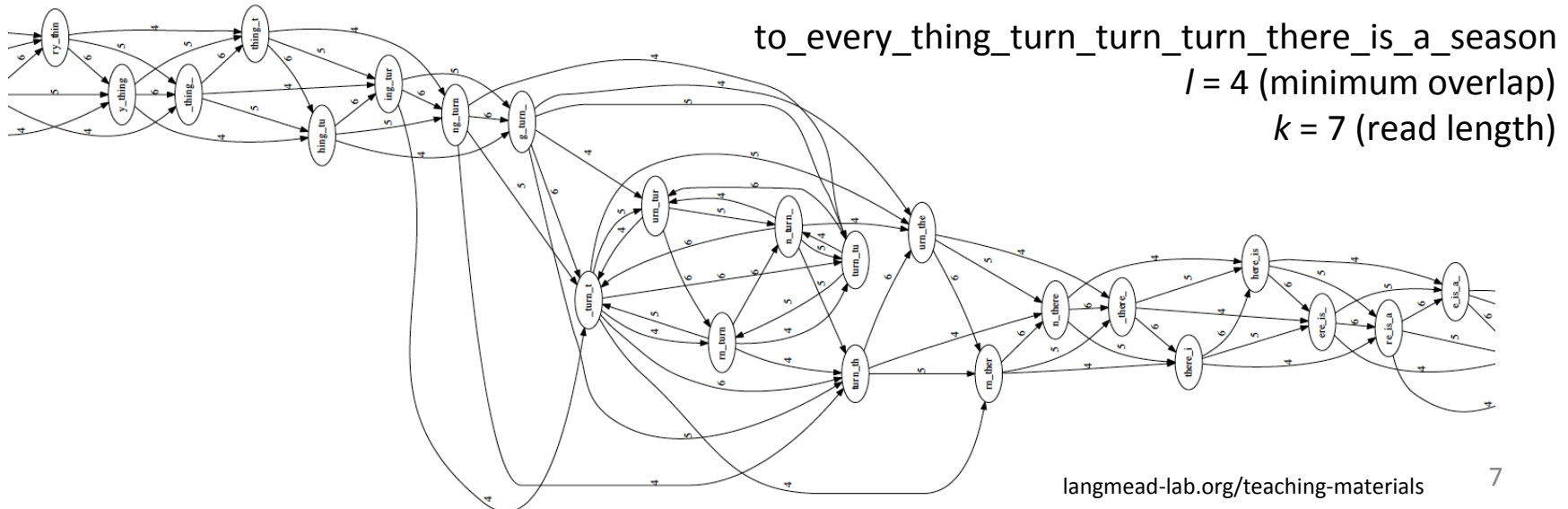
Assembly algorithms

- **Overlap layout consensus (OLC)**
 1. all-to-all pair-wise read alignments
 2. Construct graph based on overlaps
 3. Trace paths for the assembly
- **De Bruijn graph**
 1. Determine k-mer from reads
 2. Construct k-mer graph
 3. Trace paths for the assembly

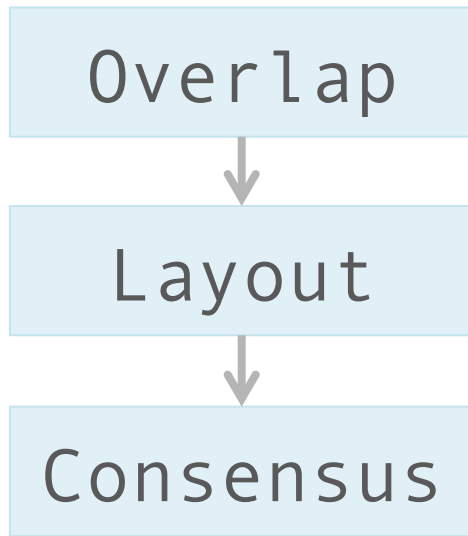
Overlap layout consensus (I)



- Overlap graph
 1. identifies all pairs of reads that overlap **sufficiently well**
 2. organizes the overlapping information into a graph containing a **node** for every read and an **edge** between any pair of reads that overlap each other.

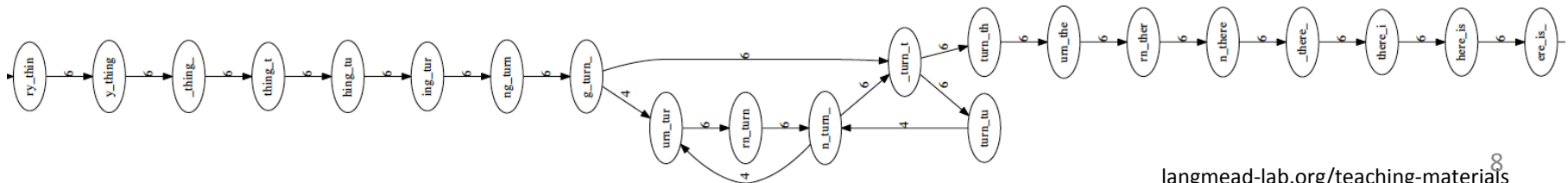
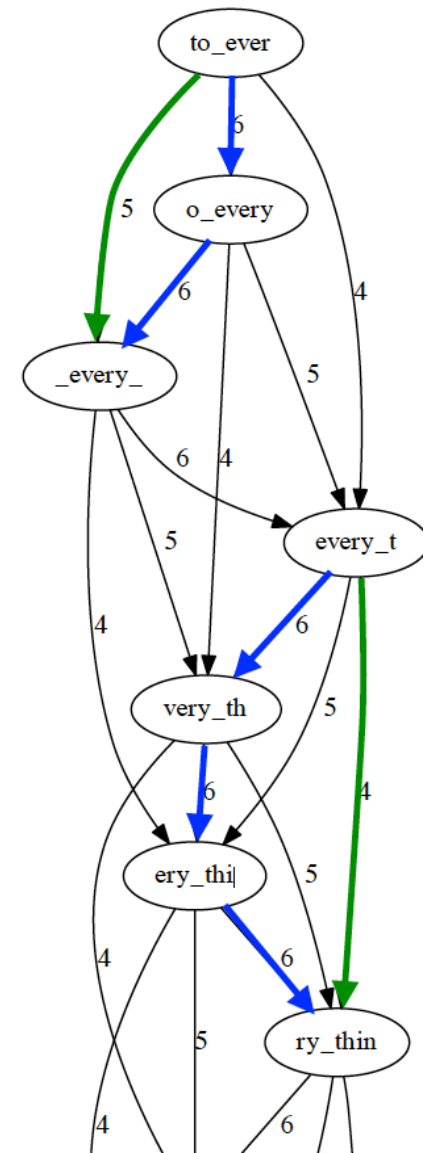


Overlap layout consensus (II)

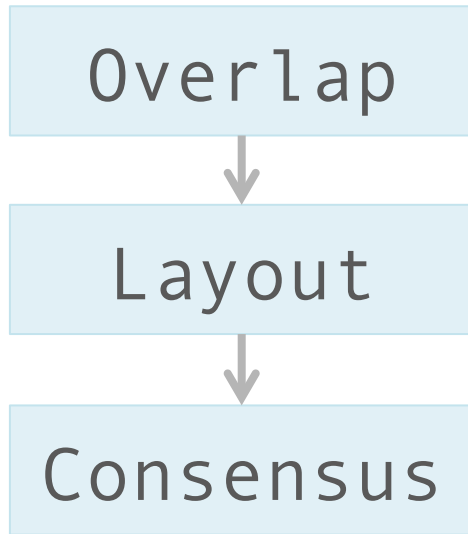


transitive edges: edges that skip one or more nodes

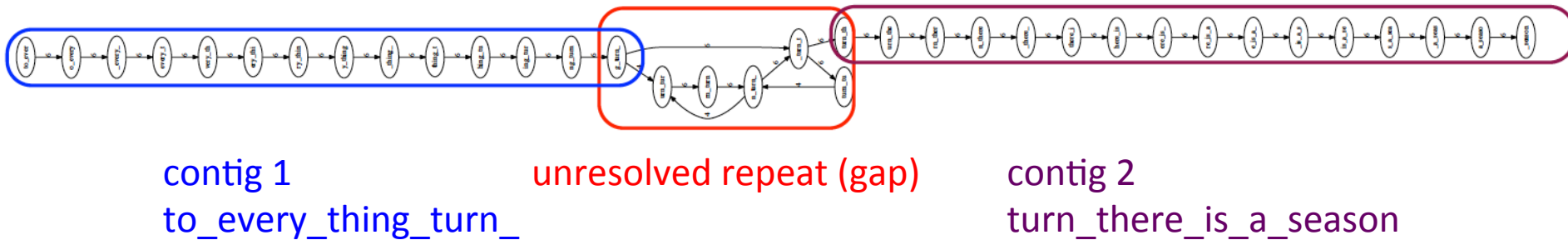
- Layout (**string graph**)
- 1. Simplifies the global overlap graph by removing redundant information (transitive edges)
- 2. Emit contigs corresponding to the non-branching stretches



Overlap layout consensus (III)

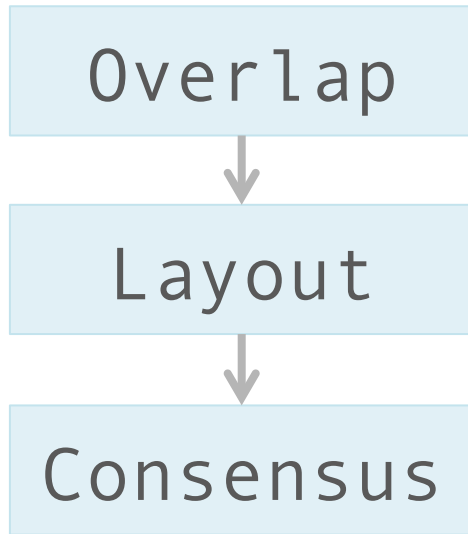


- Layout
 1. Simplifies the global overlap graph by removing redundant information
 2. Emits contigs corresponding to the non-branching paths



to every thing turn turn turn there is a season

Overlap layout consensus (IV)



- Consensus

Extract consensus sequences (major votes) based on the alignment of reads that make up a contig

```
TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

↓ ↓ ↓ ↓ ↓

```
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

The diagram illustrates the extraction of a consensus sequence from five overlapping reads. The reads are aligned, and red arrows point from the bases that are identical across all reads (the majority vote) to the final consensus sequence. The consensus sequence is TAGATTACACAGATTACTGACTTGATGGCGTAA CTA.


OLC drawbacks

- Identifying all-to-all overlaps is a slow procedure, especially when read number is millions or billions.
- Overlap graph is big when read number is huge. One node per read, and in practice the number of edges grows superlinearly with the number of reads
- The computational complexity limits its application

De Bruijn graph – k-mer

- De Bruijn graph assemblers model the relationship between **exact substrings** of length k (k-mer) extracted from input reads.

(k=3)
reads A T G G C G T



k-mer (k=3)

1 . ATG

2 . TGG

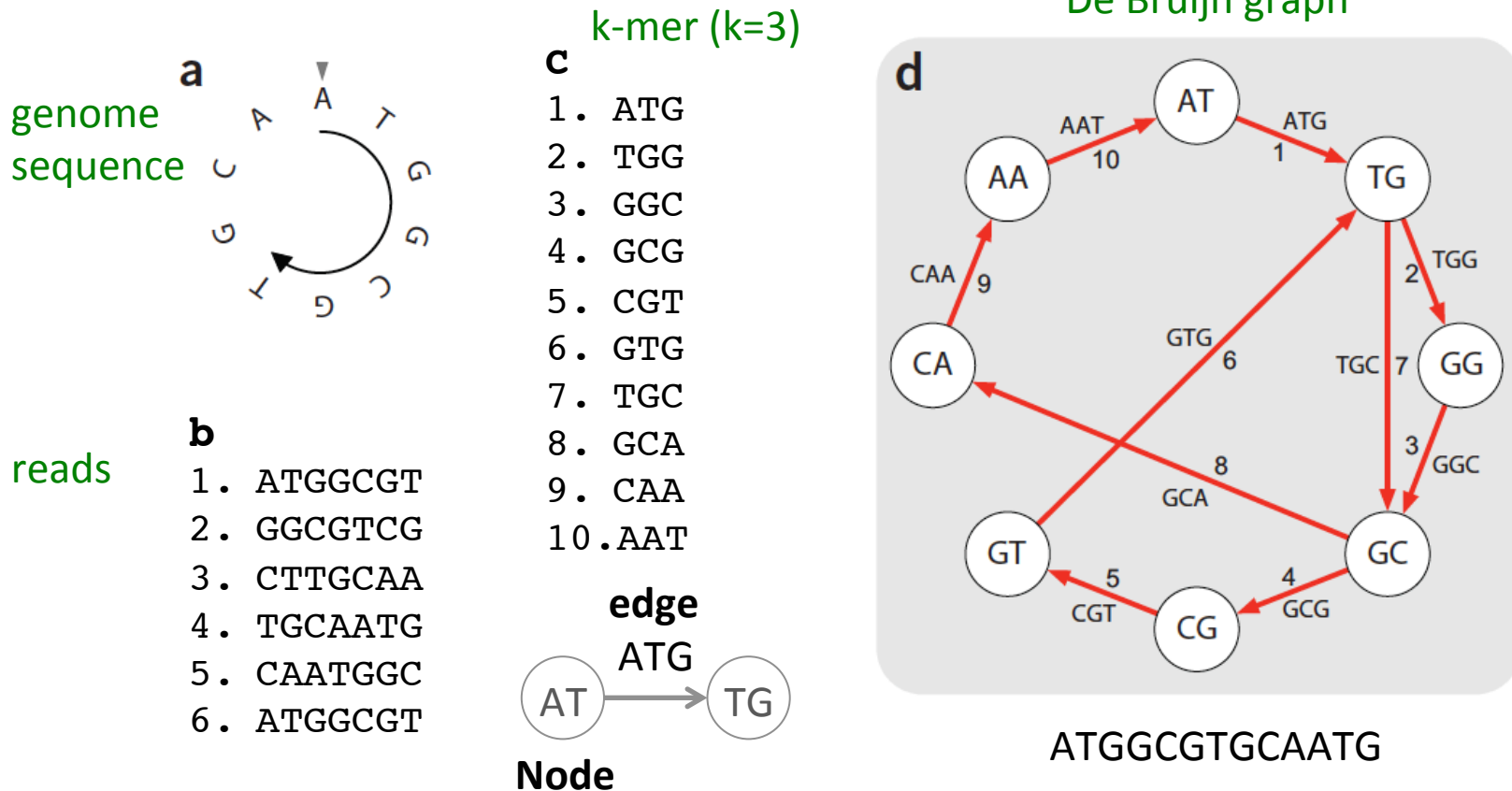
3 . GGC

4 . GCG

5 . CGT

De Bruijn graph - assembly

- De Bruijn graph assemblers model the relationship between **exact substrings** of length k (k-mer) extracted from input reads.



Sequencing errors complicate assemblies

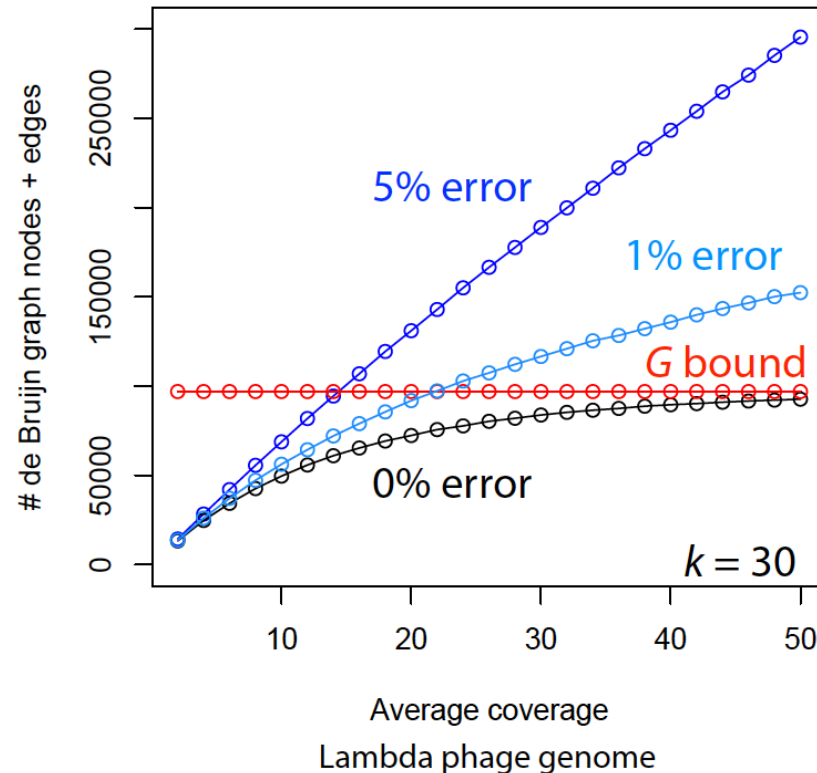
The complexity of De Bruijn graph grows when reads contain errors.

Reads

TGCGTGA
TGCGTGA
TGCGTGA
TG**G**GTGA

k-mer count profile (k=5)

TGCGT
GCGTG
CGTGA
TG**G**GT
GGGTG
GGTGA



G: genome

langmead-lab.org/teaching-materials

Sequence error correction before graph construction is a critical step for De Bruijn graph assembly.

Error identification

Reads (depth = 10)

TGCGTGATACG
TGCGTGATACG
TG**G**GTGATACG
TGCGTGATACG
TGCGTGATACG
TGCGTGATACG
TGCGTGATACG
TGCGTGATACG
TGCGTGATACG
TGCGTGATACG
TGCGTGATACG

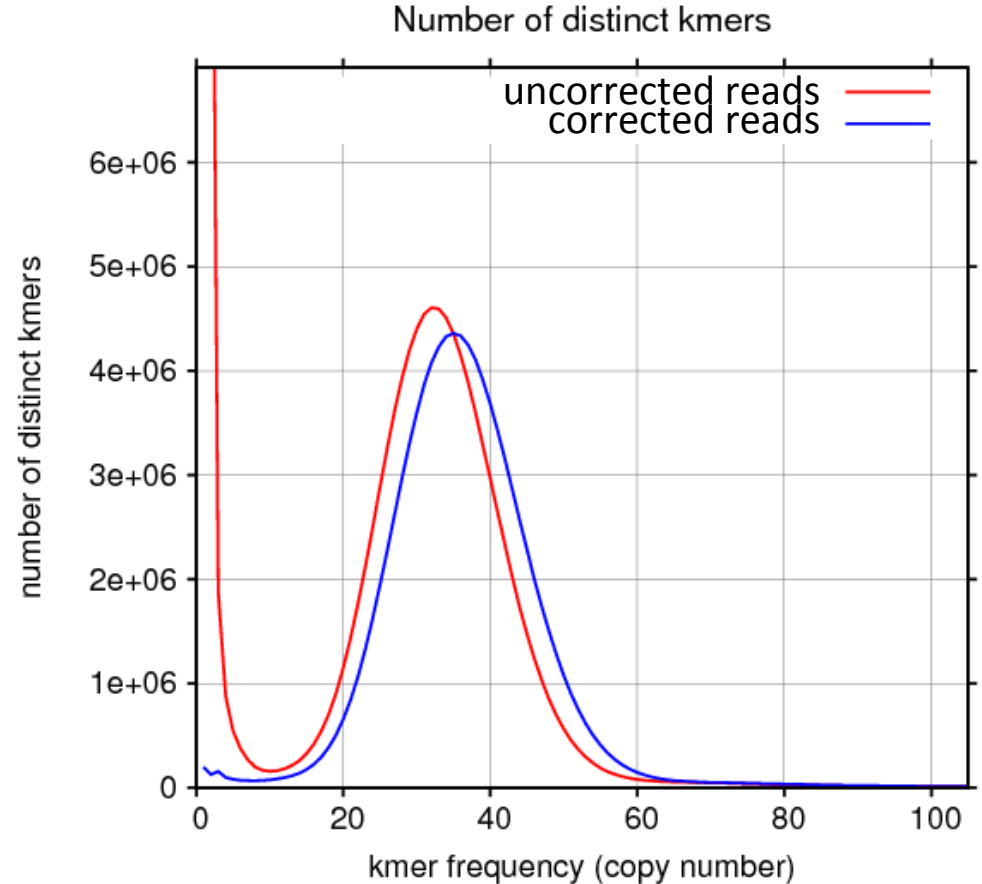
k-mer count profile (k=5)

TG G GT	1
G GTG	1
G GTGA	1
TGCGT	9
GCGTG	9
CGTGA	9
GTGAT	10
TGATA	10
GATAC	10
ATACG	10

k-mer count profile indicates where errors are

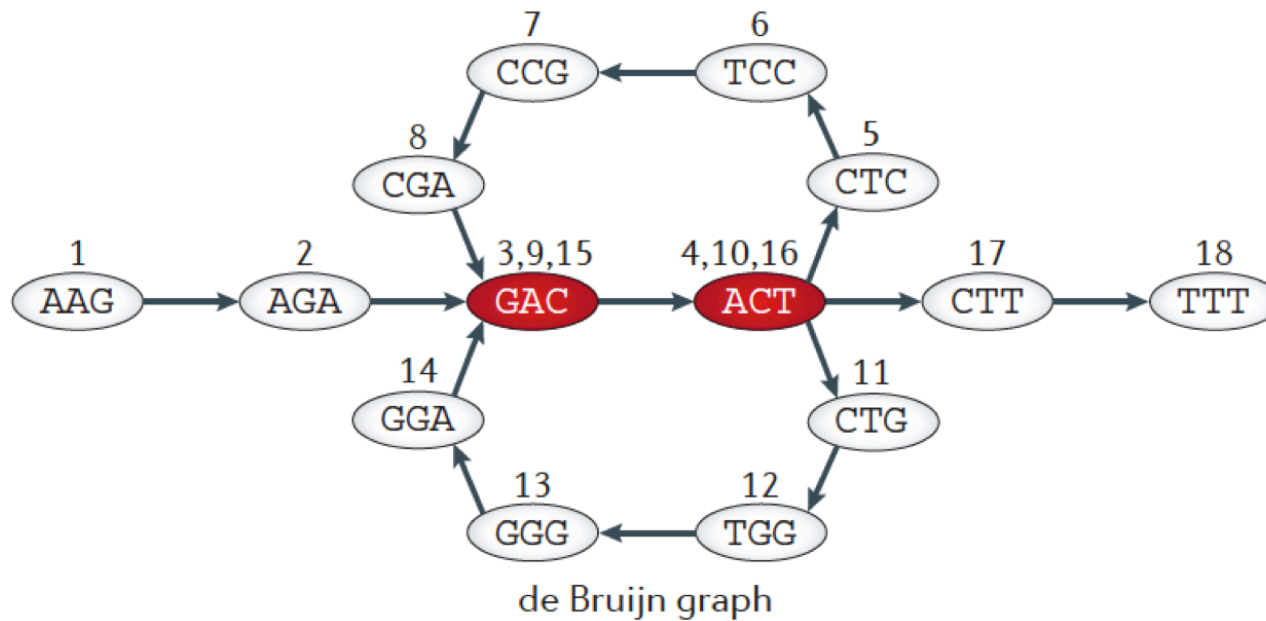
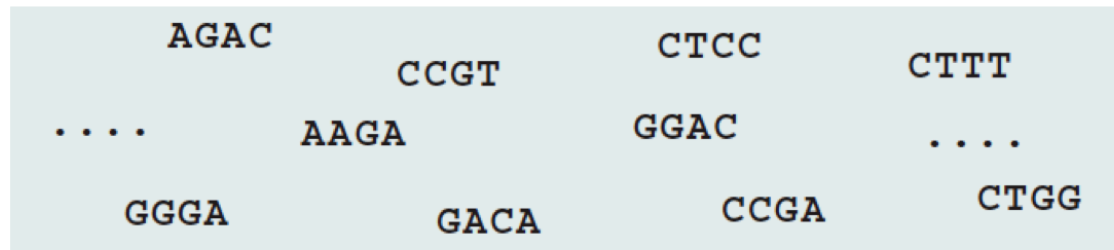
Error correction

Essentially, the error correction algorithm removes k-mers with a few copies and correct reads carrying these errors.



from Allpaths-LG

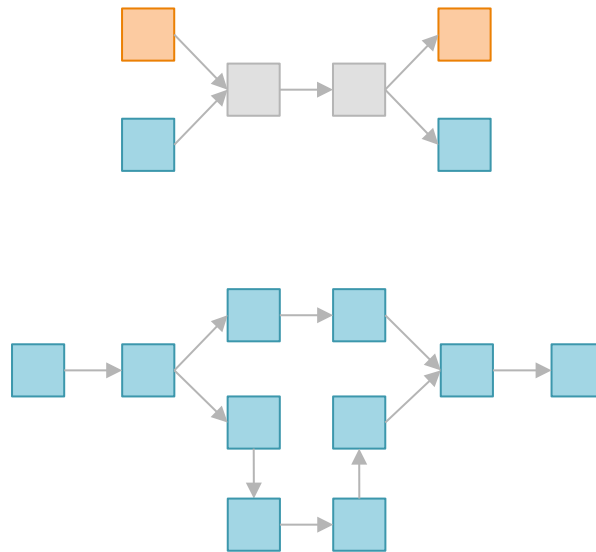
Repeats



AA**GACT**CC**GACT**GG**GACT**TT

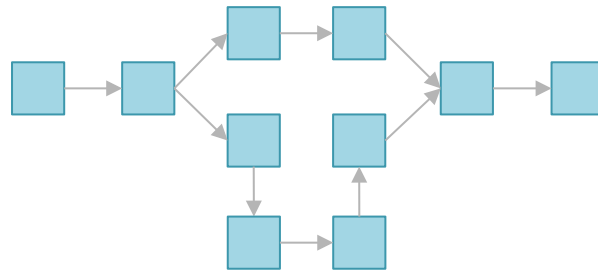
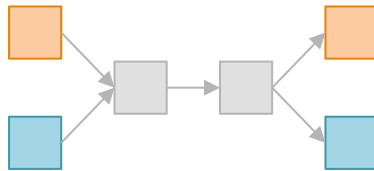
Unresolved repeats produce gaps

Long repeats, especially for those whose lengths are longer than read length, and high-copy repeats are challenging to resolve. If no other information can be used to assist in resolving repeats, gaps will be introduced.



Question

What strategies can be used to resolve repeats?



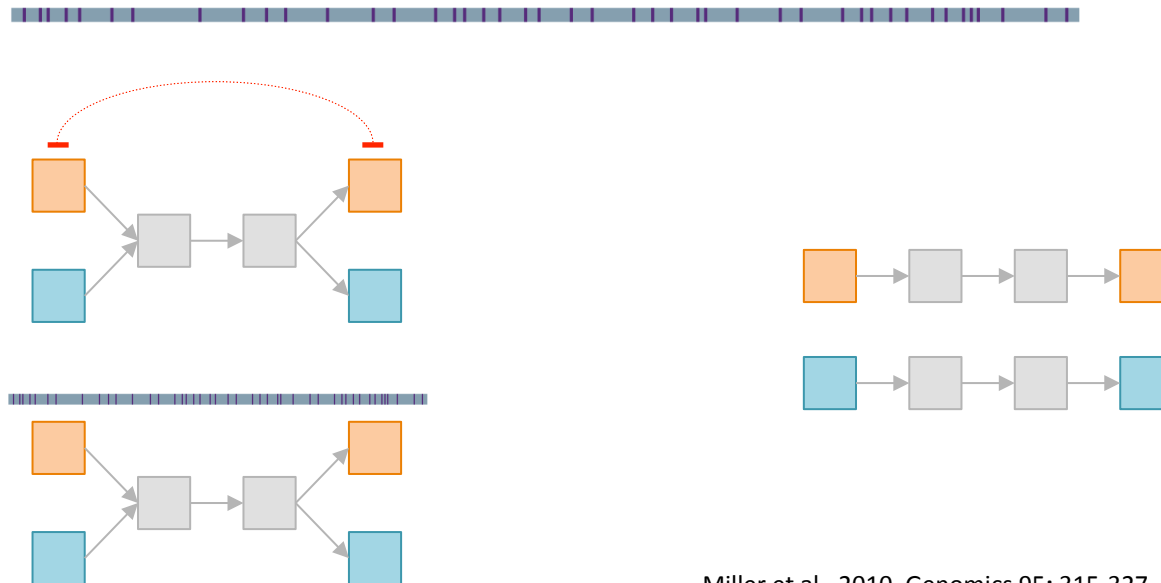
Strategies to resolve repeats

Mate-pair reads, long reads (e.g., PacBio), genetic map, physical map, and Hi-C data can resolve some repeats

Illumina mate-pair (MP) reads



PacBio long reads (average 4-8 kbp, longest over 30 kbp)



Miller et al., 2010. Genomics 95: 315-327

Other assembly gaps

- Low sequence coverage (depth)
- Sequencing biases (e.g., regions recalcitrant to sequencing)

```
CGGATCTGCGTGATACGGAATAGCCTAGCA
GATCTGCGTGATACGGAAT
ATCTGCGTGATACGGAATA
TCTGCGTGATACGGAATAG
CTGCGTGATACGGAATAGC
TGCGTGATACGGAATAGCC
CGTGATACGGAATAGCCT
6 (coverage; depth)
```

Recalcitrant genomic regions:

1. extremely high GC or AT
2. complicated secondary structure

Preferred genome sequencing reads:

1. Long reads
2. High-quality reads
3. High sequencing depth (>30x)
4. No sequencing biases

Assembly statistics – N50

- N50: A statistic used for assessing the contiguity of a genome assembly.
- The contigs in an assembly are sorted by size and added, starting with the largest. The contig N50 is the size of the contig that makes the total greater than or equal to 50% of the total contig size.
- **OR:** The contig N50 is the length of the smallest contig in the set that contains the fewest (largest) contigs whose combined length represents at least 50% of the assembly.

Assembly statistics – NG50

Problem of N50:

N50 values of the assemblies with significantly different total assembly space (lengths) are usually not comparable. Even if the same data set is used for the assembly.

50,000	50,000
30,000	40,000
10,000	30,000
	20,000
	10,000

NG50: The NG50 statistic is the same as N50 except that it is 50% of the known or estimated genome size. This allows for meaningful comparisons between different assemblies.

An example to determine N50 and NG50

- Genome size: 5 Mbp
- 20 contigs

contig N50? 360 kbp

contig NG50? 356 kbp

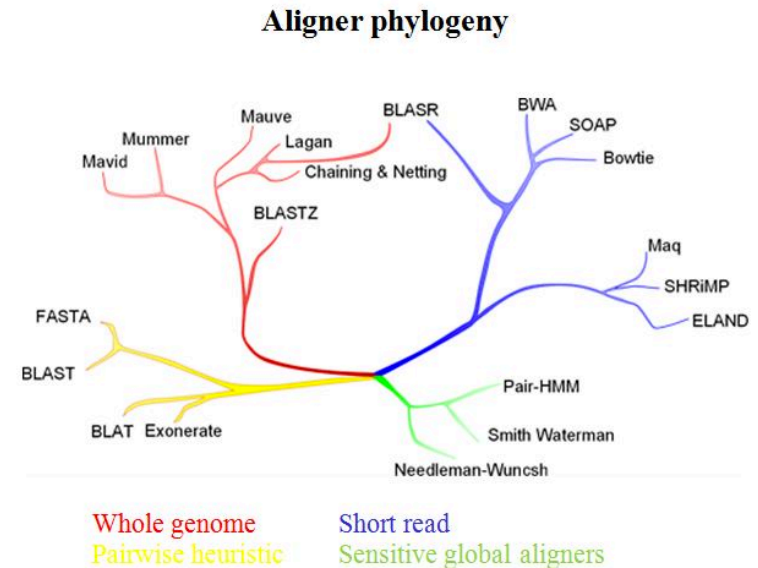
ctg_ID	ctg_size (kbp)	Accumulated (kbp)	% ASM	% genome
1	615	615	13%	12%
2	515	1,130	25%	23%
3	512	1,642	36%	33%
4	450	2,092	46%	42%
5	360	2,452	53%	49%
6	356	2,808	61%	56%
7	310	3,118	68%	62%
8	201	3,319	72%	66%
9	189	3,508	76%	70%
10	186	3,694	80%	74%
11	160	3,854	84%	77%
12	150	4,004	87%	80%
13	120	4,124	90%	82%
14	102	4,226	92%	85%
15	95	4,321	94%	86%
16	86	4,407	96%	88%
17	82	4,489	98%	90%
18	54	4,543	99%	91%
19	32	4,575	100%	92%
20	21	4,596	100%	92%
Total	4,596			

Assembly evaluation – QUAST

- QUAST: quality assessment tool for genome assemblies
 1. No. of contigs: the total number of contigs in the assembly.
 2. Largest contig: the length of the largest contig in the assembly.
 3. Total length: total assembly length
 4. Nxx (N50), NGxx (NG50)
 5. Using a reference genome, No. of misassemblies, No. INDEL per 100 kbp, and No. of mismatches per 100 kbp are reported
 6. NA50: splits contigs that contain "misassemblies" and "unaligned" sequences and uses the new contigs for N50 calculation (NA50)

An alignment tool behind QUAST - Nucmer

- **Nucmer** is a user-friendly alignment script of the MUMmer software package for DNA sequence alignment.
- For instance, a very common use for Nucmer is to determine the position and orientation of a set of sequence contigs in relation to a reference genome sequence



Citations

- Berger, B., J. Peng and M. Singh, 2013 Computational solutions for omics data. *Nature Reviews Genetics* 14: 333-346.
- Compeau, P. E. C., P. A. Pevzner and G. Tesler, 2011 How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology* 29: 987-991.
- Nagarajan, N., and M. Pop, 2013 Sequence assembly demystified. *Nature Reviews Genetics* 14: 157-167.
- Miller, J. R., S. Koren and G. Sutton, 2010 Assembly algorithms for next-generation sequencing data. *Genomics* 95: 315-327.
- Gurevich, A., V. Saveliev, N. Vyahhi and G. Tesler, 2013 QUAST: quality assessment tool for genome assemblies. *Bioinformatics* 29: 1072-1075.