Assembly

Hélène Touzet
helene.touzet@univ-lille.fr
CNRS, Bonsai, CRIStAL

DNA-seq

raw reads

trimmed and cleaned reads

- reference genome → **mapping**
  find location of the read and align them with respect to the reference
- no reference genome → **assembly**
  reconstruct the initial sequence

The assembly problem

Reconstruct this

From these

```
                CTAGGCCCTCAATTTTT
              CTCTAGGCCCTCAATTTTT
            GGCTCTAGGCCCTCATTTTTT
         CTCGGCTCTAGCCCCTCATTTT
      TATCTCGACTCTAGGCCCTCA
      TATCTCGACTCTAGGCC
  TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
```

Courtesy of Ben Langmead (Johns Hopkins University)

Why assembling reads ?

- annotation of genomes
  - discovery of new genes
  - gene order, structural variants
  - noncoding regions
  - evolutionnary genomics, phylogenomics
- transcriptome
  - reconstruction of transcripts
  - identification of alternative transcripts
- metagenomics
  - identification of species

The assembly problem

Reconstruct this

From these

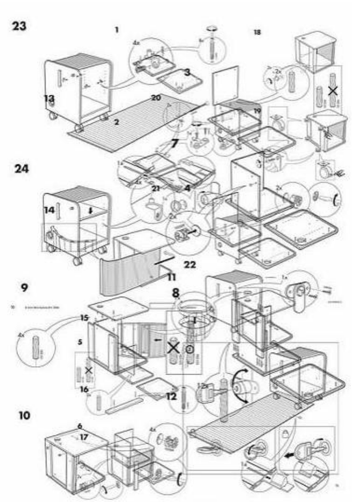
```
                    CTAGGCCCTCAATTTTT
                  CTCTAGGCCCTCAATTTTT
                GGCTCTAGGCCCTCATTTTTT
             CTCGGCTCTAGCCCCTCATTTT
          TATCTCGACTCTAGGCCCTCA
          TATCTCGACTCTAGGCC
      TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
```

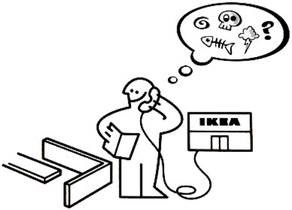
CTAGGCCCTCAATTTTT
GGCGTCTATATCT
CTCTAGGCCCTCAATTTTT
TCTATATCTCGGCTCTAGG
GGCTCTAGGCCCTCATTTTTT
CTCGGCTCTAGCCCCTCATTTT
TATCTCGACTCTAGGCCCTCA
GGCGTCGATATCT
TATCTCGACTCTAGGCC
GGCGTCTATATCTCG

From these

Reconstruct this

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT





Some assembly is required

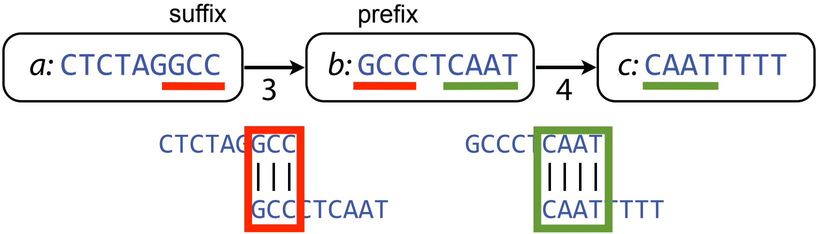
How to assemble reads ?

Historical perspective

Key features : overlaps between reads

How to assemble reads ?

Historical perspective



Key features : overlaps between reads

R_1 C T G A G A A C C T G T

R_2 C C T G T A A G A T

R_3 C T G T A C C T

R_4 G A T C T G A

R_1 C T G A G A A C C T G T G A T C T G A R_4

R_3 C T G T A C C T R_2 C C T G T A A G A T

C T G T A C C T G A G A A C C T G T A A G A T C T G A

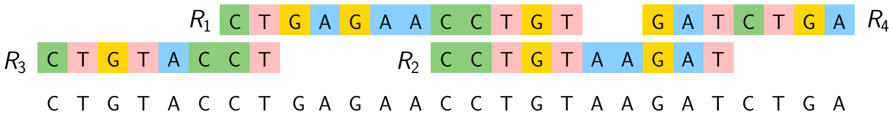
Length of the assembly : 27

R_1 C T G A G A A C C T G T C C T G T A A G A T R_2

R_4 G A T C T G A R_3 C T G T A C C T

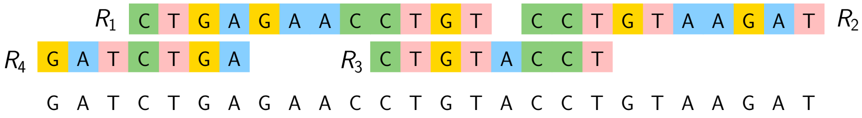
G A T C T G A G A A C C T G T A C C T G T A A G A T

Length of the assembly : 26



Length of the assembly : 27

joining together the reads in decreasing order of the quality of their overlaps



Length of the assembly : 26

trying to maximize the total length of read overlaps

Matrix of all possible overlaps

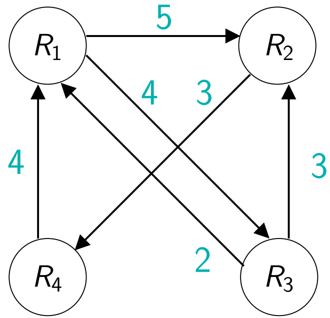
$\nearrow$	R_1	R_2	R_3	R_4
R_1		5	4	0
R_2	0		0	3
R_3	2	3		0
R_4	4	0	0	

Length of the longest suffix of R_i
which is also a prefix of R_j

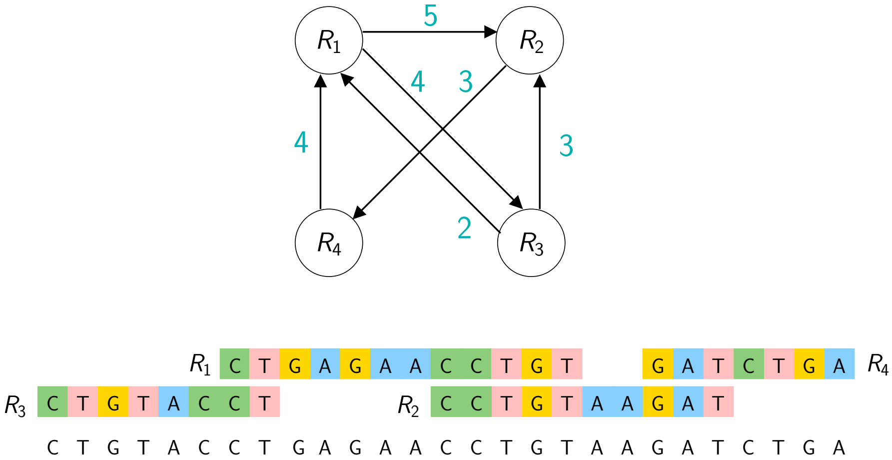
Formalisation as a graph

- Nodes : reads
- Arcs : overlaps between two reads (directed, suffix $\rightarrow$ prefix)
- Weights on the arcs : length of the overlap

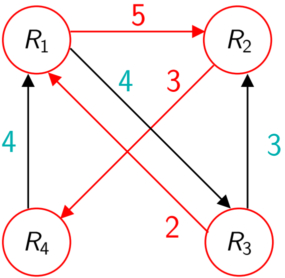
$\nearrow$	R_1	R_2	R_3	R_4
R_1		5	4	0
R_2	0		0	3
R_3	2	3		0
R_4	4	0	0	

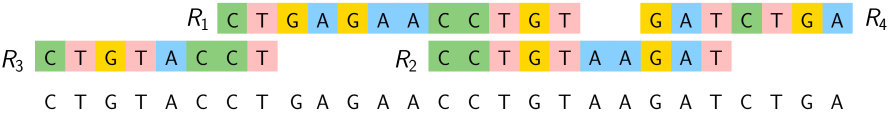


Paths in the graph

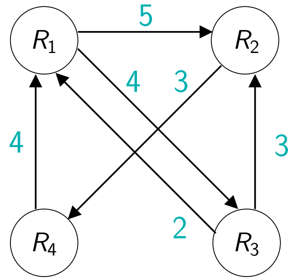


Paths in the graph



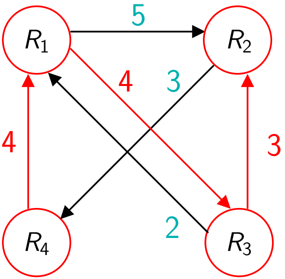


Paths in the graph




```
              R1   C T G A G A A C C T G T   C C T G T A A G A T  R2
R4  G A T C T G A           R3  C T G T A C C T
    G A T C T G A G A A C C T G T A C C T G T A A G A T
```

Paths in the graph



R_1 C T G A G A A C C T G T   C C T G T A A G A T R_2

R_4 G A T C T G A R_3 C T G T A C C T

G A T C T G A G A A C C T G T A C C T G T A A G A T

Paths in the graph

Assembly

- path that visits every node of the graph exactly once
  (Hamiltonian path)

Choice of the path

- Greedy assembly
- Maximal sum of weigths
  Shortest common superstring
- ...

several paths = several assemblies

Overlap assemblies in real life

- risk of contamination
- existence of sequencing errors
- existence of repeats
- diploid and polyploid genomes
- low coverage or unevent coverage

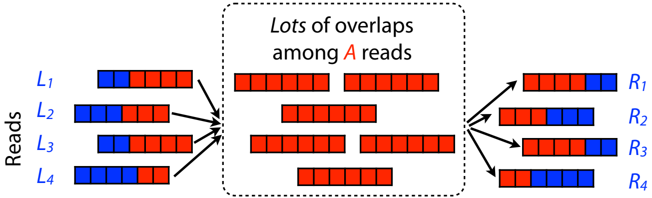
+ unable to handle the large number of NGS sequencing reads

Sequencing errors

```
    TATCTCGACTCTAGGCC
    ||||||| |||||||
TCTATATCTCGGCTCTAGG
```

- Approximate overlaps : Construction of alignments between reads
- Assembly : consensus sequence

Repeats



The region A is longer than reads

Diploidy and polyploidy

```
Read from Mother:          TATCTCGACTCTAGGCC
                            ||||||| |||||||
Read from Father:     TCTATATCTCGGCTCTAGG

Sequence from Mother: TCTATATCTCGACTCTAGGCC
Sequence from Father: TCTATATCTCGGCTCTAGGCC
```

Coverage

```
                  CTAGGCCCTCAATTTTT
                CTCTAGGCCCTCAATTTTT
              GGCTCTAGGCCCTCATTTTTT
           CTCGGCTCTAGCCCTCATTTT
        TATCTCGACTCTAGGCCCTCA
        TATCTCGACTCTAGGCC
    TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
```

Coverage = 5

```
                         CTAGGCCCTCAATTTTT
                       CTCTAGGCCCTCAATTTTT
                     GGCTCTAGGCCCTCATTTTTT
                  CTCGGCTCTAGCCCCTCATTTT
               TATCTCGACTCTAGGCCCTCA
               TATCTCGACTCTAGGCC         177 bases
           TCTATATCTCGGCTCTAGG
       GGCGTCTATATCTCG
       GGCGTCGATATCT
       GGCGTCTATATCT                     35 bases
       GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
```

Average coverage = 177 / 35 ≈ 5-fold

Overlaps - Historical perspectives

- Sanger sequencing
  - Celera (Myers, 2000)
    originally developed for the assembly of the human genome
  - SGA (Simpson, Durbin, 2012)
- not suitable for NGS short reads (Illumina)

  computationally expensive : construction of the graph, size of the graph, path discovery
- comeback with long reads (Nanopore, Pacbio)

De Bruijn graphs

- introduced in bioinformatics to deal with NGS data
- used by allmost modern short-reads assembly tools

  seminal : `Velvet` (2008), `Abyss` (2009), `SOAPdenovo2` (2012)

  state-of-the art : `SPAdes` (2012), `MaSuRCA` (2013), `Megahit` (2015)...

Genome assembly reborn : recent computational challenges. M. Pop, Briefings in Bioinformatics 2009 https://doi.org/10.1093/bib/bbp026

How to apply de Bruijn graphs to genome assembly. P.E.C. Compeau, P.A. Pevzner, G. Tesler, Nature Biotechnology 2011 doi:10.1038/nbt.2023

Rationale

- The genome can be reconstructed from the k-mers it contains
- Reads are decomposed into k-mers

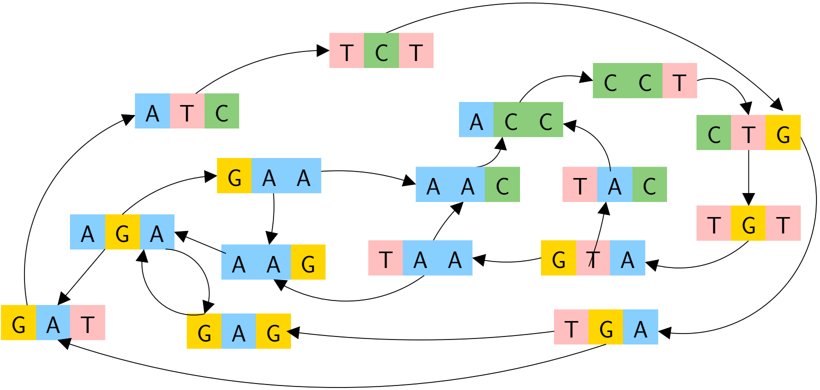
How many distinct 3-mers are they in

R_1	C	T	G	A	G	A	A	C	C	T	G	T
R_2	C	C	T	G	T	A	A	G	A	T		
R_3	C	T	G	T	A	C	C	T				
R_4	G	A	T	C	T	G	A					

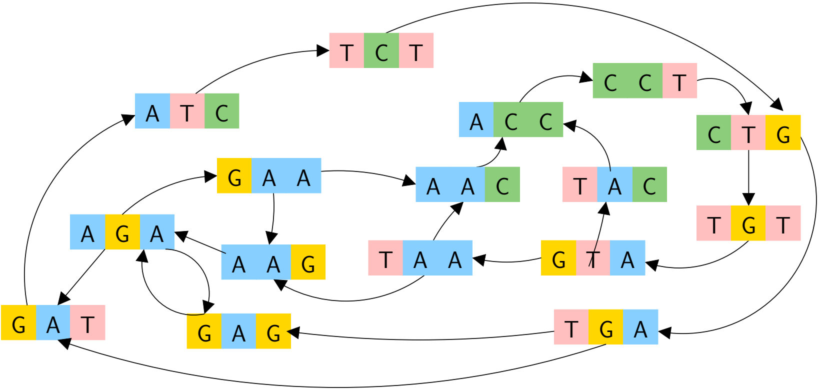
A A C	A T C	G A G	T A C
A A G	C C T	G A T	T C T
A C C	C T G	G T A	T G A
A G A	G A A	T A A	T G T

De Bruijn Graph

- Nodes : k-mers present in the reads
- Arcs : overlaps of length $k-1$ between k-mers
  Do not depend on the set of reads
- Easy to construct, low memory footprint
  Great advantage over overlap graphs

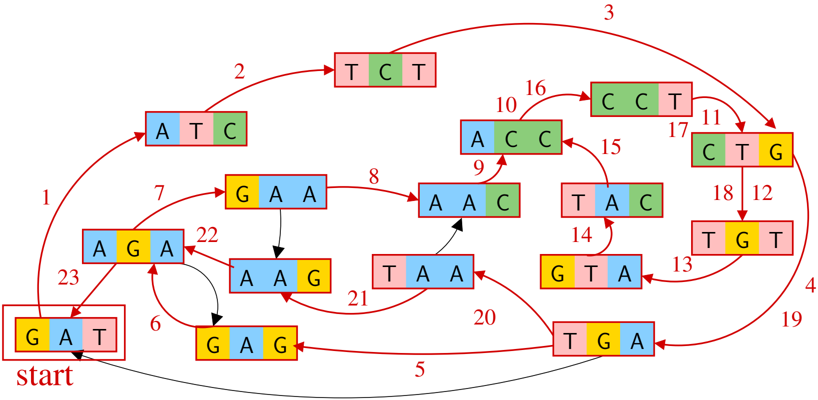


TCT
ATC
CCT
ACC
CTG
GAA
AAC
TAC
AGA
AAG
TAA
GTA
TGT
GAT
GAG
TGA



Assembly = path in the graph

Several paths = several assemblies



R_1 C T G A G A A C C T G T   C C T G T A A G A T R_2

R_4 G A T C T G A R_3 C T G T A C C T

G A T C T G A G A A C C T G T A C C T G T A A G A T

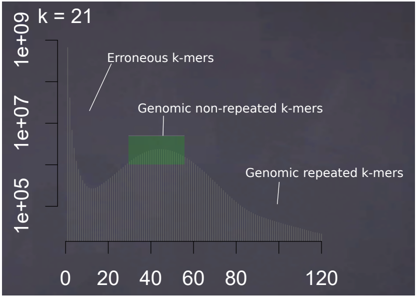
De Bruijn Graphs in practice - choice of k

Length of k-mers

- small k :
  - pro : more non-erroneous k-mers
  - cons : less signal, more random overlaps, repeat collapsing
- large k :
  - pro : higher signal, less random overlaps, less repeat collapsing
  - cons : more erroneous k-mers
- generally $k \geq 20$ (may be longer for large genomes)
- higher sequencing coverage means larger k values can be used
- multi-k assembly ($k = 21 \rightarrow k = 55 \rightarrow k = 72$)
  `IDBA`, `SPAdes`, `Megahit`

In this lecture : `SPAdes`

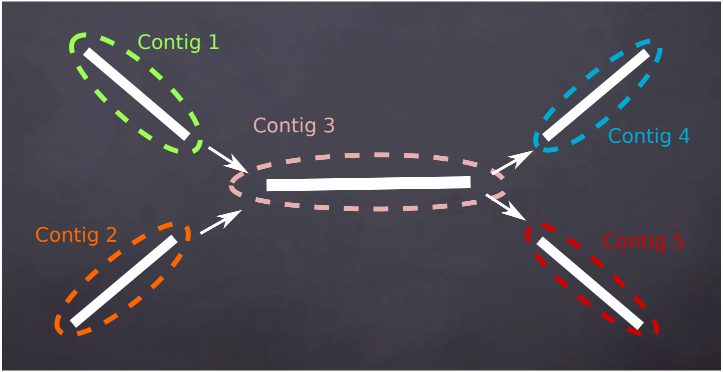
De Bruijn Graphs in practice - cleaning k-mers



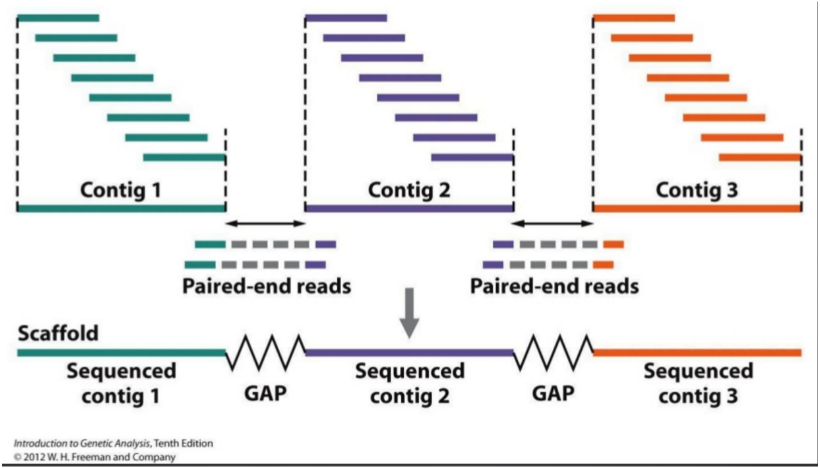
Courtesy of Rayan Chikhi (Institut Pasteur)
horizontal axis : number of occurrences
vertical axis : number of k-mers

- k-mers with low frequency are likely to contain sequencing errors
- remove k-mers with too few occurrences before the construction of the graph
- k-mer counting : `Jellyfish`, `Kmergenie`, `DSK`

De Bruijn Graphs in practice - contigs and scaffolds



Contigs $=$*simple* paths in the graph



Scaffold = link between contigs using paired-end reads
Error-prone

Short read assembly is still difficult

even with De Bruijn graphs

- risk of contamination
- existence of sequencing errors solved
- existence of repeats
- diploid and polyploid genomes
- low coverage or unevent coverage

+ unable to handle the large number of NGS sequencing reads solved

Short read assembly is still a difficult

- library design
  - longest read lengths
  - coverage $\geq$ 50x, x ploidy number
  - for 1 bacterial genome, no point going above 200x
  - BROAD recipe : several mate pairs libraries of increasing size
- assembler
  - SPAdes for small genomes
  - unclear for large genomes
  - try at least two assemblers, try different parameters
  - high computational requirements overall
- an assembly is not the absolute truth, it is a mostly complete, generally fragmented and mostly accurate hypothesis

How to compare/analyse assemblies ?

- no trivial ranking between assemblies
- no simple criteria
- assembly with high coverage and short contigs / assembly with low coverage and long contigs

Quast

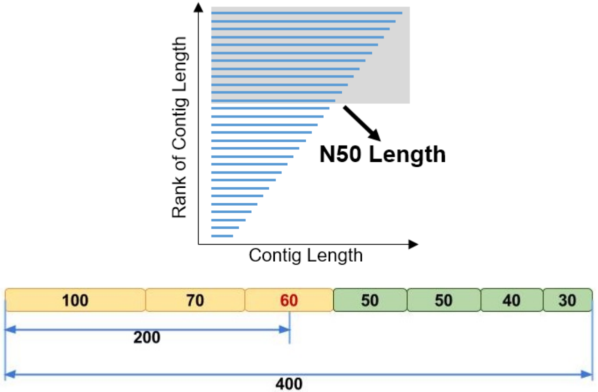
Quality Assessment Tool for Genome Assemblies

- provides a large number of statistics and metrics : contigs, missamblies, functional elements
- works both with and without a reference genome
- accepts multiple assemblies, thus is suitable for comparison

QUAST : quality assessment tool for genome assemblies. Bioinformatics 2013 `https://doi.org/10.1093/bioinformatics/btt086`

Contigs

- number of contigs
- length of the largest contig
- total number of bases in the assembly (sum of contig lengths)



- $N50$: contig length N for which 50% of all bases in the sequences are in a contig of length $L \geq N$
- $NG50$: contig length such that using equal or longer length contigs produces 50% of the expected length of the reference genome

Miassemblies (requires a reference genome)

- missassembly breakpoints : position in the contig where the left flanking sequence aligns over 1 kb away from the right flanking sequence on the reference, or they overlap >1 kb, or align on opposite strands or different chromosomes :
- metrics : total number of missambly breakpoints, number of contigs that contain misassembly breakpoints, number of bases contained in all contigs that have one or more misassemblies (`Mummer`)
- number of unaligned contigs : contigs that have no alignment to the reference sequence
- number of ambiguously mapped contigs : contigs that have multiple alignments to the reference genome

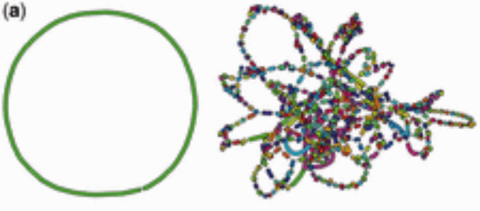
Functional elements

- genome fraction (%) : number of aligned bases in the reference, divided by the genome size.
- duplication ratio : number of aligned bases in the assembly divided by the number of aligned bases in the reference
- number of mismatches and number of indels per 100 kb
- number of genes based on a user-provided annotated list of gene positions in the reference genome
- number of predicted genes in the assembly (`GeneMark.hmm` for prokaryotes and `GlimmerHMM` for eukaryotes)

Bandage

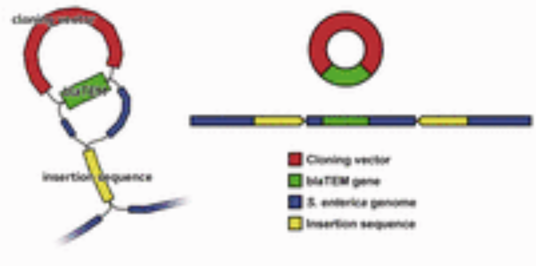
Bioinformatics Application for Navigating De novo Assembly Graphs Easily

- interactive visualization of the assembly graph (such as de Bruijn graph)
- enables BLAST searches



Left : ideal bacterial assembly with one single contig
Right : poor assembly with many short contigs

Bandage : interactive visualization of de novo genome assemblies, Bioinformatics 2015 `https://doi.org/10.1093/bioinformatics/btv383`



Left : Repeated sequences : single nodes with multiple inputs and outputs

Right, underlying gene structure deduced from Bandage visualization

16S rRNA regions of a bacterial genome assembly graph
(found in the graph with Blast)