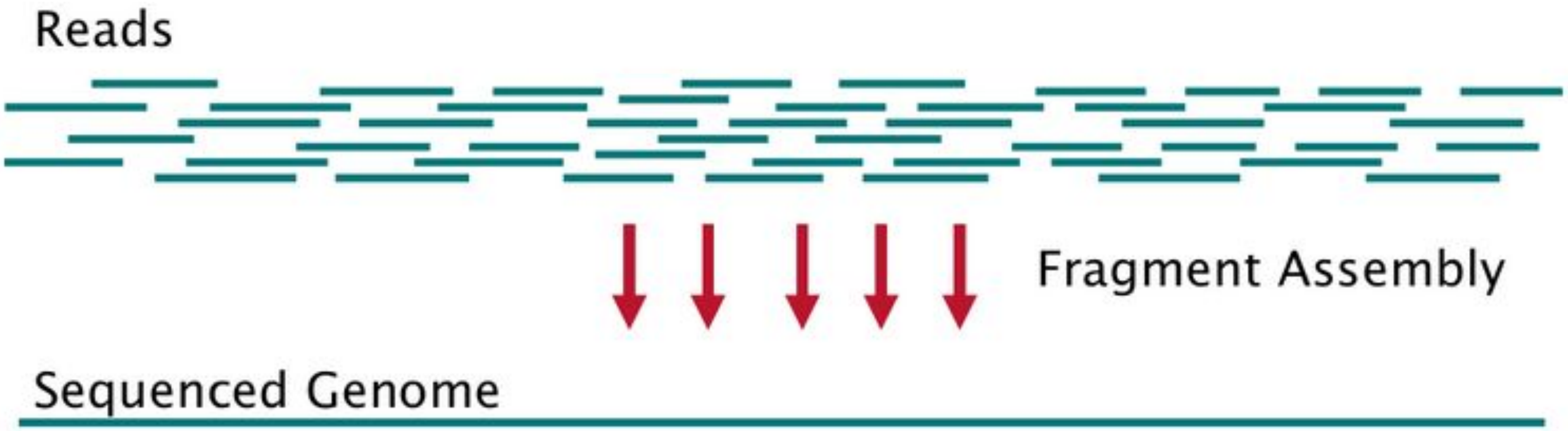
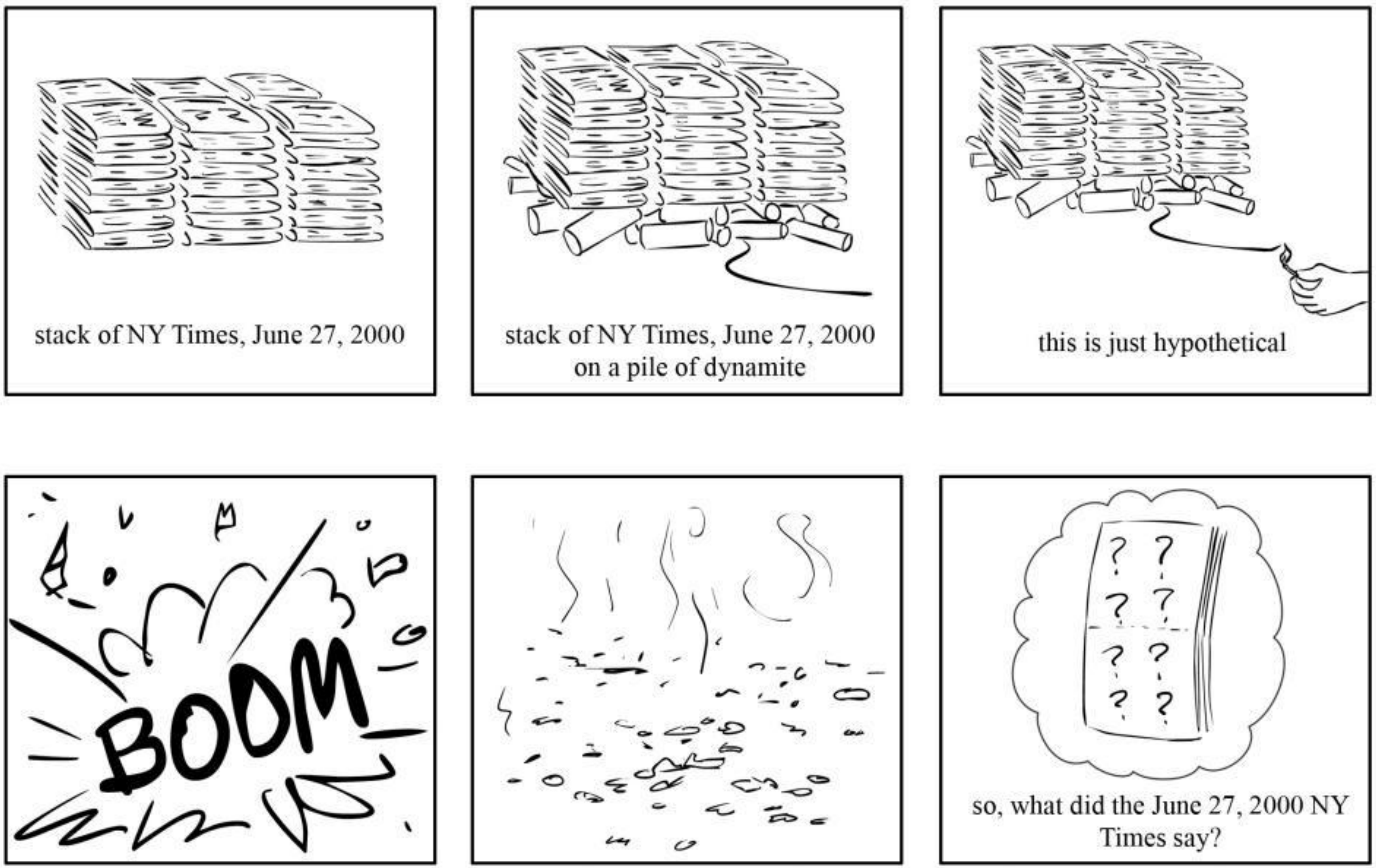
Genome assembly

Meleshko Dmitry
meleshko.dmitrii@gmail.com

De novo genome assembly



De novo whole genome assembly



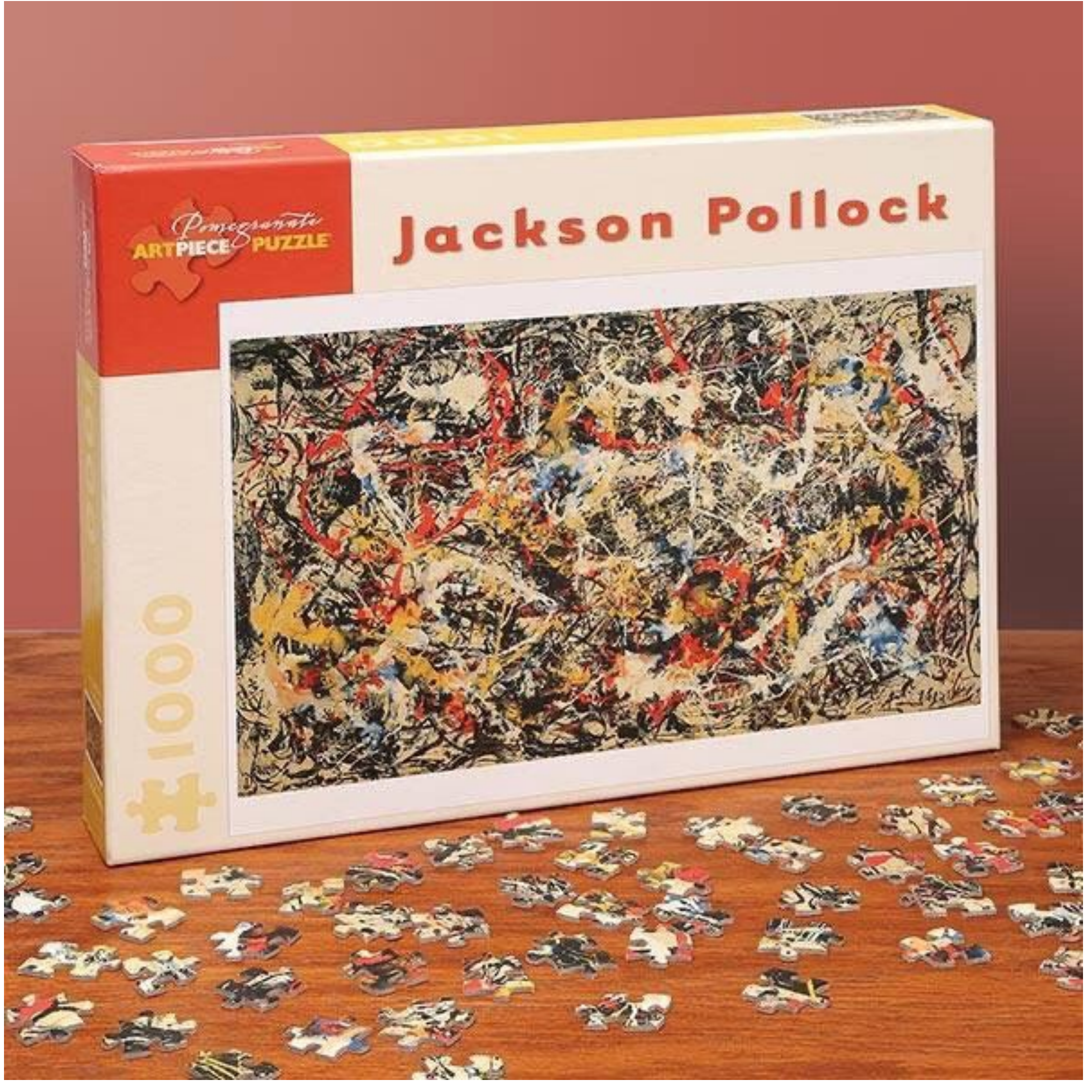
stack of NY Times, June 27, 2000

stack of NY Times, June 27, 2000 on a pile of dynamite

this is just hypothetical

so, what did the June 27, 2000 NY Times say?

Whole genome assembly



Shortest common supersequence

Given a set of strings $\{s_1, \ldots, s_n\}$, find a shortest string S containing each s_i as a substring

Shortest common supersequence

Given a set of strings $\{s_1, \ldots, s_n\}$, find a shortest string S containing each s_i as a substring

Is NP-complete

Has nothing to do with real genome assembly problem

Why to assemble?

- NGS
  - ○ Billions of short reads
  - ○ Sequencing errors
  - ○ Contaminants

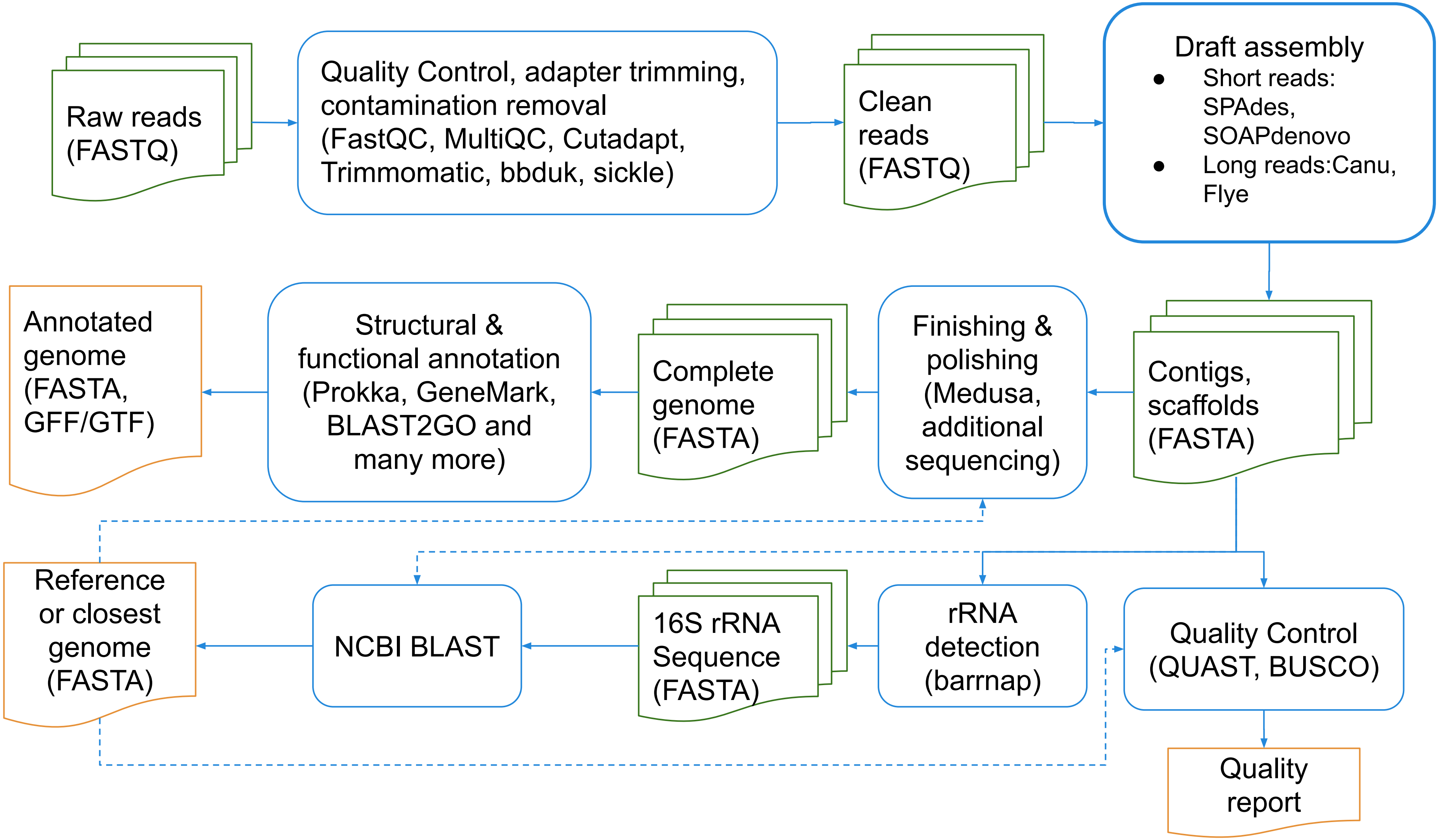
Hard to perform analysis

- Assembly
  - ✓ Corrects sequencing errors
  - ✓ Much longer sequences
  - ✓ Each genomic region is presented only once
  - × May introduce errors

Assembly types

- *De novo* genome assembly
  - Long reads
  - Short reads
  - Hybrid
- Reference-assisted genome assembly
  - Closely related species
- Transcriptome assembly
  - *De novo*
  - Reference based

De novo genome assembly



Raw reads (FASTQ)

Quality Control, adapter trimming, contamination removal (FastQC, MultiQC, Cutadapt, Trimmomatic, bbduk, sickle)

Clean reads (FASTQ)

Draft assembly

- Short reads: SPAdes, SOAPdenovo
- Long reads:Canu, Flye

Contigs, scaffolds (FASTA)

Finishing & polishing (Medusa, additional sequencing)

Complete genome (FASTA)

Structural & functional annotation (Prokka, GeneMark, BLAST2GO and many more)

Annotated genome (FASTA, GFF/GTF)

Quality Control (QUAST, BUSCO)

Quality report

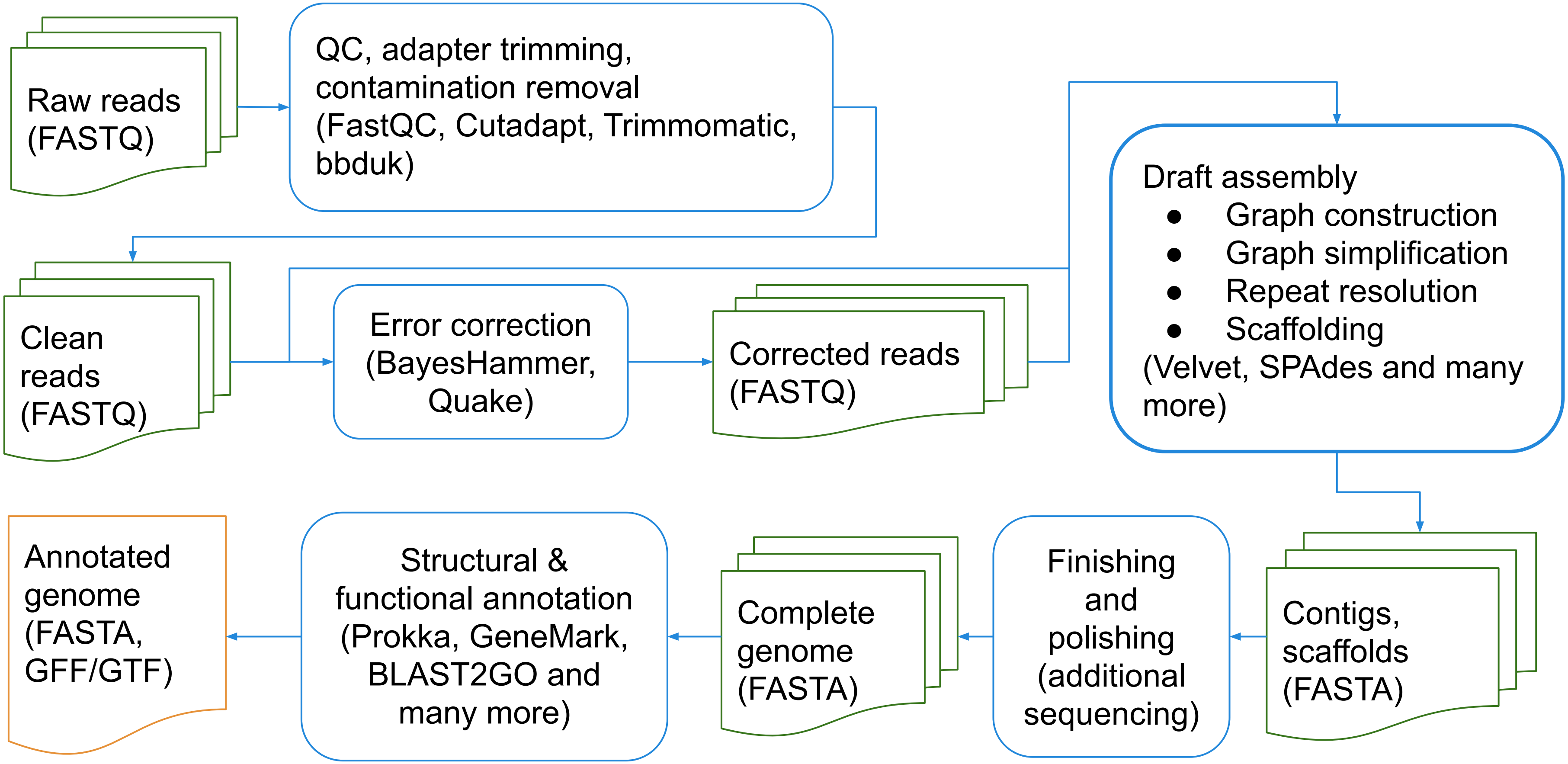
rRNA detection (barrnap)

16S rRNA Sequence (FASTA)

NCBI BLAST

Reference or closest genome (FASTA)

De novo genome assembly



Raw reads (FASTQ)

QC, adapter trimming, contamination removal (FastQC, Cutadapt, Trimmomatic, bbduk)

Clean reads (FASTQ)

Error correction (BayesHammer, Quake)

Corrected reads (FASTQ)

Draft assembly

- Graph construction
- Graph simplification
- Repeat resolution
- Scaffolding

(Velvet, SPAdes and many more)

Contigs, scaffolds (FASTA)

Finishing and polishing (additional sequencing)

Complete genome (FASTA)

Structural & functional annotation (Prokka, GeneMark, BLAST2GO and many more)

Annotated genome (FASTA, GFF/GTF)

Assembling Sanger reads

Early days

- Sanger sequencing
  - Long reads
  - Low coverage

- Overlap-Layout-Consensus (OLC)
  - Find overlaps between all reads
  - Order reads
  - Merge into consensus sequence

Finding overlaps

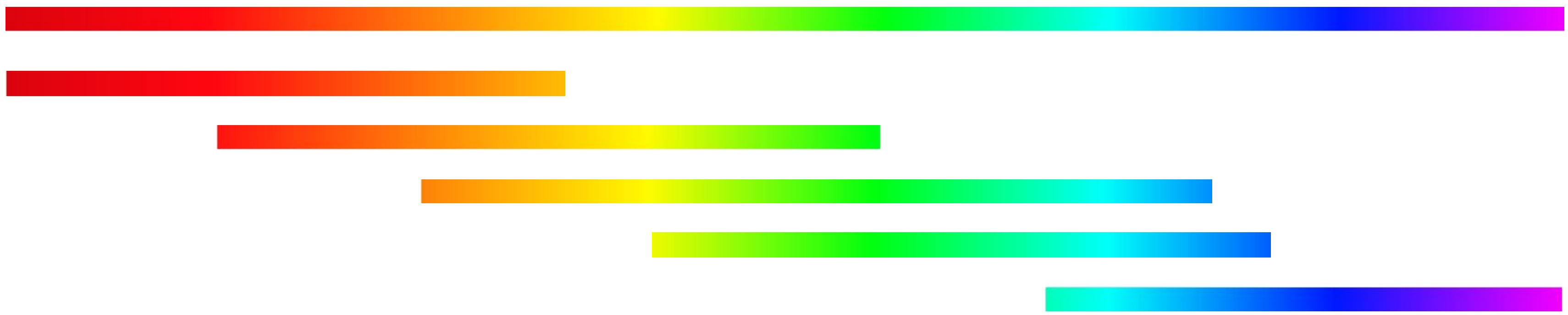
- Align reads all-to-all
  - BLAST and similar algorithms
- Ignore "insufficient" overlaps

Finding overlaps

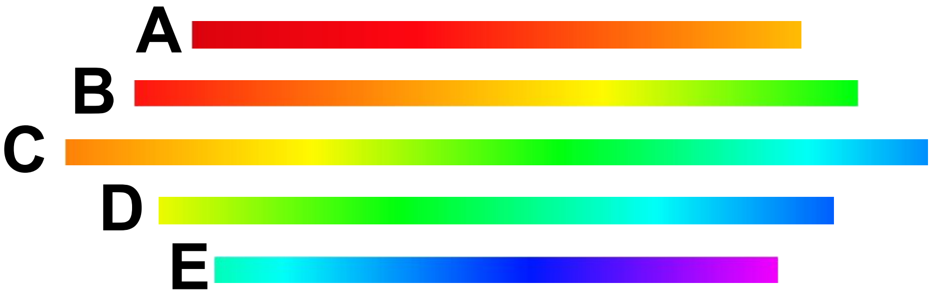
- Align reads all-to-all
  - BLAST and similar algorithms
- Ignore "insufficient" overlaps
  - At least 40bp
  - >94% similarity

Assembly example

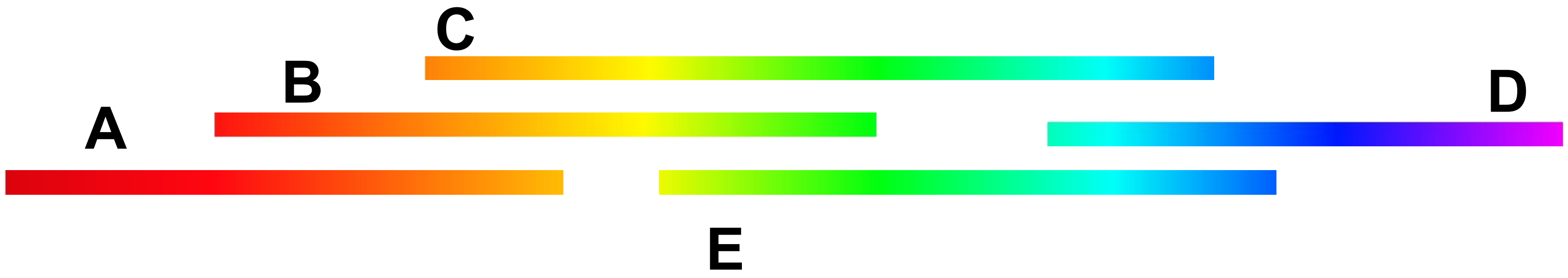
Assembly example

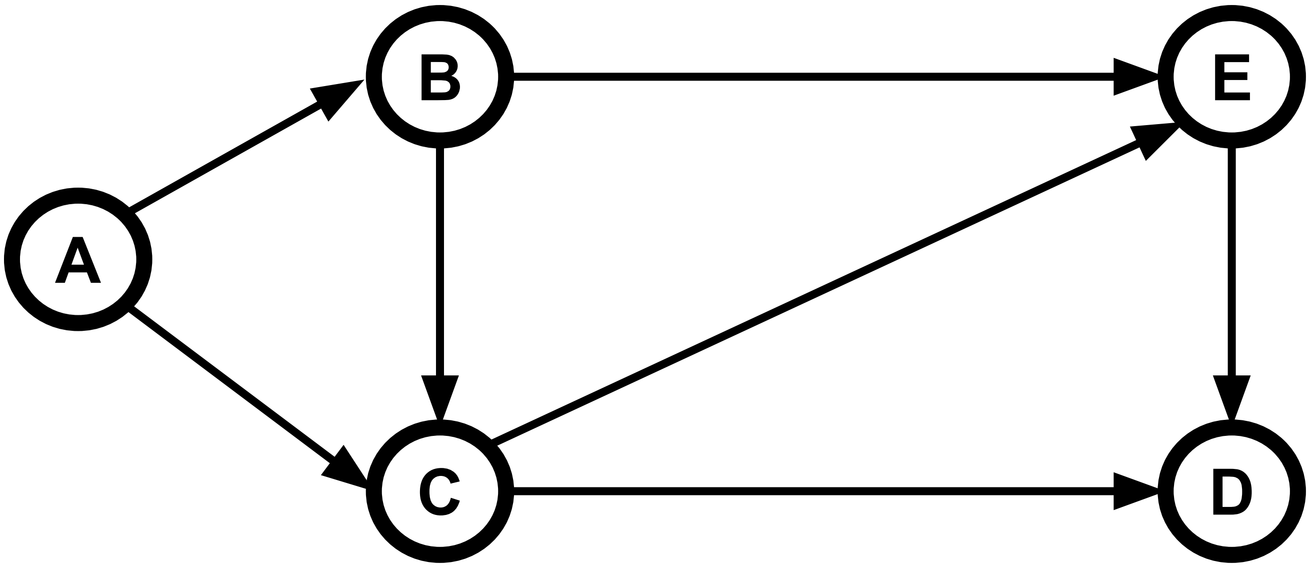


Assembly example

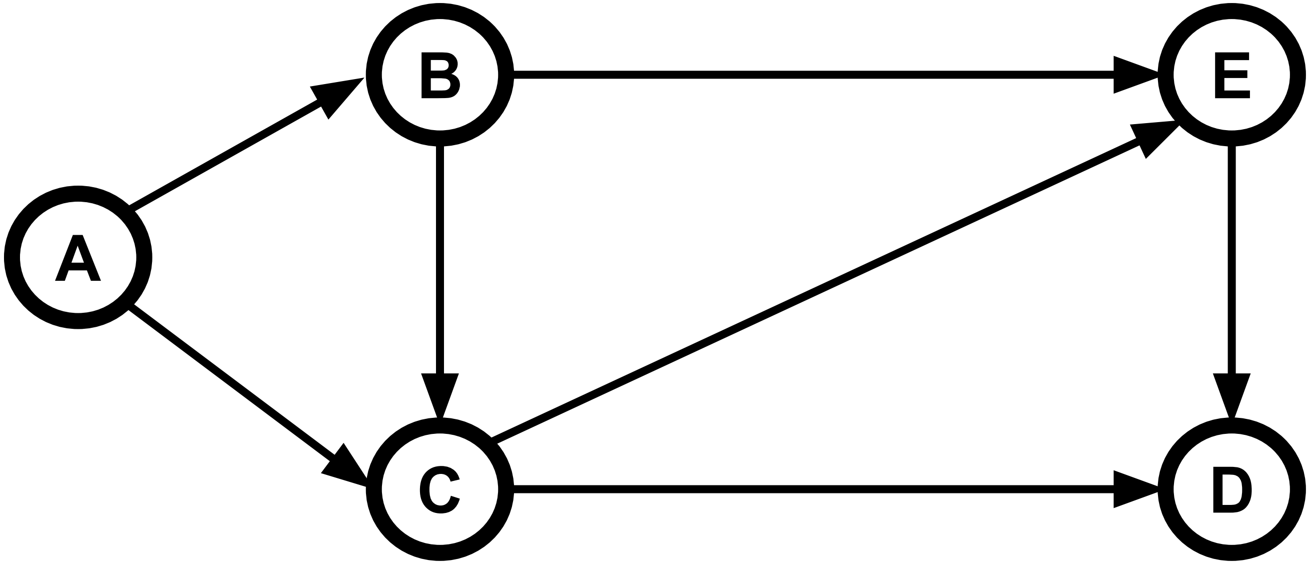


Overlap graph

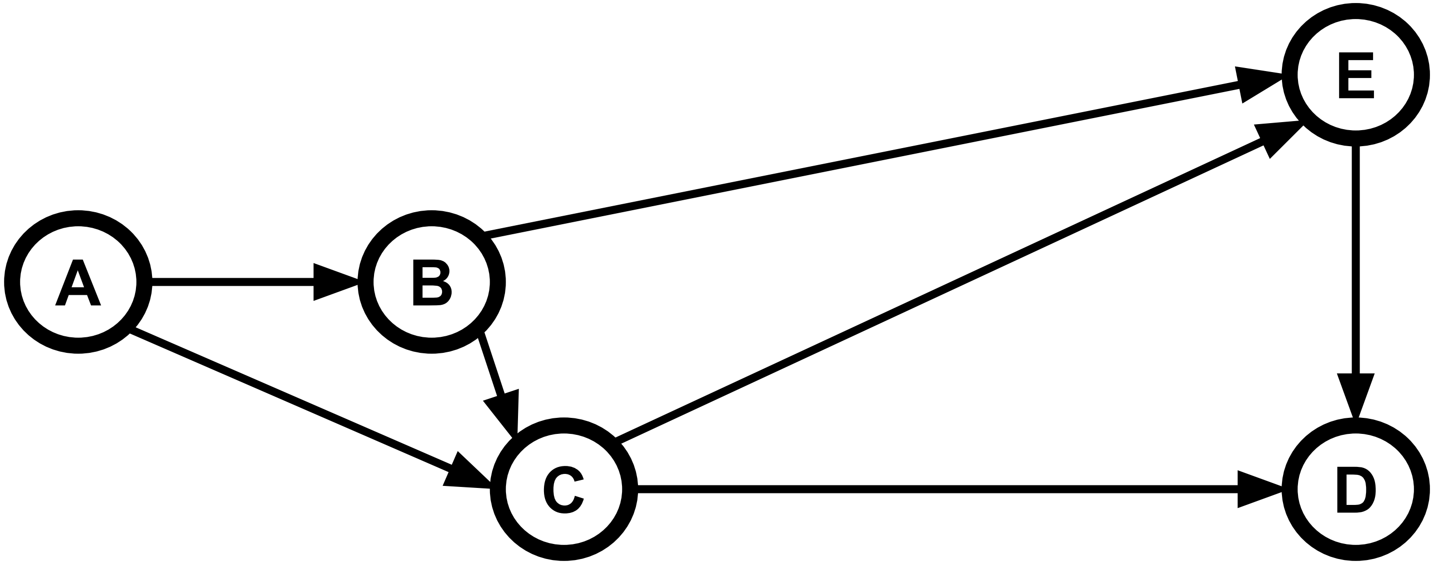




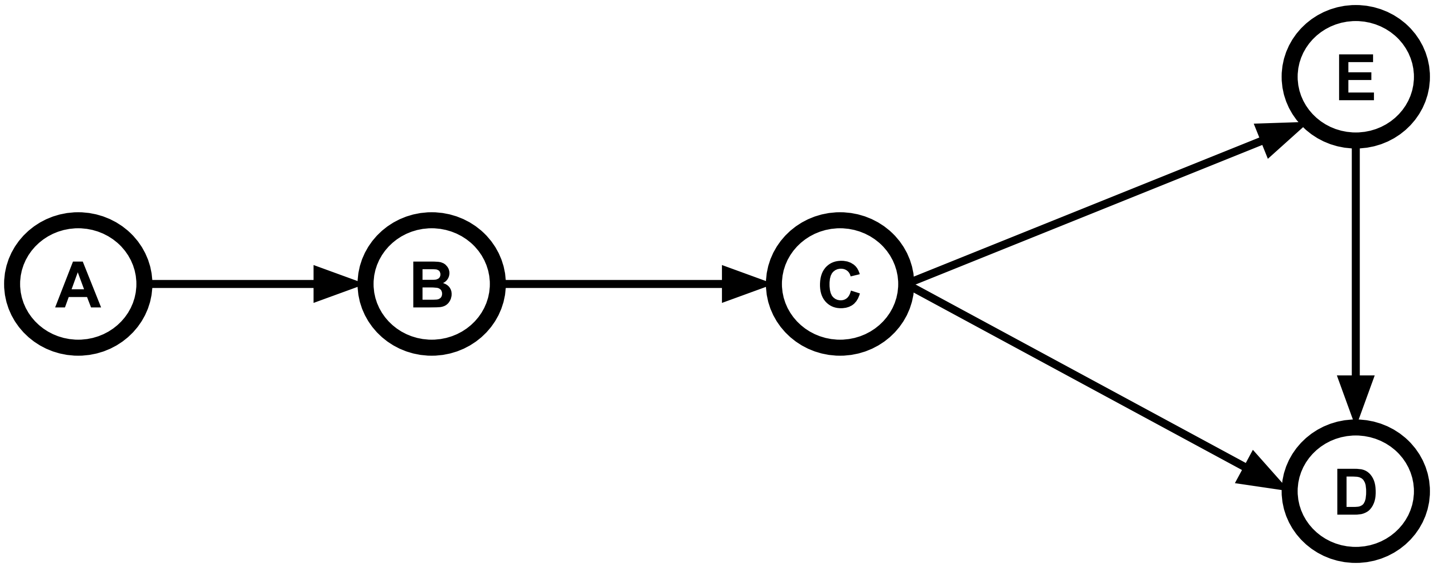
Overlap graph



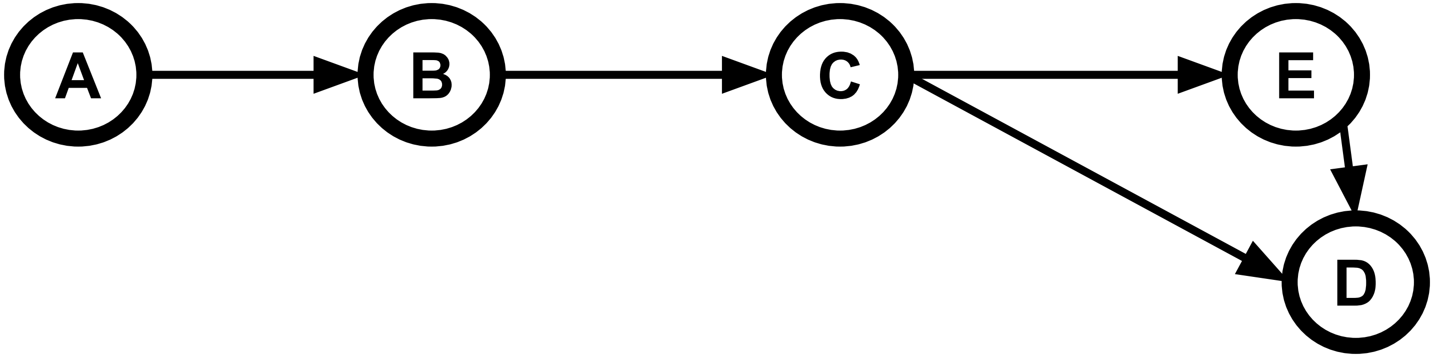
Layout



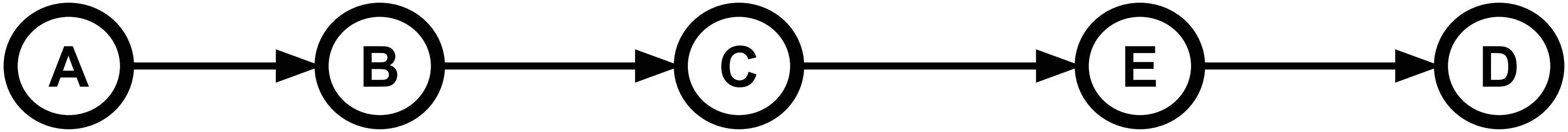
Layout



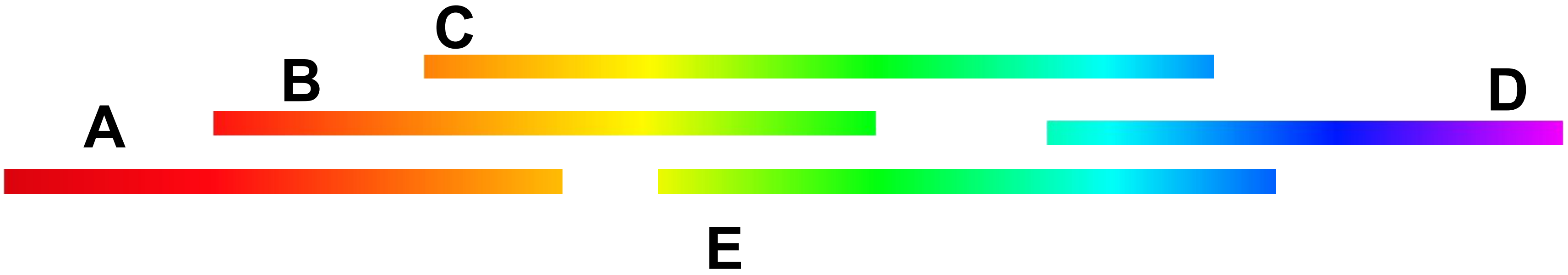
Layout

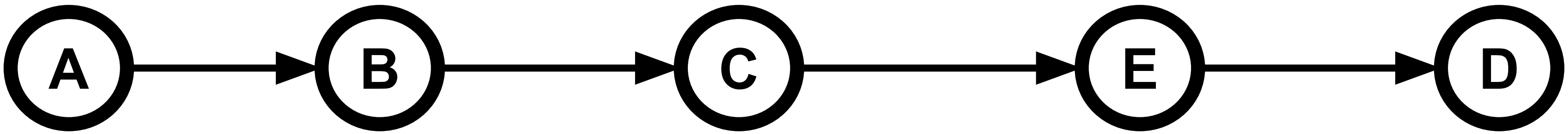


Layout

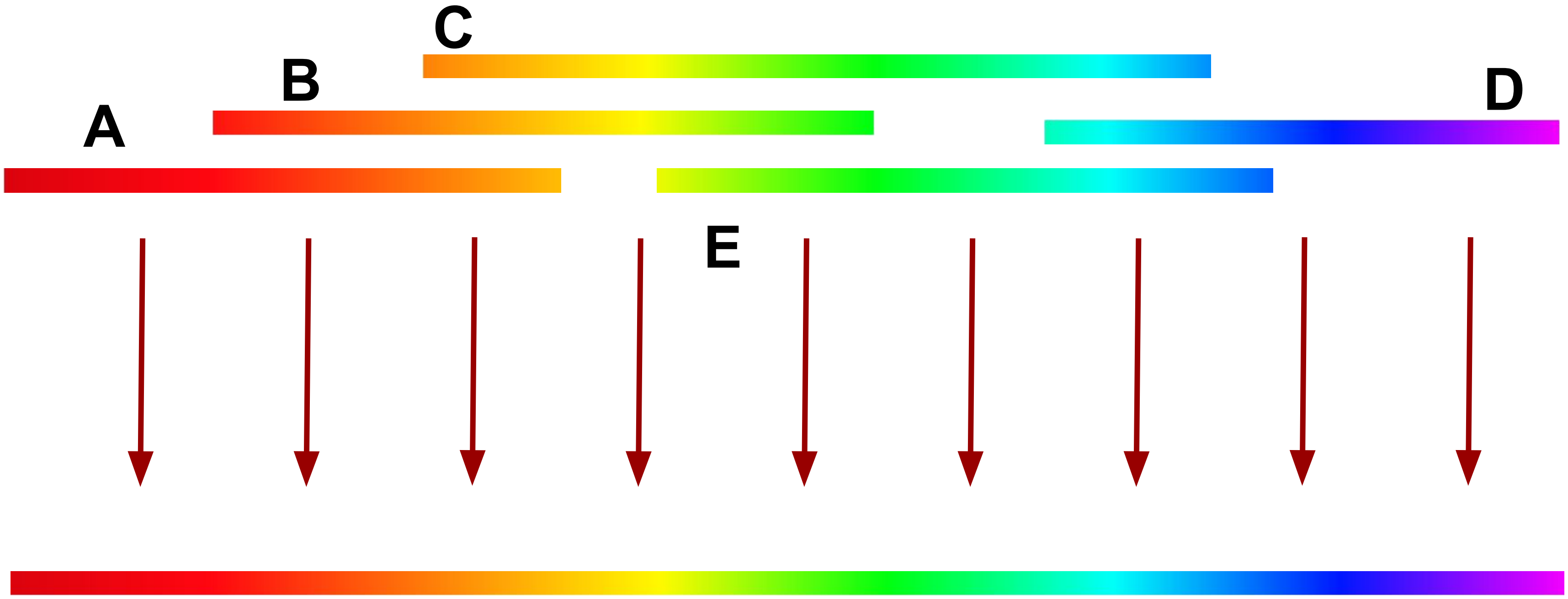


Layout





Consensus



NGS and OLC

- Overlap-Layout-Consensus is not applicable
  - Hard to find overlaps between short reads
  - Impossible to scale to such amount of reads
- De Bruijn graph approach (Pevzner et al., 2001)

- String Graph approach (Meyers, 2005)

NGS era

De Bruijn graph in a nutshell

He that mischief hatches, mischief catches

↓ Sequencing

He that mischief

mischief hatches,

hatches, mischief

, mischief catches

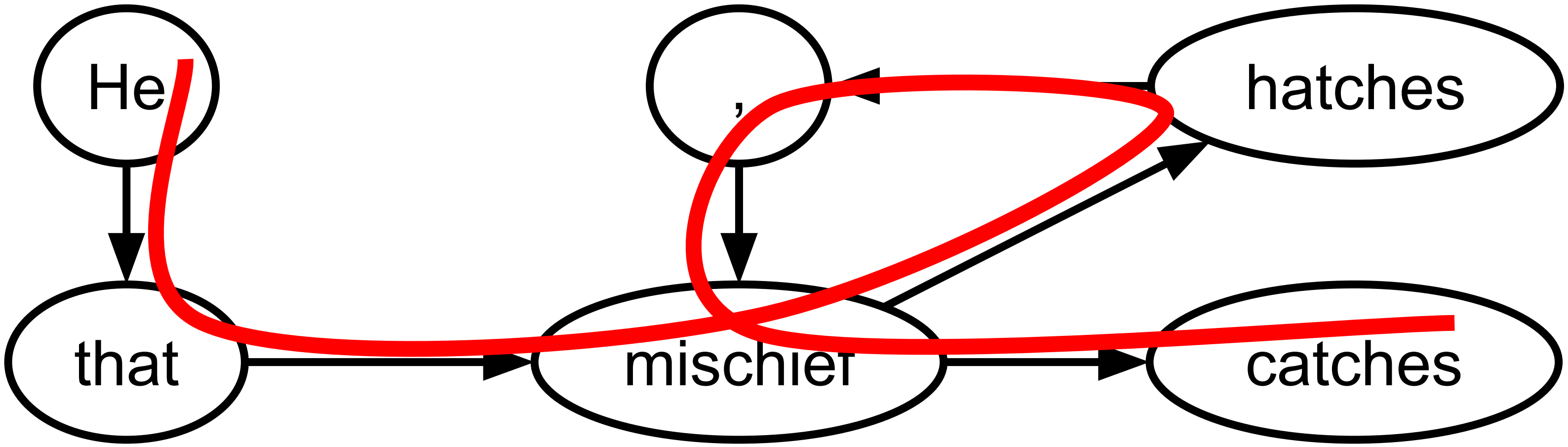
De Bruijn graph in a nutshell

, mischief catches

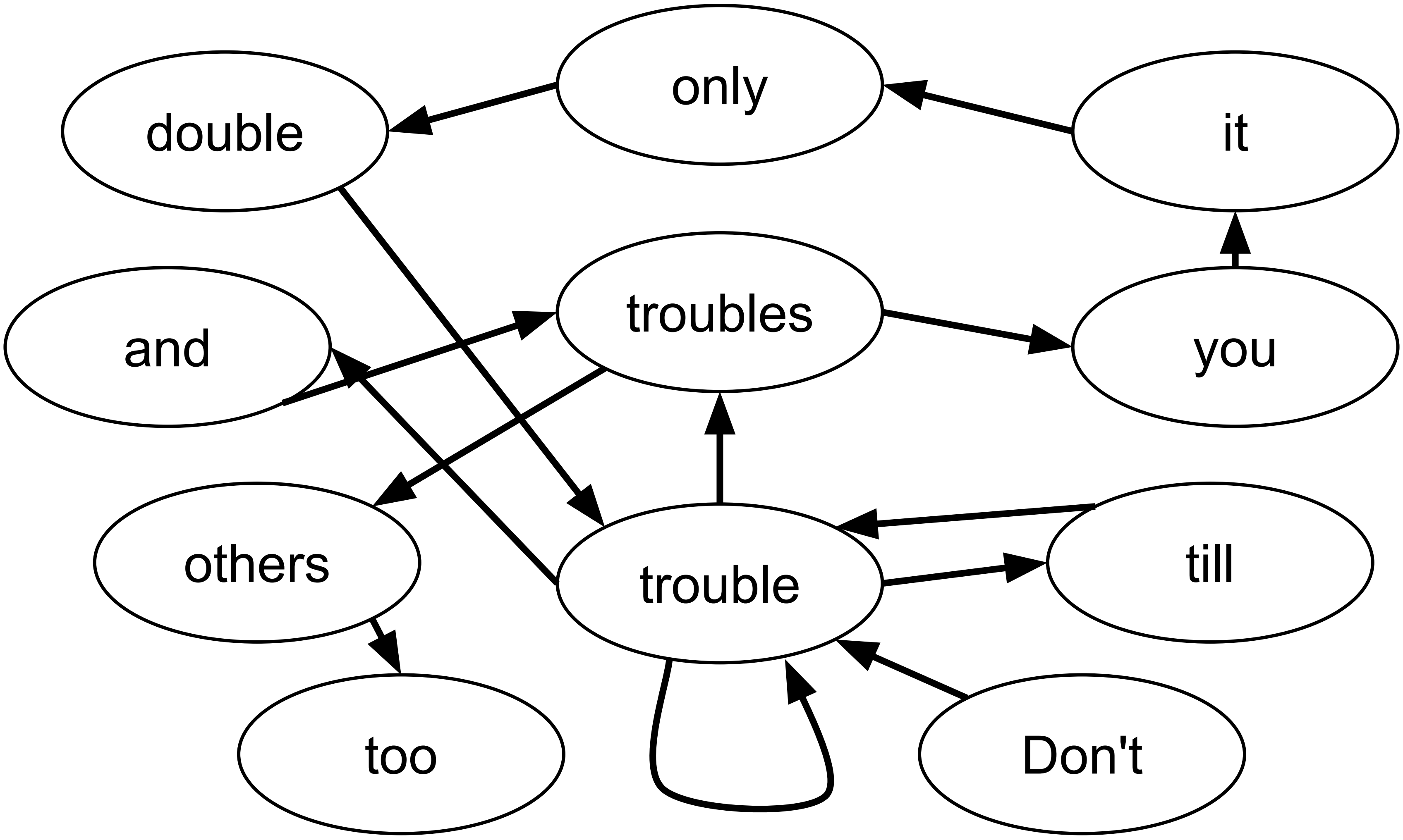
mischief hatches,

He that mischief

hatches, mischief



De Bruijn graph in a nutshell



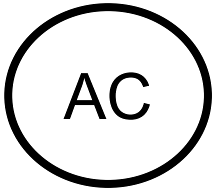
De Bruijn graph

ACGTCCGTAA

De Bruijn graph

ACGTCCGTAA

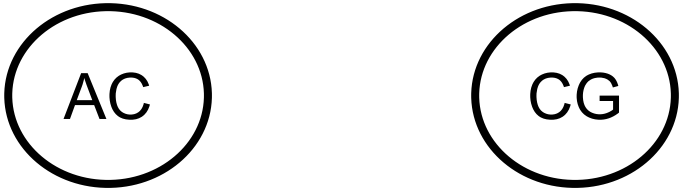
k=2



De Bruijn graph

ACGTCCGTAA

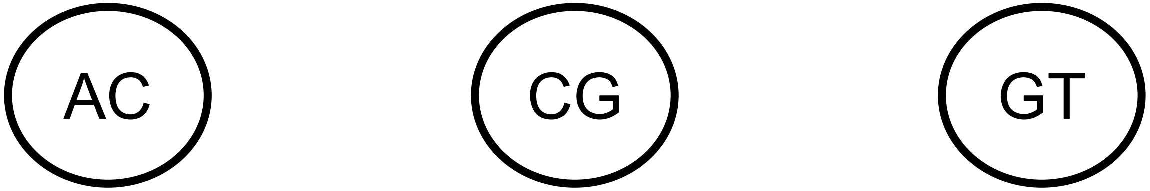
k=2



De Bruijn graph

ACGTCCGTAA

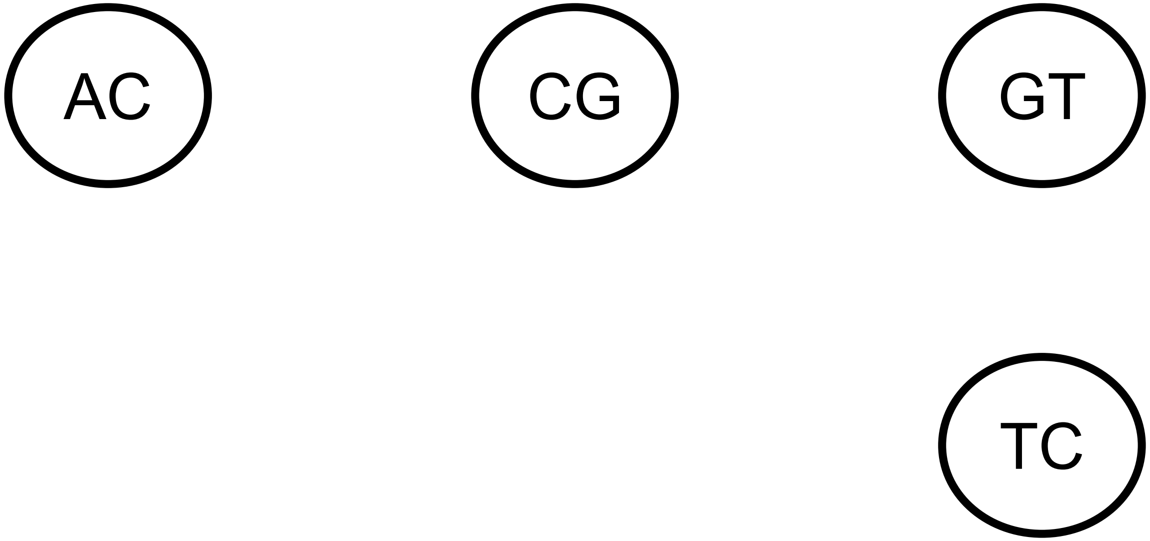
k=2



De Bruijn graph

ACGTCCCGTAA

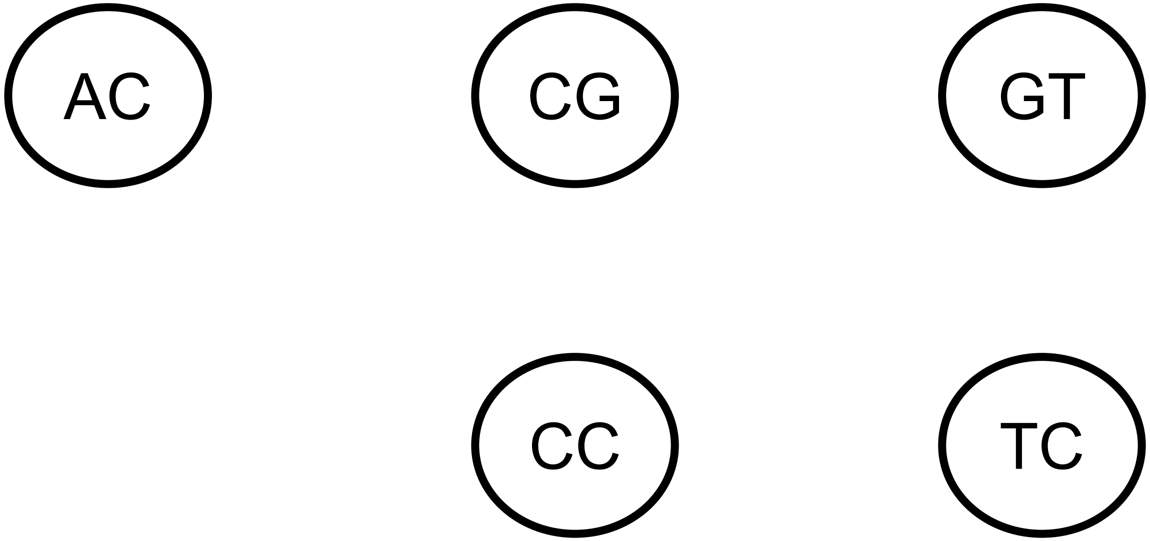
k=2



De Bruijn graph

ACGTCCGTAA

k=2



AC

CG

GT

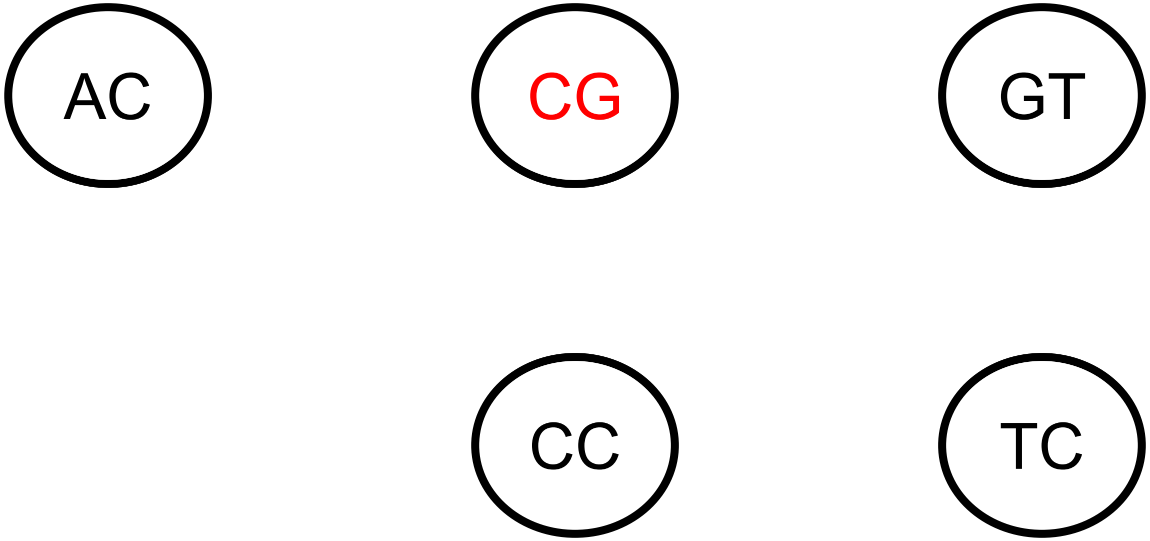
CC

TC

De Bruijn graph

ACGTCCCGTAA

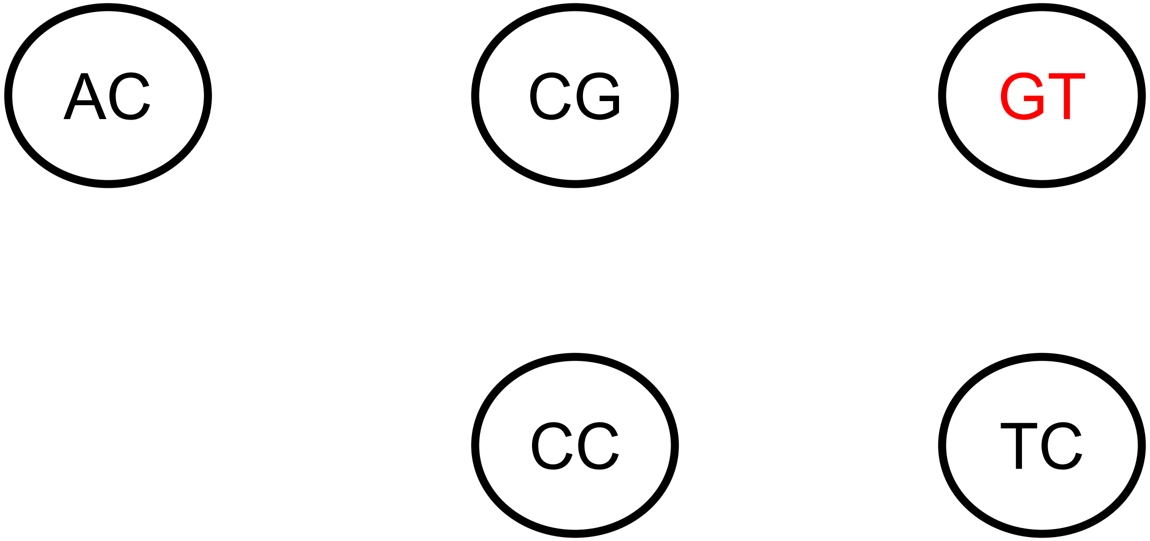
k=2



De Bruijn graph

ACGTCCGTAA

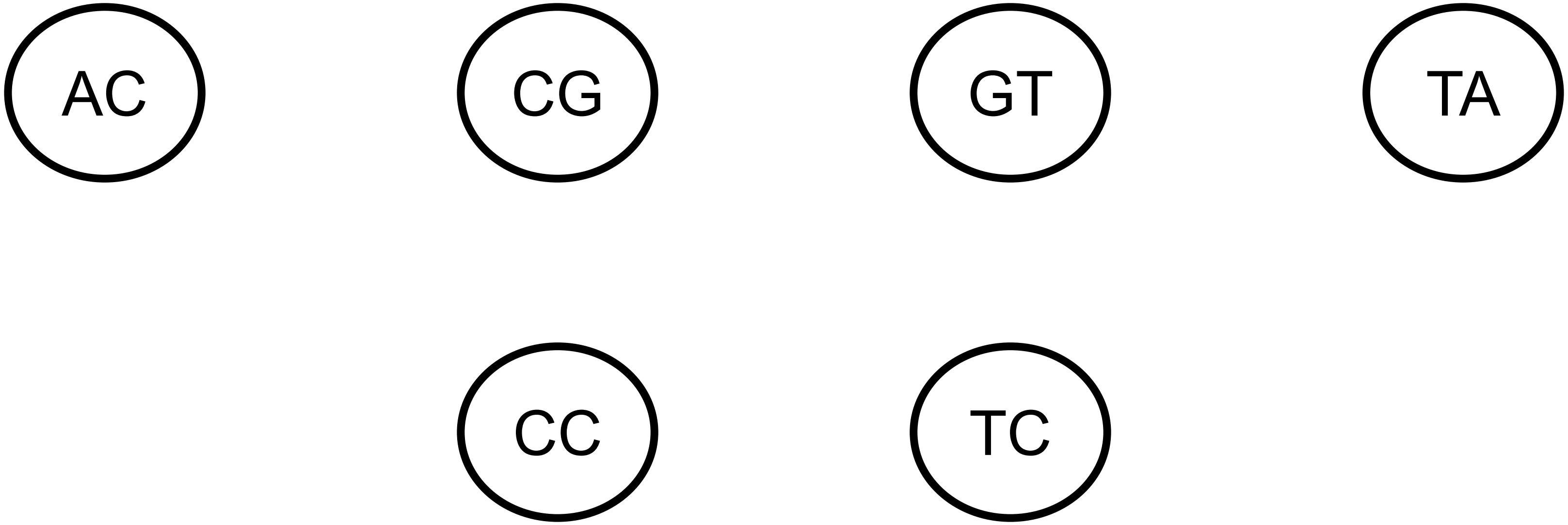
k=2



De Bruijn graph

ACGTCCCGTAA

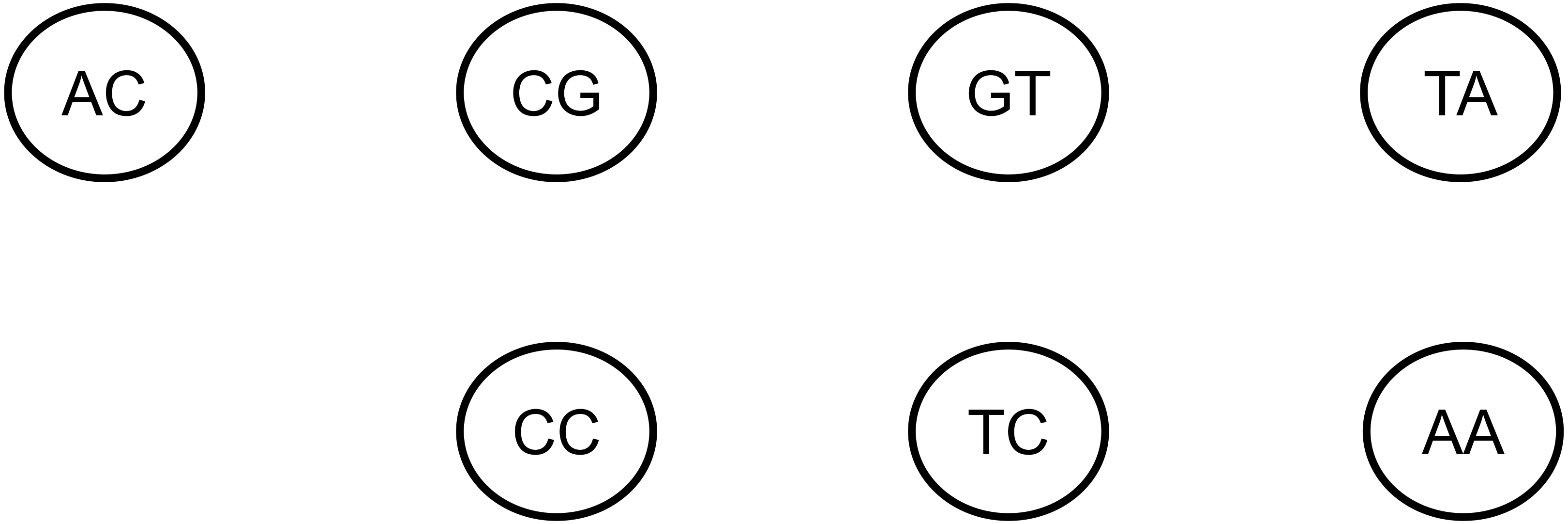
k=2



De Bruijn graph

ACGTCCGTAA

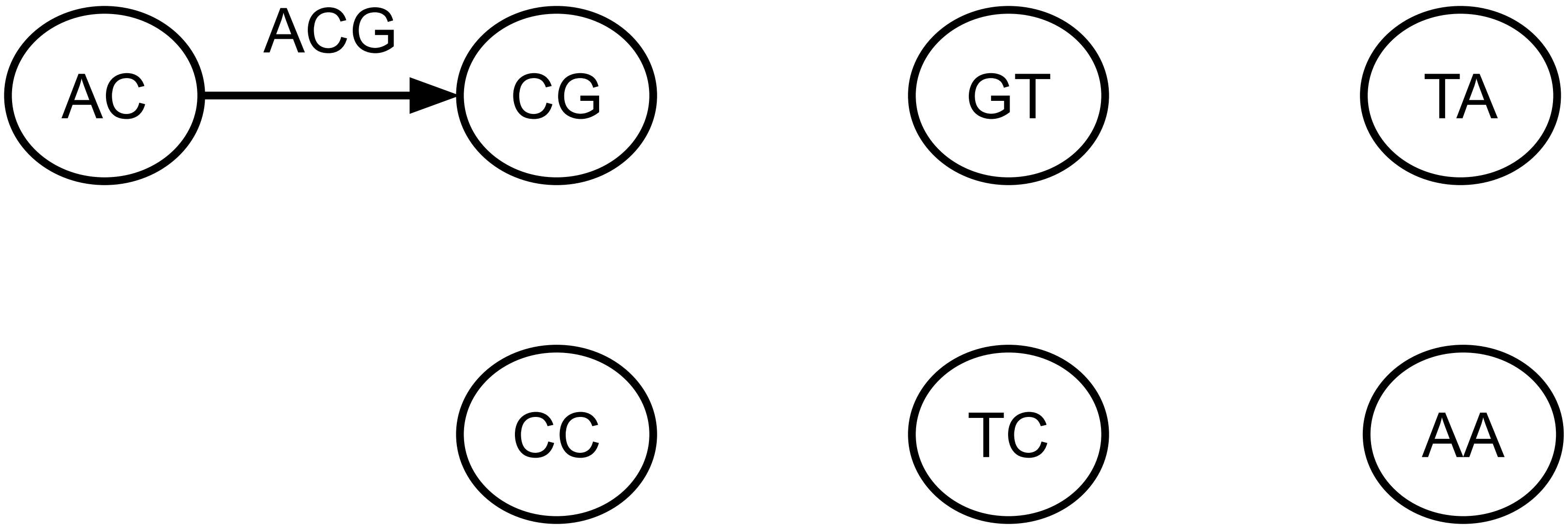
k=2



De Bruijn graph

ACGTCCGTAA

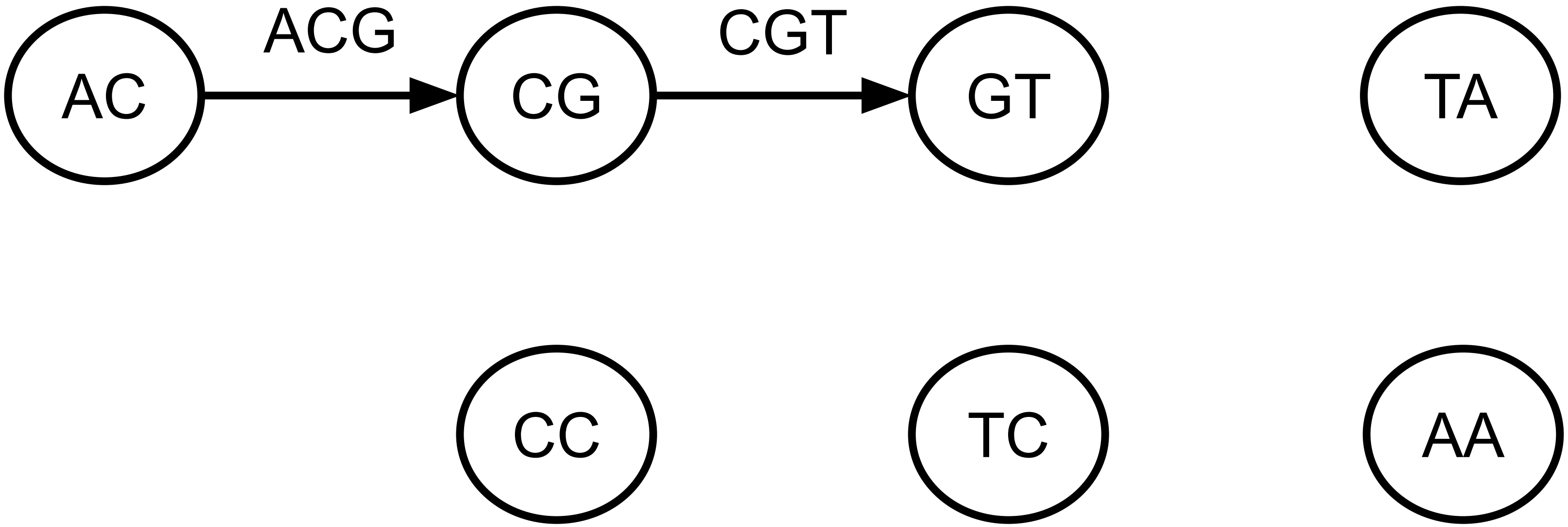
k=2



De Bruijn graph

ACGTCCGTAA

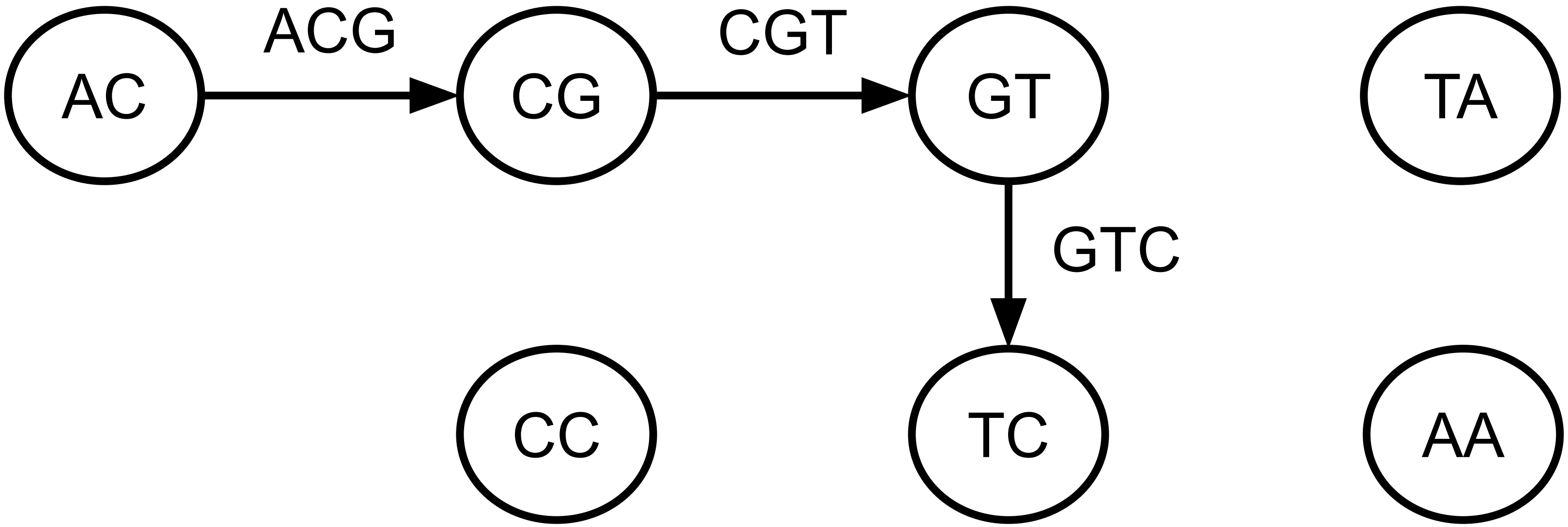
k=2



De Bruijn graph

ACGTCCGTAA

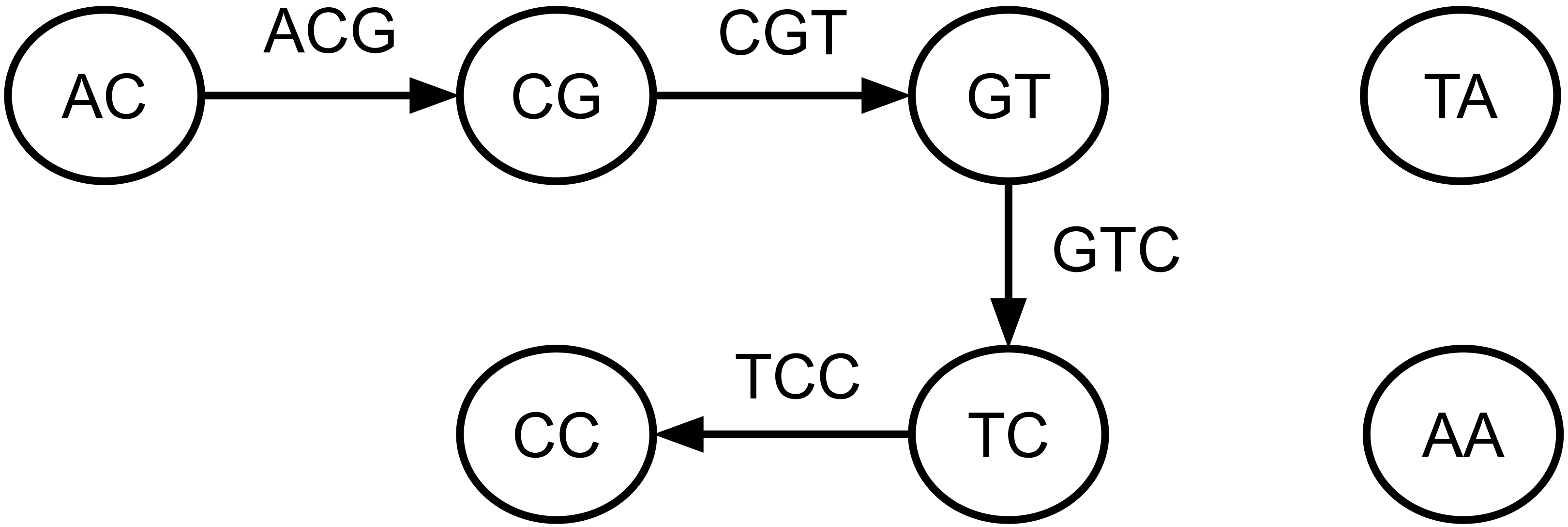
k=2



De Bruijn graph

ACGTCCGTAA

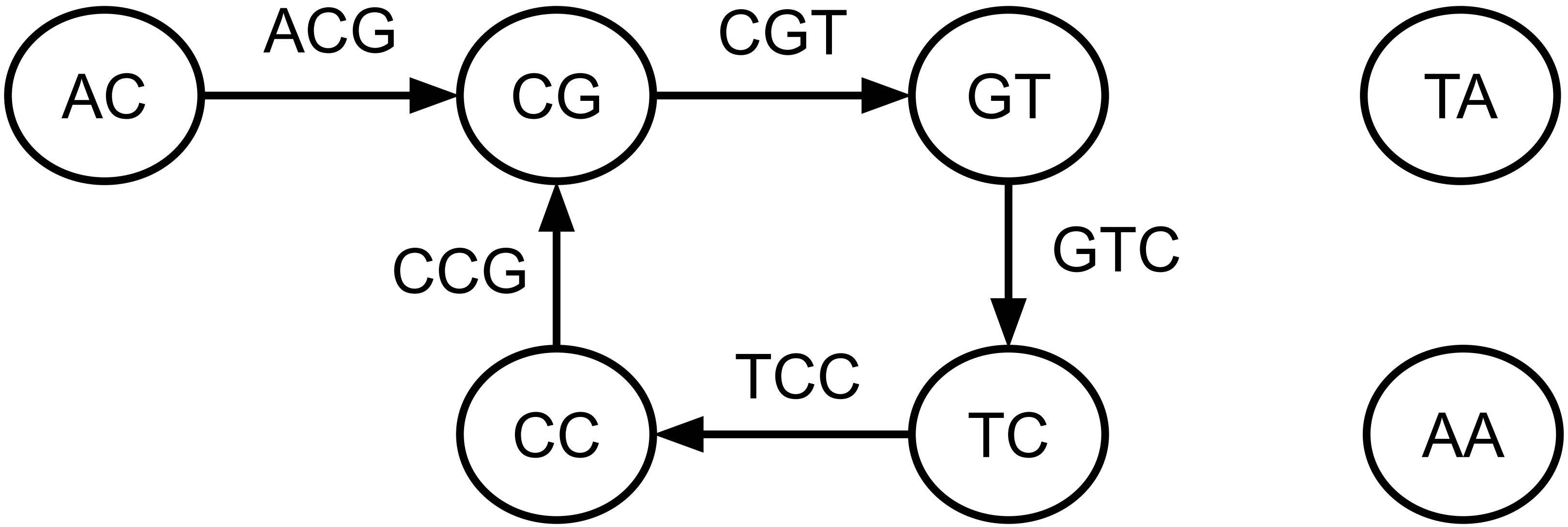
k=2



De Bruijn graph

ACGTCCGTAA

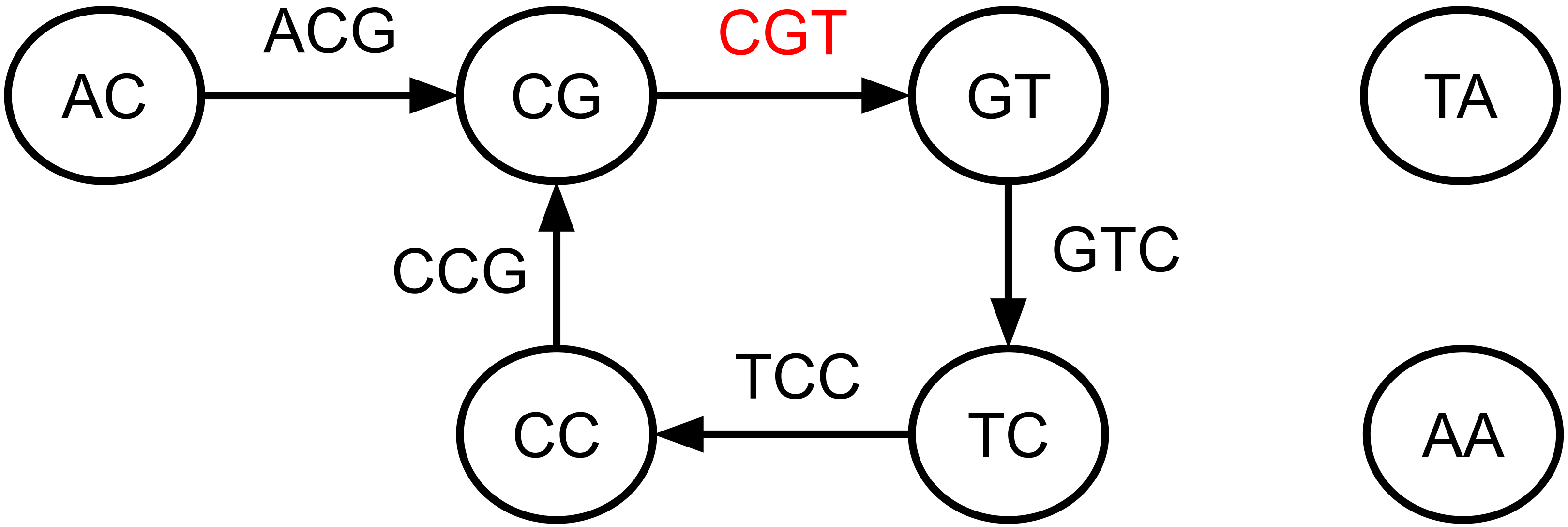
k=2



De Bruijn graph

ACGTCCCGTAA

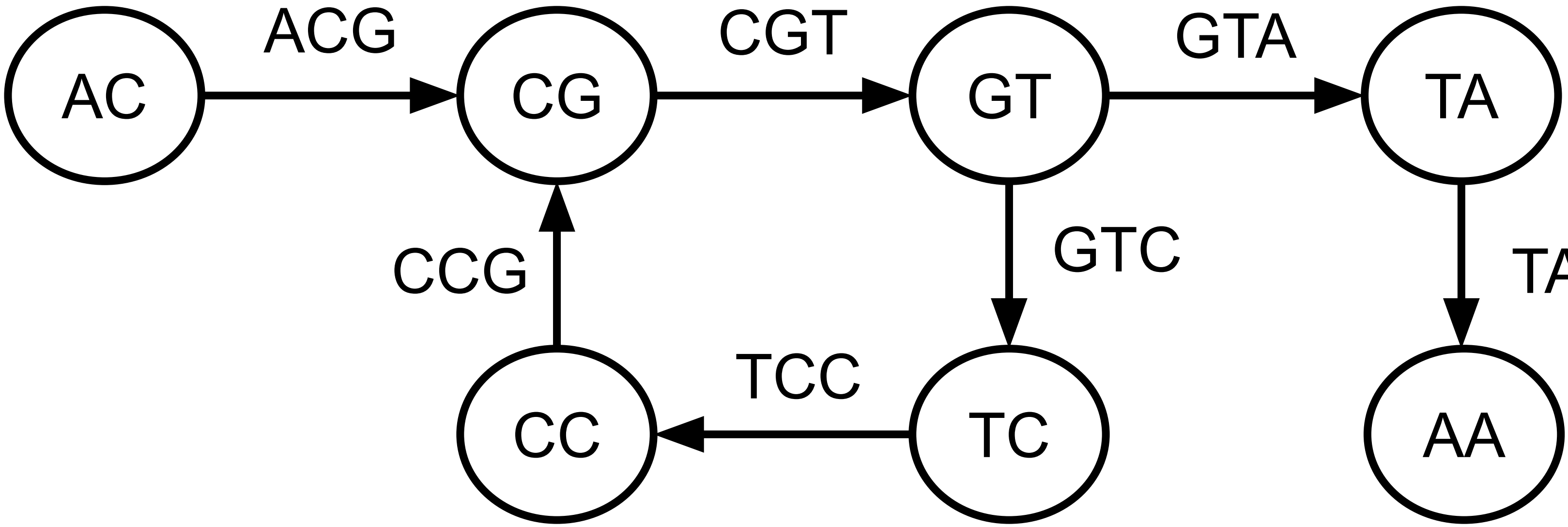
k=2



De Bruijn graph

ACGTCCGTAA

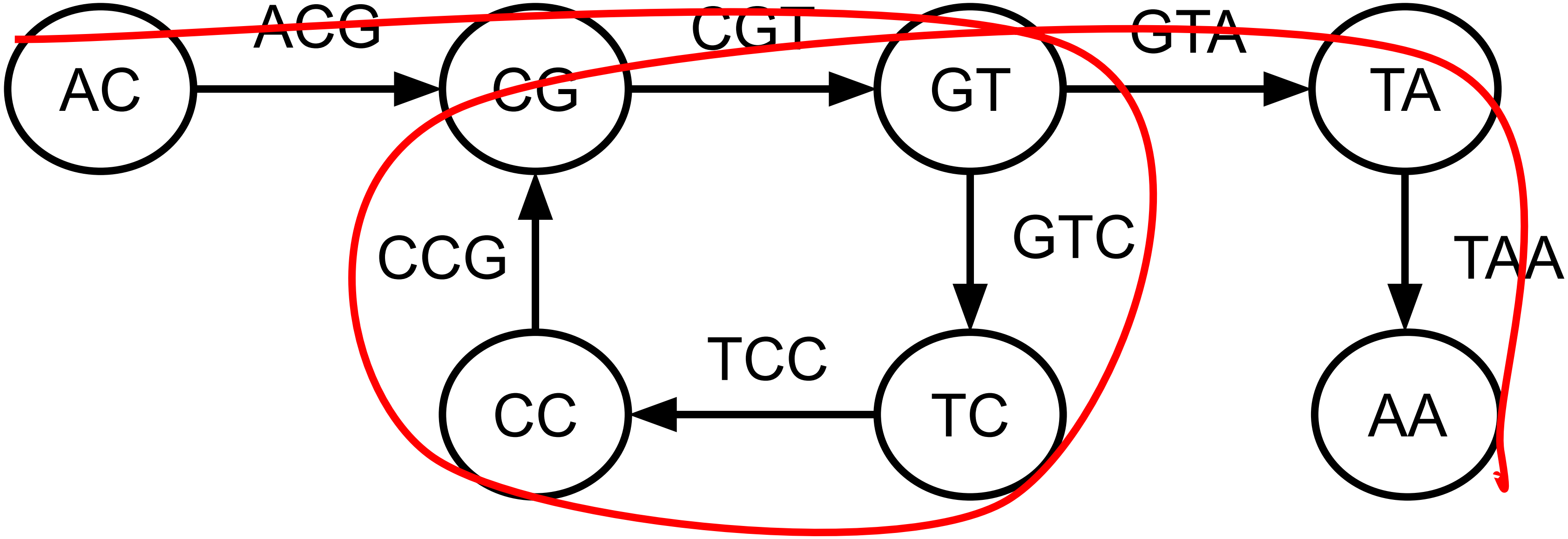
k=2



De Bruijn graph

ACGTCCGTAA

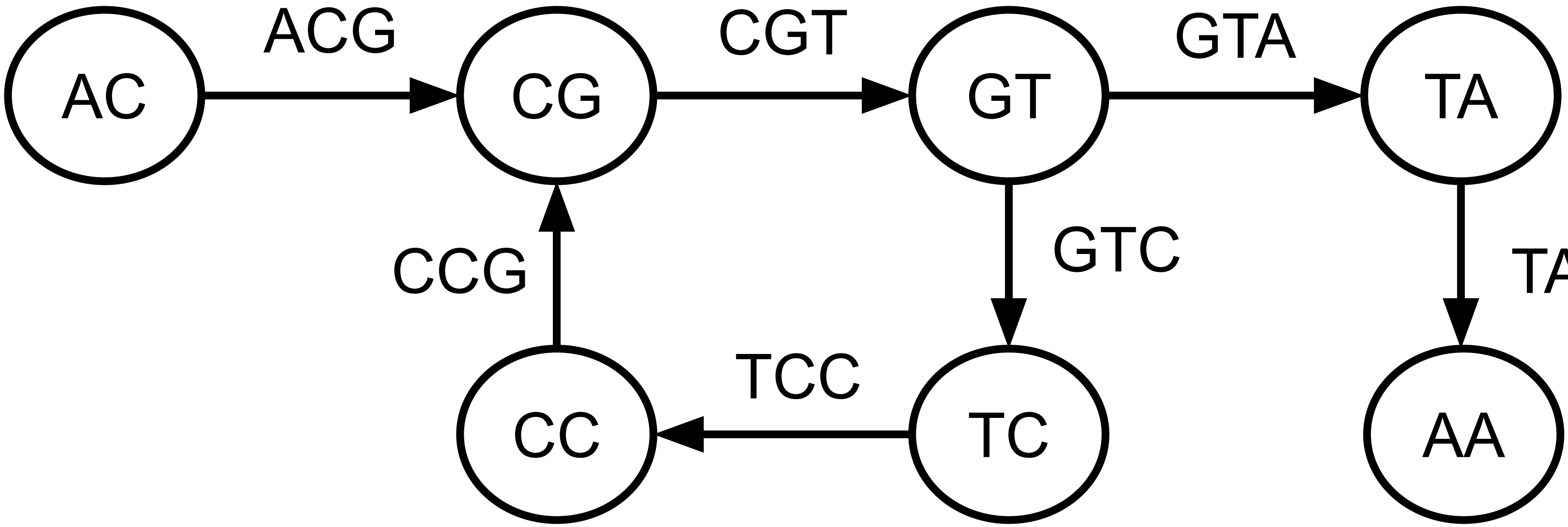
k=2



Condensed de Bruijn graph

ACGTCCGTAA

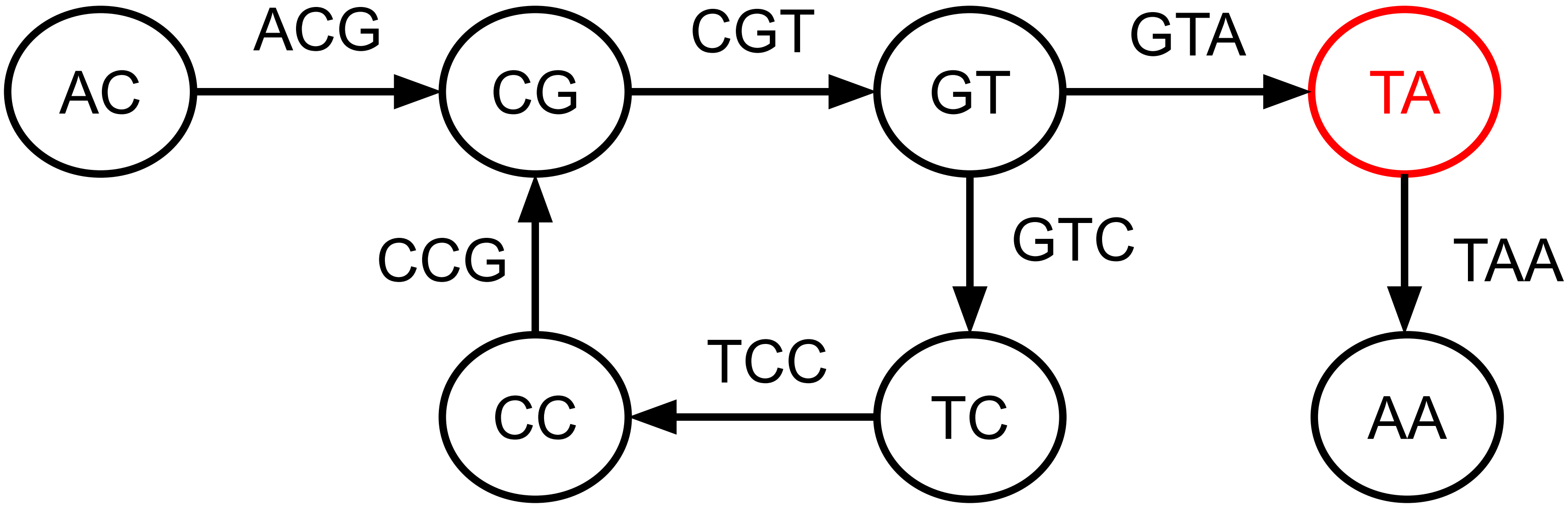
k=2



Condensed de Bruijn graph

ACGTCCGTAA

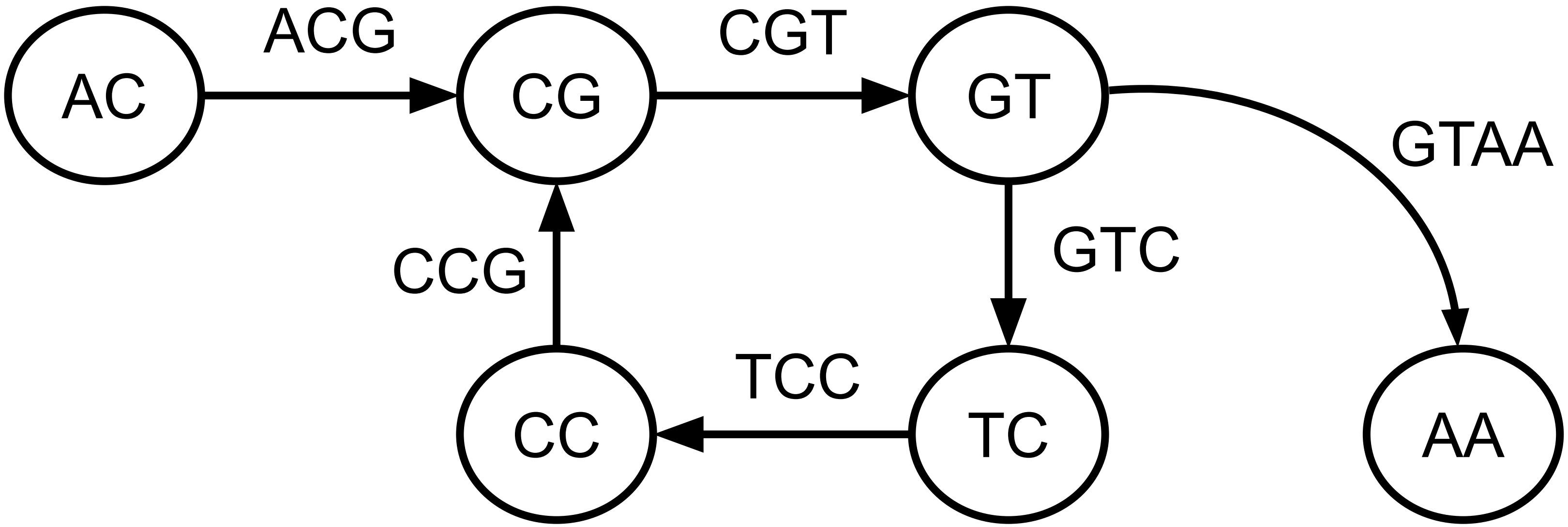
k=2



Condensed de Bruijn graph

ACGTCCGTAA

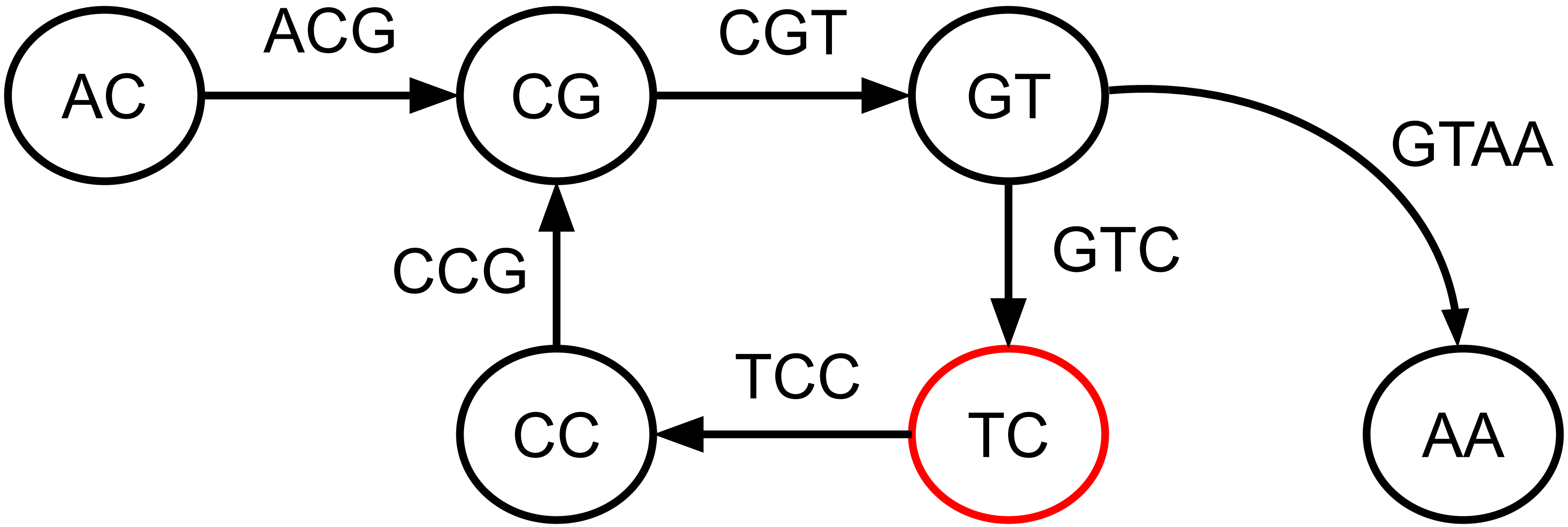
k=2



Condensed de Bruijn graph

ACGTCCGTAA

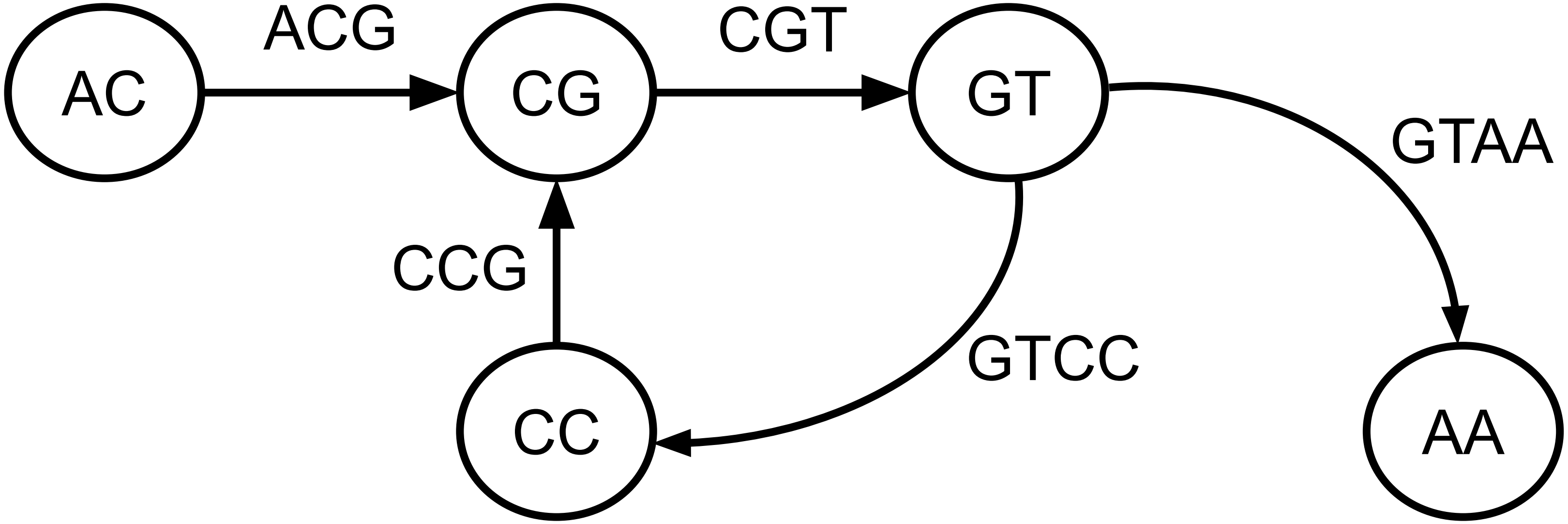
k=2



Condensed de Bruijn graph

ACGTCCGTAA

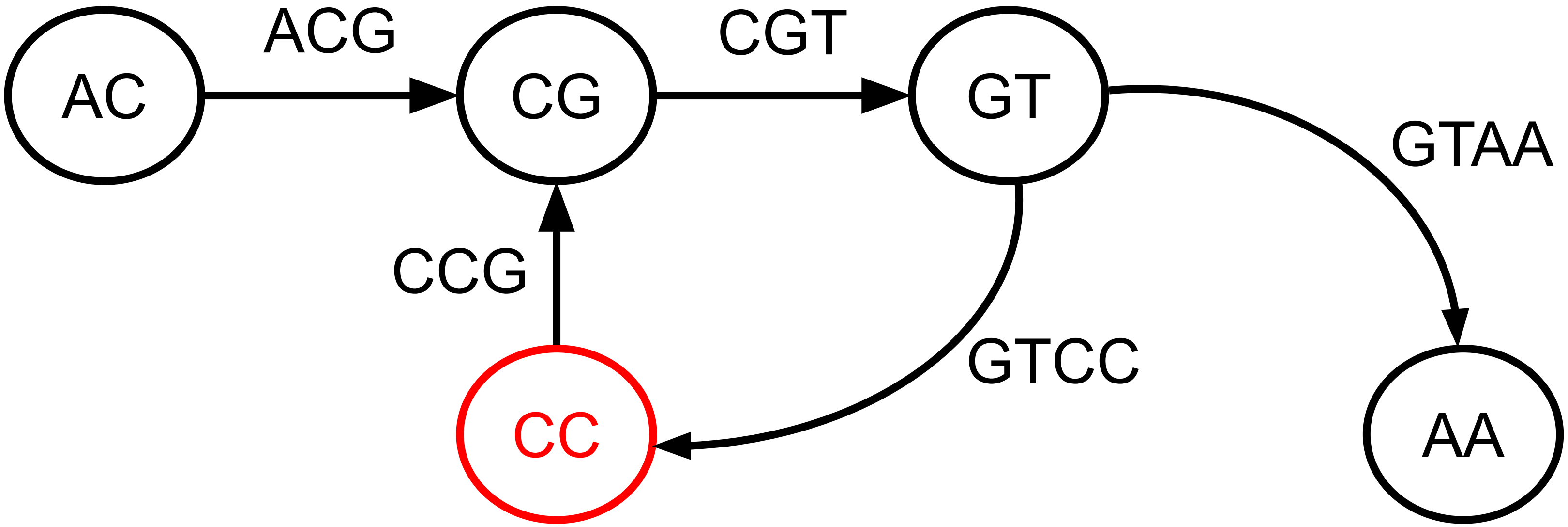
k=2



Condensed de Bruijn graph

ACGTCCGTAA

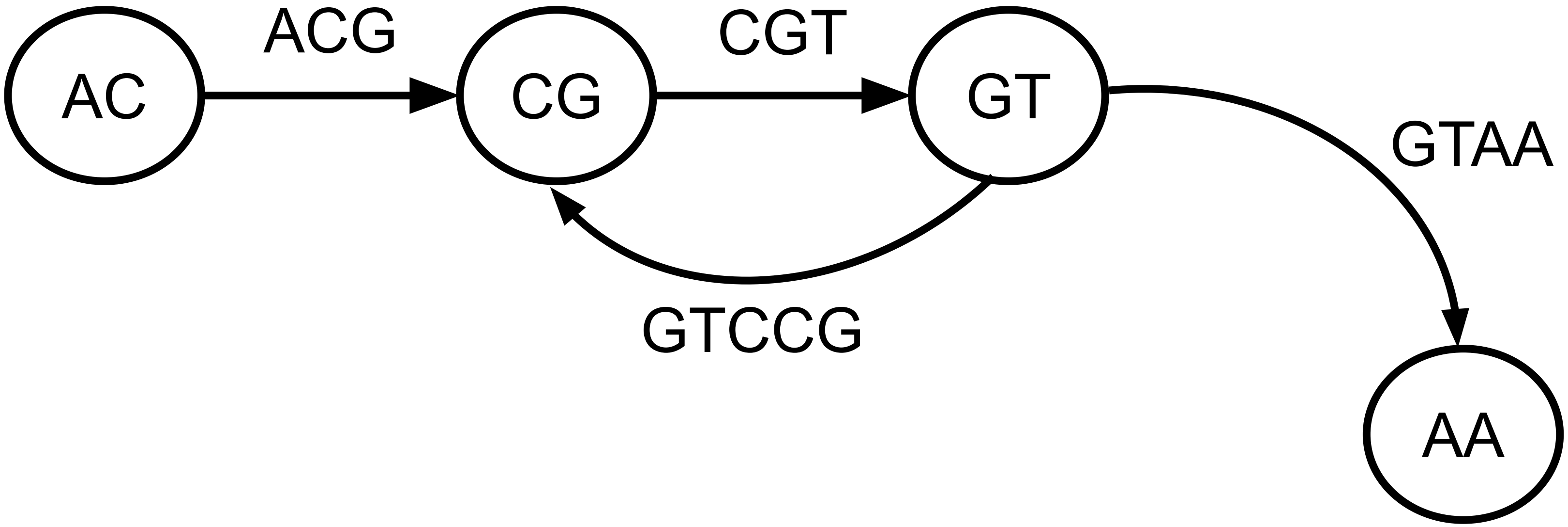
k=2



Condensed de Bruijn graph

ACGTCCGTAA

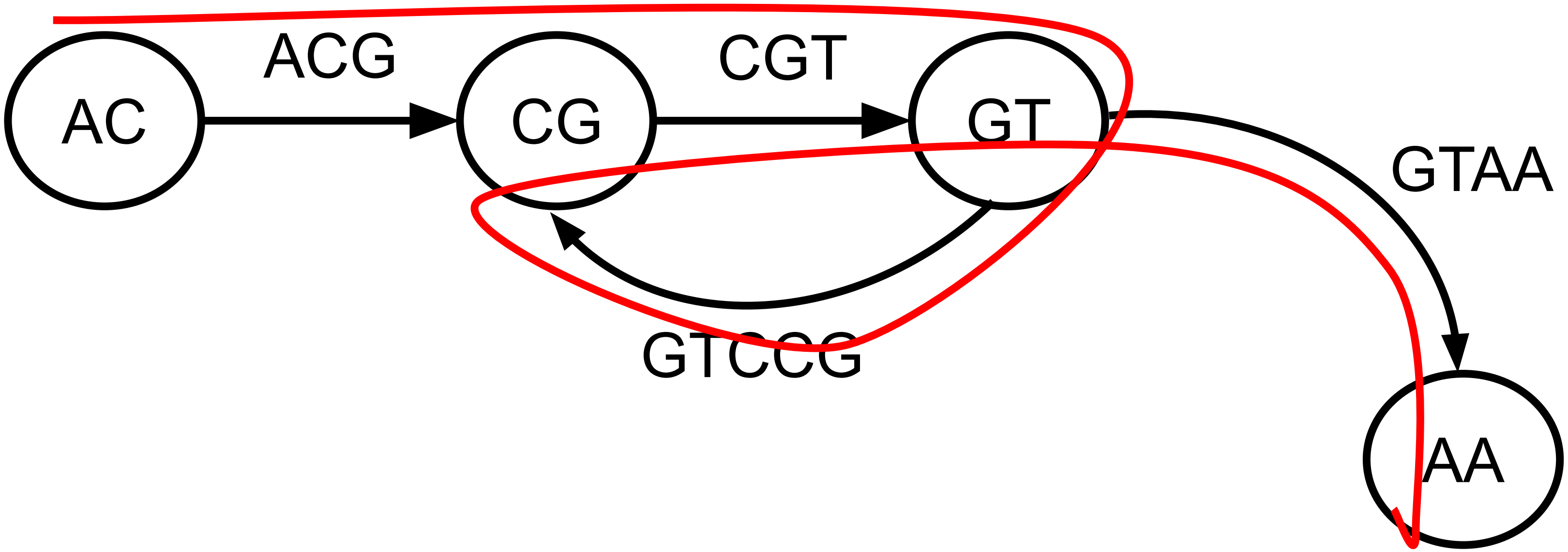
k=2



Condensed de Bruijn graph

ACGTCCGTAA

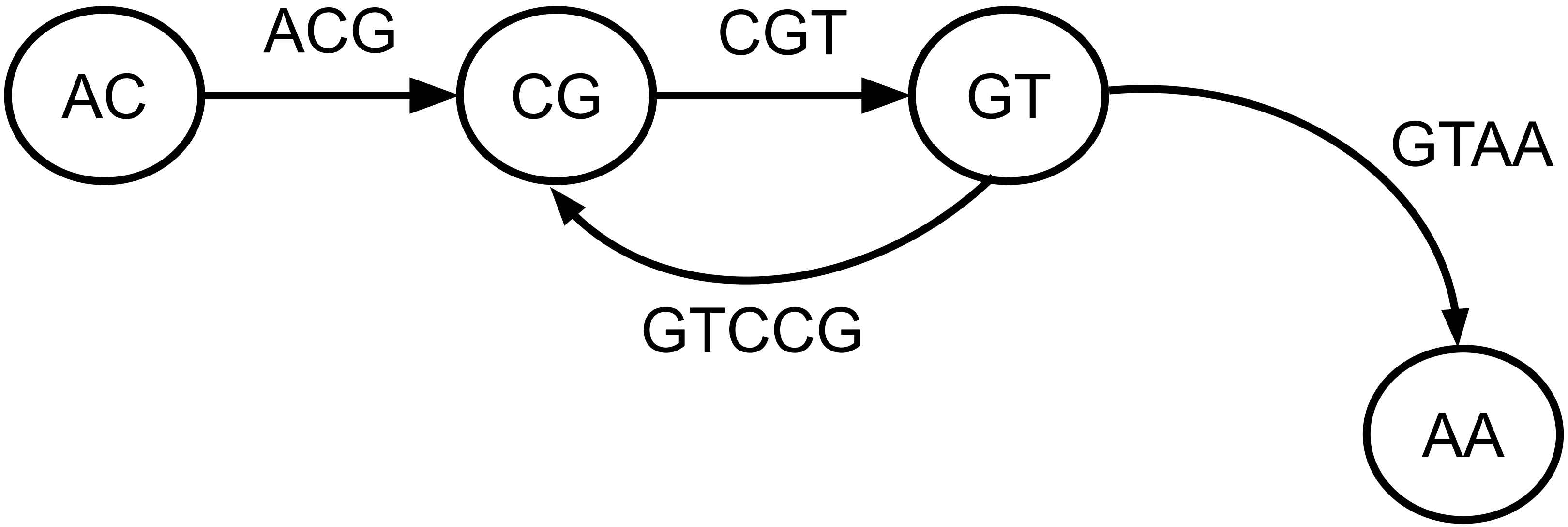
k=2



Repeats in de Bruijn graph

ACGTCCGTAA

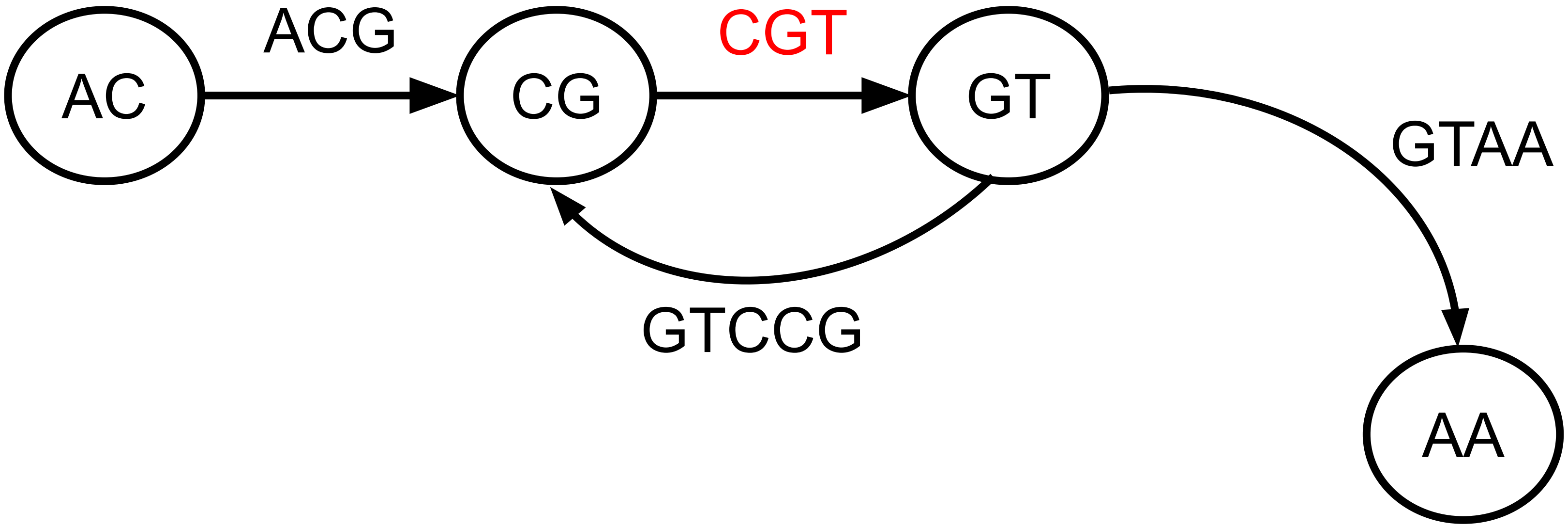
k=2



Repeats in de Bruijn graph

ACGTCCGTAA

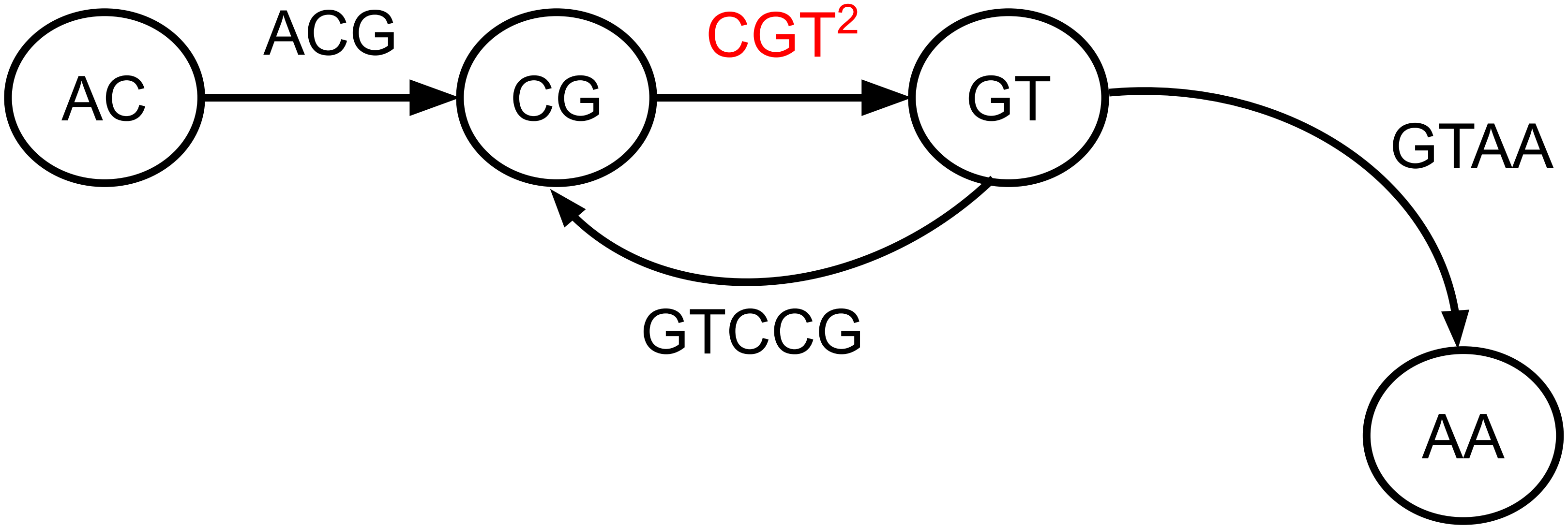
k=2



Repeats in de Bruijn graph

ACGTCCGTAA

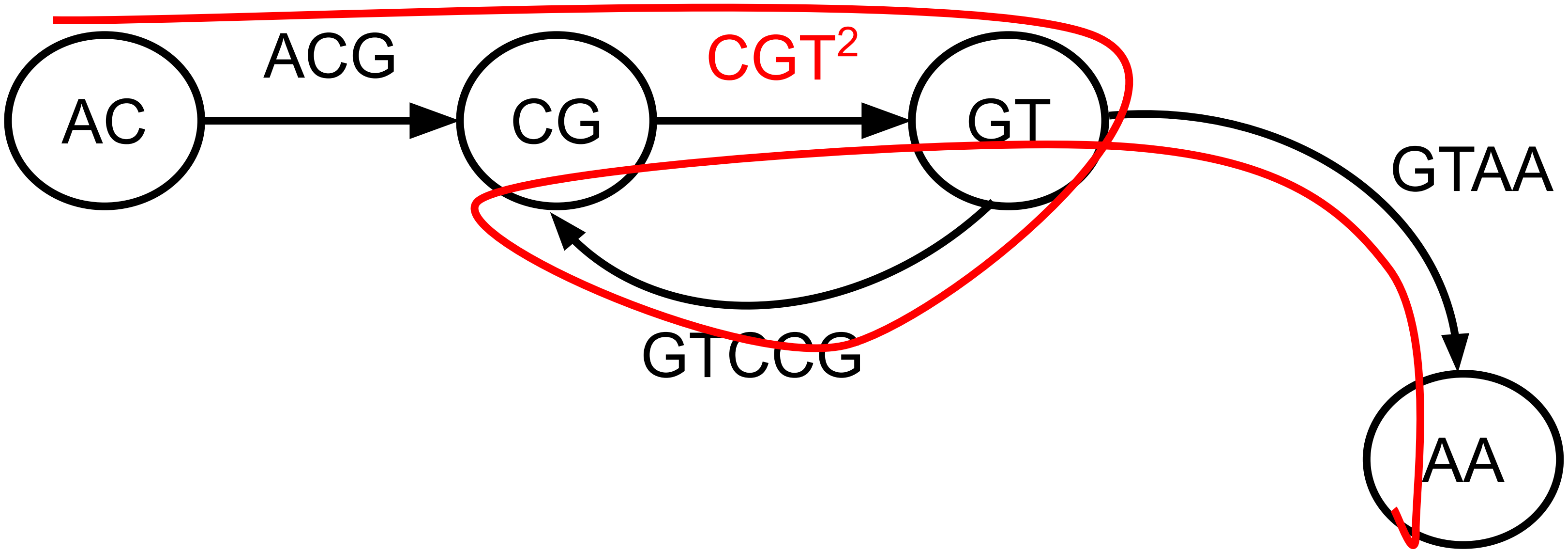
k=2



Eulerian path with multiplicities

ACGTCCGTAA

k=2



More examples

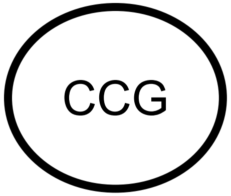
CCGTTG

TGCAGG

GTTGCA

k=3

More examples


CCG

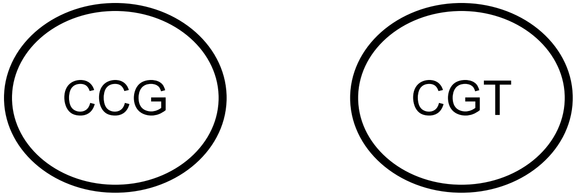
CCGTTG

TGCAGG

GTTGCA

k=3

More examples



CCG

CGT

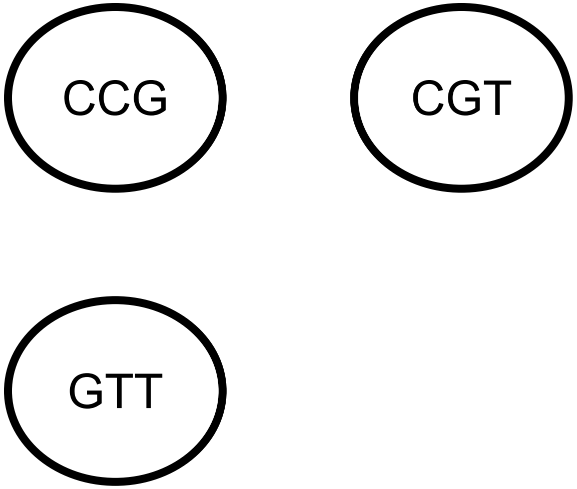
CCGTTG

TGCAGG

GTTGCA

k=3

More examples



CCG

CGT

GTT

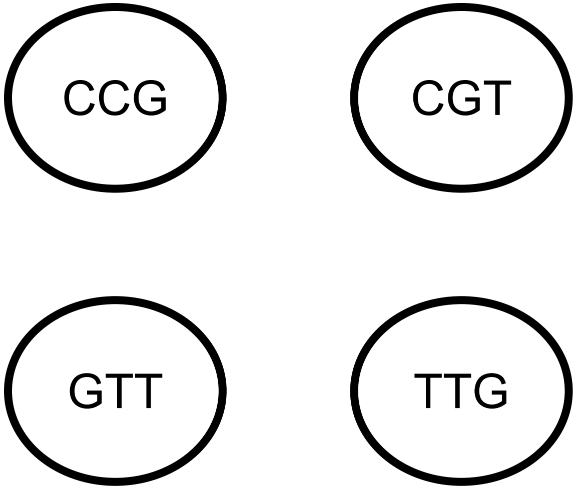
CCGTTG

TGCAGG

GTTGCA

k=3

More examples



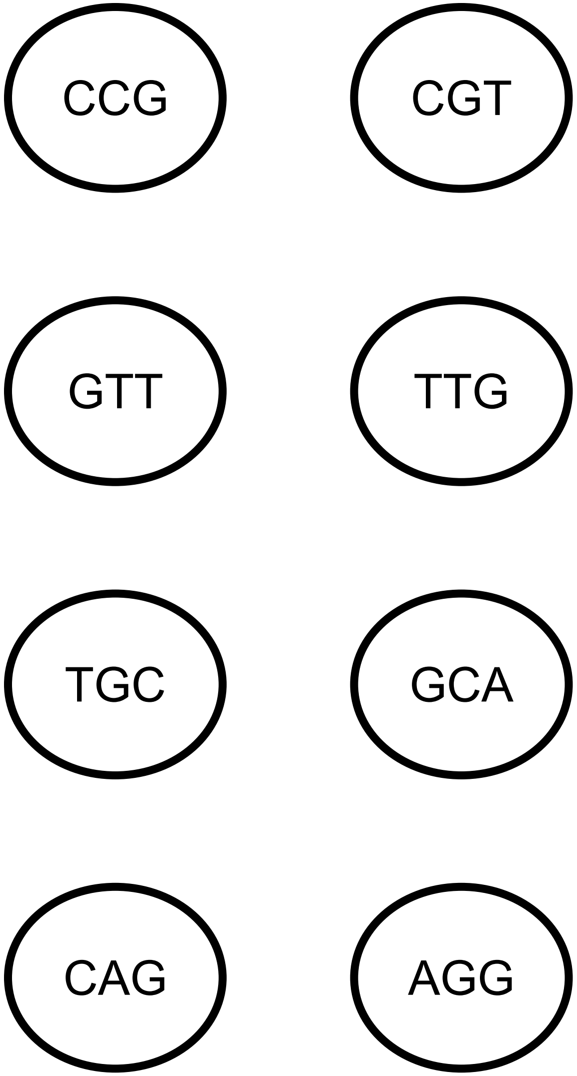
CCGTTG

TGCAGG

GTTGCA

k=3

More examples


CCG
CGT
GTT
TTG
TGC
GCA
CAG
AGG

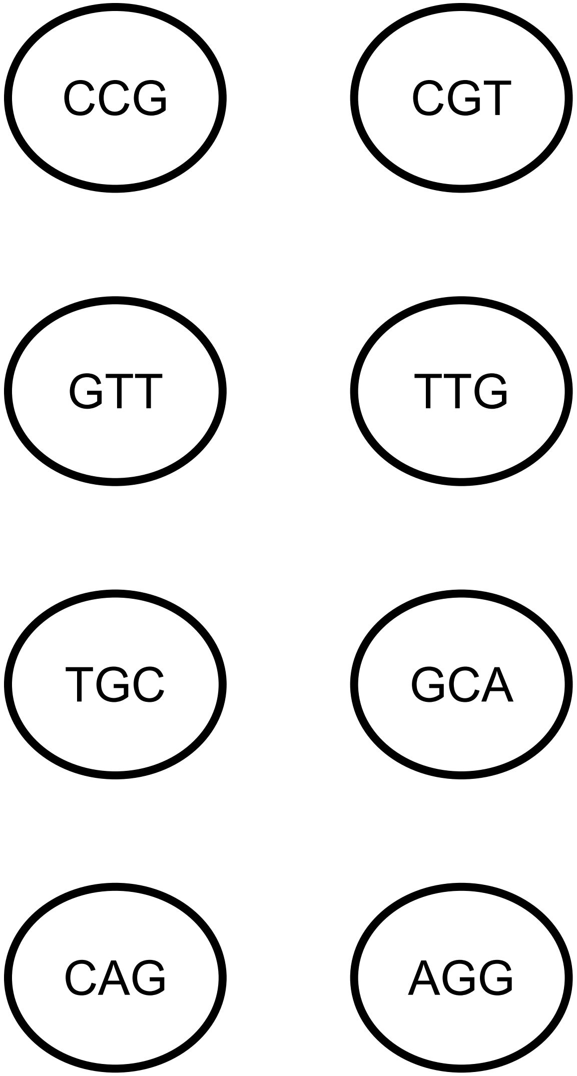
CCGTTG

TGCAGG

GTTGCA

k=3

More examples



CCG

CGT

GTT

TTG

TGC

GCA

CAG

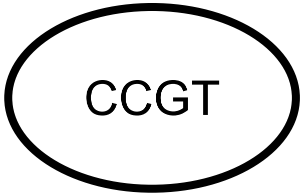
AGG

CCGTTG

TGCAGG

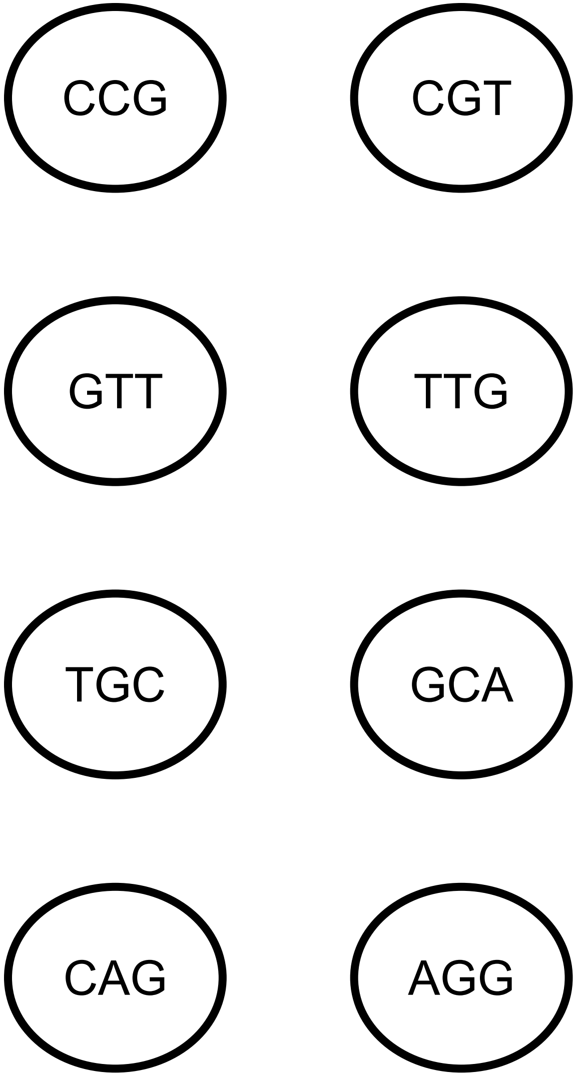
GTTGCA

k=3



CCGT

More examples

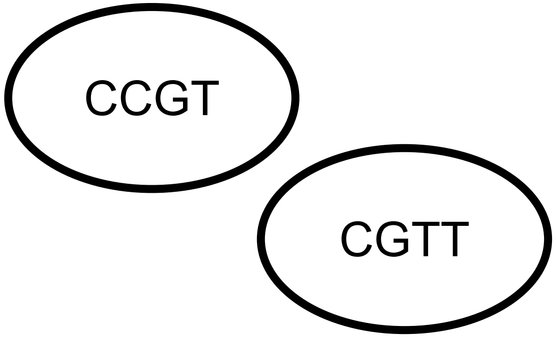


CCGTTG

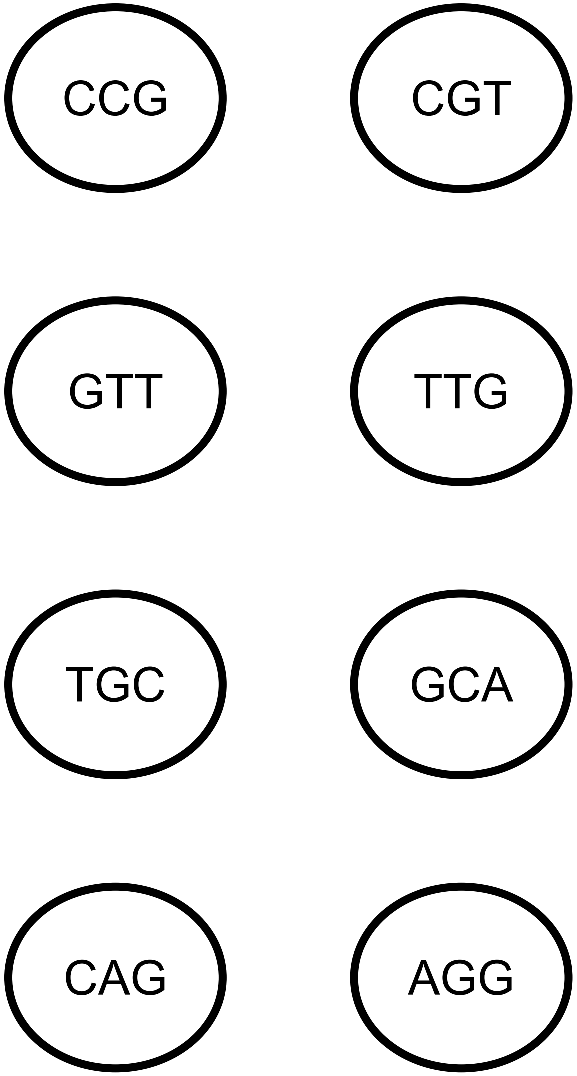
TGCAGG

GTTGCA

k=3



More examples

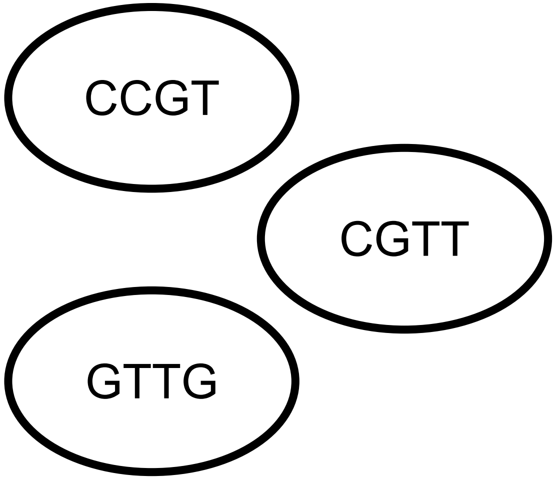


CCGTTG

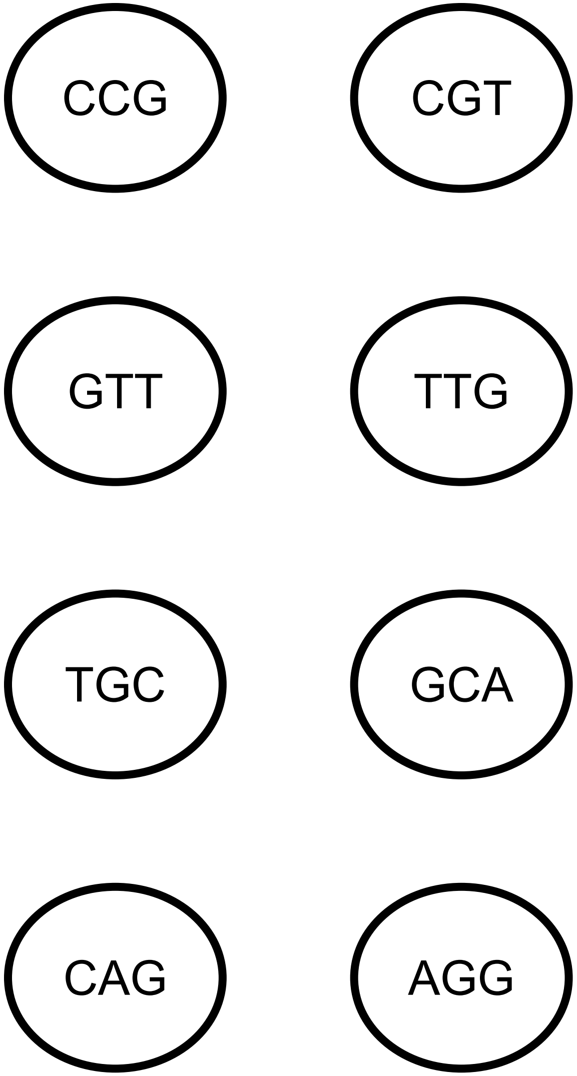
TGCAGG

GTTGCA

k=3



What about real data?

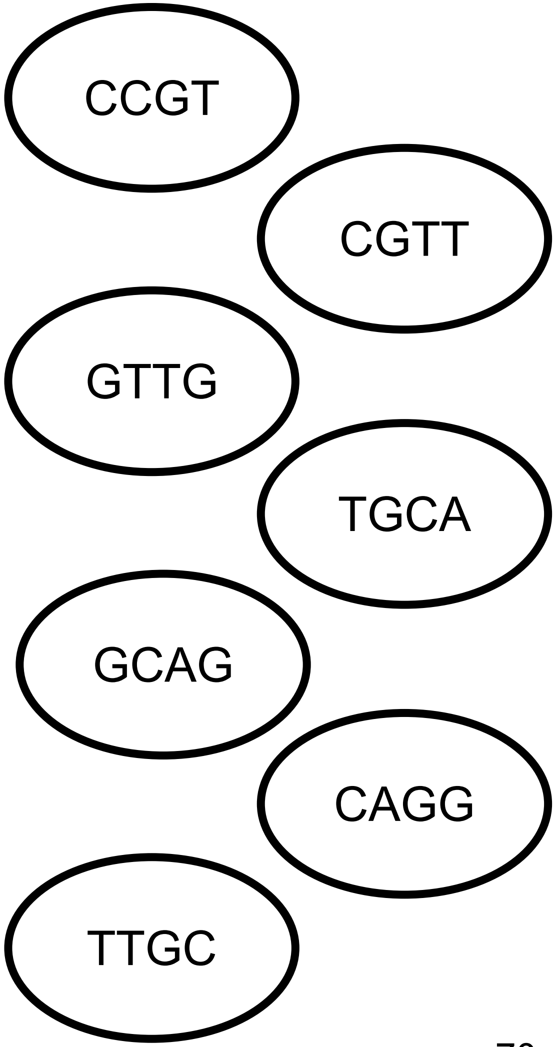


CCGTTG

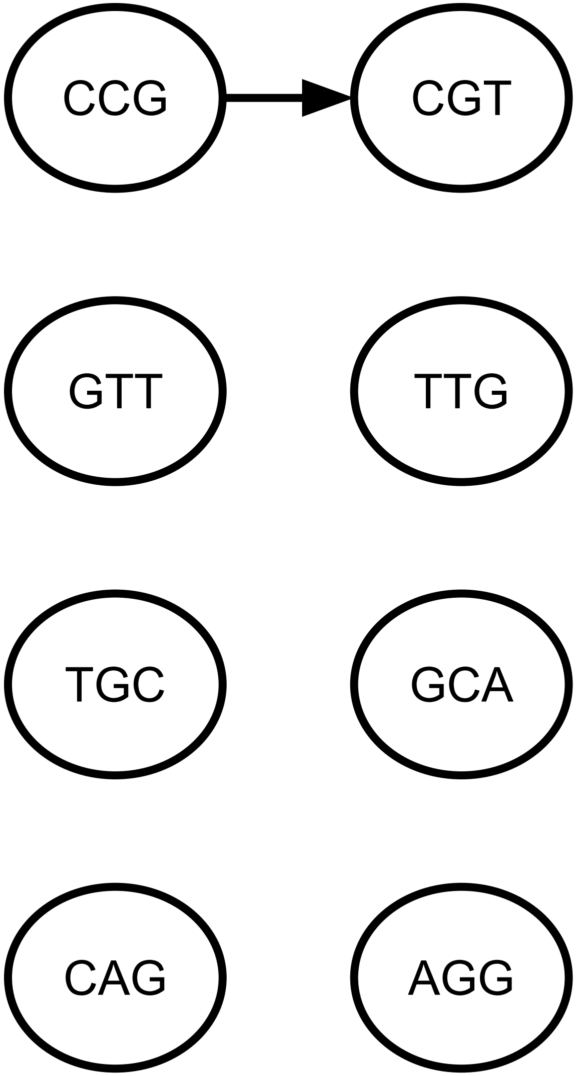
TGCAGG

GTTGCA

k=3



More examples



CCG

CGT

GTT

TTG

TGC

GCA

CAG

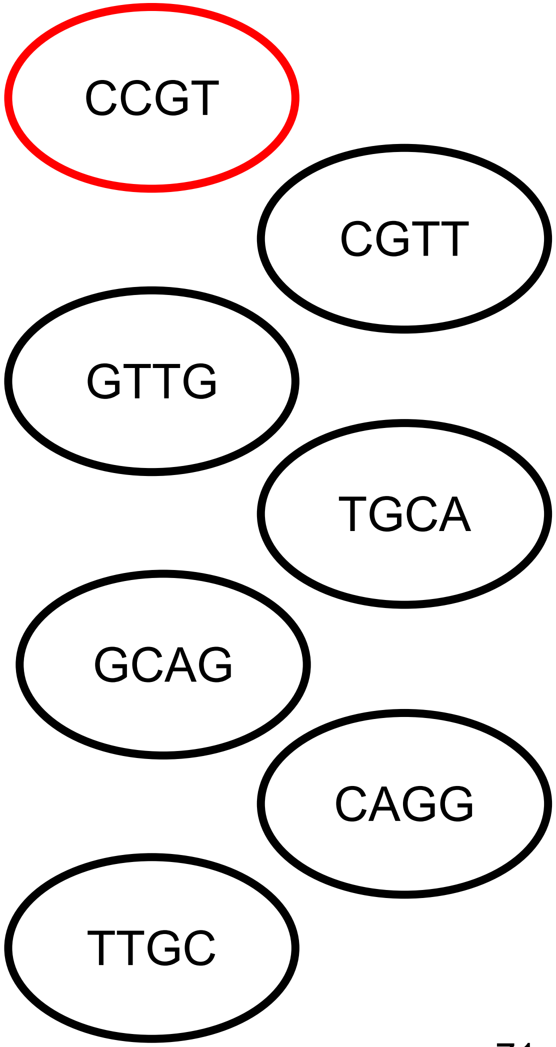
AGG

CCGTTG

TGCAGG

GTTGCA

k=3



CCGT

CGTT

GTTG

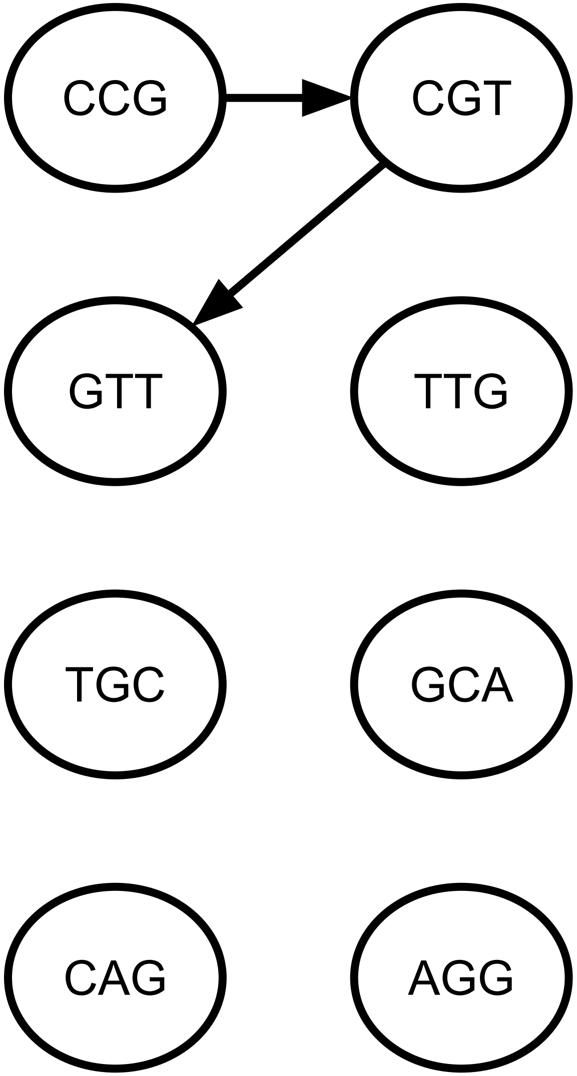
TGCA

GCAG

CAGG

TTGC

What about real data?

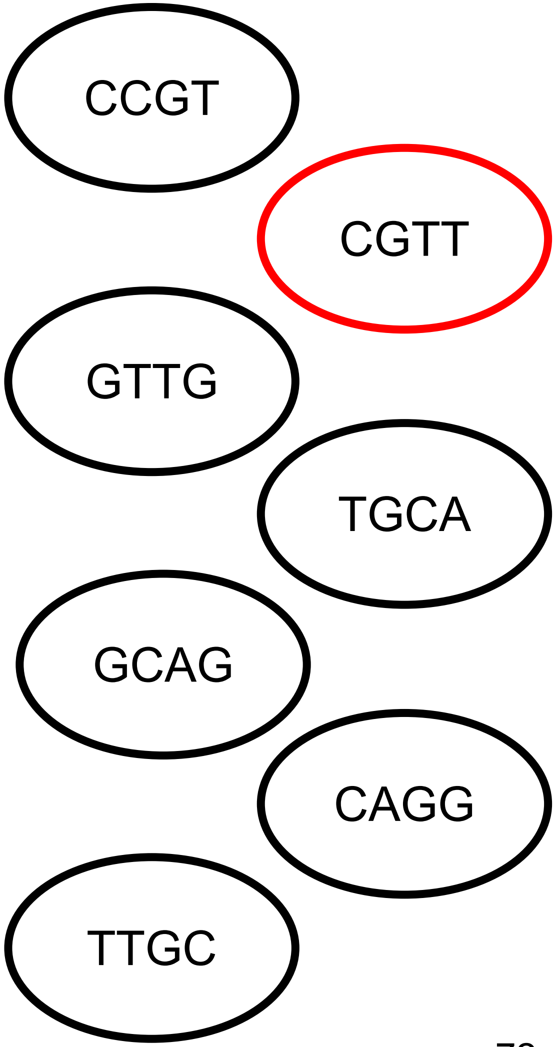


CCGTTG

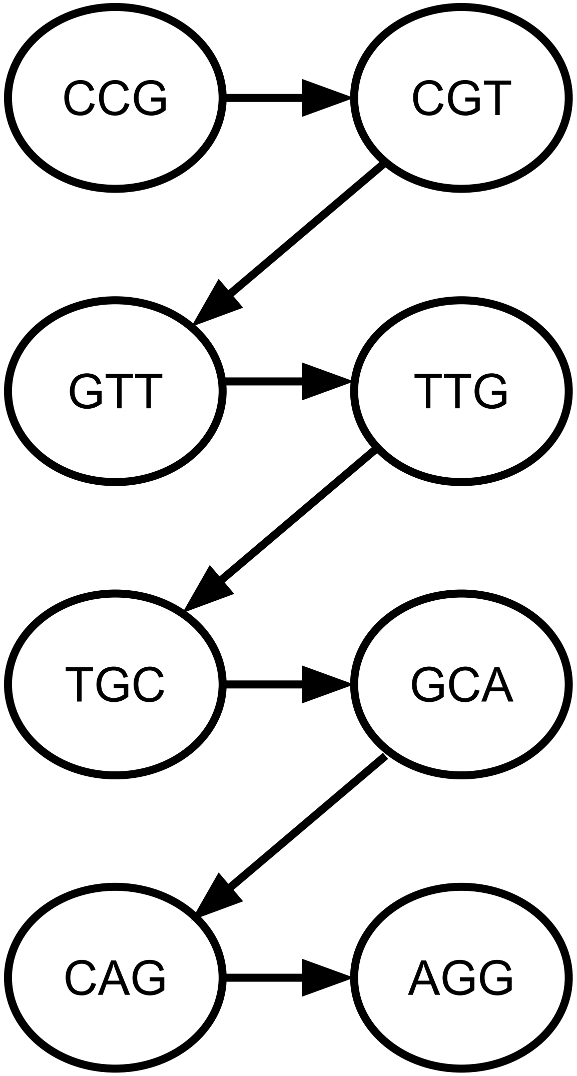
TGCAGG

GTTGCA

k=3

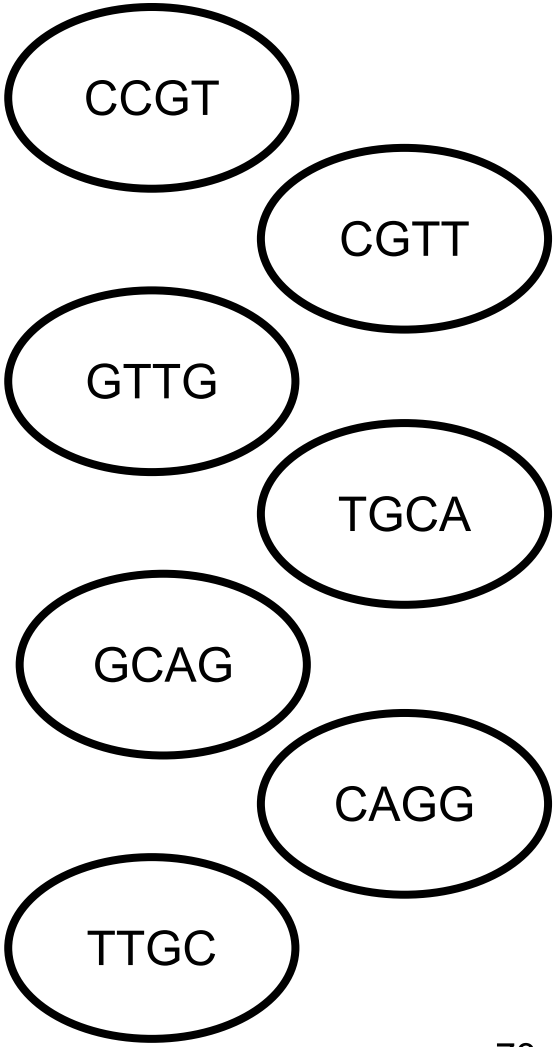


More examples

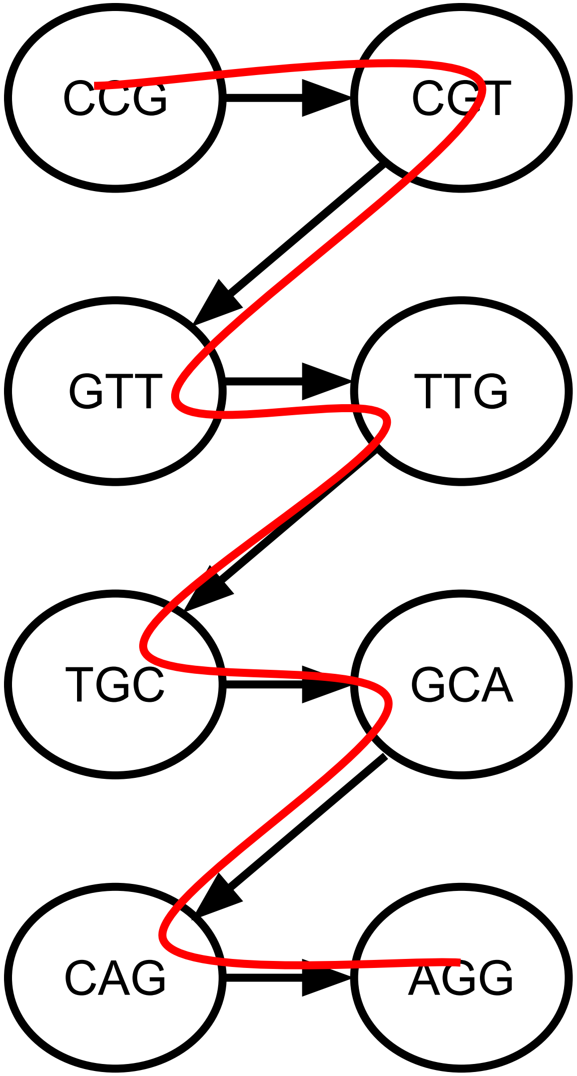


CCGTTG
TGCAGG
GTTGCA

k=3



More examples

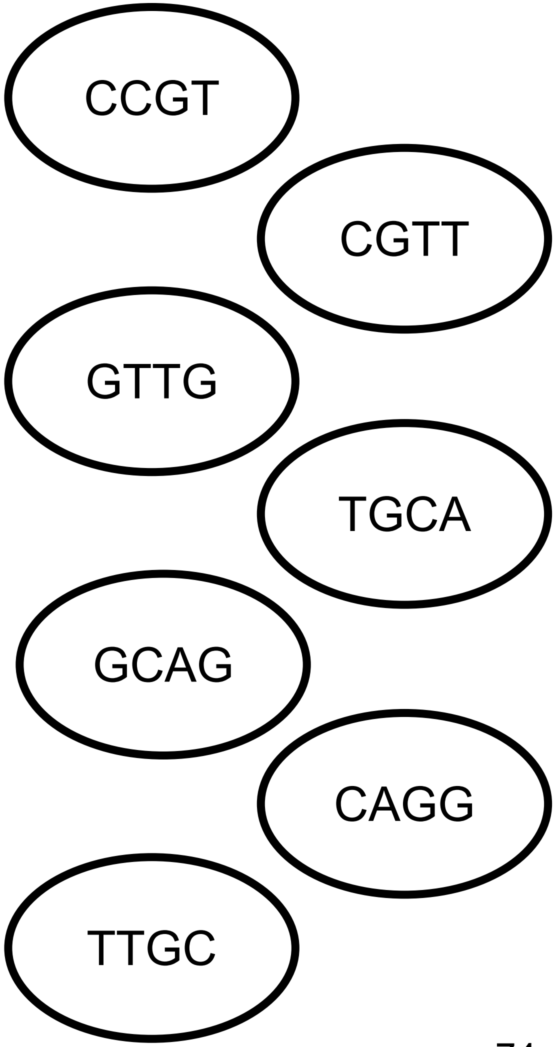


CCGTTG

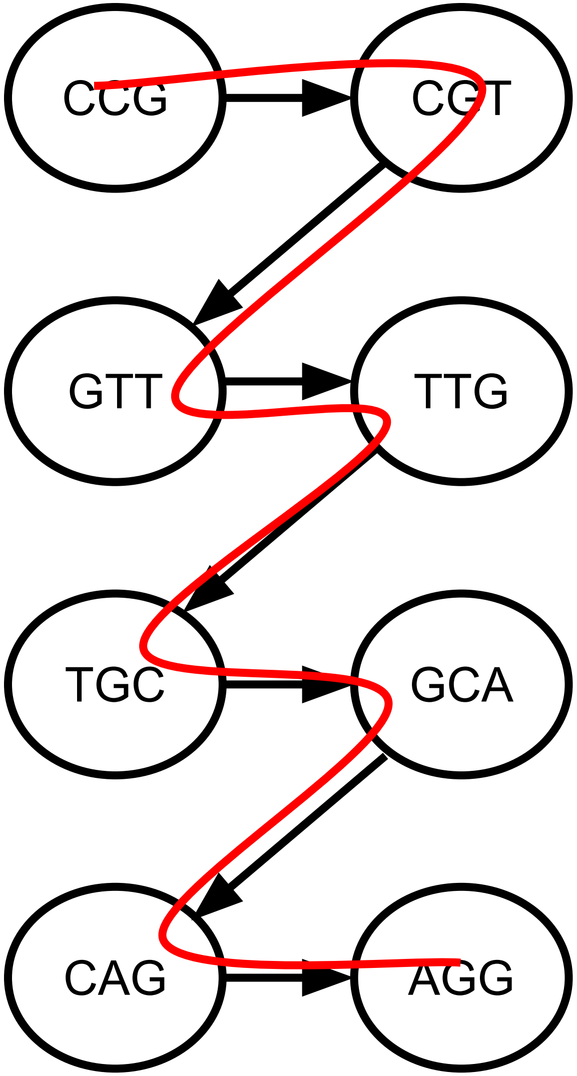
TGCAGG

GTTGCA

k=3



More examples

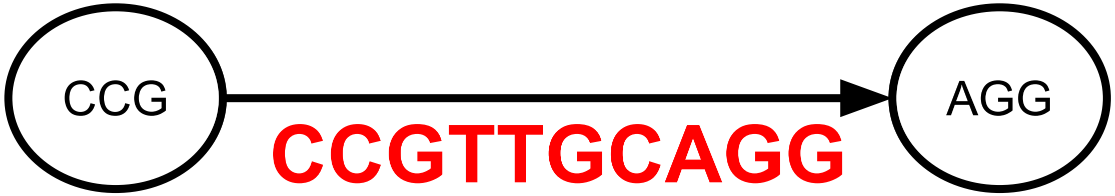


CCGTTG

TGCAGG

GTTGCA

k=3



Does k-mer size matter?

One more example

TTTCATTC

AGCTTTTC

GGGCAATA

AACGGGCA

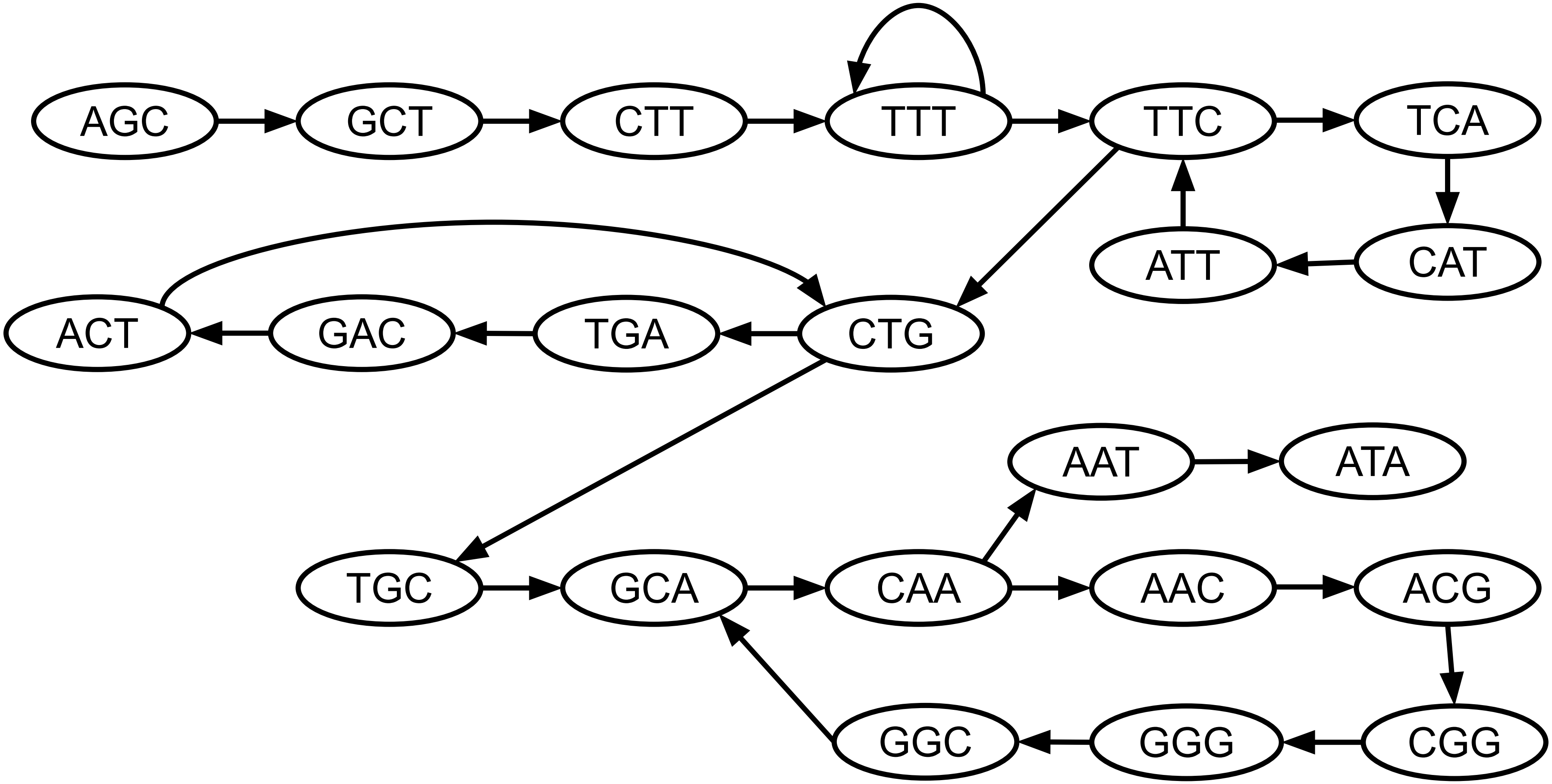
CTGCAACG

TGACTGCA

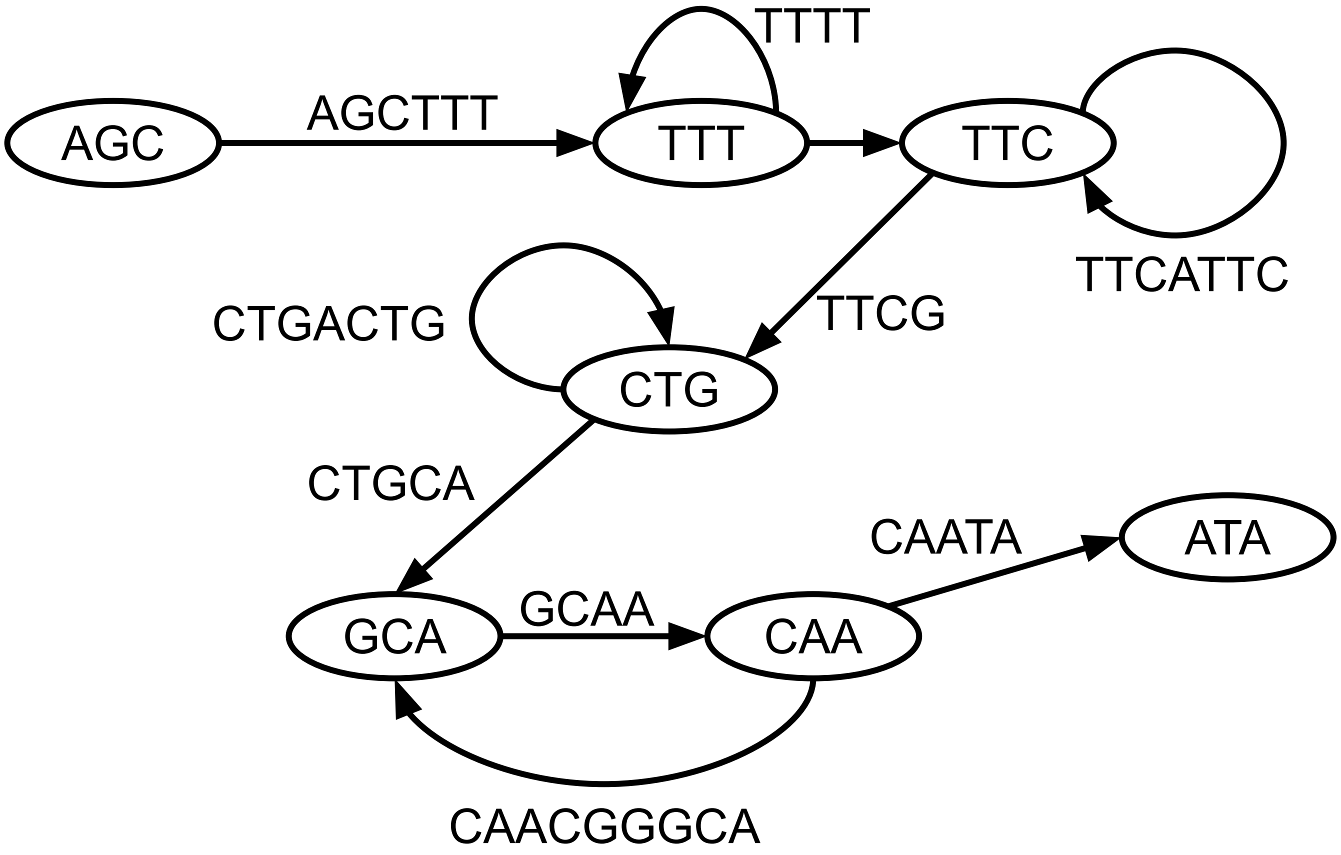
CATTCTGA

K = 3

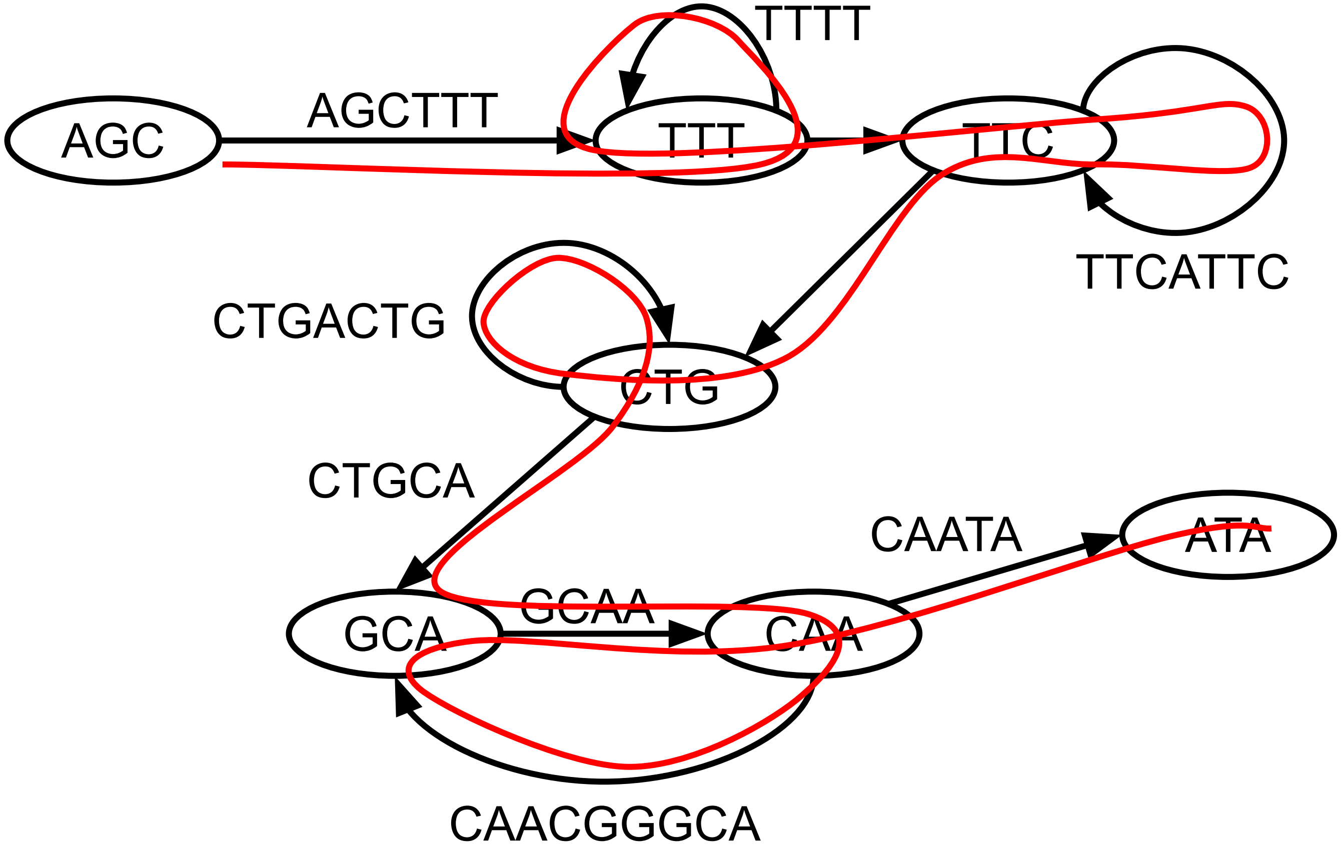
One more example



One more example



One more example



AGCTTTTCATTCTGACTGCAACGGGCAATA

One more example

TTTCATTC

AGCTTTTC

GGGCAATA

AACGGGCA

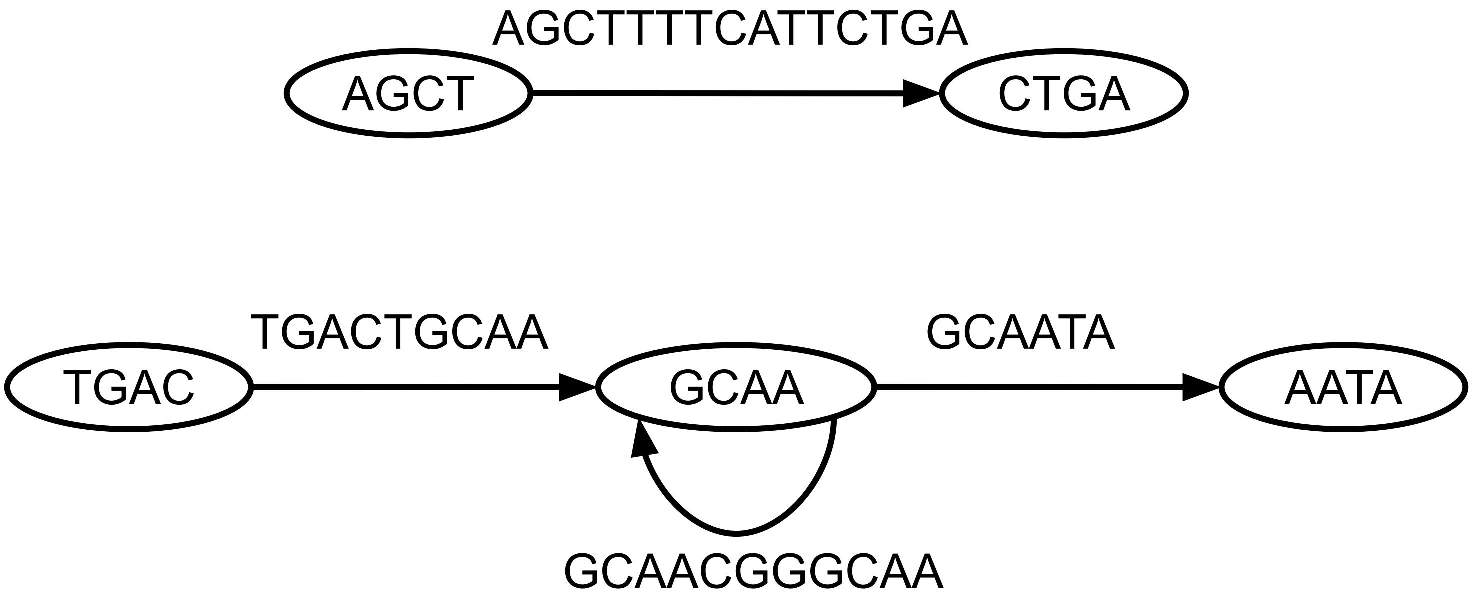
CTGCAACG

TGACTGCA

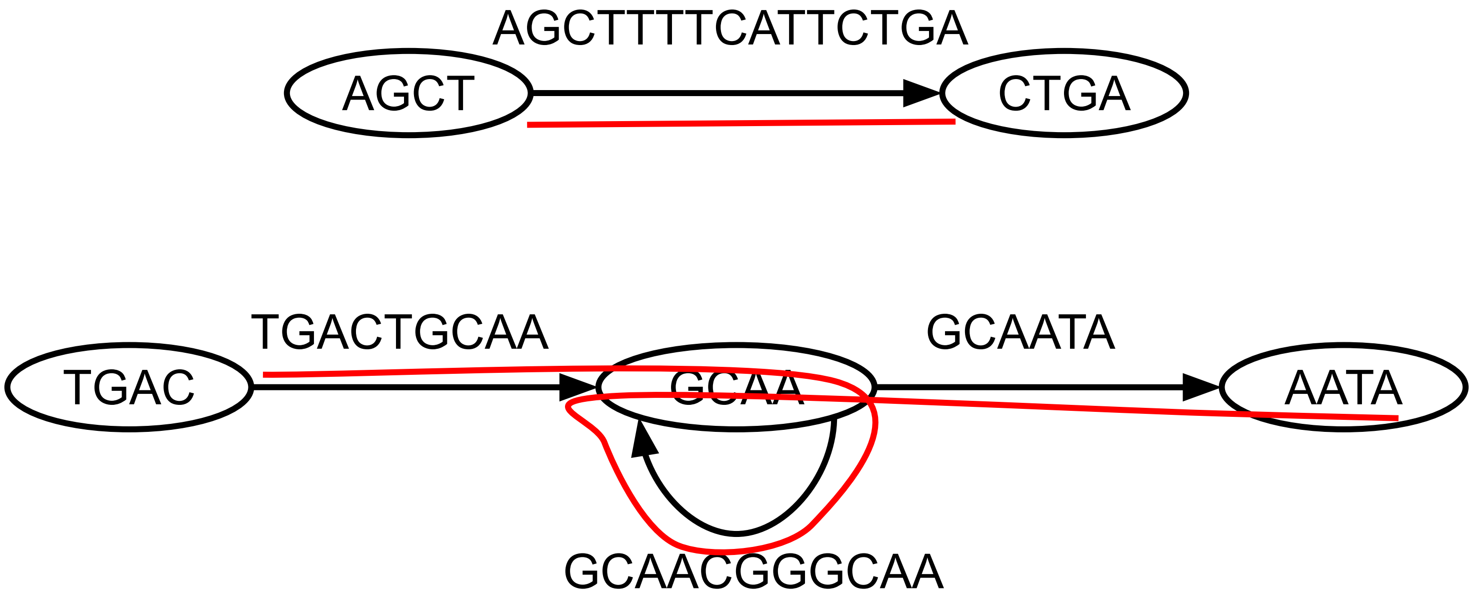
CATTCTGA

K = 4

One more example



One more example



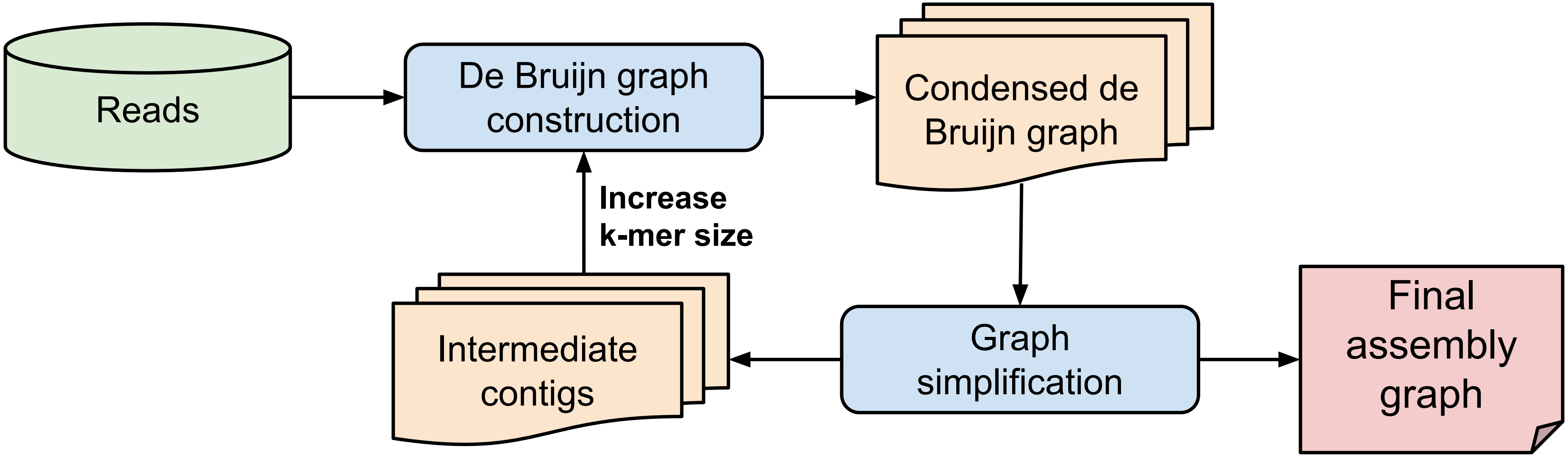
AGCTTTTCATTCTGACTGCAACGGGCAATA

How to select the *k*-mer size?

How to select the *k*-mer size?

- Small k
  - Complex graph
  - Hard to resolve repeats
- Large k
  - Gaps in the assembly
- For normal data sets *k* = *ReadLength* / 2 + ε

Iterative SPAdes run



- Smaller *k*-mer sizes are needed for reconstructing low-coverage regions
- Larger *k*-mer sizes are needed for resolving short repeats

DNA is double-stranded

DNA is double-stranded

- Add k-mer and its reverse complement
- Use odd k to avoid self-complement vertices
    - rc(AA**T**TT) = AA**A**TT
    - rc(AATT) = AATT

Removing sequencing errors

Sequencing errors in de Bruijn graph

```
CCGTTG
 CGTTAC
  GTTGCA
    TGCAGG
```

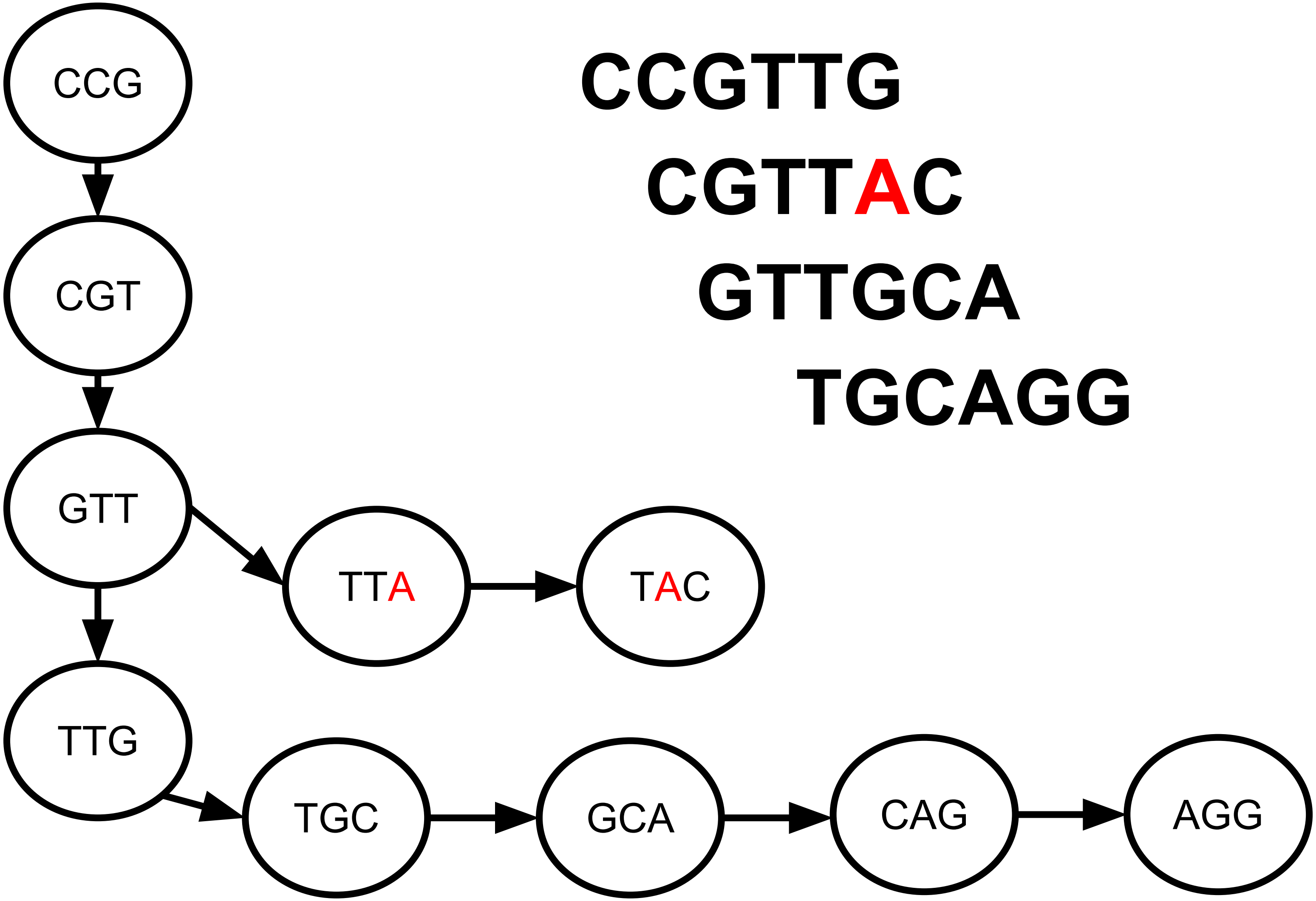
Sequencing errors in de Bruijn graph

CCGTTG

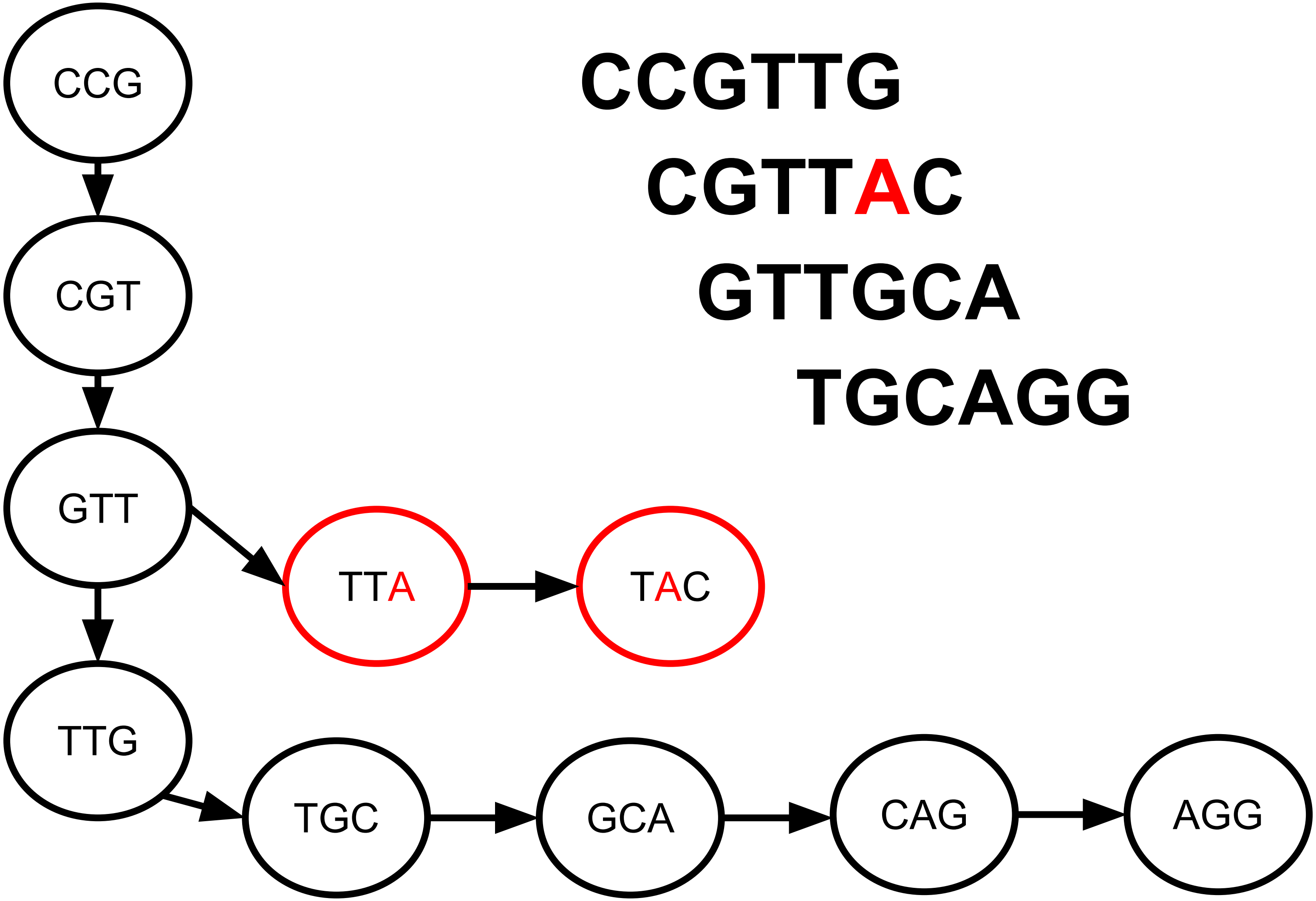
CGTTAC

GTTGCA

TGCAGG



Sequencing errors in de Bruijn graph



CCGTTG

CGTTAC

GTTGCA

TGCAGG

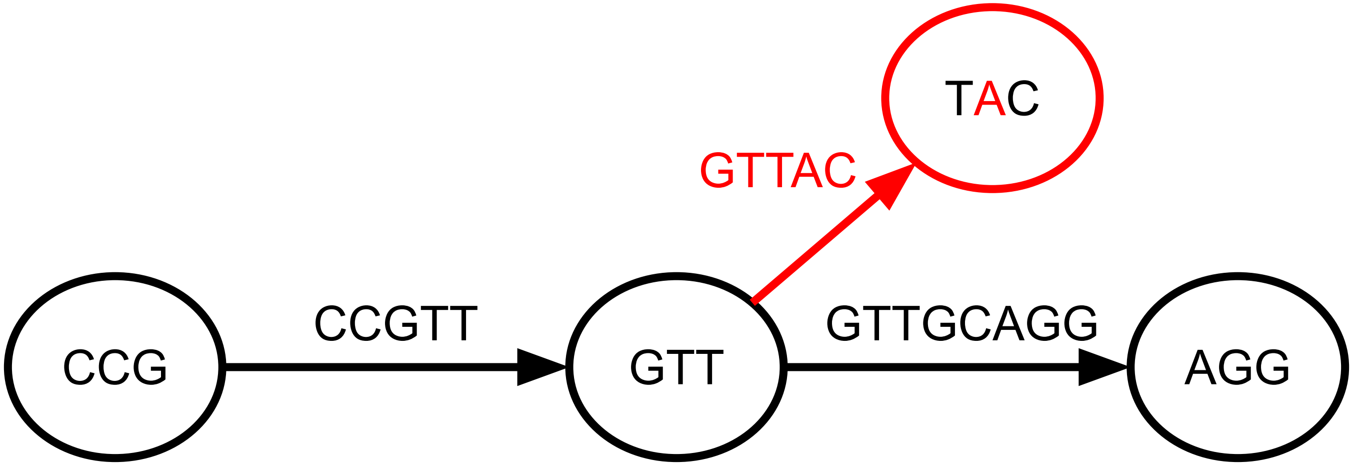
Sequencing errors in de Bruijn graph

CCGTTG

CGTTAC

GTTGCA

TGCAGG



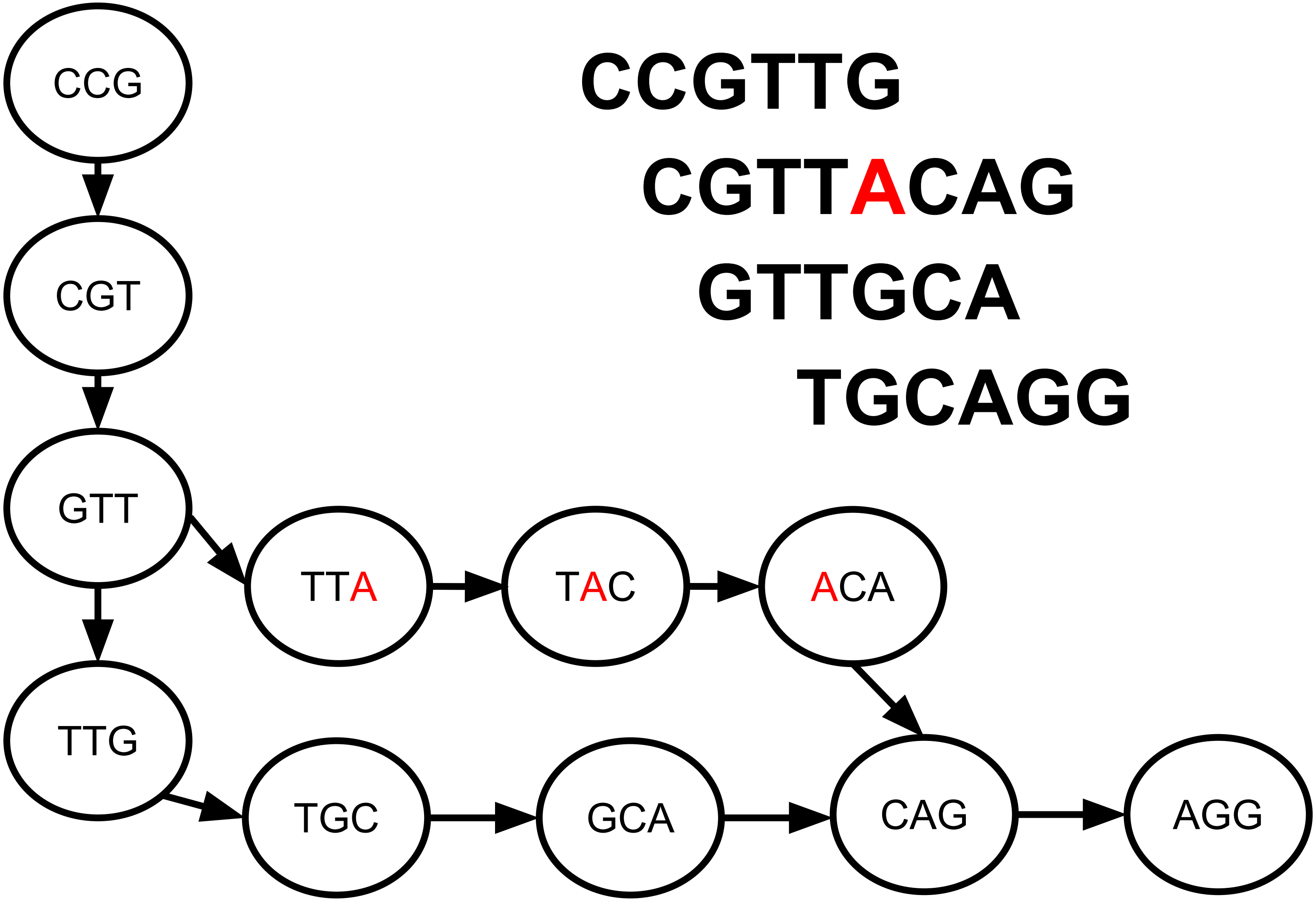
How to remove a tip?

How to remove a tip?

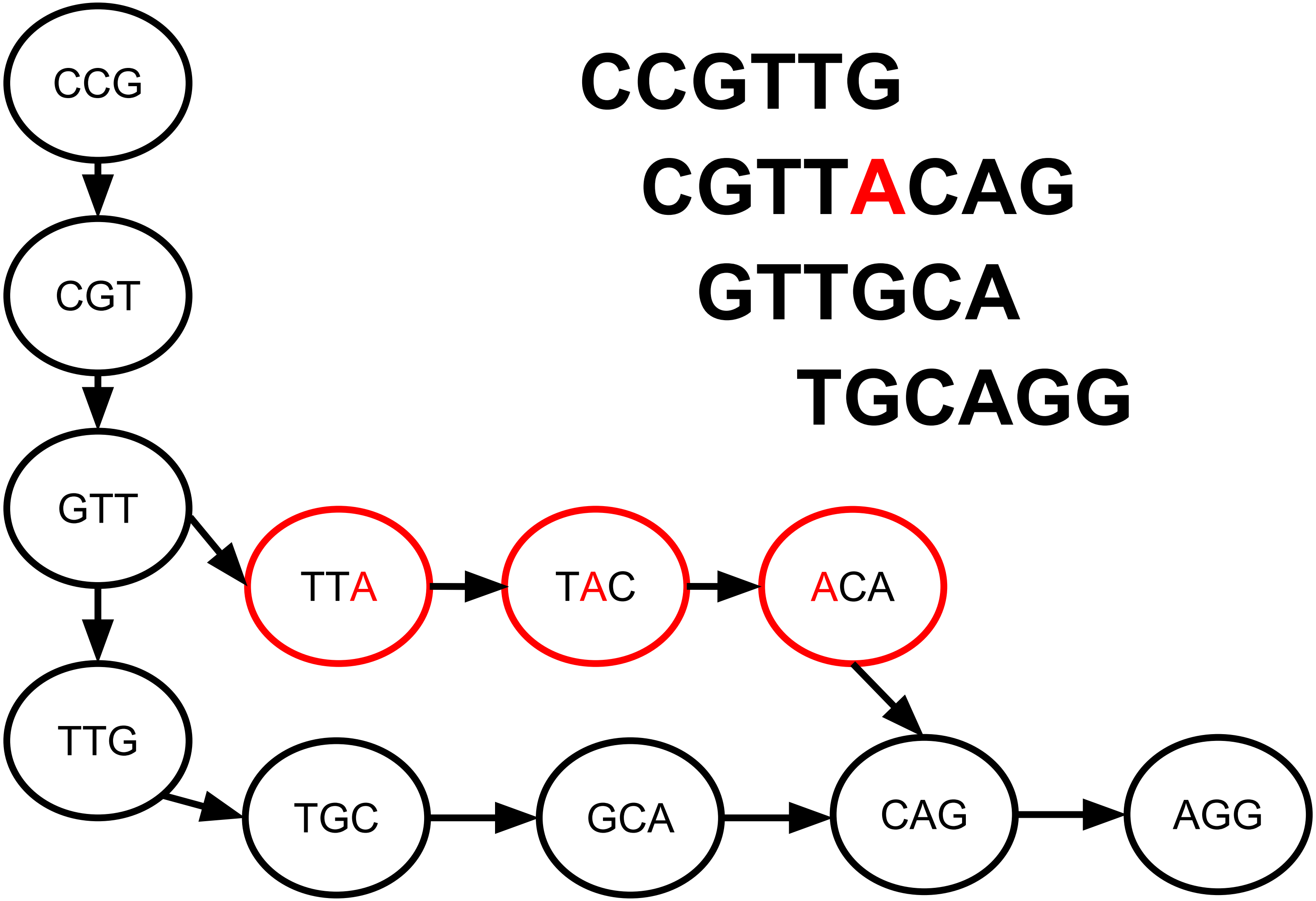
- Short length (usually less than 2 * k)
- Low coverage in respect to the main (correct path)
- Long length or high coverage — more likely to indicate a coverage gap

More about sequencing errors

CCGTTG
CGTTACAG
GTTGCA
TGCAGG



More about sequencing errors



CCGTTG

CGTTACAG

GTTGCA

TGCAGG

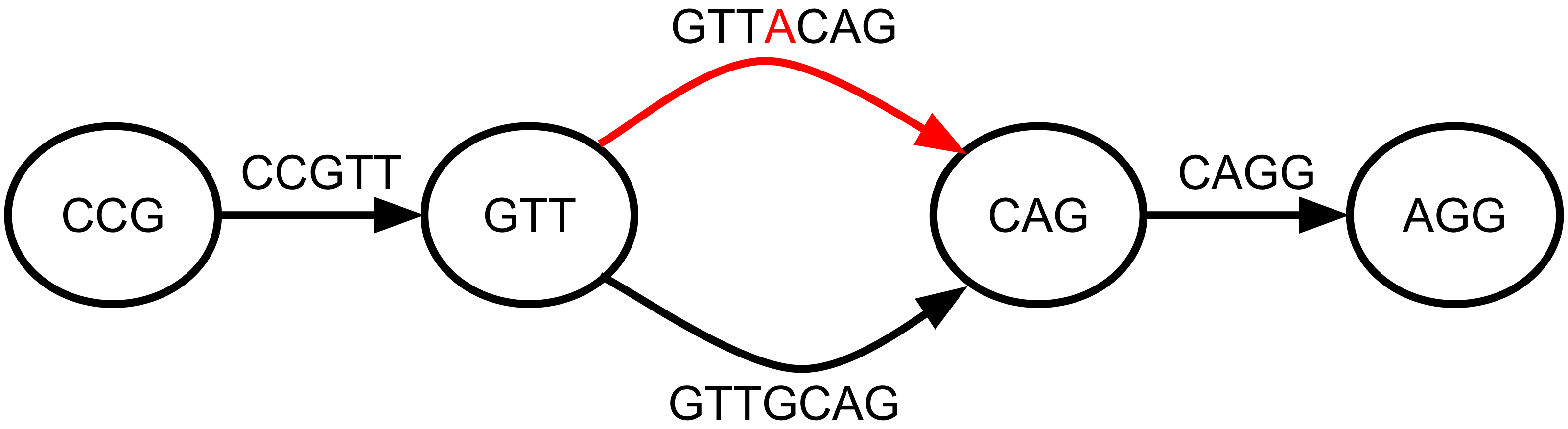
More about sequencing errors

CCGTTG

CGTTACAG

GTTGCA

TGCAGG



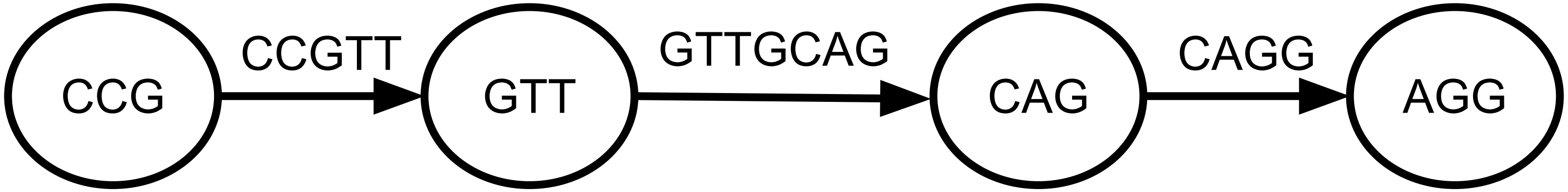
More about sequencing errors

CCGTTG

CGTTACAG

GTTGCA

TGCAGG

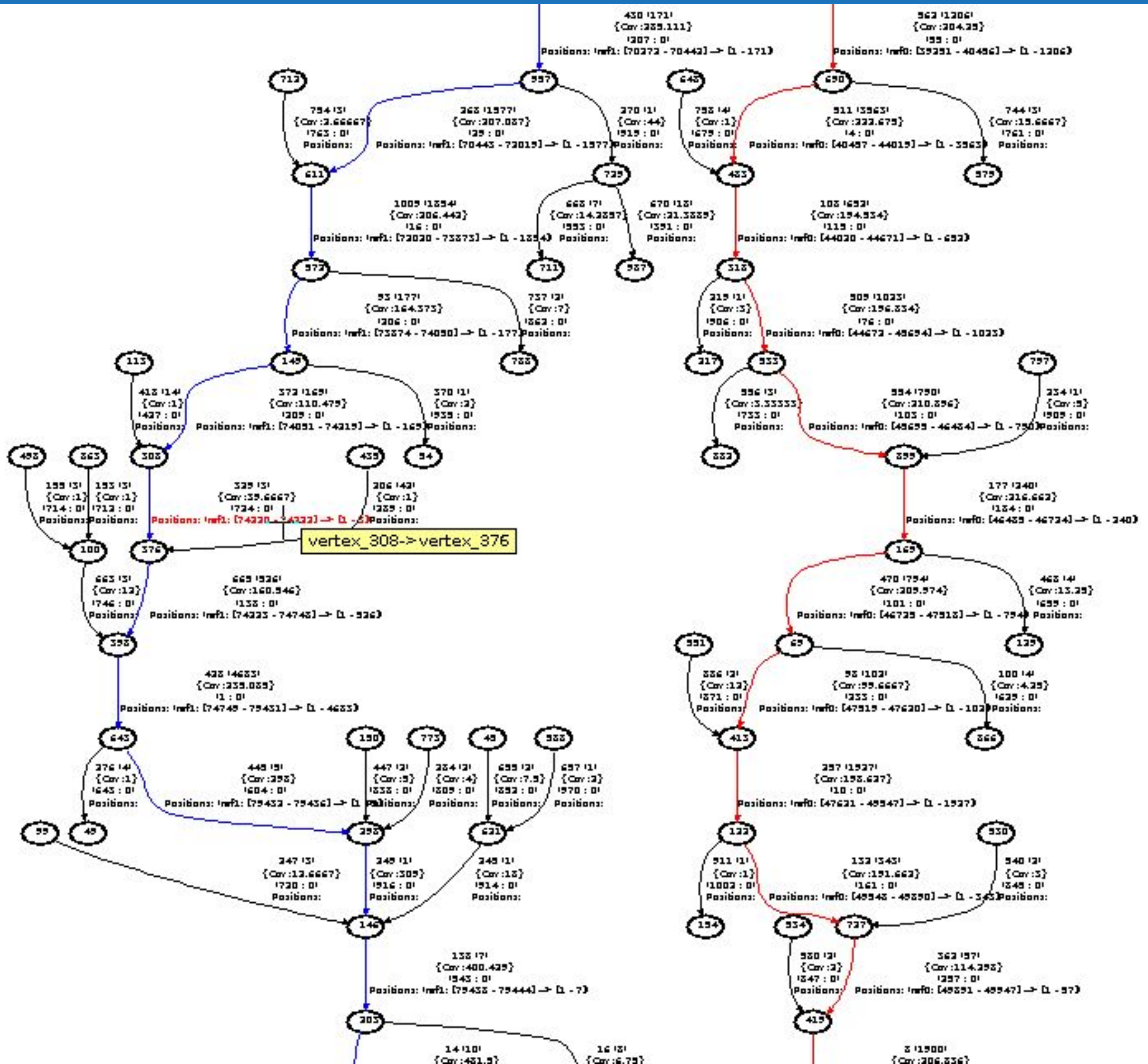


And what about bulges?

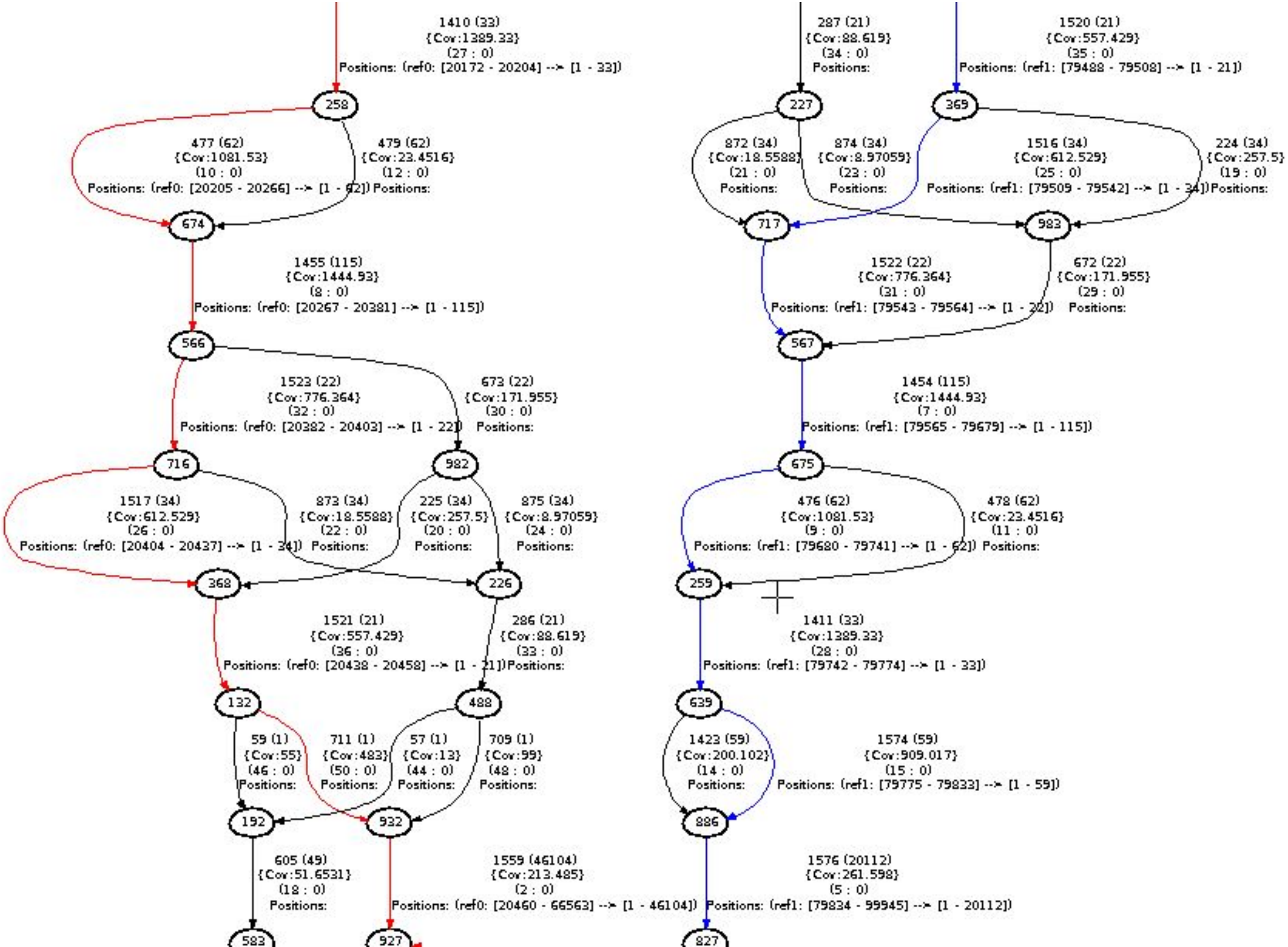
And what about bulges?

- Erroneous path has lower coverage that correct one
- Rather small length
- In case of similar coverage or bigger — more likely to be result of diploidy

Real life



Real life


1410 (33)
{Cov:1389.33}
(27 : 0)
Positions: (ref0: [20172 - 20204] --> [1 - 33])
258
477 (62)
{Cov:1081.53}
(10 : 0)
Positions: (ref0: [20205 - 20266] --> [1 - 62])
479 (62)
{Cov:23.4516}
(12 : 0)
Positions:
674
1455 (115)
{Cov:1444.93}
(8 : 0)
Positions: (ref0: [20267 - 20381] --> [1 - 115])
566
1523 (22)
{Cov:776.364}
(32 : 0)
Positions: (ref0: [20382 - 20403] --> [1 - 22])
673 (22)
{Cov:171.955}
(30 : 0)
Positions:
716
982
1517 (34)
{Cov:612.529}
(26 : 0)
Positions: (ref0: [20404 - 20437] --> [1 - 34])
873 (34)
{Cov:18.5588}
(22 : 0)
Positions:
225 (34)
{Cov:257.5}
(20 : 0)
Positions:
875 (34)
{Cov:8.97059}
(24 : 0)
Positions:
368
226
1521 (21)
{Cov:557.429}
(36 : 0)
Positions: (ref0: [20438 - 20458] --> [1 - 21])
286 (21)
{Cov:88.619}
(33 : 0)
Positions:
132
488
59 (1)
{Cov:55}
(46 : 0)
Positions:
711 (1)
{Cov:483}
(50 : 0)
Positions:
57 (1)
{Cov:13}
(44 : 0)
Positions:
709 (1)
{Cov:99}
(48 : 0)
Positions:
192
932
605 (49)
{Cov:51.6531}
(18 : 0)
Positions:
1559 (46104)
{Cov:213.485}
(2 : 0)
Positions: (ref0: [20460 - 66563] --> [1 - 46104])
583
927
287 (21)
{Cov:88.619}
(34 : 0)
Positions:
1520 (21)
{Cov:557.429}
(35 : 0)
Positions: (ref1: [79488 - 79508] --> [1 - 21])
227
369
872 (34)
{Cov:18.5588}
(21 : 0)
Positions:
874 (34)
{Cov:8.97059}
(23 : 0)
Positions:
1516 (34)
{Cov:612.529}
(25 : 0)
Positions: (ref1: [79509 - 79542] --> [1 - 34])
224 (34)
{Cov:257.5}
(19 : 0)
Positions:
717
983
1522 (22)
{Cov:776.364}
(31 : 0)
Positions: (ref1: [79543 - 79564] --> [1 - 22])
672 (22)
{Cov:171.955}
(29 : 0)
Positions:
567
1454 (115)
{Cov:1444.93}
(7 : 0)
Positions: (ref1: [79565 - 79679] --> [1 - 115])
675
476 (62)
{Cov:1081.53}
(9 : 0)
Positions: (ref1: [79680 - 79741] --> [1 - 62])
478 (62)
{Cov:23.4516}
(11 : 0)
Positions:
259
1411 (33)
{Cov:1389.33}
(28 : 0)
Positions: (ref1: [79742 - 79774] --> [1 - 33])
639
1423 (59)
{Cov:200.102}
(14 : 0)
Positions:
1574 (59)
{Cov:909.017}
(15 : 0)
Positions: (ref1: [79775 - 79833] --> [1 - 59])
886
1576 (20112)
{Cov:261.598}
(5 : 0)
Positions: (ref1: [79834 - 99945] --> [1 - 20112])
827

Velvet assembler simplification

- Tip clipping
- Bulge removal
- Removing erroneous connections

Velvet tip clipping

- Remove only if shorter than 2 * k
- Coverage is lower than of any alternative paths
- Iteratively process over the graph until no tips are left

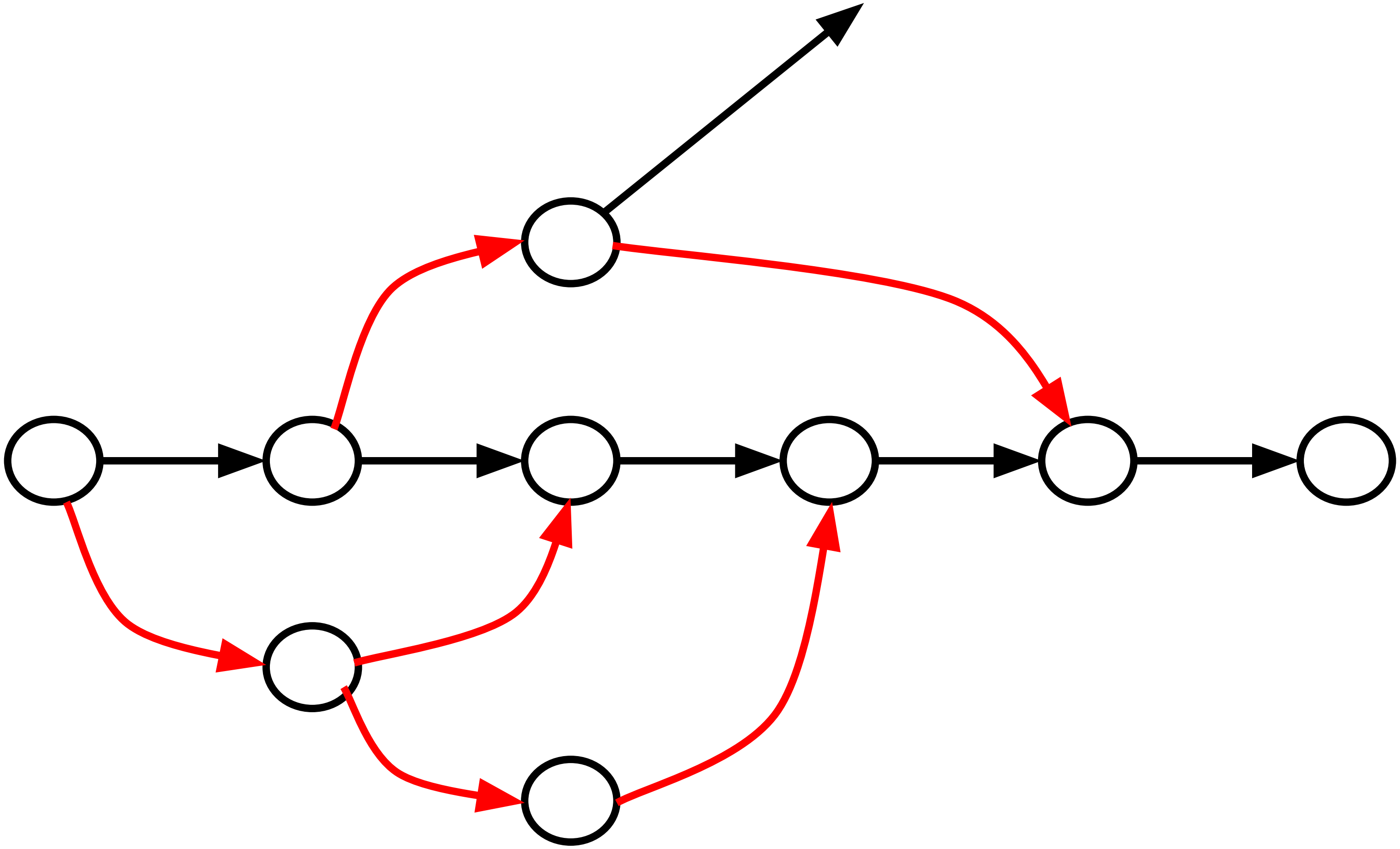
Velvet “tour bus” algorithm

- Distance between vertices A and B is

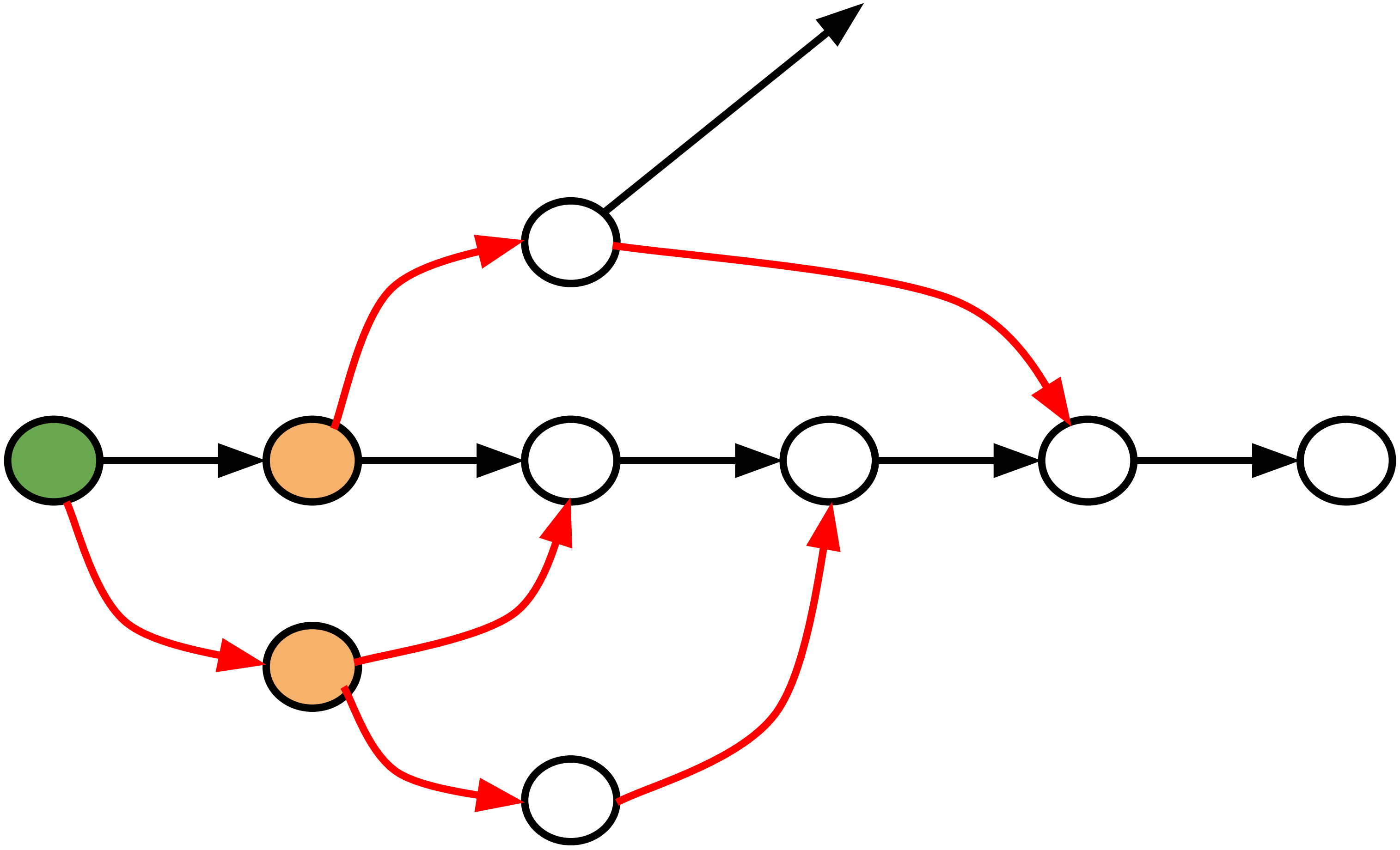
  $D(A, B) = \mathit{length}(E_{AB}) / \mathit{coverage}(E_{AB})$

  - Allows to go through reliable paths faster
- Start BFS from arbitrary node
- As soon as we came to already visited vertex
  - Align to alternative paths
  - Project low-covered path onto the main one

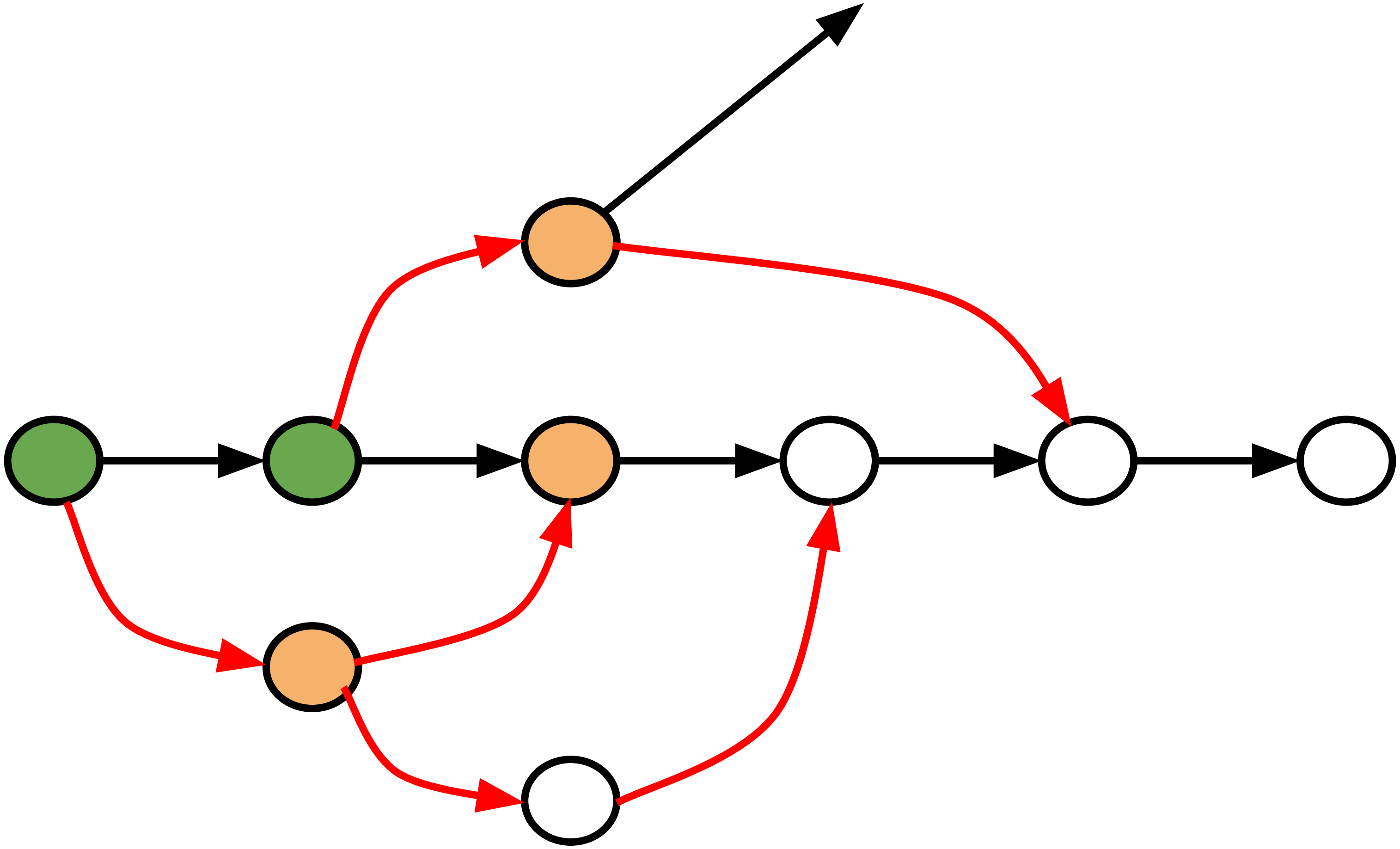
Velvet “tour bus” algorithm



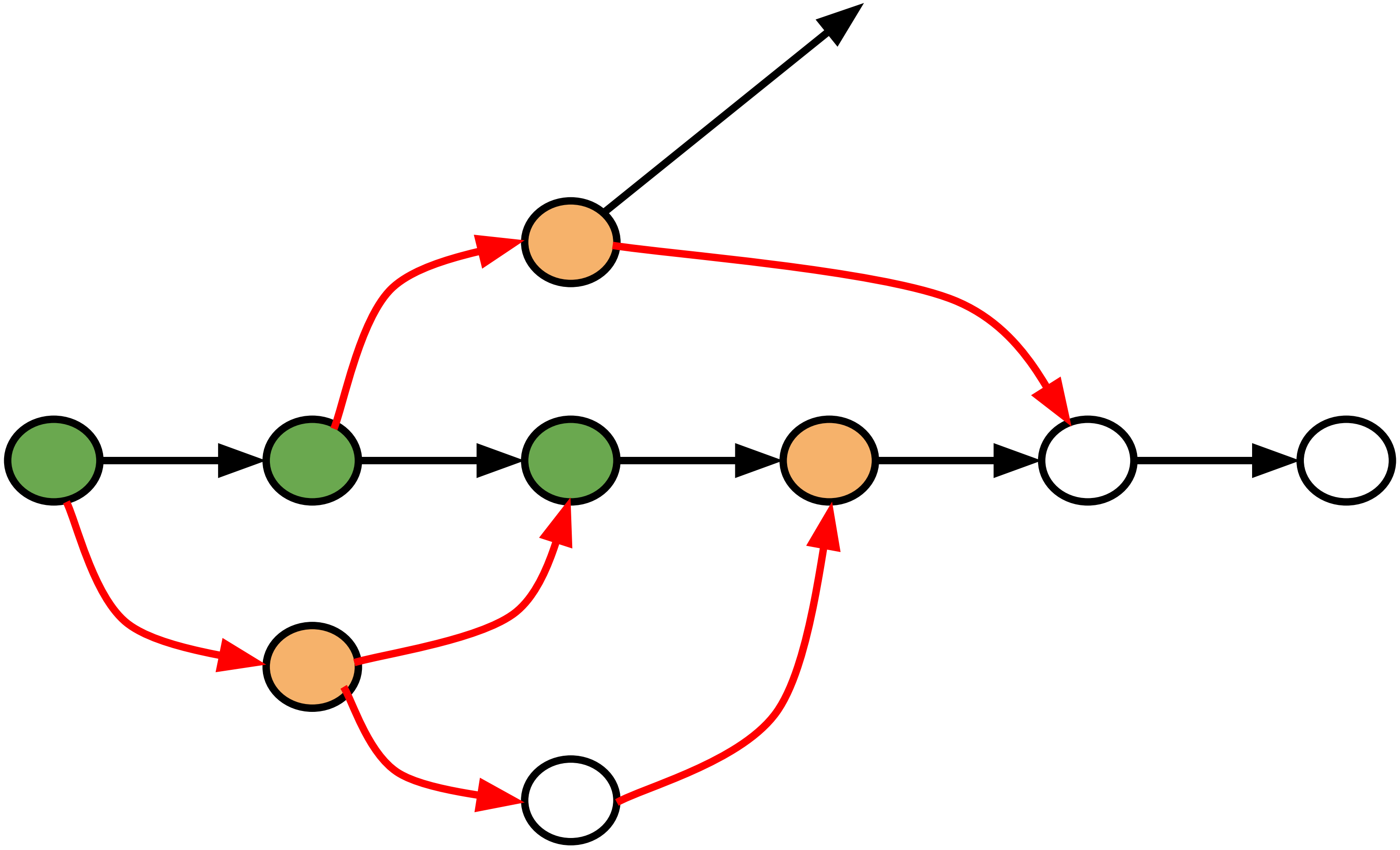
Velvet “tour bus” algorithm



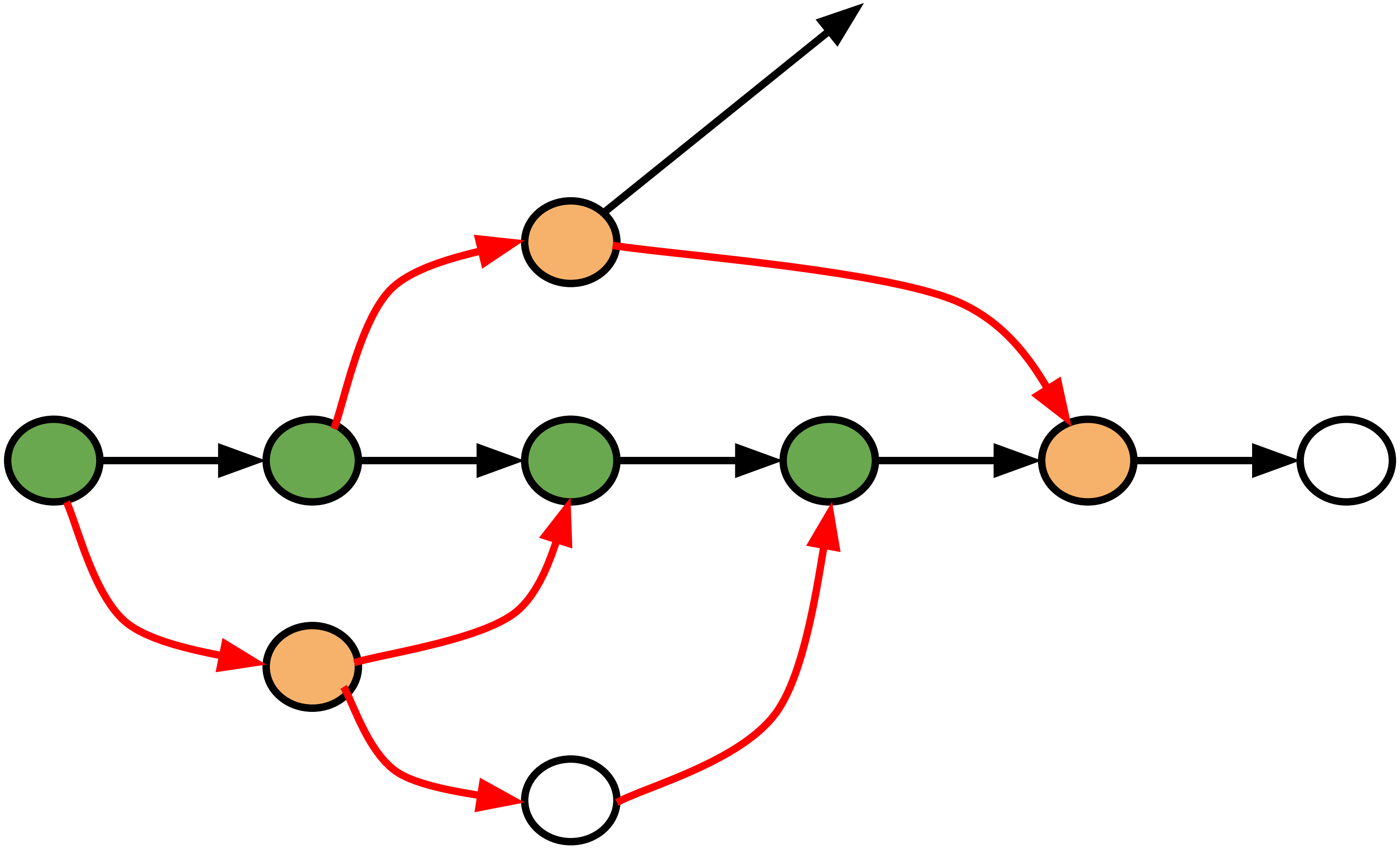
Velvet “tour bus” algorithm



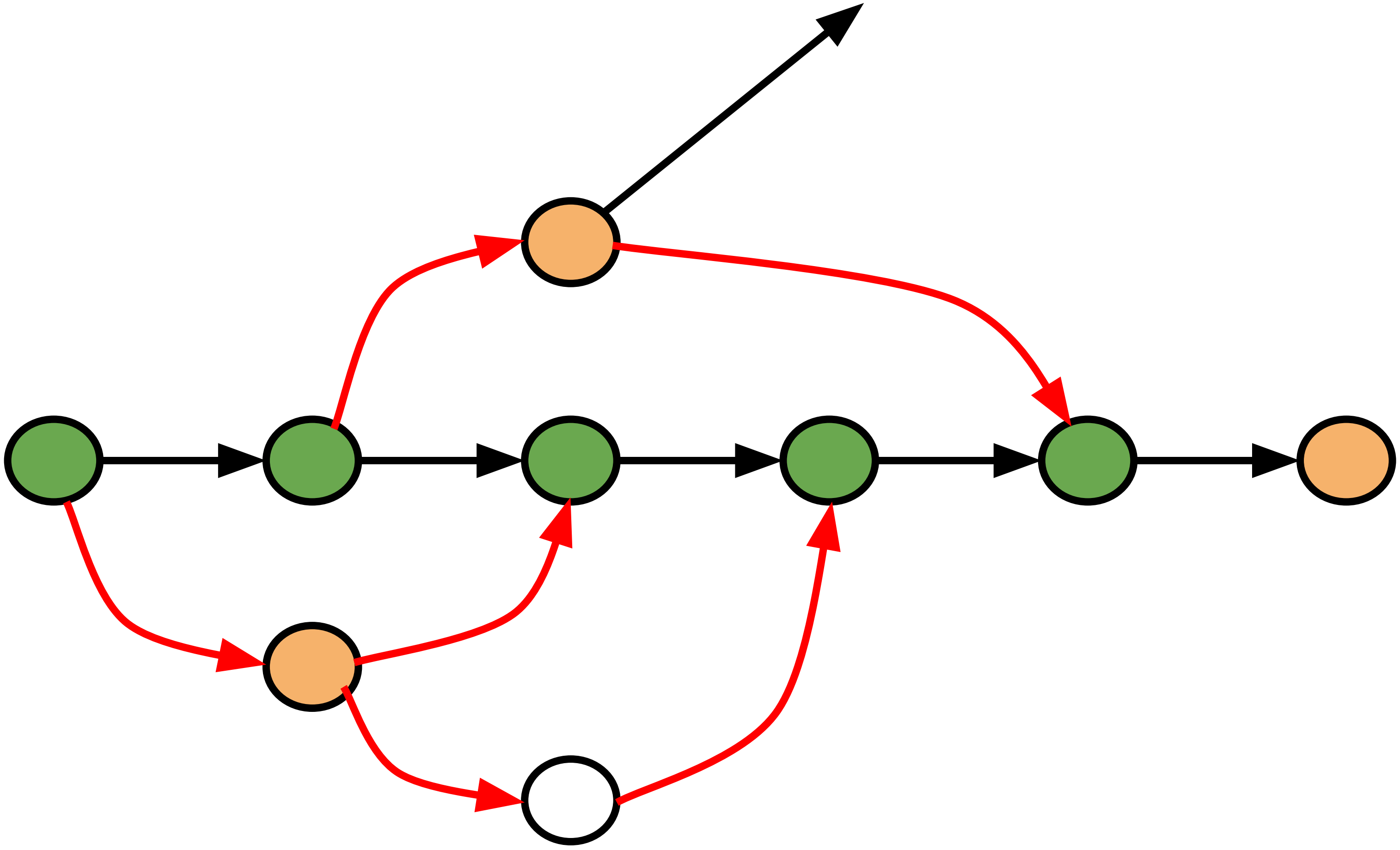
Velvet “tour bus” algorithm



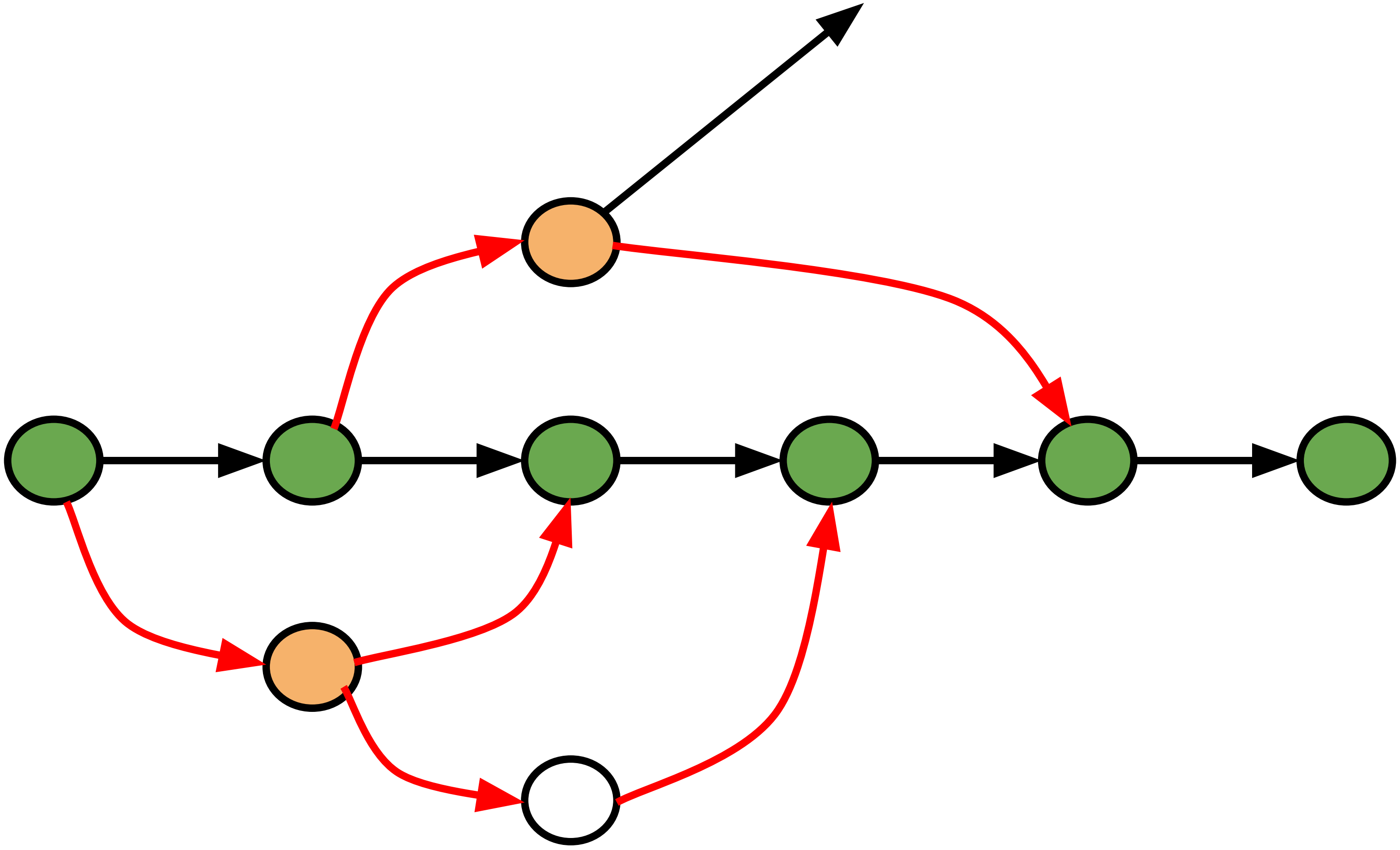
Velvet “tour bus” algorithm



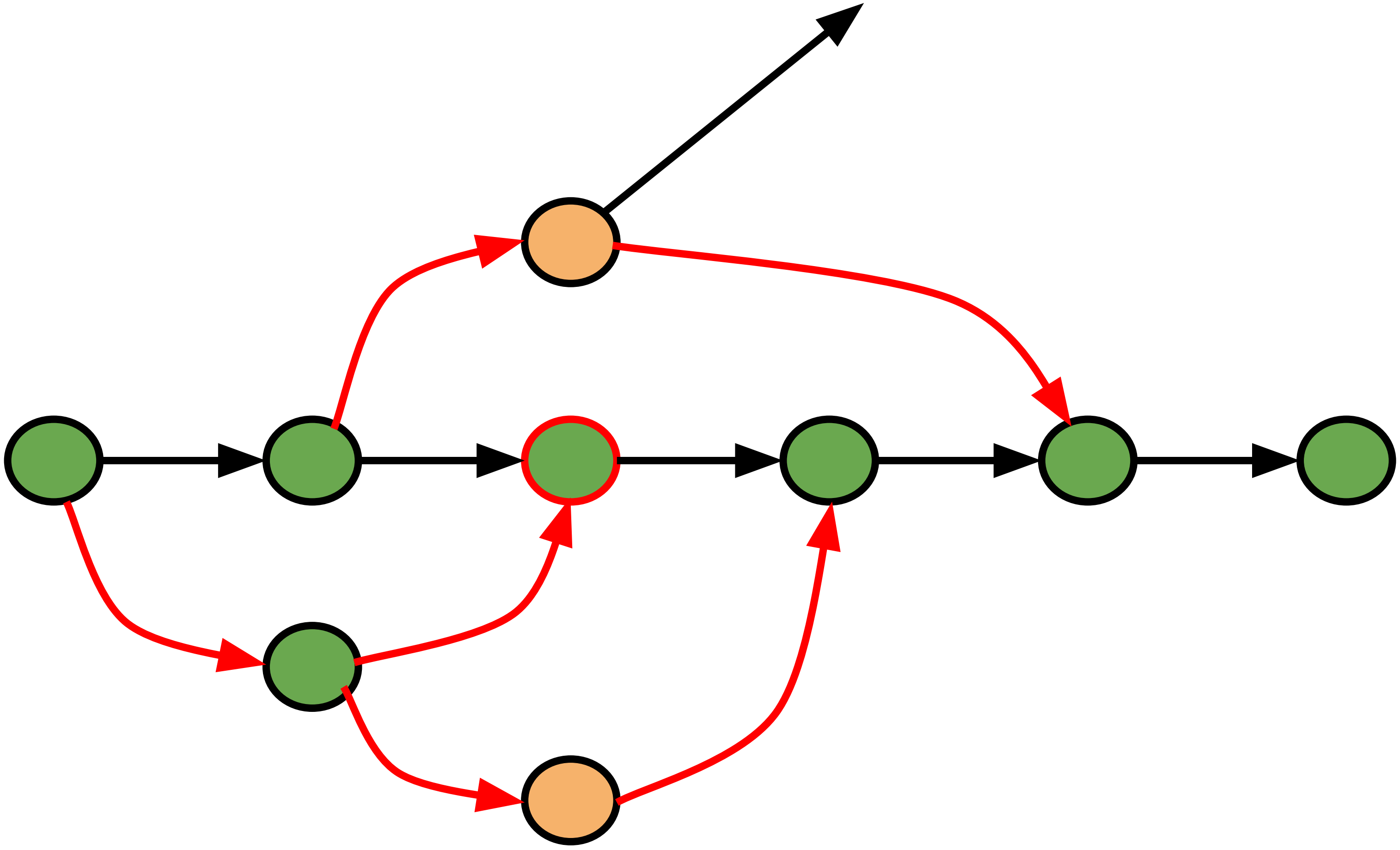
Velvet “tour bus” algorithm



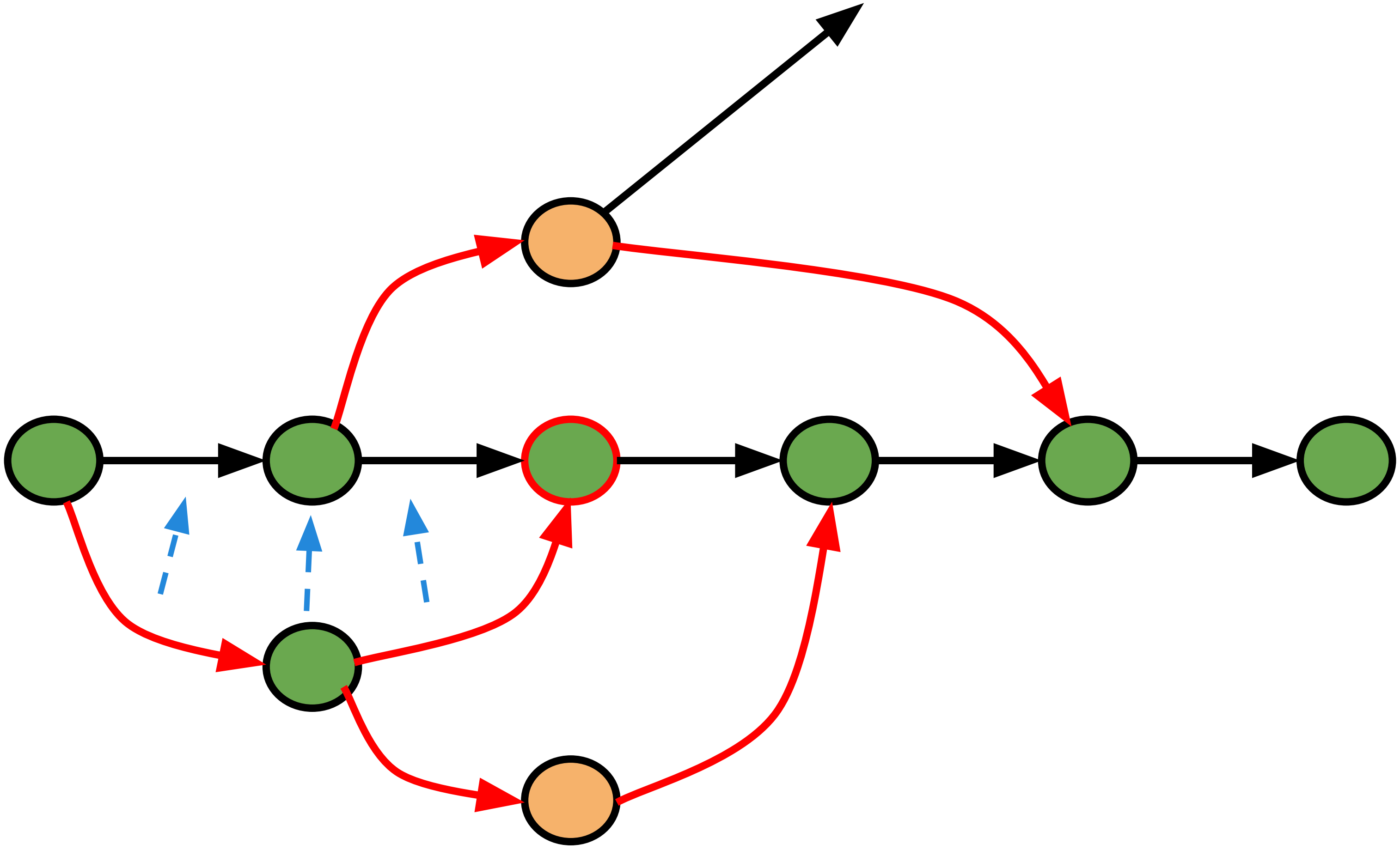
Velvet “tour bus” algorithm



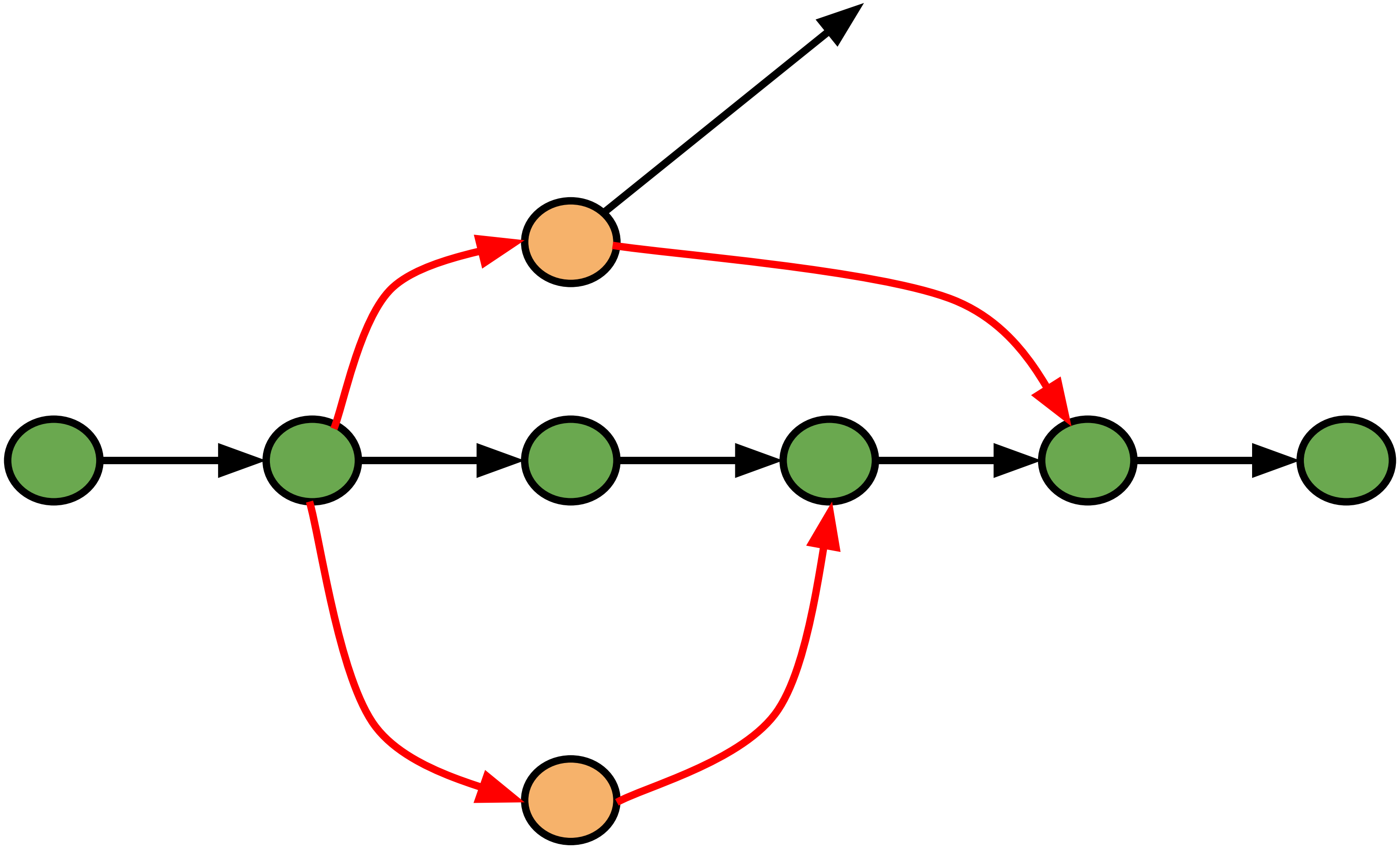
Velvet “tour bus” algorithm



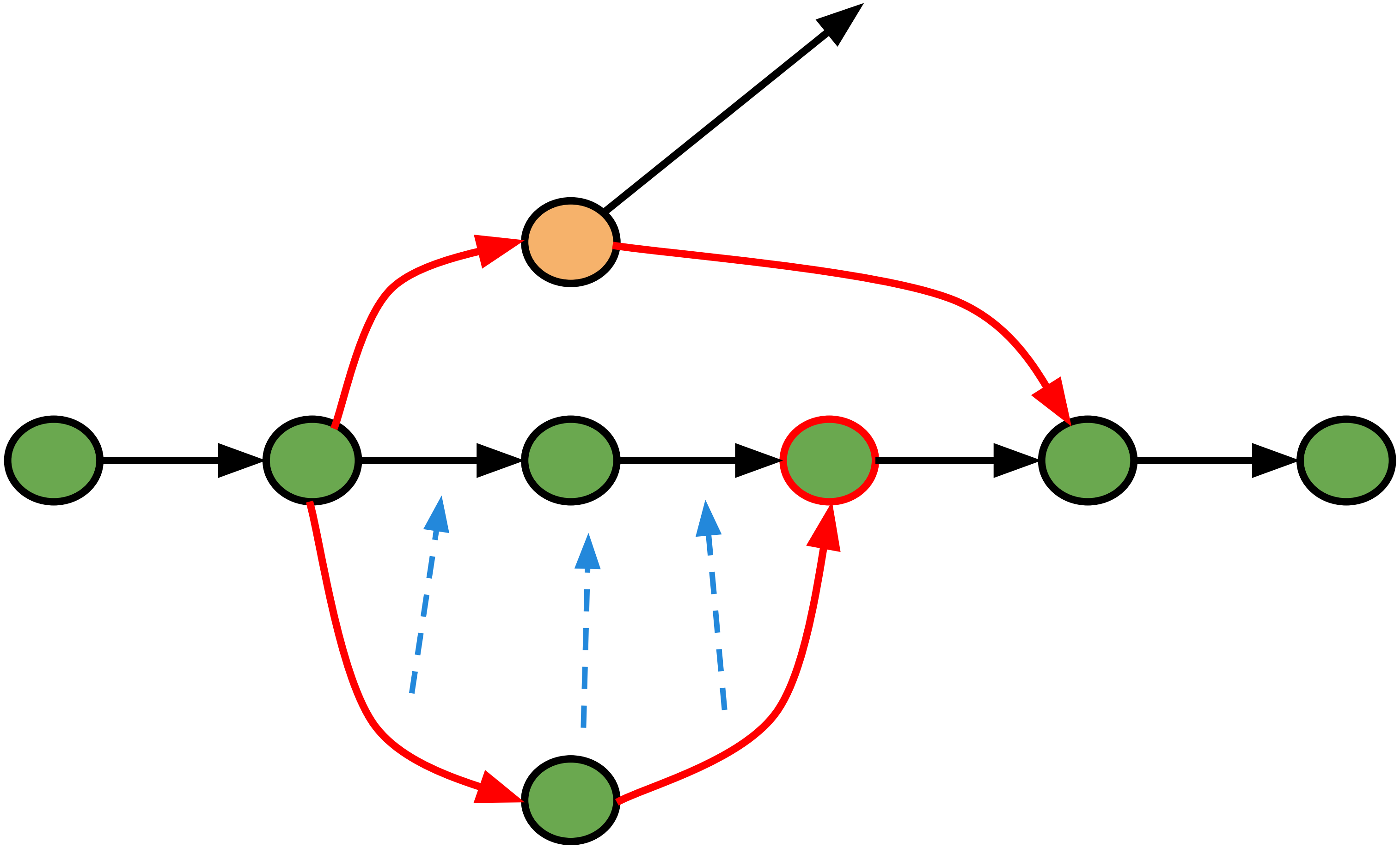
Velvet “tour bus” algorithm



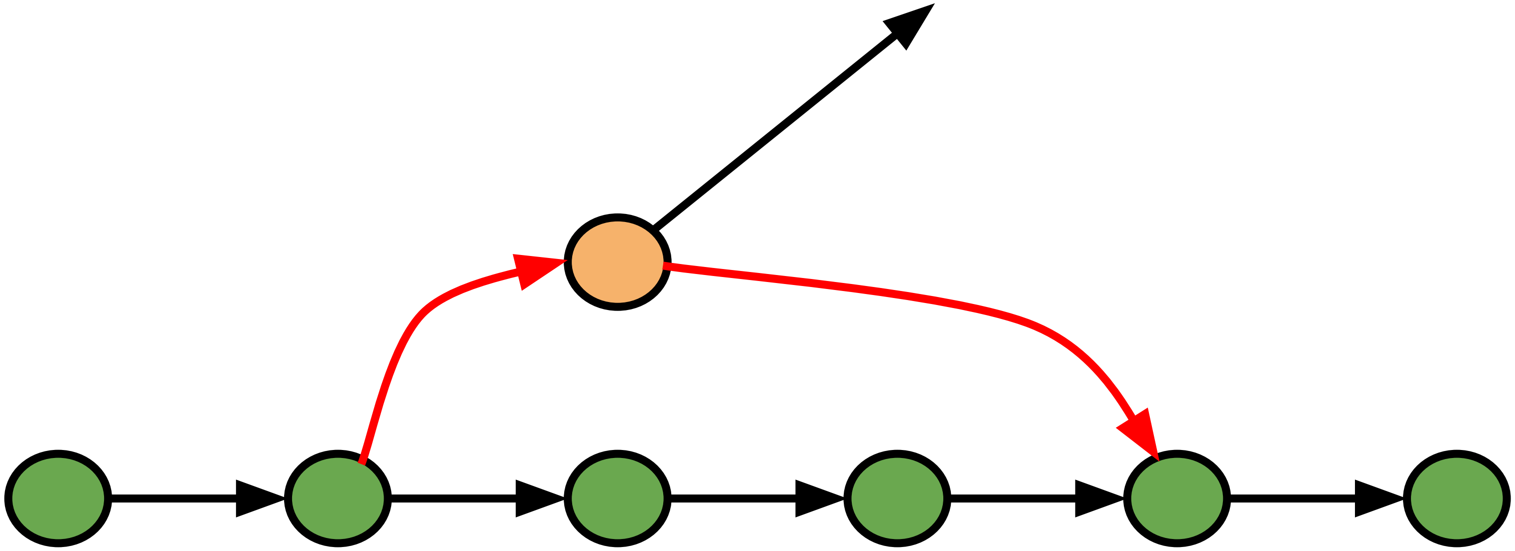
Velvet “tour bus” algorithm



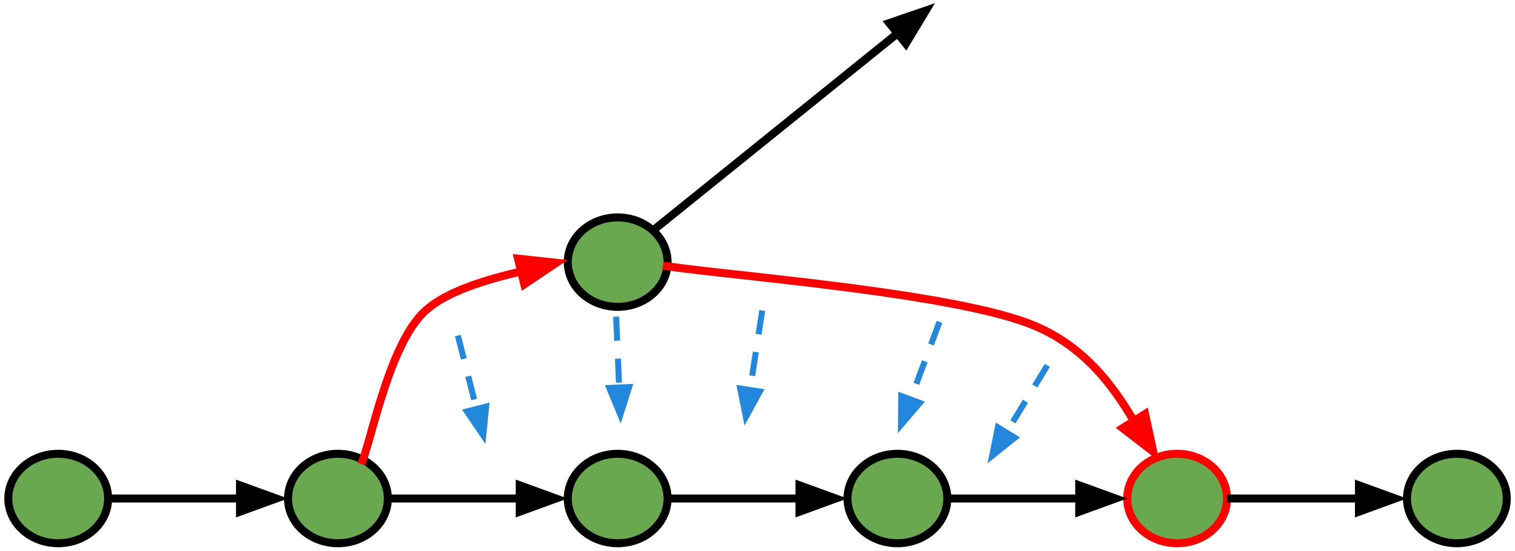
Velvet “tour bus” algorithm



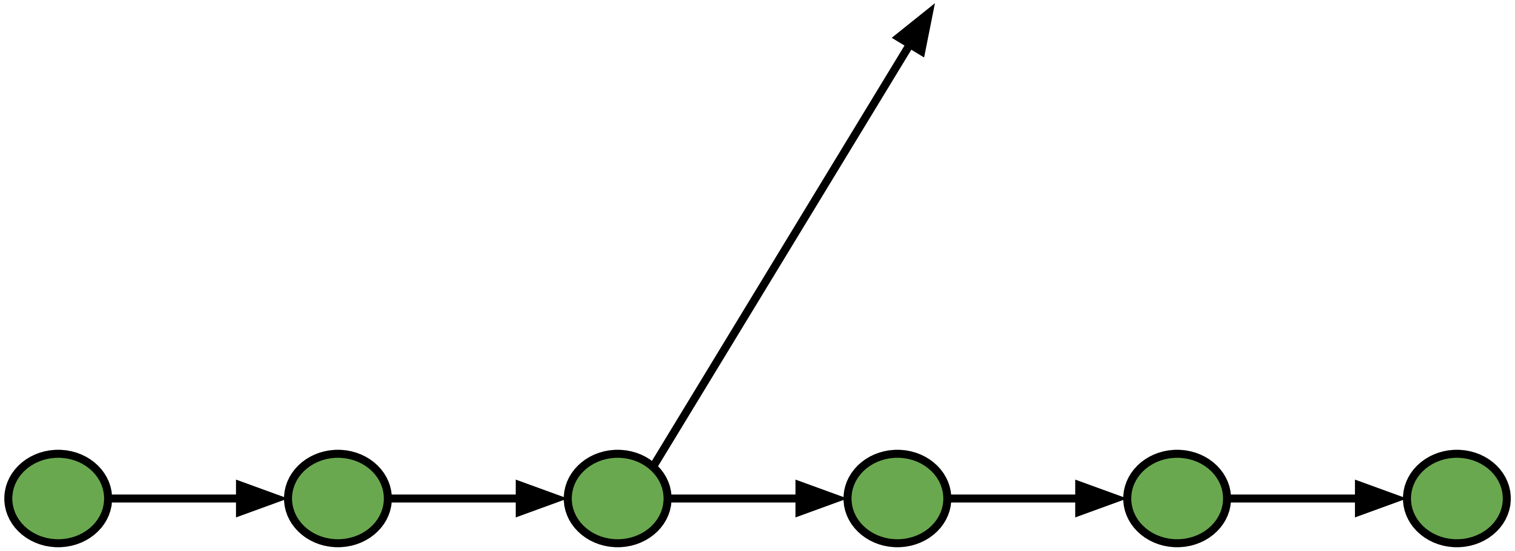
Velvet “tour bus” algorithm



Velvet “tour bus” algorithm



Velvet “tour bus” algorithm



Velvet erroneous connection remover

- Erroneous connections don’t have any recognized topological structure
- Have low coverage
- Removed using simple coverage cutoff

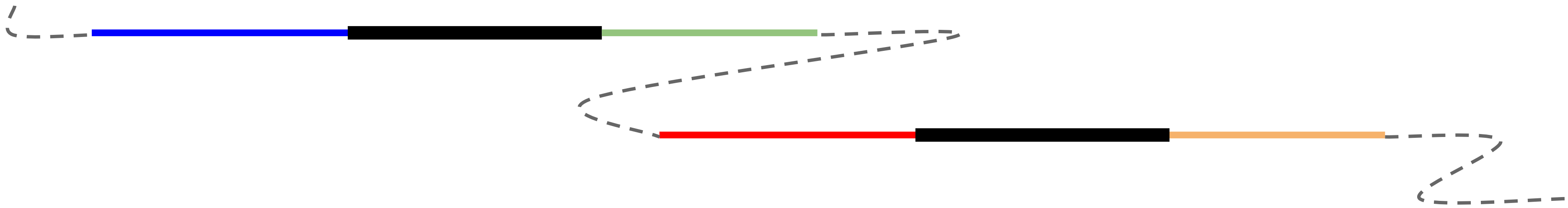
Homework

Implement de Bruijn graph

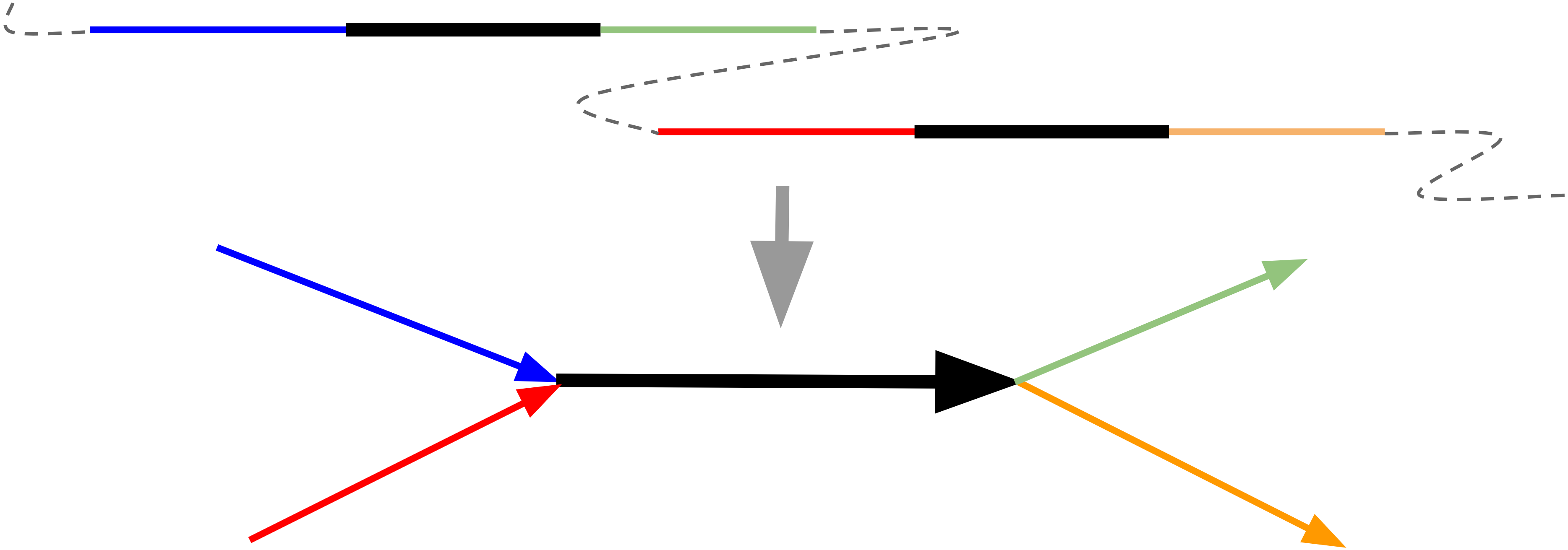
- Construction from FASTA/FASTQ
- Condensation
- Output to FASTA/DOT/…
- Simplification

Resolving repeats

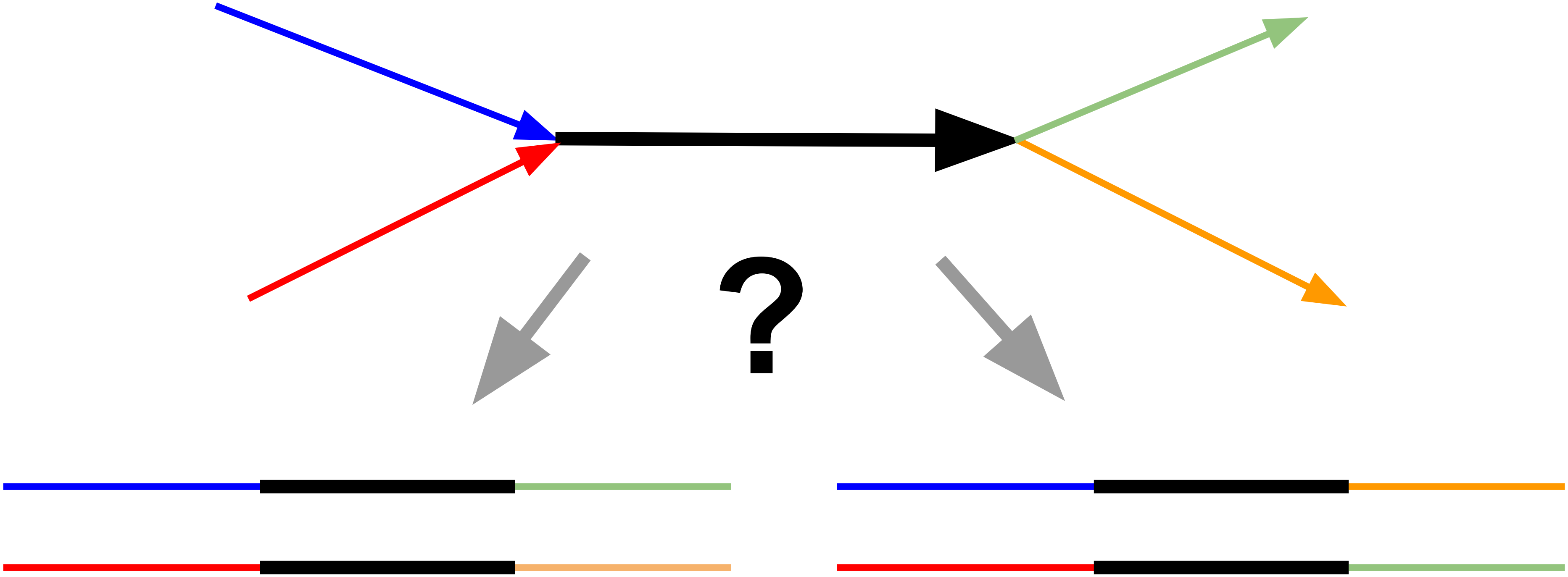
Resolving repeats



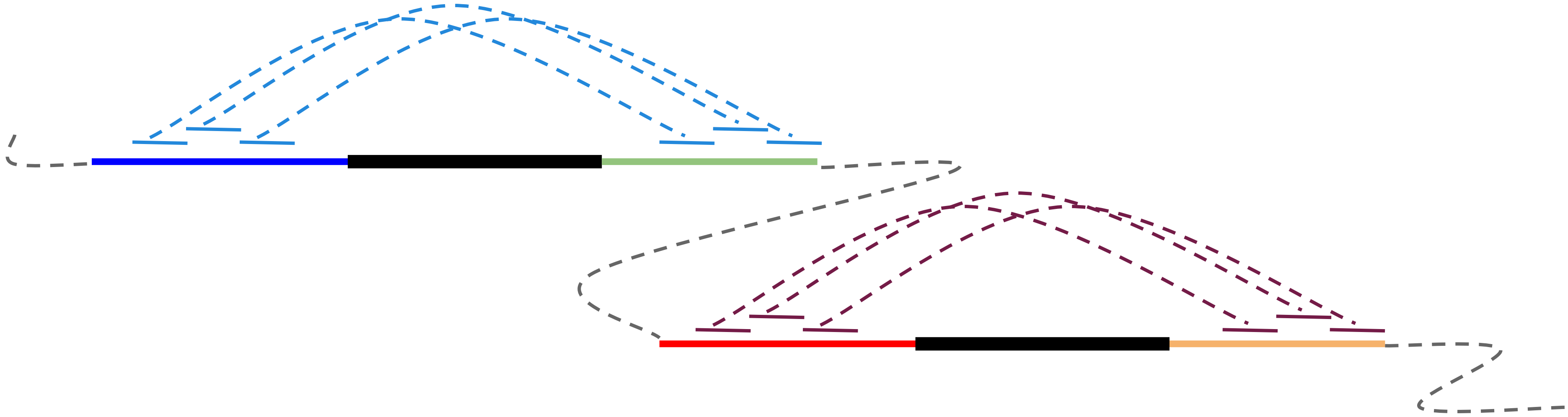
Resolving repeats



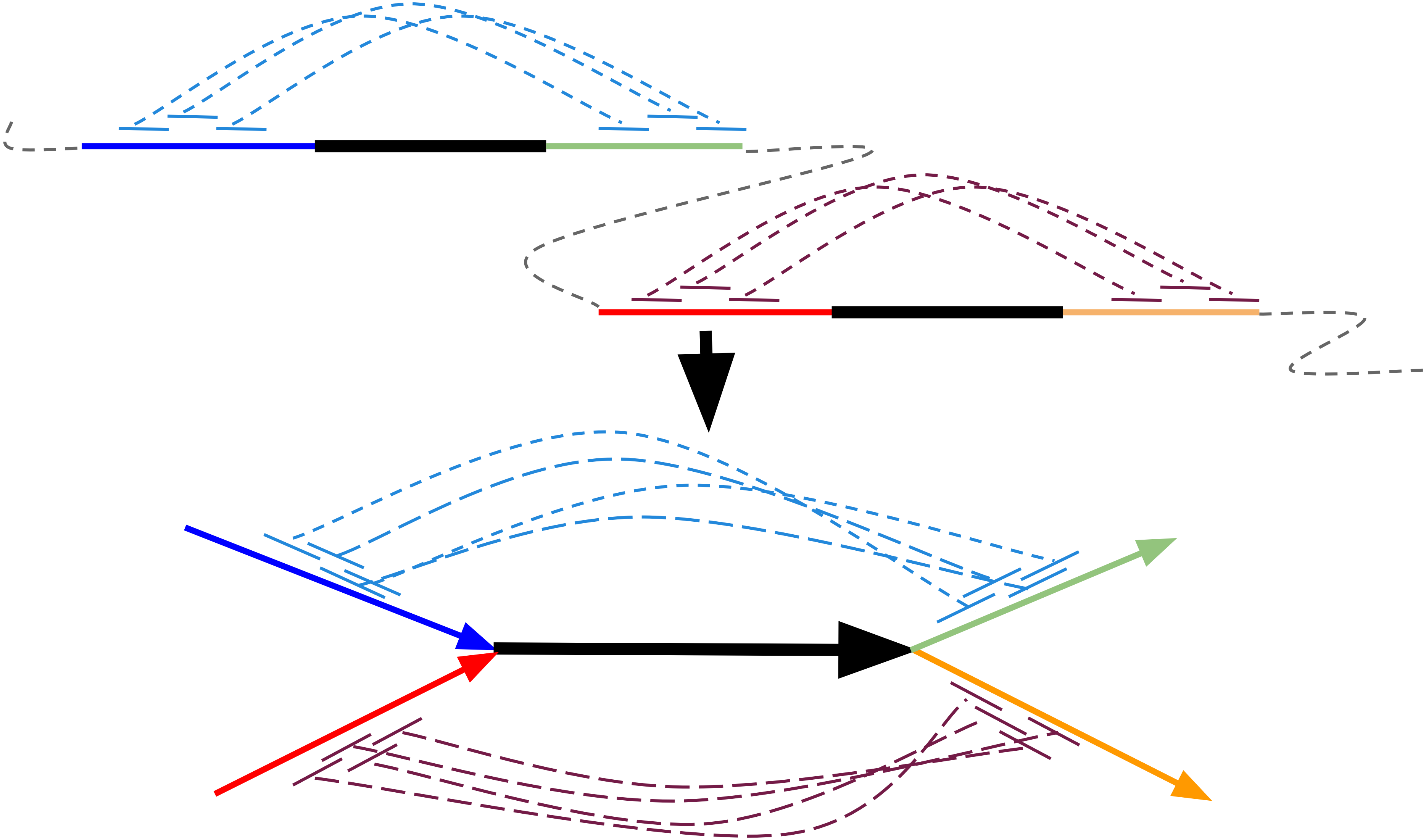
Resolving repeats



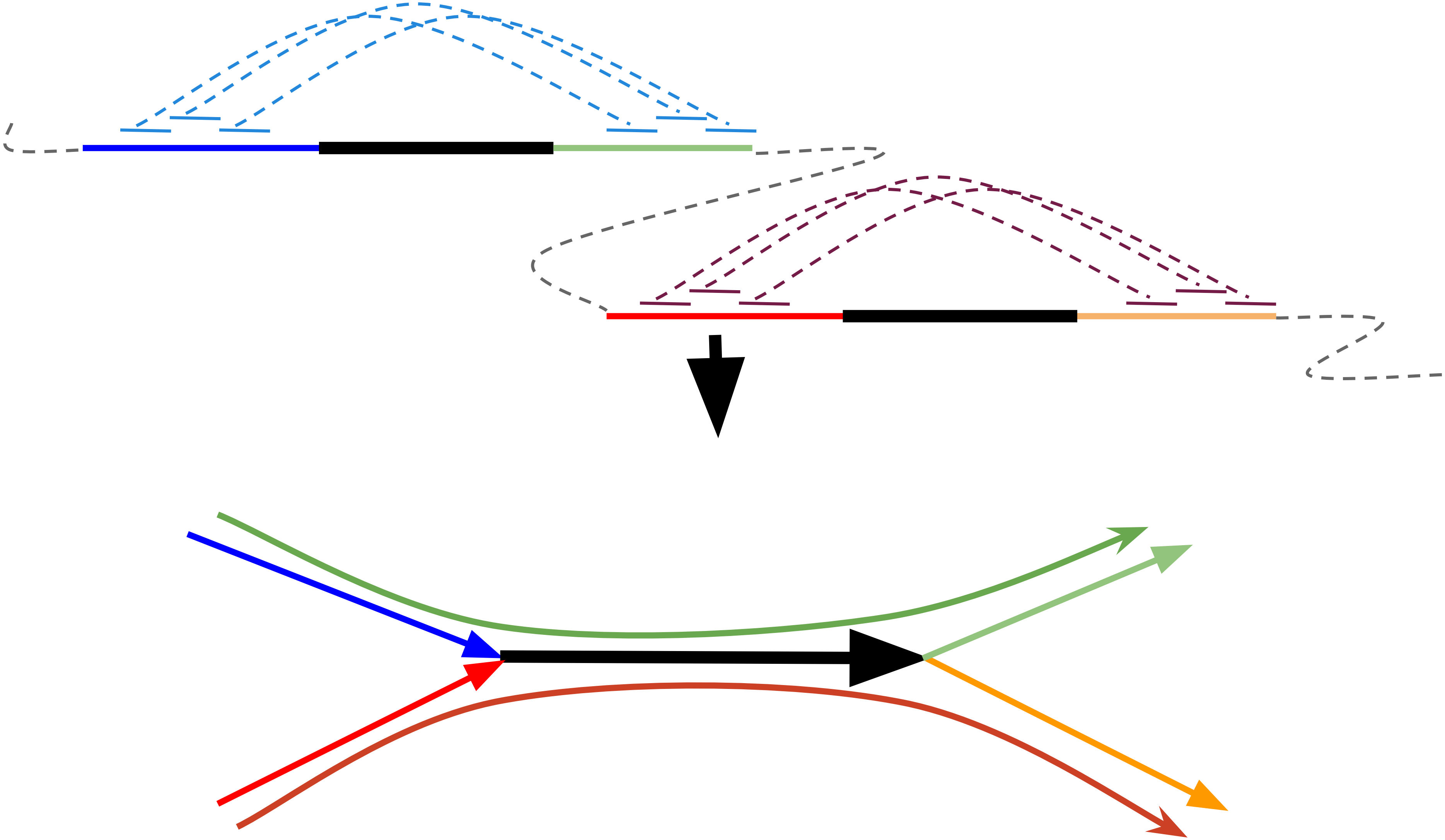
Resolving repeats



Resolving repeats

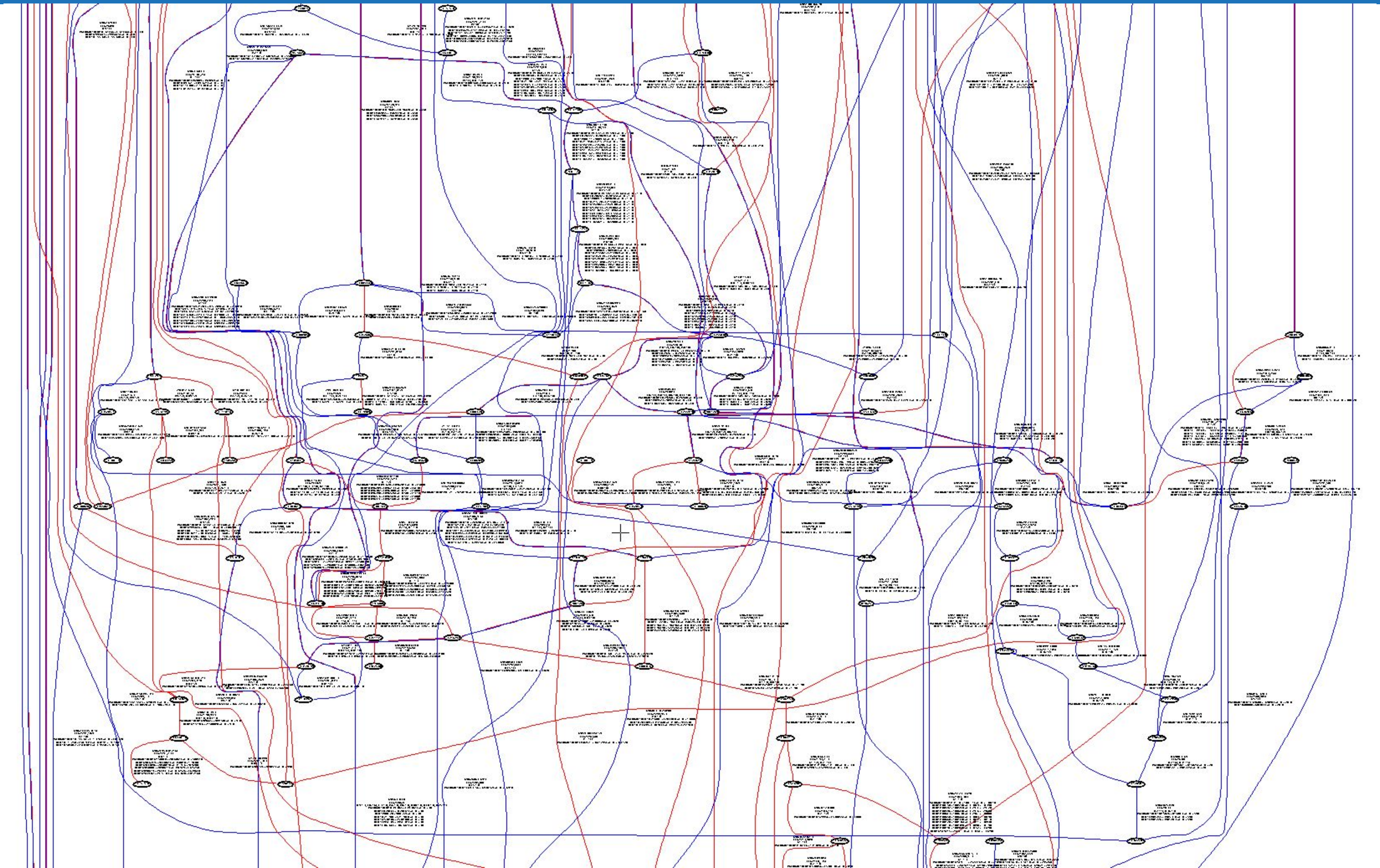


Resolving repeats



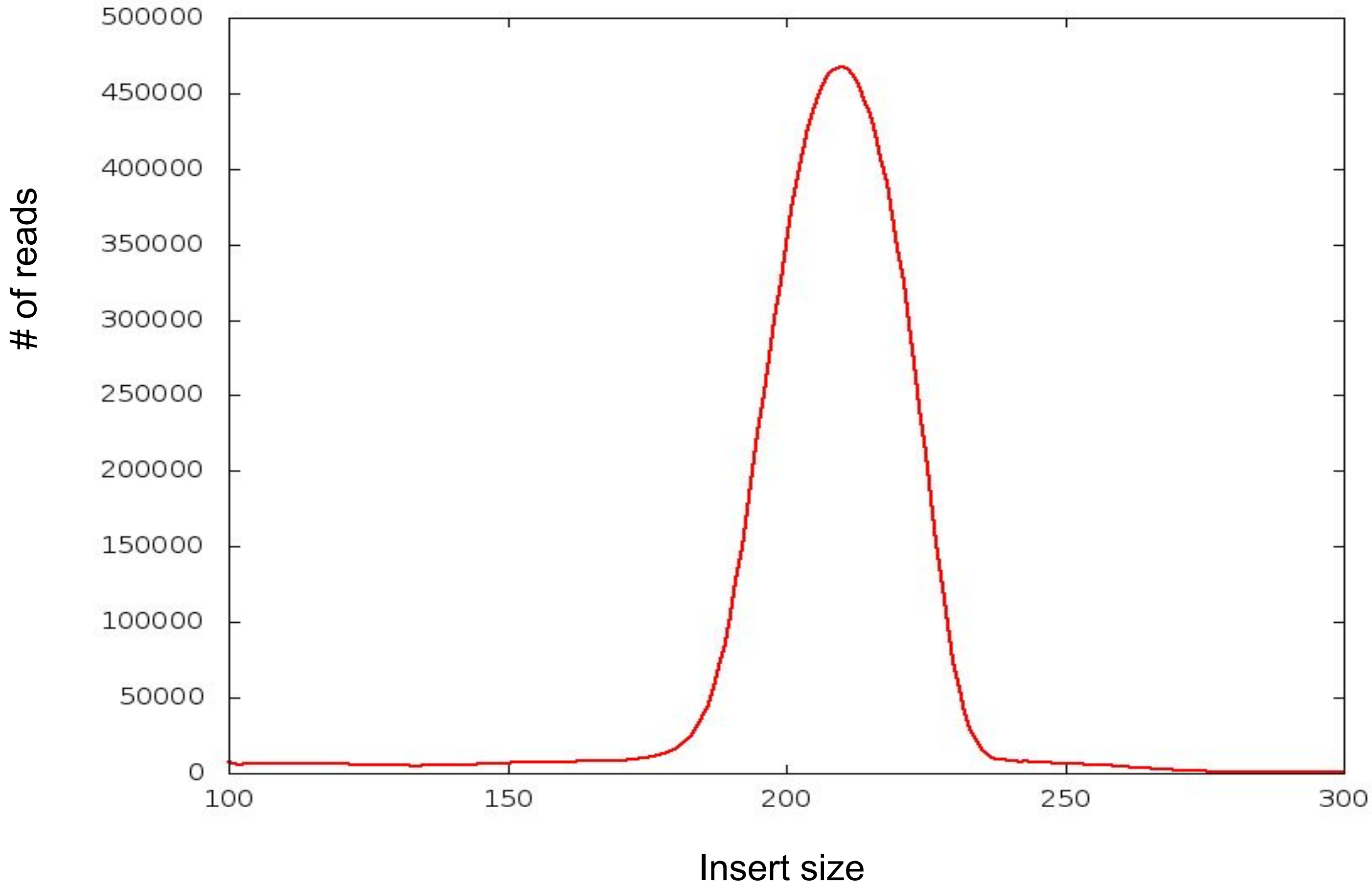
Real life

Part of *E.coli* genome, K = 99

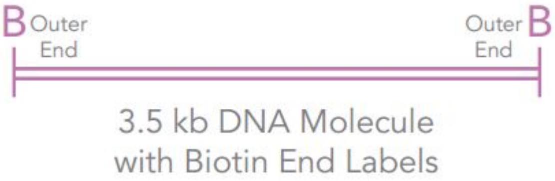


Insert size distribution

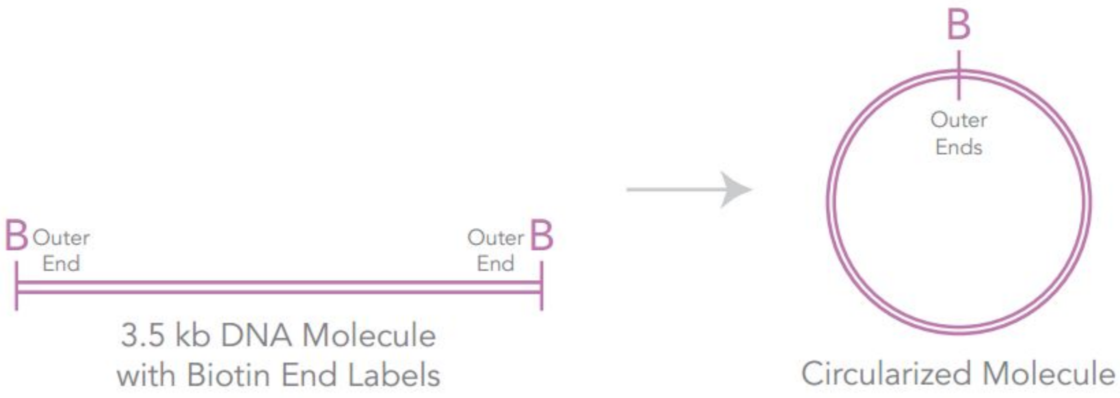
Paired-end reads



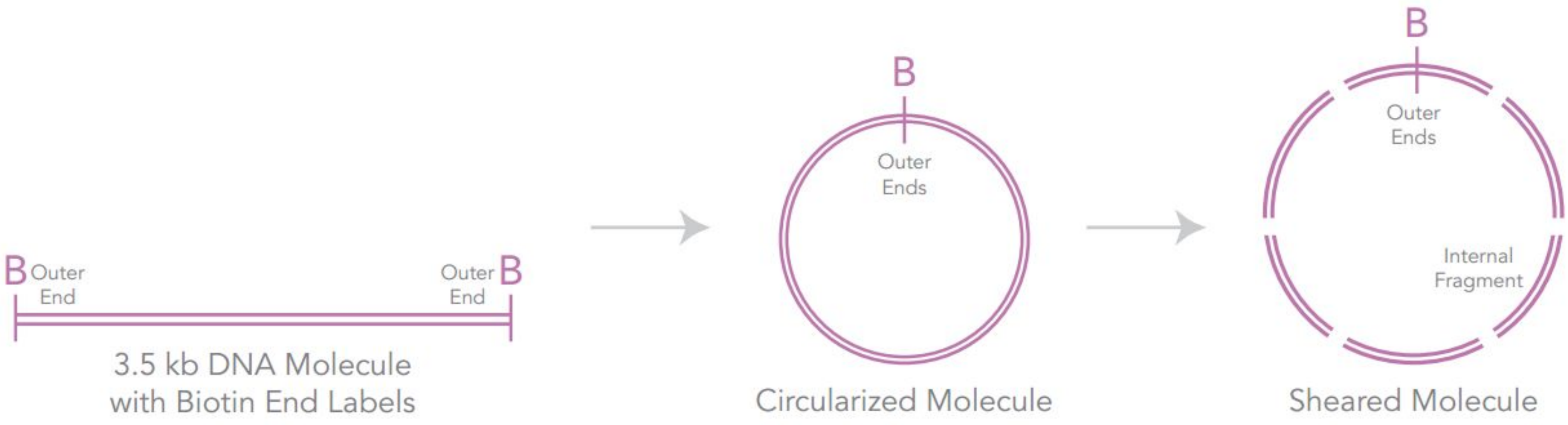
Mate-pairs


B
Outer End
Outer End
B
3.5 kb DNA Molecule with Biotin End Labels

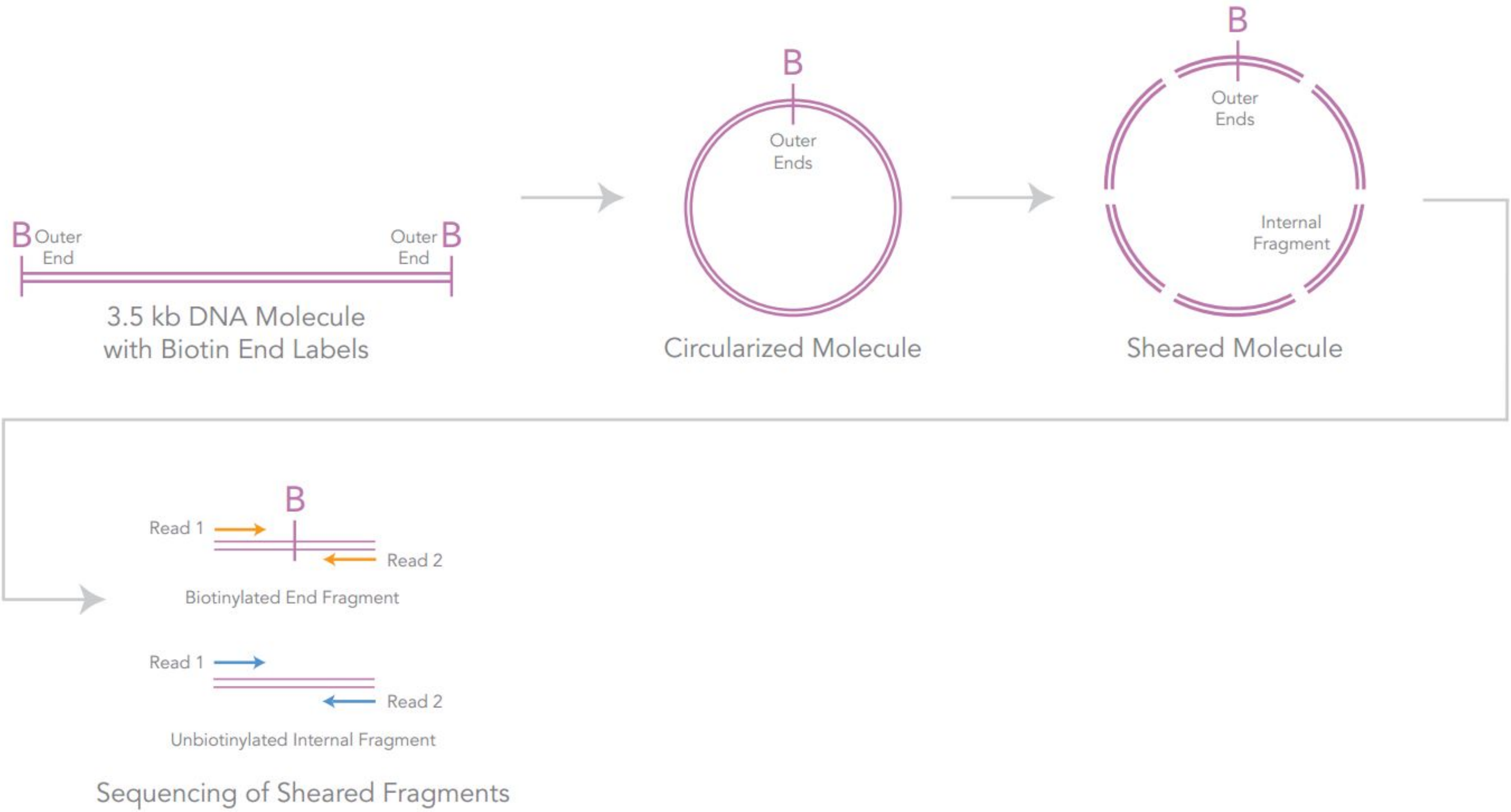
Mate-pairs


B
Outer
End
Outer
End
B
3.5 kb DNA Molecule
with Biotin End Labels
B
Outer
Ends
Circularized Molecule

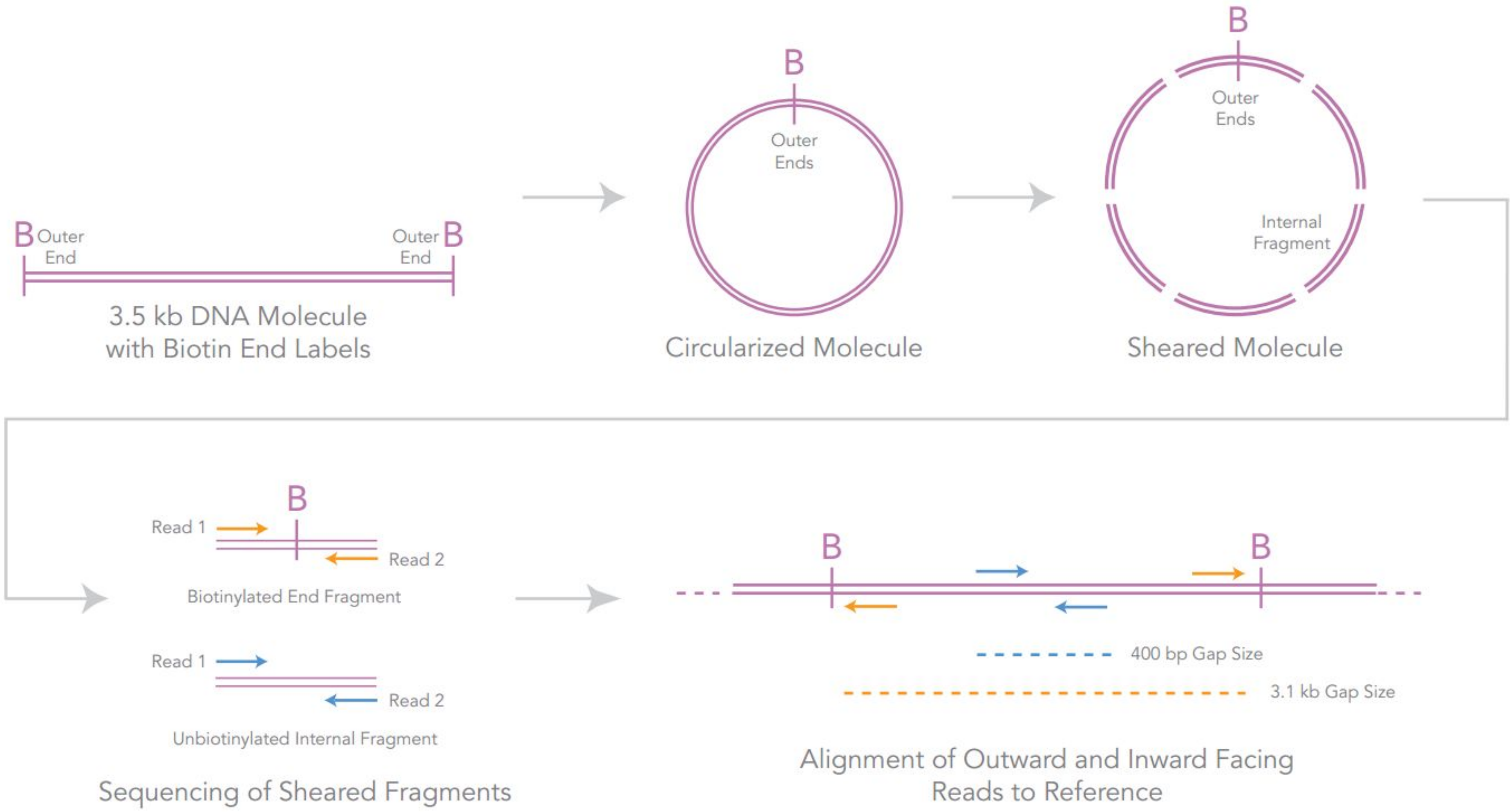
Mate-pairs


B
Outer End
Outer End
B
3.5 kb DNA Molecule with Biotin End Labels
B
Outer Ends
Circularized Molecule
B
Outer Ends
Internal Fragment
Sheared Molecule

Mate-pairs

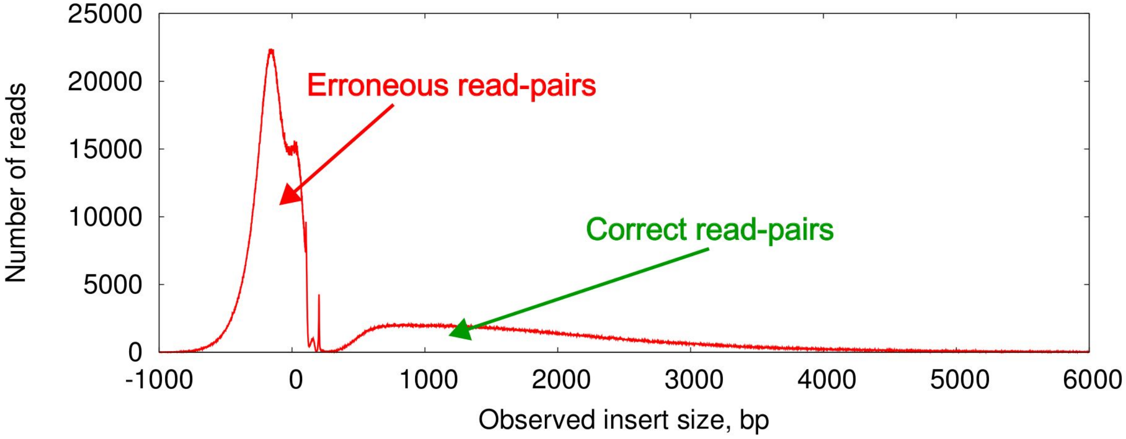

B
Outer End
Outer End
B
3.5 kb DNA Molecule with Biotin End Labels
B
Outer Ends
Circularized Molecule
B
Outer Ends
Internal Fragment
Sheared Molecule
B
Read 1
Read 2
Biotinylated End Fragment
Read 1
Read 2
Unbiotinylated Internal Fragment
Sequencing of Sheared Fragments

Mate-pairs


B
Outer End
Outer End
B
3.5 kb DNA Molecule with Biotin End Labels
B
Outer Ends
Circularized Molecule
B
Outer Ends
Internal Fragment
Sheared Molecule
B
Read 1
Read 2
Biotinylated End Fragment
Read 1
Read 2
Unbiotinylated Internal Fragment
Sequencing of Sheared Fragments
B
B
400 bp Gap Size
3.1 kb Gap Size
Alignment of Outward and Inward Facing Reads to Reference

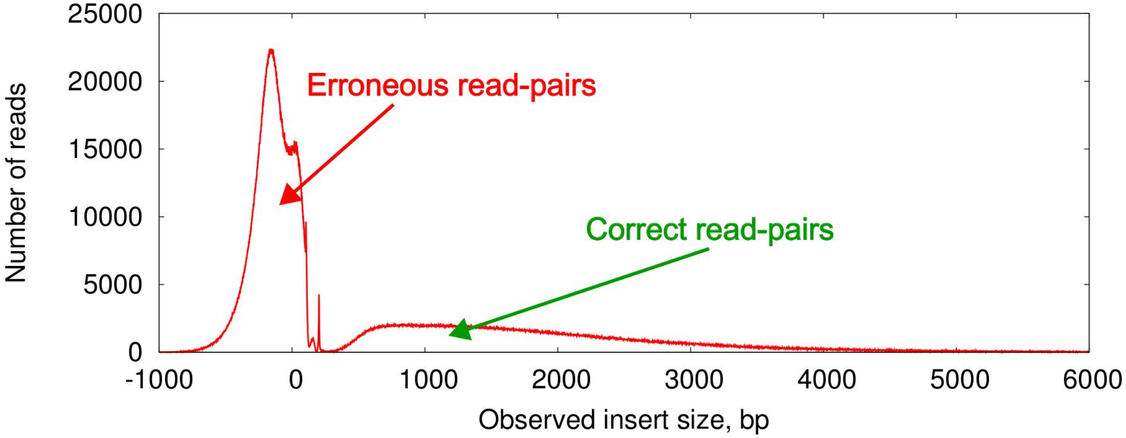
Mate-pairs

Conventional mate-pairs:

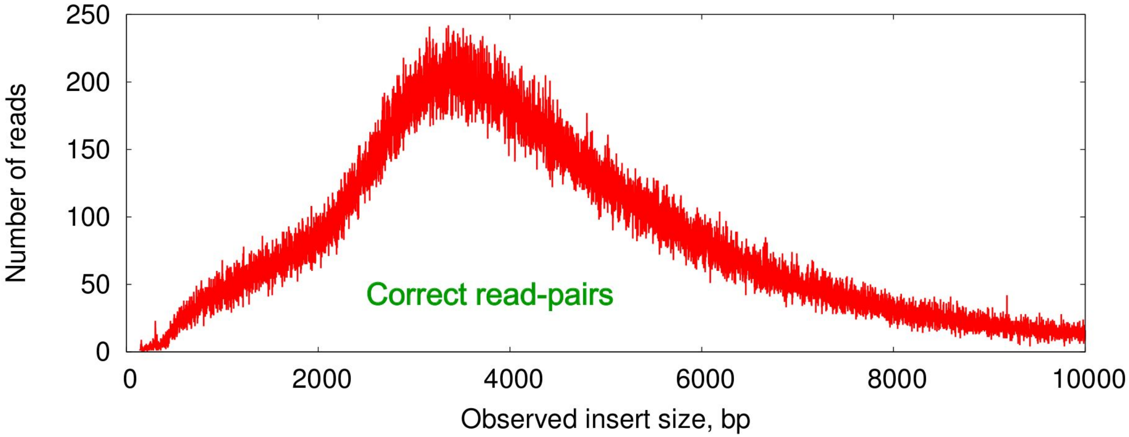


Mate-pairs

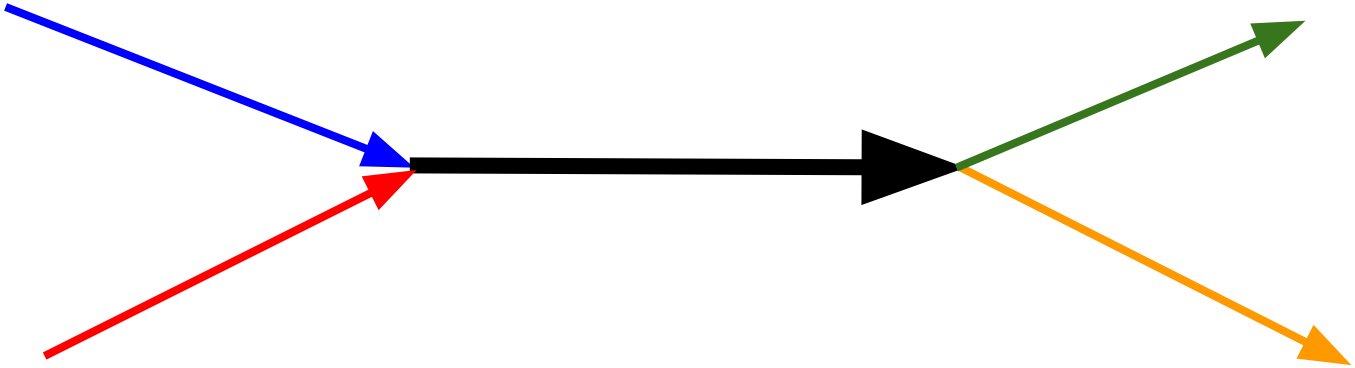
Conventional mate-pairs:



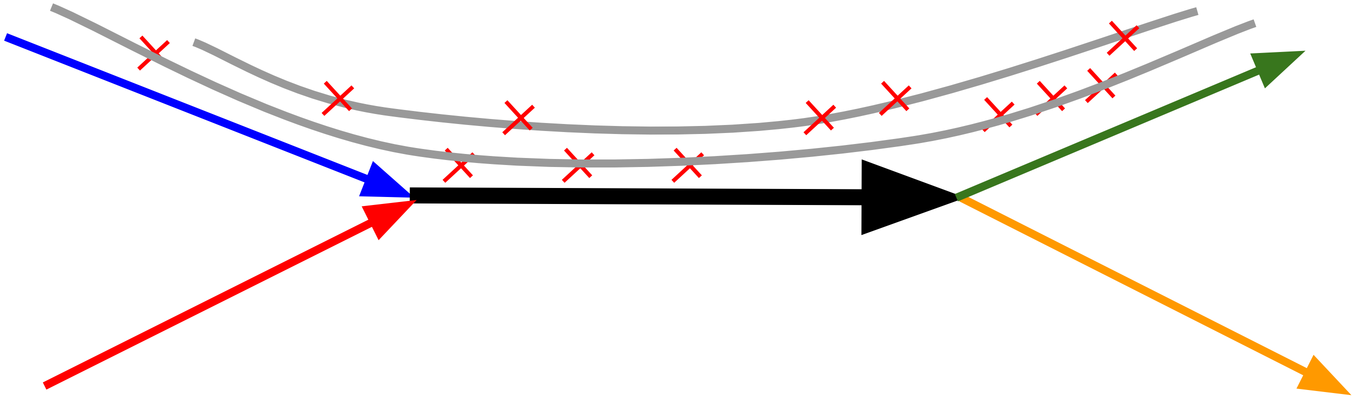
Illumina Nextera mate-pairs:



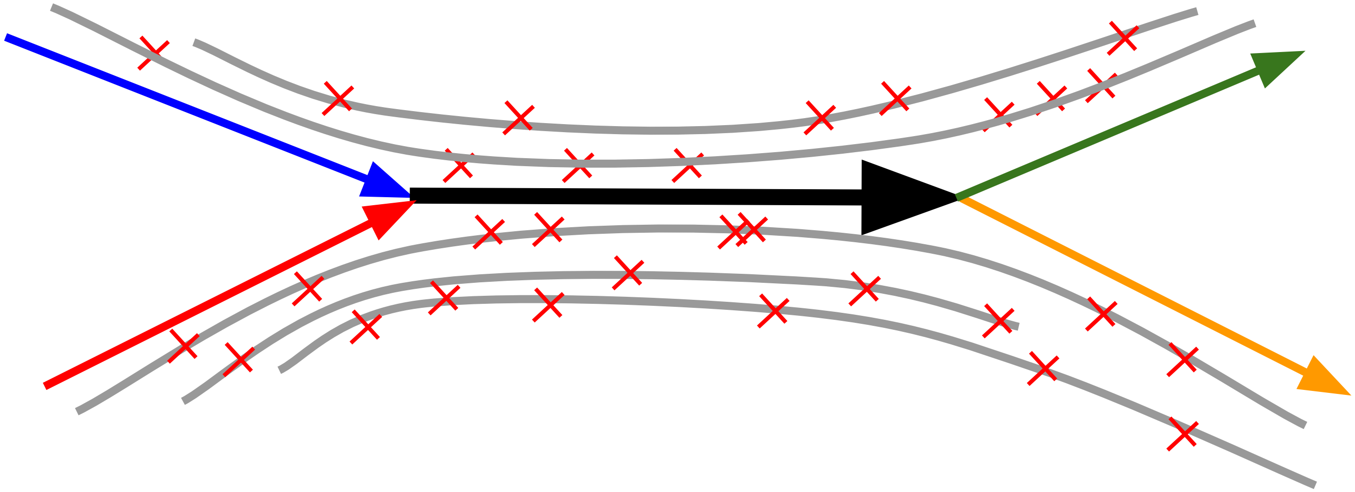
Long reads to the rescue



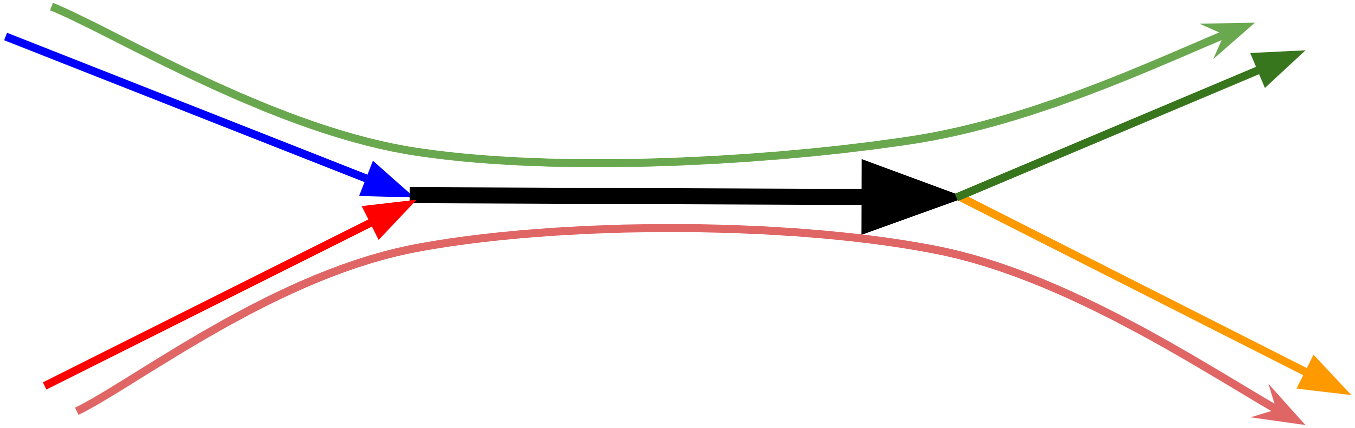
Long reads to the rescue



Long reads to the rescue



Long reads to the rescue



Sequencing technologies

	SANGER SEQUENCING	illumina	Roche 454 SEQUENCING	ion torrent by life technologies	PACBIO		Oxford NANOPORE Technologies
Protocols		HiSeq, MiSeq			Subreads	CCS / HiFi	MinIon
Read length	500-900	25-300	400-1100	200-400	20K-100K	5K-20K	1K-3M
Error rate	0.001-0.1%	0.1-1%	1%	1-2%	2-10%	0.1-2%	5-15%
Error type	Indels & Mismatches	Mismatches only	Indels & Mismatches	Indels & Mismatches	Indels & Mismatches	Indels & Mismatches	Indels & Mismatches
Comments	Remains the golden standard	Error rate grows at the end of read	Problems with homopolymers	Problems with homopolymers	Errors distributed randomly	Error rate depends on sequencing settings	Typically several deletions in a row
Cost	$$$$$	$$	$$	$$	$$$	$$$	$

PacBio only assembly

Thm:

Perfect assembly possible iff

a) errors random

b) sampling is Poisson

c) reads long enough 2 solve repeats.

Note: e-rate not needed

Gene Meyers' twitter

New long reads vs Sanger assembly

High error rate => overlap detection is harder

- miniasm
- MHAP
- …

Thank you!

Questions?