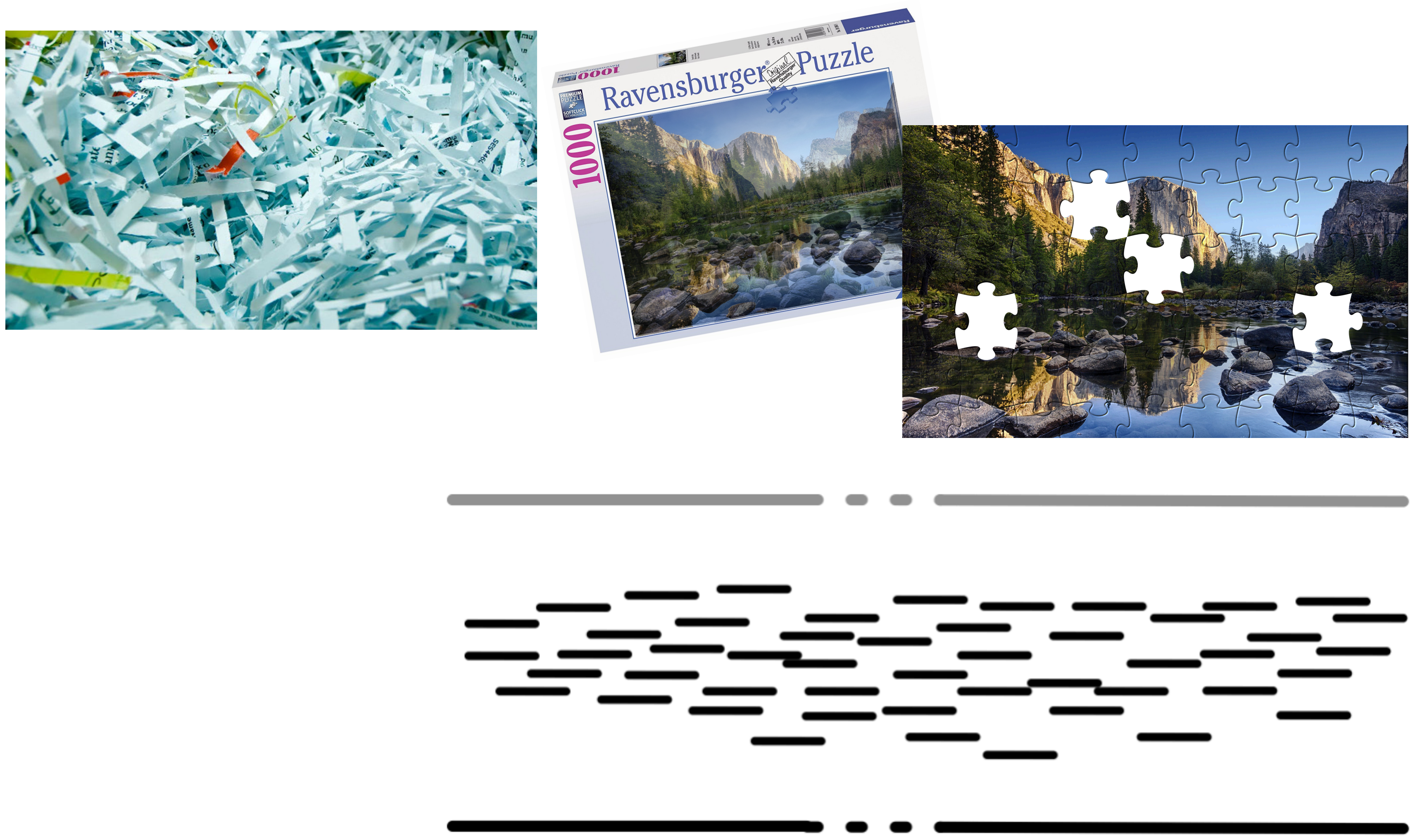
Genome Assembly and Quality Control

Bioinformatics Workshop for *M. tuberculosis* Genomics and Phylogenomics

July 9-14, 2018 @The Philippine Genome Center



Ulas Karaoz, PhD

Ecology Department
Berkeley Lab

BERKELEY LAB
Lawrence Berkeley National Laboratory
Bringing Science Solutions to the World

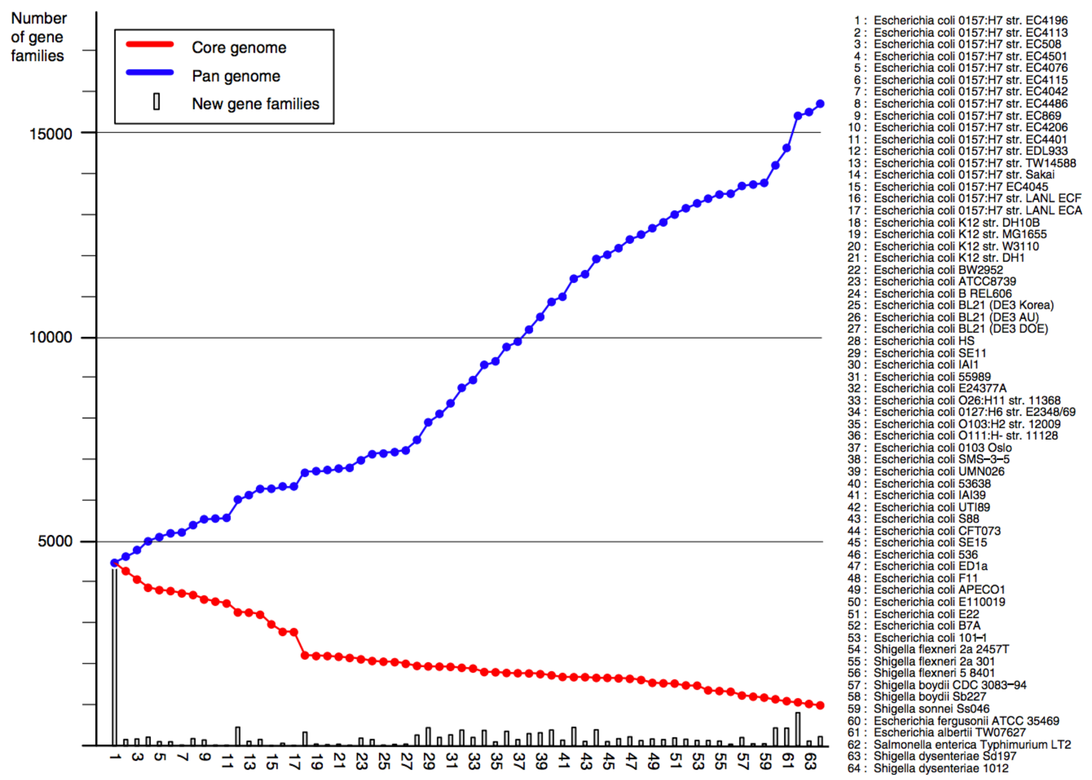
https://eesa.lbl.gov/profiles/ulas-karaoz, Email: ukaraoz@lbl.gov, Twitter: @ukaraoz

Learning Objectives

- How assemblers work
- Assembly algorithms for short and long reads
- Challenges for the assembly

Why Assemble Genomes?

- Reference isn't available
- Question/update/correct "reference" genome
- Discover novel gene content
- Discover novel insertions or SNPs in distant organisms
- Just because we can now?

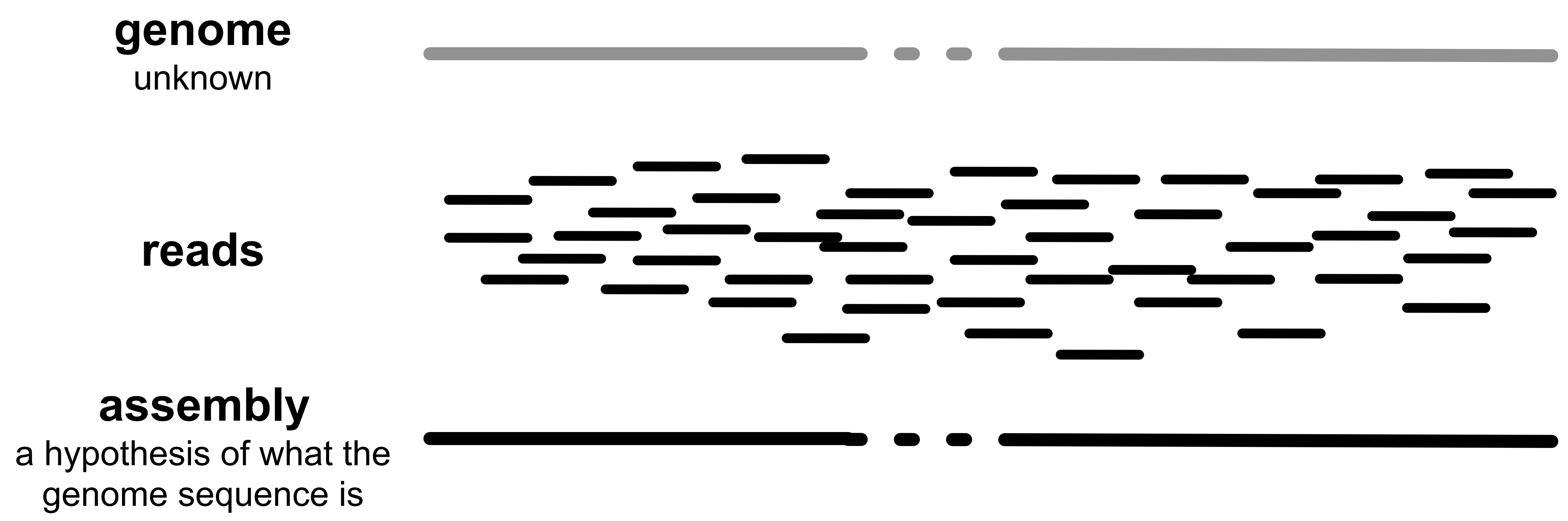


Lukjancenko. Microbial Ecology 2010. Comparison of 61 Sequenced Escherichia coli Genomes

Embrace Reality

An assembly is generally:

- fragmented
- only partly covers the genome



Assembly jargon

Read: sequence that is outputted by the sequencer

Paired read: a pair of reads, each from either end of the same fragment

Single read: a read from one end of the fragment

***k*-mer:** any sequence of length *k*

Contig: gap-less assembled sequence

Scaffold: ordered contigs with gaps

Gap: stretches with unknown/unresolveable sequence

Whole-genome Shotgun Sequencing

shotgun = random fragmentation

input DNA GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT

amplified DNA

GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT
GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT
GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT
GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT
GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT

fragmented DNA

GGCGGTAG GGGTATT TATATGCTTTTTT
CGGTAGC ATATGCT TTTTTT
AGCGCGGG
GGCGGT GGTATTATTT TTTTTT
CGGGTATTA TTATATATG
GGCGGT TGCTTTTTT
GTATTATTTAT

Whole-genome Shotgun Sequencing

given these fragments

```
GGCGGTAG
GGCGGT
GGCGGT
CGGTAGC
AGCGCGGG
CGGGTATTA
GGGTATT
GTATTATTTAT
ATATGCT
GGTATTATTT
TTATATATG
TATATGCTTTTTT
TTTTTT
TTTTTT
TGCTTTTT
```

reconstruct this

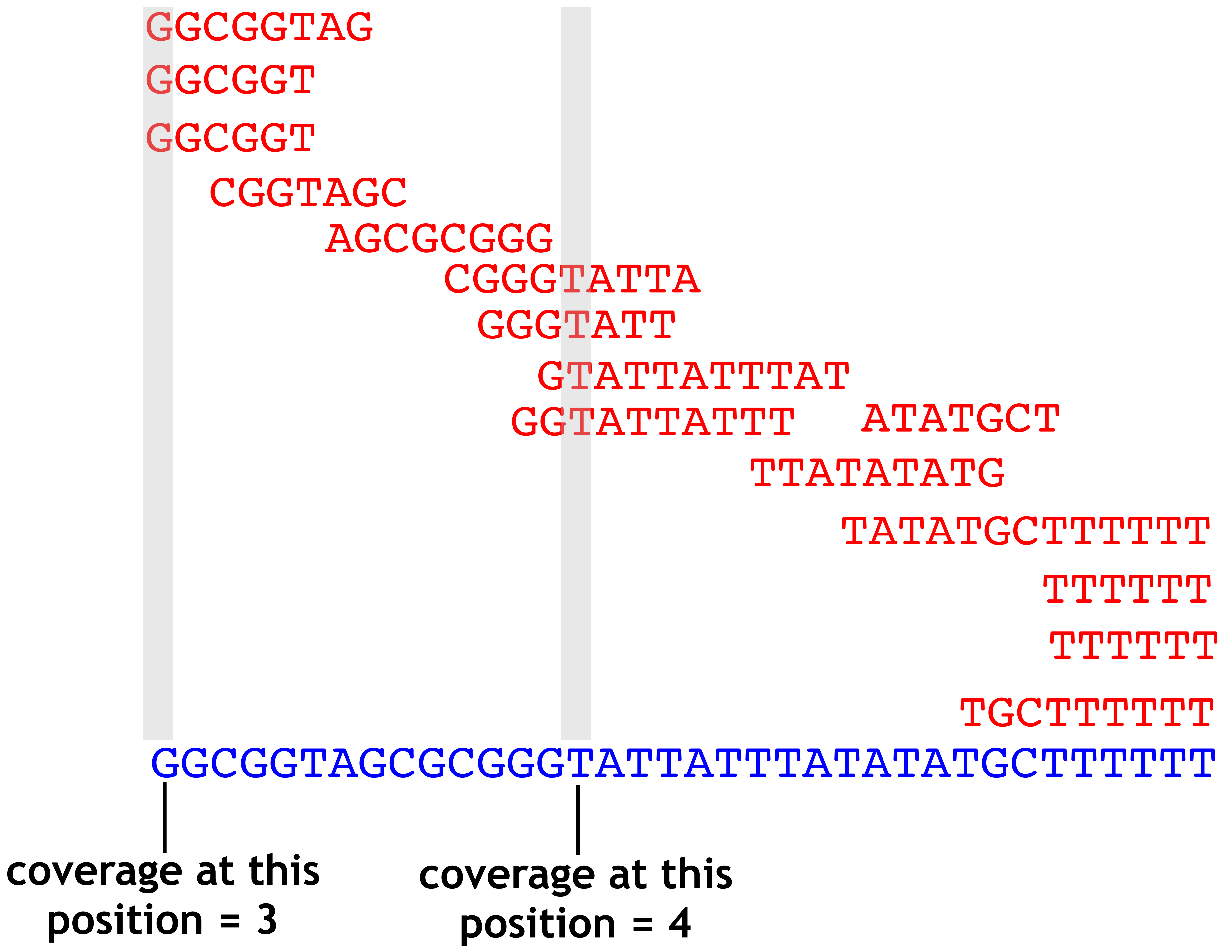
```
GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT
```

Whole-genome Shotgun Sequencing

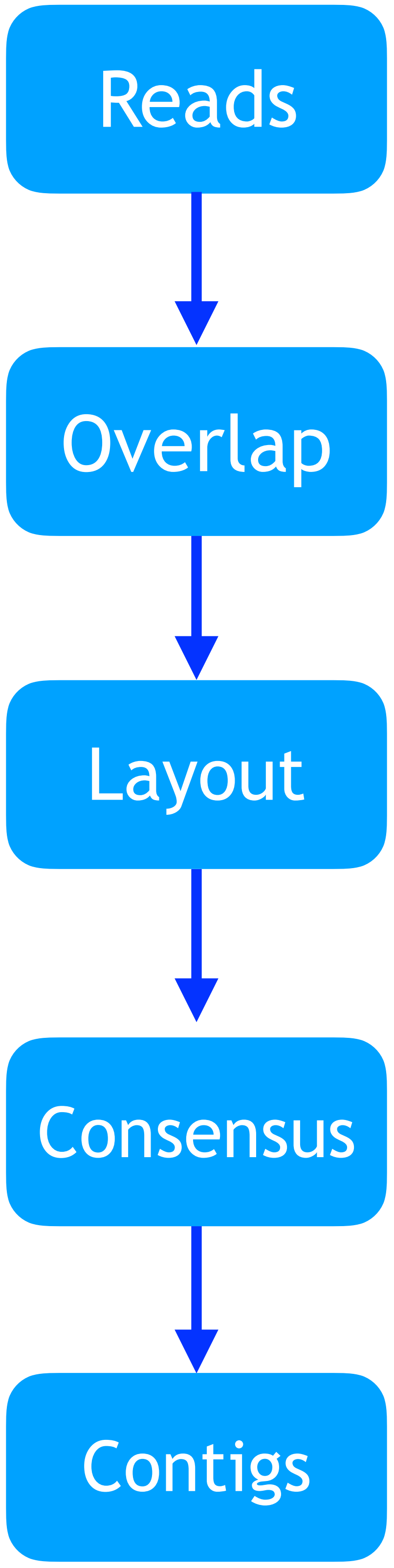
```
GGCGGTAG
GGCGGT
GGCGGT
  CGGTAGC
      AGCGCGGG
          CGGGTATTA
           GGGTATT
             GTATTATTTAT
            GGTATTATTT   ATATGCT
                     TTATATATG
                        TATATGCTTTTTT
                               TTTTTT
                                TTTTTT
                             TGCTTTTTT

GGCGGTAGCGCGGGTATTATTTATATATGCTTTTTT
```

Key concept: (Depth of) Coverage



Long Read Assembly



Find significant overlaps between reads, build a graph

<u>**"overlap-layout-consensus"**</u>

Bundle

Determine most likely base

Long Read Assembly: Overlap

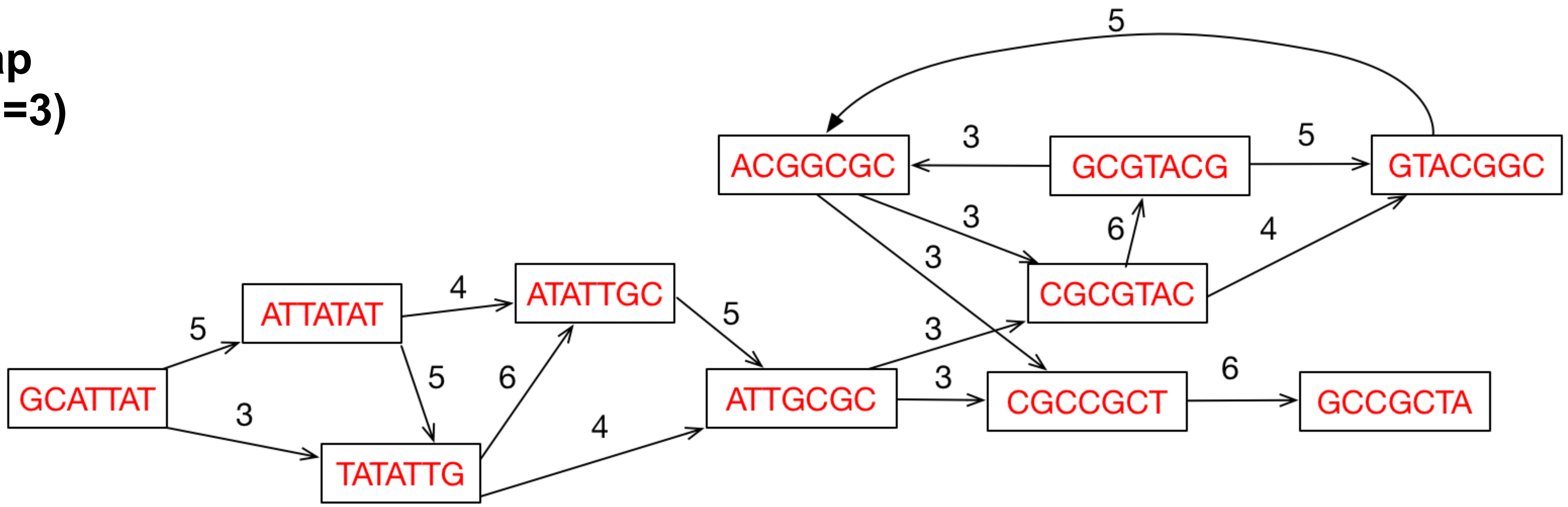
unknown string

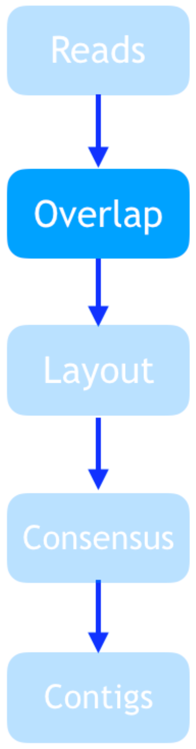
```
GCATTATATATTGCGCGTACGGCGCCGCTACA
```

short fragments

```
GCATTA
  ATTATAT
      TATATTG
       ATATTGC
         ATTGCGC
             CGCGTAC
              GCGTACG
                GTACGGC
                  ACGGCGC
                      CGCCGCT
                       GCCGCTACA
```

overlap graph (l=3)





Long Read Assembly: Layout

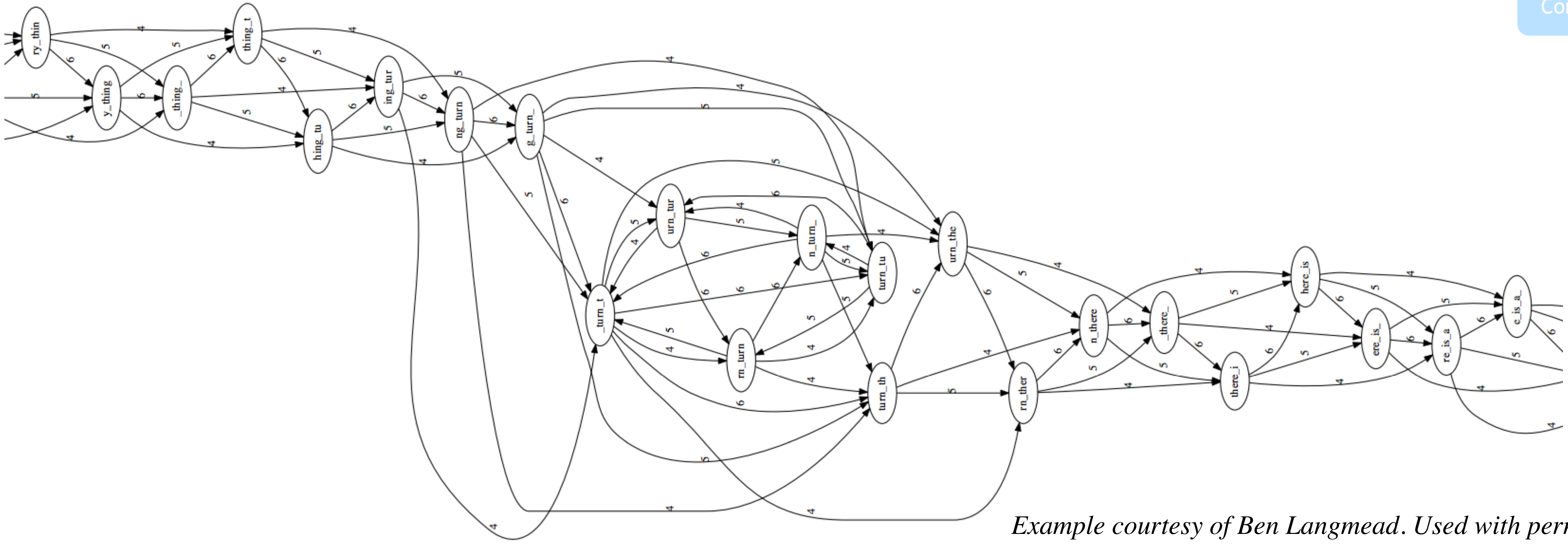
Consider the following sentence:

"to every thing turn turn turn there is a season"

with:

read length = 7, l (overlap length) = 4

unknown string to_every_thing_turn_turn_turn_there_is_a_season



Reads → Overlap → Layout → Consensus → Contigs

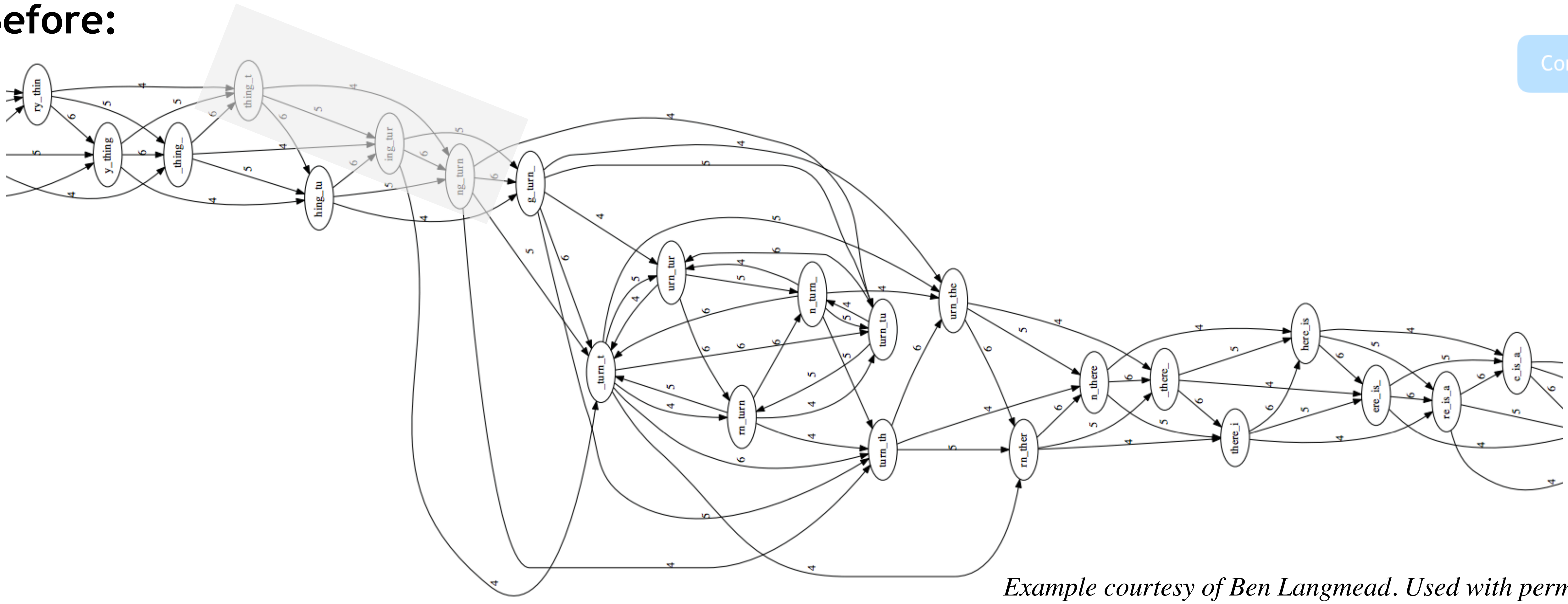
Example courtesy of Ben Langmead. Used with permission.
http://www.langmead-lab.org/teaching-materials/

Long Read Assembly: Layout

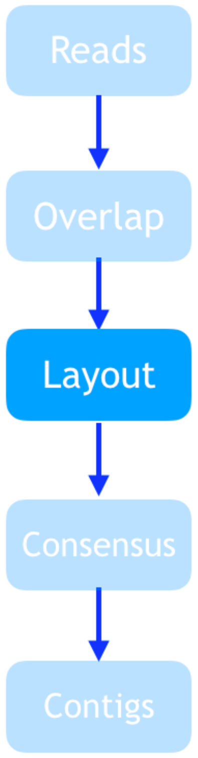
Remove transitively inferrible connections, starting with connections skipping one node:

$$\begin{tikzcd} \text{thing\_t} \arrow[r, "5"] \arrow[rr, bend left, "4 \;\times"] & \text{ing\_tur} \arrow[r, "6"] & \text{ng\_turn} \end{tikzcd}$$

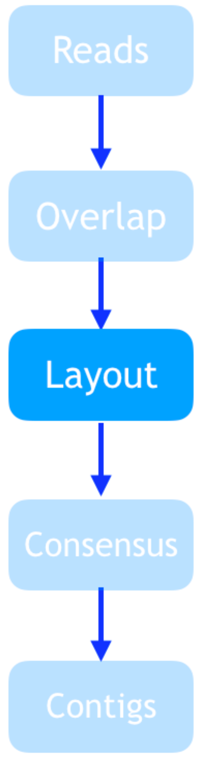
Before:



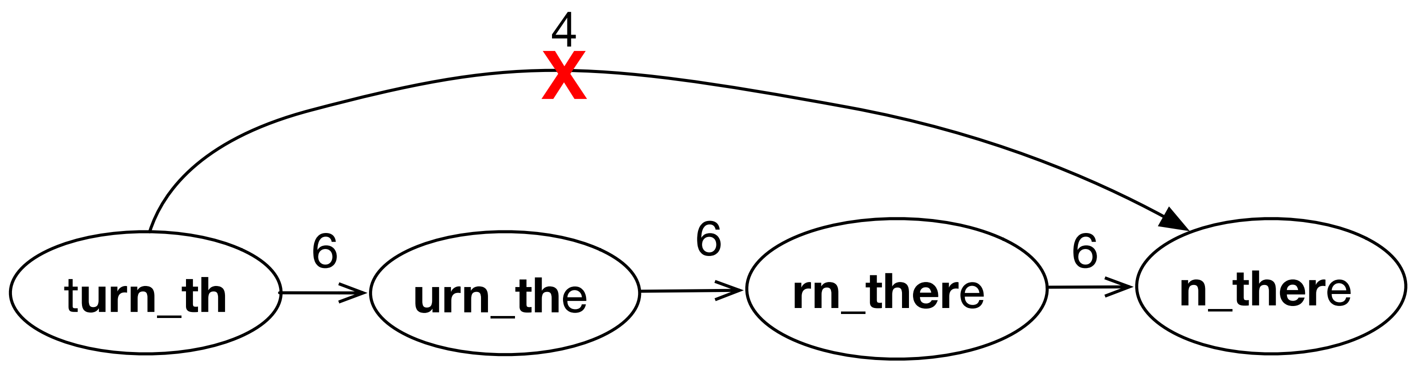
Example courtesy of Ben Langmead. Used with permission.
http://www.langmead-lab.org/teaching-materials/



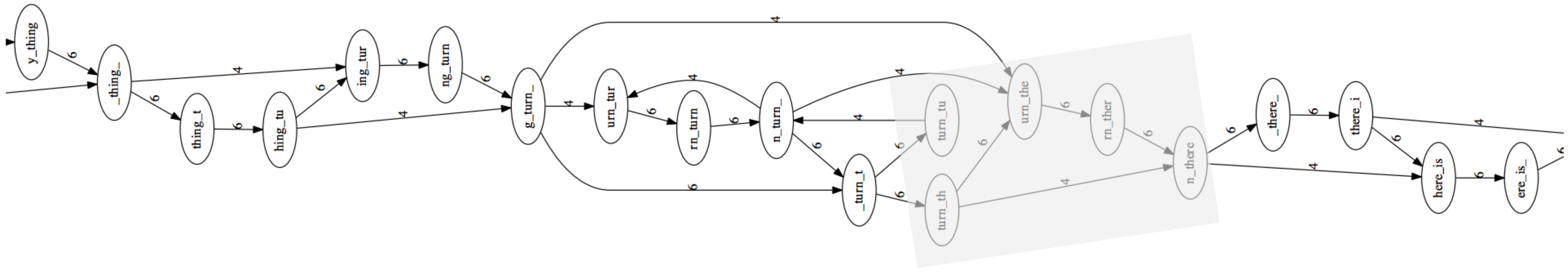
Long Read Assembly: Layout



Remove transitively inferrible connections, starting with connections skipping one or *two* nodes:



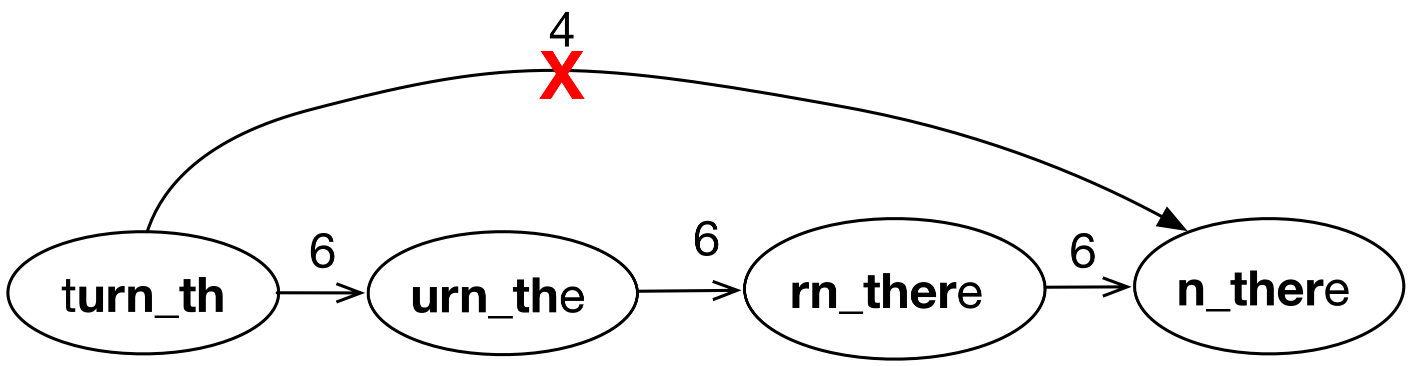
After:

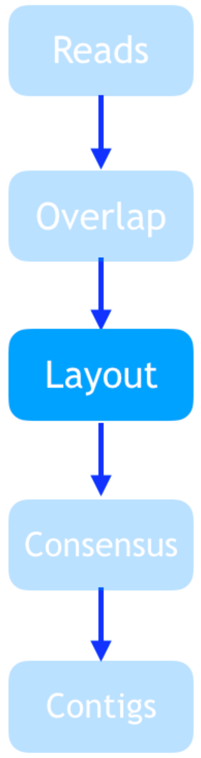


Example courtesy of Ben Langmead. Used with permission.
http://www.langmead-lab.org/teaching-materials/

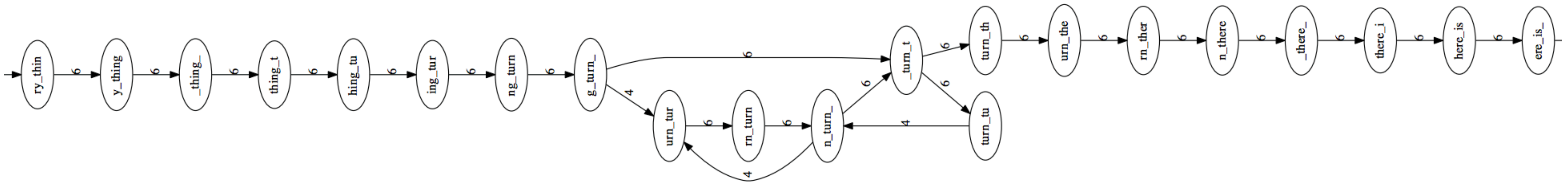
Long Read Assembly: Layout

Remove transitively inferrible connections, starting with connections skipping one or *two* nodes:





Simpler:

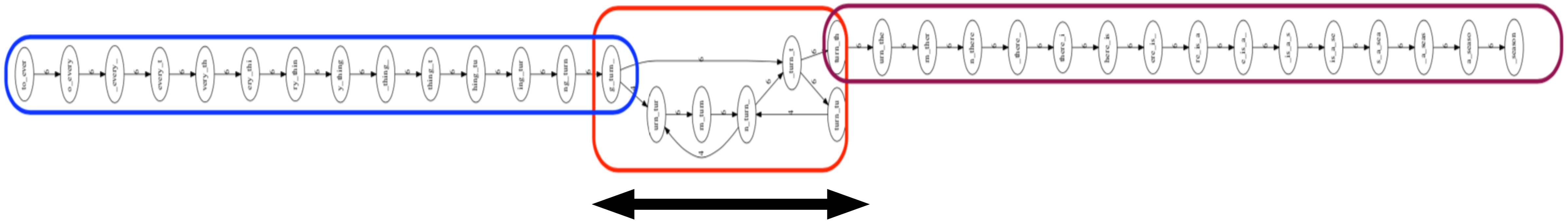


Example courtesy of Ben Langmead. Used with permission.
http://www.langmead-lab.org/teaching-materials/

Long Read Assembly: Layout

Contigs are non-branching "*contiguous*" stretches

`to_every_thing_turn_turn_turn_there_is_a_season`



`to_every_thing_turn` `turn_there_is_a_season`

contig 1 — repeat, cannot be resolved — contig 2

Example courtesy of Ben Langmead. Used with permission.
http://www.langmead-lab.org/teaching-materials/

Long Read Assembly: Consensus

$$\begin{tikzcd}
\text{Reads} \arrow[d] \\
\text{Overlap} \arrow[d] \\
\text{Layout} \arrow[d] \\
\boxed{\text{Consensus}} \arrow[d] \\
\text{Contigs}
\end{tikzcd}$$

reads that make up a contig, lined-up

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

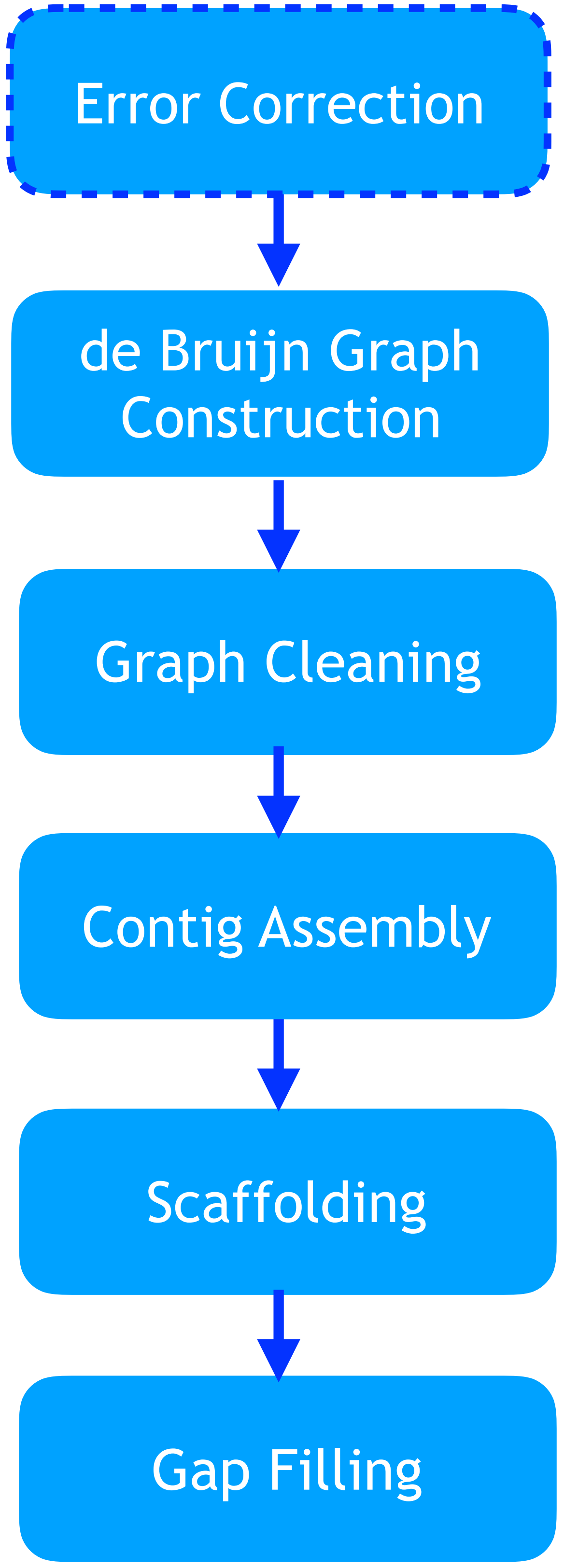
consensus base calls

```
TAGATTACACAGATTACTGACTTGATGGCGTAA-CTA
```

Example courtesy of Ben Langmead. Used with permission.
http://www.langmead-lab.org/teaching-materials/

Short Read Assembly

"**de Bruijn graph based**"

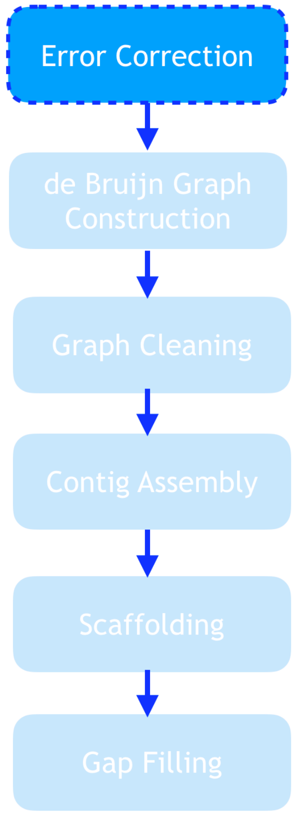


Short Read Assembly: k-mer

***k*-mer:** any sequence of length *k*

***mer*:** From Ancient Greek μέρος (*méros*, “part”)
e.g. poly-mer, mono-mer

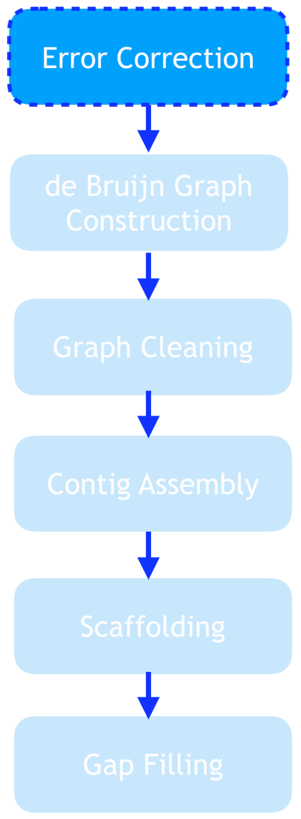
```
              S:  GGCGGTAGCGCG
   a 4-mer of S:      GTAG
all 3-mers of S:  GGC
                   GCG
                    CGG
                     GGT
                      GTA
                       TAG
                        AGC
                         GCG
                          CGC
                           GCG
```



Short Read Assembly: Error correction using k-mers

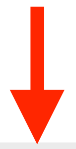
- Consider a read with a single error

ACGATGCATCGACTATGTACGATCGATCGATTACGAGATCAGCTACTAGCATCTACGATAG



Short Read Assembly: Error correction using k-mers

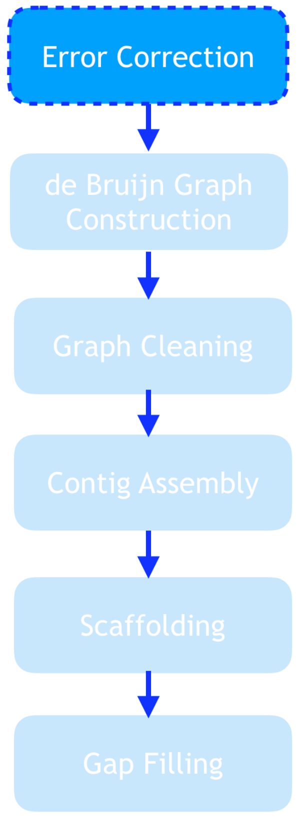
- Consider a read with a single error



ACGATGCATCGACTATGTACGATCGATCGATTACGAGATCAGCTACTAGCATCTACGATAG

- We can count the number of times each k-mer in the read is present in all reads

"k-mers containing errors appear few times"



Short Read Assembly: Error correction using k-mers

- Consider a read with a single error

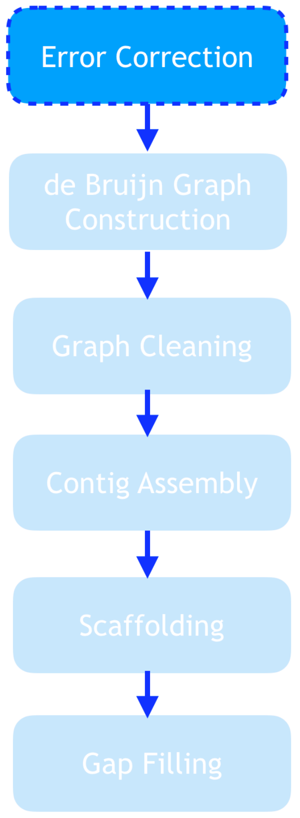
ACGATGCATCGACTATGTACGATCGATCGATTACGAGATCAGCTACTAGCATCTACGATAG

- We can count the number of times each k-mer in the read is present in all reads

"k-mers containing errors appear few times"

count(ACGATGCATCGACTATGTAC)=100

count(CGAGATCAGCTACTAGCATC)=1



Short Read Assembly: Error correction using k-mers

- Consider a read with a single error

ACGATGCATCGACTATGTACGATCGATCGATTACGAGATCAGCTACTAGCATCTACGATAG

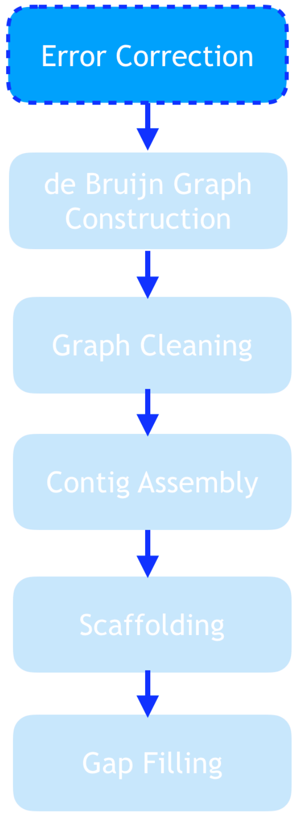
- We can count the number of times each k-mer in the read is present in all reads

"k-mers containing errors appear few times"

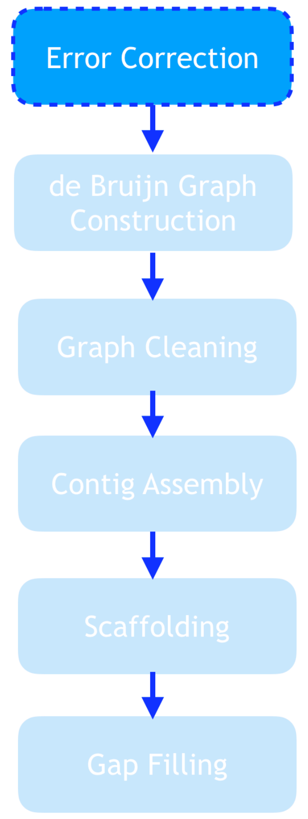
count(ACGATGCATCGACTATGTAC)=100

count(CGAGATCAGCTACTAGCATC)=1

- To correct: replace rare k-mers with common k-mers



Short Read Assembly: Error correction using k-mers



- Consider a read with a single error

ACGATGCATCGACTATGTACGATCGATCGATTACGAGATCAGCTACTAGCATCTACGATAG

- We can count the number of times each k-mer in the read is present in all reads

"k-mers containing errors appear few times"

count(ACGATGCATCGACTATGTAC)=100

count(CGAGATCAGCTACTAGCATC)=1

- To correct: replace rare k-mers with common k-mers
- Many *k*-mer based correctors are available:
  - Quake, sga, soapdenovo, bfc, bless, lighter, musket

Short Read Assembly: Error correction using k-mers

- Consider a read with a single error

ACGATGCATCGACTATGTACGATCGATCGATTACGAGATCAGCTACTAGCATCTACGATAG

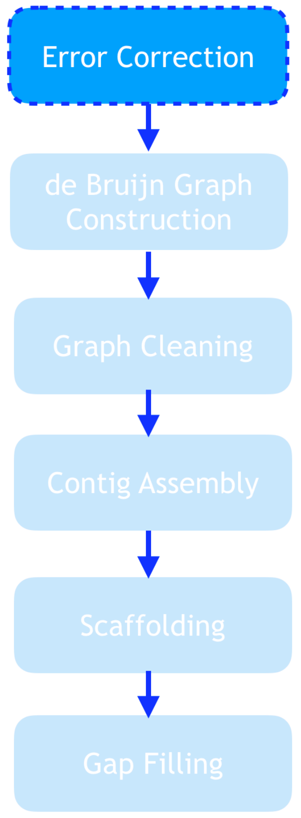
- We can count the number of times each k-mer in the read is present in all reads

"k-mers containing errors appear few times"

count(ACGATGCATCGACTATGTAC)=100

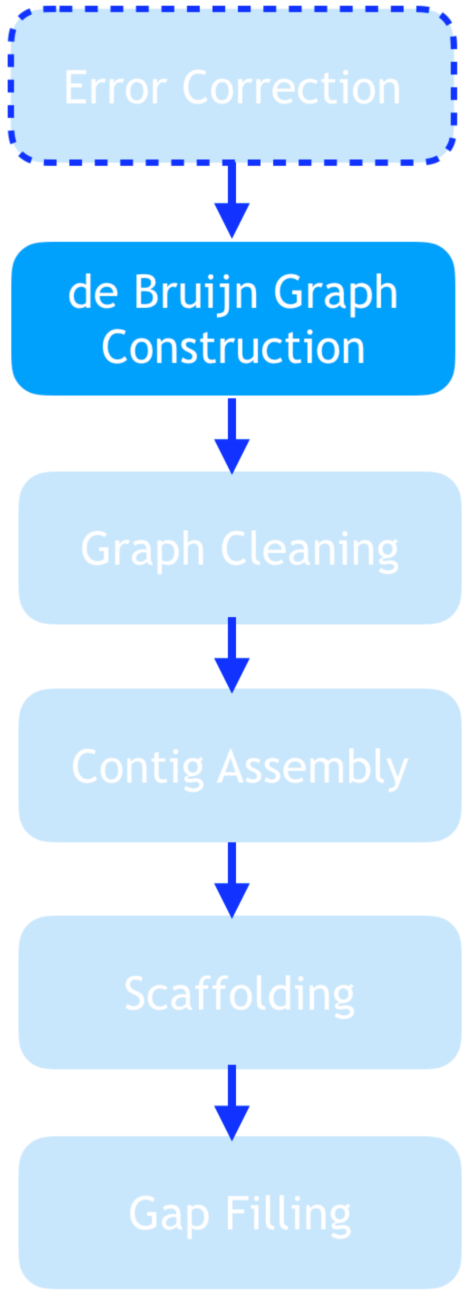
count(CGAGATCAGCTACTAGCATC)=1

- To correct: replace rare k-mers with common k-mers
- Many *k*-mer based correctors are available:
  - Quake, sga, soapdenovo, bfc, bless, lighter, musket
- Alternative error-correction strategy: find inexact overlaps between reads
  - very slow, impractical for large datasets



Short Read Assembly: de Bruijn Graphs

- Computing overlaps between pairs of short reads is computationally infeasible
- de Bruijn graph assemblers break reads into *k-mers* and link adjacent *k-mers* with an edge

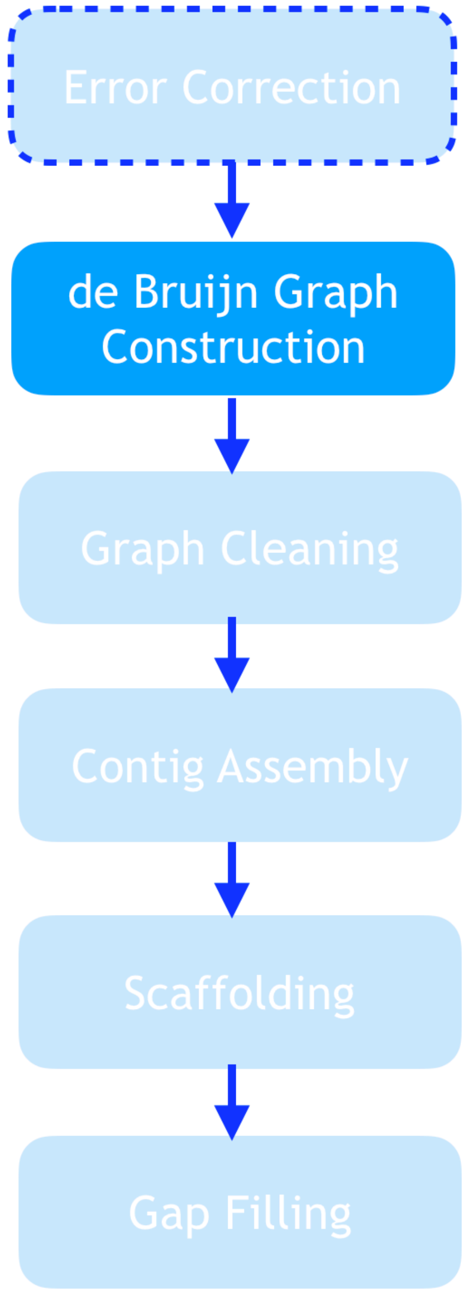


de Bruijn, 1946
Idury et al., 1995
Pevzner et al., 2001

Short Read Assembly: de Bruijn Graphs

- Computing overlaps between pairs of short reads is computationally infeasible
- de Bruijn graph assemblers break reads into *k-mers* and link adjacent *k-mers* with an edge

reads: CCGTTA, TTACGTT, TACGTT, CGTTCG, GTTCGA



de Bruijn, 1946
Idury et al., 1995
Pevzner et al., 2001

Short Read Assembly: de Bruijn Graphs

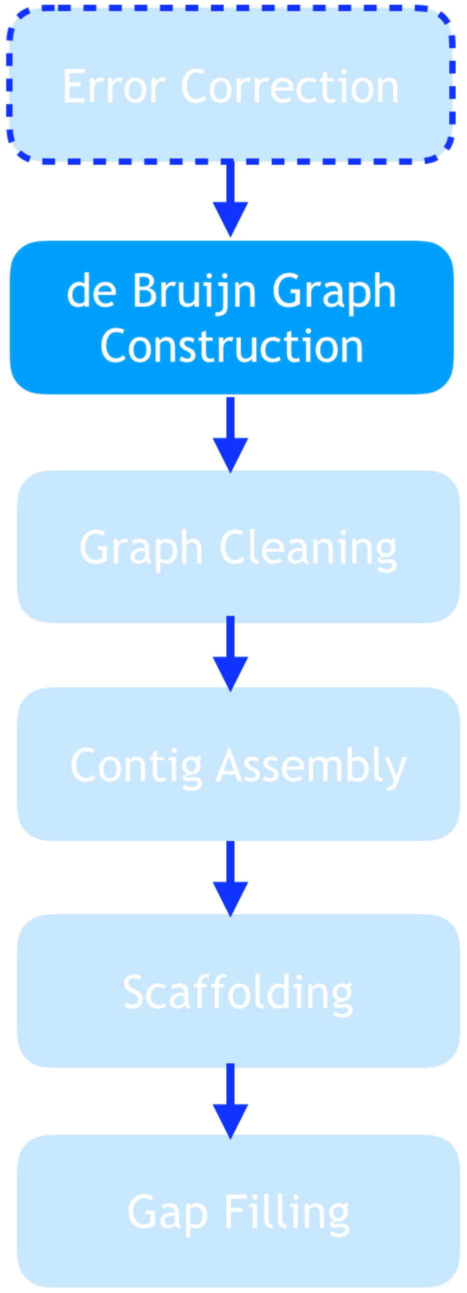
- Computing overlaps between pairs of short reads is computationally infeasible
- de Bruijn graph assemblers break reads into *k-mers* and link adjacent *k-mers* with an edge

reads: CCGTTA, TTACGTT, TACGTT, CGTTCG, GTTCGA

de Bruijn graph for k=4:

$$\begin{tikzcd}
 & & \text{GTTC} \arrow[r] & \text{TTCG} \arrow[r] & \text{TCGA} & \\
\text{CCGT} \arrow[r] & \text{CGTT} \arrow[ur] \arrow[r] & \text{GTTA} \arrow[r] & \text{TTAC} \arrow[r] & \text{TACG} \arrow[r] & \text{ACGT} \arrow[llll, bend left]
\end{tikzcd}$$

de Bruijn, 1946
Idury et al., 1995
Pevzner et al., 2001

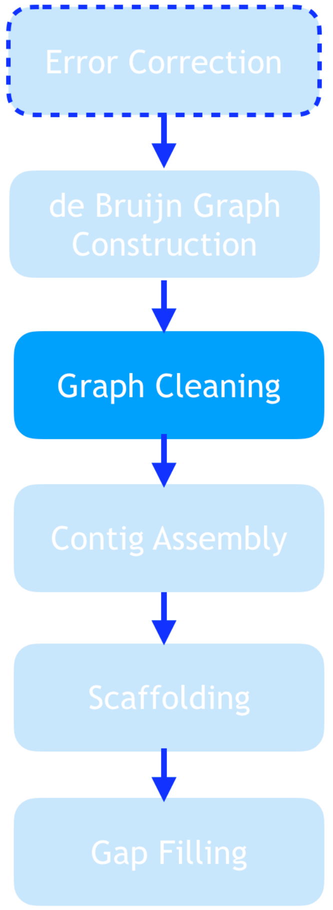


Short Read Assembly: Graph Artefacts

read1: GTATCGATCGACTAGCTACGACTAGCTACGATCGACTACGATCA

read2: TCGATCGACTAGCTACGACTAGCTACGATCGACTACGAACAGC

read3: GATCGACTAGCTACGACTAGCTACGATCGACTACGATCGGCATC

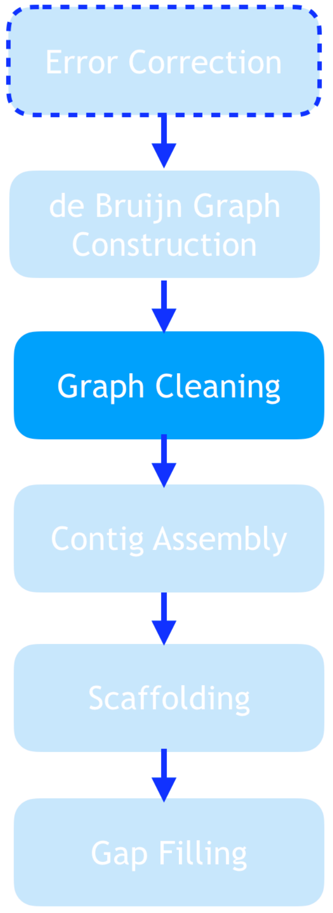


Short Read Assembly: Graph Artefacts

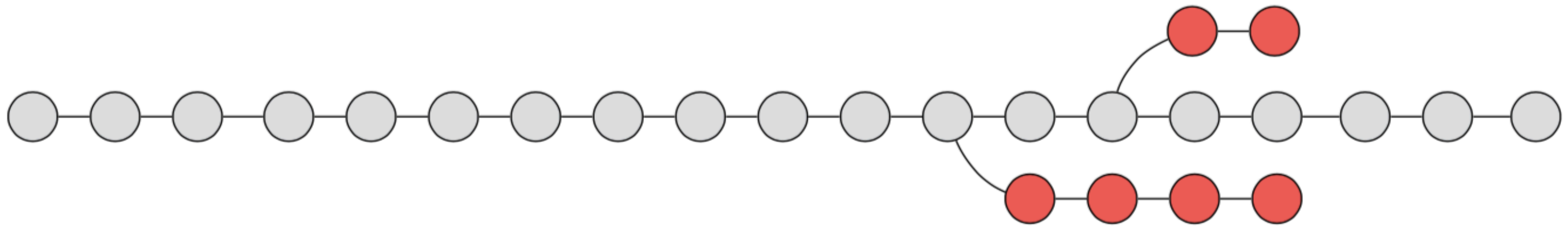
read1: GTATCGATCGACTAGCTACGACTAGCTACGATCGACTACGATCA

read2: TCGATCGACTAGCTACGACTAGCTACGATCGACTACGAACAGC

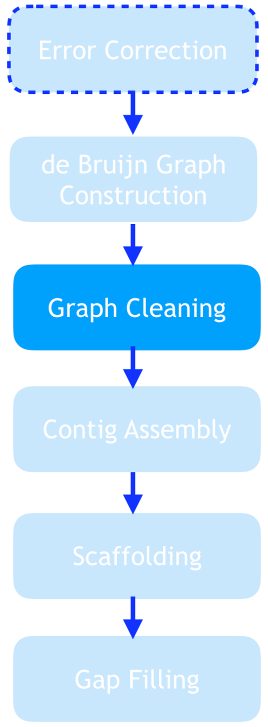
read3: GATCGACTAGCTACGACTAGCTACGATCGACTACGATCGGCATC



Tips



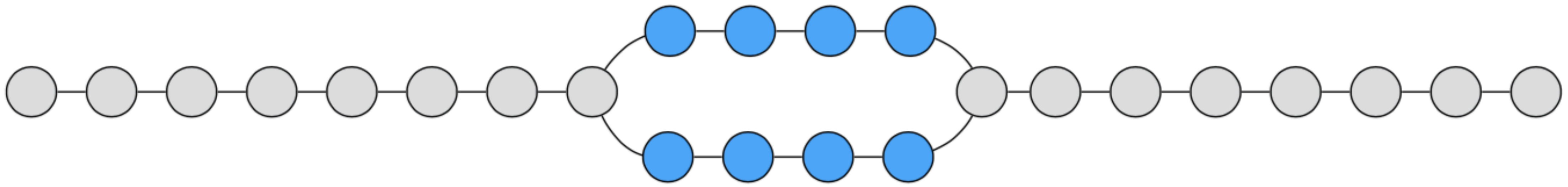
Short Read Assembly: Graph Artefacts



allele1: GACTAGCTATATCGATCGATCGATCGATCTCTAGACTACGACTGAAATC

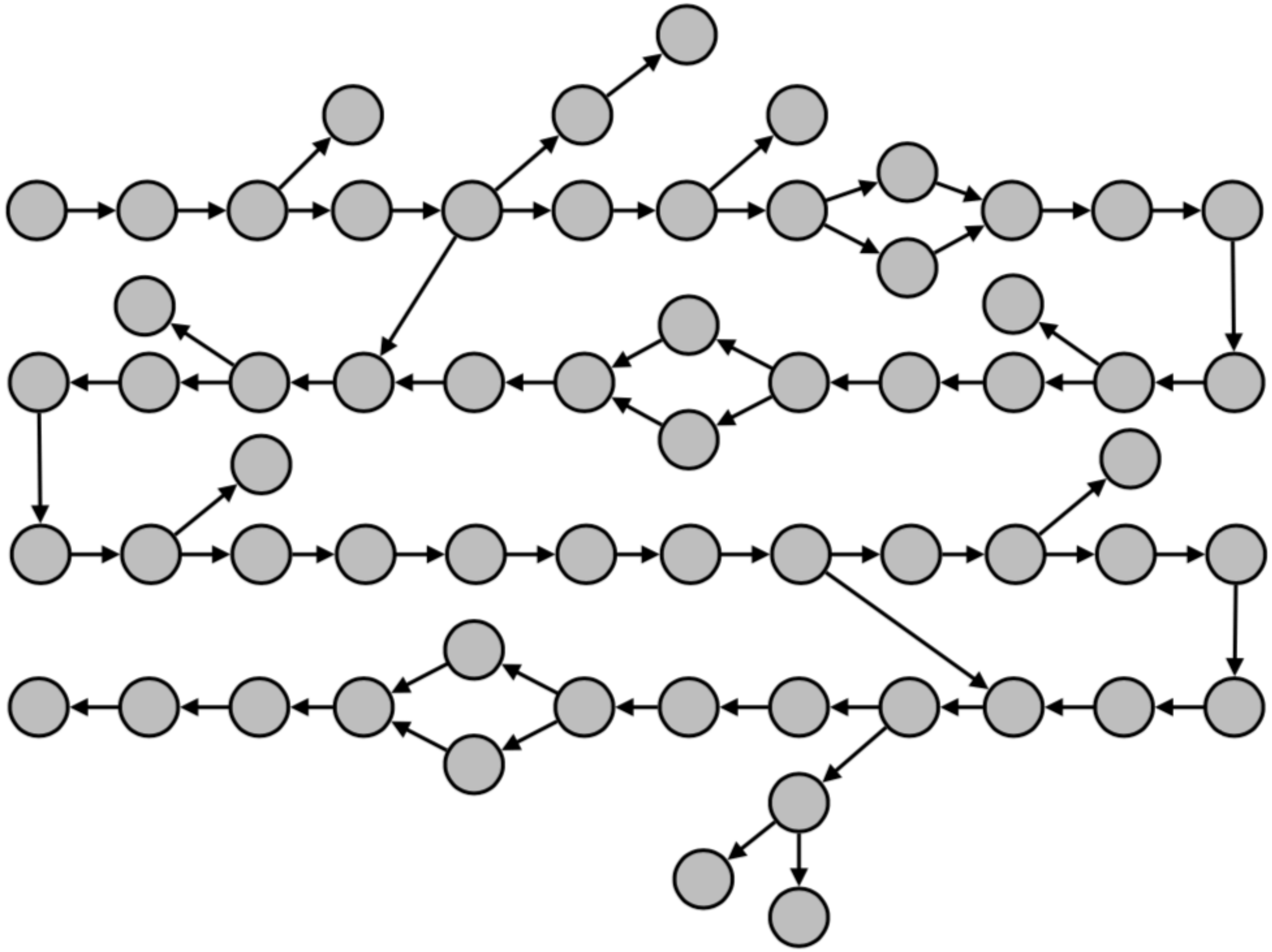
allele2: GACTAGCTATATCGATCGATCGATGGATCTCTAGACTACGACTGAAATC

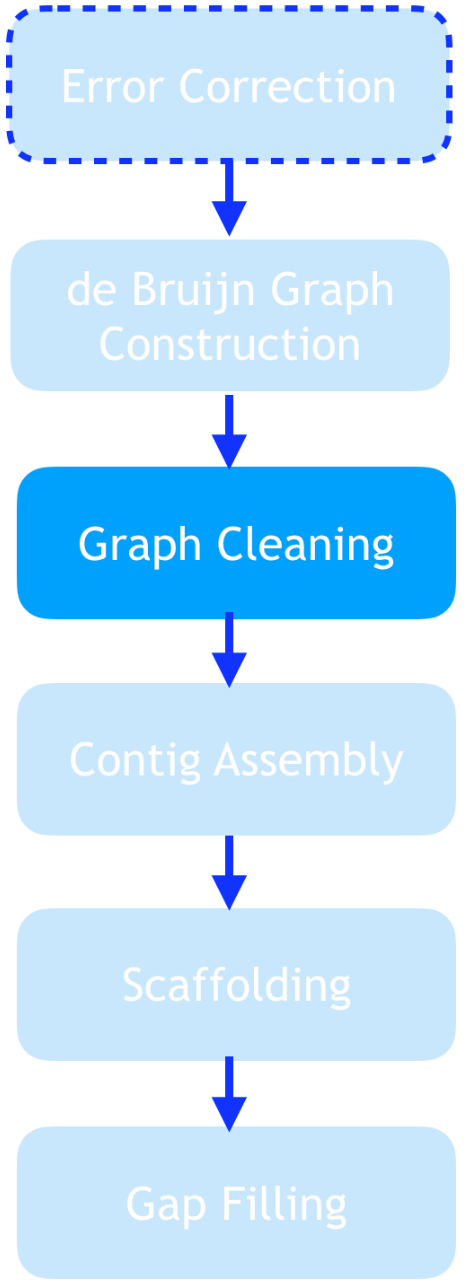
Bubbles



Bioinformatics Workshop for *M. tuberculosis* Genomics and Phylogenomics
@The Philippine Genome Center

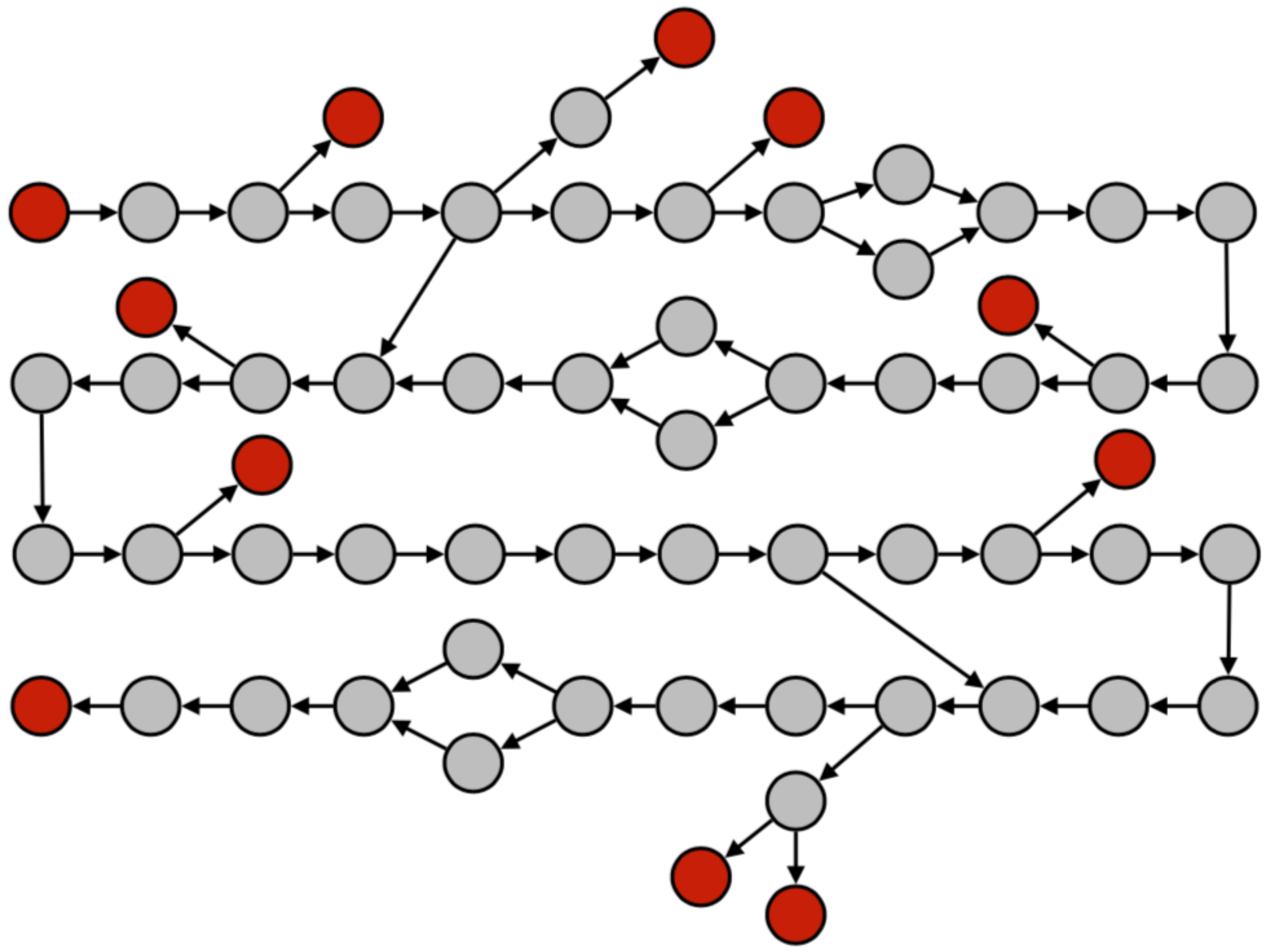
Short Read Assembly: Graph Cleaning

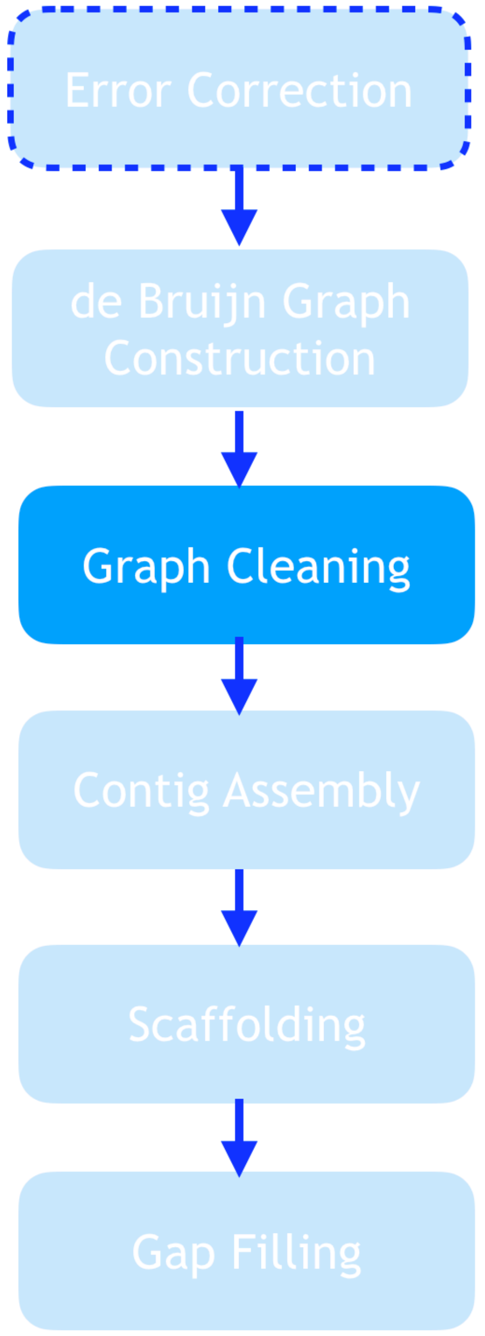




Short Read Assembly: Graph Cleaning

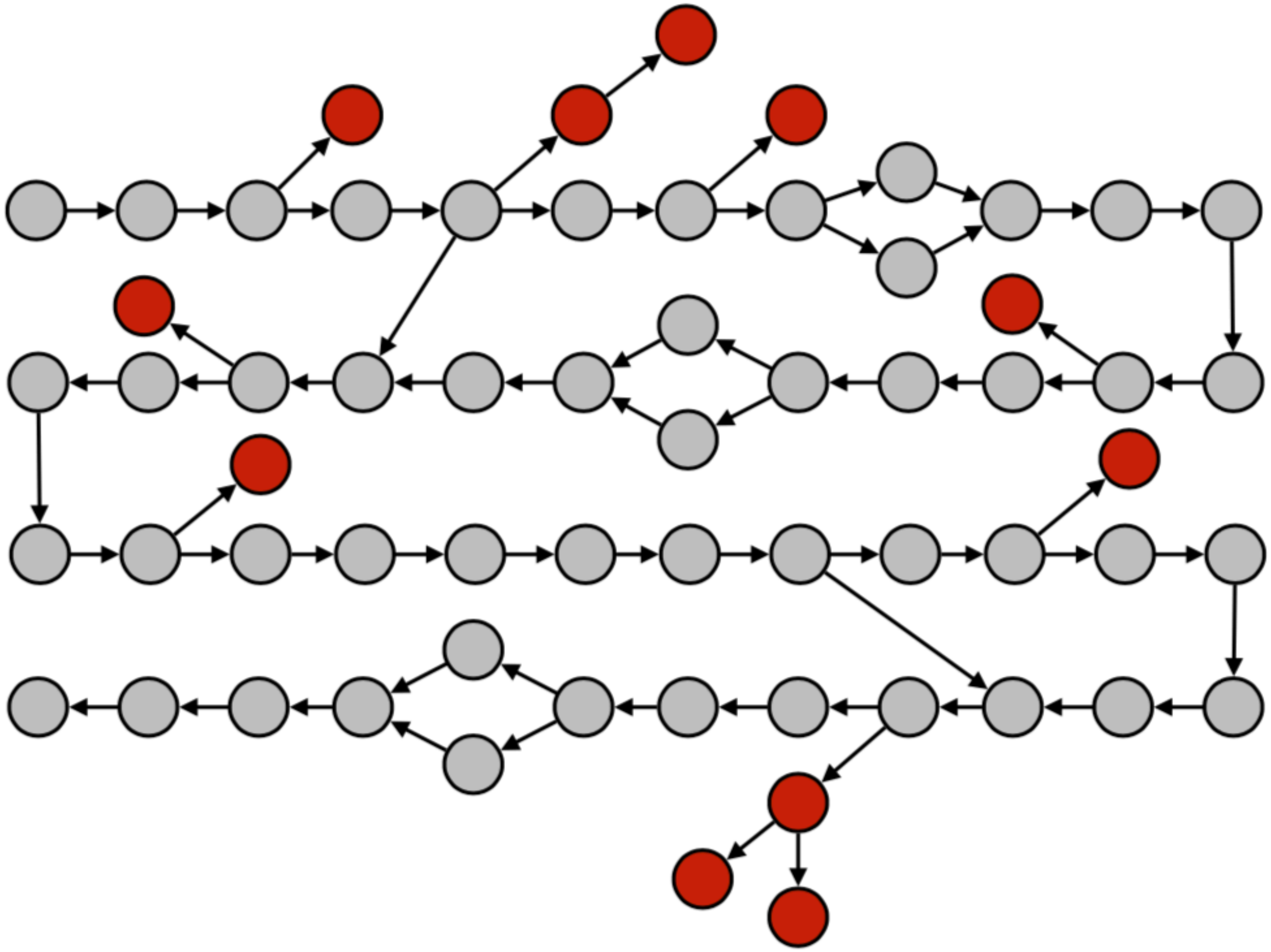
Tip Removal

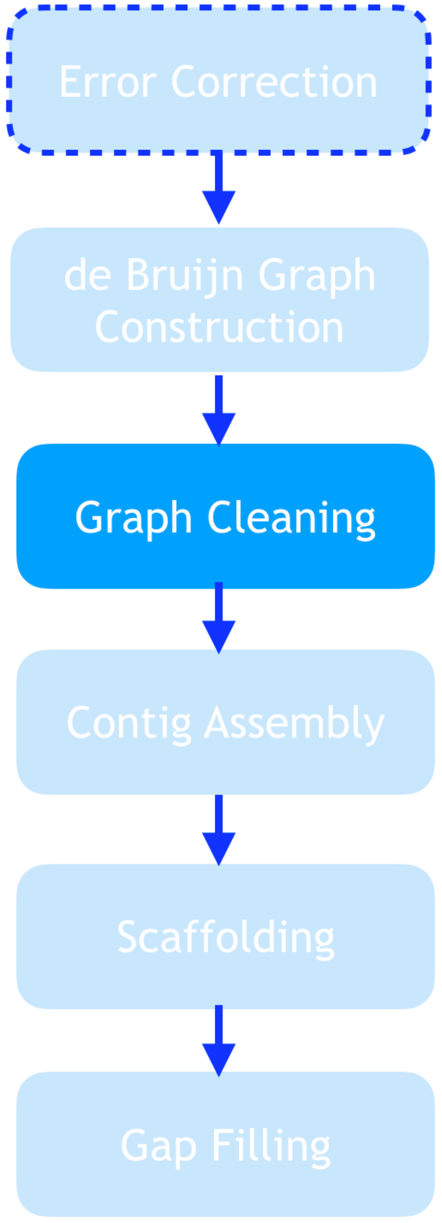




Short Read Assembly: Graph Cleaning

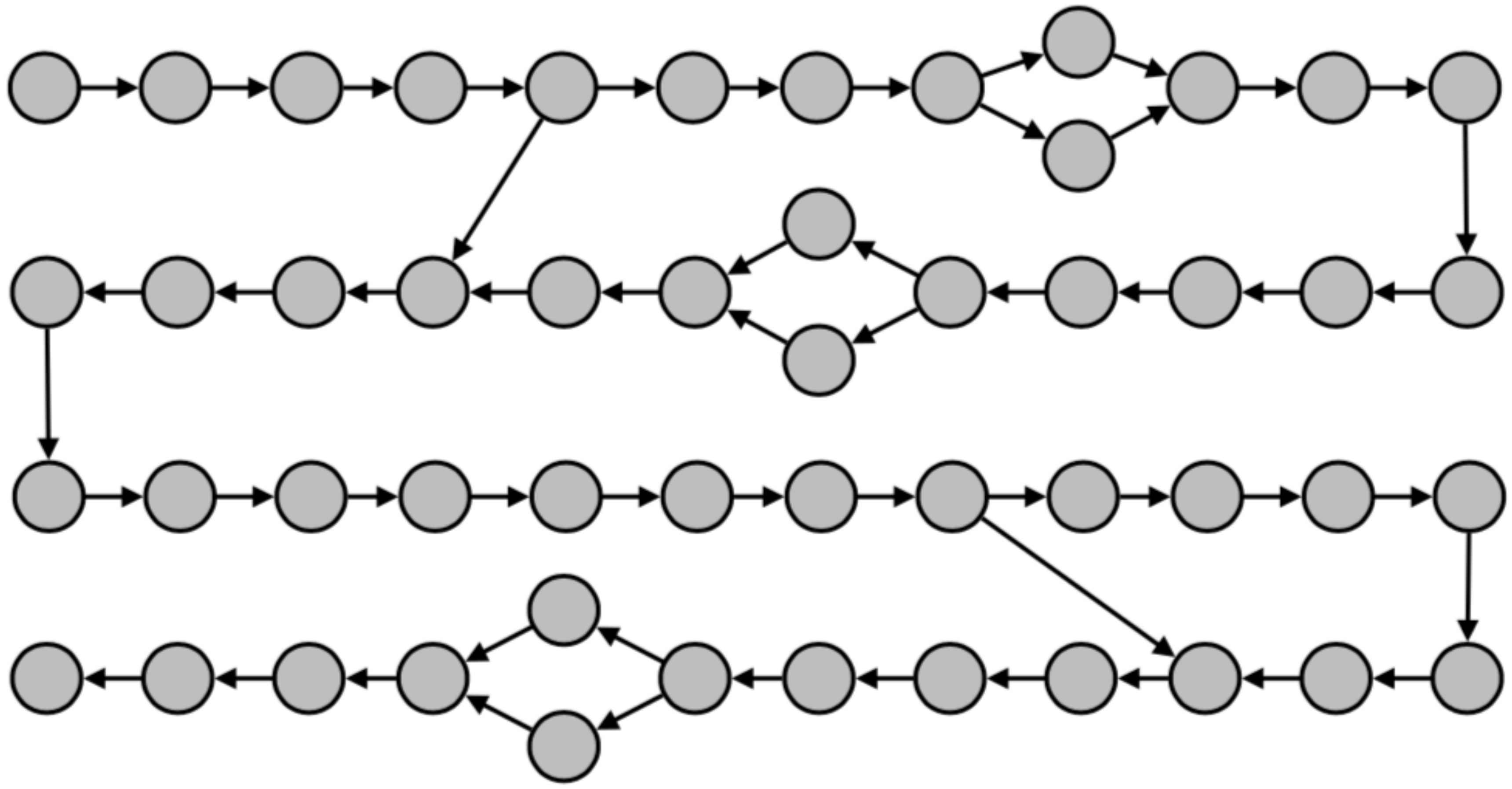
Tip Removal

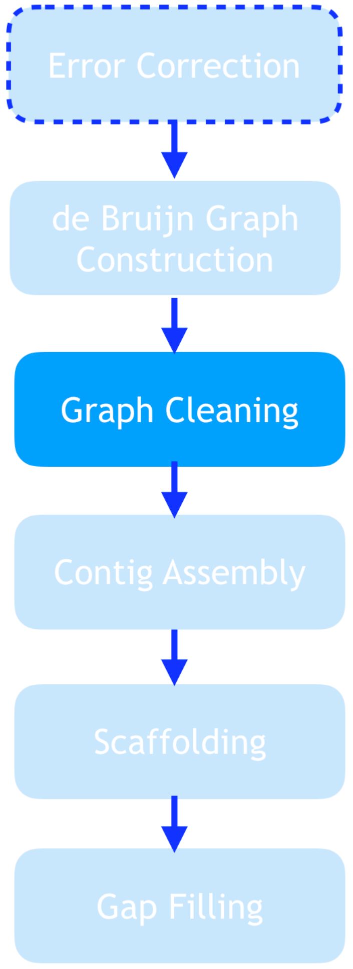




Short Read Assembly: Graph Cleaning

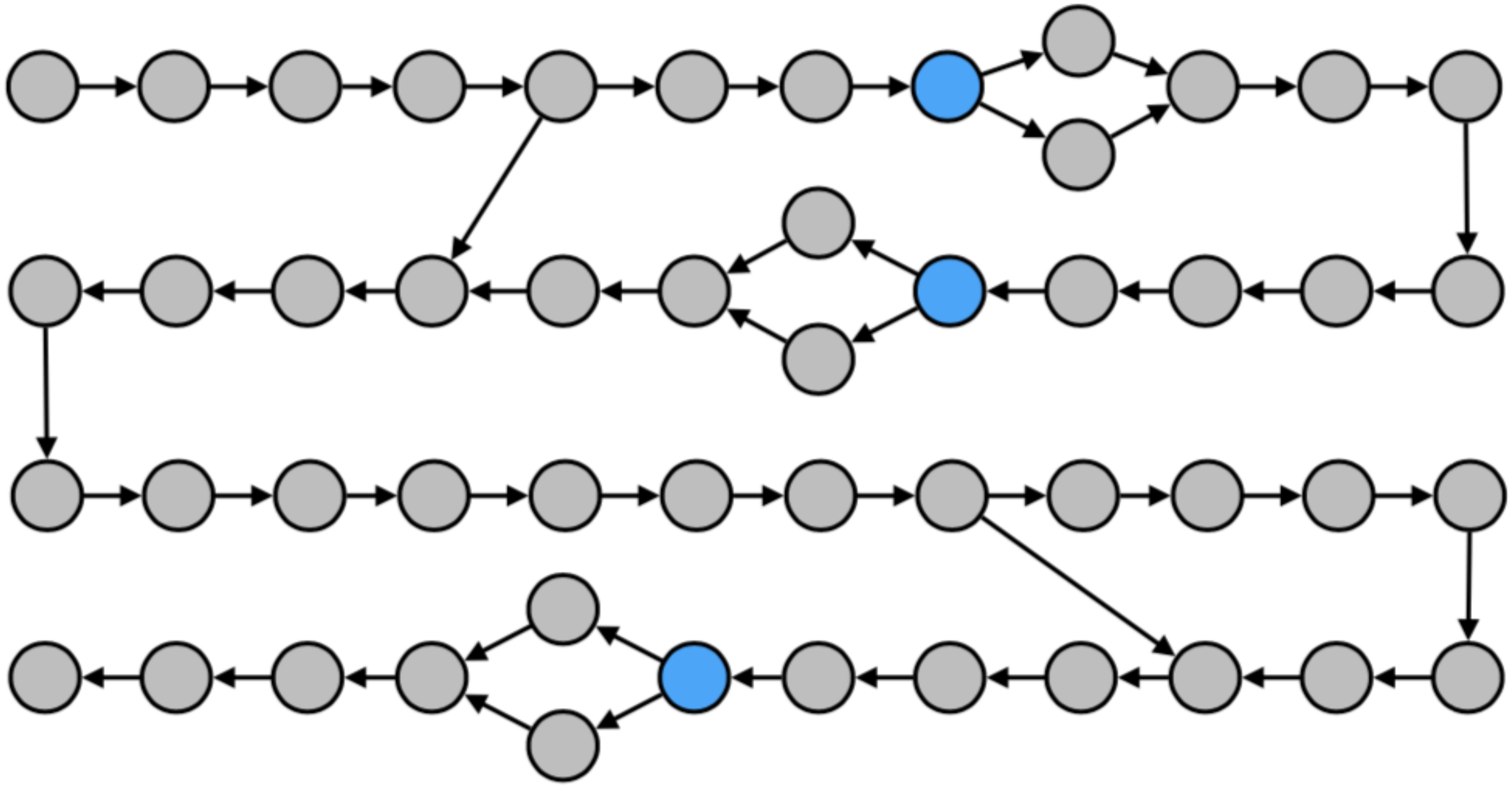
Tip Removal

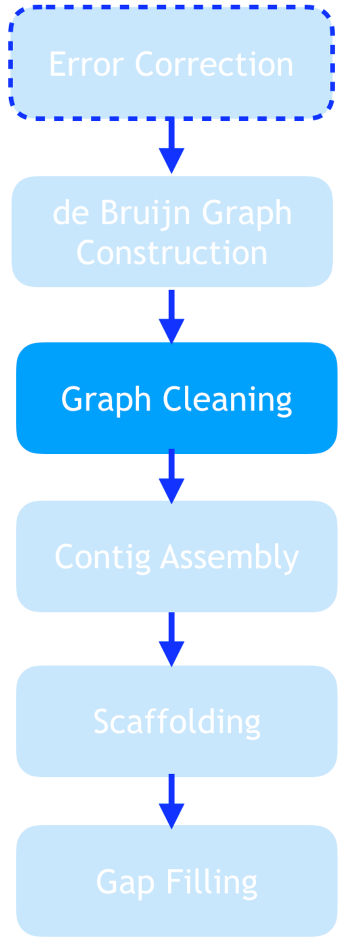




Short Read Assembly: Graph Cleaning

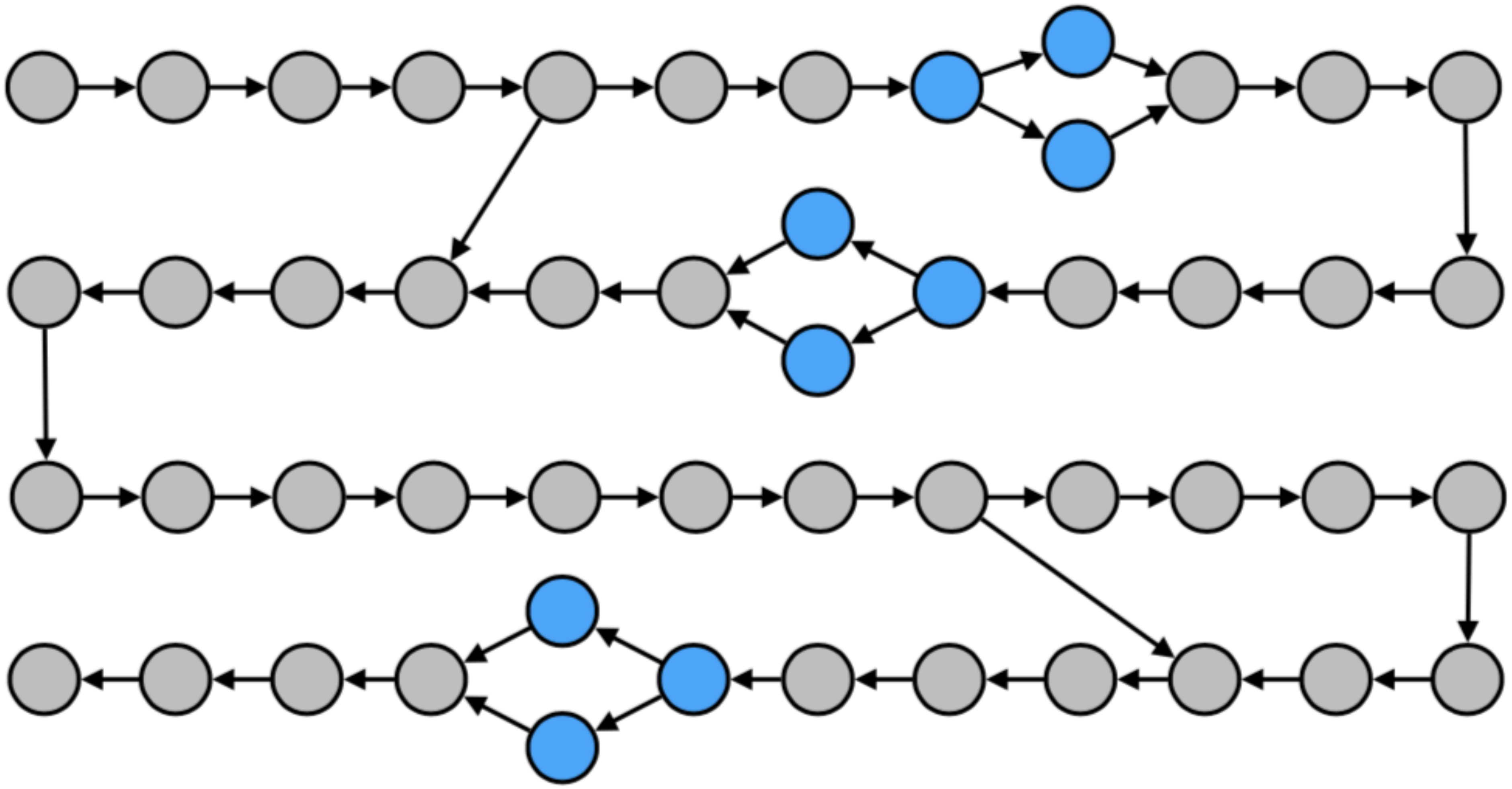
Bubble Removal

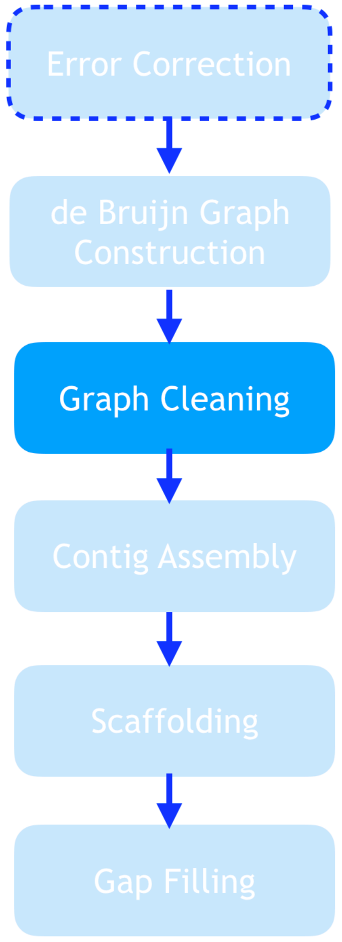




Short Read Assembly: Graph Cleaning

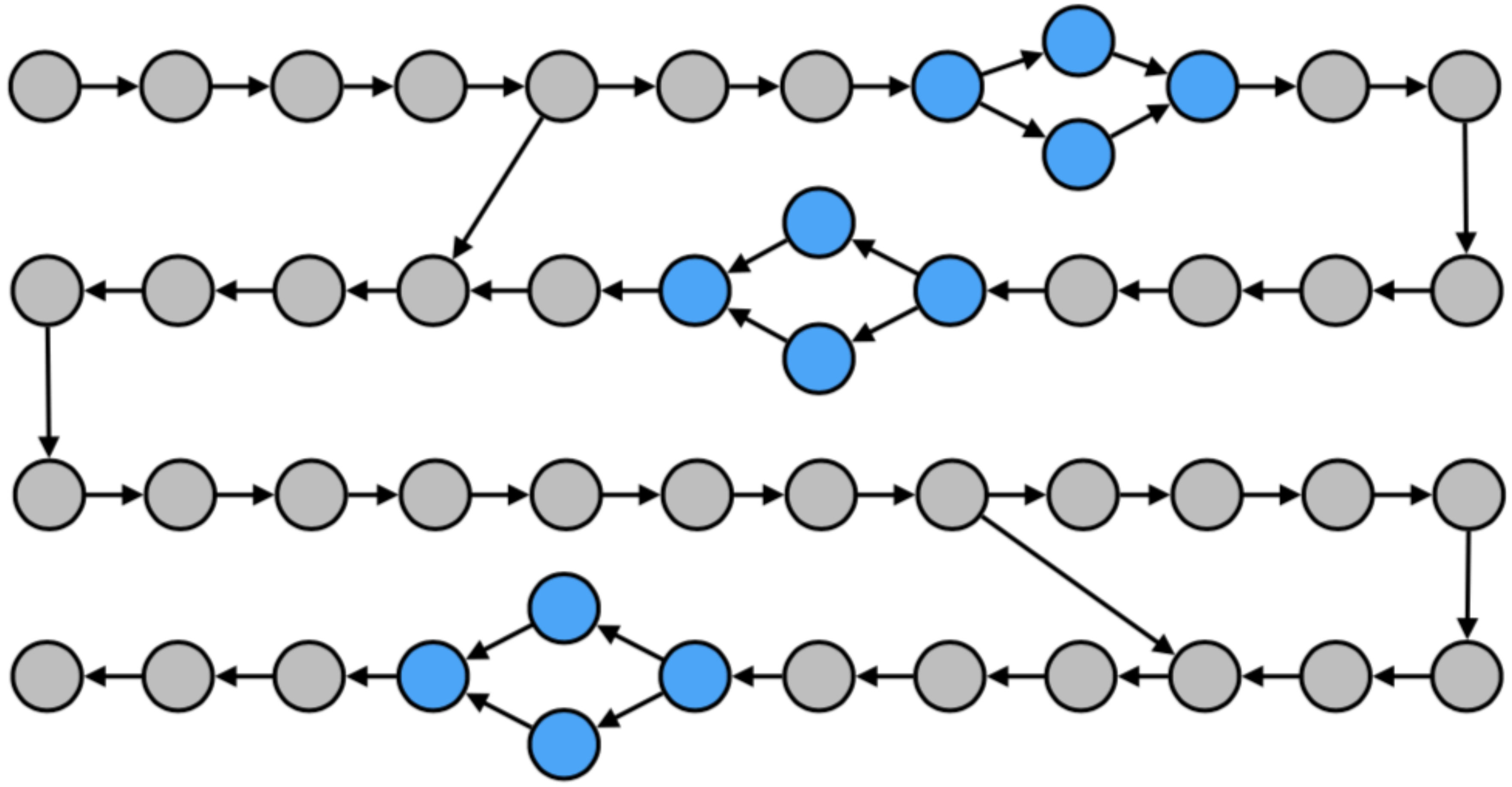
Bubble Removal

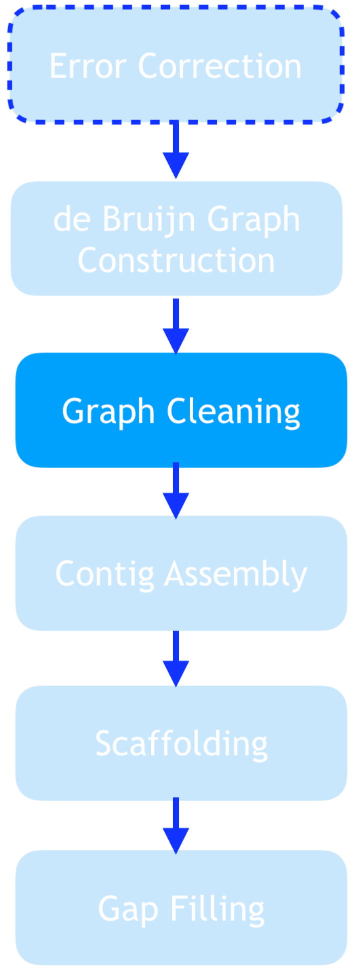




Short Read Assembly: Graph Cleaning

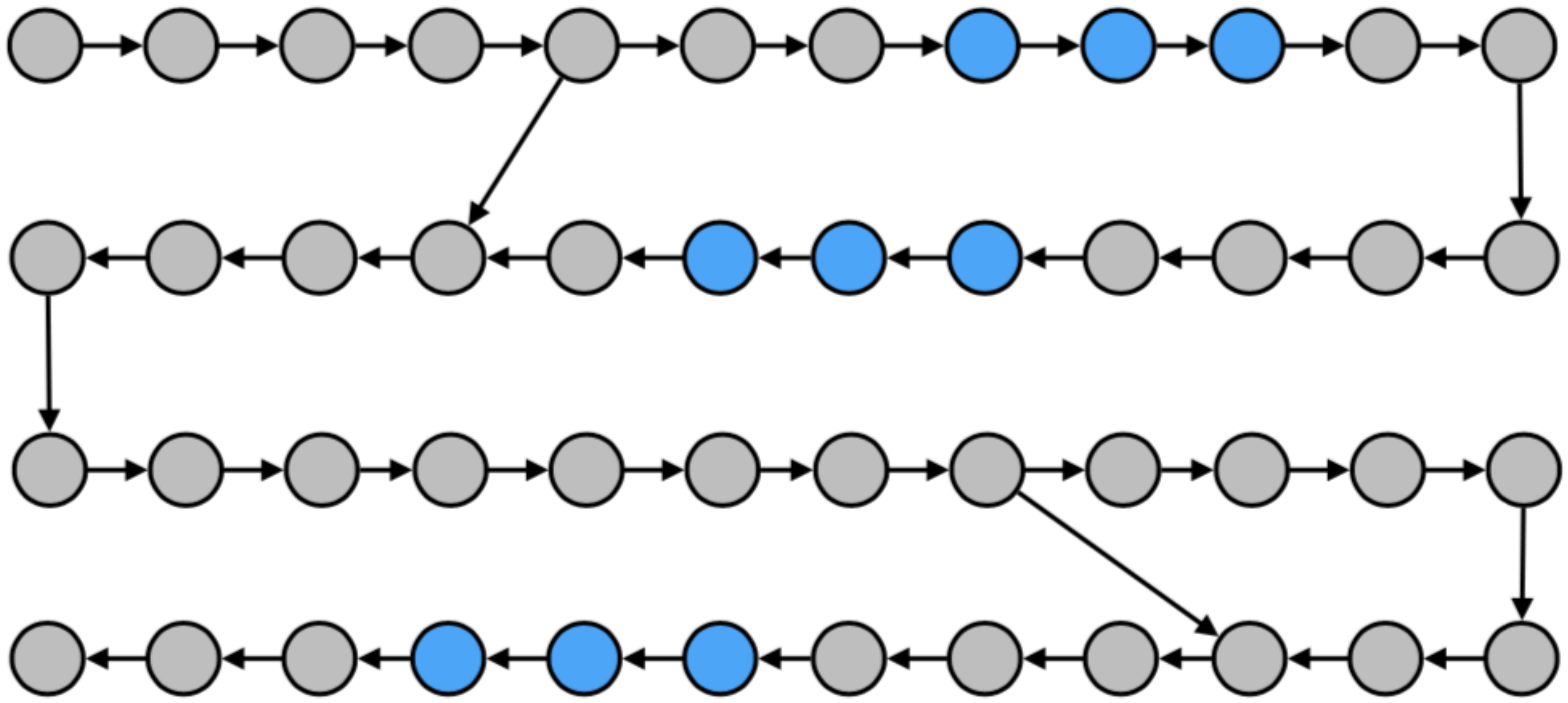
Bubble Removal





Short Read Assembly: Graph Cleaning

Bubble Removal



Error Correction

de Bruijn Graph Construction

Graph Cleaning

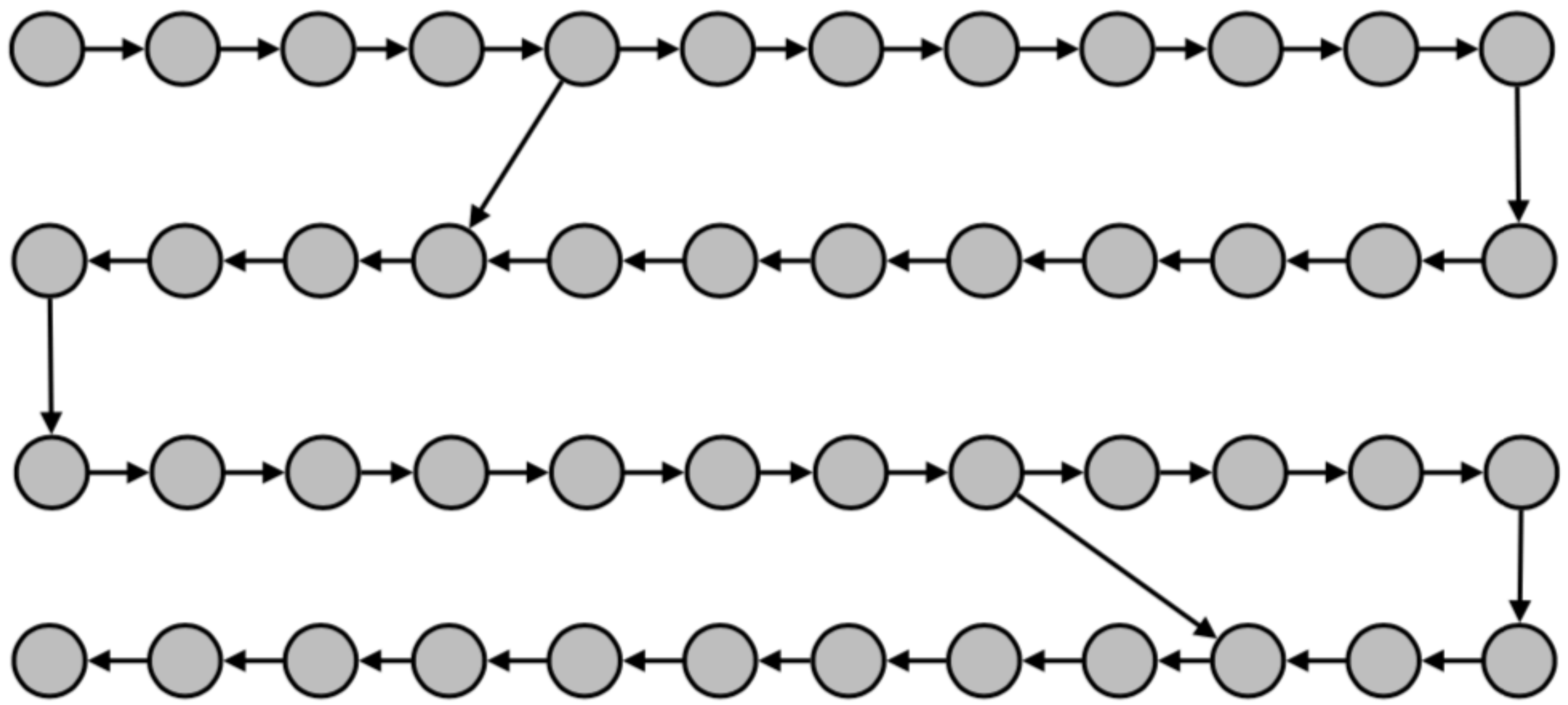
Contig Assembly

Scaffolding

Gap Filling

Short Read Assembly: Contig Assembly

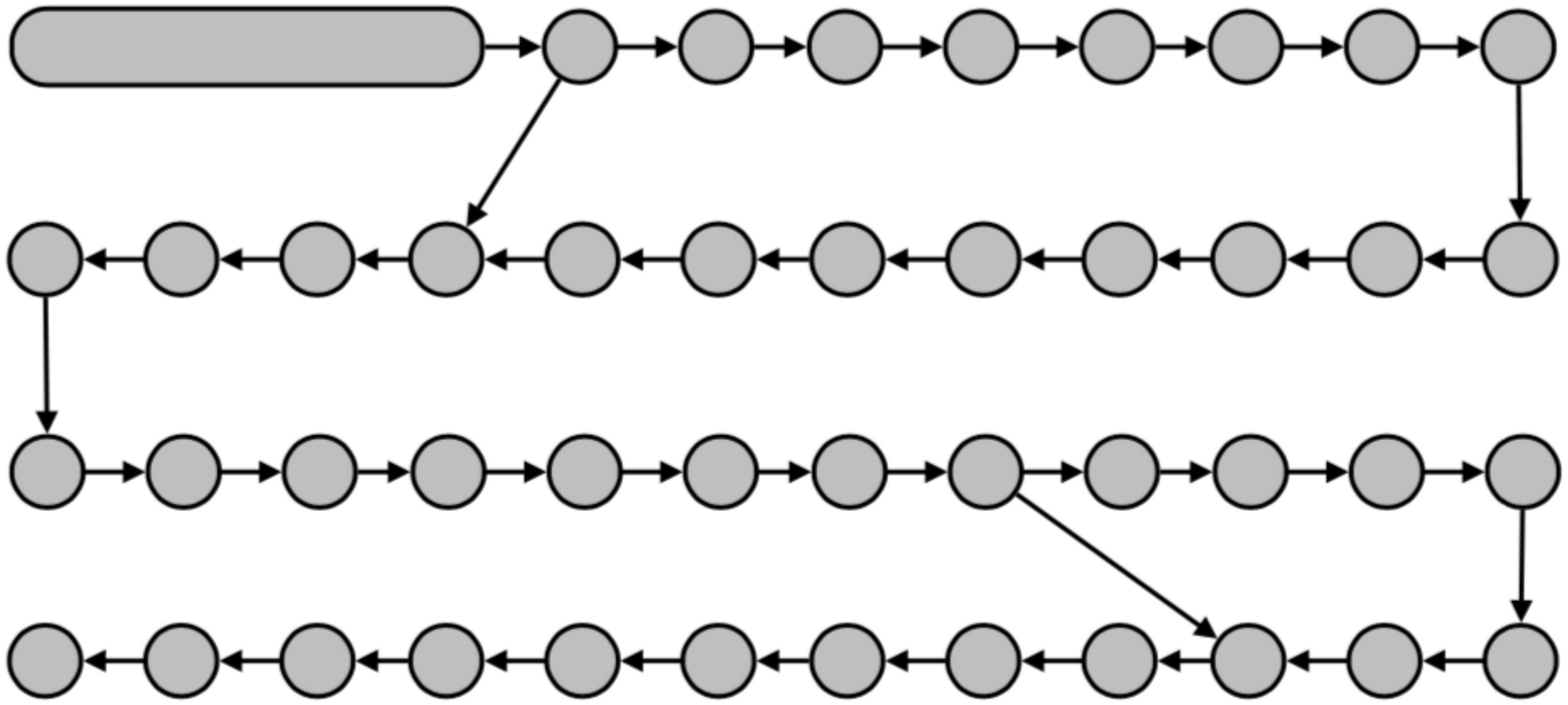
Contig Assembly

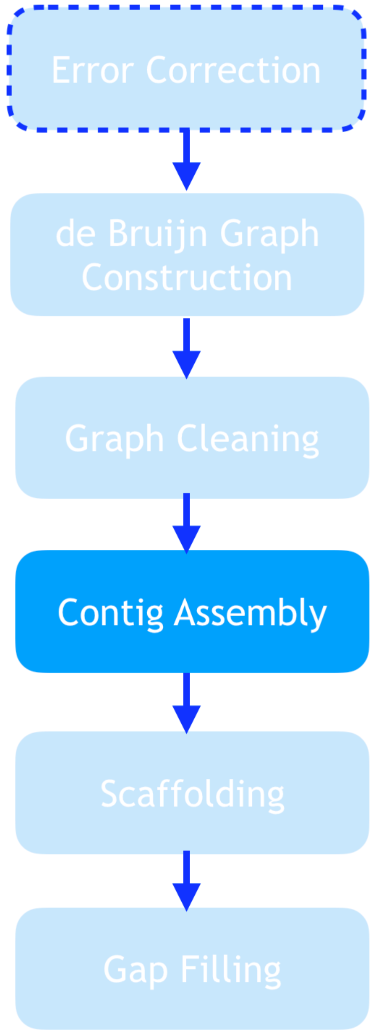


- Error Correction
- de Bruijn Graph Construction
- Graph Cleaning
- **Contig Assembly**
- Scaffolding
- Gap Filling

Short Read Assembly: Contig Assembly

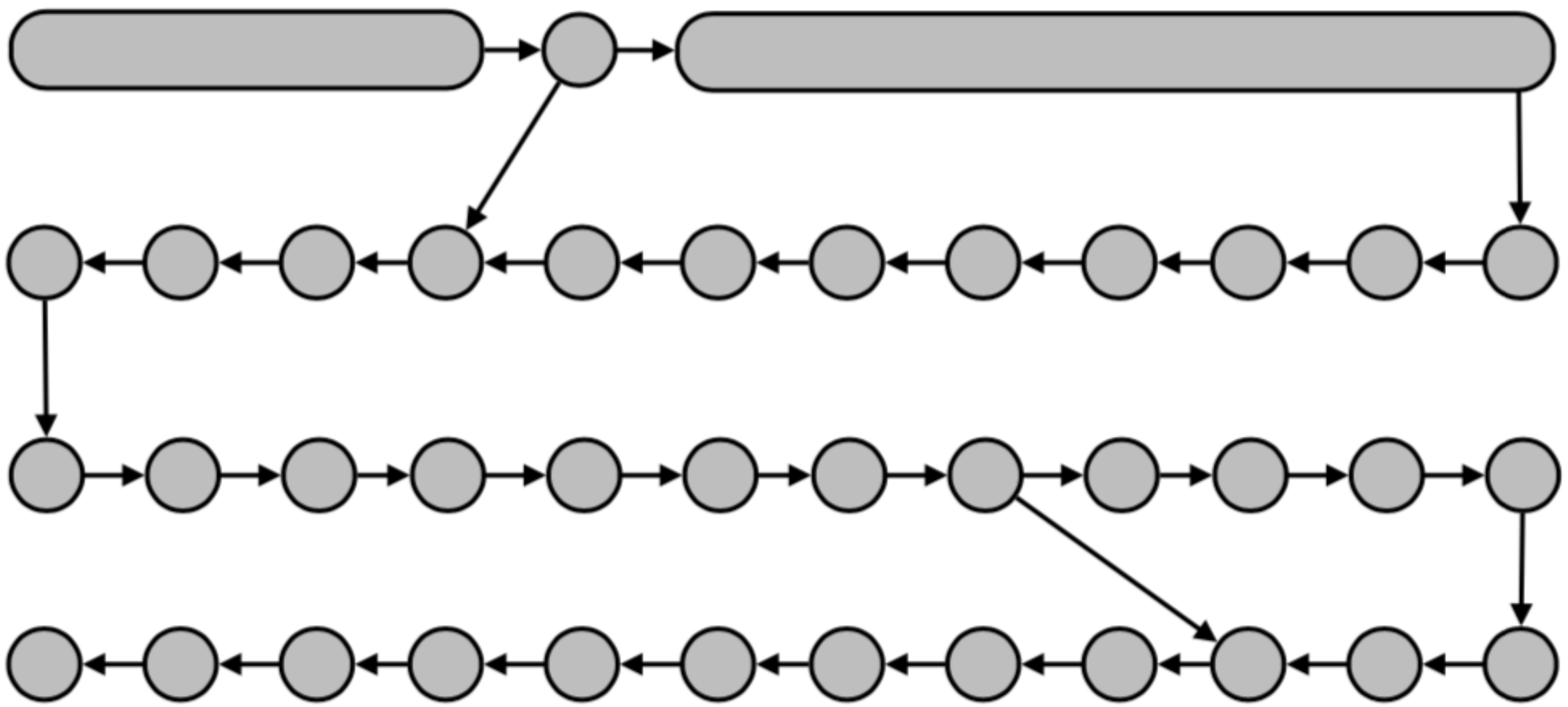
Contig Assembly





Short Read Assembly: Contig Assembly

Contig Assembly



Error Correction

de Bruijn Graph Construction

Graph Cleaning

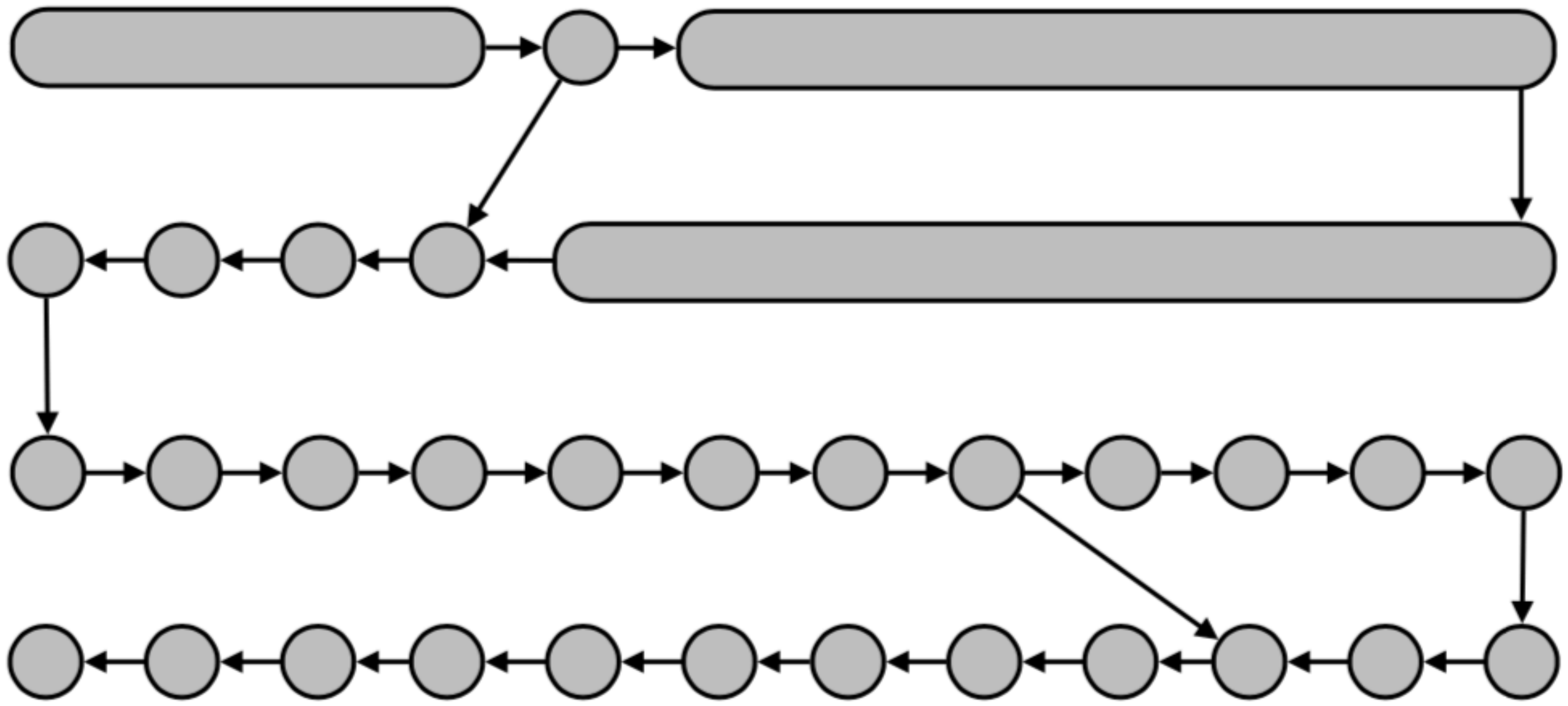
Contig Assembly

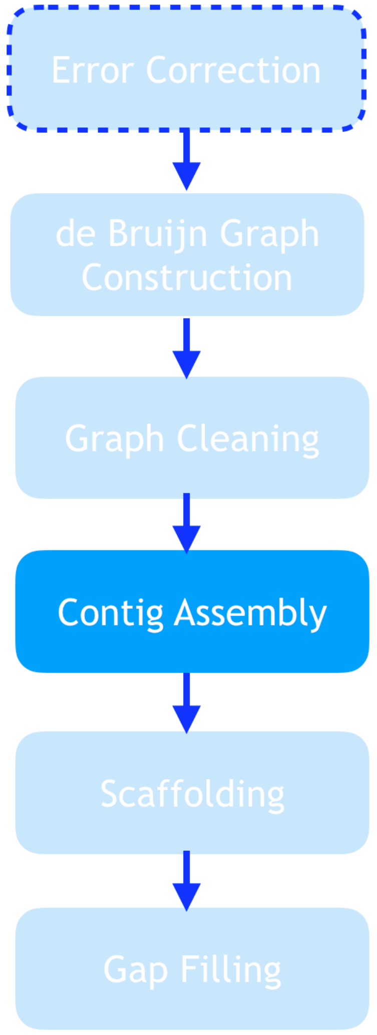
Scaffolding

Gap Filling

Short Read Assembly: Contig Assembly

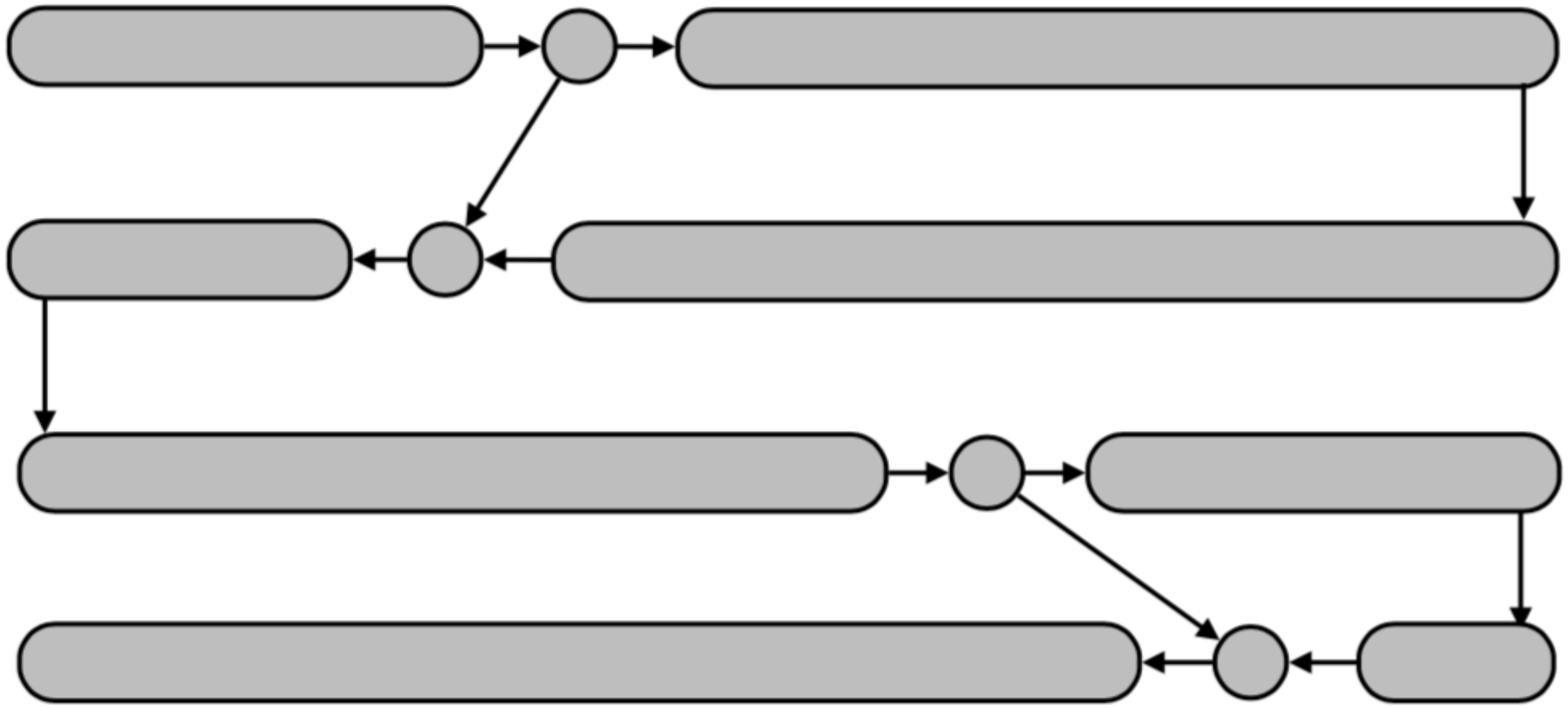
Contig Assembly

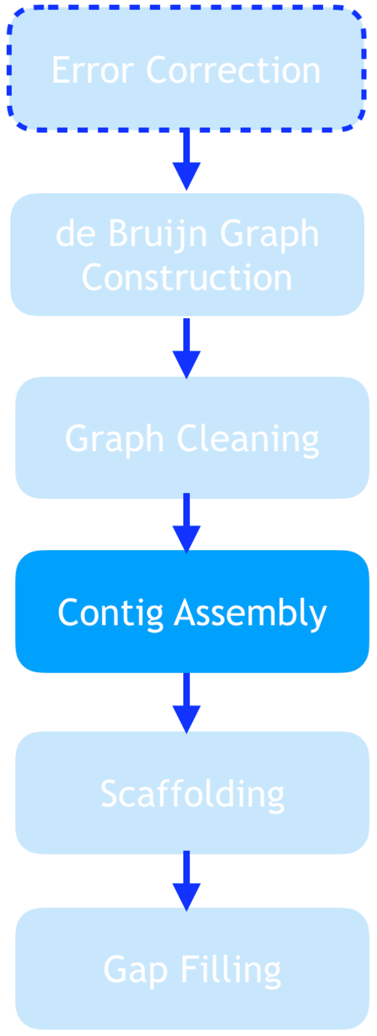




Short Read Assembly: Contig Assembly

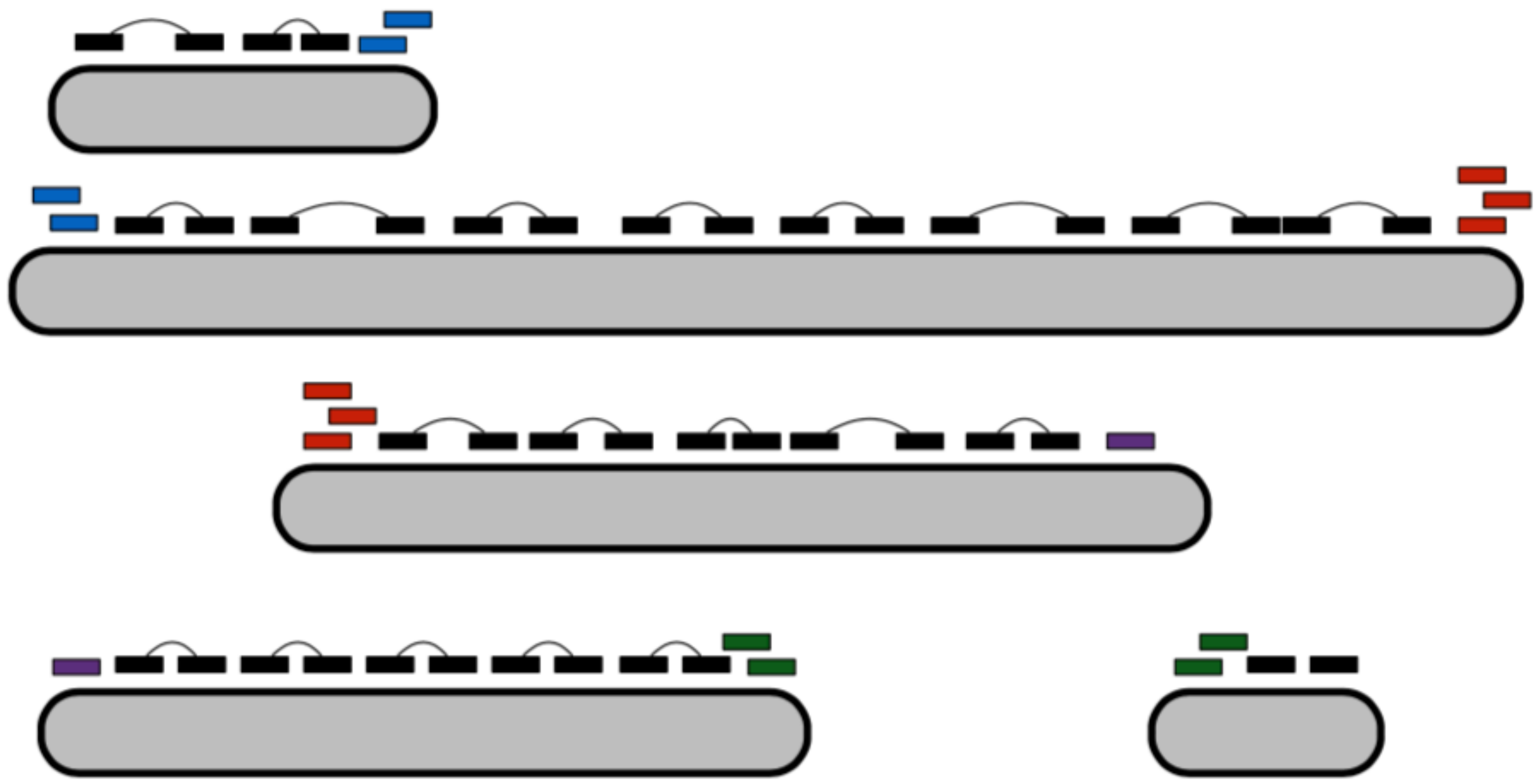
Contig Assembly

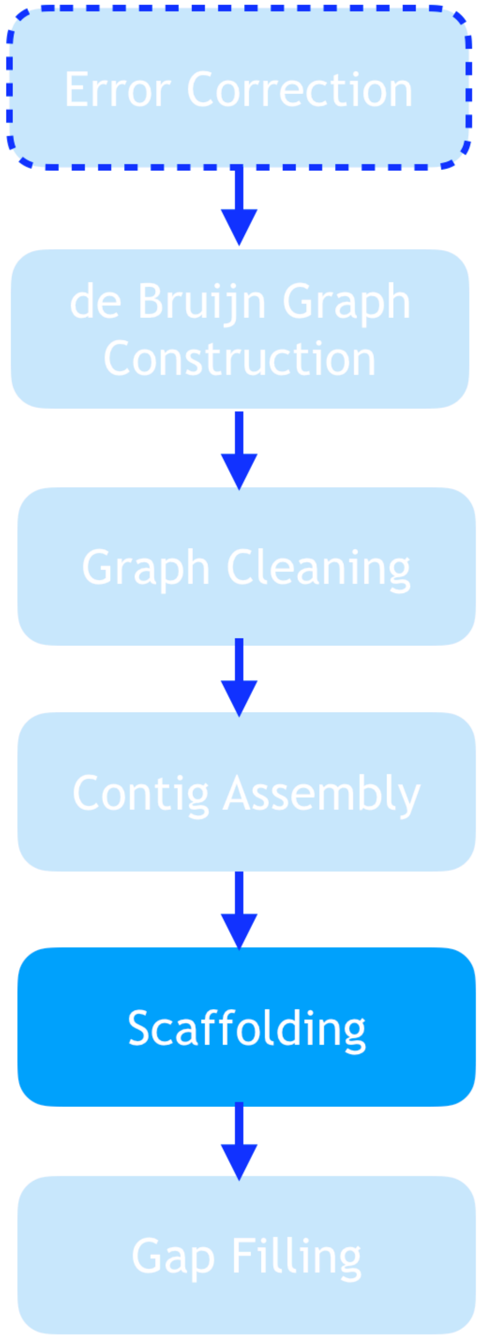




Short Read Assembly: Scaffolding

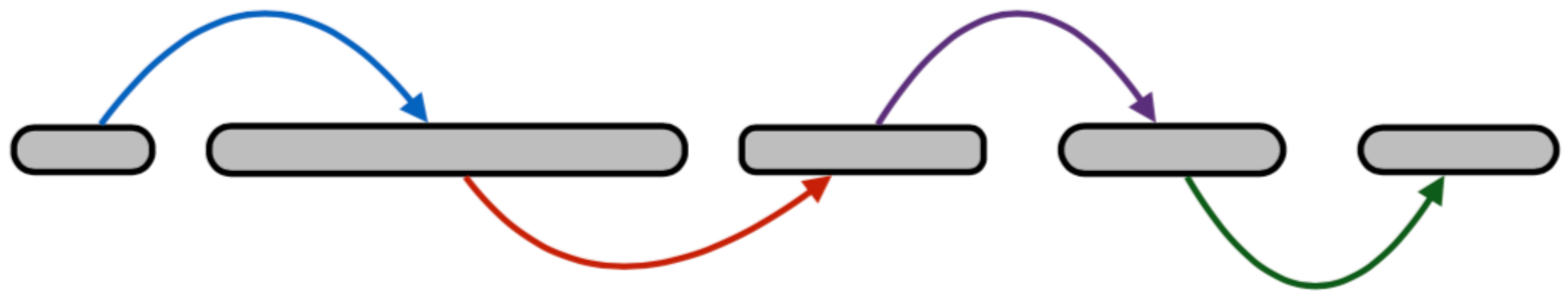
- To **scaffold**, first need to map (=place) reads to contigs

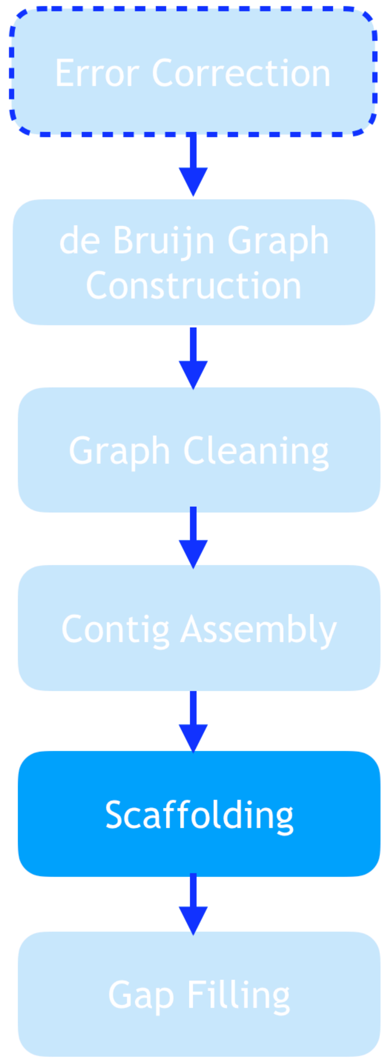




Short Read Assembly: Scaffolding

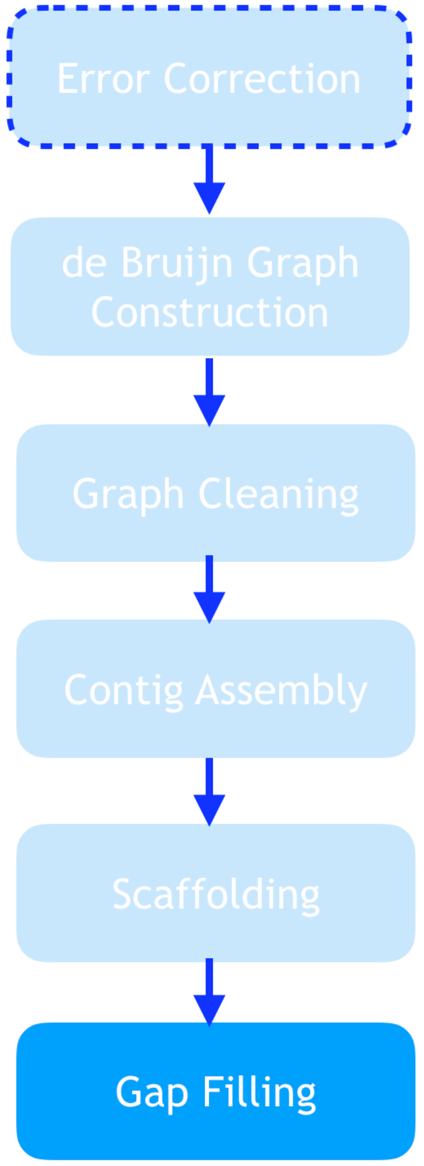
- Read pairs help to build a "scaffold graph"
- Estimate distances between contigs using fragment size distribution

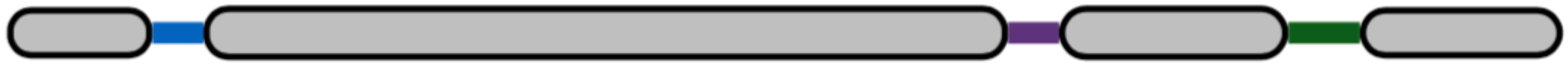




Short Read Assembly: Gap-filling

- Scaffolds will contain gaps ("NNNNNN’")
- Can use local assembly to fill these in
  - some gapfiller programs: sga gapfill, GapCloser from SOAPdenovo
- Can fill gaps using other sequencing technology (e.g. PacBio)





Assembly: What might you expect?

- Bacterial genomes:
  - Short reads: typically will get 100s of contigs (10-100 kbp average length)
  - Long reads: handful of contigs (typically 1-5), sometimes one containing the whole genome

- Long read data:
  - expensive
  - low base level accuracy
  - right technology depends on the scientific question and budget

Assembly Quality

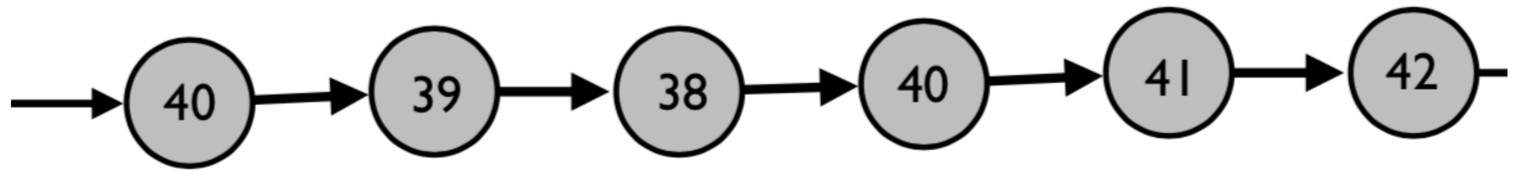
- Contiguity
- Completeness
- Accuracy

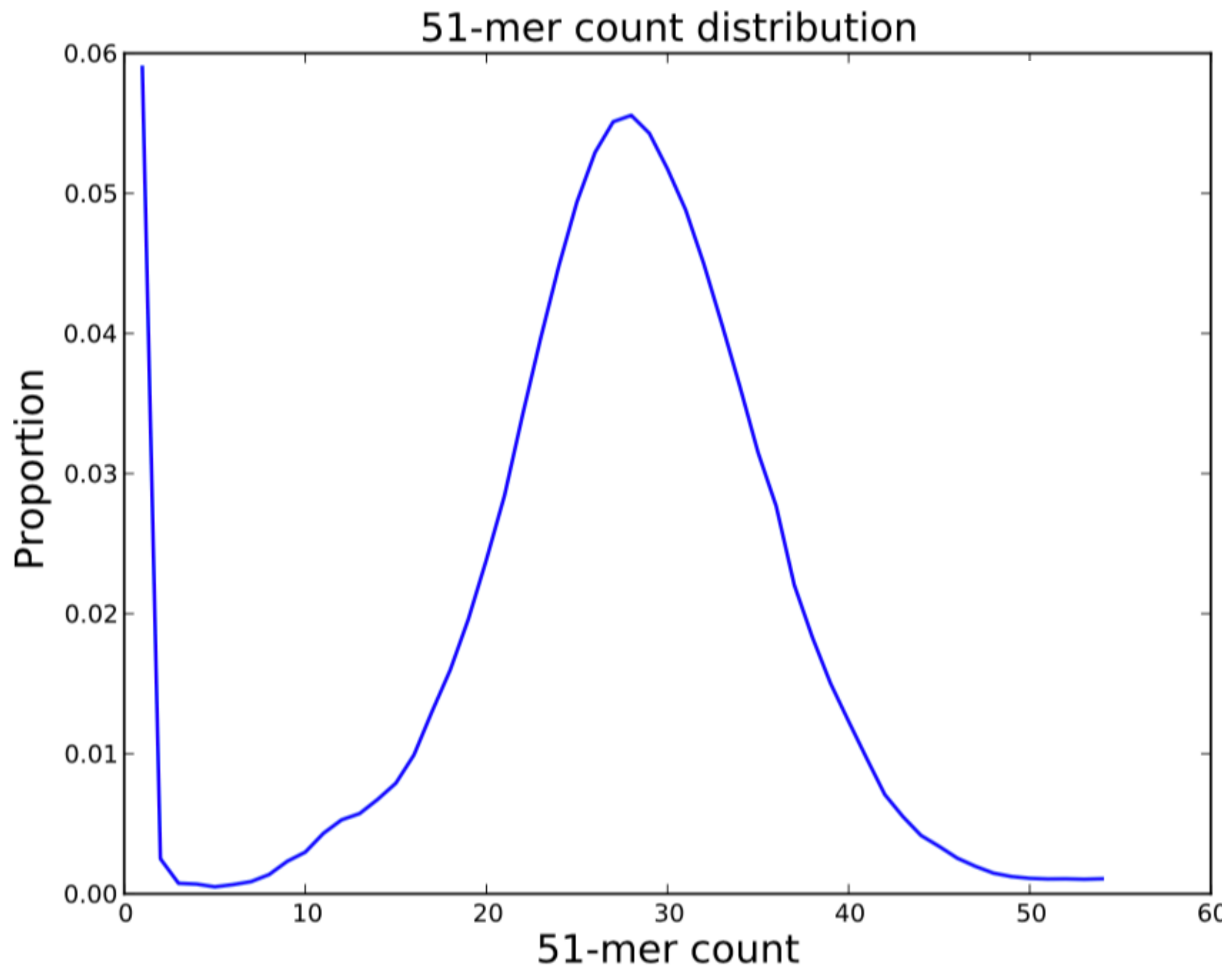
Assembly: What makes it harder?

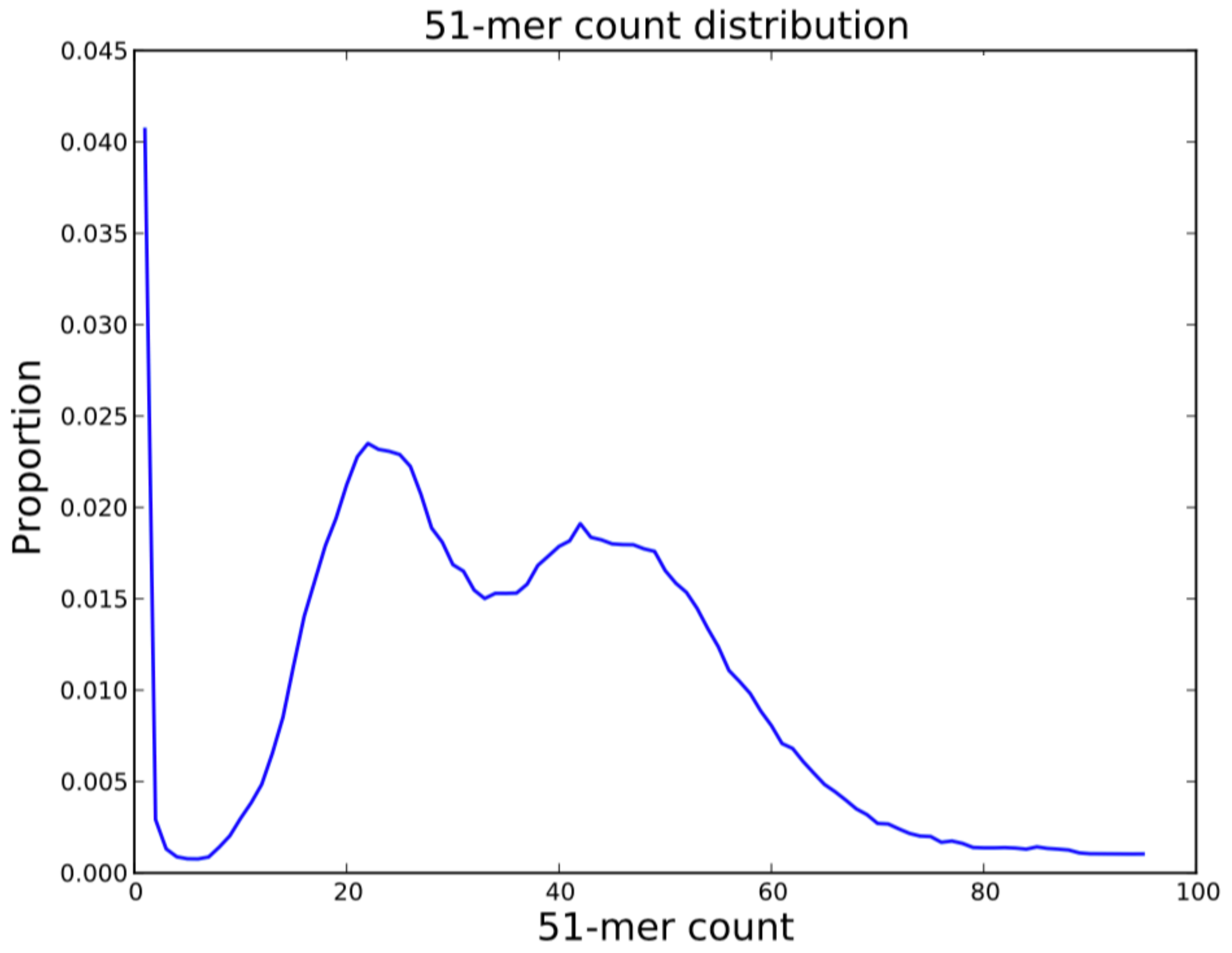
- Repetitive sequence
- Low coverage
- Biased sequencing coverage
- High error rate
- Chimeric reads
- Sequencing adapters not cleaned before input to the assembler
- Sample contamination
- Sequencing a mixed population, high heterozygosity

sga-preqc (Simpson J.): computes several useful statistics to assess quality pre-assembly-*http://github.com/jts/sga*

Assessing Assembly Difficulty: *k-mer* coverage



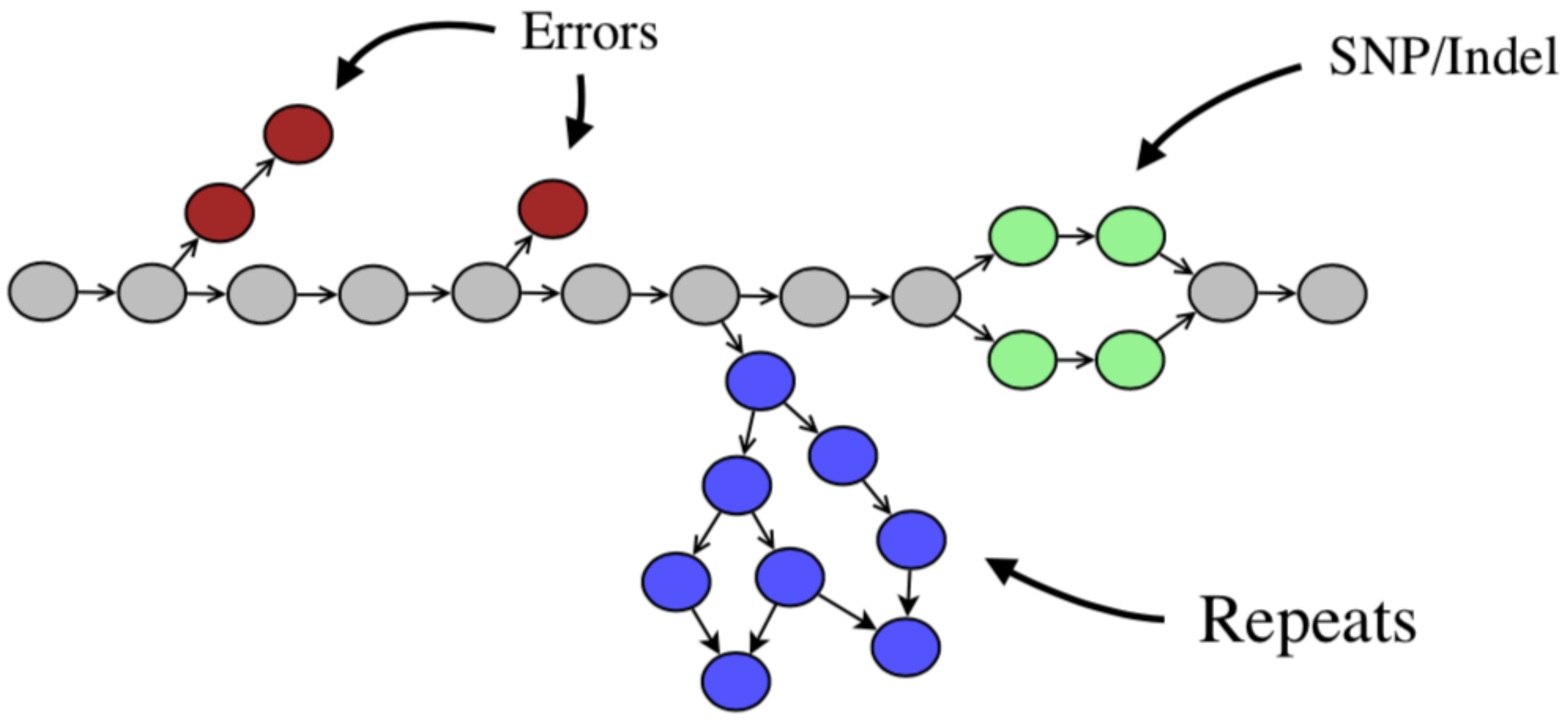


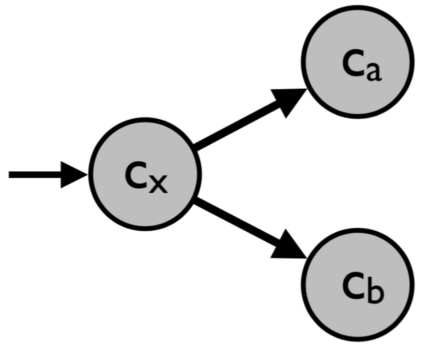


Easy: single k-mer peak, few low coverage k-mers

Hard: bimodal, might indicate contamination, mixed populations, high heterozygosity

Back to the Graph Structure

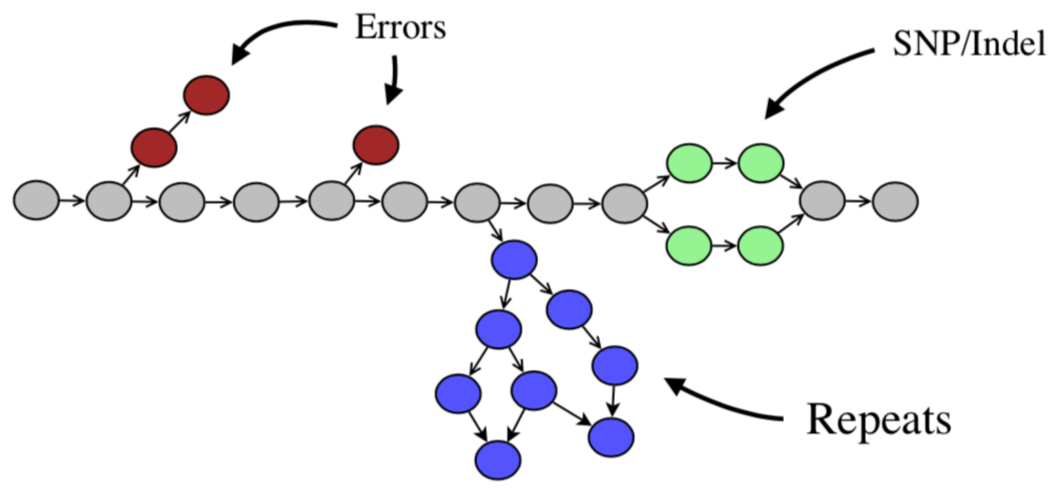


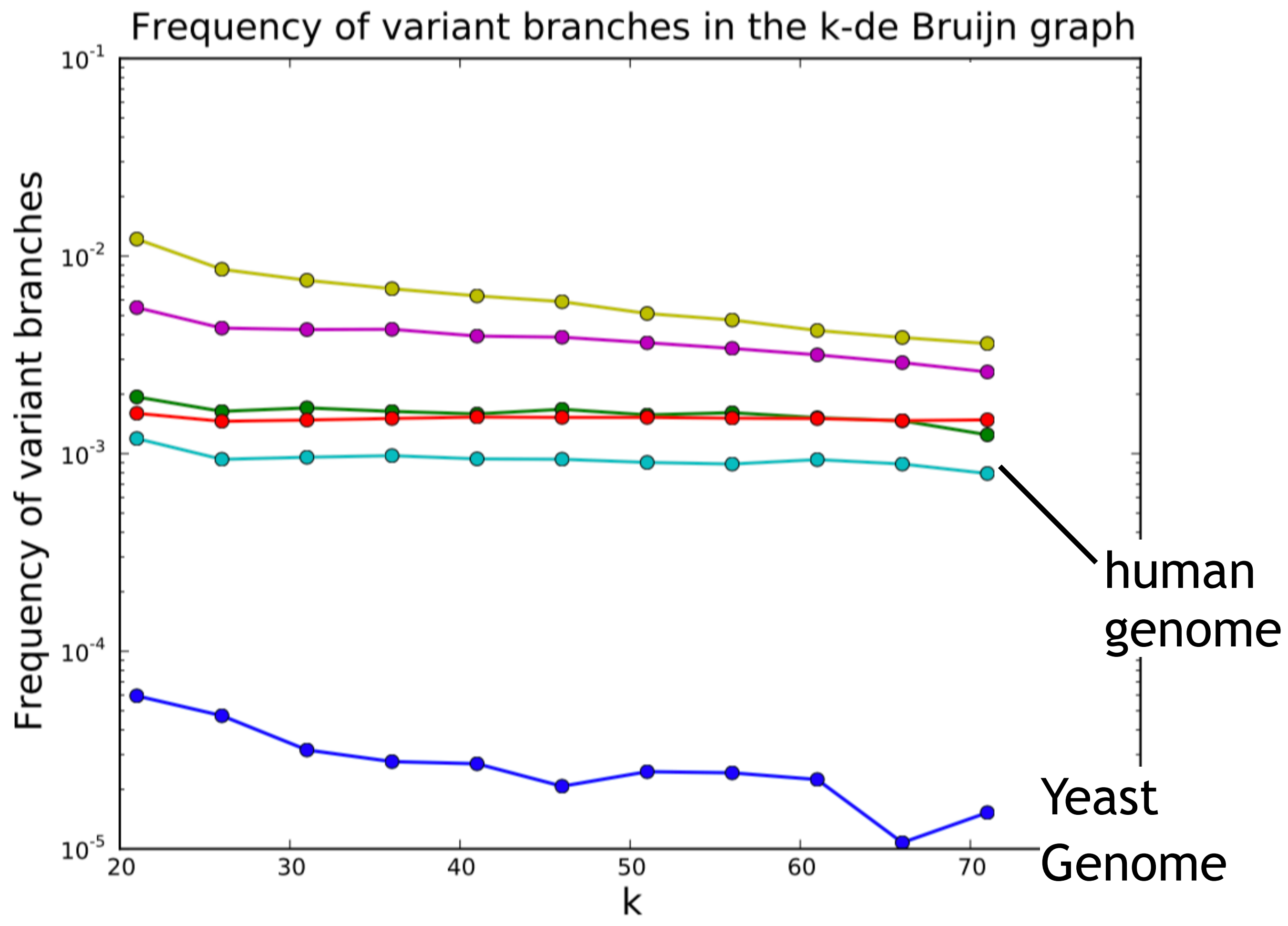


sga-preqc (Simpson J.): computes several useful statistics to assess quality pre-assembly-*http://github.com/jts/sga*

Assessing Assembly Difficulty: Variant Branch Rate

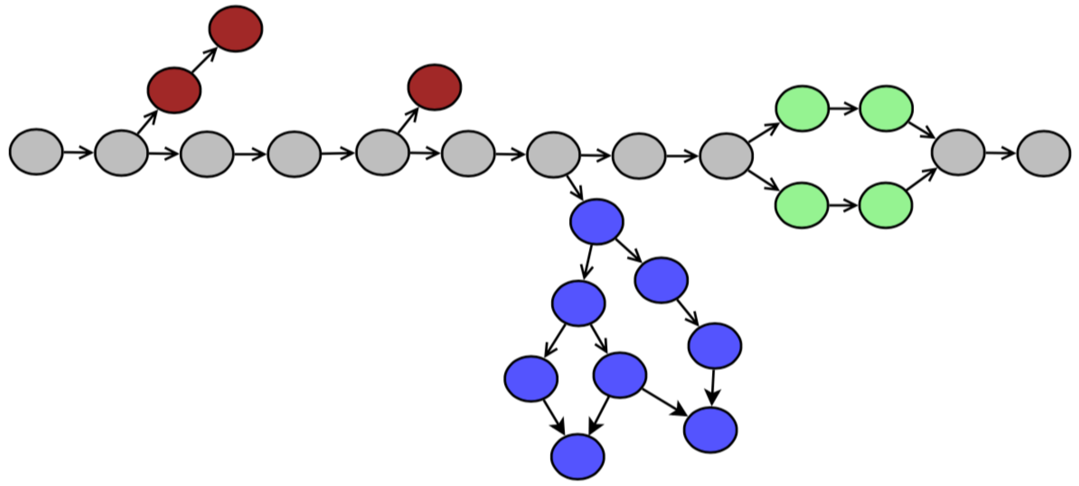
- Measure branch rates to assess assembly difficulty

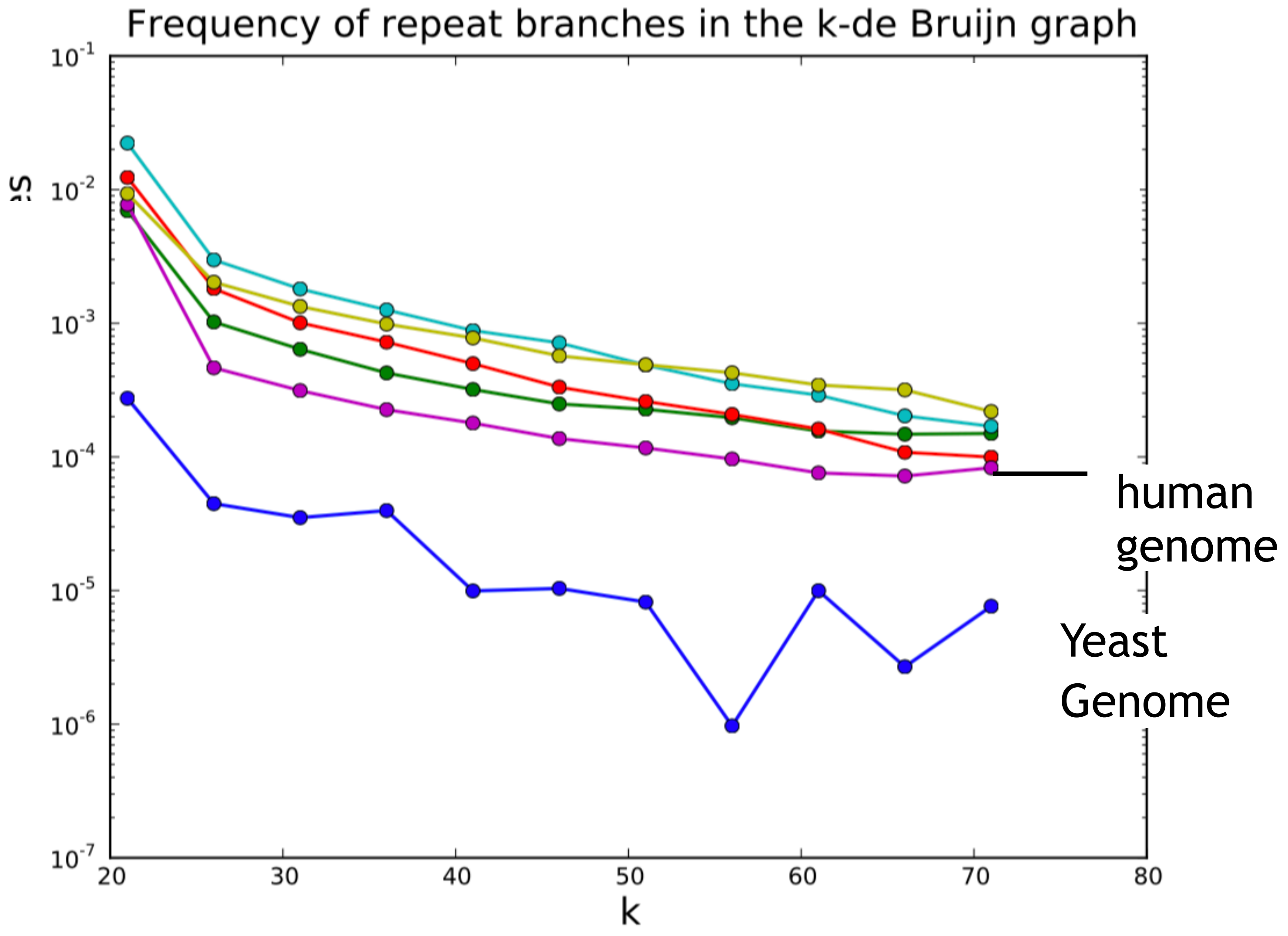




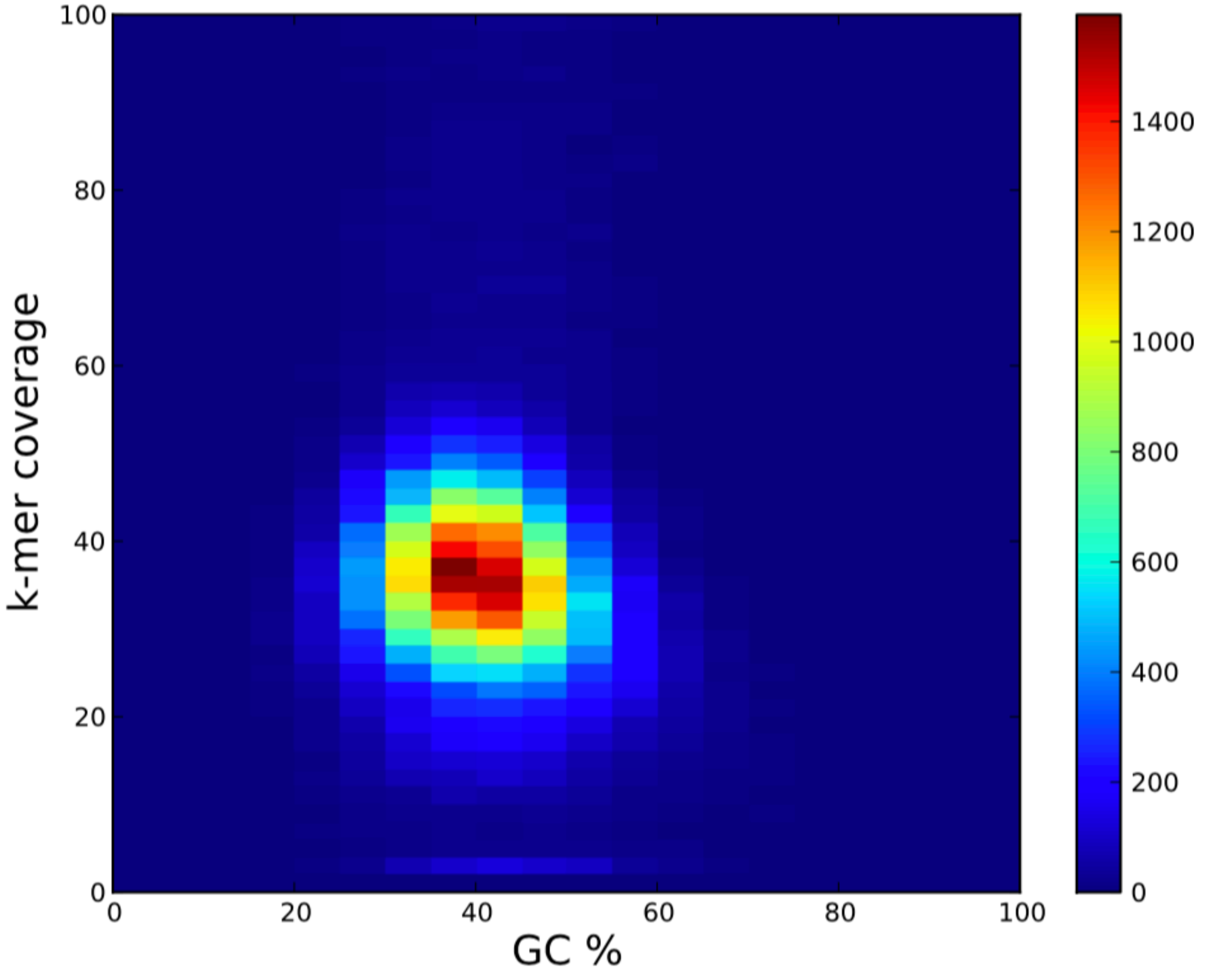
sga-preqc (Simpson J.): computes several useful statistics to assess quality pre-assembly-*http://github.com/jts/sga*

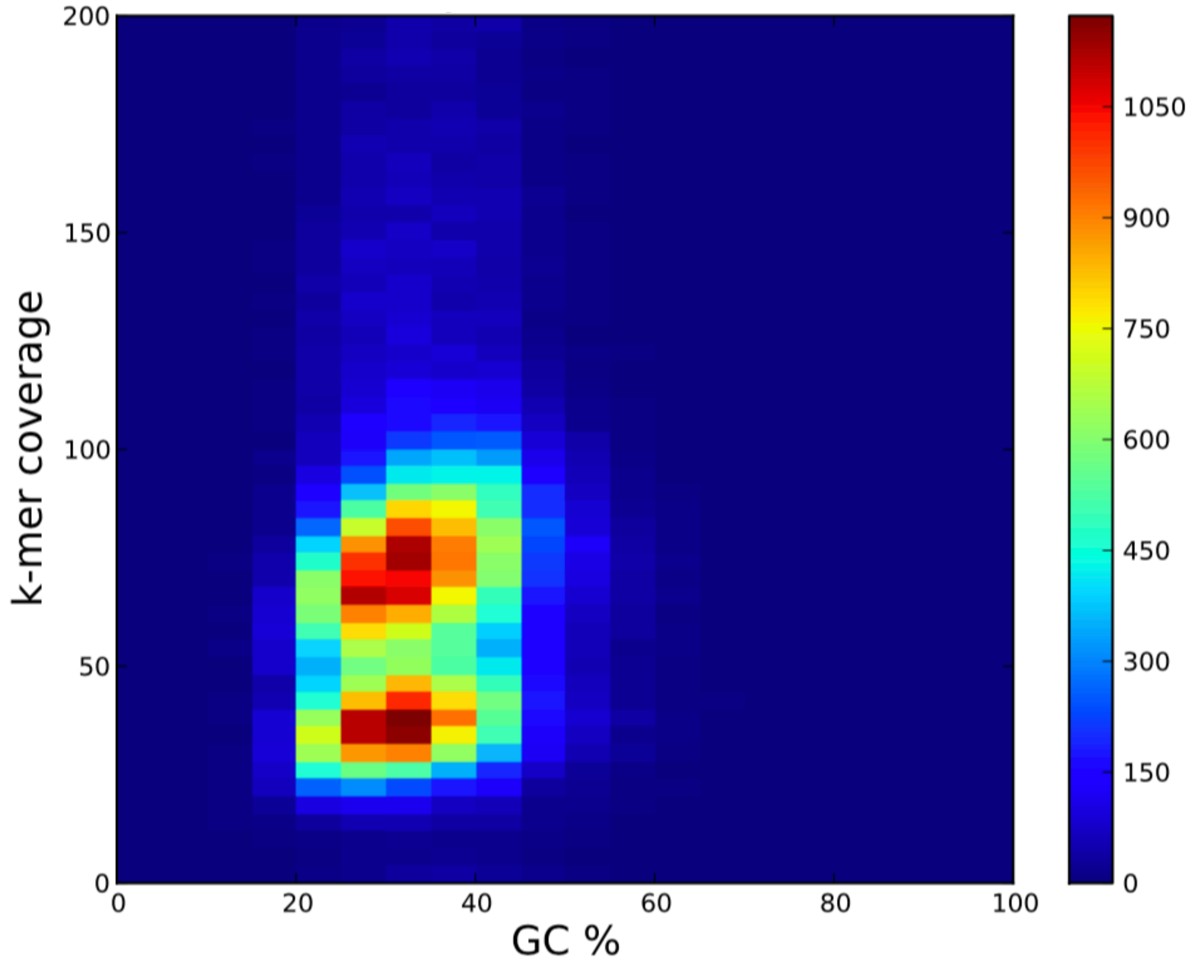
Assessing Assembly Difficulty: Repeat Branch Rate





Assessing Assembly Difficulty: GC Bias





Easier: unimodal

Harder: multimodal

Assembly for Short and Long Reads

- Long reads (PacBio/Nanopore)
    - >10 kb reads common
    - High error rate (5-15%)
    - Key challenge: computationally overcoming high error rate
- Short reads (Illumina)
    - high accuracy, high throughput (read: high coverage)
    - short read makes it hard to resolve repeats
    - Key challenge: efficiently assemble millions of short reads

Take Homes

- Assembly is a "hypothesis" about what the genome is
- Short and long read assemblers work quite differently
- Long read assembly is generally better but more expensive
- BUT: long read contigs will have lower accuracy at base level
- A variety of factors determine if an assembly is easy or hard. We can preqc these factors before assembly.