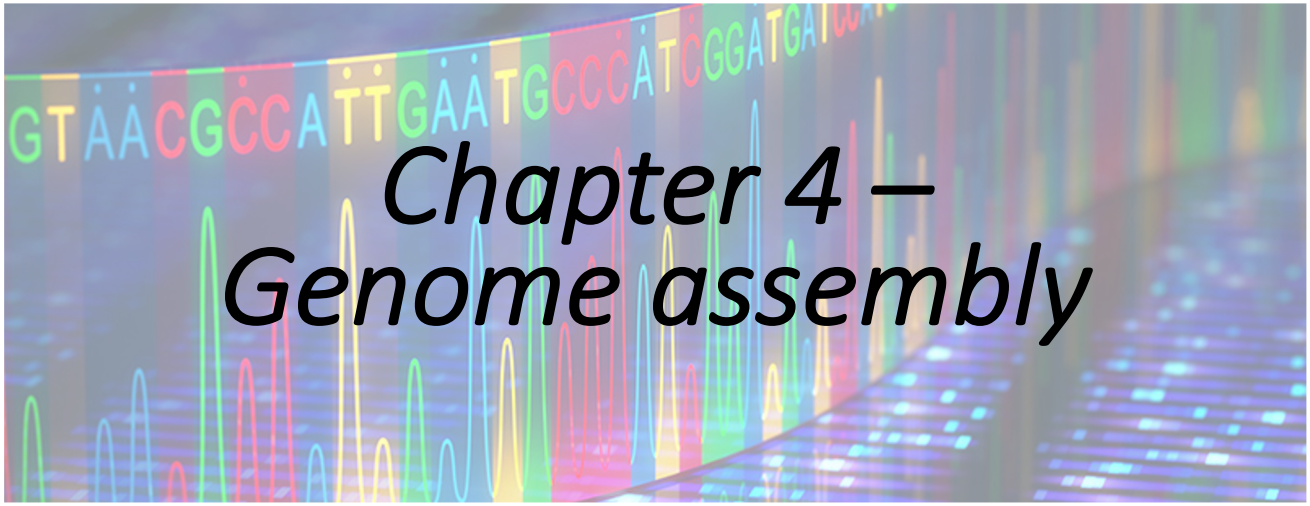
Genomics & Bioinformatics



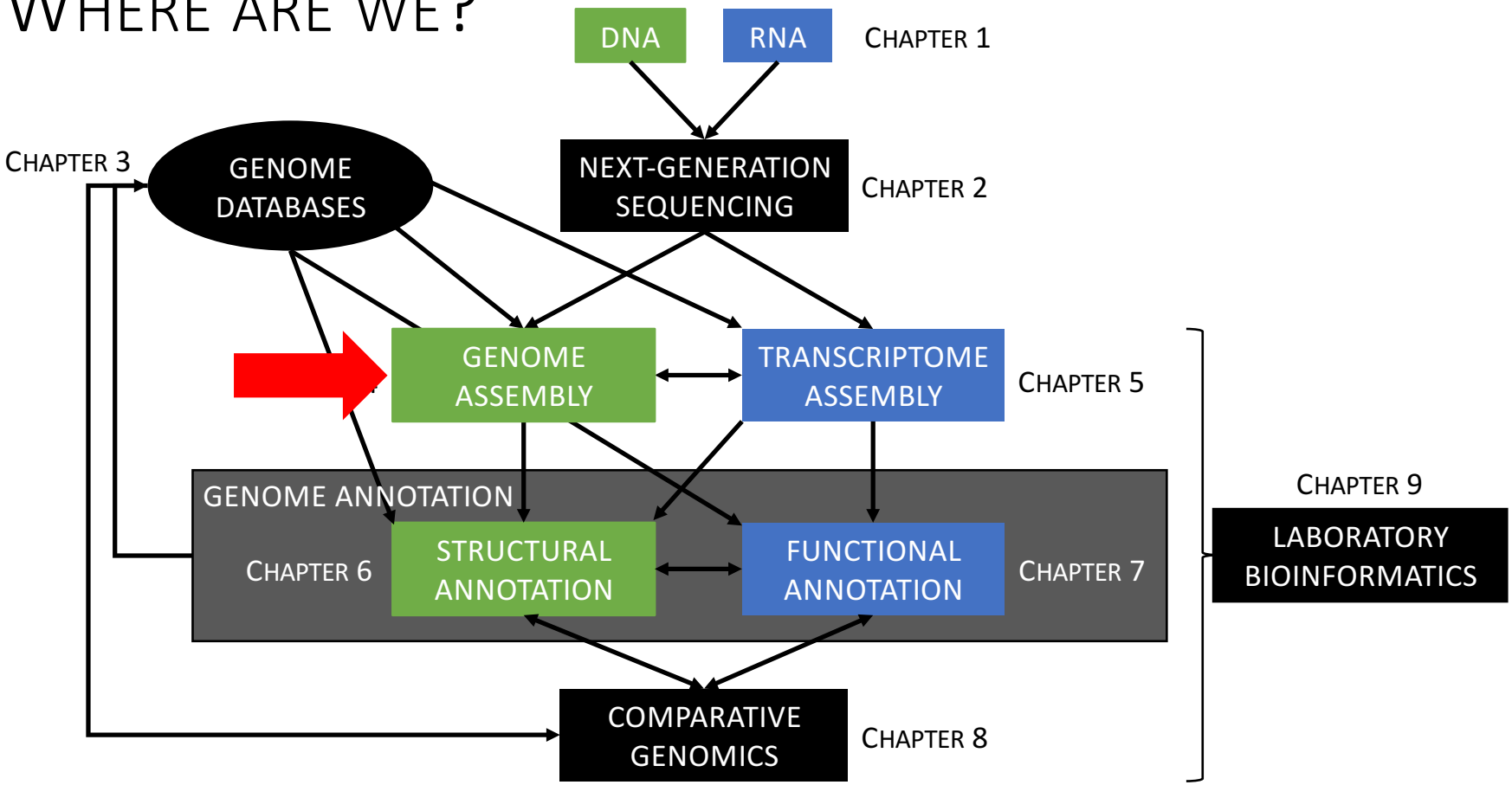
Chapter 4 – Genome assembly

BIOL 497, 597

Boise State University

Spring 2022

WHERE ARE WE?


DNA
RNA
CHAPTER 1
CHAPTER 3
GENOME DATABASES
NEXT-GENERATION SEQUENCING
CHAPTER 2
GENOME ASSEMBLY
TRANSCRIPTOME ASSEMBLY
CHAPTER 5
GENOME ANNOTATION
CHAPTER 6
STRUCTURAL ANNOTATION
FUNCTIONAL ANNOTATION
CHAPTER 7
CHAPTER 9
LABORATORY BIOINFORMATICS
COMPARATIVE GENOMICS
CHAPTER 8

OBJECTIVES

- Training students in producing a draft nuclear genome for a non-model organism using Illumina data.
- The whole-genome shotgun (WGS) sequencing dataset published by Zhang et al. (2017) on the orchid species *Apostasia shenzhenica* (2n=2x=68) is used as case-study.
- Prepared a presentation summarizing study and applied methodology.

LETTER

doi:10.1038/

Apostasia genome and the evolution of orch

Zhang[1]*, Ke-Wei Liu[1]*, Zhen Li[2,3]*, Rolf Lohaus[2,3]*, Yu-Yun Hsiao[4,5]*, Shan-Ce Niu[1,6], Jie-Yu Wang[1,7], [2,3]†, Qing Xu[1], Li-Jun Chen[1], Kouki Yoshida[8], Sumire Fujiwara[9], Zhi-Wen Wang[10], Yong-Qiang Zhang[1] Meina Wang[1], Guo-Hui Liu[1], Lorenzo Pecoraro[1], Hui-Xia Huang[1], Xin-Ju Xiao[1], Min Lin[1], Xin-Yi You-Yi Chen[4,5], Song-Bin Chang[4,5], Shingo Sakamoto[9], Masaru Ohme-Takagi[9,11], Masafumi Yagi[12], g-Yu Shen[13], Chuan-Ming Yeh[11], Yi-Bo Luo[6], Wen-Chieh Tsai[4,5,13], Yves Van de Peer[2,3,14] &

OBJECTIVES

The chapter is subdivided into three parts:

- **PART 1:** Preparing/cleaning Illumina reads for *de novo* nuclear genome assembly and inferring genome size and complexity.
- **PART 2:** *De novo* genome assembly.
- **PART 3:** Validation of draft genome.

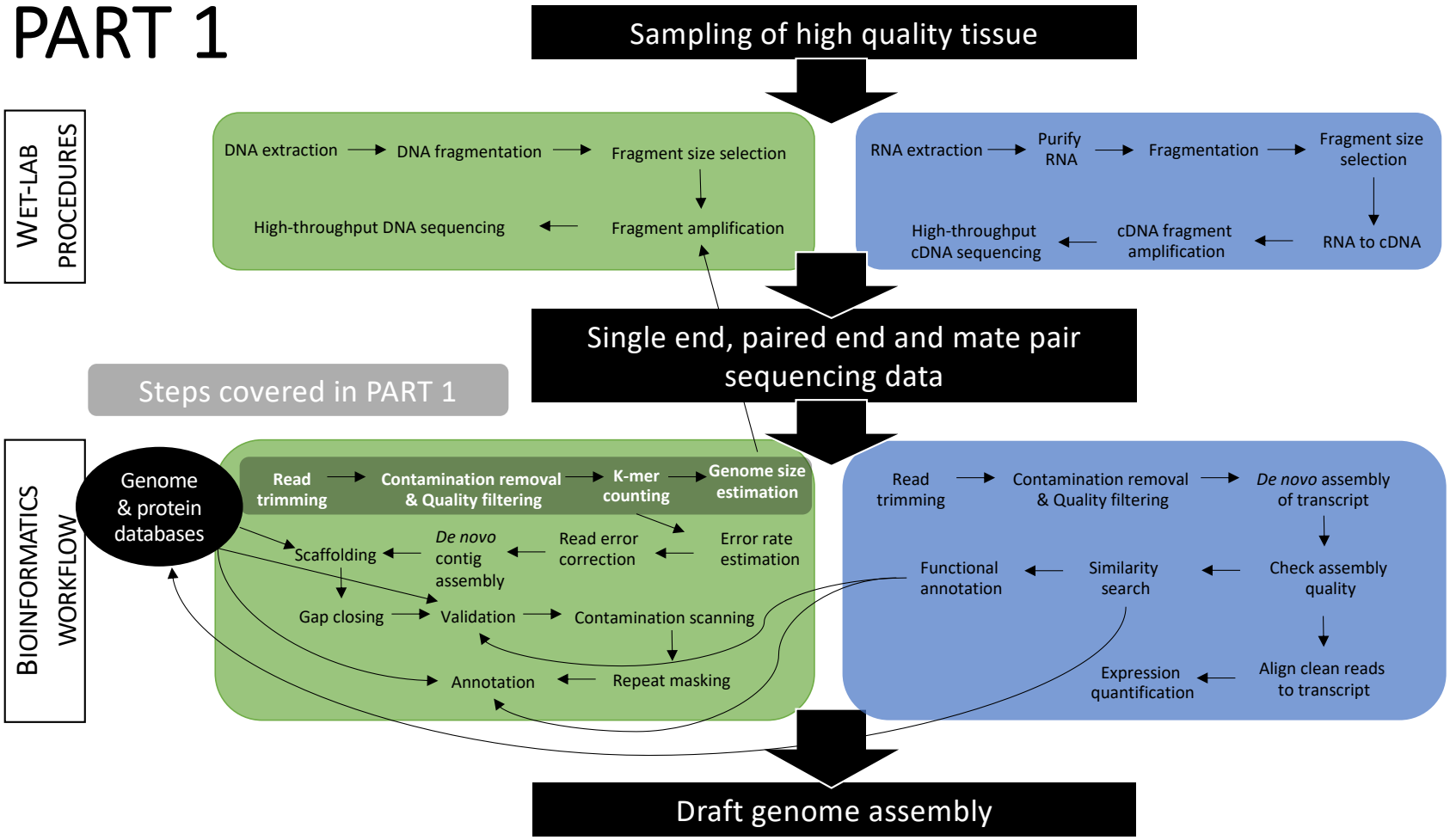
LETTER

doi:10.1038/

Apostasia genome and the evolution of orch

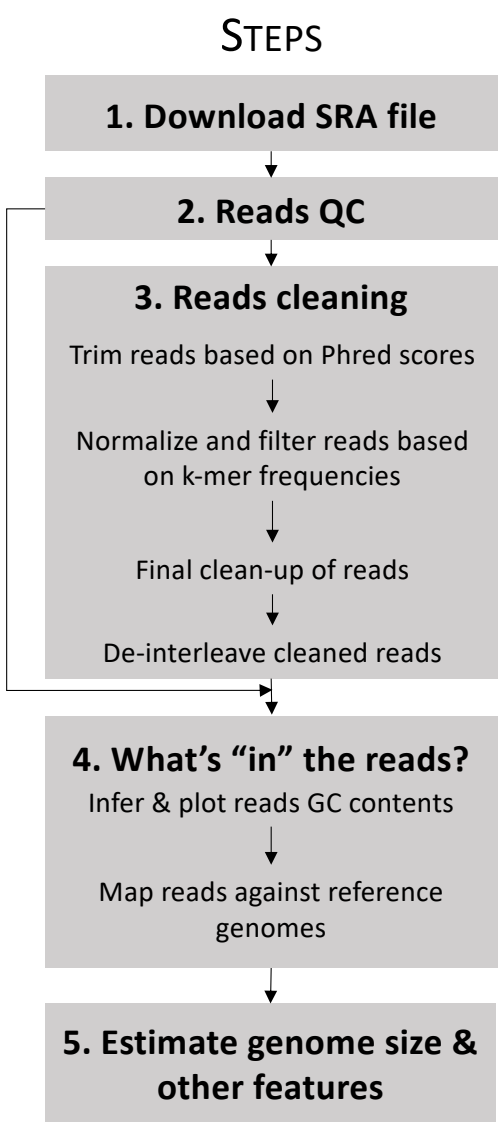
Zhang[1]*, Ke-Wei Liu[1]*, Zhen Li[2,3]*, Rolf Lohaus[2,3]*, Yu-Yun Hsiao[4,5]*, Shan-Ce Niu[1,6], Jie-Yu Wang[1,7], [2,3]†, Qing Xu[1], Li-Jun Chen[1], Kouki Yoshida[8], Sumire Fujiwara[9], Zhi-Wen Wang[10], Yong-Qiang Zhang[1] Meina Wang[1], Guo-Hui Liu[1], Lorenzo Pecoraro[1], Hui-Xia Huang[1], Xin-Ju Xiao[1], Min Lin[1], Xin-Yi You-Yi Chen[4,5], Song-Bin Chang[4,5], Shingo Sakamoto[9], Masaru Ohme-Takagi[9,11], Masafumi Yagi[12], g-Yu Shen[13], Chuan-Ming Yeh[11], Yi-Bo Luo[6], Wen-Chieh Tsai[4,5,13], Yves Van de Peer[2,3,14] &

PART 1


Sampling of high quality tissue
WET-LAB PROCEDURES
DNA extraction
DNA fragmentation
Fragment size selection
High-throughput DNA sequencing
Fragment amplification
RNA extraction
Purify RNA
Fragmentation
Fragment size selection
High-throughput cDNA sequencing
cDNA fragment amplification
RNA to cDNA
Single end, paired end and mate pair sequencing data
Steps covered in PART 1
BIOINFORMATICS WORKFLOW
Genome & protein databases
Read trimming
Contamination removal & Quality filtering
K-mer counting
Genome size estimation
Scaffolding
De novo contig assembly
Read error correction
Error rate estimation
Gap closing
Validation
Contamination scanning
Annotation
Repeat masking
Read trimming
Contamination removal & Quality filtering
De novo assembly of transcript
Functional annotation
Similarity search
Check assembly quality
Expression quantification
Align clean reads to transcript
Draft genome assembly

STEPS

5 major steps required to prepare reads for *de novo* nuclear genome assembly



STEPS	SOFTWARE	FILE FORMATS
1. Download SRA file	fastq-dump	fastq
2. Reads QC	FastQC	html
3. Reads cleaning	seqtk & khmer	fastq
Trim reads based on Phred scores	seqtk trimfq	
Normalize and filter reads based on k-mer frequencies	khmer normalize-by-median.py khmer filter-abund.py	
Final clean-up of reads	seqtk seq	
De-interleave cleaned reads	khmer split-paired-reads.py	
4. What's "in" the reads?		
Infer & plot reads GC contents	GC_content.pl & R	fasta, txt, pdf
Map reads against reference genomes	bwa, samtools, count_fastq.sh	fasta, fastq, bam, sa
5. Estimate genome size & other features	jellyfish, GenomeScope & R	fastq, histo, R, pdf

BIOINFORMATIC TOOLS & PUBLICATIONS

- Although all software are available on your Linux computers, the instructor encourages you to look at their documentations and associated papers.
- This exercise will help gaining better understanding of software' methodologies and applicability.

S1: SRA FILE

NCBI Site map | All databases | Search

Sequence Read Archive

Main | Browse | Search | Download | Submit | Software | Trace Archive | Trace Assembly | Trace BLAST

Studies | Samples | Analyses | Run Browser | Run Selector | Provisional SRA

WGS of Apostasia shenzhenica: 180 insert size (SRR5759389)

Metadata | Analysis (alpha) | Reads | Download

Run	Spots	Bases	Size	GC content	Published	Access Type
SRR5759389	84.1M	15.1Gbp	11.3G	35.5%	2017-06-27	public

This run has 2 reads per spot:

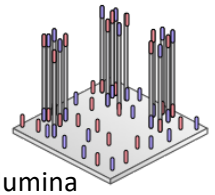
L=90, 100%	L=90, 100%

Legend

Experiment	Library Name	Platform	Strategy	Source	Selection	Layout
SRX2959224 to BLAST	Apostasia180	Illumina	WGS	GENOMIC	PCR	PAIRED

Design:

180 insert size library on Illumina



Illumina sequencing clusters (= Spots)

Key statistics

N bases = N. spots * reads length (bp)

N bases = $84.1e^6$ * 180 (90 + 90) = $15.1e^9$bp = 15.1Gbp

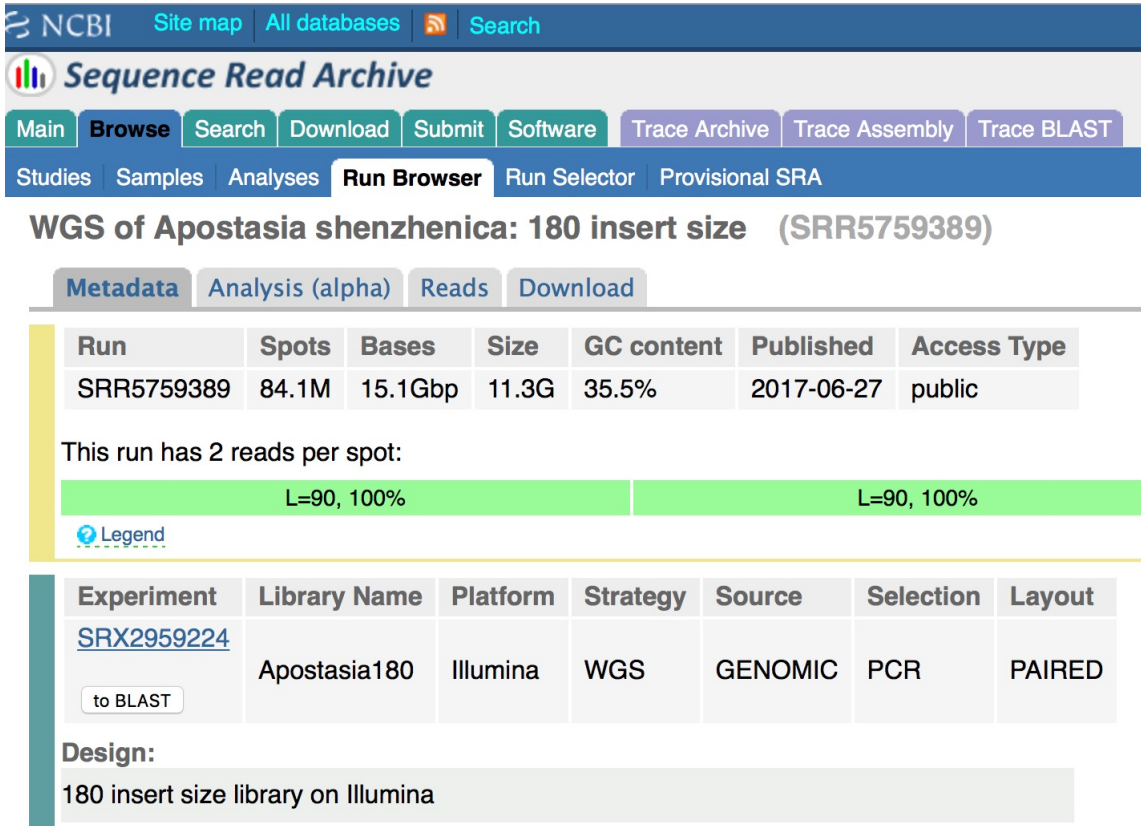
Raw genome coverage (x) = N bases / Genome size (haploid)

Raw genome coverage (x) = $15.1e^9$bp / $471.0e^6$ = 32x

This means that every bp in the genome has been sequenced 32 times.

S1: SRA FILE

- Use *fastq-dump* (implemented in the SRA Toolkit) to download WGS raw data.
- Split PE reads, but store both reads (R1 and R2) in the same file using the interleave *fastq* format.




```
svenbuerki@BIO-sven-lab:~/Documents/Kmers_analyses/SRA/output$ head SRR5759389_pe12.fastq
@SRR5759389.1./1
ACTCTTACATTCGATGTATCAGTAAAGCTGAAGTTTGAAAAGCTCTAAAGAAGATGAAATAAAAAAGCATTAGGACTGGATGACATCTGA
+
@@@DFFFFHHHGHEHHHGIGIJJGGDHGIG>GGIHGGIJIGDGIJJE4BFCDGHHIIGGIHCHCHGEEA=7@E4@??ECHFFFDFEEEED
@SRR5759389.1./2
GACAGCCTTTCAGTCTTCAAAATGACATTAAATAATCTAGTGAGCTAATAAACTTCTTCATTTTCCAAGCACTTCCAAATATCTATTCAG
+
@@@FFDDDDHHGDIIIIIIJGEHGECHEGHJJGGHIGIGHGH@FHFIGIIFHIJIHIIIGIIIJHJGIIIIGIJJJJJIIIIIHHHHHHF
```

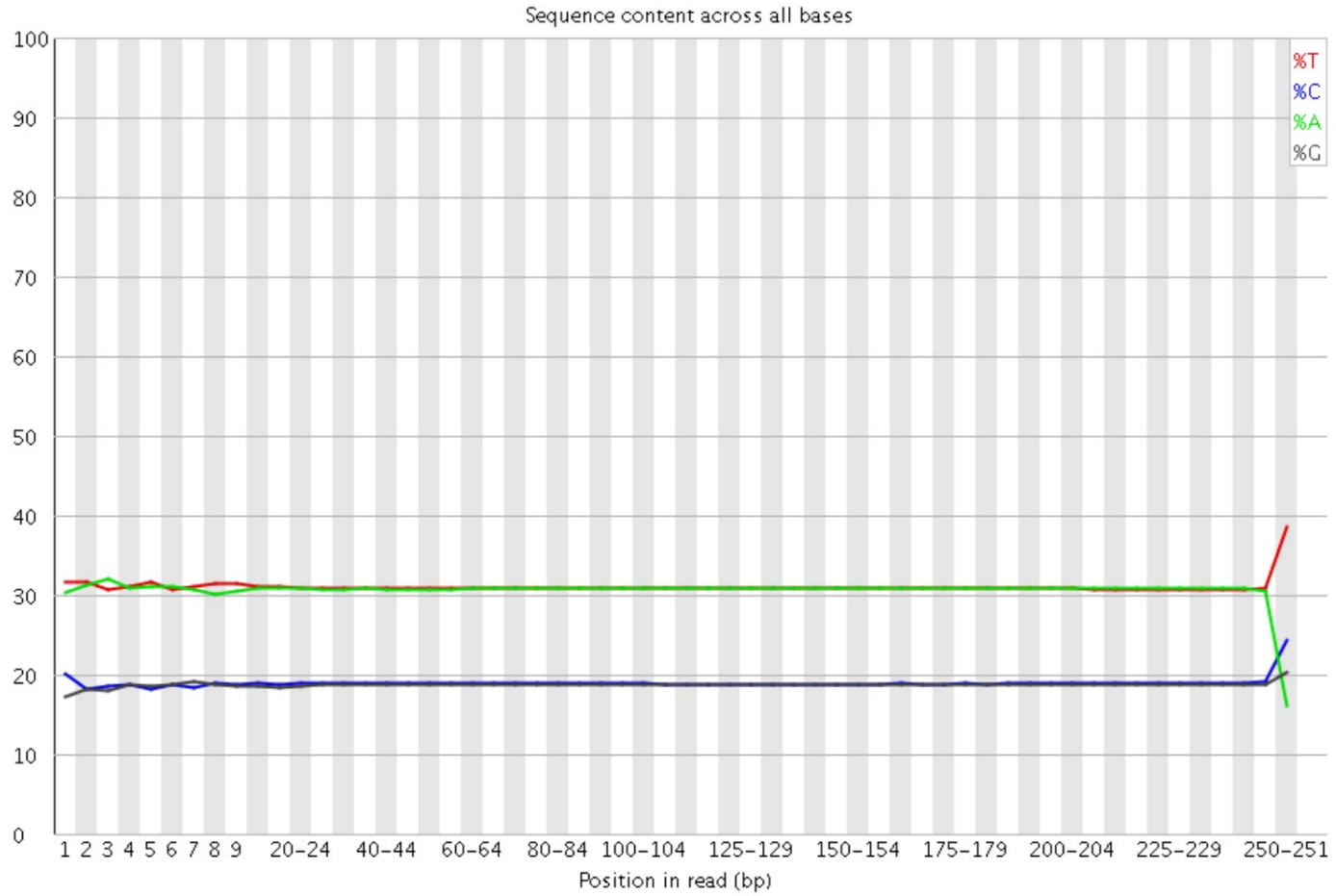
Look @FastQC data

FastQC Report S2: READS QCS

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Per base sequence content

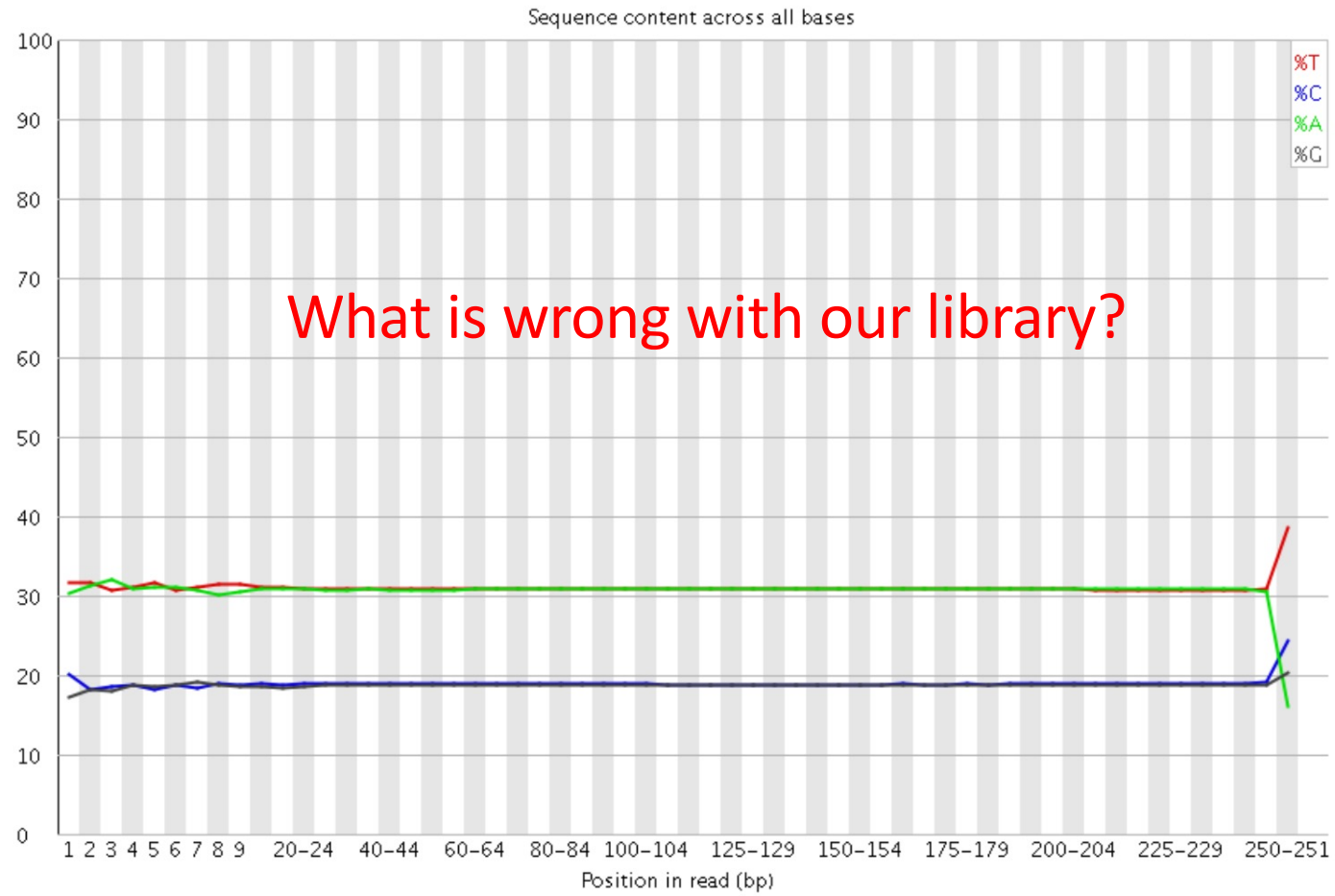


FastQC Report S2: Reads QCs

Summary

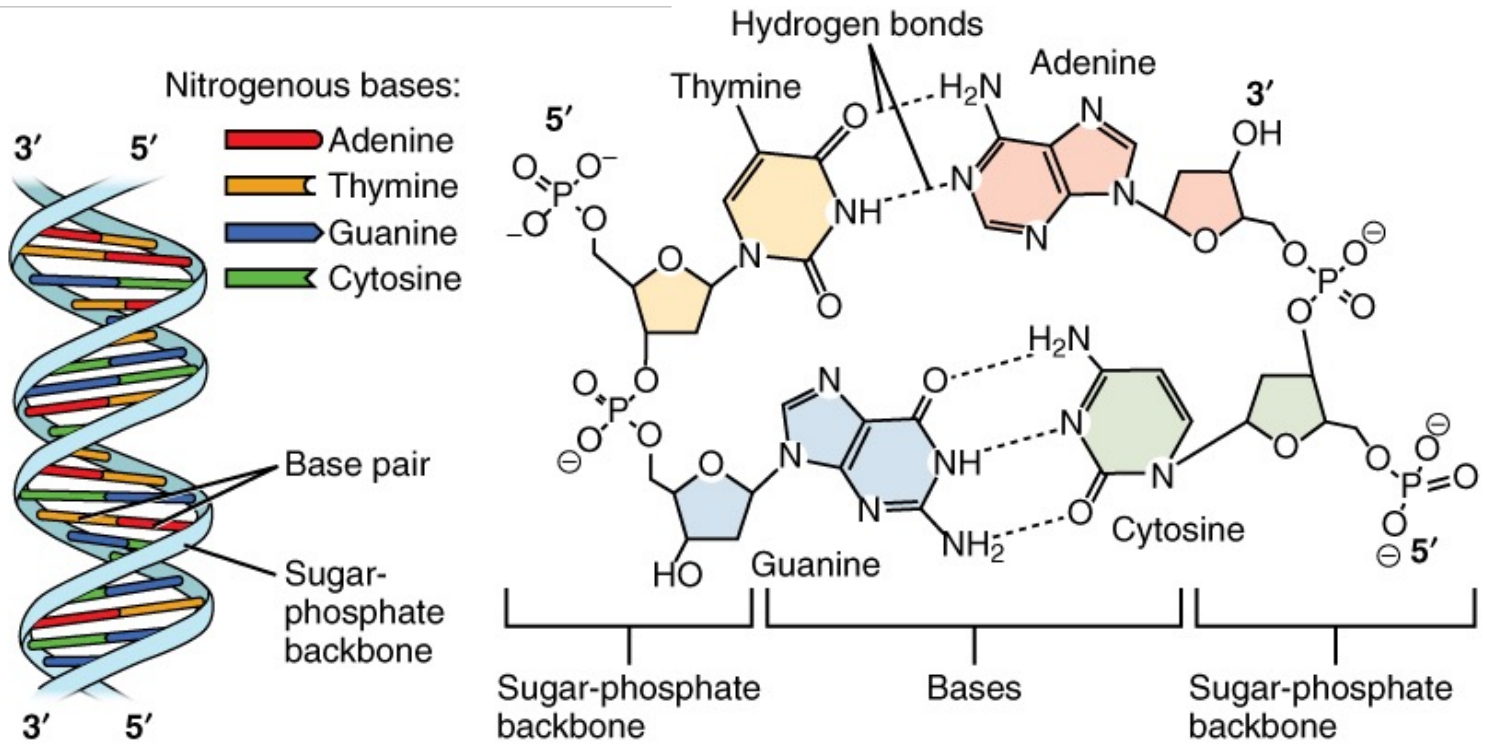
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Per base sequence content



What is wrong with our library?

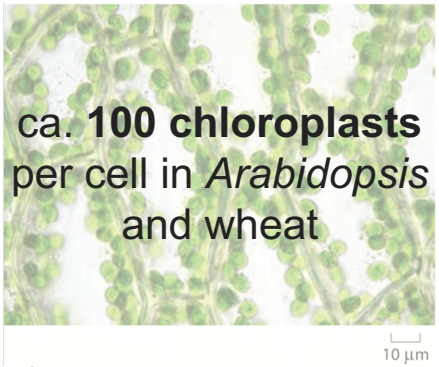
DNA – HYDROGEN BONDS

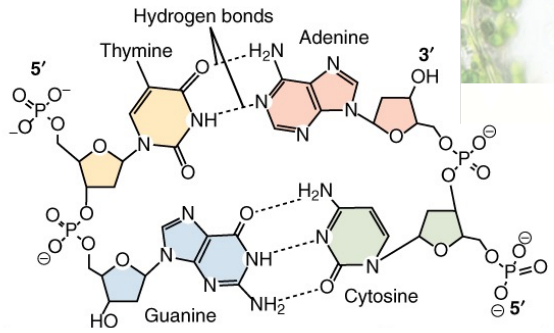


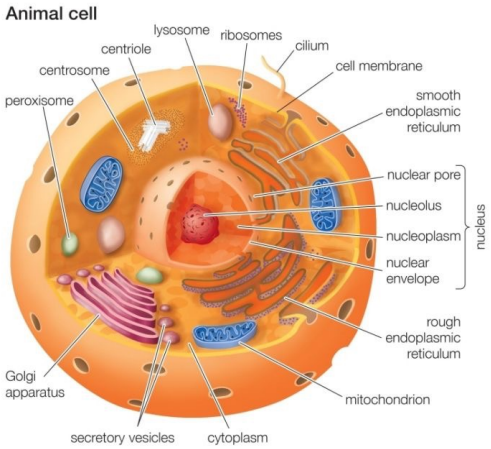
T-A: 2 hydrogen bonds
G-C: 3 hydrogen bonds

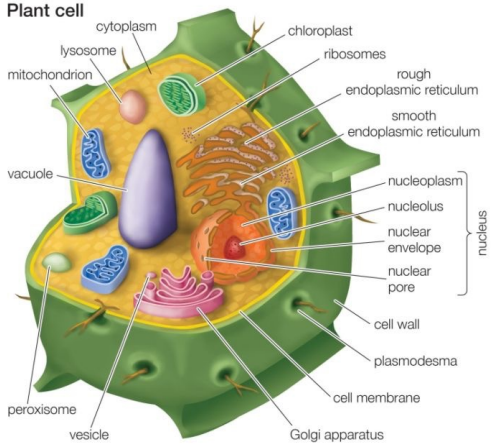
DNA – HYDROGEN BONDS

- Mitochondrial and chloroplastic genomes are enriched in AT.
- Nuclear genome is enriched in GC.
- On average, plastid DNA GC-content is ~37%, whereas nrDNA GC-content is ~41%.
- These genome structural properties can be used to filter reads in bioinformatics pipelines and study gene trafficking between genomes.





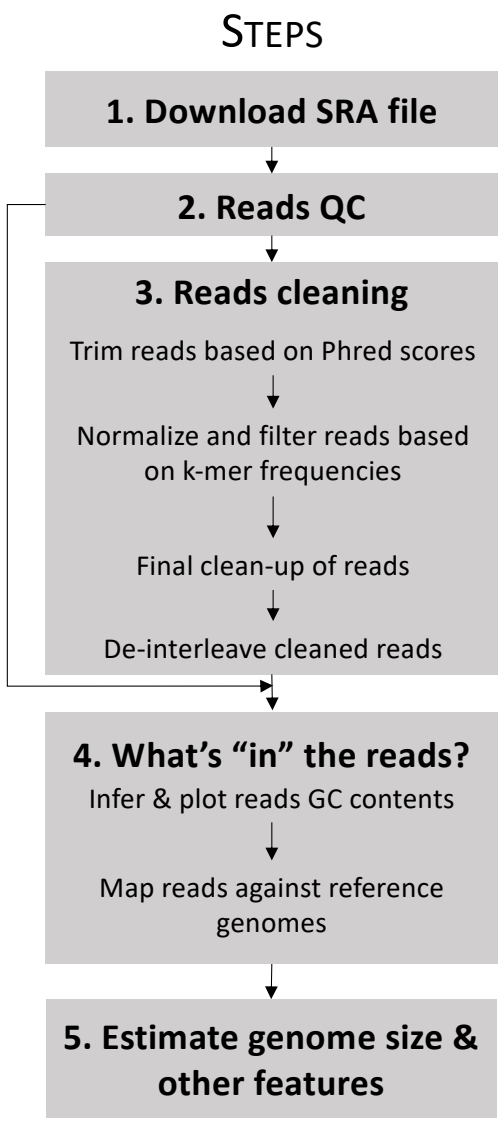




S3: READS CLEANING

Reads will be cleaned/trimmed based on:

- Phred quality scores (33) to conduct a first round of trimming.
- K-mer frequencies (k=21) to:
  - ✓Normalize high coverage reads (higher than 100x) based on median reads coverage.
  - ✓Filter low abundance reads (where PCR/sequencing errors will most likely take place).
- A final round of cleaning by removing low quality bases, short sequences, and non-paired reads.
- Reads will be formatted for *de novo* genome assembly using *SOAPdenovo2*.



What is a k-mer?

- A **k-mer** is a substring of length *k* in a string of DNA bases or sequence.
- For a given sequence of length *L*, and a k-mer size of *k*, the total number of k-mers possible (*n*) equals:

$$n = (L - k) + 1$$

- For instance, for a sequence of length 9 (*L*), and a k-mer length of 2 (*k*) the total number of k-mers equals:

$$n = (9 - 2) + 1 = 8$$

- Example: All eight 2-mers of the sequence "AATTGGCCG" are AA, AT, TT, TG, GG, GC, CC, CG

COUNTING K-MERS

- Most studies provide an estimate of sequencing coverage prior to assembly (e.g. 32x in our example), but the real coverage distribution will be influenced by:
  - ✓DNA quality,
  - ✓Library preparation,
  - ✓Local GC content,
  - ✓Genome complexity.
- One way of rapidly examining the coverage distribution (and genome complexity) before assembling a reference genome is to chop your raw sequence reads into short "k-mers" of 21 nucleotides, and count how often you get each possible k-mer.

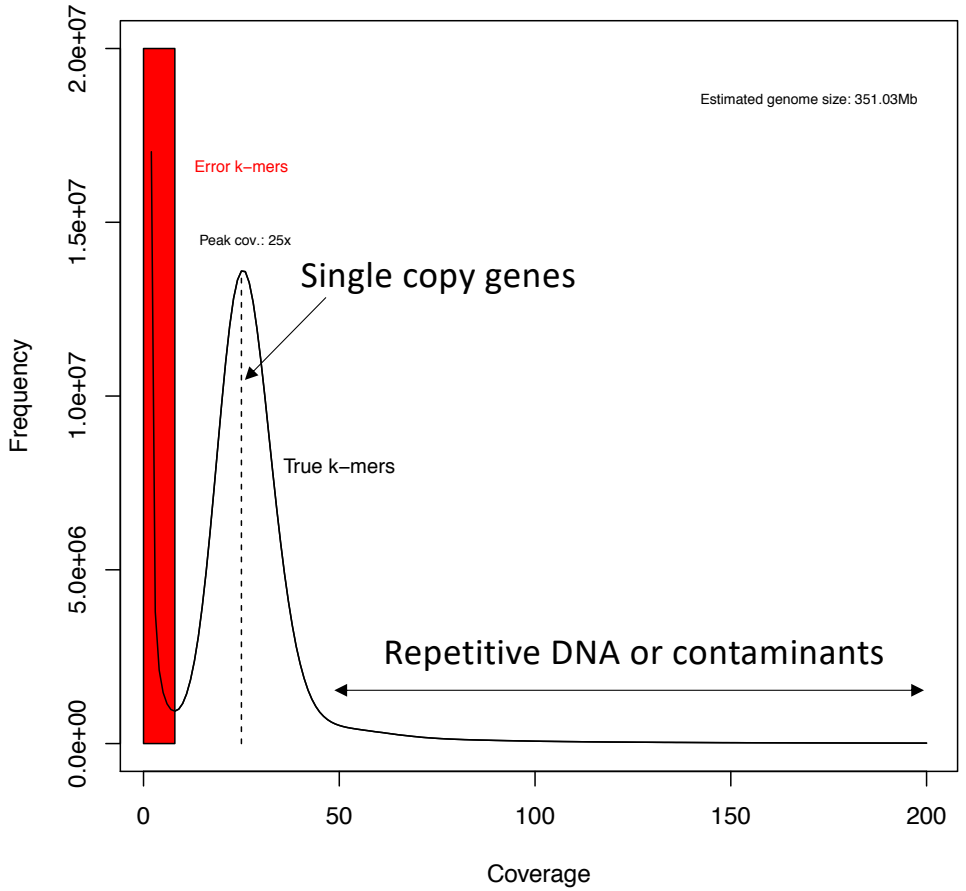
COUNTING K-MERS

By counting k-mers you will find out that:

- ✓**Many sequences are extremely rare.** They are either PCR or sequencing errors or could be rare somatic mutations. Such sequences could confuse assembly software; eliminating them can decrease subsequent memory & CPU requirements.
- ✓**Other sequences may exist at 10,000x coverage.** These could be pathogens/contaminants or repetitive elements. Often, there is no benefit to retaining all copies of such sequences because the assembly software will be confused by them; while retaining a small proportion such reads could significantly reduce CPU, memory and space requirements (this is especially important for this course).

K-MER GRAPH TO ESTIMATE KEY GENOMIC FEATURES

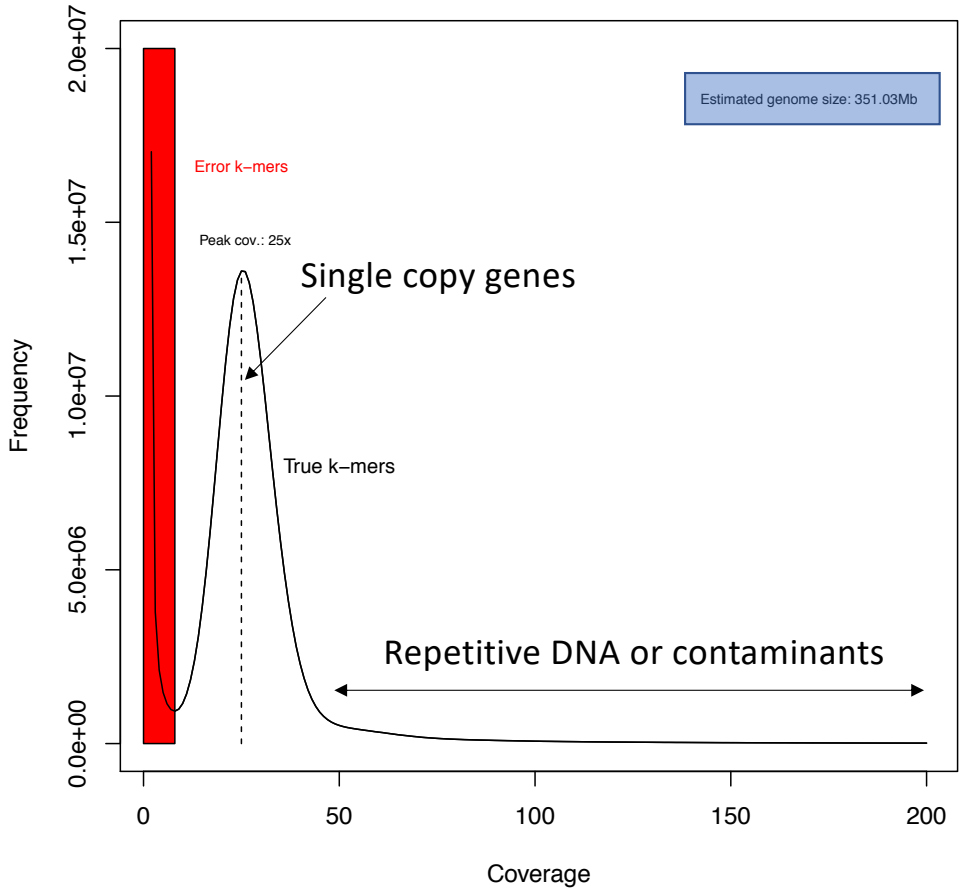
- Unique K-mers (1x; in red) are potential PCR and/or sequencing errors.
- The peak @25x represents the haploid genome (single copy genes). There are $1.4e^{7}$ unique 21-mers (frequency) that have been observed 25 times (coverage).
- The tail of the distribution (coverage >100x) most likely represents repetitive DNA or contaminants.



K-MER GRAPH TO ESTIMATE KEY GENOMIC FEATURES

- Haploid genome size (N) is equal to:
  - ✓*N = Total numbers of k-mers / Peak of coverage (25x)*
  - ✓*N = Area under the curve / Peak of coverage (25x)*

Remember this key concept for Step 5



K-MERS & PCR ERRORS – AN EXAMPLE

- **This “real” sequence "AATTGGCCG"**

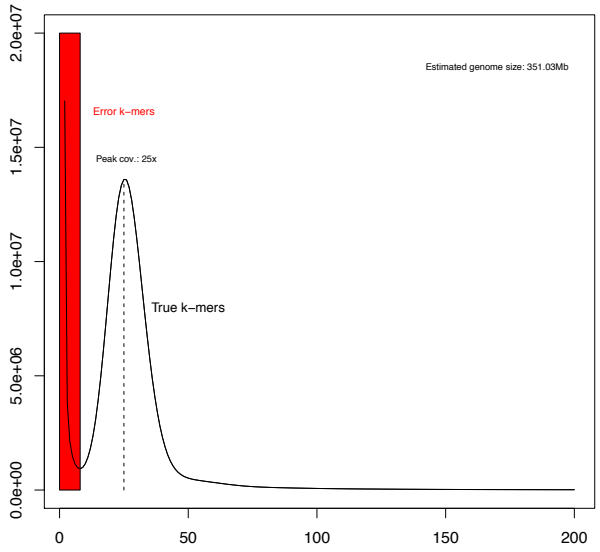
All 3-mers of the sequence are AAT, ATT, TTG, TGG, GGC, GCC, CCG

K-MERS & PCR ERRORS – AN EXAMPLE

- **Now consider that the 4th letter (T) is replaced with a C in the sequence to simulate a PCR error: "AATCGGCCG"**

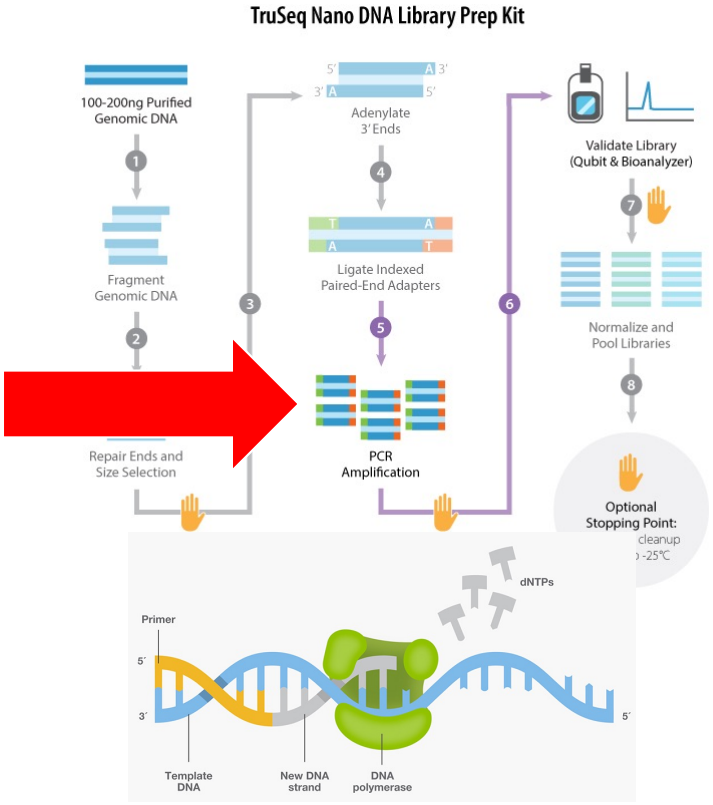
All 3-mers of this "biased" sequence are AAT, **ATC**, **TCG**, **CGG**, GGC, GCC, CCG.

The k-mers in bold are the incorrect 3-mers that are now unique and end up at the beginning of the graph.



K-MERS & PCR ERRORS – AN EXAMPLE

- **This error most likely takes place during the DNA library preparation:**



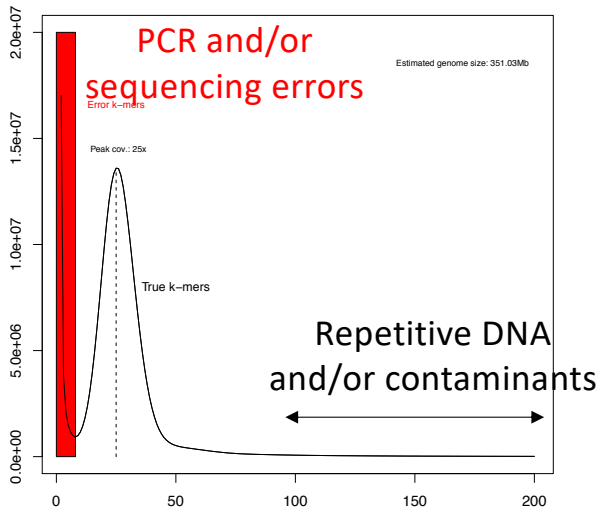
- **Or it could also be a sequencing error…**

Type	Instrument	Primary Errors	Single-pass Error Rate (%)	Final Error Rate (%)
Short reads	3730xl (capillary)	substitution	0.1-1	0.1-1
	454 All models	indel	1	1
	Illumina All Models	substitution	~0.1	~0.1
	Ion Torrent – all chips	indel	~1	~1
	SOLiD – 5500xl	A-T bias	~5	≤0.1
Long reads	**Oxford Nanopore**	deletions	≥4*	4*
	PacBio RS	Indel	~13	≤1

S3: READS CLEANING

Here, a k-mer approach is applied to:

- Filter low coverage reads to minimize the effect of PCR and/or sequencing errors on the *de novo* assembly.
- Normalize high coverage reads (>100x) based on median coverage to optimize RAM requirements for *de novo* assembly.



STEPS

1. Download SRA file

2. Reads QC

3. Reads cleaning

Trim reads based on Phred scores

Normalize and filter reads based on k-mer frequencies

Final clean-up of reads

De-interleave cleaned reads

4. What's "in" the reads?

Infer & plot reads GC contents

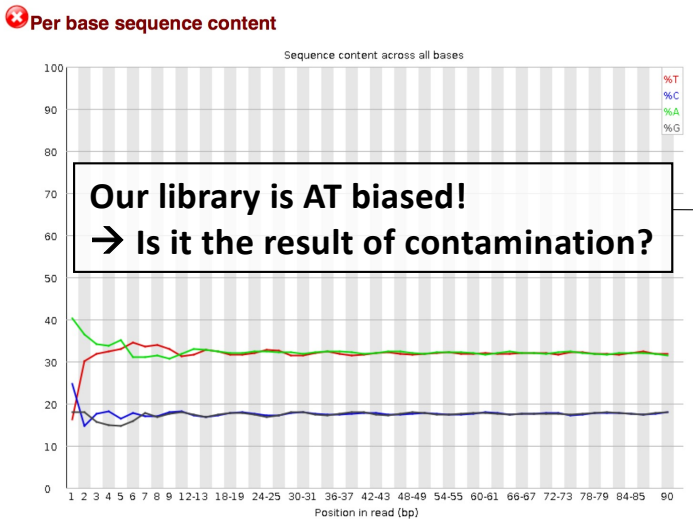
Map reads against reference genomes

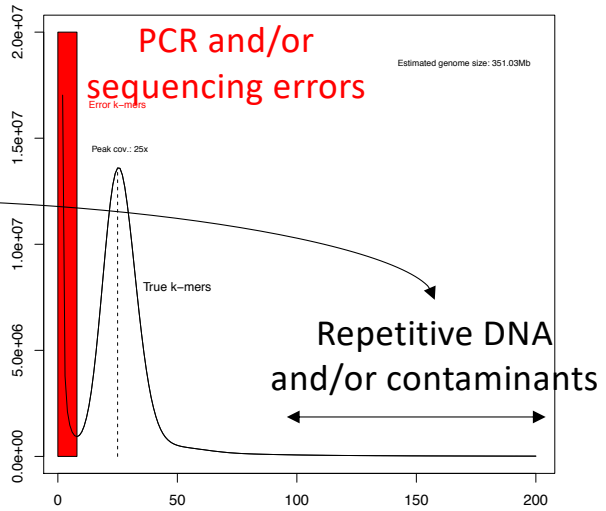
5. Estimate genome size & other features

S4: WHAT'S "IN" THE READS?

Here, we want to:

- Assess potential contaminants by inferring and plotting reads GC contents.





STEPS

1. Download SRA file

2. Reads QC

3. Reads cleaning

Trim reads based on Phred scores

Normalize and filter reads based on k-mer frequencies

Final clean-up of reads

De-interleave cleaned reads

4. What's "in" the reads?

Infer & plot reads GC contents

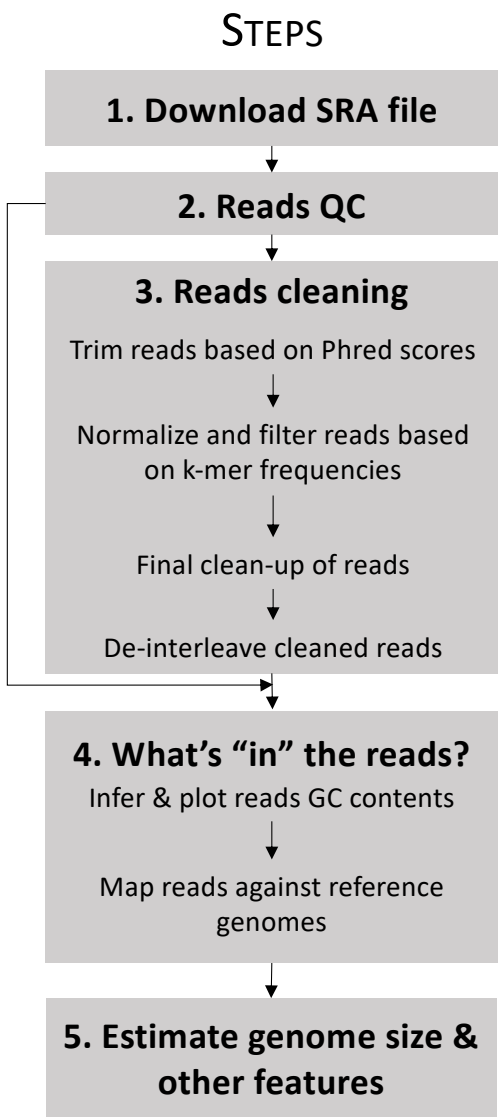
Map reads against reference genomes

5. Estimate genome size & other features

S4: WHAT'S "IN" THE READS?

Here, we want to:

- A GC plot from an **uncontaminated library would be expected to produce a smooth, unimodal distribution**.
- **Shoulders, or in more extreme cases a bimodal distribution, could indicate the presence of sequence reads from an organism with a different GC content**, which is most likely a contaminant.

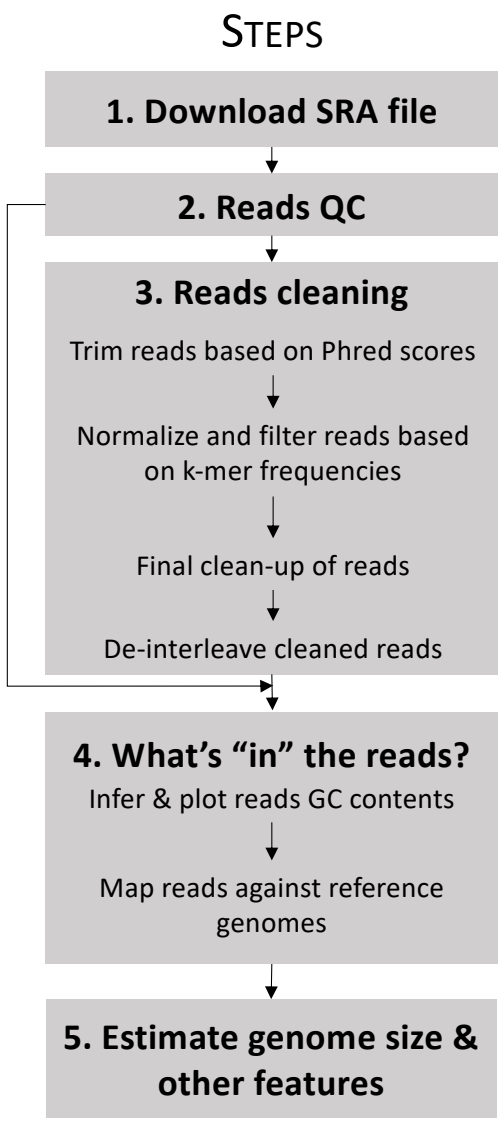


S4: WHAT'S "IN" THE READS?

Here, we want to:

- Map reads against reference genomes (using BWA*) to assess proportions of reads from:
  - ✓Nuclear genome
  - ✓Chloroplast genome
  - ✓Other

*BWA is a software package for mapping low-divergent sequences against a large reference genome



S4: WHAT'S "IN" THE READS?

NCBI Resources How To

Sequence Set Browser Show help

Project: PEFY01 Search List of all Projects

PEFY00000000.1 Apostasia shenzhenica

Master | Contigs | Download | History

# of Contigs:	12,380
# of Proteins:	21,743
# of Scaffolds/Chrs:	2,985
Total length:	322,899,837 bp
BioProject:	PRJNA310678
BioSample:	SAMN04453324
Keywords:	WGS
Annotation:	Scaffolds
Organism:	Apostasia shenzhenica – show lineage
Biosource:	/country = China: Shenzhen /ecotype = Shenzhen /isolate = ASH160606 /mol_type = genomic /tissue_type = stem; leaf
WGS:	PEFY01000001:PEFY01012380
Scaffolds:	KZ451883:KZ454867 2,985 scaffolds, 21,743 proteins, total length is 348,733,136 bases
Reference:	The Apostasia genome and the evolution of orchids : Nature 549 (7672), 379-383 (2017) – show 35 authors
Submission:	Submitted (25-OCT-2017) Shenzhen Key Laboratory for Orchid Conservation and Utilization, The National Orchid Conservation Center of China, Wangtong Road, Shenzhen 518114, China – Liu,Z.-J.

Apostasia wallichii chloroplast DNA, complete genome

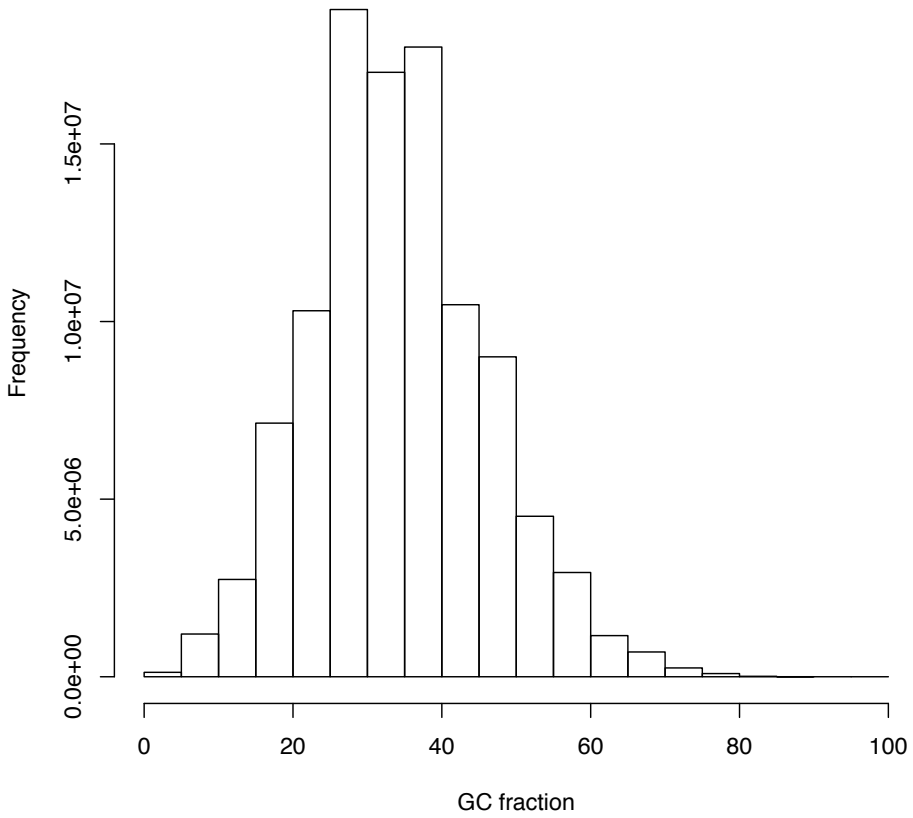
GenBank: LC199394.1

FASTA Graphics

Go to:

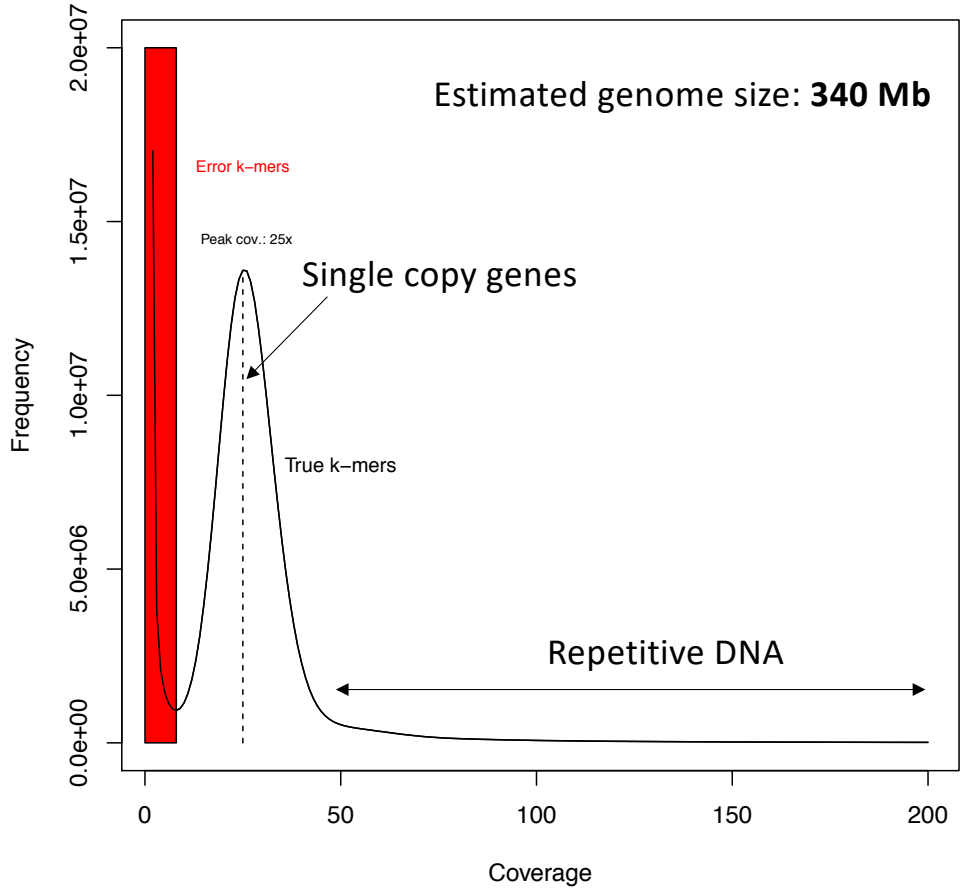
```
LOCUS       LC199394              156126 bp    DNA     circular PLN 02-NOV-2017
DEFINITION  Apostasia wallichii chloroplast DNA, complete genome.
ACCESSION   LC199394
VERSION     LC199394.1
KEYWORDS    .
SOURCE      chloroplast Apostasia wallichii
  ORGANISM  Apostasia wallichii
            Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
            Spermatophyta; Magnoliophyta; Liliopsida; Asparagales; Orchidaceae;
            Apostasioideae; Apostasia.
REFERENCE   1
  AUTHORS   Niu,Z., Pan,J., Zhu,S., Li,L., Xue,Q., Liu,W. and Ding,X.
  TITLE     Comparative Analysis of the Complete Plastomes of Apostasia
            wallichii and Neuwiedia singapureana (Apostasioideae) Reveals
            Different Evolutionary Dynamics of IR/SSC Boundary among
            Photosynthetic Orchids
  JOURNAL   Front Plant Sci 8, 1713 (2017)
   PUBMED   29046685
  REMARK    DOI:10.3389/fpls.2017.01713
            Publication Status: Online-Only
REFERENCE   2  (bases 1 to 156126)
  AUTHORS   Niu,Z.T., Zhu,S.Y. and Ding,X.Y.
  TITLE     Direct Submission
  JOURNAL   Submitted (28-NOV-2016) Contact:Niu Zhitao Nanjing Normal
            University, College of Life Sciences; No.1, Wenyuan Road, Nanjing,
            Jiangsu 210023, China
FEATURES             Location/Qualifiers
     source          1..156126
                     /organism="Apostasia wallichii"
                     /organelle="plastid:chloroplast"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:280454"
     misc_feature    1..83035
                     /note="large single copy region (LSC)"
     gene            complement(41..1102)
                     /gene="psbA"
```

LIBRARY IS NOT CONTAMINATED



- GC profile is shifted towards lower GC values.
- Overall library contains >98% of reads belonging to nuclear genome.

ESTIMATED GENOME SIZE IS CLOSE TO EXPECTED VALUE



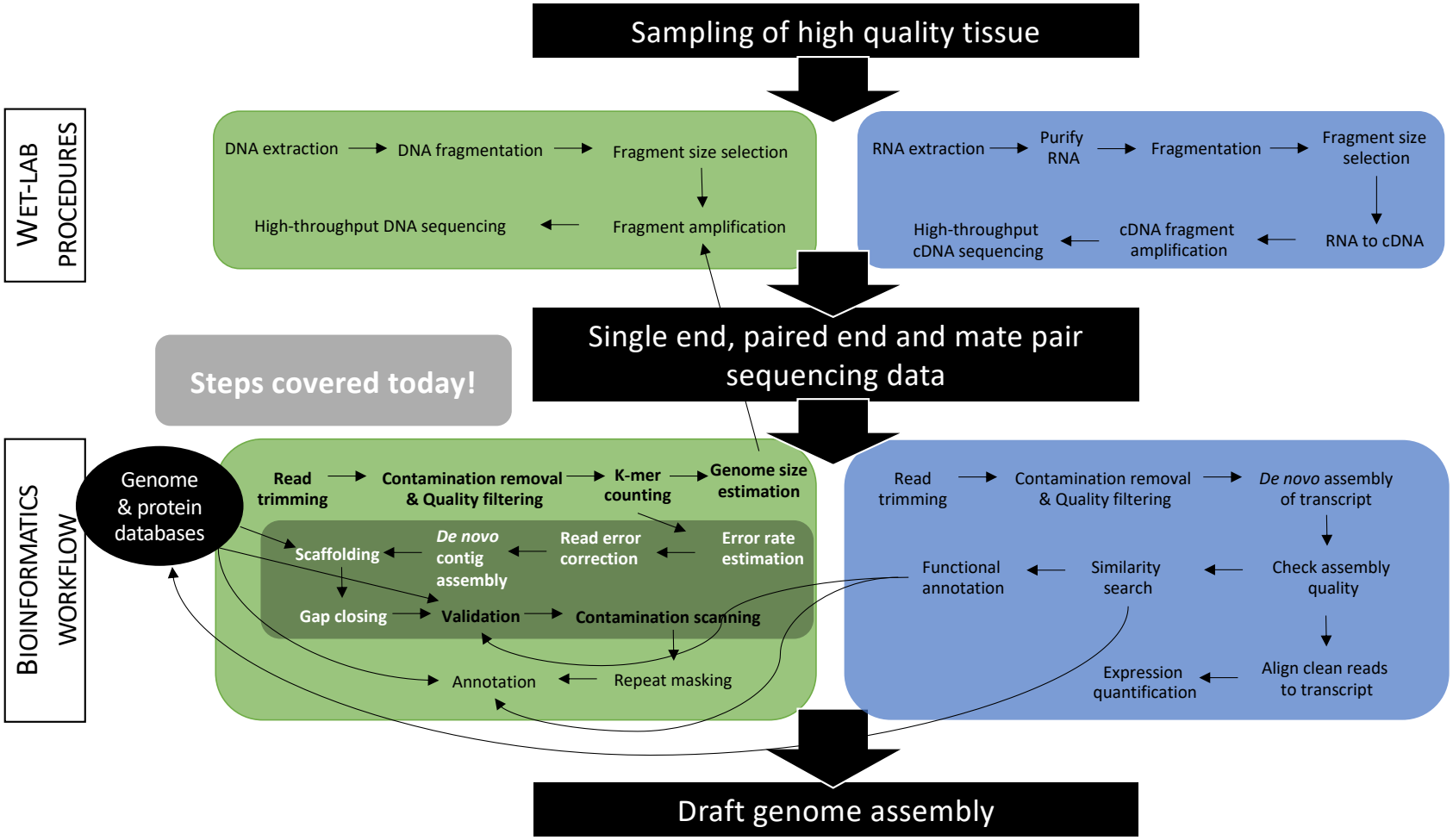
- Based on library, estimated genome size is ca. 340 Mb.
- Not far from the value obtained by Zhang *et al.:* **349 Mb**
- This means that we might have enough data to reconstruct at least the single-copy genes, which are sequenced ca. 20x times.
- Validate with GenomeScope.

PARTS 2, 3 – DE NOVO ASSEMBLY AND VALIDATION

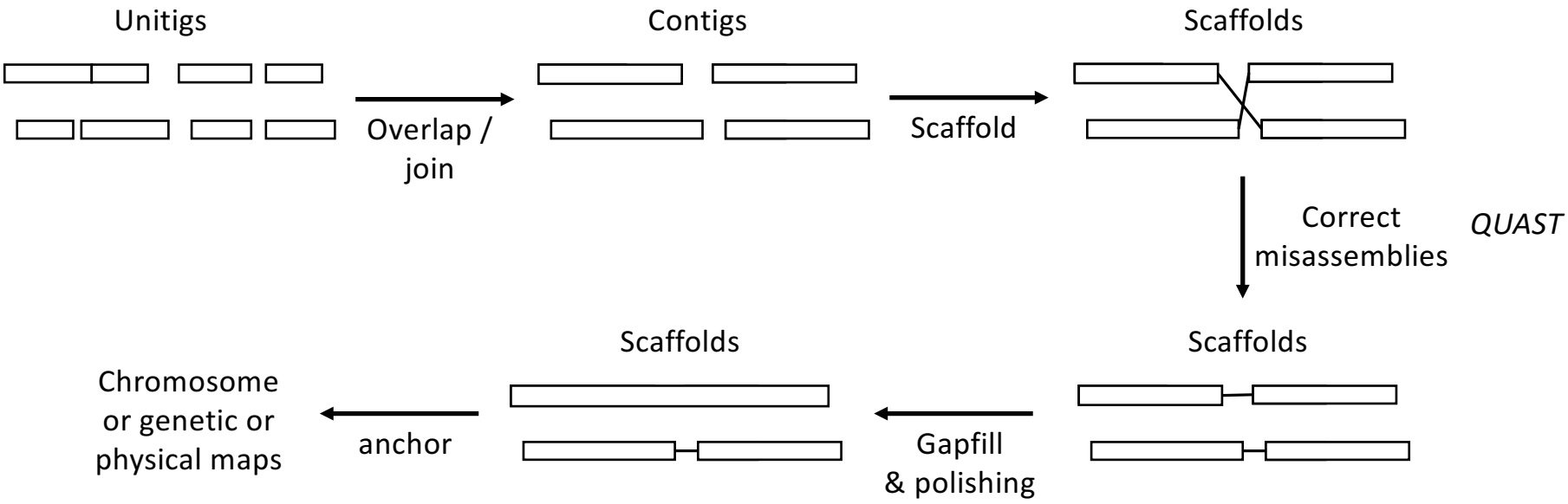
- Learn to set-up and perform a *de novo* genome assembly based on cleaned Illumina reads using *SOAPdenovo2*.

- Provide theoretical knowledge on *de novo* genome assembly methods. Focusing on de Bruijn graphs.

- Validate the *de novo* genome assembly using *QUAST*.


Sampling of high quality tissue
WET-LAB PROCEDURES
DNA extraction
DNA fragmentation
Fragment size selection
High-throughput DNA sequencing
Fragment amplification
RNA extraction
Purify RNA
Fragmentation
Fragment size selection
RNA to cDNA
cDNA fragment amplification
High-throughput cDNA sequencing
Single end, paired end and mate pair sequencing data
Steps covered today!
BIOINFORMATICS WORKFLOW
Genome & protein databases
Read trimming
Contamination removal & Quality filtering
K-mer counting
Genome size estimation
Error rate estimation
Read error correction
De novo contig assembly
Scaffolding
Gap closing
Validation
Contamination scanning
Annotation
Repeat masking
Read trimming
Contamination removal & Quality filtering
De novo assembly of transcript
Check assembly quality
Align clean reads to transcript
Expression quantification
Functional annotation
Similarity search
Draft genome assembly

OVERVIEW OF THE *DE NOVO* ASSEMBLY WORKFLOW



Contig: A contiguous sequence of bases.
Unitig: A type of contig for which there are no competing choices in terms of internal overlaps (they usually stop before a repeat sequence).
Scaffold: A sequence of contigs separated by gaps (Ns).

WHAT IS THE BEST ASSEMBLER FOR OUR DATA?

- *SOAPdenovo2* vs. *ALLPATHS-LG*.
- Both algorithms are adapted to Illumina reads.
- *ALLPATHS-LG* requires high sequencing (>100x) coverage to assemble genome.
- *ALLPATHS-LG* requires a minimum of 2 paired-end libraries: one short and one long.

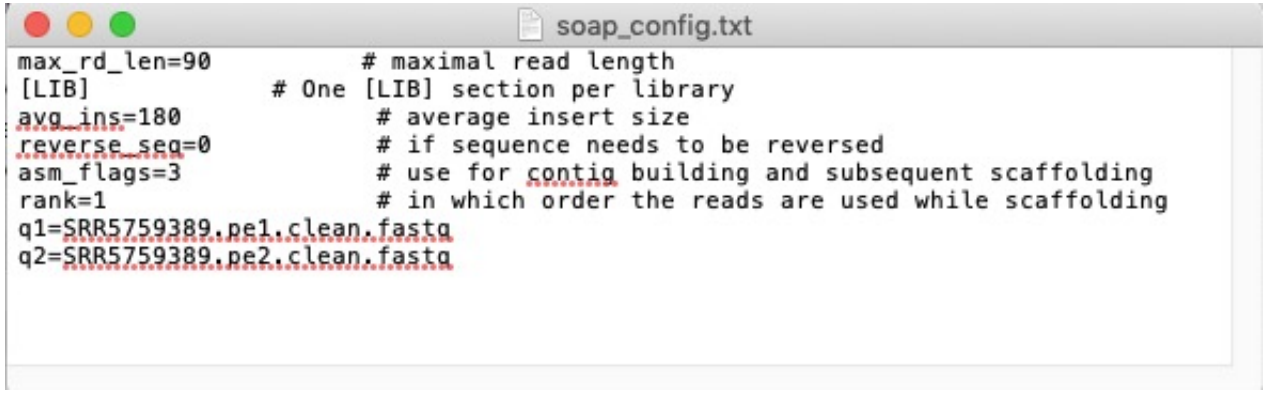
→ **We have only 20-25x coverage (for single-copy genes) and one library with an insert-size of max. 180 (2x 90 bp)!**

SOAPdenovo2

- This program is made up of six modules handling:
  1. Read error correction.
  2. **de Bruijn graph construction.**
  3. **Contig assembly.**
  4. Paired-end reads mapping.
  5. Scaffold construction.
  6. Gap closure.

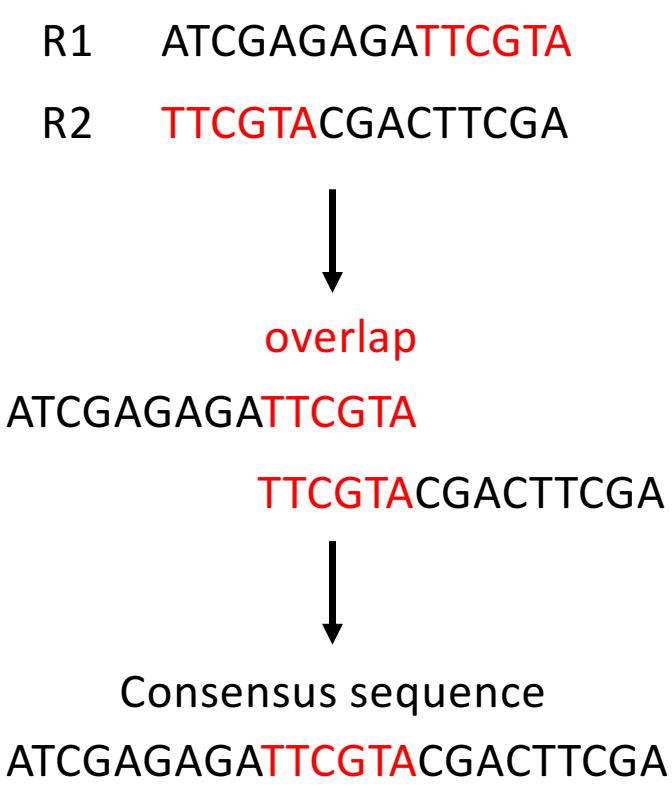
SOAPdenovo2 – Setting up the analysis

- **Step 1:** Create a folder and copy the de-interleaved cleaned paired-end *fastq* files.
- **Step 2:** Create a configuration file providing the settings of the analysis.
- **Step 3:** Run the *de novo* genome assembly analysis. THIS ANALYSIS TAKES A WHILE TO RUN!

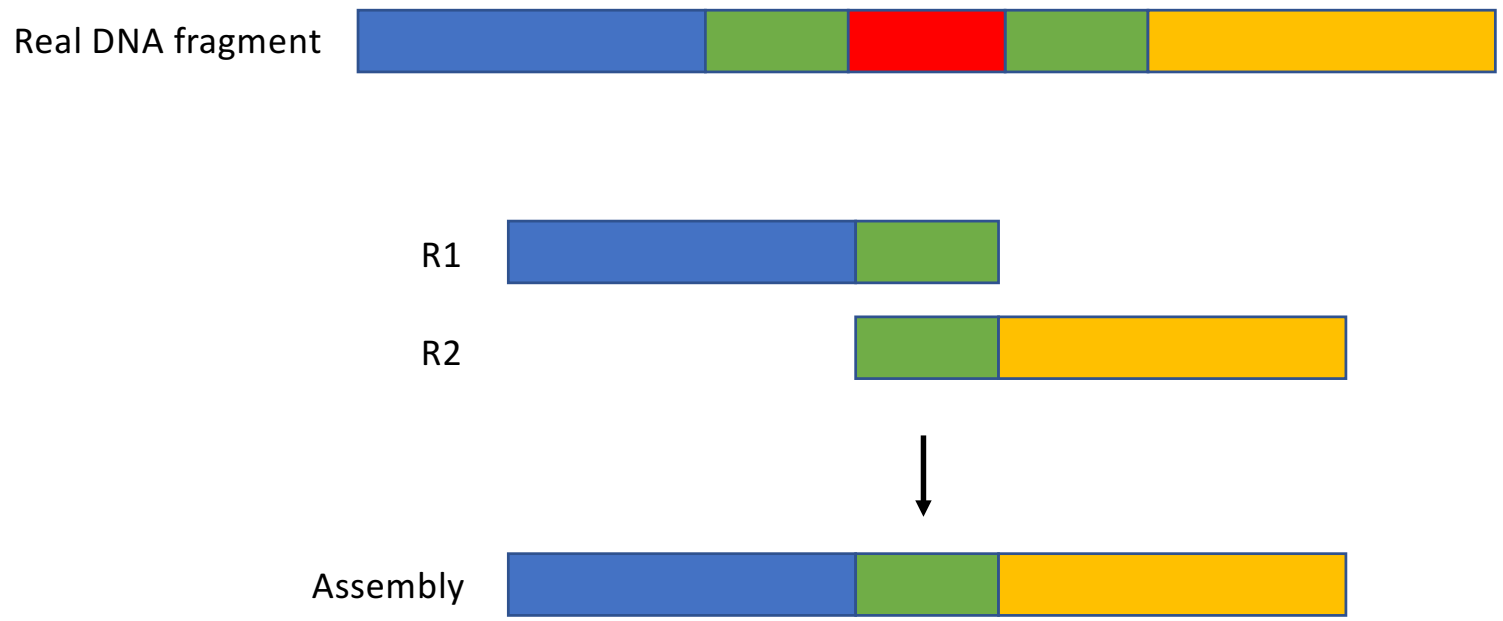


THE OVERLAP-LAYOUT-CONSENSUS (OLC) METHOD

- Traditional method used to assemble long reads (i.e. Sanger reads).
- The assembler identifies overlaps between various long reads.
- Based on those overlaps, it subsequently merges the read fragments into longer sequences.
- This method poorly performs with repetitive DNA regions.



THE OVERLAP-LAYOUT-CONSENSUS (OLC) METHOD



- Green segments are nearly identical.
- OLC may erroneously connect the blue and orange segments, and skip the red segment in between.

THE OVERLAP-LAYOUT-CONSENSUS (OLC) METHOD

- **To properly handle repetitive DNA regions OLC programs:**
  1. Mask repetitive and low-complexity regions.
  2. Assemble the remaining genome into many contigs and scaffolds.
  3. Then an expensive completion step is employed to merge scaffolds into super-scaffolds and fill up the repeats.
- **An OLC assembler needs to constantly guess whether slight variation between two overlapping segments is due to repeats or error.** This can be done by using e.g. phred quality scores.

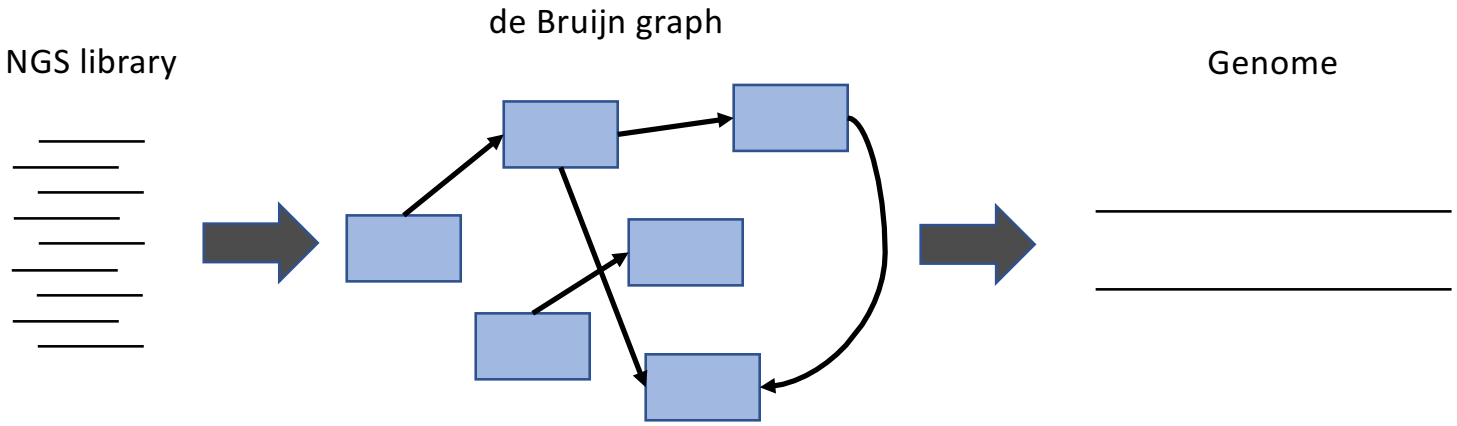
GENOMIC OLC ASSEMBLERS

- **CANU:** Assembler designed for high-noise single-molecule sequencing (e.g. PacBio, Oxford Nanopore).
- **MIRA:** This program is capable of performing assemblies from a wide range of sequence types (e.g. Sanger, Illumina, PacBio).
- **SGA:** The string graph assembler (SGA) uses a modified approach to conventional OLC assemblers. It makes use of an FM-index to accelerate the initial identification of read overlaps making the OLC approach more tenable for assemblies consisting of large numbers of reads. It has considerably lower memory overheads than a de Bruijn graph based assembler.

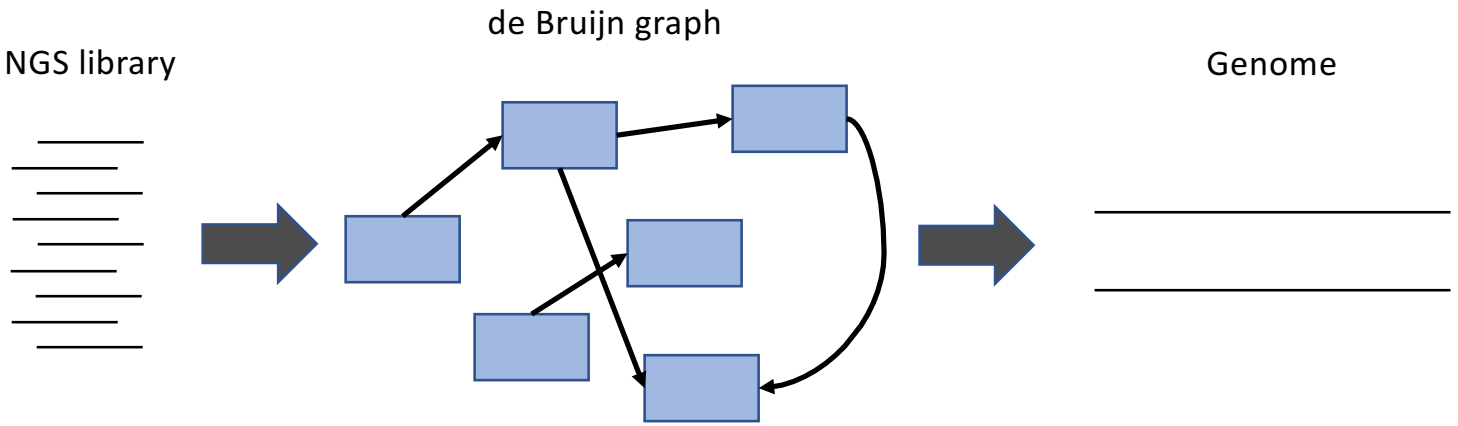
DE BRUIJN GRAPH IN A NUTSHELL

To construct a de Bruijn graph of any genome with k-mer of any size:

1. The reads are split into its k-mer components.
2. k-mers are connected based on whether they have k-1 common nucleotides.
3. De Bruijn graph is then used to reconstruct genome sequence.



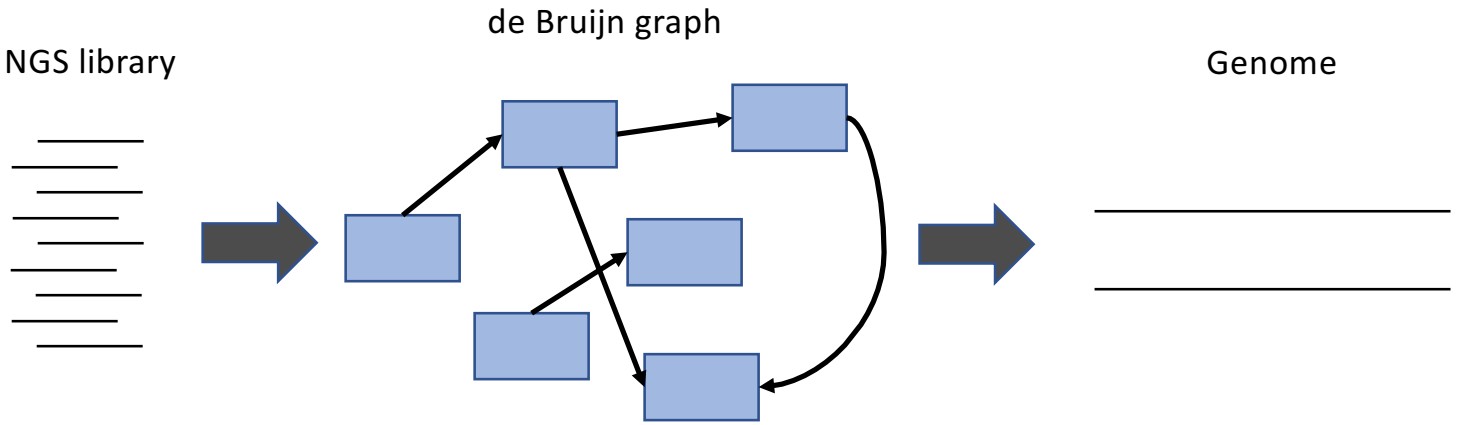
DE BRUIJN GRAPH IN A NUTSHELL



Step 1

- Split short reads into smaller pieces (k-mers).
- K-mers retain enough characteristics of the genome to allow its reconstruction, yet are short enough to provide detailed statistics to perform error corrections.

DE BRUIJN GRAPH IN A NUTSHELL

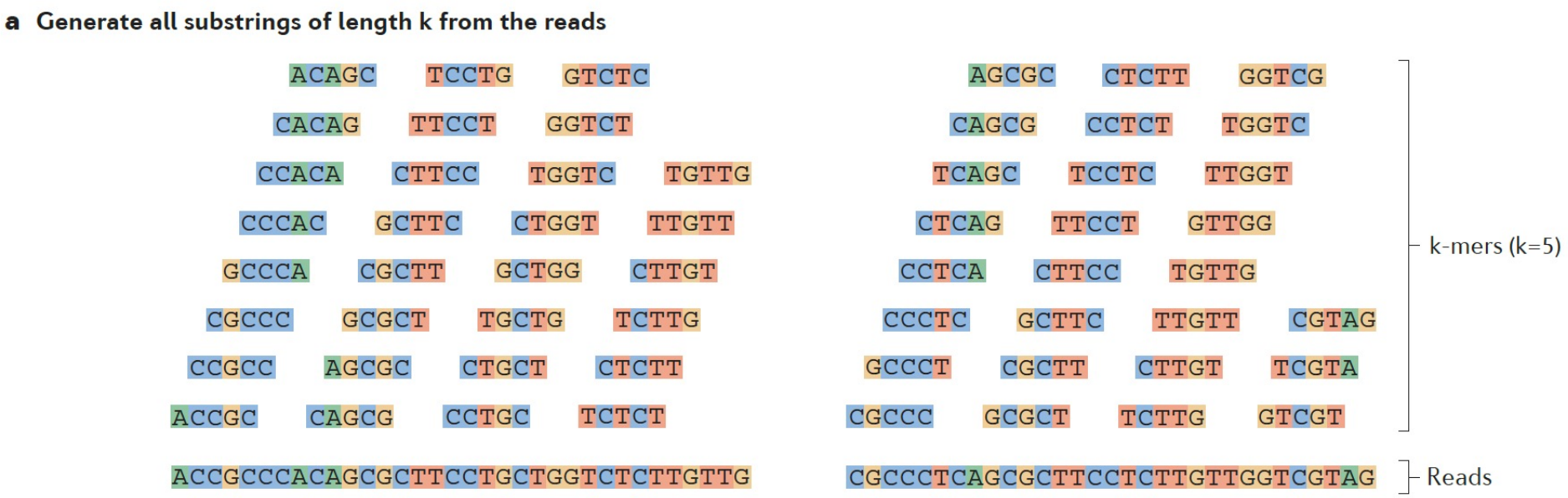


Steps 2 & 3

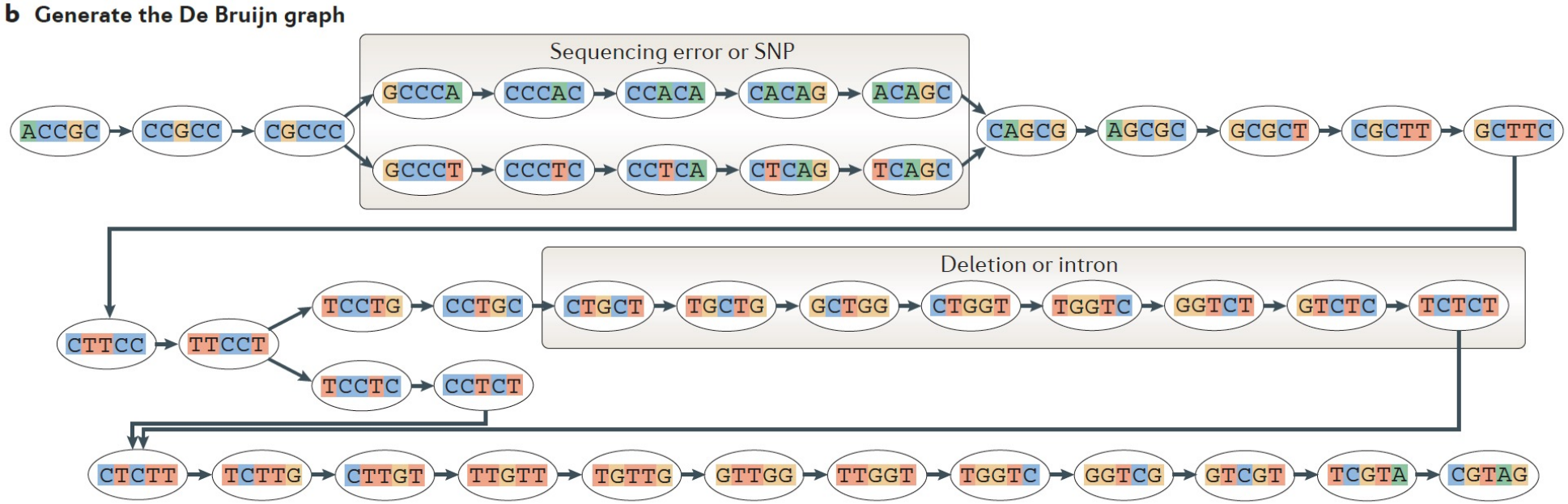
- Connect k-mers (using overlap of k-1) into a de Bruijn graph.
- De Bruijn graph is then used to reconstruct genome sequence.

Based on Martin & Wang (2011), *Nature Reviews*

DE BRUIJN DE NOVO ASSEMBLY – STEP 1 GENERATE K-MERS

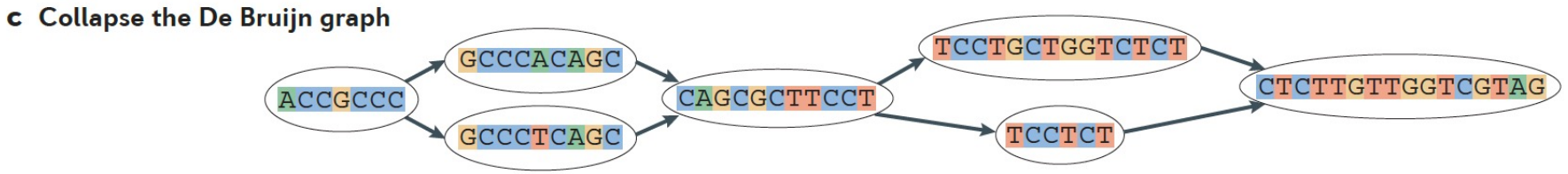


DE BRUIJN DE NOVO ASSEMBLY – STEP 2 GENERATE DE BRUIJN

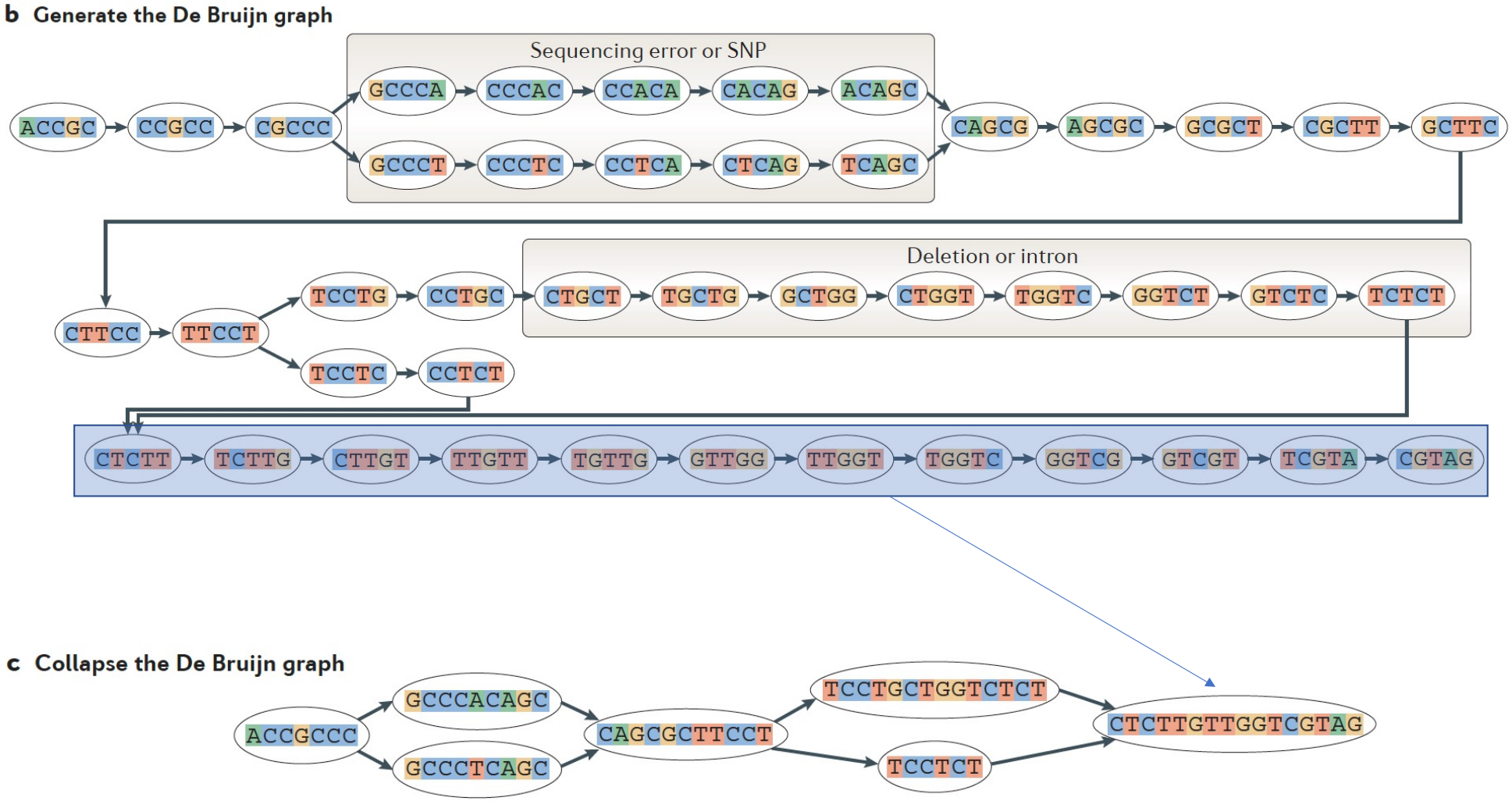


Connect nodes (unique k-mers) only when they have a k-1 overlap

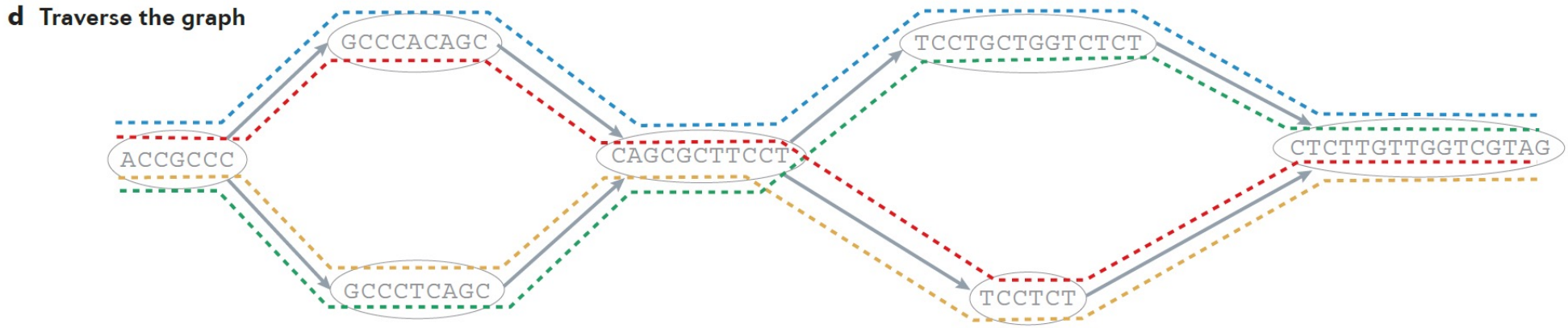
DE BRUIJN DE NOVO ASSEMBLY – STEP 3 COLLAPSE DE BRUIJN



Chaines of adjacent nodes in the graph are collapsed into a single node.


b Generate the De Bruijn graph
Sequencing error or SNP
ACCGC
CCGCC
CGCCC
GCCCA
CCCAC
CCACA
CACAG
ACAGC
GCCCT
CCCTC
CCTCA
CTCAG
TCAGC
CAGCG
AGCGC
GCGCT
CGCTT
GCTTC
Deletion or intron
CTTCC
TTCCT
TCCTG
CCTGC
TCCTC
CCTCT
CTGCT
TGCTG
GCTGG
CTGGT
TGGTC
GGTCT
GTCTC
TCTCT
CTCTT
TCTTG
CTTGT
TTGTT
TGTTG
GTTGG
TTGGT
TGGTC
GGTCG
GTCGT
TCGTA
CGTAG
c Collapse the De Bruijn graph
ACCGCCC
GCCCACAGC
GCCCTCAGC
CAGCGCTTCCT
TCCTGCTGGTCTCT
TCCTCT
CTCTTGTTGGTCGTAG

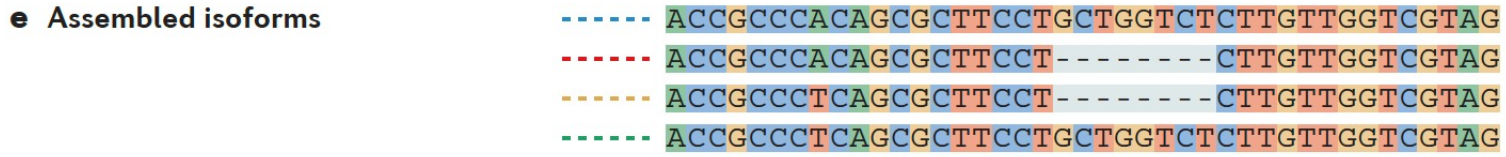
DE BRUIJN DE NOVO ASSEMBLY – STEP 4 TRAVERSE THE GRAPH



Here, we can include PE information to favor some paths over others. This allow assembling the most likely DNA sequences

Traversing: A method for systematically visiting all nodes in a mathematical graph

DE BRUIJN DE NOVO ASSEMBLY – STEP 5 ASSEMBLE SEQ.



This is an example from RNA-Seq

DE BRUIJN GRAPH OF A SMALL SEQUENCE

- Infer de Bruijn graph from an already assembled genome sequence.
- Edges are drawn between node pairs (k-mers) to connect nodes with an overlap of k-1 (6).
- Simple graph since none of the 7-mers appeared more than once in the original sequence.

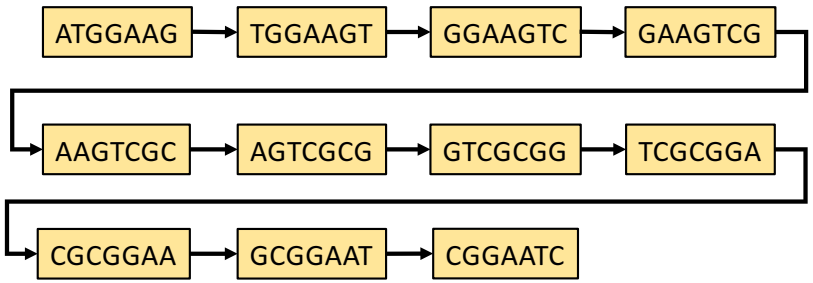
Sequence ATGGAAGTCGCGGAATC

7-mers ATGGAAG
TGGAAGT
GGAAGTC
GAAGTCG
AAGTCGC
AGTCGCG
GTCGCGG
TCGCGGA
CGCGGAA
GCGGAAT
CGGAATC

Step 1: Generate set of k-mers

Step 2: Infer directed graph showing overlaps between k-mers

de Bruijn graph



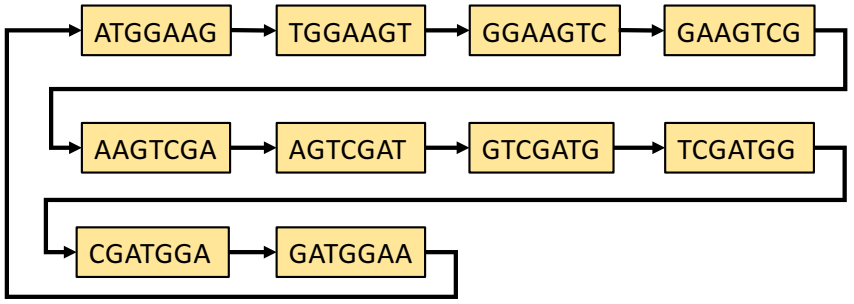
DE BRUIJN GRAPH OF A SMALL SEQUENCE

- Here the 5’-most and 3'-most 7-mers are identical (in blue) → **Creating redundancy in de Bruijn graph**
- The de Bruijn graph has one less node due to merger of those two identical nodes.
- A loop connects the 2 ends of the graph.

Sequence ATGGAAGTCGATGGAAG

7-mers
ATGGAAG
TGGAAGT
GGAAGTC
GAAGTCG
AAGTCGA
AGTCGAT
GTCGATG
TCGATGG
CGATGGA
GATGGAA
ATGGAAG

de Bruijn graph



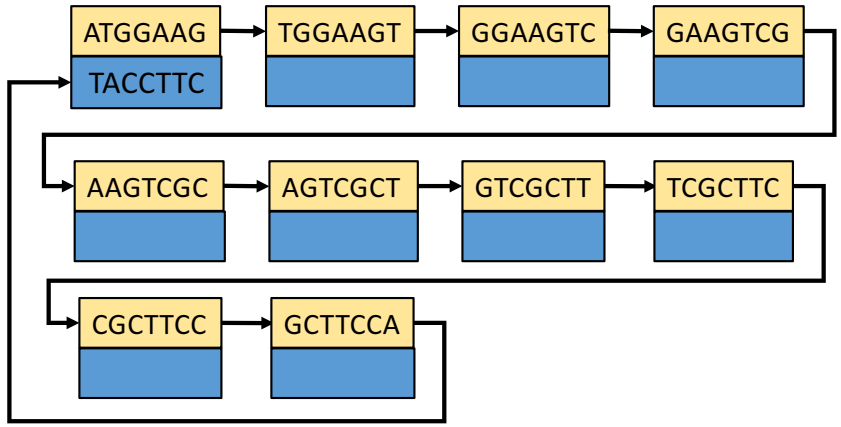
DOUBLE-STRANDED NATURE OF GENOME

- Although nodes displayed in previous examples did not show sequences from both strands, in reality, **each node of a de Bruijn graph is double-stranded**.
- Here, the **3'-most 7-mer is the reverse complement of the 5'-most 7-mer**.

Sequence ATGGAAGTCGCTTCCAT

7-mers
ATGGAAG
TGGAAGT
GGAAGTC
GAAGTCG
AAGTCGC
AGTCGCT
GTCGCTT
TCGCTTC
CGCTTCC
GCTTCCA
CTTCCAT

de Bruijn graph



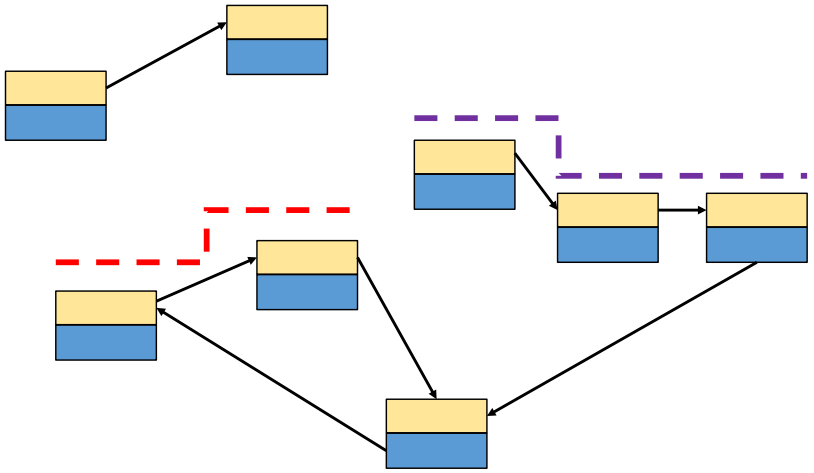
K-MERS SHOULD BE OF ODD LENGTH

- De Bruijn assemblers use k-mers of odd length (e.g. 21, 23, 25).
- If k-mers are of even length, some k-mers can be reverse complements of themselves (e.g. ATATATATATAT). **Even k-mers will create ambiguity in the de Bruijn graph and make its resolution difficult.**
- Palindromic k-mers can be avoided with odd k-mer size, because the reverse complement of center nucleotide is different from the nucleotide itself.

GENOME ASSEMBLY USING DE BRUIJN GRAPHS

De Bruijn graph-based algorithms solve the genome assembly problem in 2 steps:

1. A **de Bruijn graph is constructed** from all sequencing reads.
2. The **de Bruijn graph is then traversed to determine its underlying genome sequence**.



What is lost in de Bruijn graphs?

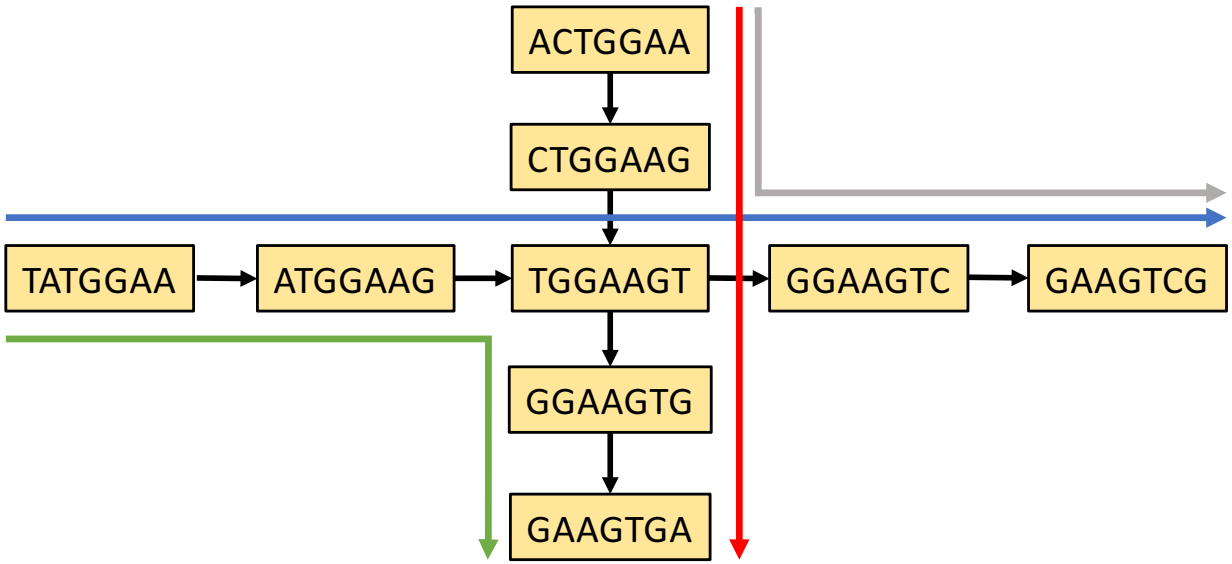
- **de Bruijn graphs do not preserve long-range positional information.** This means that one cannot go back from the de Bruijn graph to the read!
- **By converting a long read into a de Bruijn graph, we lose what was already known about that part of the genome.** The loss is proportional to the length of the read.
- This issue is especially troublesome for repeat DNA regions where long reads could help with the assembly of the genome sequence.

VARYING K-MER SIZES IN DE BRUIJN GRAPHS

- **To solve the previous issue, de Bruijn assemblers will be ran analyses with different k-mer sizes (21, 23, 25, 27, etc.) in order to find the best assembly.**
- Why does the method work? Let us present an intuitive explanation.

VARYING K-MER SIZES IN DE BRUIJN GRAPHS

- The graph can be traversed through 4 paths, but are they all real?

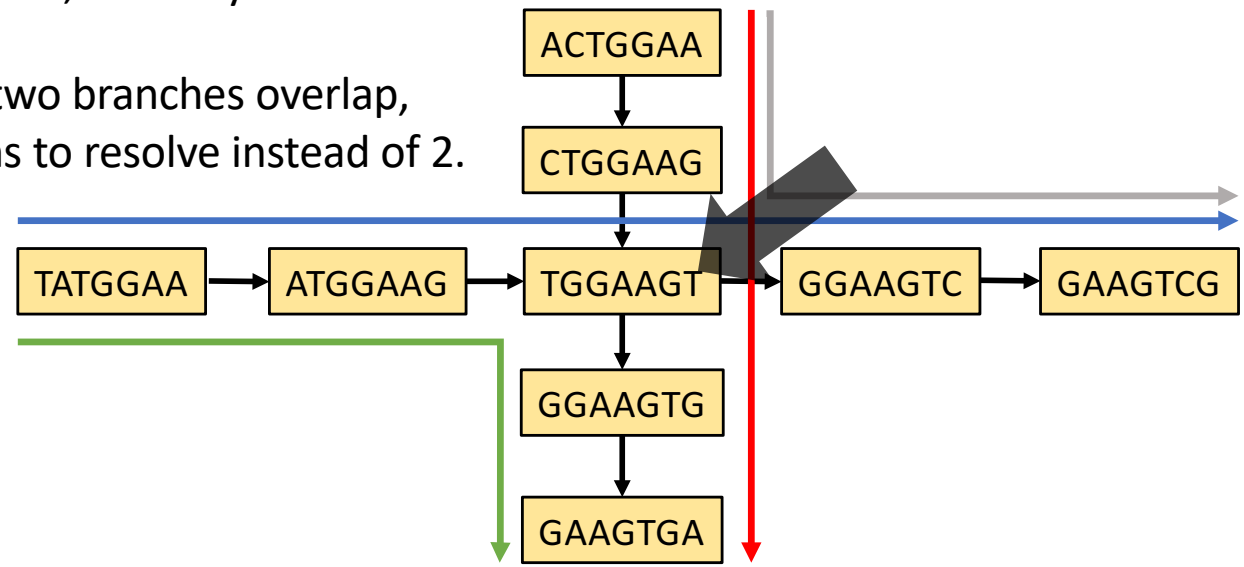


Real sequences: ACTGGAAGTGA and TATGGAAGTCG

K=7

VARYING K-MER SIZES IN DE BRUIJN GRAPHS

- The original sequences were likely from non-repetitive regions of a genome, but they have a common k-mer.
- This common k-mer made two branches overlap, giving the assembler 4 paths to resolve instead of 2.

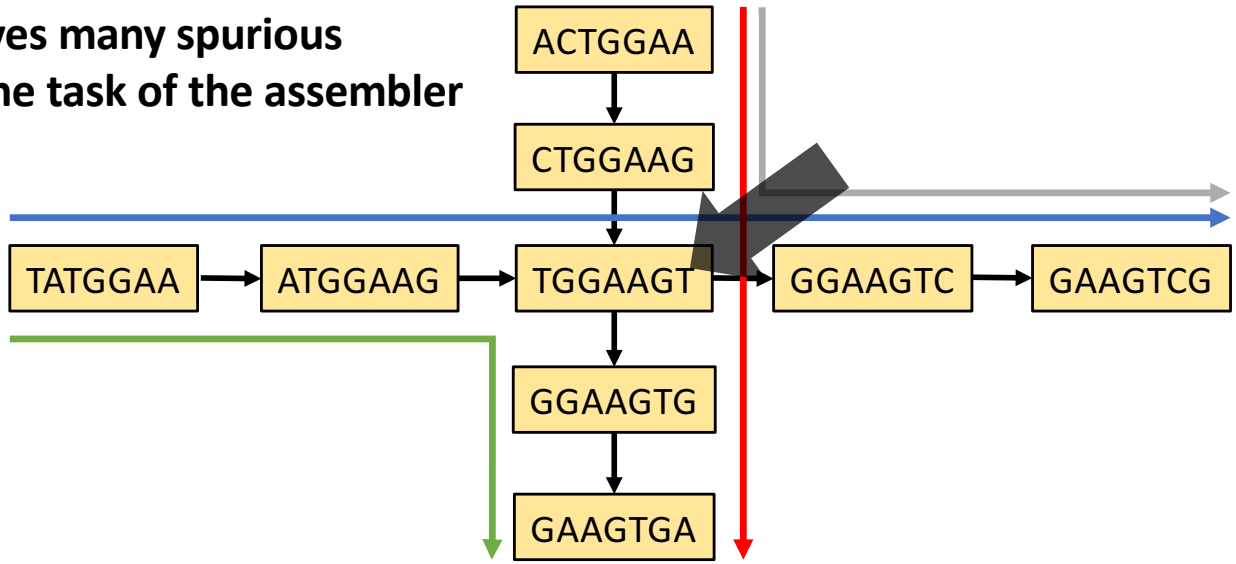


Real sequences: ACTGGAAGTGA and TATGGAAGTCG

K=7

VARYING K-MER SIZES IN DE BRUIJN GRAPHS

- The 2 paths will separate by changing the k-mer size from 7 to 9.
- **Increasing k-mer size resolves many spurious ambiguities, thus making the task of the assembler easier.**



Real sequences: ACTGGAAGTGA and TATGGAAGTCG

MEMORY REQUIREMENT AND K-MER DISTRIBUTION

- Researchers trying to assemble genomes or transcriptomes from NGS libraries will face these two problems:
  1. How to set k-mer parameters to get the best assembly.
  2. How to complete the assembly within RAM limits of the computer.

MEMORY REQUIREMENT AND K-MER DISTRIBUTION

- If all reads are perfect (no errors), they will all match the de Bruijn graph of the genome sequence.
- Irrespective of whether the genome is sequenced at 10x or 1000x depth, **the size of the de Bruijn graph will be limited by the size of the underlying genome and not the volume of data**.
- This means that you have to **do a good job at cleaning your reads** prior to *de novo* assembly to minimize the impact of errors and reduce RAM memory!

MEMORY REQUIREMENT AND K-MER DISTRIBUTION

However, we don’t live in a perfect world and all libraries have errors.

→ These errors make the assembly more problematic and therefore more RAM memory is required!

SELECTING THE BEST GENOME ASSEMBLER

- **Sequencing technology:**
  - ✓Short reads are only appropriate for de Bruijn graph assemblers.
  - ✓Long reads are better adapted to OLC assemblers. The Illumina 250 bp reads (obtained with the MiSeq platform) can be analyzed with OLC assemblers (e.g. your cholroplast genome)
- **Genome size and complexity:**
  - ✓All assemblers are capable of assembling simple prokaryotic genomes.
  - ✓Some assemblers are not capable of assembling larger genomes, which may be due to, e.g. excessive memory requirements, or difficulties in handling heterozygous polyploid genomes.
- **Source of sequencing data:** *De novo* assembly initially targeted at genomic sequences, but it is now adapted to *de novo* transcriptome assembly and metagenomics.

DE BRUIJN GENOME ASSEMBLERS

- **VELVET:** Assembler capable of producing assemblies from very short, early NGS reads (i.e. 25bp), but it can also handle longer (i.e. 454) reads to scaffold contig sequences. **High memory needs**.
- **SPADES:** Developed for single-cell and **prokaryotic** sequences. It incorporates an initial read error correction phase to reduce sequencing errors present in the input reads, before building a de Bruijn (utilizing multiple sizes of k-mer).
- **ABySS:** Capable of assembling **mammalian-sized** genomes from short reads. Built around MPI parallelization. It can make use of paired k-mers consisting of 2 k-mers separated by a fixed distance. It is equivalent to a single large k-mer spanning the length of the k-mer pair.

GAP CLOSING USING LONG READS

- Sequencing biases, repetitive genomic features, genomic polymorphism, and other complicating factors all come together to make some regions difficult or impossible to assemble.
- **The best draft genomes will contain gaps and other imperfections.**
- Traditionally, draft genomes were upgraded to “phase 3 finished” using time-consuming and expensive Sanger-based manual finishing processes.
- An approach is implemented in **PBJelly** allowing to **perform gap closing on draft genomes using long-reads** from either the PacBio or Oxford Nanopore platforms.