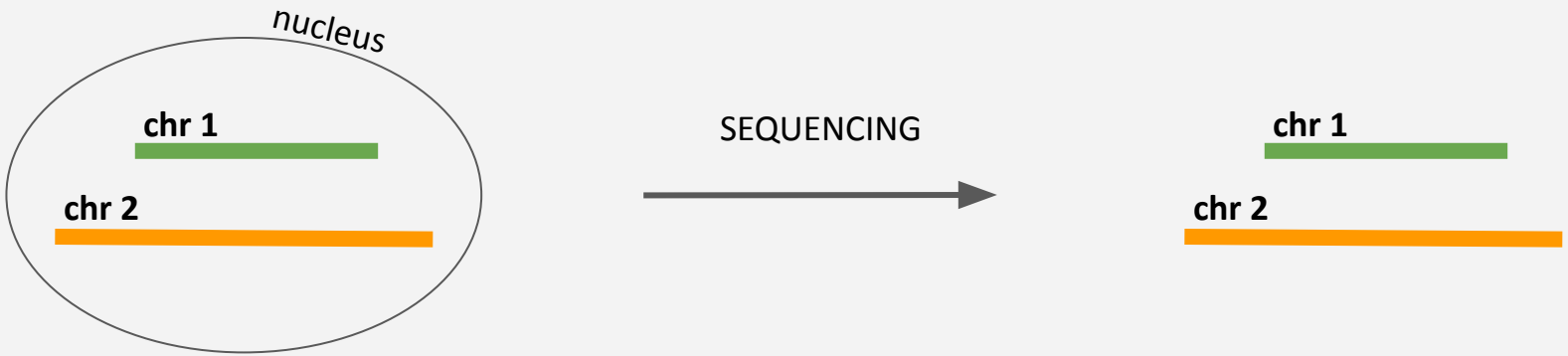
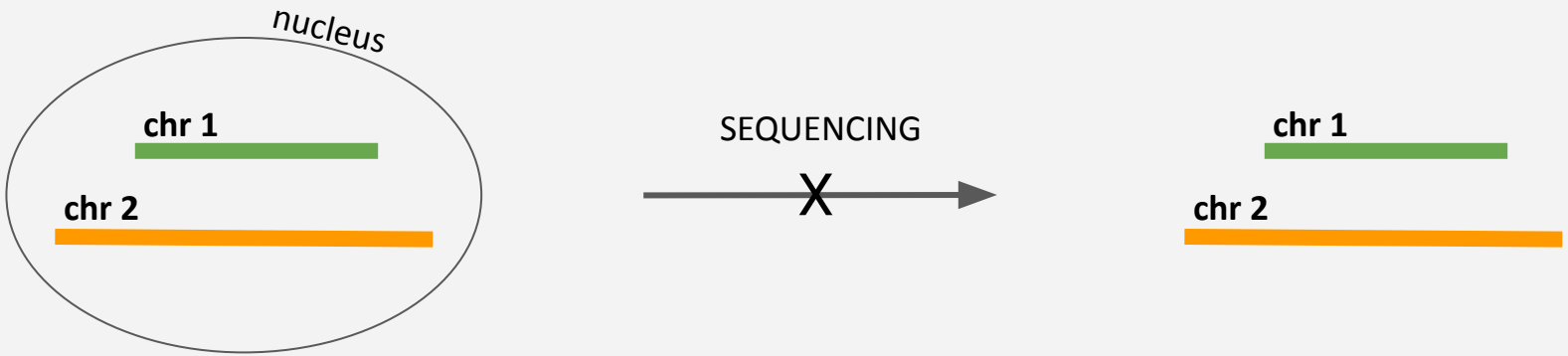
De novo genome assembly

Nadège Guiglielmoni

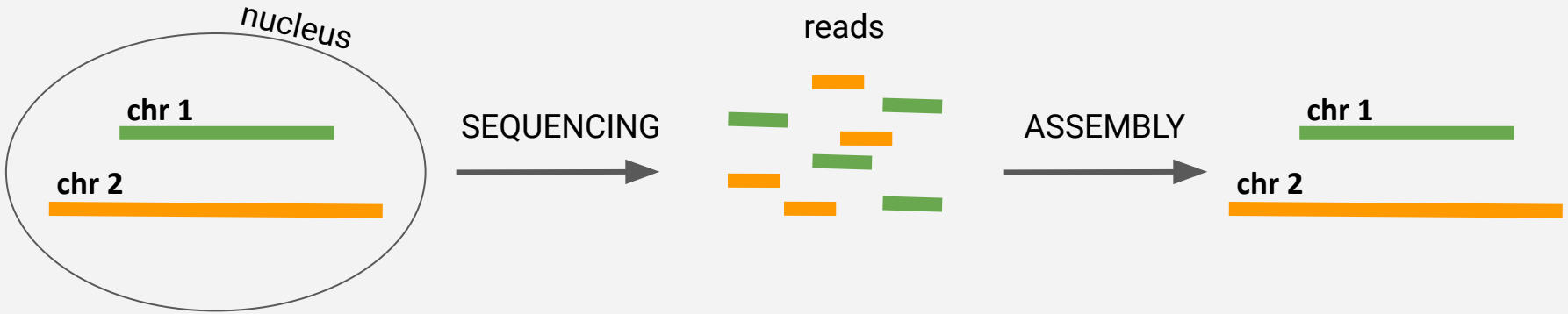
Introduction



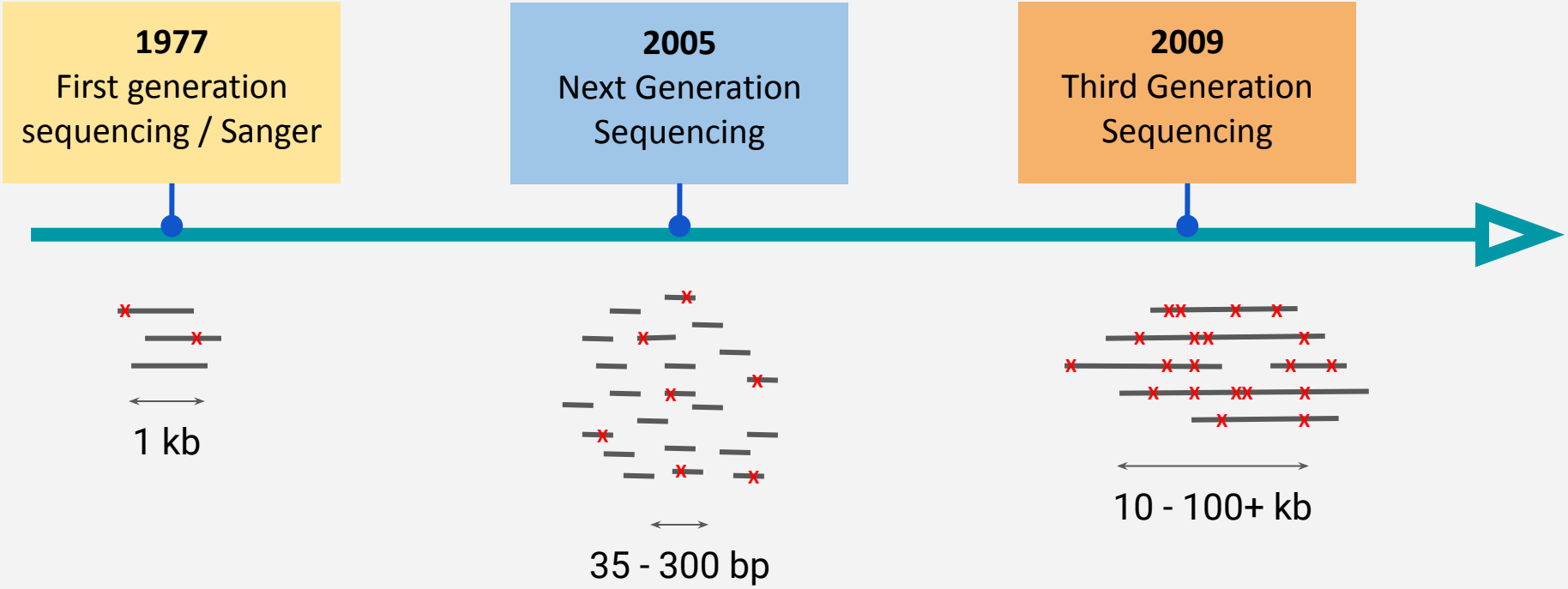
Introduction



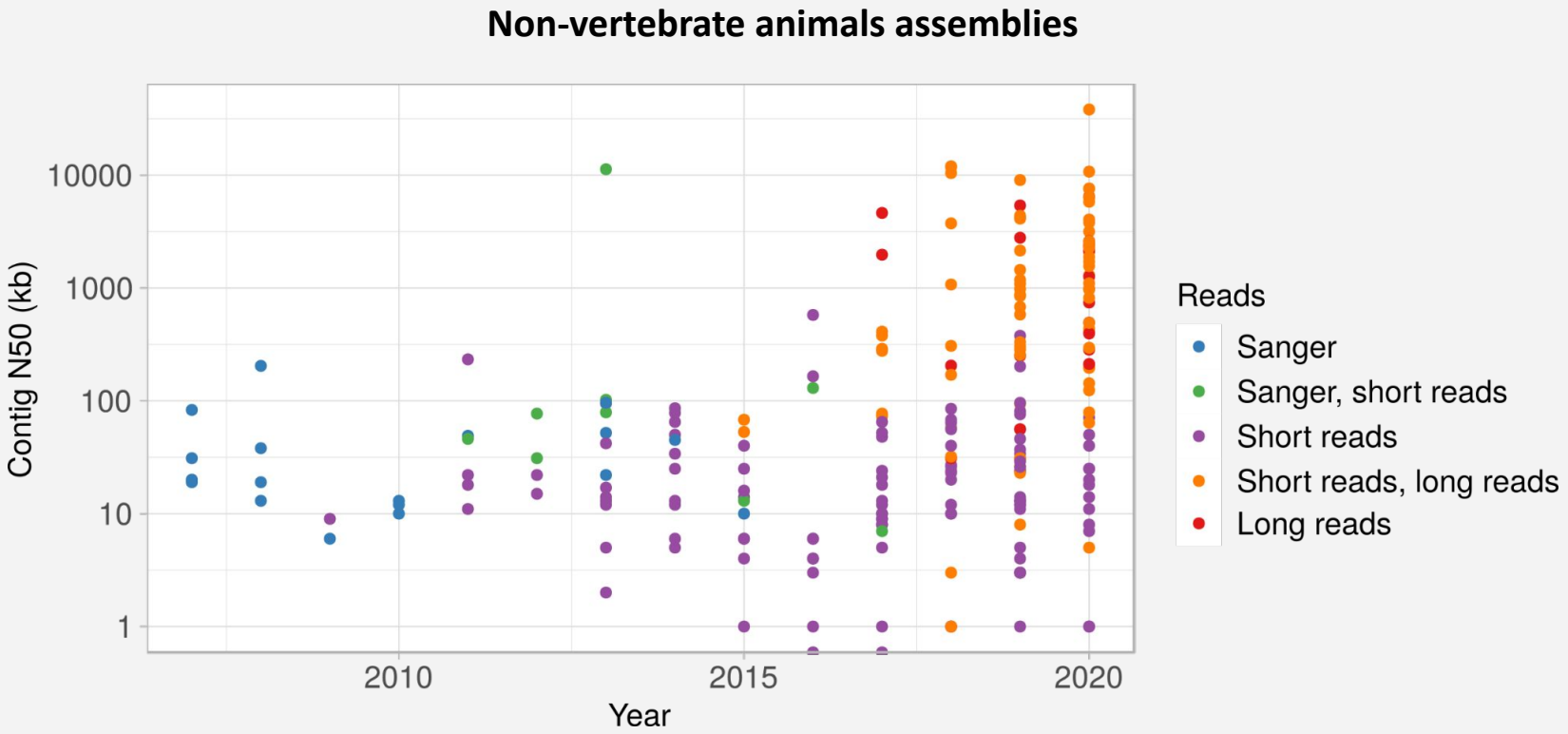
Introduction



Introduction



Introduction



Introduction

- Sequencing data
- Assembly algorithms
- Reads pre-processing
- Assembly post-processing
- Scaffolding approaches

Sequencing data

illumina

ATTTGTACGGACATAGTAAAAGCATGCGCGCCATGCGCGCAACTAG

TACGGACA

AAAAGCAT

CAGGCGCG

GCAACTAG

ATTTGTAC

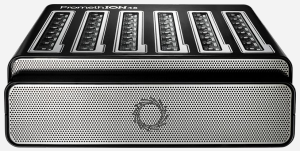
TATTAAAA

TGCGCGCC

CGCGCCAT

Length 35 - 700 base pairs
Low error rate

Sequencing data



ATTTGTACGGACATAGTAAAAGCATGCGCGCCATGCGCGCAACTAG

TGCGGACATAGTA--AGCATGC

ATTTTTGTACGGACAAAGTA--A

ATGCGCGCGCCATG

GCGTCATGC----AACTAG

A--AGCATGCGCGCGCCATGC----AACT

Length 10 - 100+ kilobases
High error rate

Sequencing data

PACBIO®

ATTTGTACGGACATAGTAAAAGCATGCGCGCCATGCGCGCAACTAG

TACCGGACATAGTA--AGCATCGC

ATTTTTGTACGGACAAAGTAAAA ATGCGCGCGCCAATG

GCG-CATGC----AACTAG

A--AGCATTGCGCGCCCCATGCGCGCAAACT

Length ~20 kilobases
High error rate

Sequencing data

PACBIO®

HiFi

ATTTGTACGGACATAGTAAAAGCATGCGCGCCATGCGCGCAACTAG

TACGGACATAGTAAAAGCATCGC

ATTTGTACGGACATAGTAAAA ATGCGCGCCATG

GCGCCATGCGCGCAACTAG

AAAAGCATGCGCGCCATGCGCGCAAACT

Length ~20 kilobases

Low error rate

Assembly algorithms

Assemble the reads: reconstitute, from the set of reads, the suite of bases (A,T,G and C) characteristic of this genome

De novo assembly: no reference

Source: http://www.langmead-lab.org/teaching-materials/

Assembly algorithms: overlap graphs

Overlap: length-*l* suffix of X matches length-*l* prefix of Y, where *l* is given

l = 3

X: CTCTAGGCC
Y: TAGGCCCTC

Assembly algorithms: overlap graphs

Overlap: length-*l* suffix of X matches length-*l* prefix of Y, where *l* is given

l = 3

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

Assembly algorithms: overlap graphs

Overlap: length-*l* suffix of X matches length-*l* prefix of Y, where *l* is given

l = 3

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

Assembly algorithms: overlap graphs

Overlap: length-*l* suffix of X matches length-*l* prefix of Y, where *l* is given

l = 3

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

Assembly algorithms: overlap graphs

Overlap: length-*l* suffix of X matches length-*l* prefix of Y, where *l* is given

l = 3

a length-6 prefix of Y matches a suffix of X

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

X: CTCTAGGCC
Y: TAGGCCCTC

Assembly algorithms: overlap graphs

Graph vocabulary:

- **vertex/node**
- **edge/arc**: connects vertices
- **directed graph** (digraph): set of vertices and directed edges

Overlap graph = directed graph

- nodes = reads
- edges = overlaps betwen reads

Assembly algorithms: overlap graphs

Vertices (reads): { *a:* CTCTAGGCC, *b:* GCCCTCAAT, *c:* CAATTTTT }

Edges (overlaps): { (*a*, *b*), (*b*, *c*) }


```
CTCTAGGCC          GCCCTCAAT
   |||                |||||
   GCCCTCAAT          CAATTTTT
```

Assembly algorithms

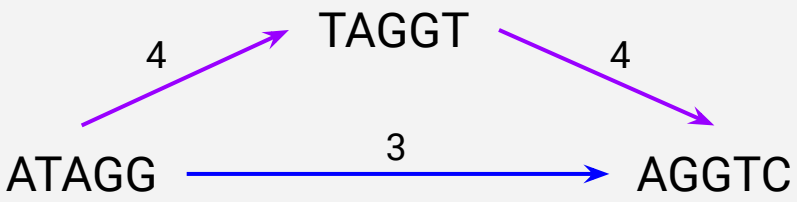
- **Greedy approach:** find the shortest common superstring by merging the best overlapping reads
- **Overlap-Layout-Consensus:** disantangle repeats by looking for the shortest generalised Hamiltonian path in the overlap graph
- **de Bruijn graphs**

Assembly algorithms: Overlap-Layout-Consensus

Overlap-Layout-Consensus (OLC): disantangle repeats by looking for the shortest generalised Hamiltonian path in the overlap graph

Hamiltonian path: goes through each node once

Layout: simplify the graph by removing redundant edges



Assembly algorithms: Overlap-Layout-Consensus

Consensus step

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

Take reads that make up a contig and line them up

```
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

Take *consensus*, i.e. majority vote

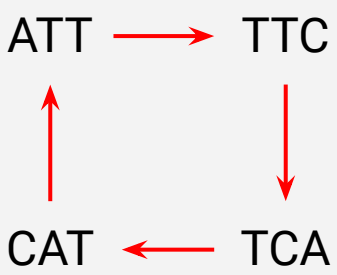
Assembly algorithms: de Bruijn graphs

de Bruijn graphs (DBG): connects words of a *k* length with (*k*-1)-long overlaps

k-mers: *k*-length words in a highly accurate genomic dataset (Illumina, HiFi)

node centric DBG: *k*-mers are nodes, overlaps are edges

edge centric DBG: overlaps are nodes, *k*-mers are edges



Assembly algorithms: de Bruijn graphs

Eulerian path: goes through every edge once

Figure 1 Bridges of Königsberg problem. (**a**) A map of old Königsberg, in which each area of the city is labeled with a different color point. (**b**) The Königsberg Bridge graph, formed by representing each of four land areas as a node and each of the city's seven bridges as an edge.

How to apply de Bruijn graphs to genome assembly, Compeau et *al.*, 2011

Assembly algorithms: de Bruijn graphs

k = 4

ATTATAT CGCGTAC ATTGCGC GCATTAT ACGGCGC

TATATTG GTACGGC GCGTACG ATATTGC

Assembly algorithms: de Bruijn graphs

k = 4 ATTATAT CGCGTAC ATTGCGC GCATTAT ACGGCGC

TATATTG GTACGGC GCGTACG ATATTGC

ATTA→TTAT→TATA→ATAT

CGCG→GCGT→CGTA→GTAC

ATTG→TTGC→TGCG→GCGC

GCAT→CATT→ATTA→TTAT

ACGG→CGGC→GGCG→GCGC

TATA→ATAT→TATT→ATTG

GTAC→TACG→ACGG→CGGC

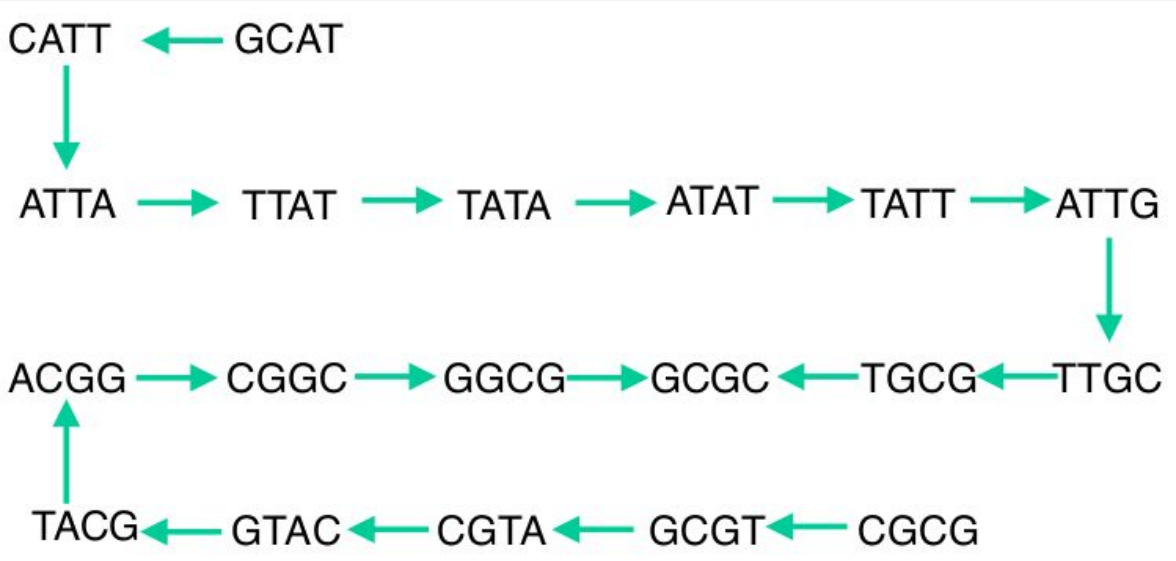
GCGT→CGTA→GTAC→TACG

ATAT→TATT→ATTG→TTGC

Assembly algorithms: de Bruijn graphs

k = 4 ATTATAT CGCGTAC ATTGCGC GCATTAT ACGGCGC

TATATTG GTACGGC GCGTACG ATATTGC



Assembly algorithms: de Bruijn graphs

k = 4 ATTATAT CGCGTAC ATTGCGC GCATTAT ACGGCGC

TATATTG GTACGGC GCGTACG ATATTGC

After compaction:

GCATTATATTGCG → GCGC ← CGCGTACGGCG

Two unitigs: GCATTATATTGCGC, CGCGTACGGCGC

Reads pre-processing

Remove adaptors

- **Illumina**: fastqc, cutadapt, Trimmomatic…
- **Nanopore**: Nanopore tools, Porechop

Filtering

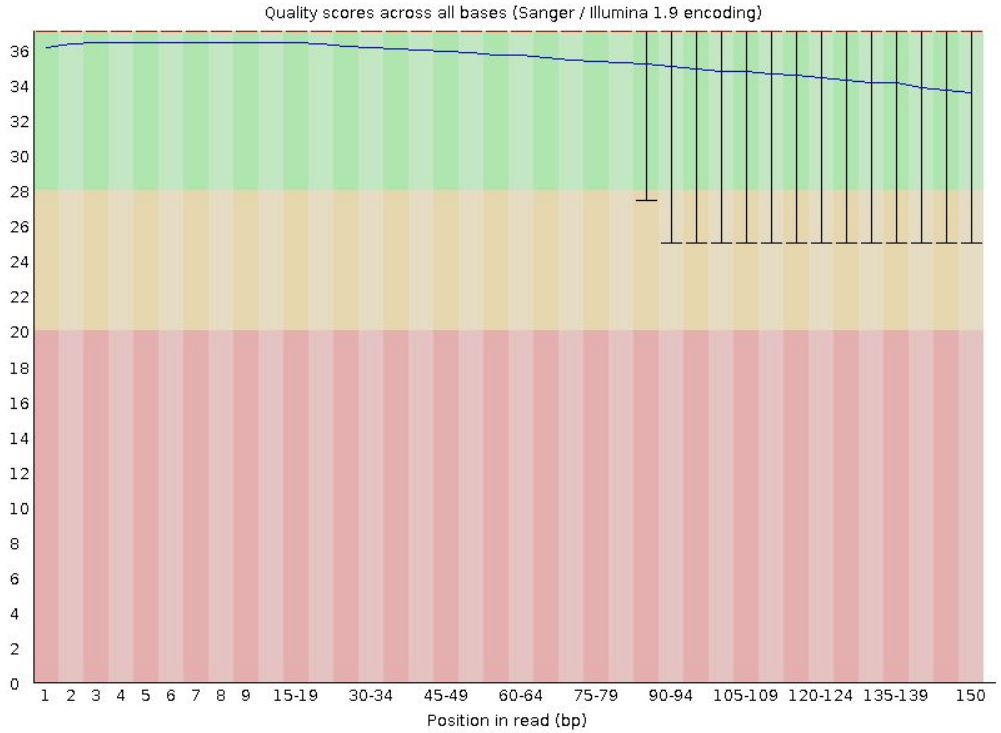
- **Long reads:** Filtlong

Reads pre-processing

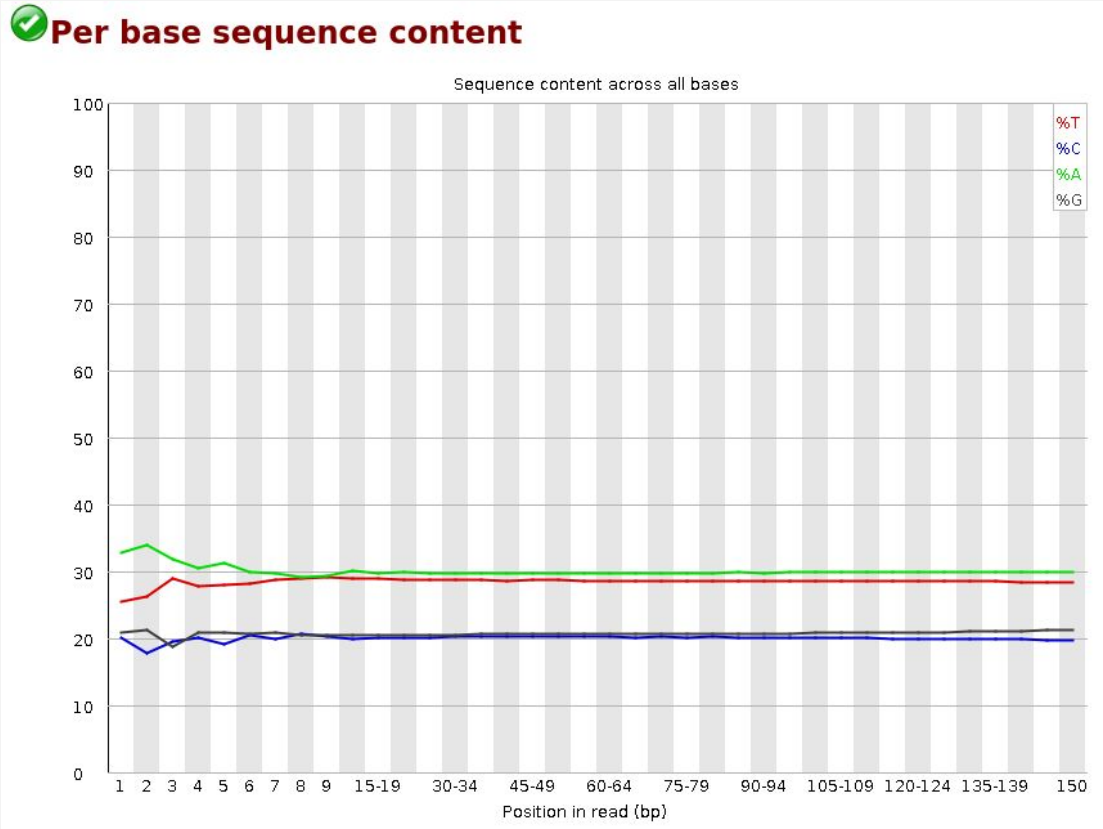
Summary

- Basic Statistics
- Per base 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
- Kmer Content

Per base sequence quality



Reads pre-processing

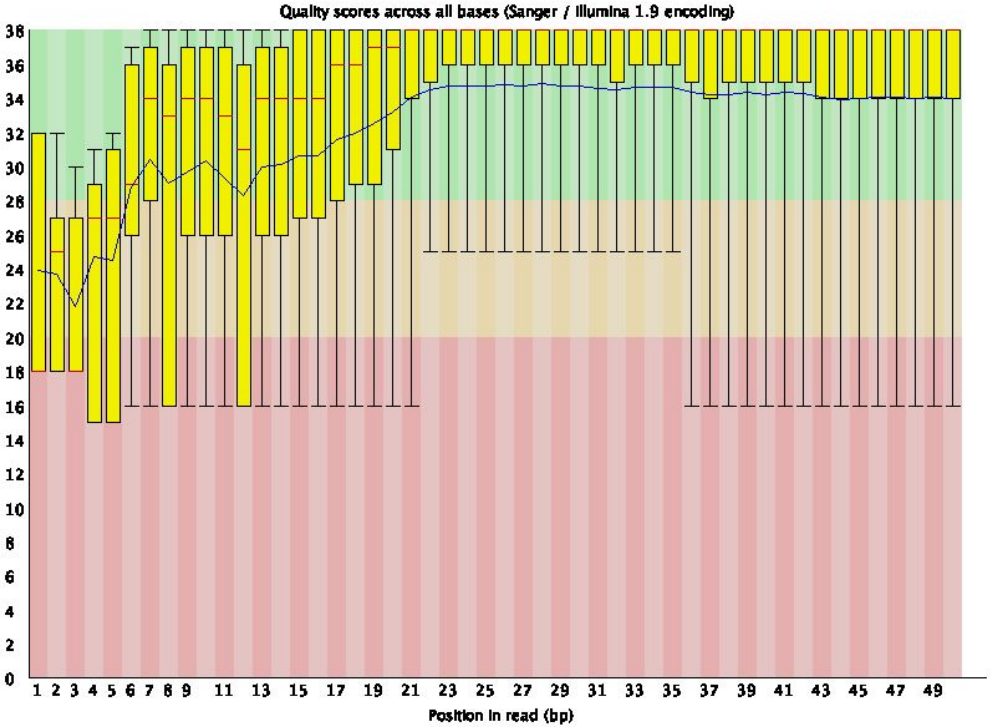


Reads pre-processing

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 quality



Reads pre-processing

Reads correction: reduce error rate of long reads

- **self correction:** long reads only

  Canu, NextDenovo, Daccord, CONSENT...

- **hybrid correction:** long reads & short reads

  Ratatosk, LoRDEC, CoLoRMAP, proovread…

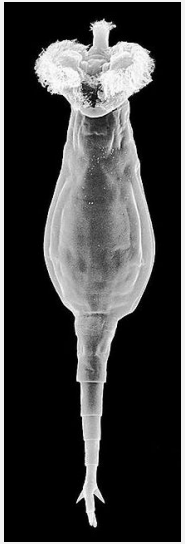
→ Less trending with high-accuracy long reads

Assembly post-processing

- **Polishing**: reduce errors
- **Haplotig purging**: remove uncollapsed haplotypes
- **Scaffolding**: increase contiguity
- **Gap filling**: find missing sequences

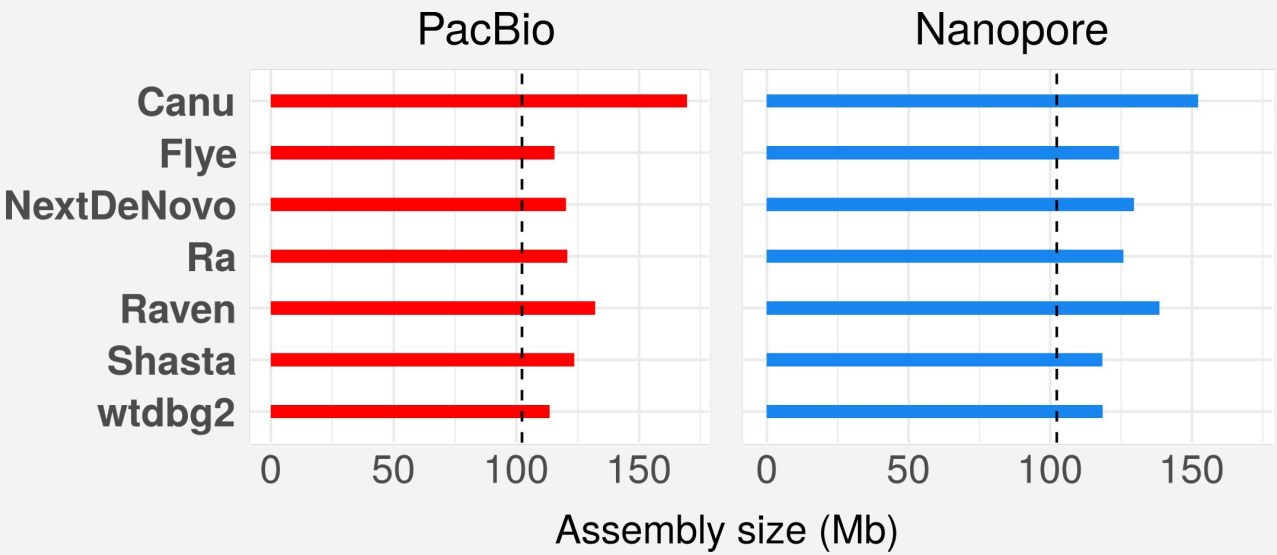
Assembly post-processing: haplotig purging

Adineta vaga



Who Needs Sex (or Males) Anyway?
Liza Gross, PloS Biology, 2007

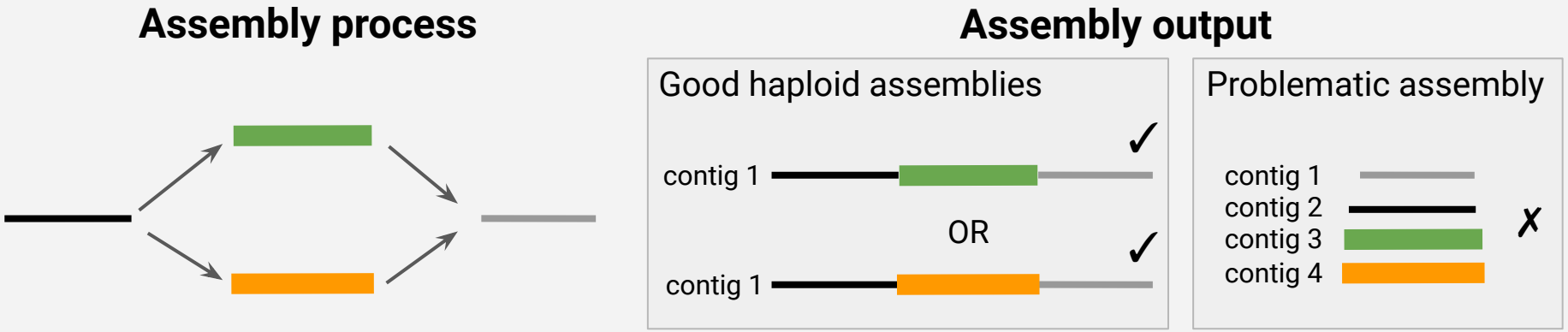
Expected haploid size 102 Mb



Assembly post-processing: haplotig purging

Haplotype 1 ATTACCAGTCTCAA**TGGATGGCTACTC**TTTGACGATAGCT

Haplotype 2 ATTACCAGTCTCAA**AGGCTGCTAGTG**TTTGACGATAGCT



Assembly post-processing: haplotig purging

HaploMerger2

HaploMerger2: rebuilding both haploid sub-assemblies from high-heterozygosity diploid genome assembly

Shengfeng Huang*, Mingjing Kang and Anlong Xu

Identifying and removing haplotypic duplication in primary genome assemblies

Dengfeng Guan[1,2], Shane A. McCarthy [2], Jonathan Wood[3], Kerstin Howe [3], Yadong Wang[1,*] and Richard Durbin [2,3,*]

purge_dups

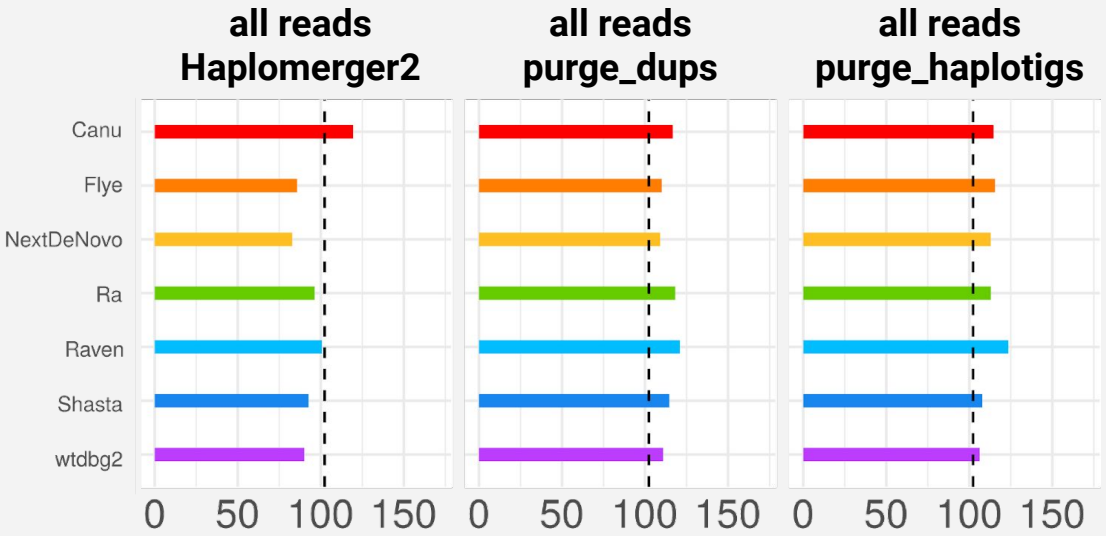
Purge Haplotigs

Purge Haplotigs: allelic contig reassignment for third-gen diploid genome assemblies

Michael J. Roach*, Simon A. Schmidt and Anthony R. Borneman

Assembly post-processing: haplotig purging

PacBio assemblies

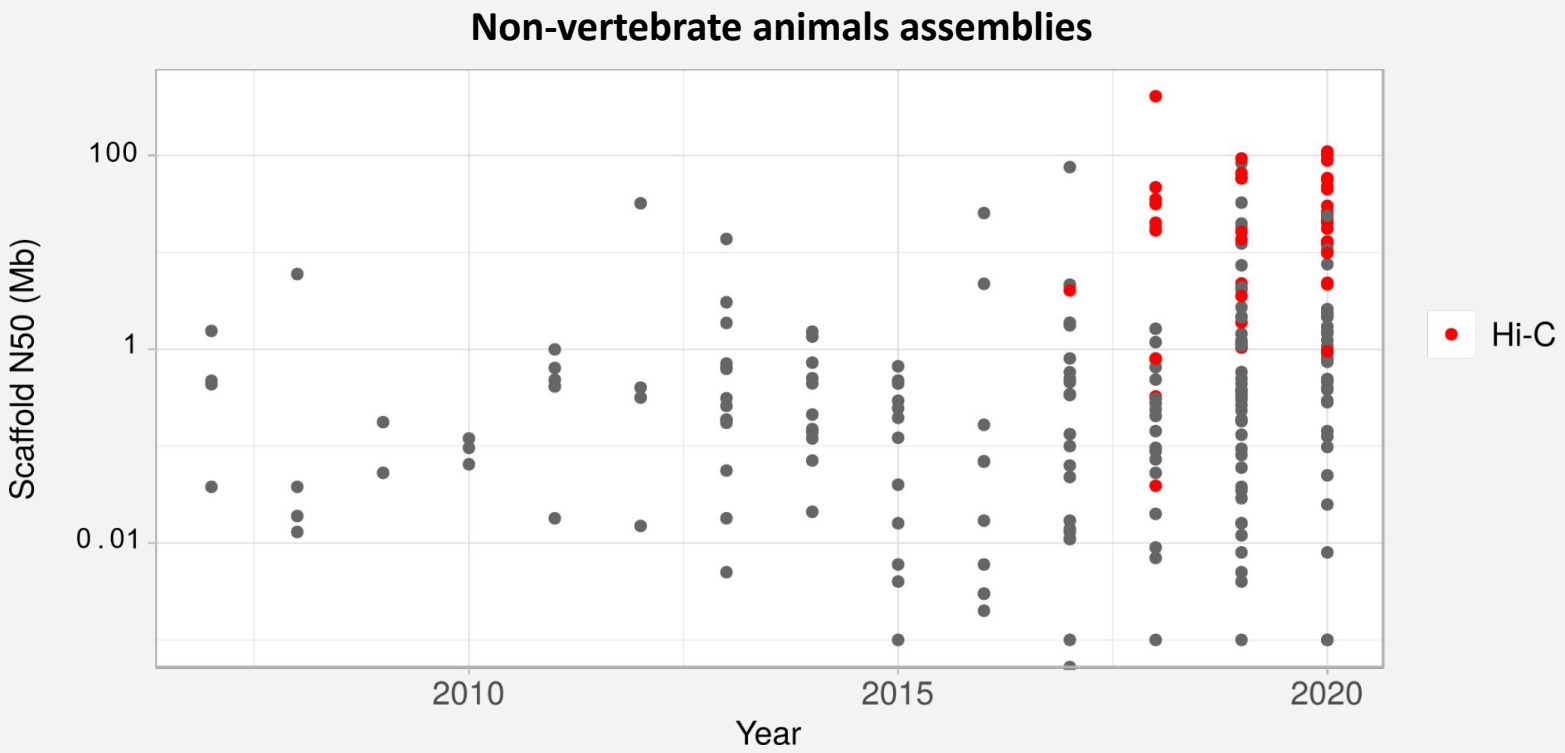


Scaffolding approaches

Scaffolding: grouping and orienting contigs to build chromosome-level scaffolds

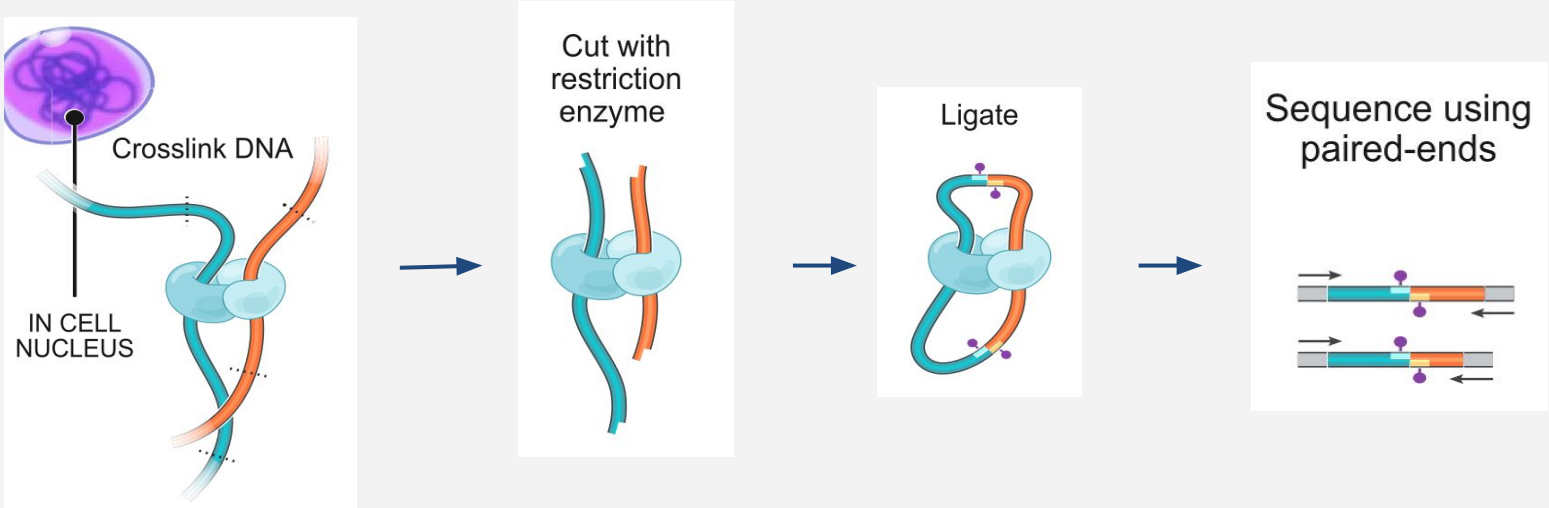
- **Mate-pairs**: short reads with a long insert
- **Long reads**
- **Genetic maps**: ordered markers
- **Optical maps**: ordered markers
- **Linked reads:** barcoded short reads
- **Hi-C/3C/Proximity ligation**

Scaffolding approaches: Hi-C scaffolding



Scaffolding approaches: Hi-C scaffolding

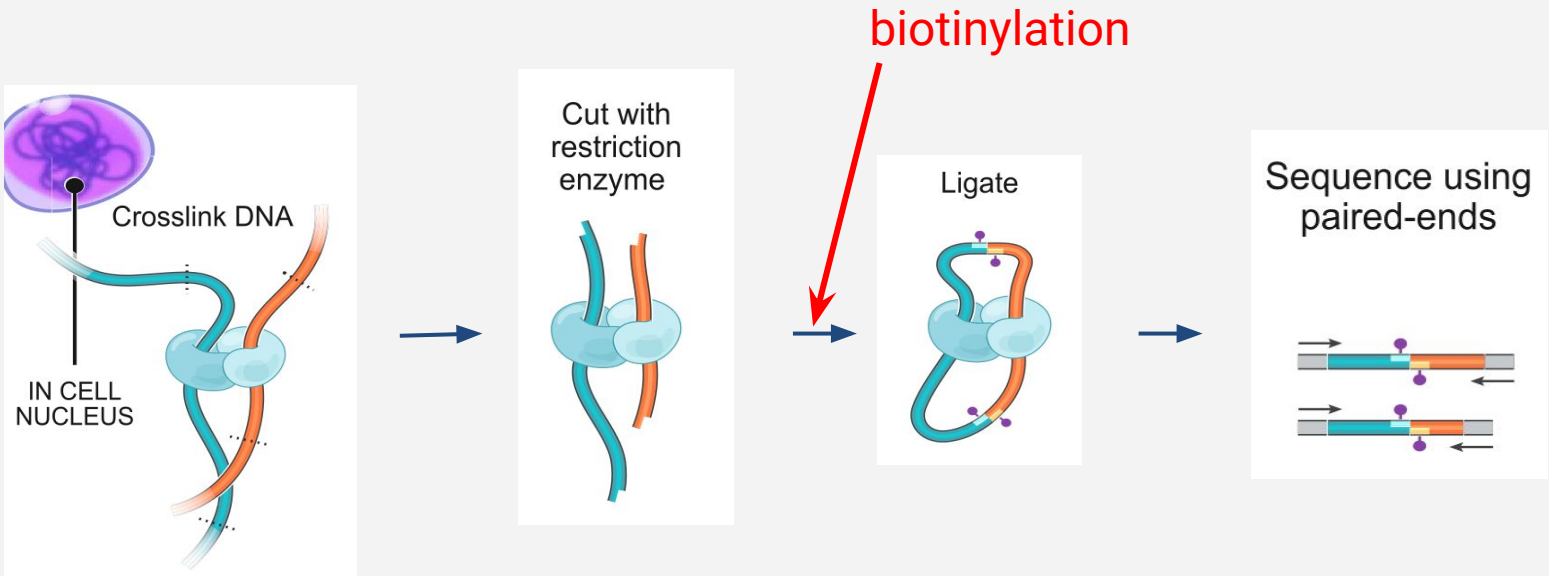
3C



A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Rao et *al.*, 2014

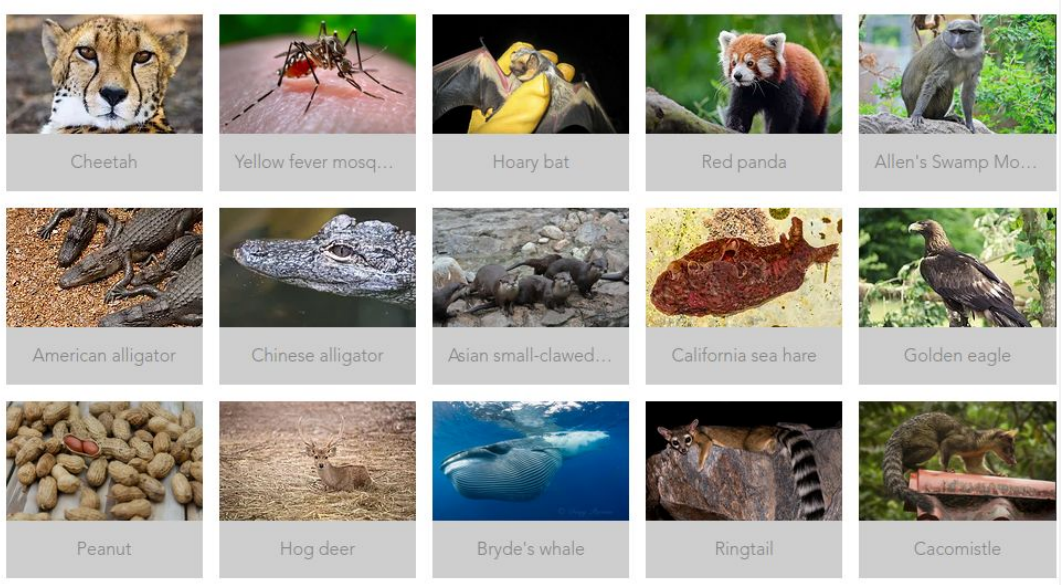
Scaffolding approaches: Hi-C scaffolding

Hi-C



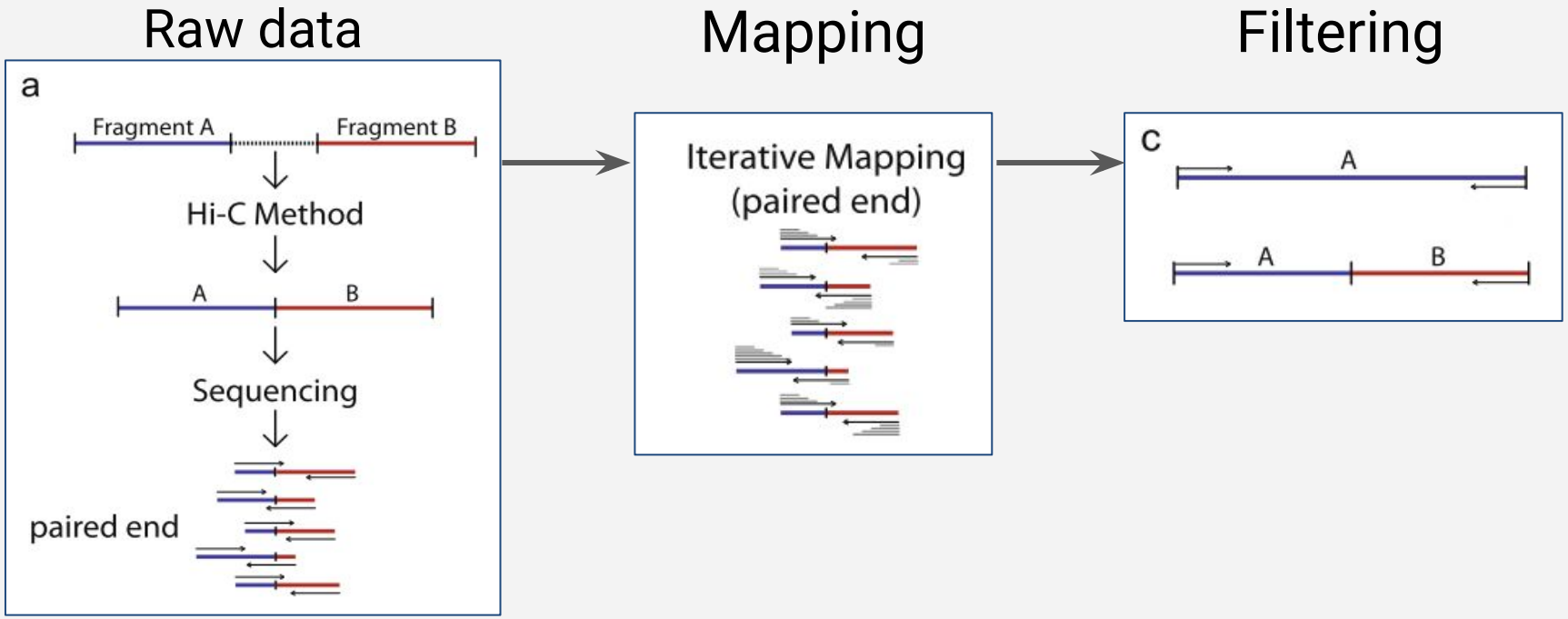
A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Rao et *al.*, 2014

Scaffolding approaches: Hi-C scaffolding



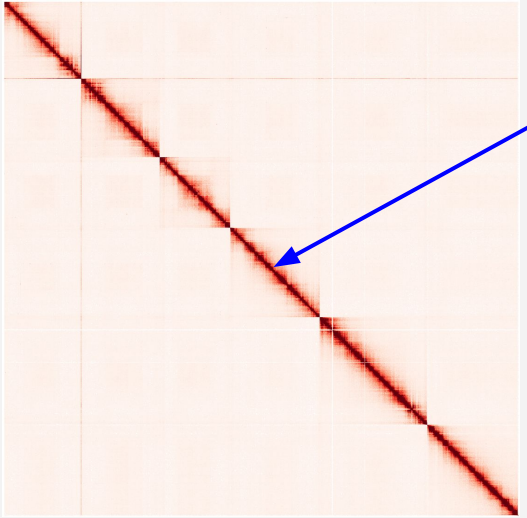
www.dnazoo.org

Scaffolding approaches: Hi-C scaffolding



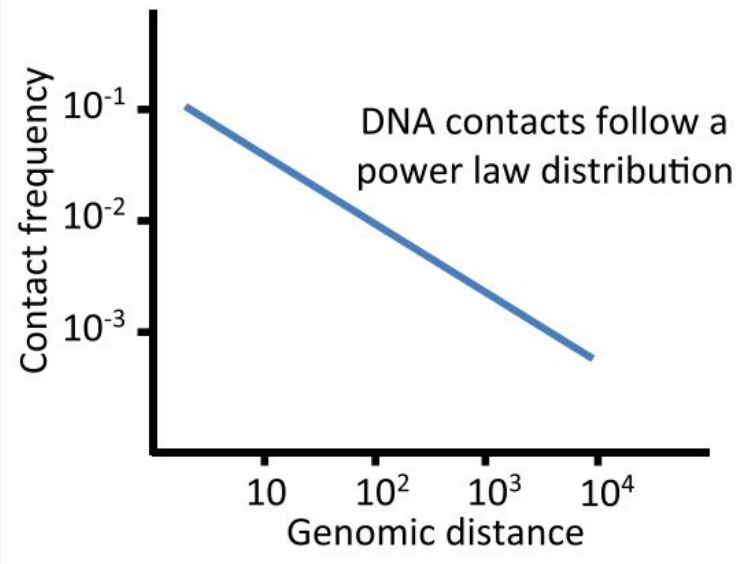
The Hitchhiker's guide to Hi-C analysis: Practical guidelines. Lajoie et *al.*, 2015

Scaffolding approaches: Hi-C scaffolding



Contact map of
Caenorhabditis elegans

contact frequency = f(genomic distance)

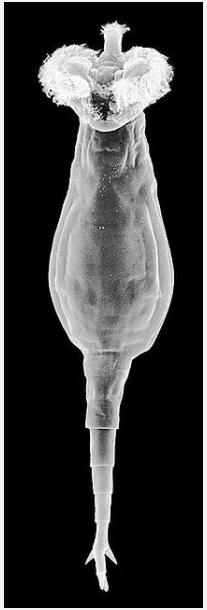


Contact genomics: scaffolding and phasing (meta)genomes using chromosome 3D physical signatures. Flot et *al.*, 2015

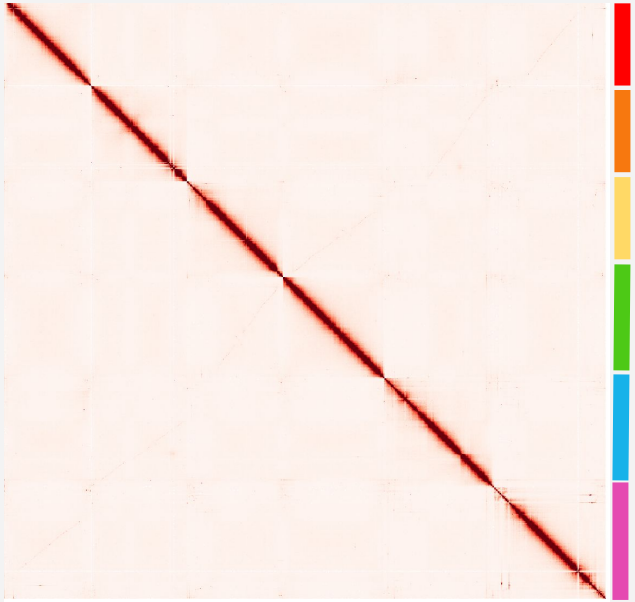
Scaffolding approaches: Hi-C scaffolding

Adineta vaga (rotifer)

6 scaffolds



Who Needs Sex (or Males) Anyway?
Liza Gross, PloS Biology, 2007



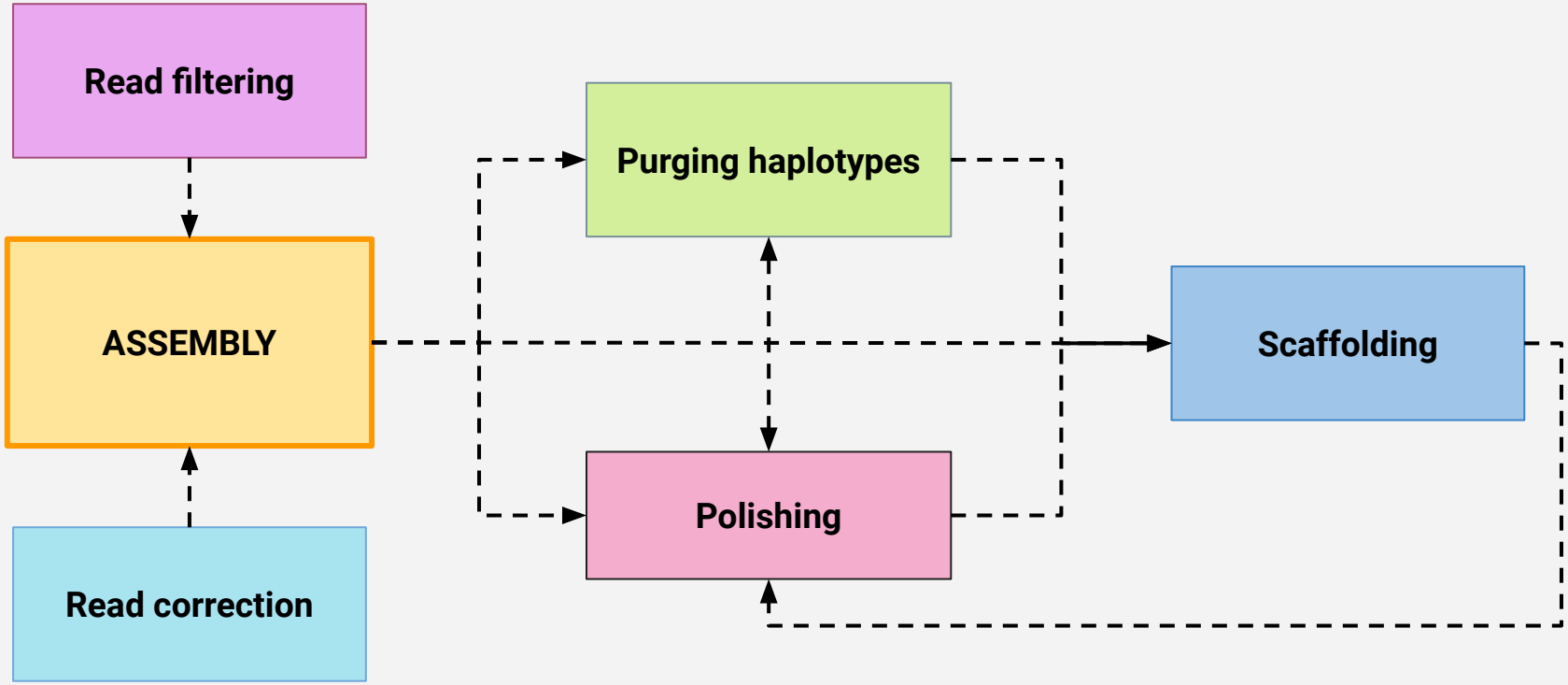
Hi-C contact map of *Adineta vaga*

Scaffolding approaches: gap filling

GATTCCGGAGACCTANNNNNNNNNNNNNNNNNNNNNNNNNNATATTTGTCAGAC

- Short reads: GapFiller, GAPPadder, Sealer
- Long reads: FGAP, GMCloser, LR_Gapcloser, PBJelly, PGcloser, TGS-GapCloser

Assembly pipeline



Thank you for your attention!
Questions?