NHMW Workshop IV

De novo genome assembly

Martin Kapun & Andreas Kroh

Program for today

Concepts - before lunch (ca. 90')

- The basics of NGS
- Types of assembly algorithms
- Assembly workflow
- Quality control

Program for today

Concepts - before lunch (ca. 90')

- The basics of NGS
- Types of assembly algorithms
- Assembly workflow
- Quality control

Hands-on - after lunch (ca. 120')

- Raw data QC
- Estimating the genome size
- Assembly with SPAdes and Flye
- Assembly QC

Genome Assembly

A genome assembly is an attempt to accurately represent an entire genome sequence from a large set of short DNA sequences.

.

Source: Keith Bradnam

Genome Assembly

A genome assembly is **an attempt** to accurately represent an entire genome sequence from a large set of short DNA sequences.

.

Source: Keith Bradnam

Genome Assembly

A genome assembly is an attempt to accurately **represent an entire genome sequence** from a large set of short DNA sequences.

.

Source: Keith Bradnam

Genome Assembly

A genome assembly is an attempt to accurately represent an entire genome **sequence from a large set** of short DNA sequences.

.

Source: Keith Bradnam

Genome Assembly

A genome assembly is an attempt to accurately represent an entire genome sequence from a large set of **short** DNA sequences.

.

Source: Keith Bradnam

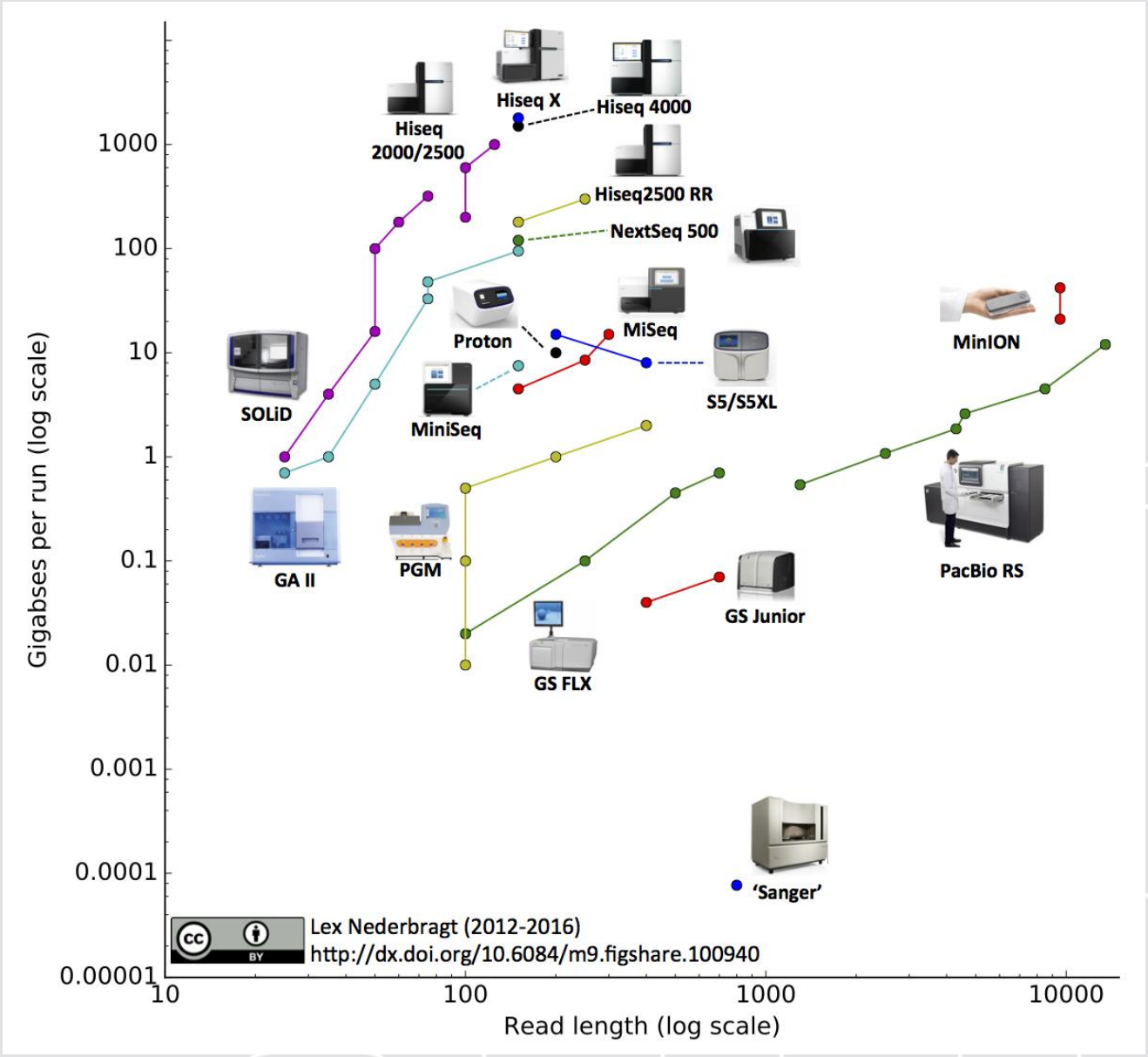
Genome Assembly

A genome assembly is an attempt to accurately represent an entire genome sequence from a large set of short DNA sequences.

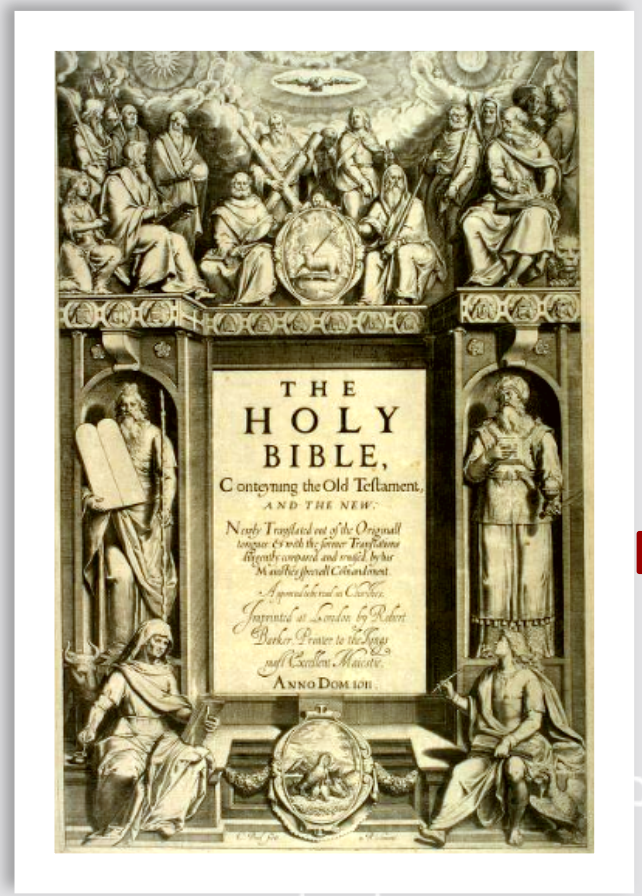
It's a bit like trying to do the hardest jigsaw puzzle you can imagine!

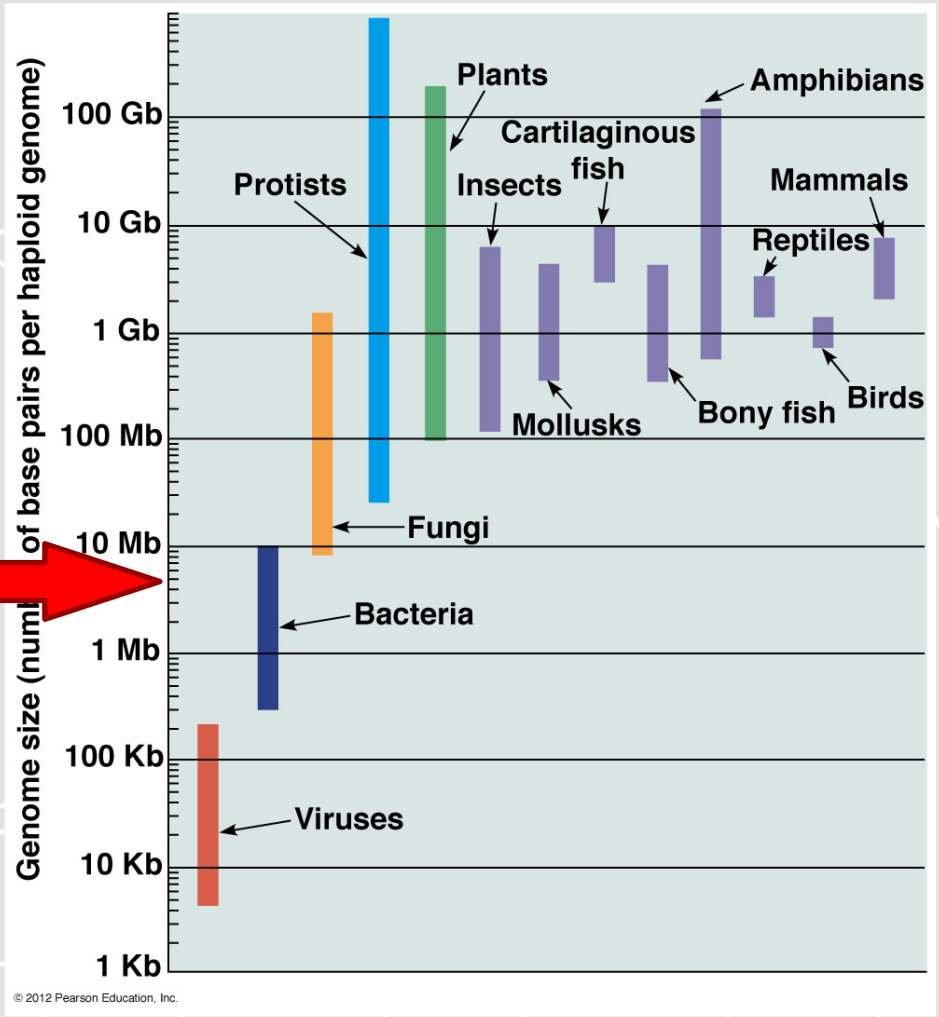
Source: Keith Bradnam

New Sequencing technologies



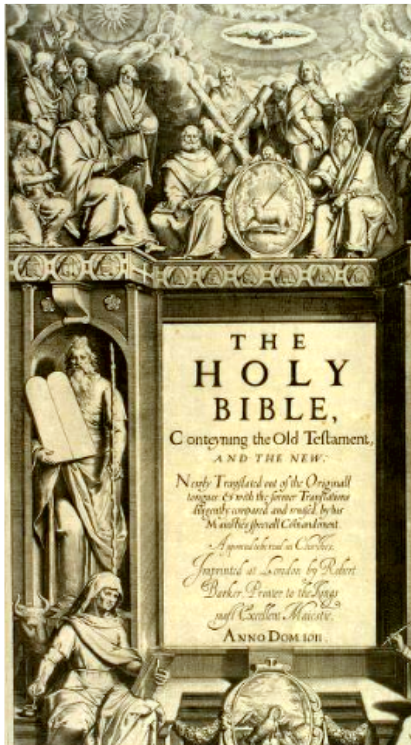
Genome-size variation





Next Generation Sequencing

The idea behind it



The Holy Bible: *c.* 3.5 million characte.

Next Generation Sequencing

The idea behind it

The Holy Bible: *c.* 3.5 million characters (excl. spaces) x 500,000 (!)

Next Generation Sequencing

The idea behind it

Next Generation Sequencing

The idea behind it

GAATTCTTCAGGTAGCTTCCTAGGGTTTCCAAGGCAATACAA

AGGTAGCTTCCTAGGGTTTCCAAGGCAATACAAGAAGAATTTT

TTCTTCAGGTAGCTTCCTAGGGTTTCCAAGG

Problem:

- numerous fragments of the same genomic region exist

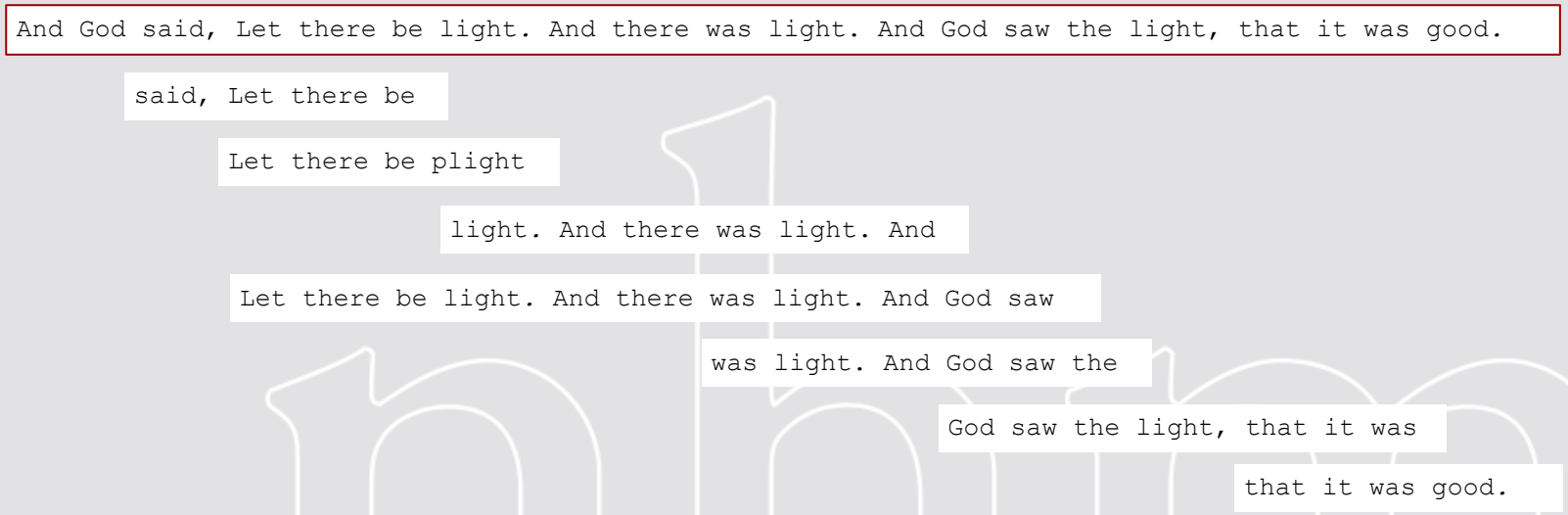
Benefit:

- numerous fragments of the same genomic region exist

What to do with these data?

The idea behind it

Reference guided assembly (read mapping/resequencing)



What to do with these data?

The idea behind it

Reference guided assembly (read mapping/resequencing)

De-novo assembly

Read types

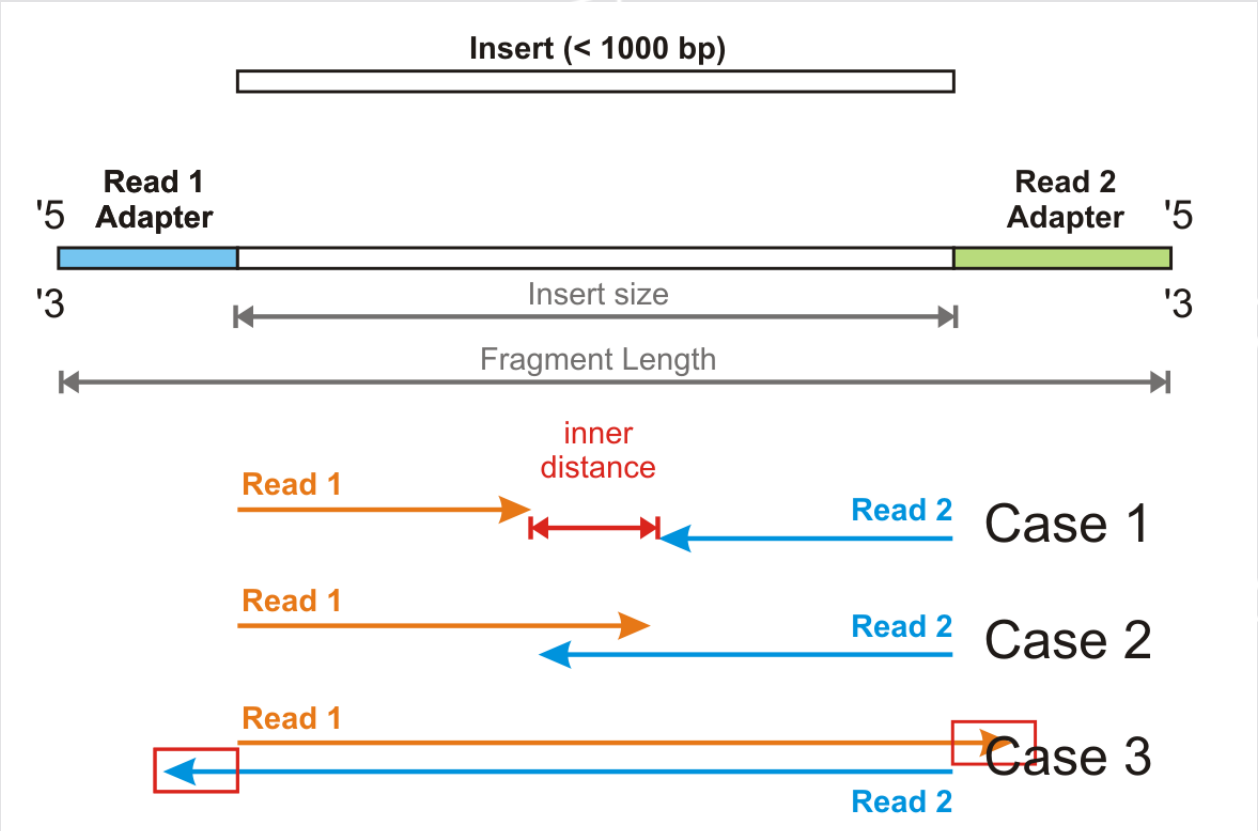
Single-End vs. Paired-End

Figure 4. Paired-End Sequencing and Alignment

Paired-end sequencing enables both ends of the DNA fragment to be sequenced. Because the distance between each paired read is known, alignment algorithms can use this information to map the reads over repetitive regions more precisely. This results in much better alignment of the reads, especially across difficult-to-sequence, repetitive regions of the genome.

Read types

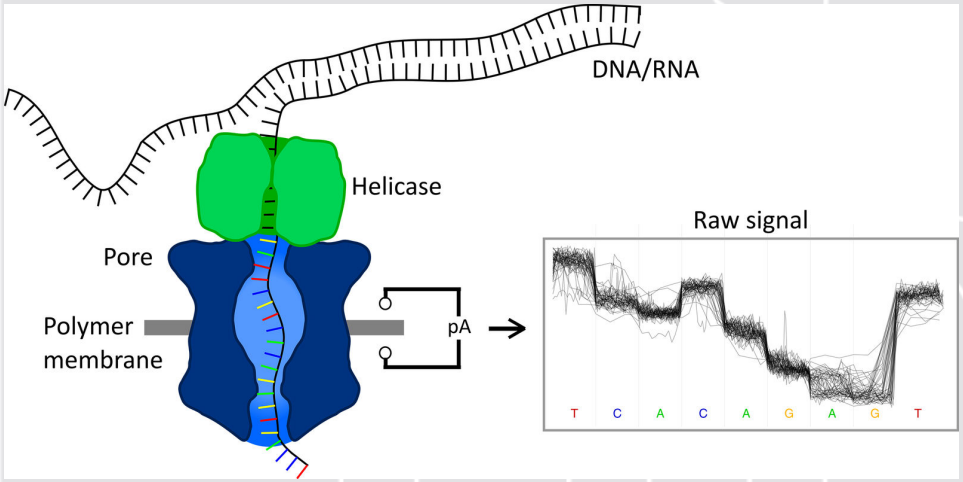
Single-End vs. **Paired-End**



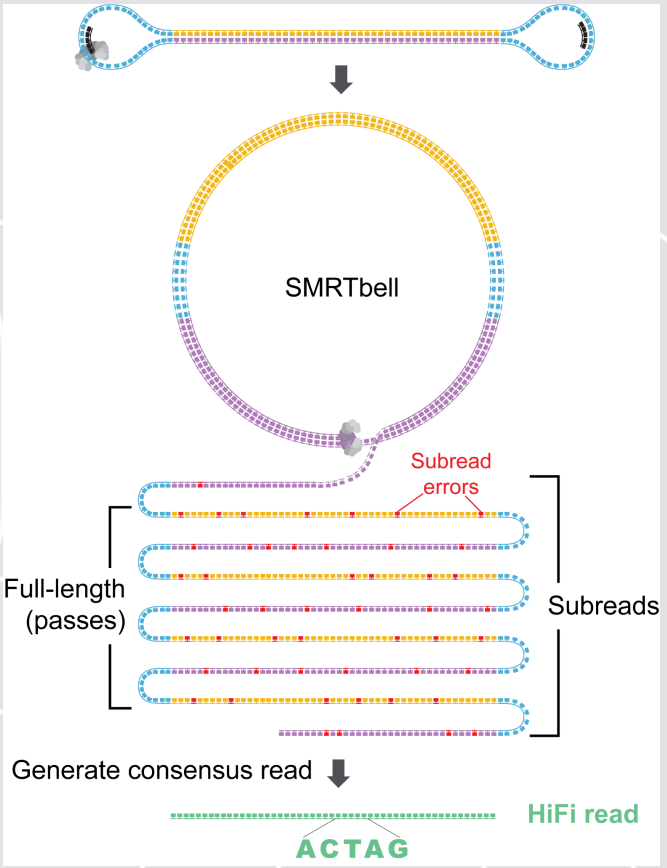
Read types

Single-End vs. Paired-End vs. **Single Molecule**

Oxford Nanopore



Pacific Biosciences



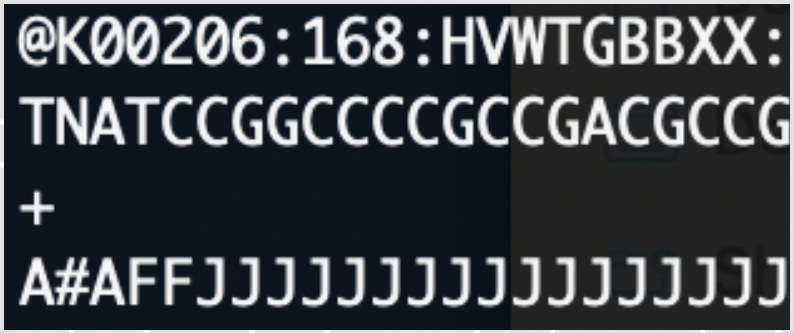
Next Generation Sequencing

Read technologies

Raw data

- (FASTA)
- FASTQ

Header (starting with an @)
Sequence
Second header
Quality string (PHRED)



Next Generation Sequencing

Read Quality

Phred Quality Scores

000		(nul)	016	▶	(dle)	032	sp	048	0	064	@	080	P	096	`	112	p
001	☺	(soh)	017	◀	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002	☻	(stx)	018	↕	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	♥	(etx)	019	‼	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	♦	(eot)	020	¶	(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005	♣	(enq)	021	§	(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	♠	(ack)	022	▬	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	•	(bel)	023	↨	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	◘	(bs)	024	↑	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009		(tab)	025	↓	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010		(lf)	026		(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂	(vt)	027	←	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012	♀	(np)	028	∟	(fs)	044	,	060	<	076	L	092	\	108	l	124	|
013		(cr)	029	↔	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014	♫	(so)	030	▲	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	☼	(si)	031	▼	(us)	047	/	063	?	079	O	095	_	111	o	127	⌂

- quality string: "?" **(ASCII:** 63)
- PHRED: 63 - 33 = 30
- error prob.: 10^-(30/10) -> 0.001
- 1/0.001 = 1000, which means that every 1000th base with a PHRED score of 30 is wrongly scored

Next Generation Sequencing

Read Quality

Phred Quality Scores

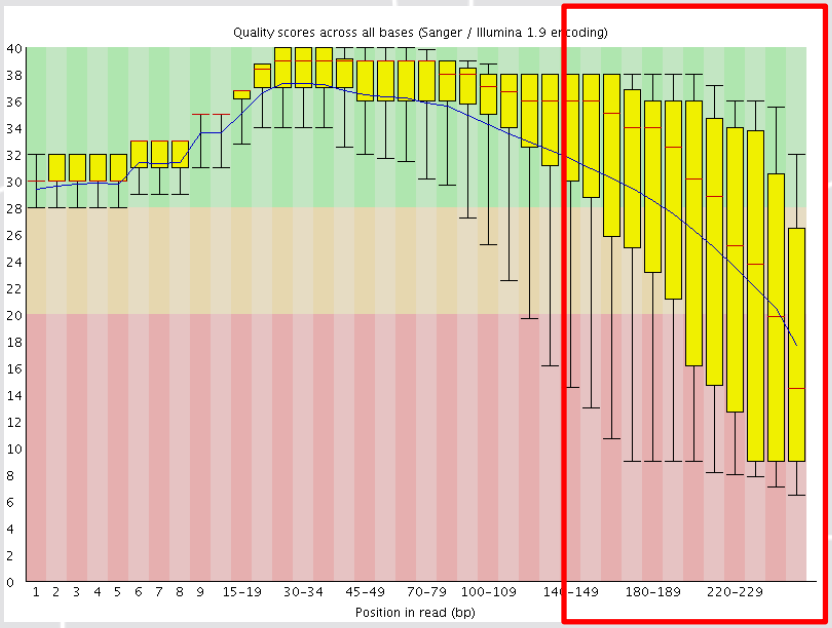
Table 1: Quality Scores and Base Calling Accuracy

Phred Quality Score	Probability of Incorrect Base Call	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1,000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%

Data Quality

Short reads

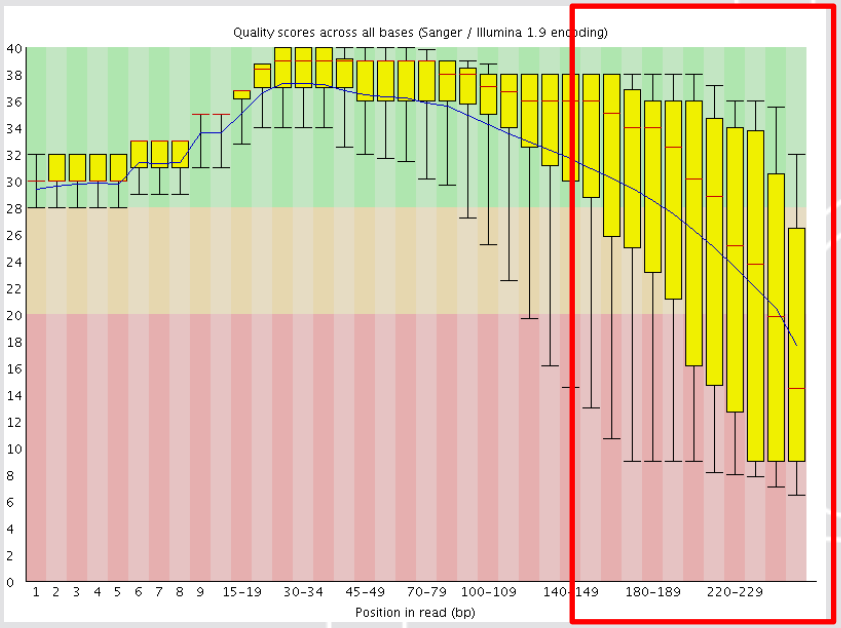
150 bp - 250 bp

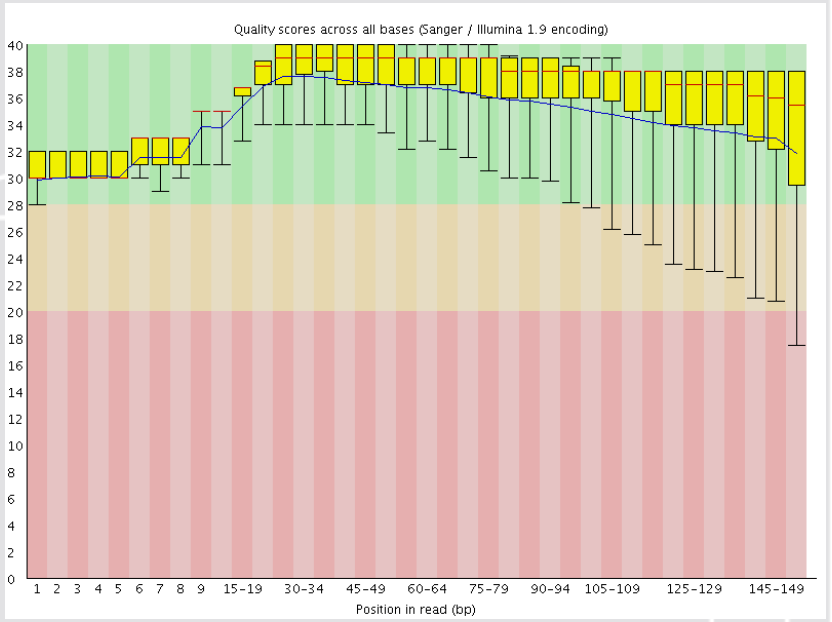


Data Quality

Short reads

Quality trimming (removal of low qual bases; adapters)

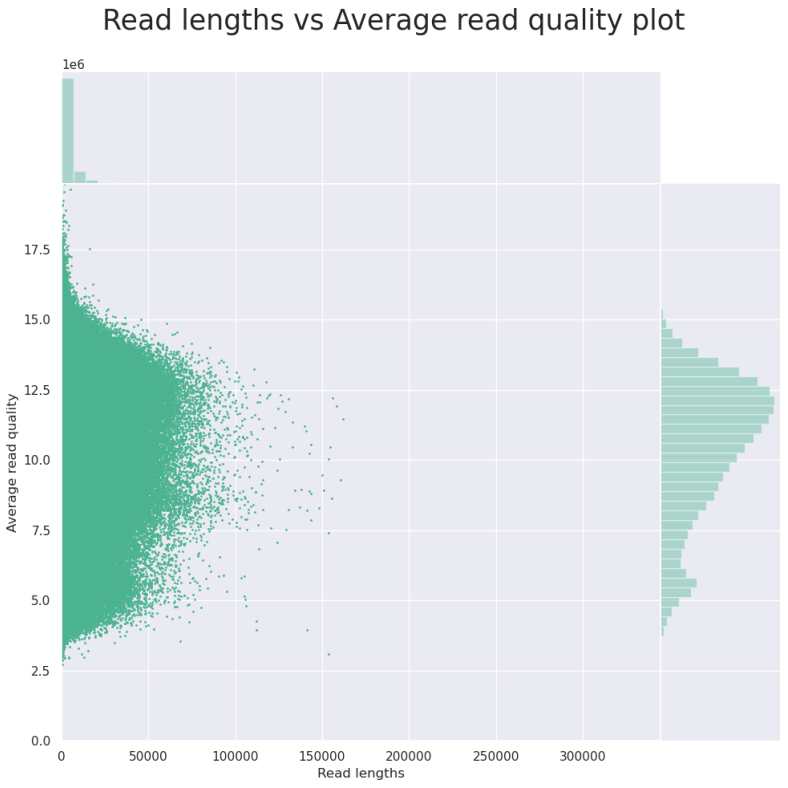




Data Quality

Long reads - Oxford Nanopore

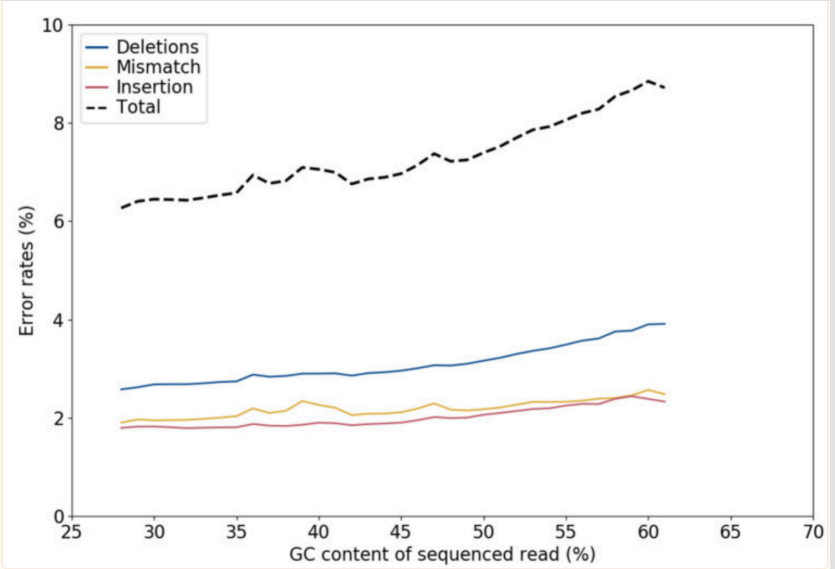
< 200,000bp

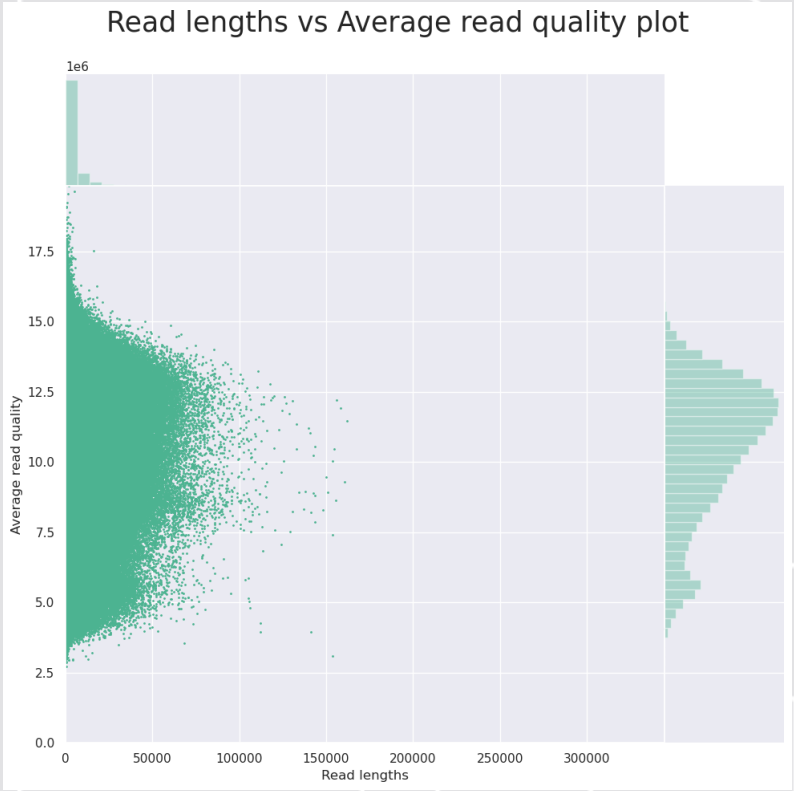


Data Quality

Long reads - Oxford Nanopore

< 200,000bp



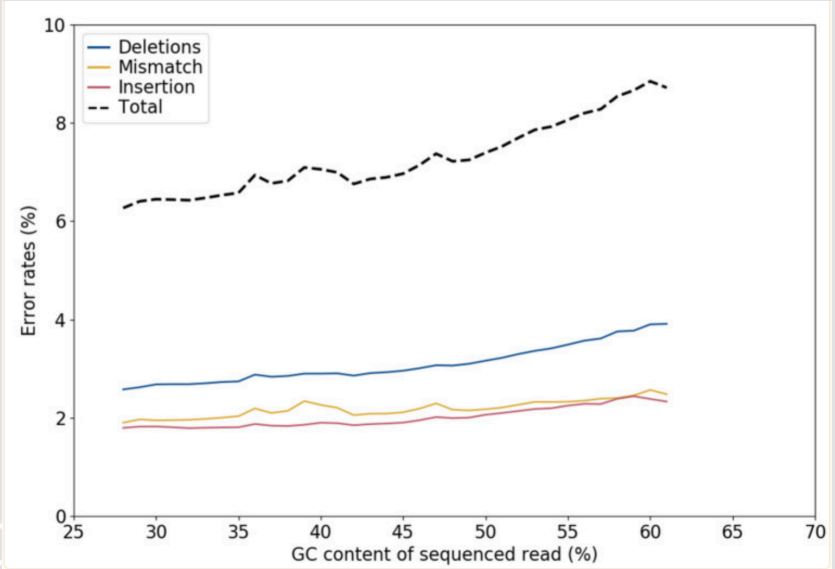


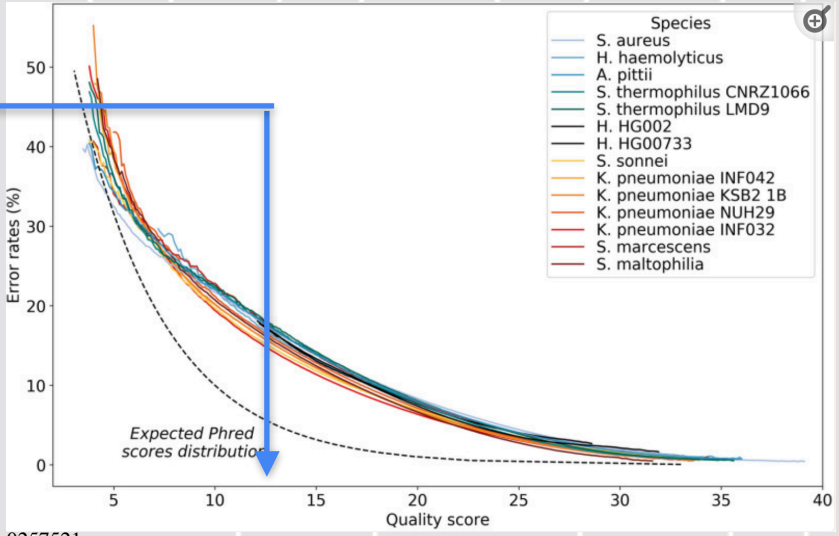
Delahaye, C. & Nicolas, J. 2021. Sequencing DNA with nanopores: Troubles and biases. *PLoS One* **16**: e0257521.

Data Quality

Long reads - Oxford Nanopore

< 200,000bp



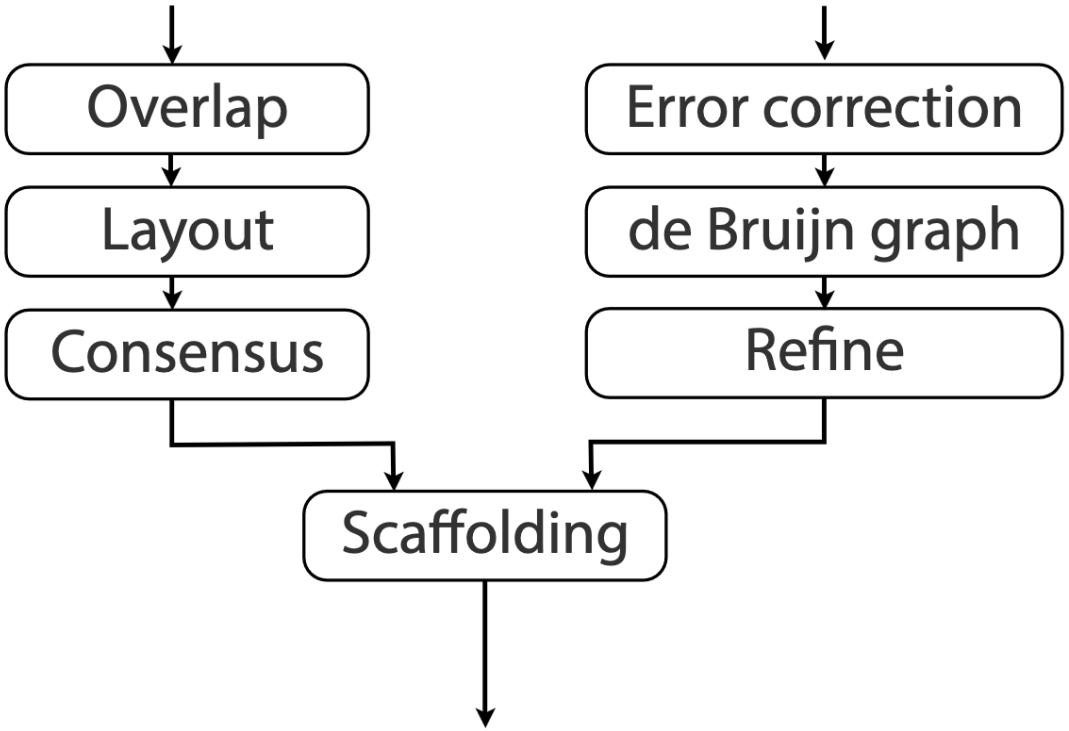


Delahaye, C. & Nicolas, J. 2021. Sequencing DNA with nanopores: Troubles and biases. *PLoS One* **16**: e0257521.

Assembly algorithms

Alternative 1: Overlap-Layout-Consensus (OLC) assembly

Alternative 2: de Bruijn graph (DBG) assembly

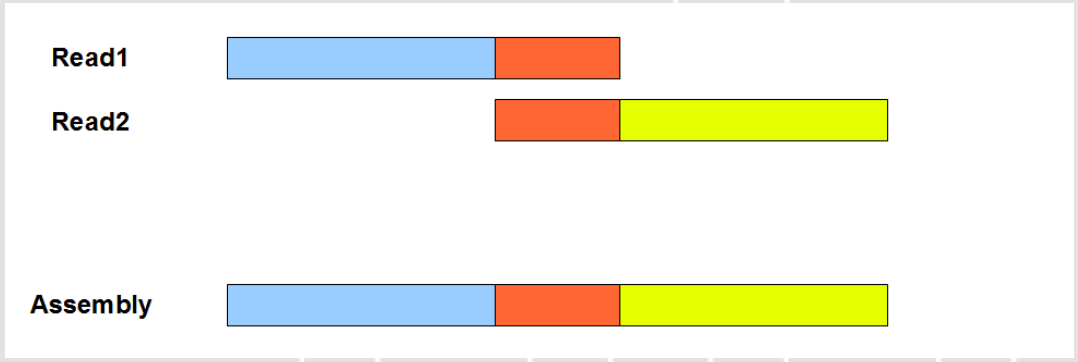


https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

Assembly algorithms

(1) Overlap-consensus-layout (OCL) method

- e.g. *Flye* assembler (long reads)



And God called the firmament Heaven

and the gathering together

And God said, Let the earth

the gathering together of

And God said, Let there be

gathering together of the

And God called the dry

together of the waters

And God saw that it was good.

of the waters he called

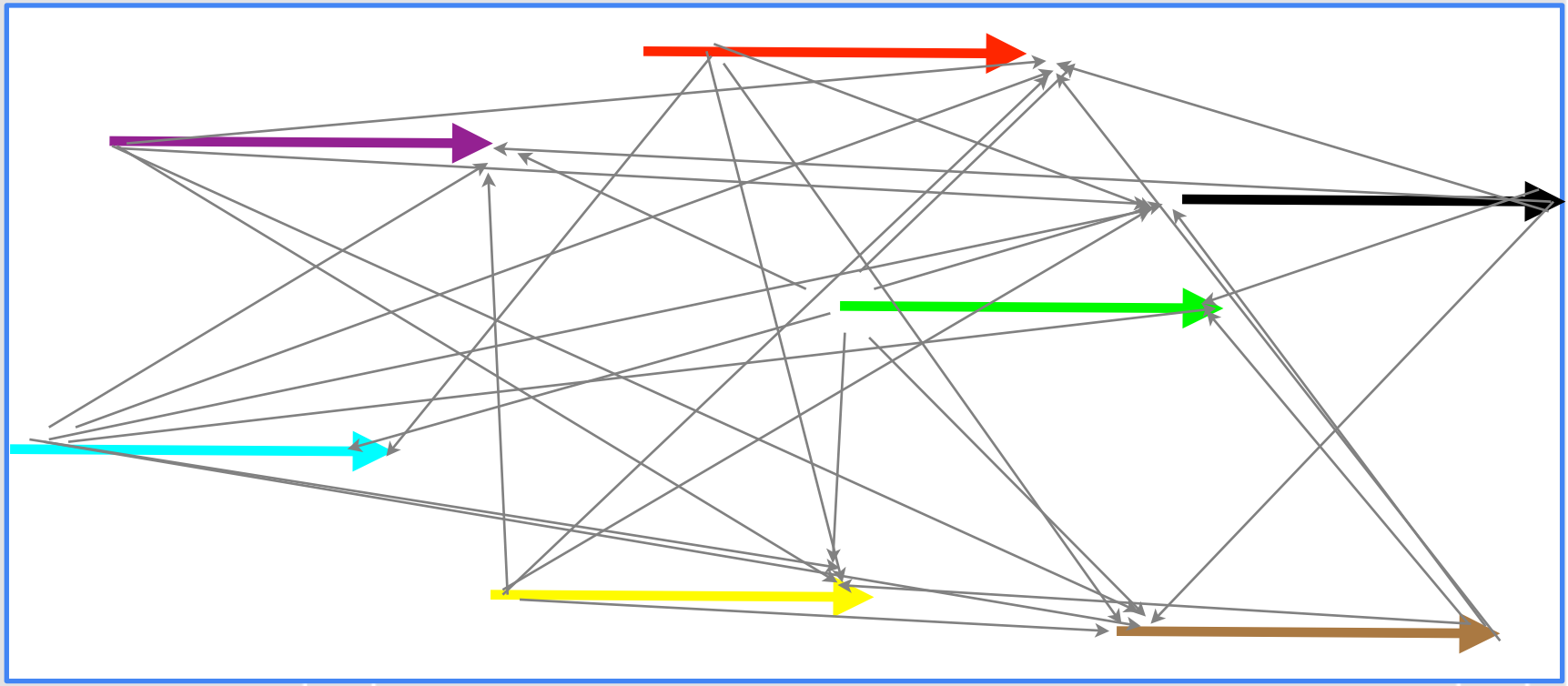
he called Seas. And God

and the gathering together of the waters he **called Seas**. And God

Assembly algorithms

(1) Overlap-consensus-layout (OCL) method

- e.g. *Flye* assembler (long reads)

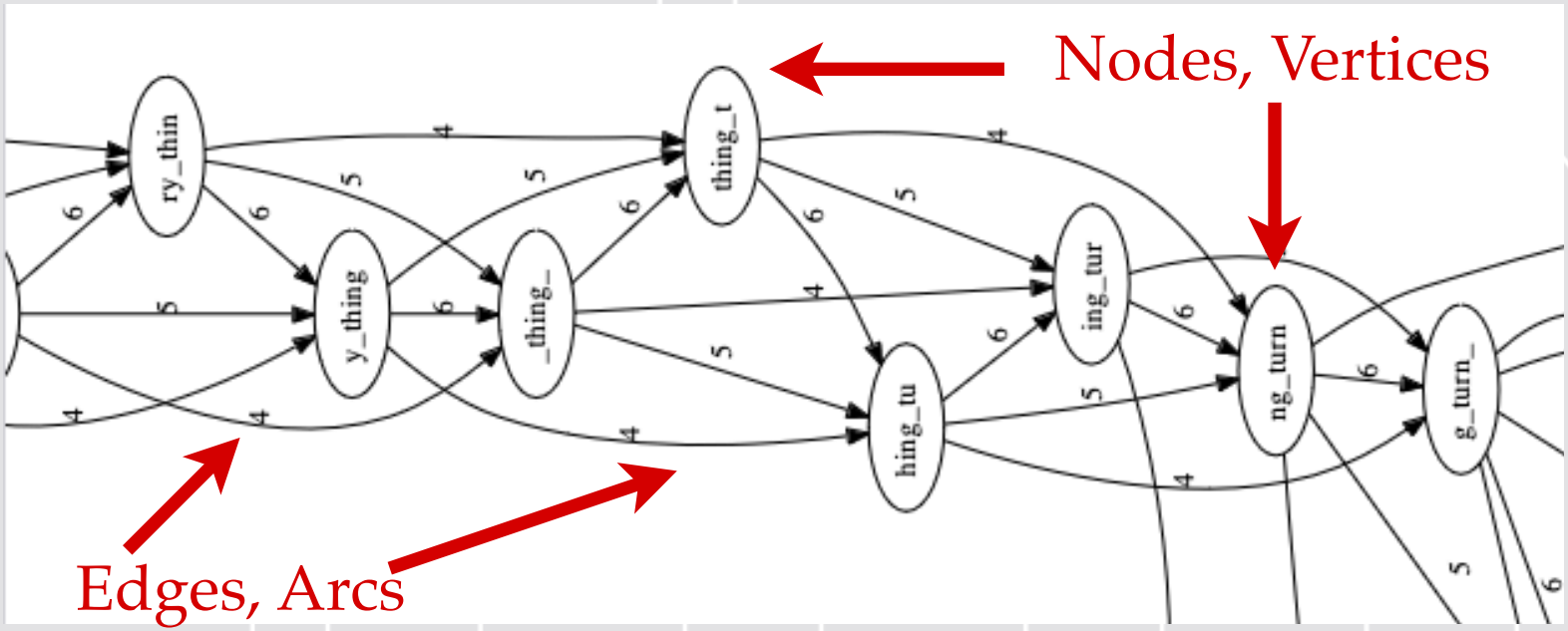


Assembly algorithms

(1) Overlap-consensus-layout (OCL) method

- **Layout**

```
to_every_thing_turn_turn_turn_there_is_a_season
```



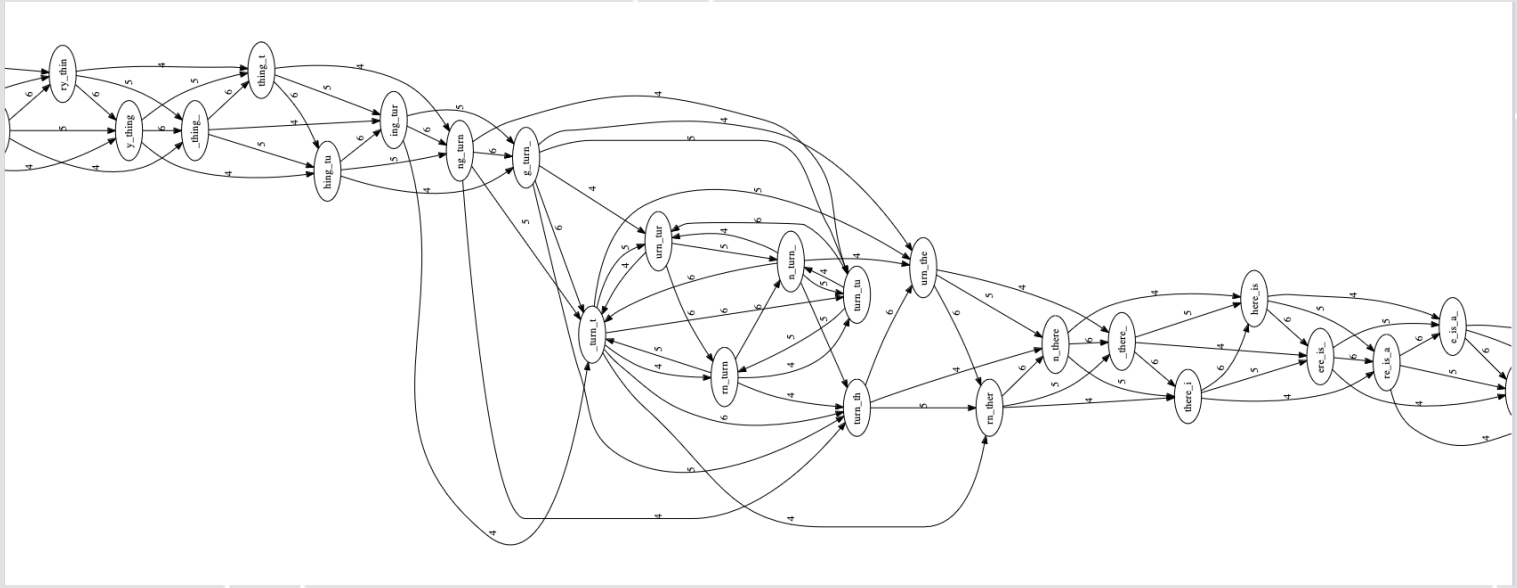
https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

Assembly algorithms

(1) Overlap-consensus-layout (OCL) method

- **Layout** `to_every_thing_turn_turn_turn_there_is_a_season`

BEFORE



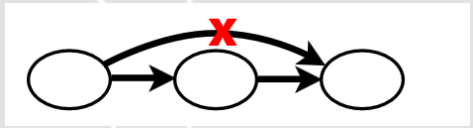
https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

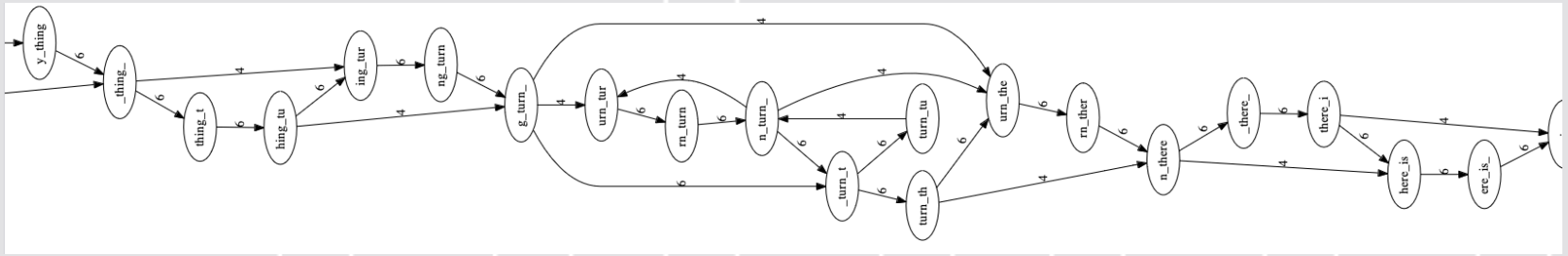
Assembly algorithms

(1) Overlap-consensus-layout (OCL) method

- **Layout** `to_every_thing_turn_turn_turn_there_is_a_season`

Remove transitively-inferrible edges, starting with edges that skip one node





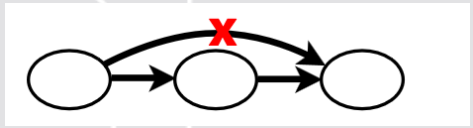
https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

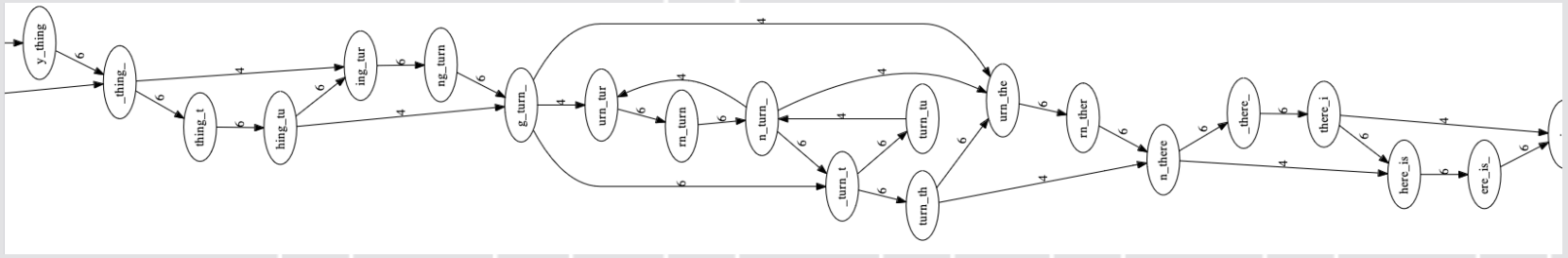
Assembly algorithms

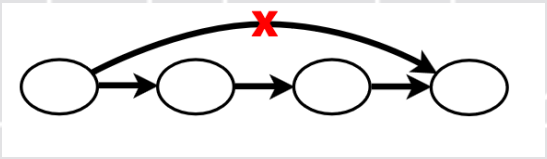
(1) Overlap-consensus-layout (OCL) method

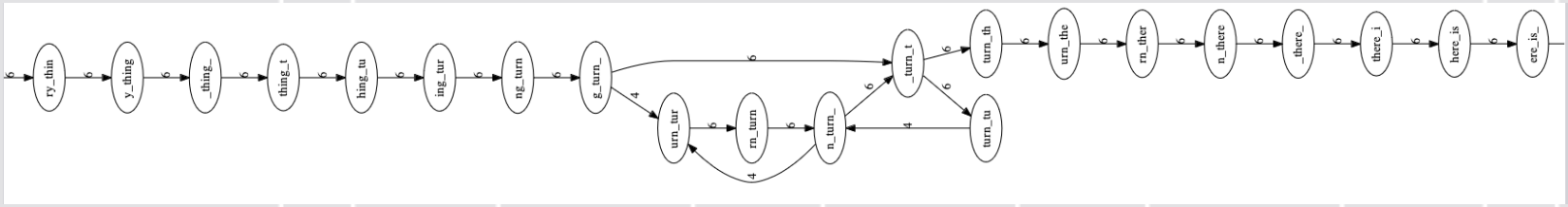
- **Layout** `to_every_thing_turn_turn_turn_there_is_a_season`

Remove transitively-inferrible edges, starting with edges that skip one node









https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

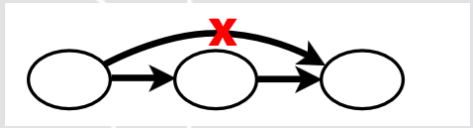
Assembly algorithms

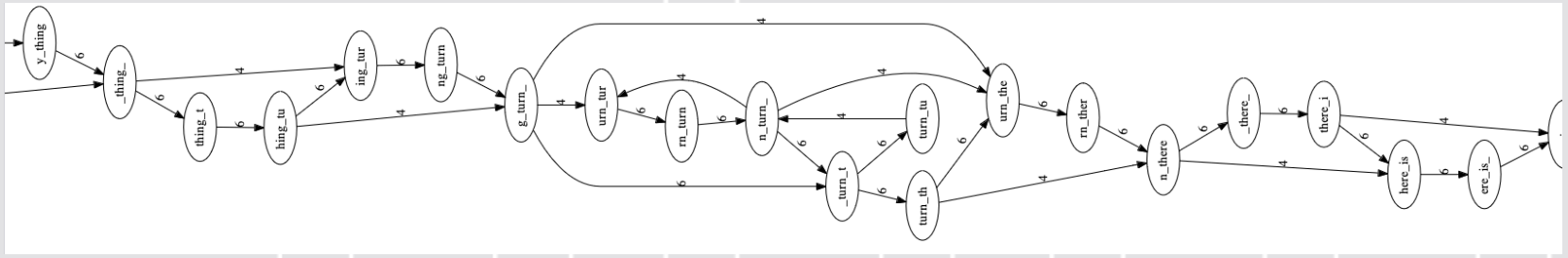
(1) Overlap-consensus-layout (OCL) method

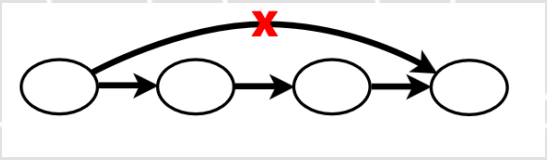
- **Layout**

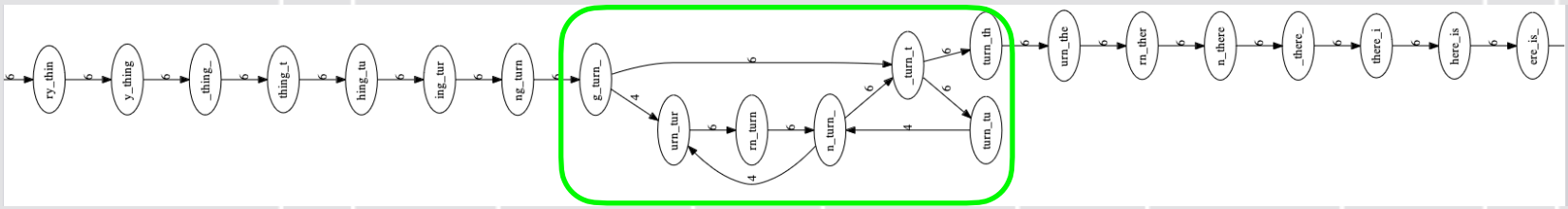
```
to_every_thing_turn_turn_turn_there_is_a_season
```

Remove transitively-inferrible edges, starting with edges that skip one node









https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_olc.pdf

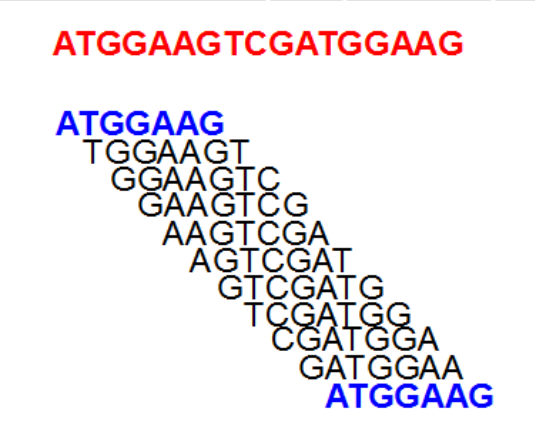
Assembly algorithms

(2) de-Bruijn-graph based method

- e.g. *SPAdes* assembler (short reads)

K-mer

The term k-mer refers to all the possible substrings of length k that are contained in a string



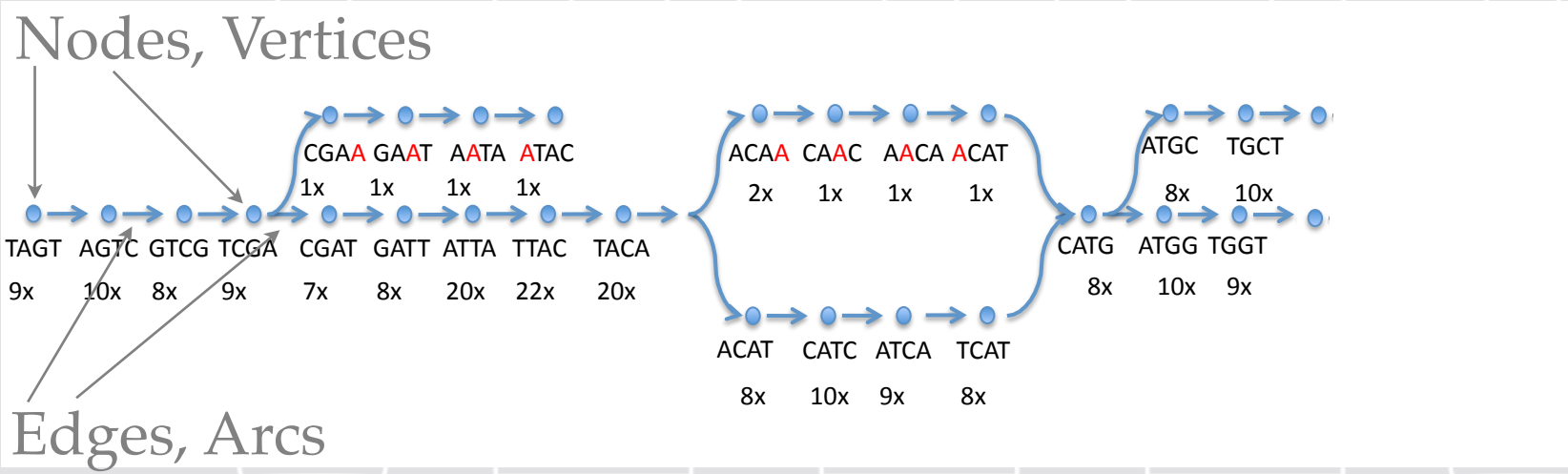
# of overlapping k-mers = *s - k + 1*

s = sequence length
k = k-mer length

Assembly algorithms

(2) de-Bruijn-graph based method

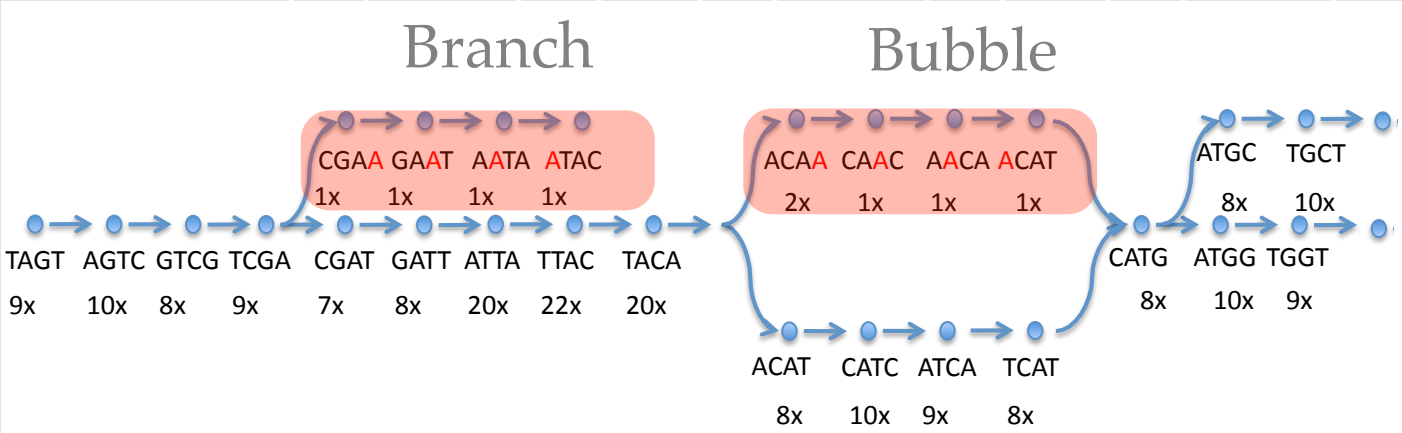
- distributed de Bruijn Graph



Assembly algorithms

(2) de-Bruijn-graph based method

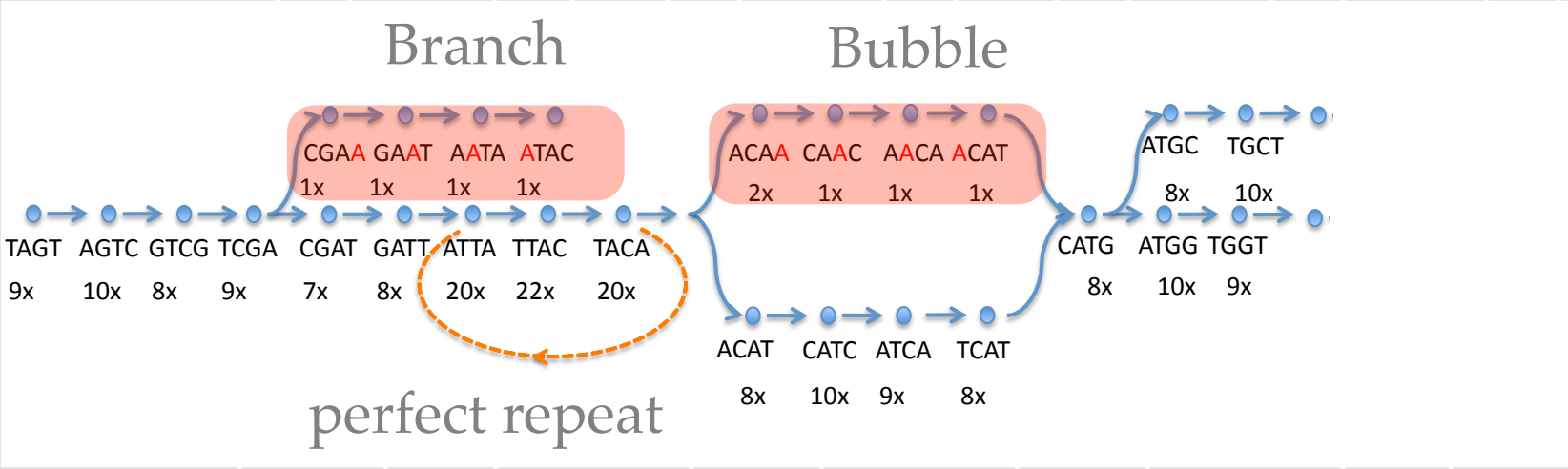
- distributed de Bruijn Graph
- removal of reading errors based on coverage threshold (popping bubbles, trimming branches)



Assembly algorithms

(2) de-Bruijn-graph based method

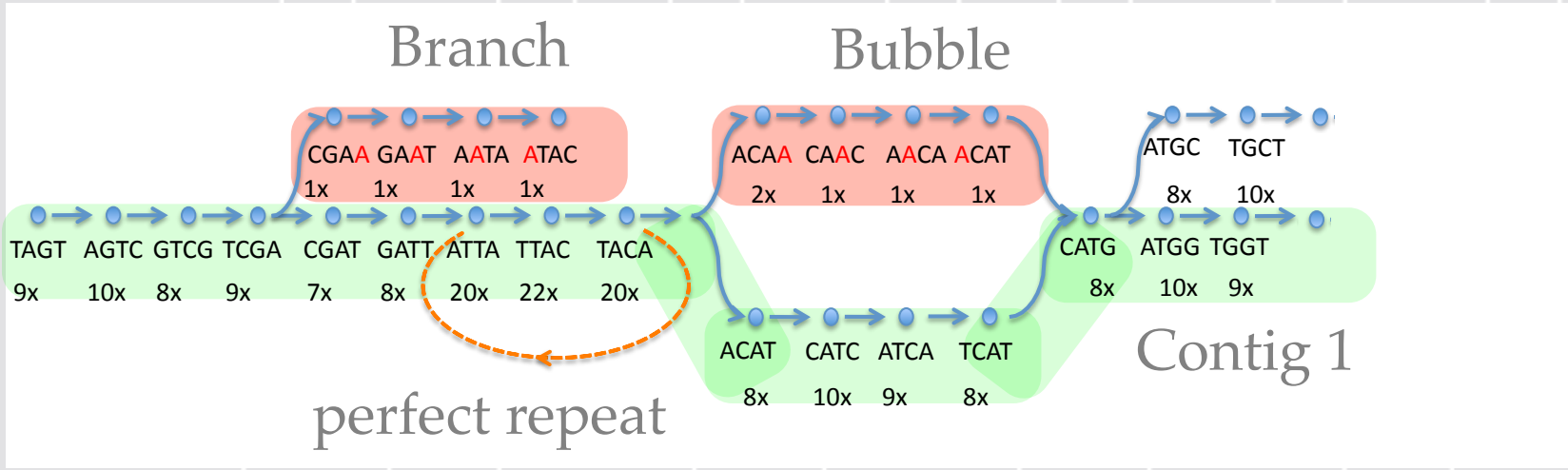
- distributed de Bruijn Graph
- removal of reading errors based on coverage threshold (popping bubbles, trimming branches)
- perfect repeats??



Assembly algorithms

(2) de-Bruijn-graph based method

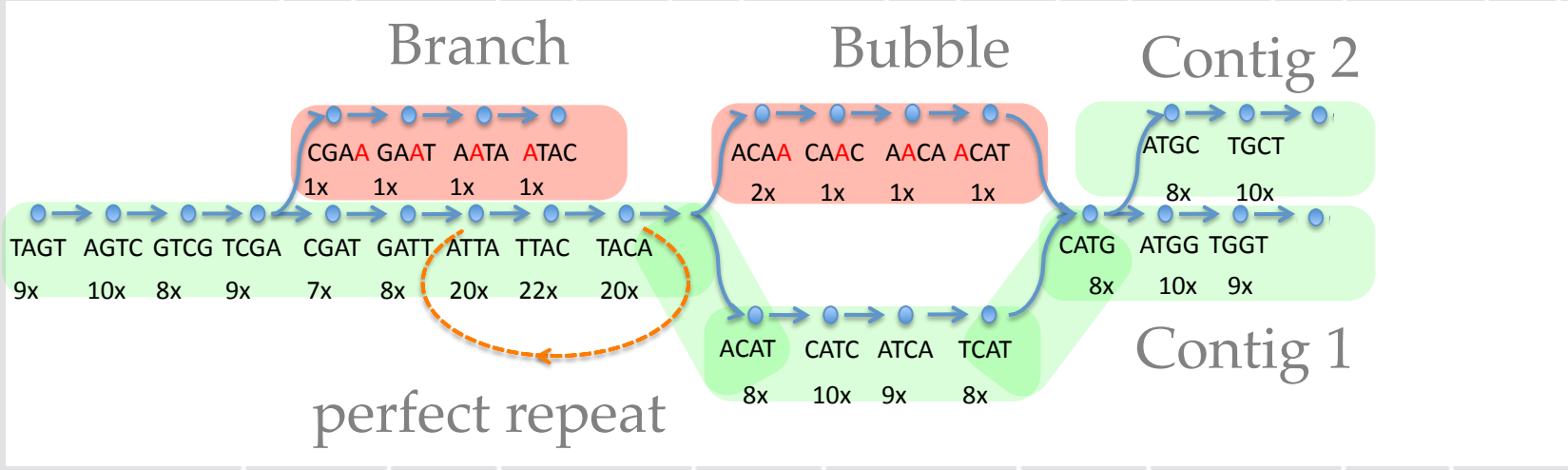
- distributed de Bruijn Graph
- removal of reading errors based on coverage threshold (popping bubbles, trimming branches)
- perfect repeats??
- merging vertex by unambiguous edges



Assembly algorithms

(2) de-Bruijn-graph based method

- distributed de Bruijn Graph
- removal of reading errors based on coverage threshold (popping bubbles, trimming branches)
- perfect repeats??
- merging vertex by unambiguous edges



Assembly algorithms

(2) de-Bruijn-graph based method

- e.g. *SPAdes* assembler (short reads)

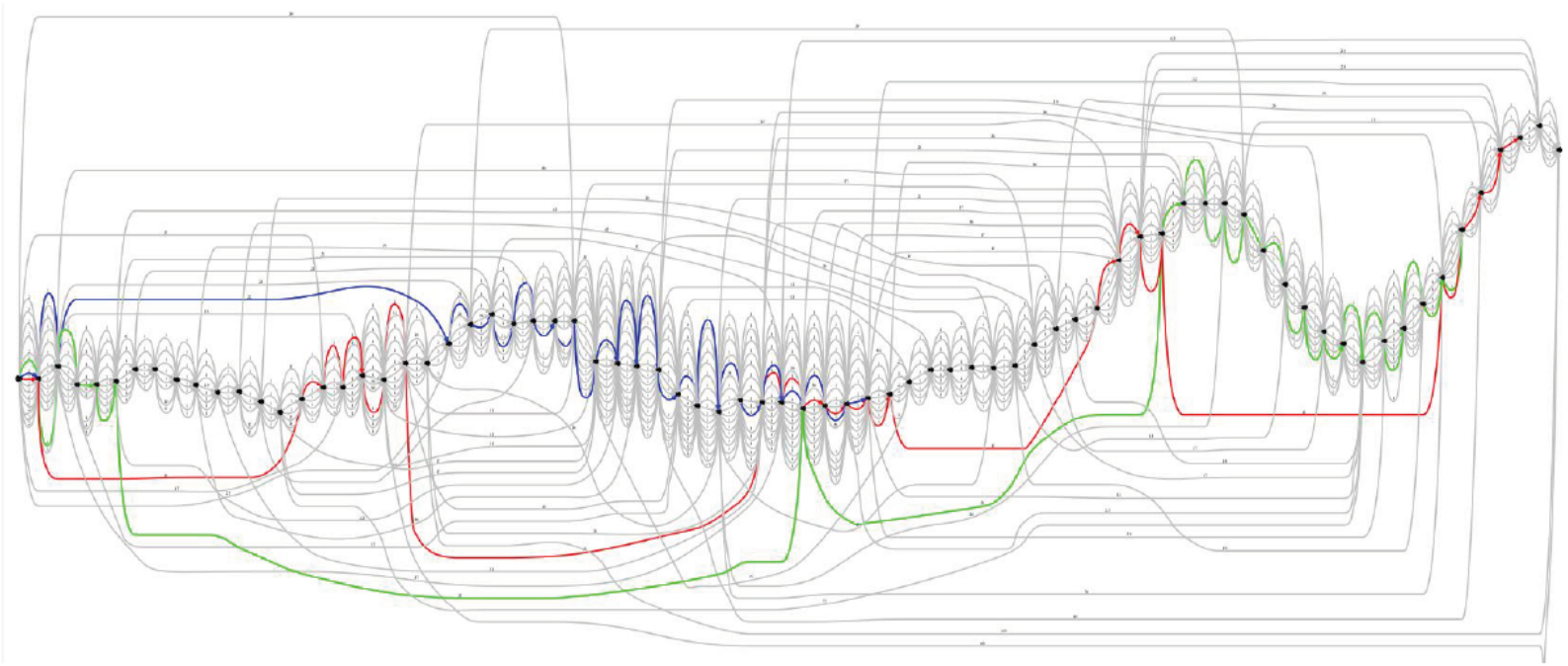
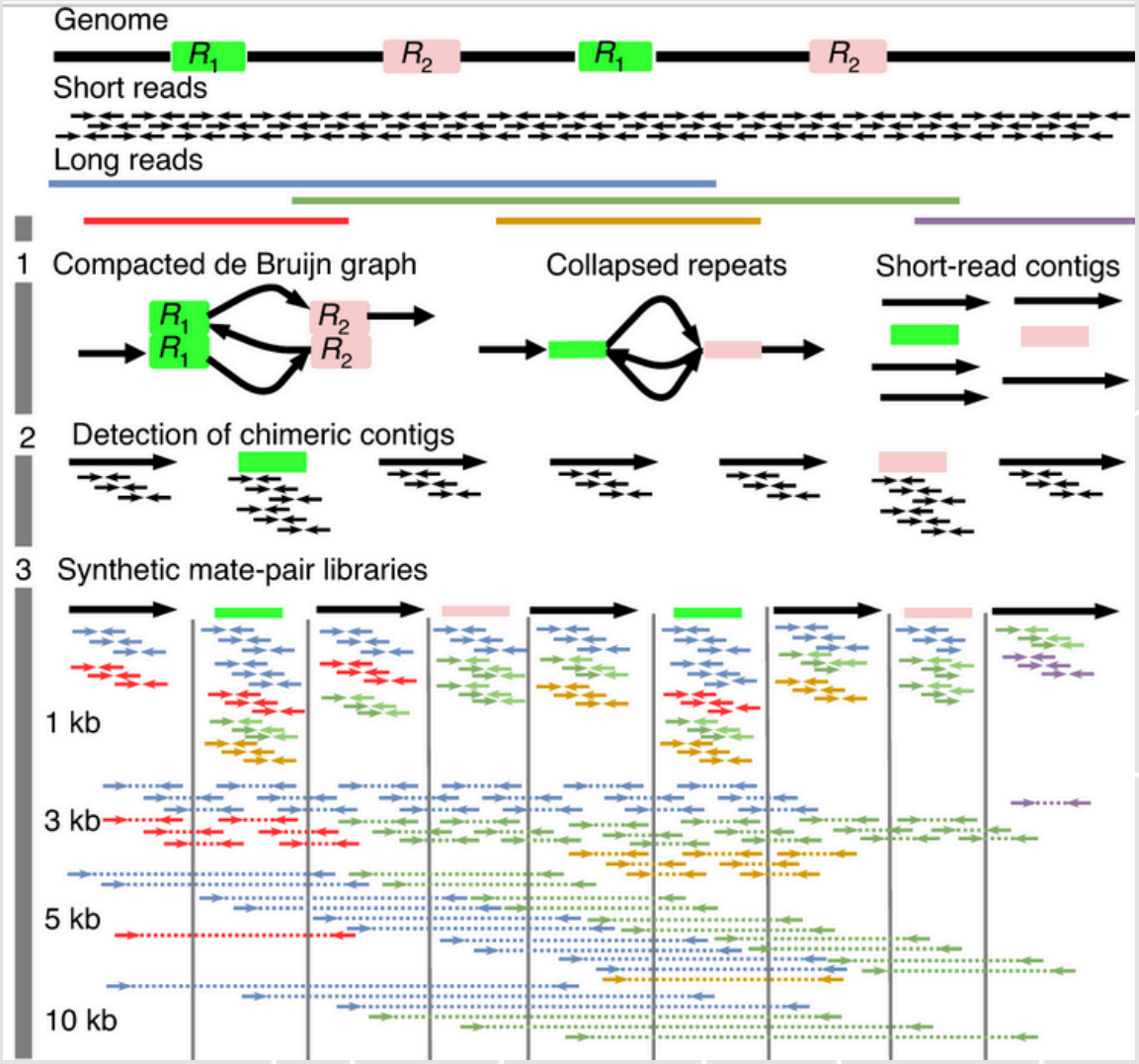


Fig. S1. A small subgraph of the A-Bruijn graph constructed from 76 (15,8)-mers appearing in segments of 55 reads covering a short 100-nucleotide region (starting at position 2,100,000 in *E. coli* genome). Three out of 55 read-paths are highlighted in blue, red, and green.

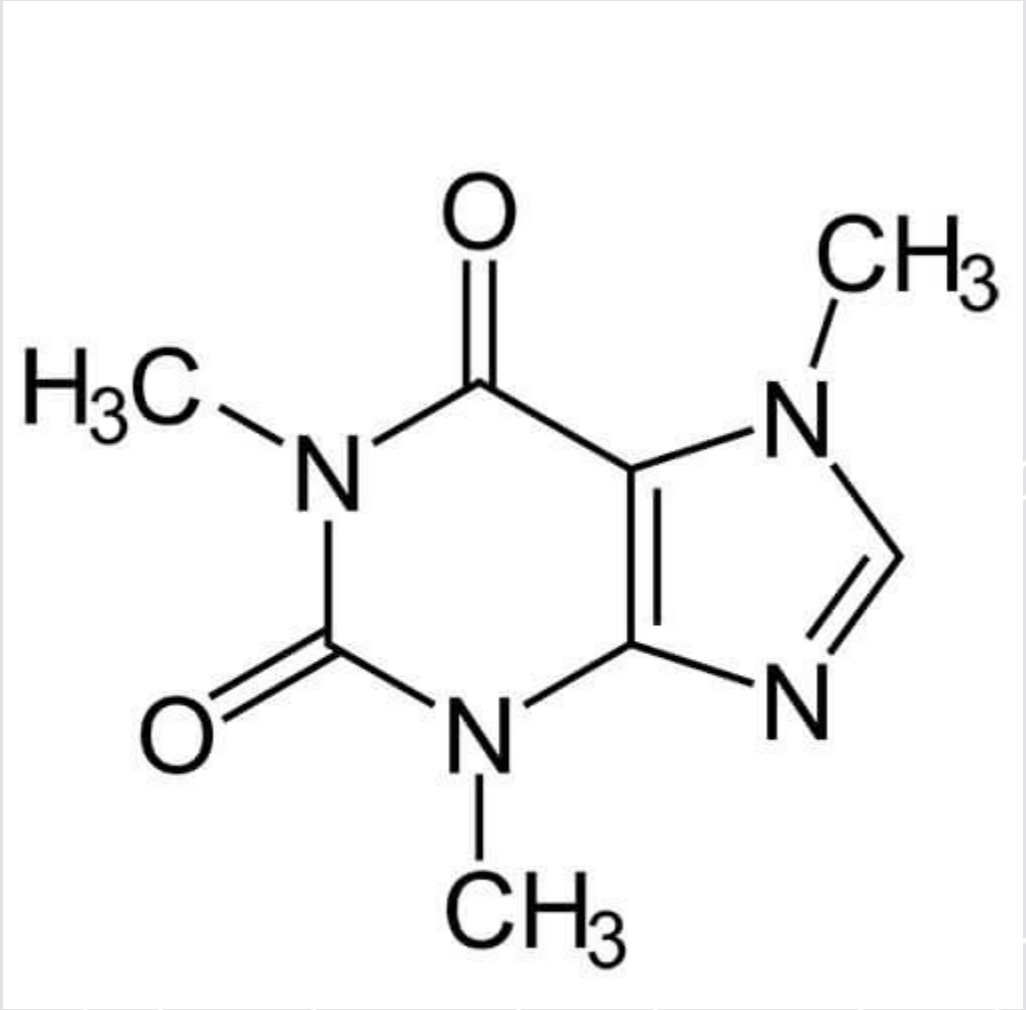
Hybrid Assembly

(3) Hybrid assembly - WENGAN

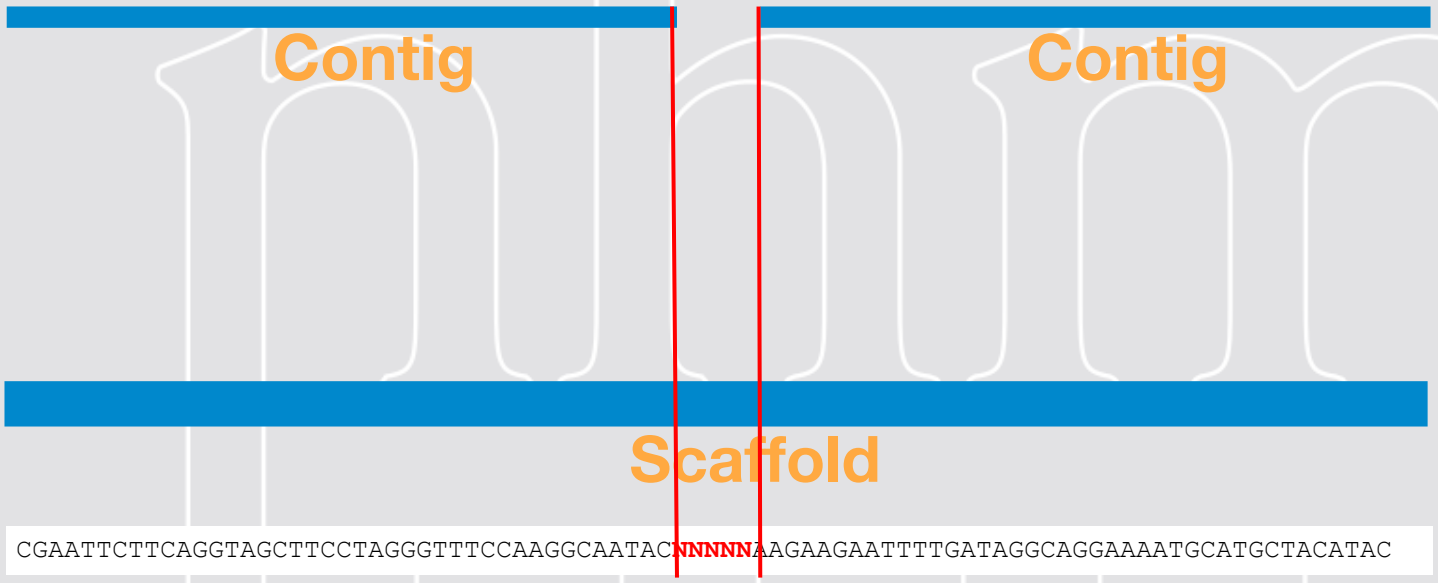


https://www.nature.com/articles/s41587-020-00747-w

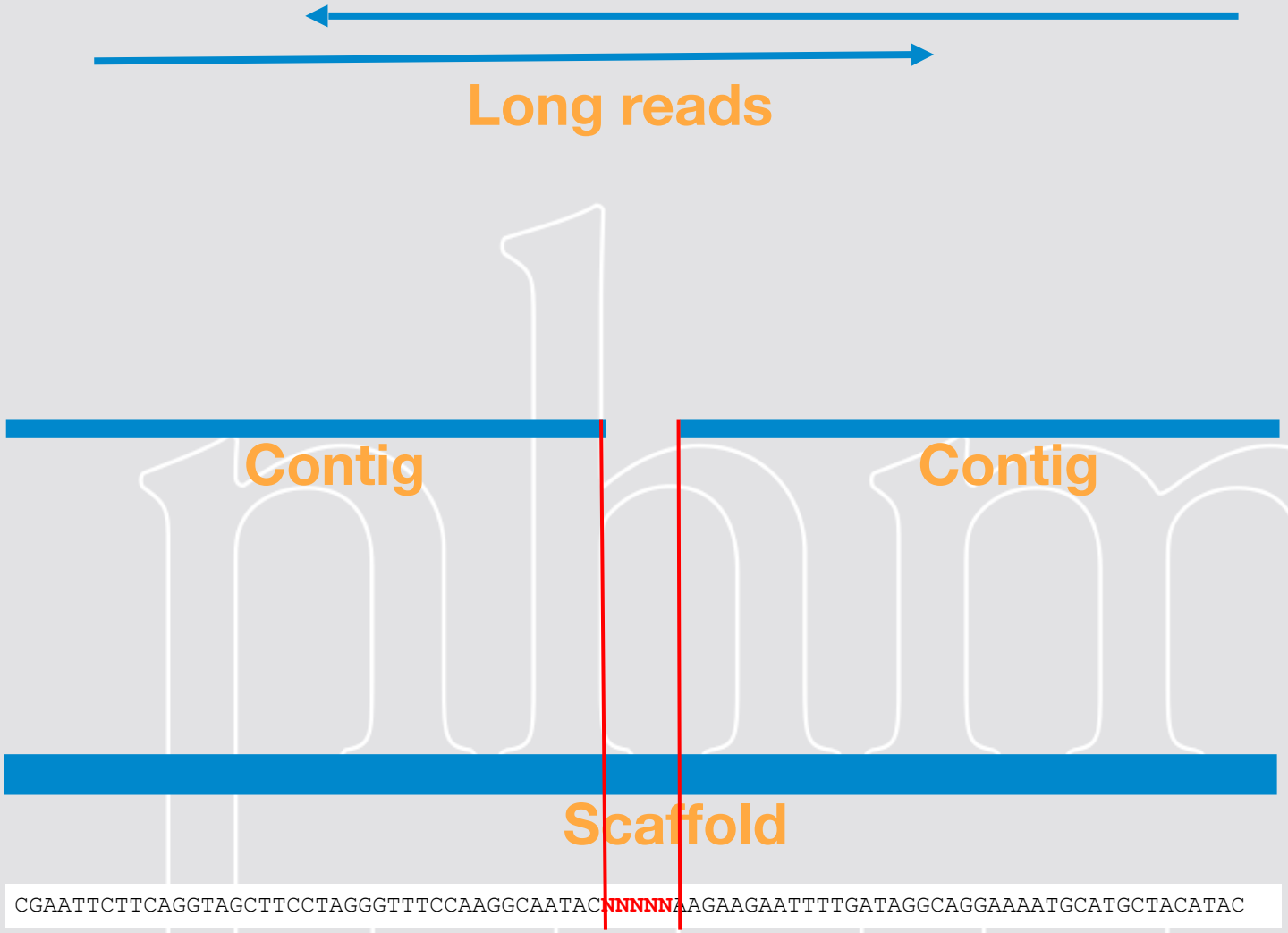
10’ Break!



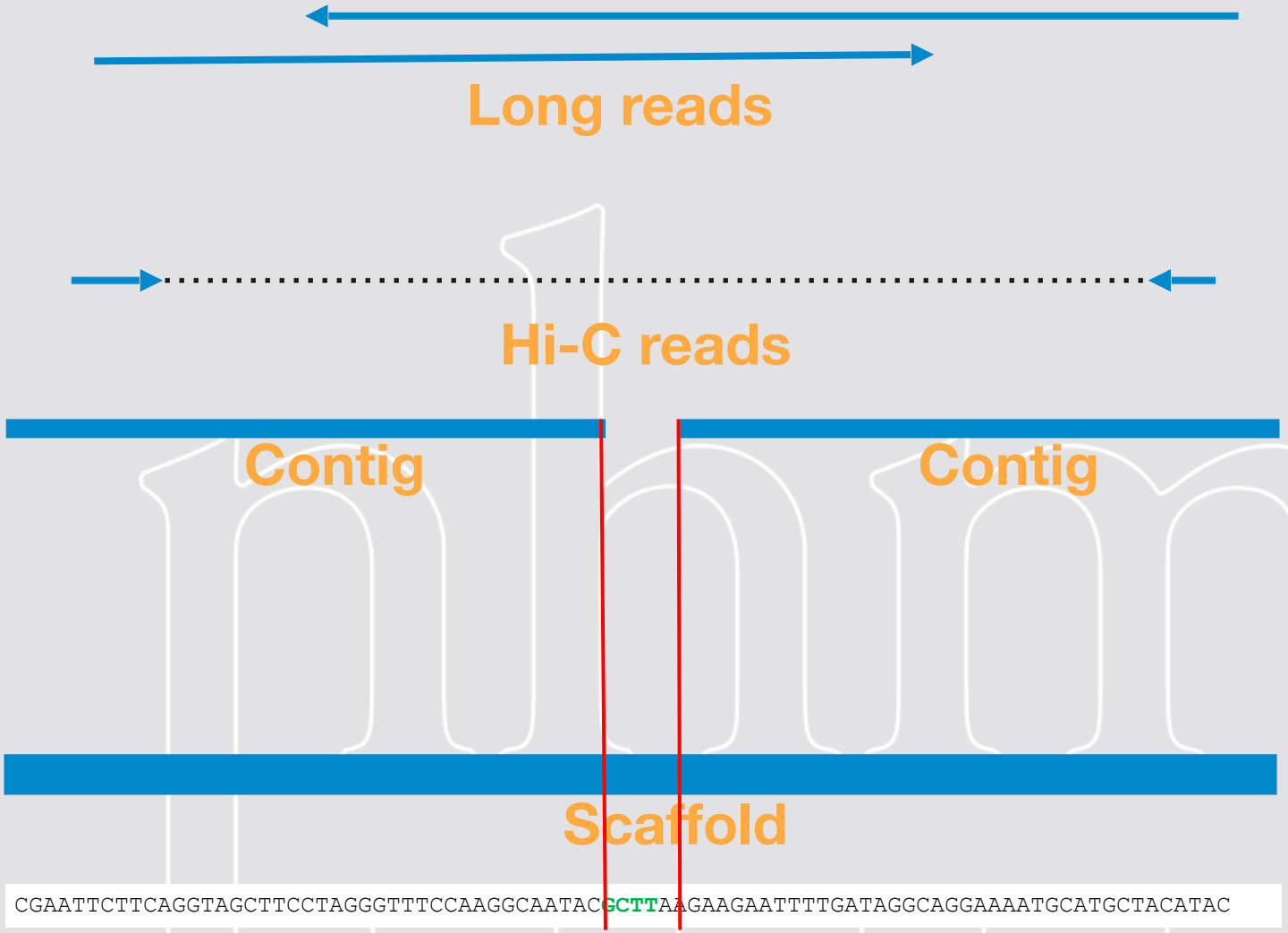
Scaffolding



Scaffolding

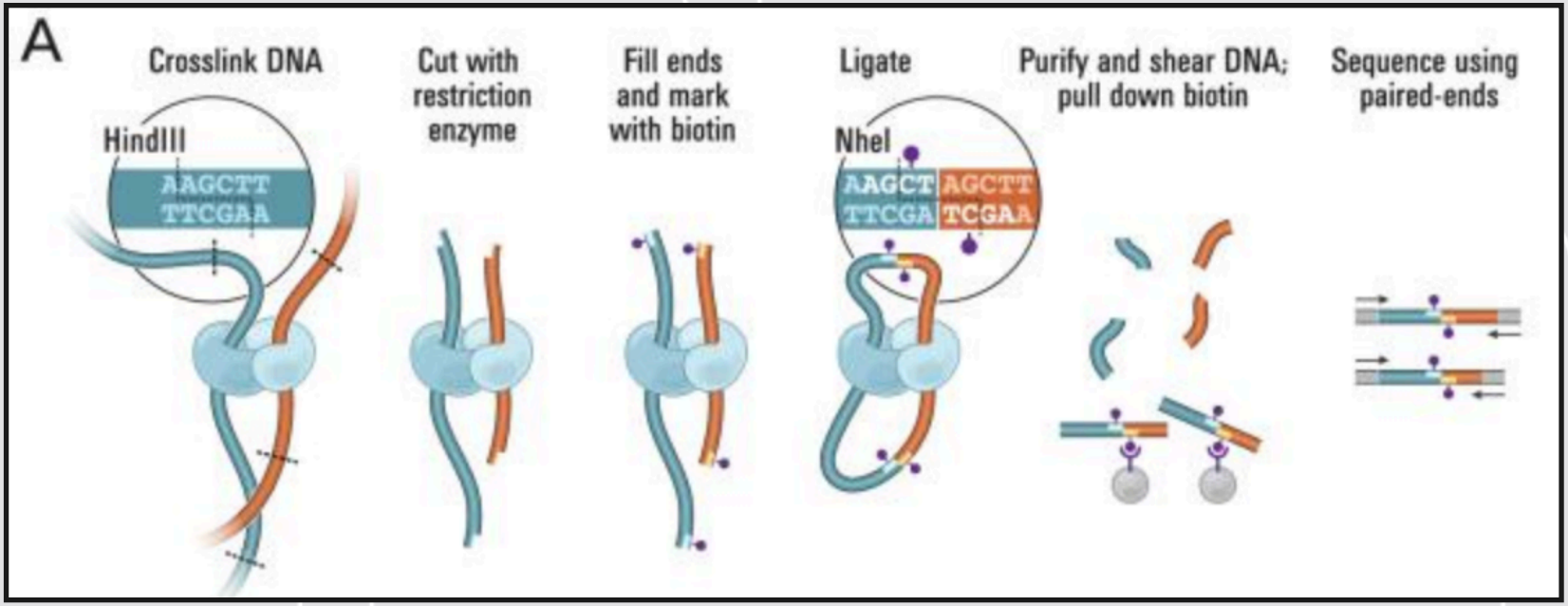


Scaffolding



Hi-C sequencing

Hi-C (Genomic Chromosome-Conformation-Capture)



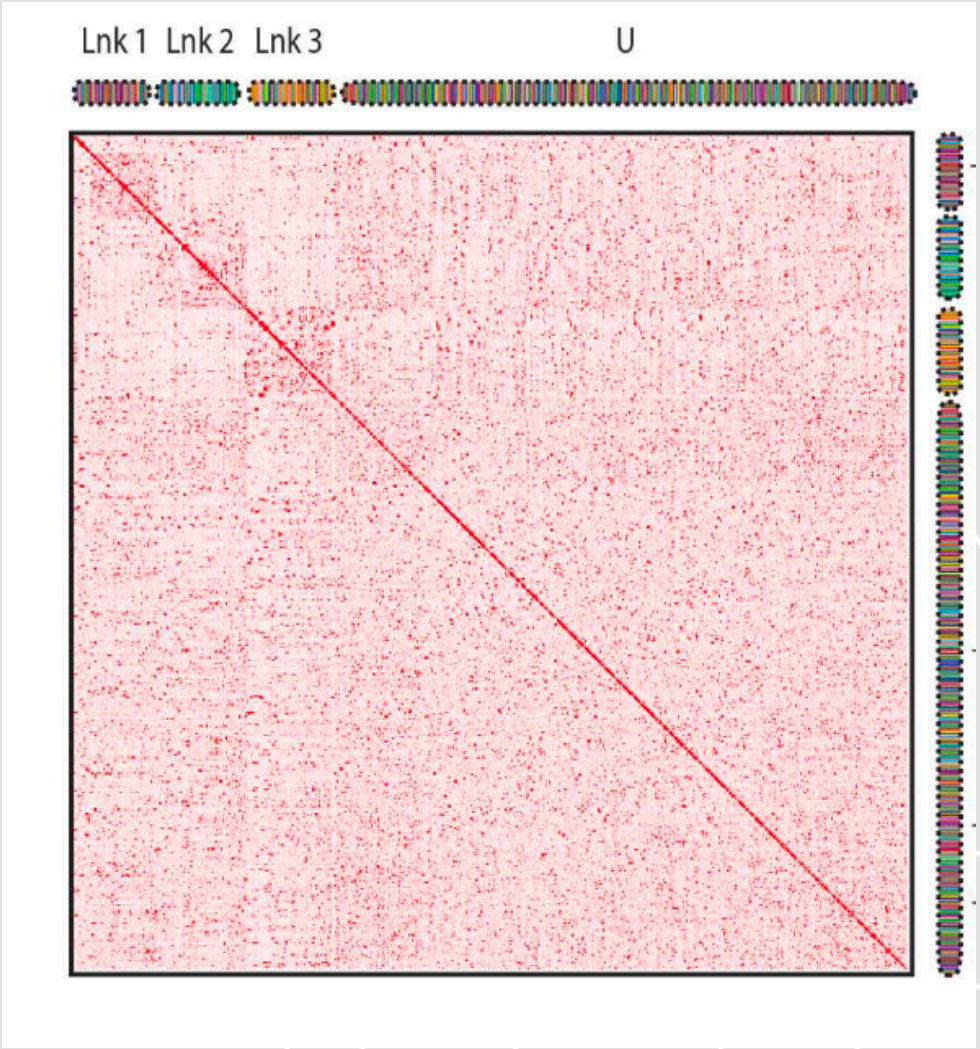
Lieberman-Aiden, E., van Berkum, N.L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., *et al*. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**: 289–293.

Hi-C sequencing

- **Order** of genome synteny **correct**
- Structure in plot according to **physical distance**
- Structure in plot according to **chromatin structure**

Lieberman-Aiden, E., van Berkum, N.L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., *et al*. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**: 289–293.

Hi-C sequencing



- **Order** of contigs **wrong**
- **Order** of genome synteny **wrong**
- No structure in the contact map

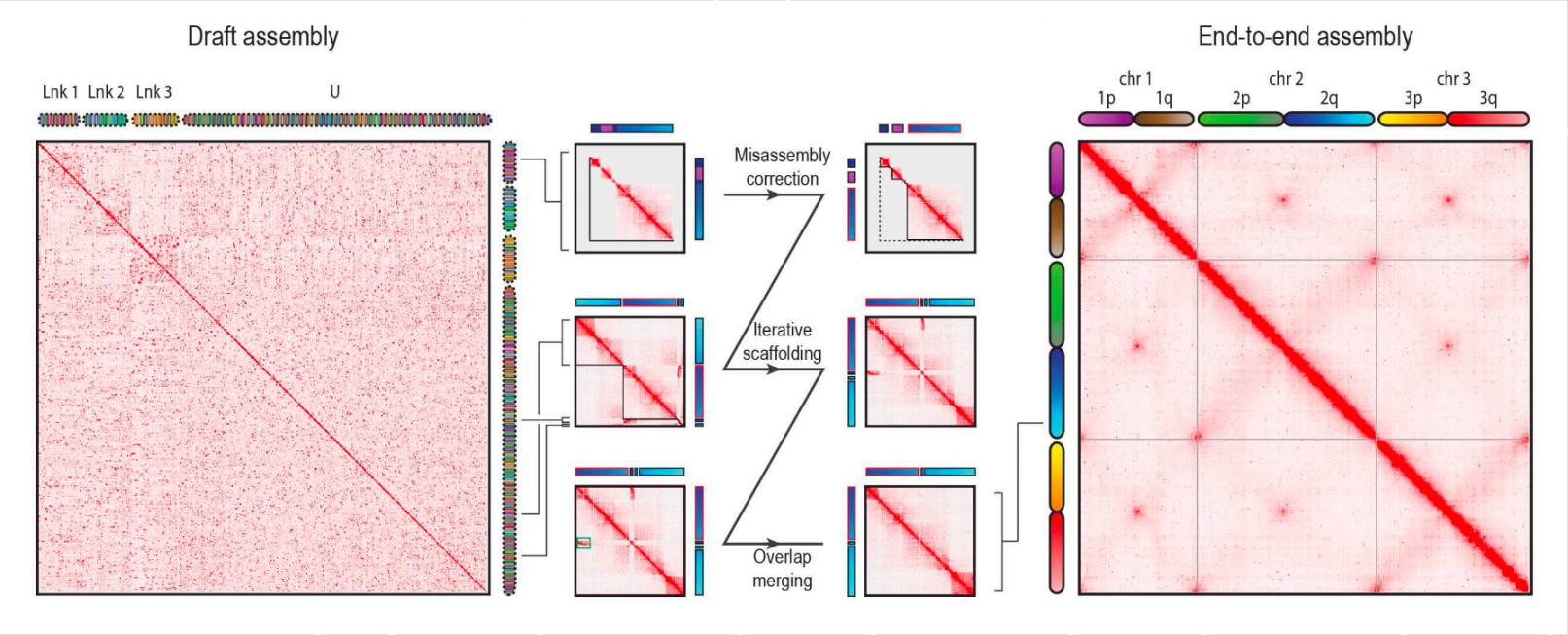
Lieberman-Aiden, E., van Berkum, N.L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., *et al*. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**: 289–293.

Hi-C sequencing

Hi-C reads

Contig

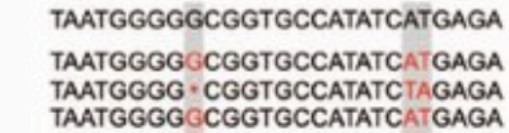
Contig



Lieberman-Aiden, E., van Berkum, N.L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., *et al*. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**: 289–293.

Polishing

PROCESS	RESULT	
Pilon protocol	Assembly improvement (Fasta)	Variation detection (VCF)
Evaluate alignment pileups	Identify and fix base errors	Identify SNPs and small indels
Scan read coverage and alignment discrepancies	Identify potential local misassemblies	Identify larger insertions and deletions
Reassemble across gaps and discrepant regions	Attempt to fill gaps and fix local misassemblies	Attempt to build out the full sequence of larger insertions



Lieberman-Aiden, E., van Berkum, N.L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., *et al*. 2009. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**: 289–293.

K-mers revisited

A relationship between k-mers and genome-size

GATCCTACTGATGC (L = 14) on decomposition of k-mers of length k = 8,

Total number of k-mer's generated will be

n = (L - k) + 1
= (14 - 8) + 1
= 7

GATCCTAC, ATCCTACT, TCCTACTG, CCTACTGA,
CTACTGAT, TACTGATG, ACTGATGC

https://bioinformatics.uconn.edu/genome-size-estimation-tutorial/

K-mers revisited

A relationship between k-mers and genome-size

GATCCTACTGATGC (L = 14) on decomposition of k-mers of length k = 8,

Total number of k-mer's generated will be

n = (L - k) + 1
= (14 - 8) + 1
= 7

GATCCTAC, ATCCTACT, TCCTACTG, CCTACTGA,
CTACTGAT, TACTGATG, ACTGATGC

k=18		
Genome Sizes	Total K-mers of k=18	% error in genome estimation
L	N=(L-K)+1	
100	83	17
1000	983	1.7
10000	9983	0.17
100000	99983	0.017
1000000	999983	0.0017

https://bioinformatics.uconn.edu/genome-size-estimation-tutorial/

K-mers revisited

A relationship between k-mers and genome-size

For sequences > 1mb:

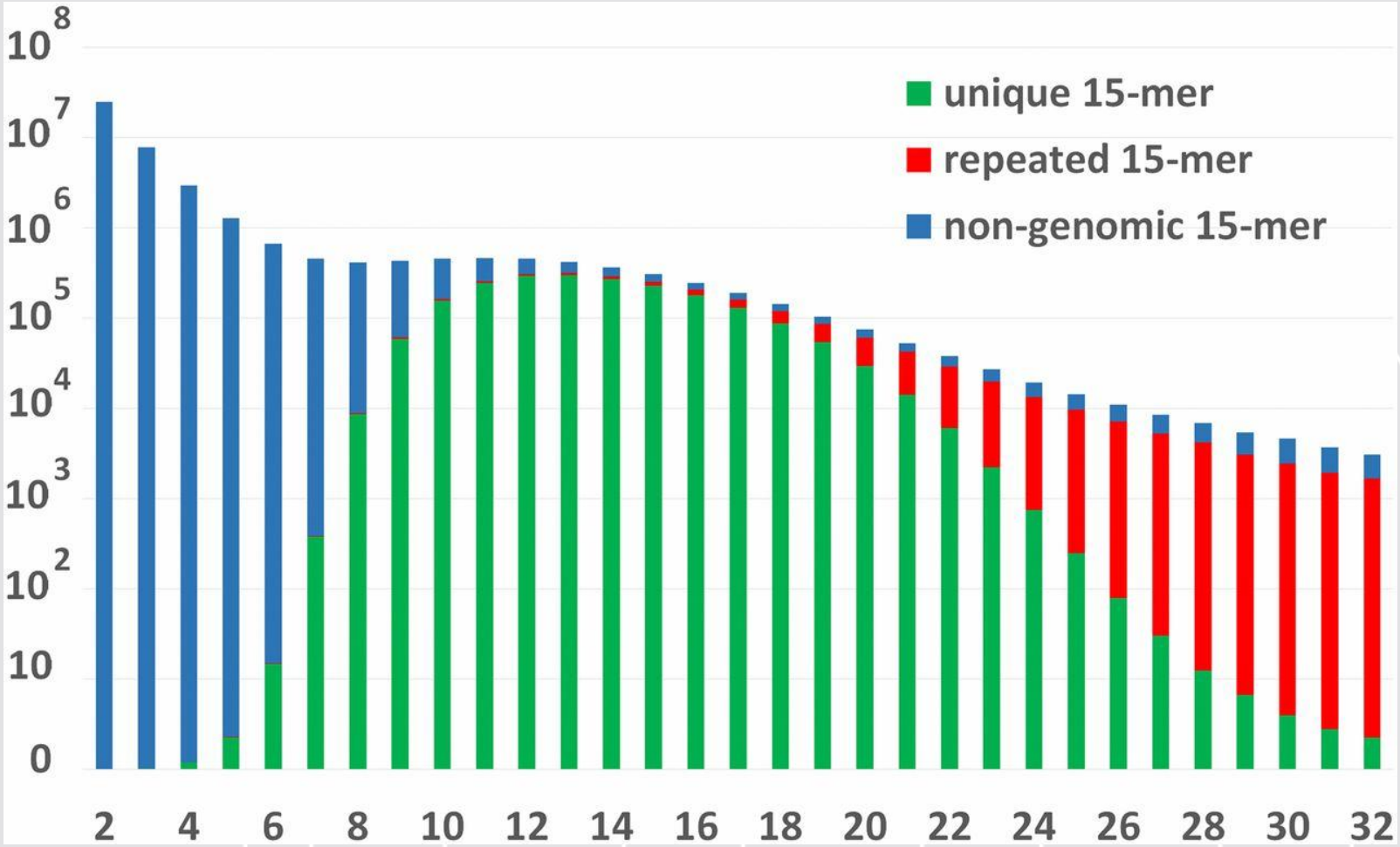
unique k-mers ~ sequence length

Genomesize ~ #unique k-mers/coverage

BUT, Read Error, Repeats & Heterozygosity!!

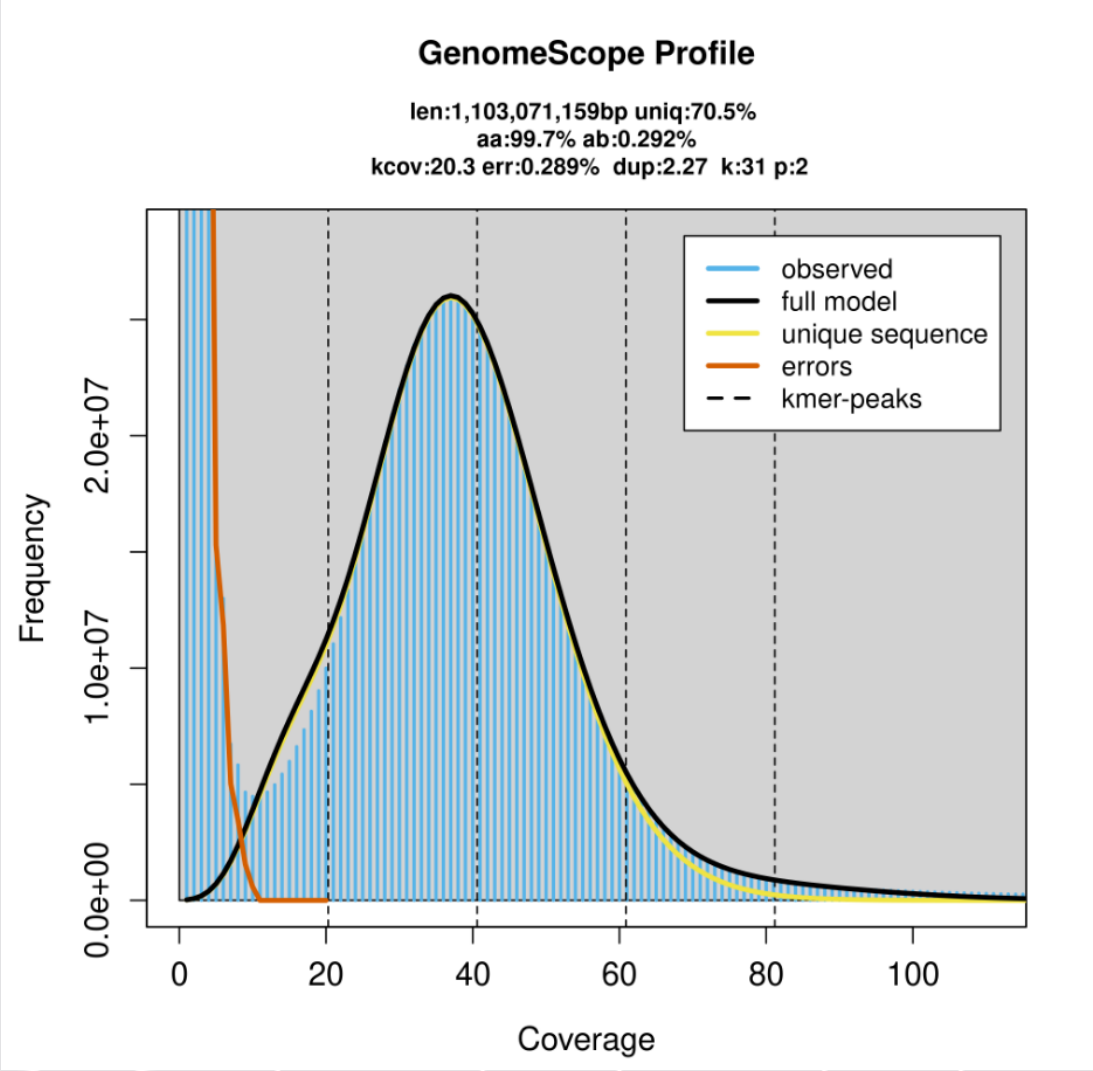
K-mers revisited

K-mer, Read Error, Repeats & Heterozygosity



K-mers revisited

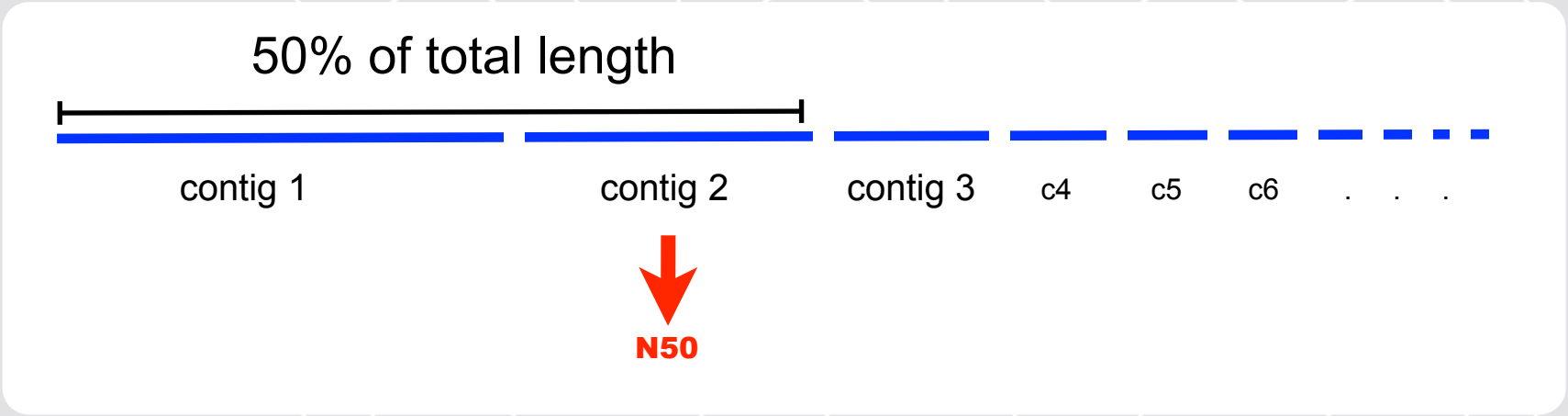
Model-based estimate of genome-size with *Genomescope*



Assembly QC metrics

Standard Metrics (*QUAST*):

- Longest Contig; longest Scaffold;
- **N50**: 50% of total assembly length is comprised by contigs, larger or equal to the N50 contig

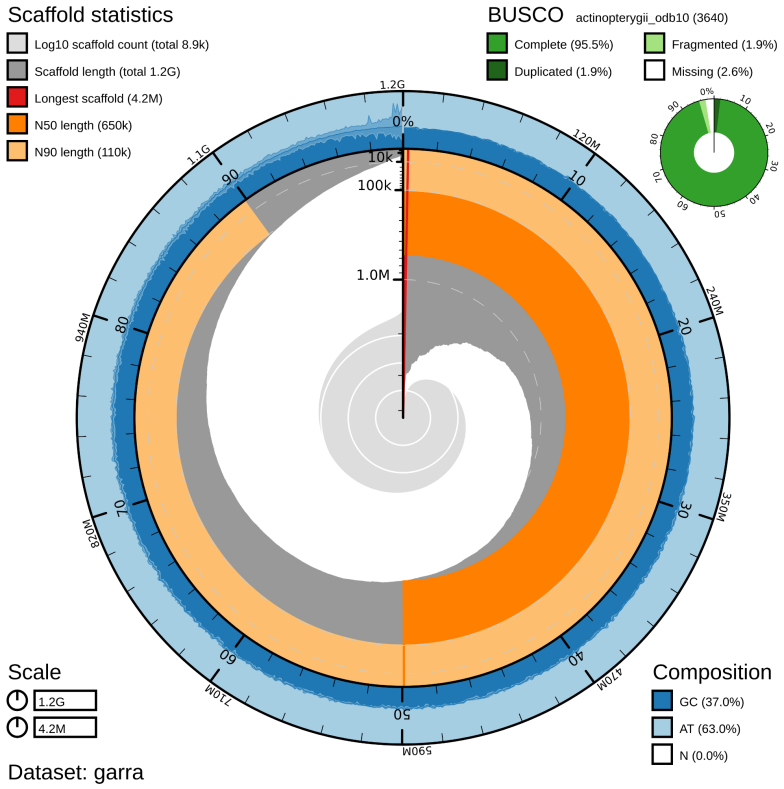


Assembly QC metrics

NHMW Workshop IV

Standard Metrics (*QUAST*):

- Longest Contig; longest Scaffold; N50
- **Snailplot**

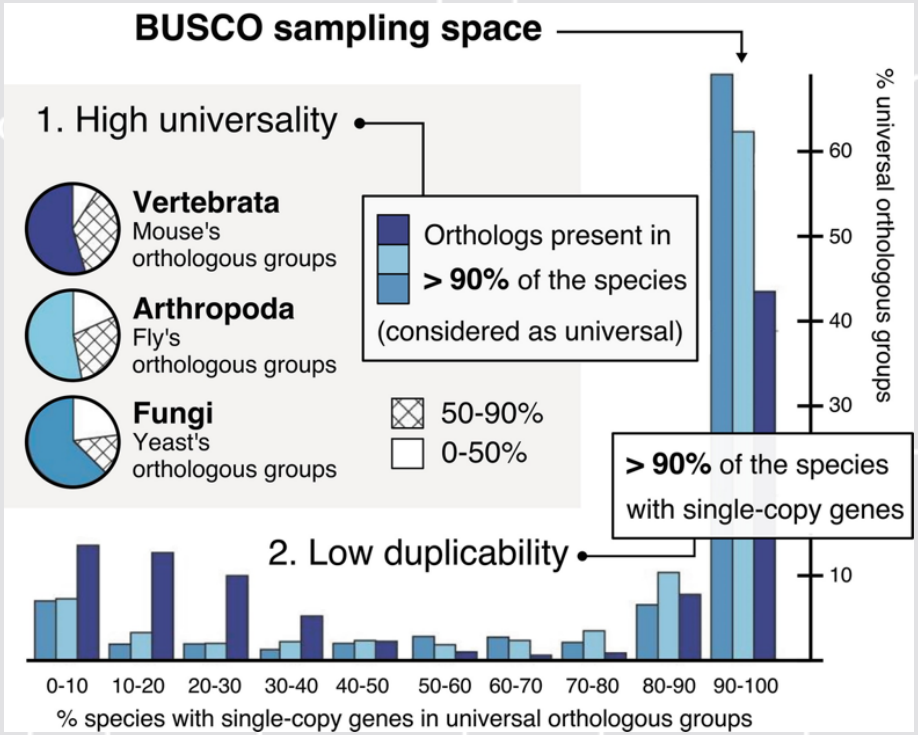


- light/dark blue: AT/GC content [%]
- light orange: N90 [y-axis: length]
- dark orange: N50 [y-axis: length]
- dark grey: [y-axis: scaffold lengths]
- light grey: [count of scaffolds]

Assembly QC metrics

Benchmarking Universal Single-Copy Orthologue (BUSCO)

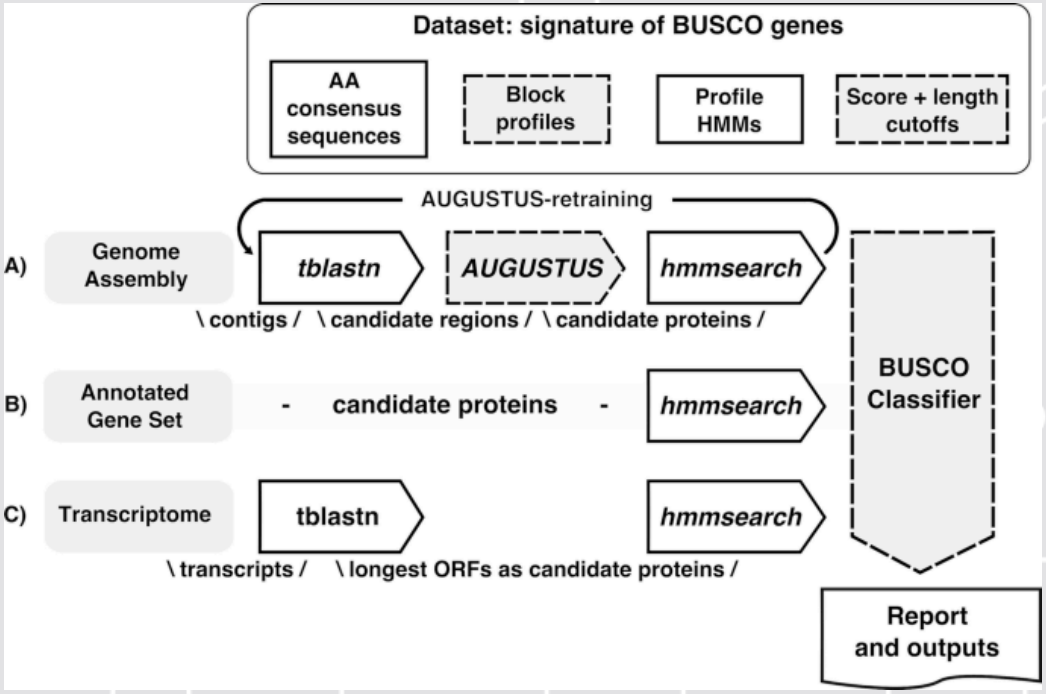
- Estimate assembly quality based on presence and completeness of conserved single copy genes.



Assembly QC metrics

Benchmarking Universal Single-Copy Orthologue (BUSCO)

- Estimate assembly quality based on presence and completeness of conserved single copy genes.

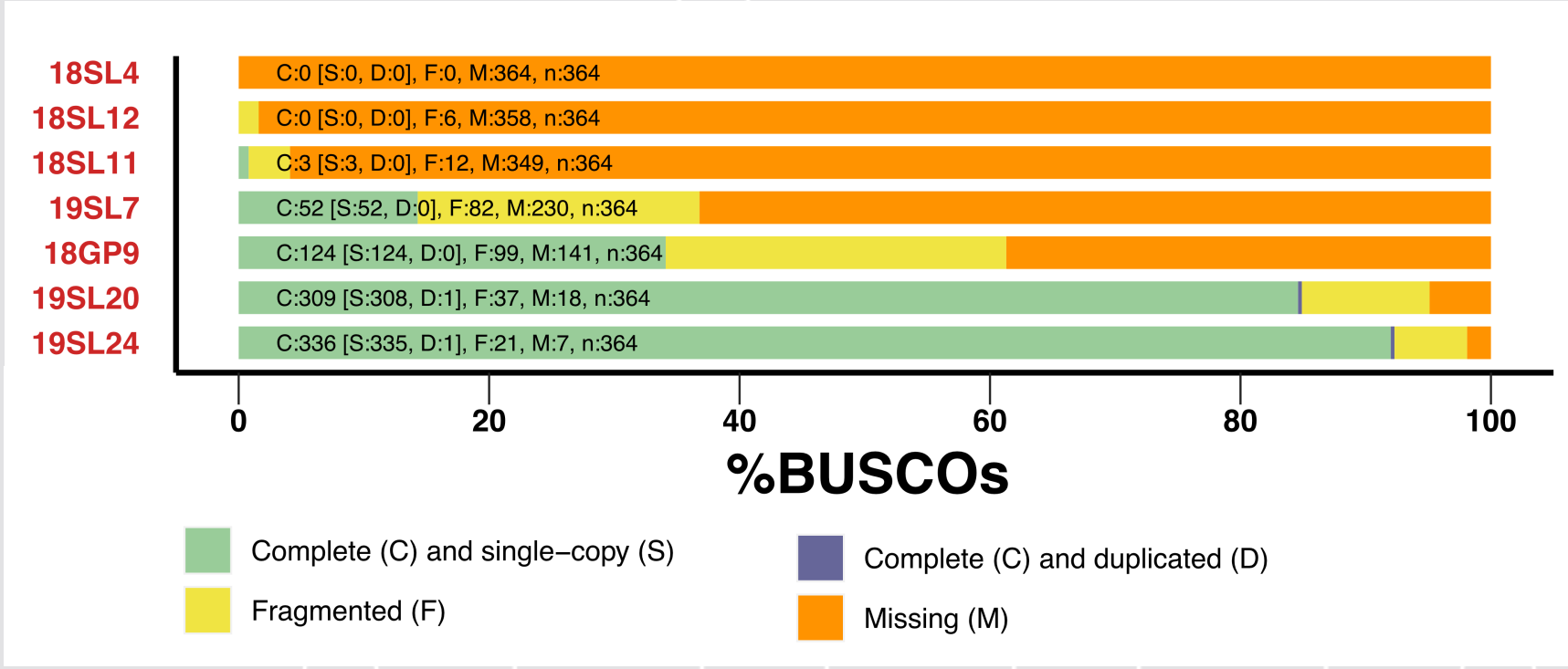


Seppey, M., Manni, M. & Zdobnov, E.M. 2019. BUSCO: Assessing Genome Assembly and Annotation Completeness. *Gene Prediction* 227–245. Humana, New York, NY.

Assembly QC metrics

Benchmarking Universal Single-Copy Orthologue (BUSCO)

- Estimate assembly quality based on presence and completeness of conserved single copy genes.



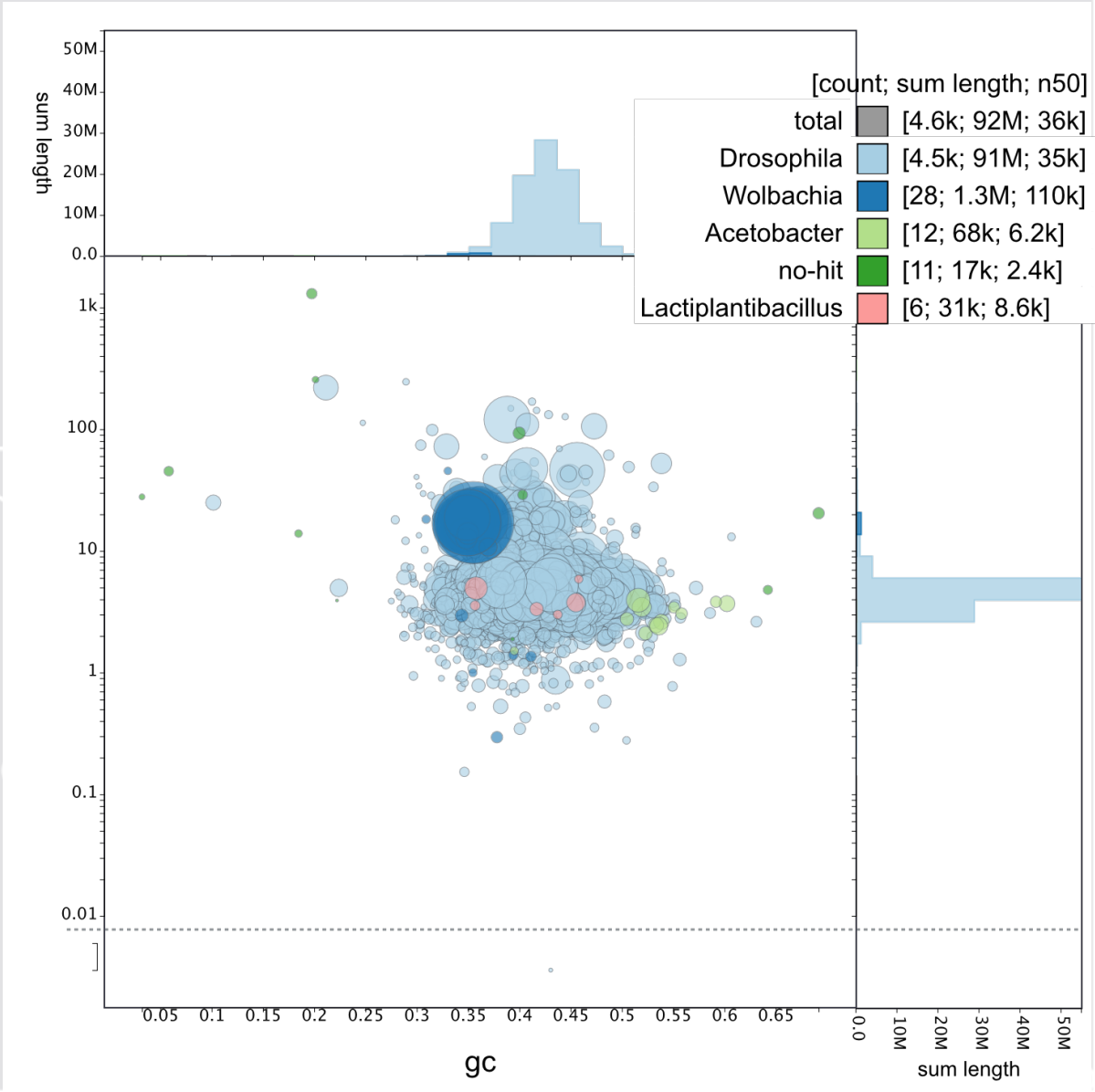
Seppey, M., Manni, M. & Zdobnov, E.M. 2019. BUSCO: Assessing Genome Assembly and Annotation Completeness. *Gene Prediction* 227–245. Humana, New York, NY.

Assembly QC metrics

BLOBtools

Combines various assembly-QC analyses

- **BLAST** -> similarity of contigs to taxa
- **MAPPING** -> coverage variation
- **BUSCO**



Hands on…

https://github.com/nhmvienna/Workshop_IV_DeNovoAssembly

README.md

Workshop IV - De-novo assembly

- The slides to this Workshop can be found here
- The recordings of the Workshop can be found on the NHM intranet under `I:\Public\mkapun\FrontiersInMolecularSystematics\Workshop_IV_DeNovoAssembly`
- check out the previous workshop for instructions on how to work on the Phyloserver
- In your home directory on the server, clone this repository by typing the following command in your commandline

```
git clone https://github.com/nhmvienna/Workshop_IV_DeNovoAssembly
```

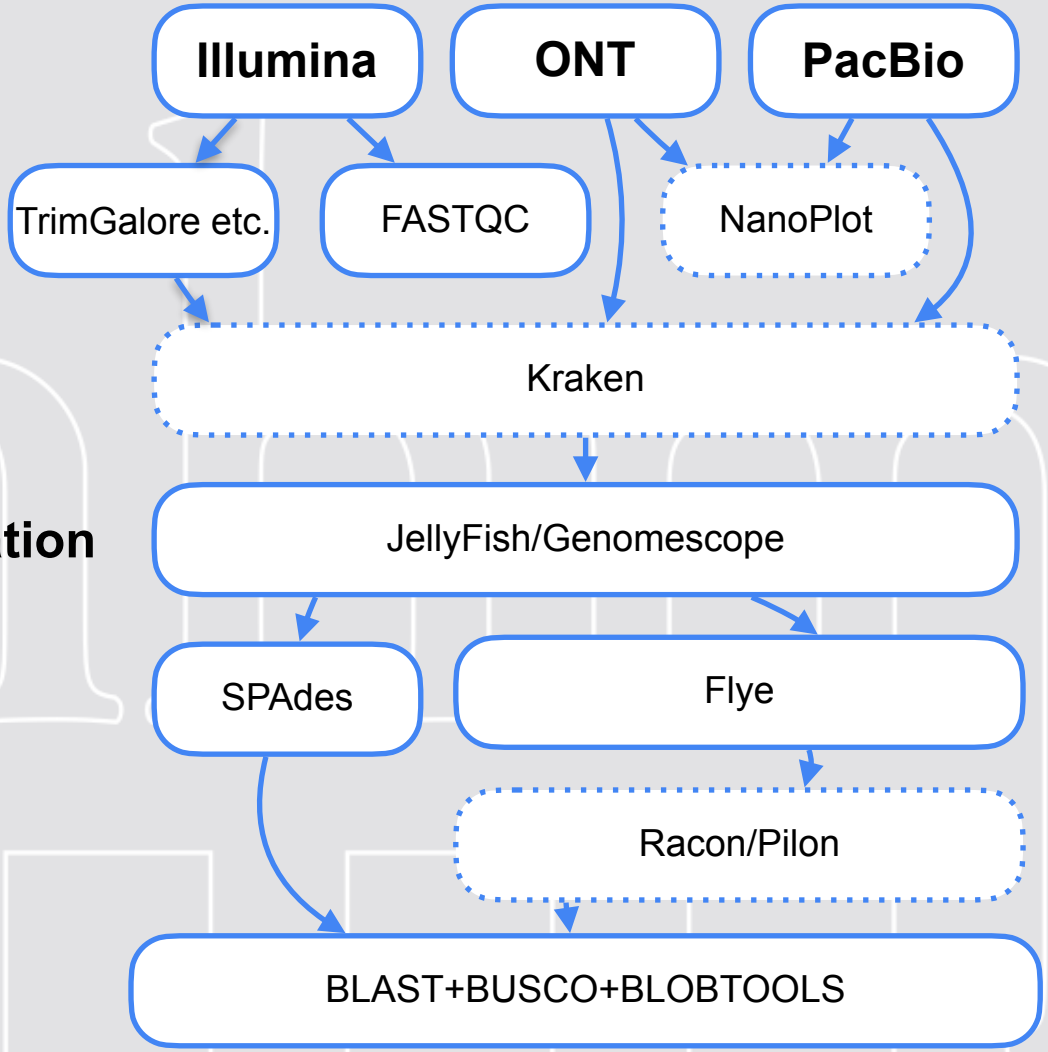
- Then, follow the instruction in shell/assembly_pipeline.sh

For a fully automated Pipeline, see here: https://github.com/nhmvienna/AutDeNovo

AutDeNovo

https://github.com/nhmvienna/AutDeNovo

1) Copying raw data
2) Filter, trim, QC
3) Decontamination
4) Genomesize estimation
5) De-novo assembly
6) Polishing
7) Assembly QC



Thank you!!!


PHASE TWO: INTERPRETATION
I THINK I FOUND A CORNER PIECE.
3 BILLION PIECES
GENOME