Session 2.3 – Ensamblado

Isabel Cuesta

BU-ISCIII

Unidades Comunes Científico Técnicas – SGSAFI-ISCIII

13 al 17 Noviembre 2023

3ª Edición

Programa Formación Continua, ISCII

Assembly

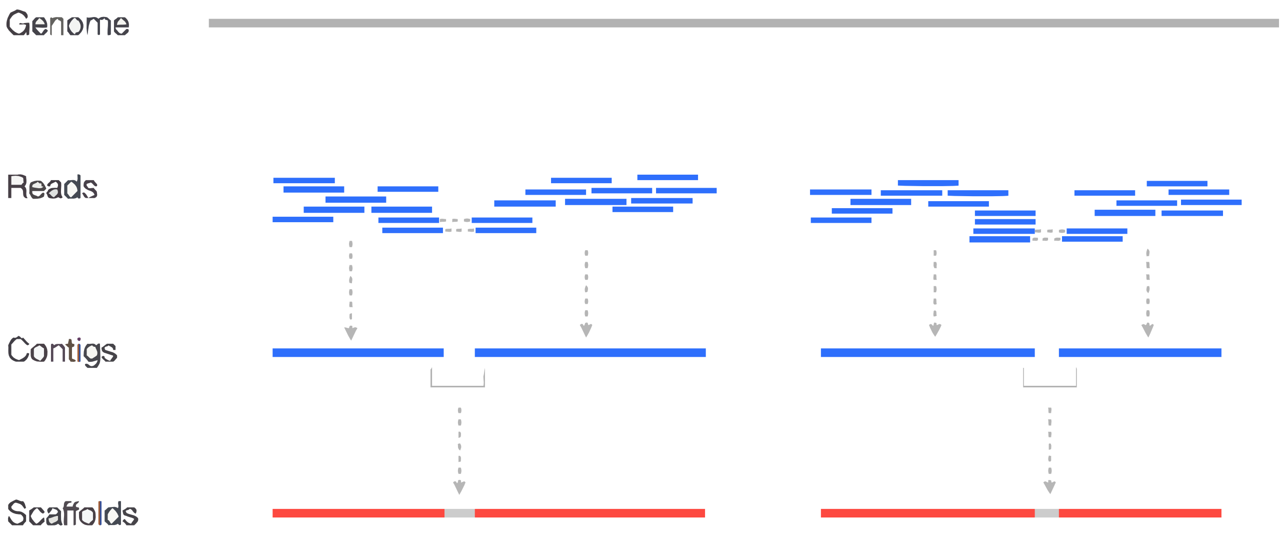
Reconstruct the sequence of the original DNA from shorter DNA sequences or small fragments known as reads

- ***De novo:*** with no previous knowledge of the genome to be assembled. It overlap the end of the end of each read in order to create a longer sequence.

- ***Assembly with reference:*** A similar but not identical genome guides the assembly process. Map reads over supplied genome.

Assembly: contig y scaffold

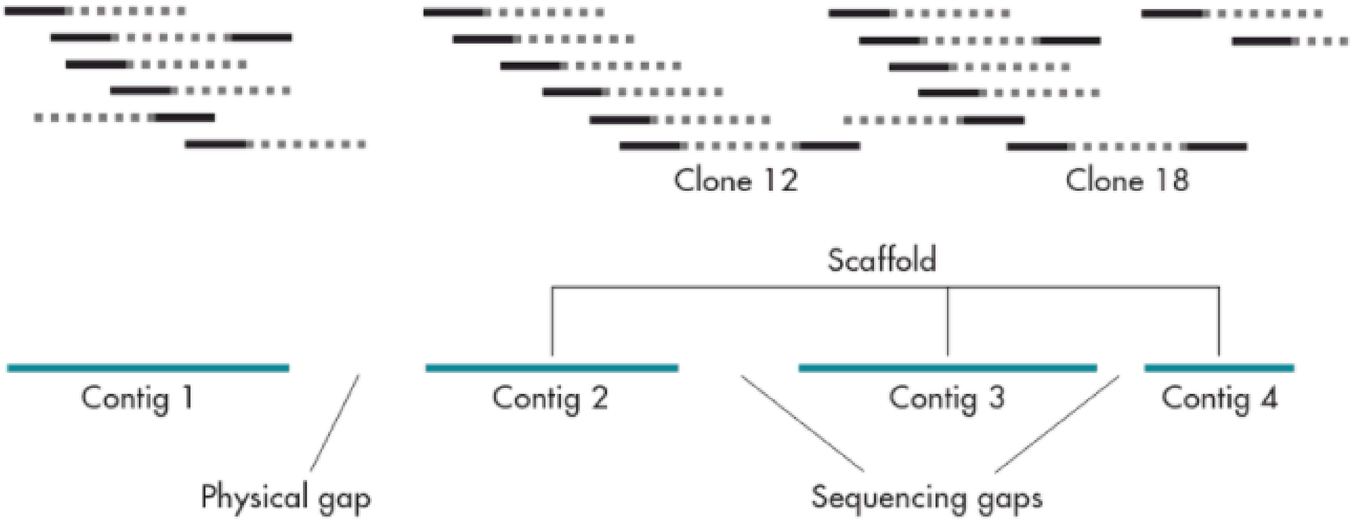
- **Contig:** continuous sequence made up of overlaping shorter sequences
- **Scaffold:** two or more contigs located and rearranged according to spatial information (pair-end, mate pair, reference)



https://www.biostars.org/p/253222/

Assembly: gaps

- **Sequencing gaps:** Position and orientation known by spatial information
- **Physical gaps:** No information about adjacent contigs



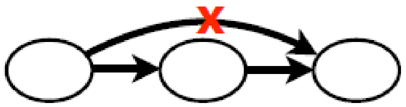
Gene Cloning, Lodge *et al*.

Assembly: Algorithms

- **Overlap, Layout, Consensus (OLC - overlap graph):**
  - O - first overlaps among all the reads are found
  - L - then it carries out a layout of all the reads and overlaps information on a graph
    - Removes redundant and low quality overlaps
  - C - and finally the consensus sequence is inferred

Ex. Newbler, Mira, Celera Assembler, CAP3, PCAP, Phrap, Phusion.

```
X: CTCGGCCCTAGG
      ||| |||||
Y:    GGCTCTAGGCCC
```



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

TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

Take reads that make up a contig and line them up

Take *consensus*, i.e. majority vote

https://pt.slideshare.net/anton_alexandrov/combining-de-bruijn-graph-overlap-graph-and-microassembly/12?smtNoRedir=1

Assembly: Algorithms

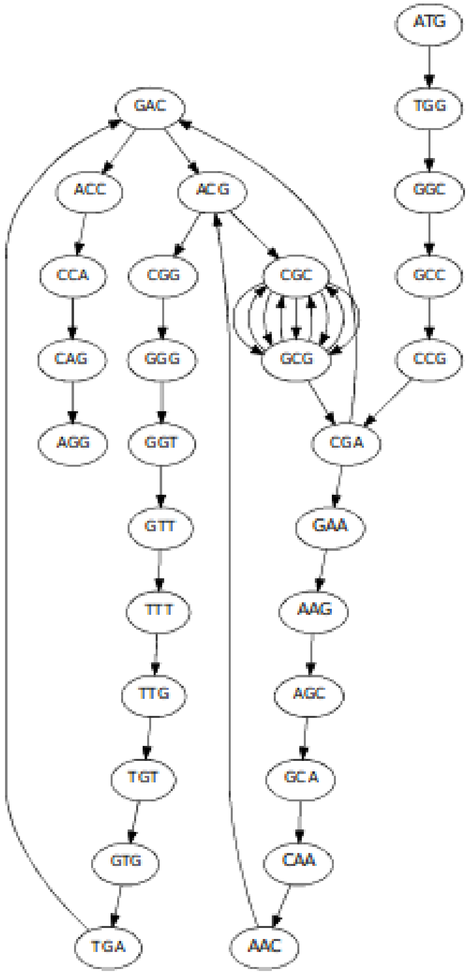
- **De Brujin Graph (DBG: k-mer graph)**

Chopping reads into much shorter k-mers (fixed length fragments) and then using all the k-mers to form a DBG and infer the contigs.

- Nodes in the graph are k-mers

- Edges represent consecutive k-mers (which overlap by k-n symbols)

Ex. SPAdes, ABySS, Velvet, AllPaths, Soap….

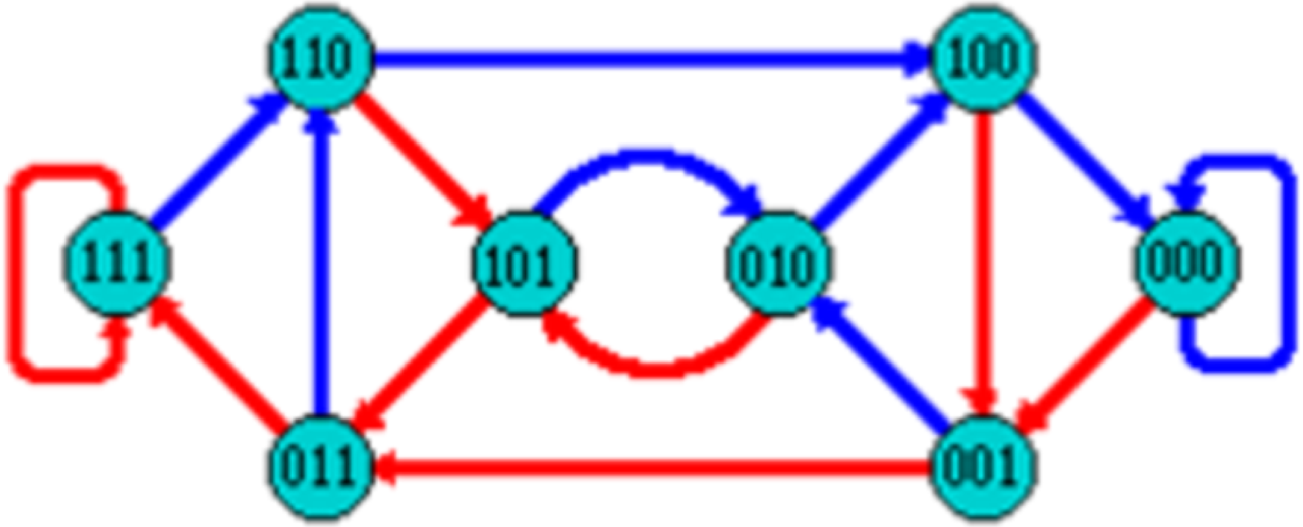


https://medium.com/@han_chen

de Bruijn Graphs

- A directed graph of sequences of symbols
- Nodes in the graph are k-mers
- Edges represent consecutive k-mers (which overlap by k-1 symbols)

Consider the 2 symbol alphabet (0 & 1) de Bruijn Graph for k =3



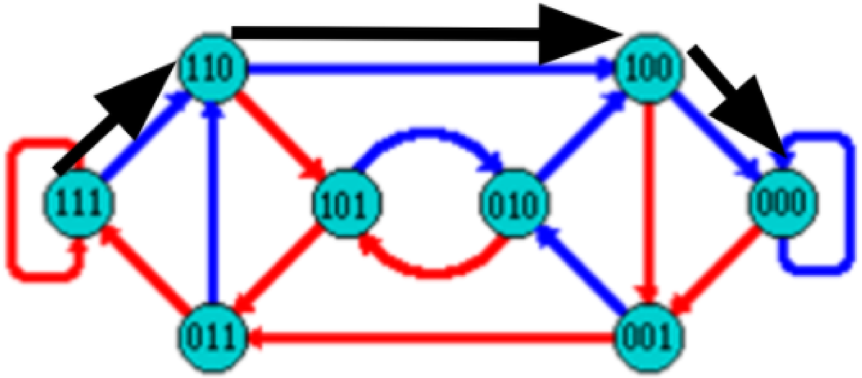
https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

Producing sequences

- Sequences of symbols are produced by moving through the graph

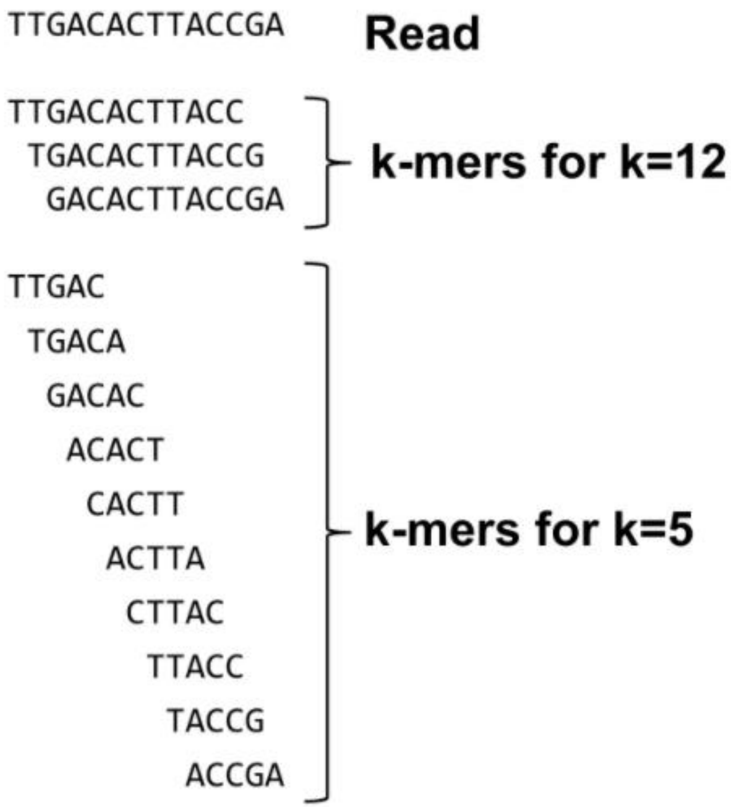
e.g. 111000 = 111 -> 110 -> 100 -> 000

```
1 1 1
  1 1 0
    1 0 0
      0 0 0
-----------
1 1 1 0 0 0
```



https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

What are K-mers?



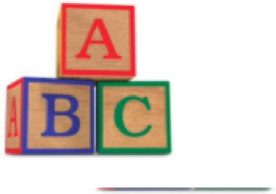
TTGACACTTACCGA Read

TTGACACTTACC
TGACACTTACCG
GACACTTACCGA
k-mers for k=12

TTGAC
TGACA
GACAC
ACACT
CACTT
ACTTA
CTTAC
TTACC
TACCG
ACCGA
k-mers for k=5

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

K-mers de Bruijn graph



Example #1:

HAPPI   PINE   INESS   APPIN

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

K-mers de Bruijn graph

Example #1:

HAPPI PINE INESS APPIN

All 4-mers:

HAPP PINE INES APPI

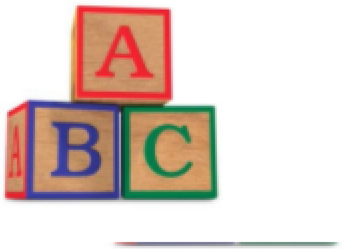
APPI NESS PPIN

Unique 4-mers:

HAPP APPI PINE PPIN INES NESS

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

K-mers de Bruijn graph



Example #1:

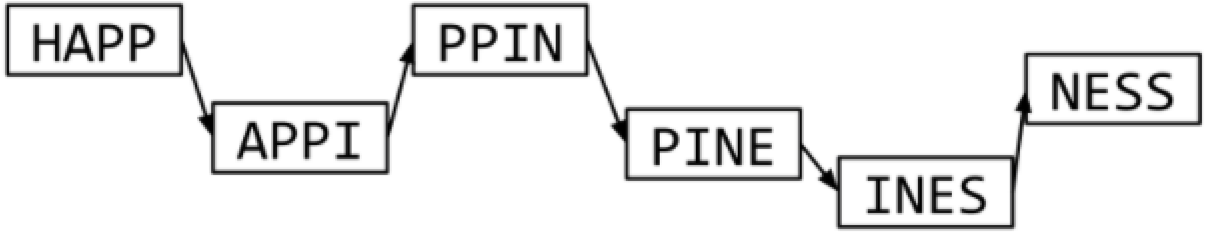
HAPPI PINE INESS APPIN

k = 4 k-mers:

HAPP APPI

PINE PPIN

INES NESS



https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

K-mers de Bruijn graph

Example #1:

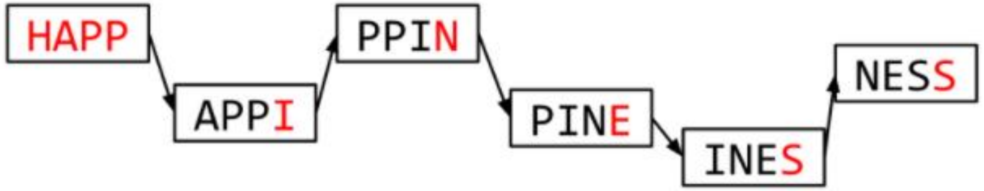
HAPPI PINE INESS APPIN

k = 4 k-mers:

HAPP APPI

PINE PPIN

INES NESS



HAPP

APPI

PPIN

PINE

INES

NESS

HAPPINESS

Easy!

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

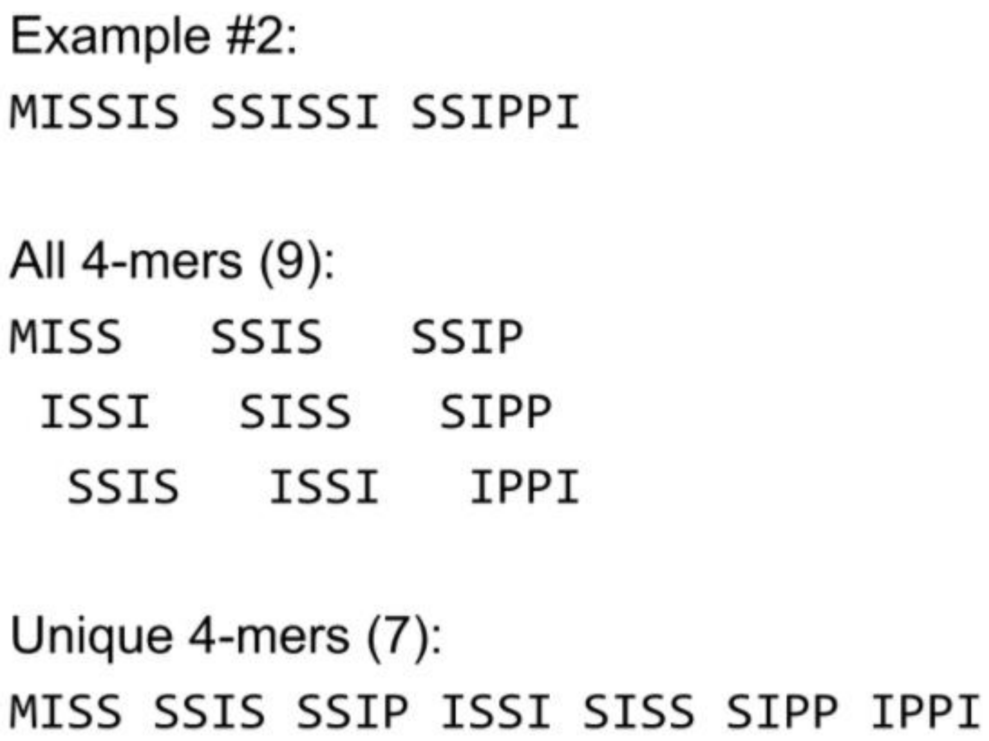
The problem of repeats

Example #2:

```
MISSIS SSISSI SSIPPI
```

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

The problem of repeats



Example #2:

```
MISSIS SSISSI SSIPPI
```

All 4-mers (9):

```
MISS   SSIS   SSIP
 ISSI   SISS   SIPP
  SSIS   ISSI   IPPI
```

Unique 4-mers (7):

```
MISS SSIS SSIP ISSI SISS SIPP IPPI
```

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

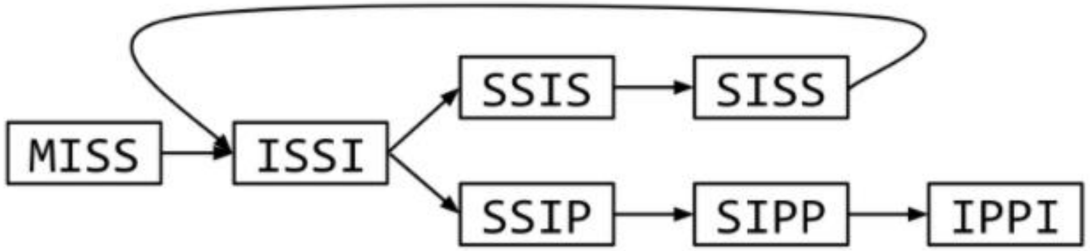
The problem of repeats

Example #2:

MISSIS SSISSI SSIPPI

All 4-mers:

MISS ISSI SSIS SISS SSIP SIPP IPPI



https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

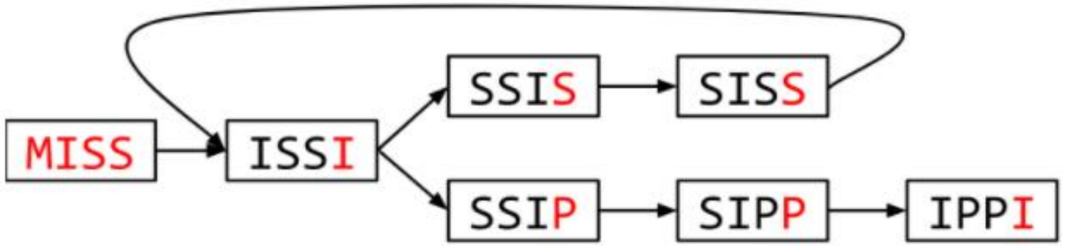
The problem of repeats

Example #2:

MISSIS SSISSI SSIPPI

All 4-mers:

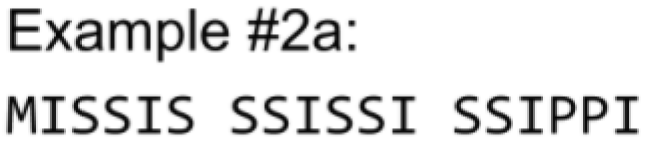
MISS ISSI SSIS SISS SSIP SIPP IPPI



MISSISSIPPI or MISSISSISSISSIPPI or ...

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

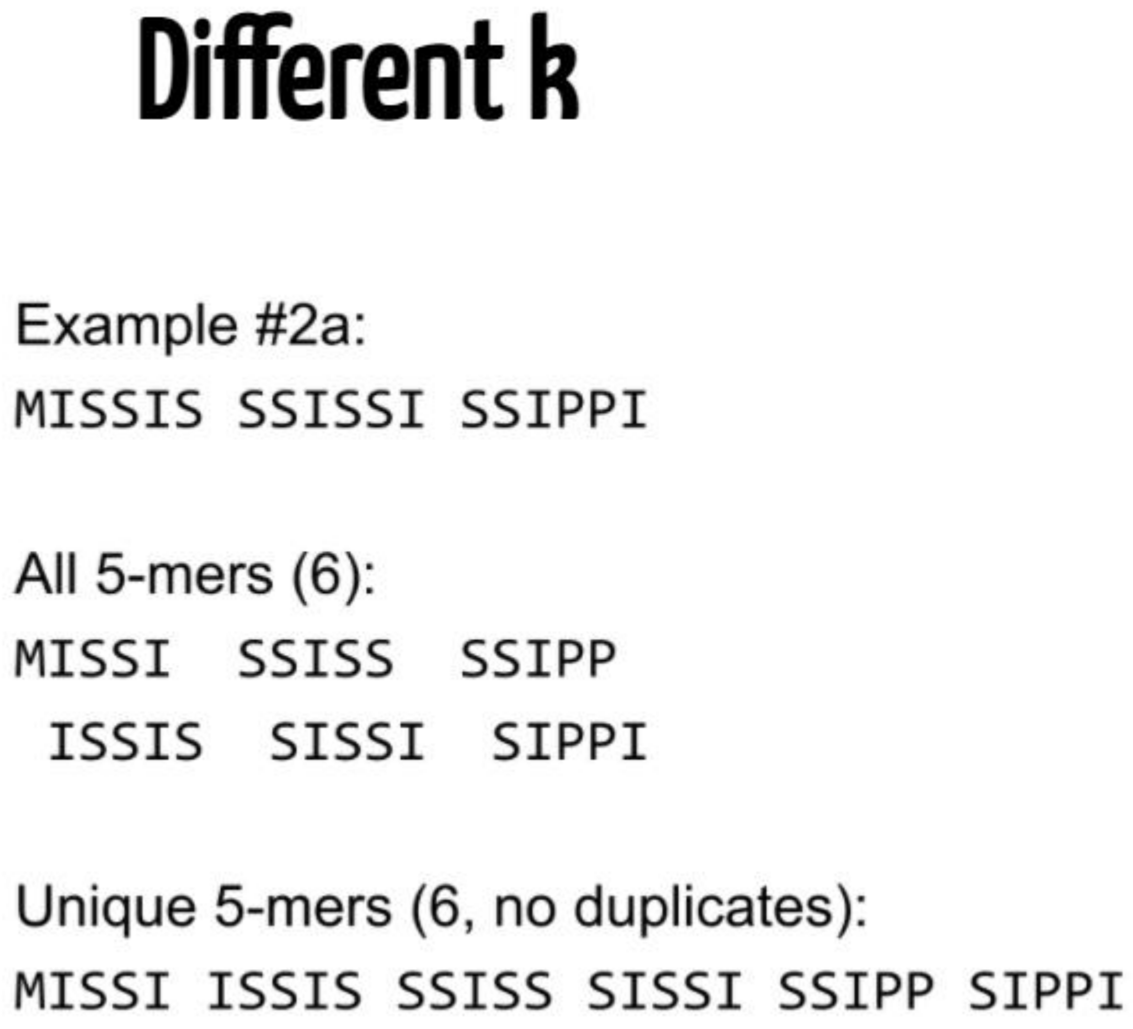
Different k



Example #2a:

MISSIS SSISSI SSIPPI

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23



Different k

Example #2a:

```
MISSIS SSISSI SSIPPI
```

All 5-mers (6):

```
MISSI  SSISS  SSIPP
 ISSIS  SISSI  SIPPI
```

Unique 5-mers (6, no duplicates):

```
MISSI ISSIS SSISS SISSI SSIPP SIPPI
```

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

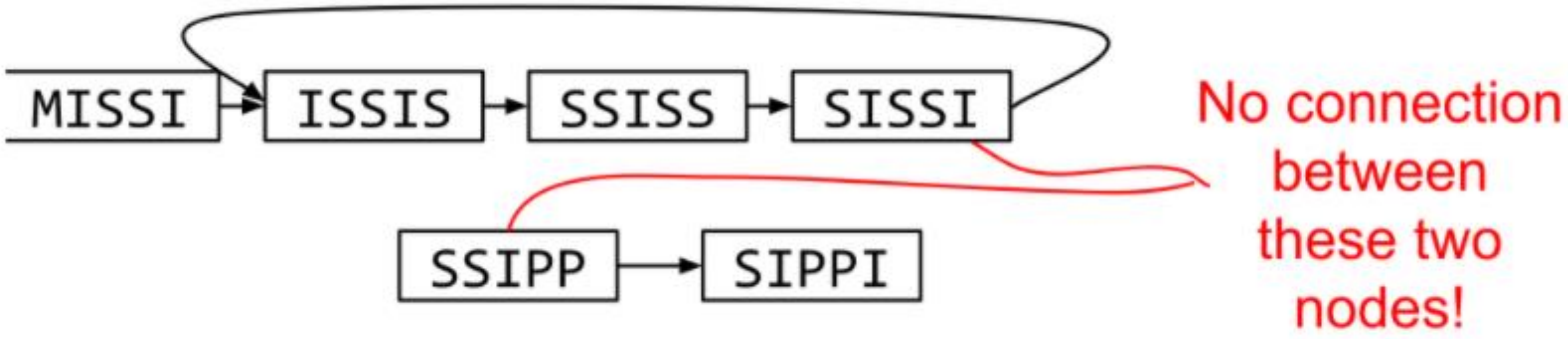
Different k

Example #2a:

MISSIS SSISSI SSIPPI

This time k = 5 k-mers:

MISSI ISSIS SSISS SISSI SSIPP SIPPI



https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

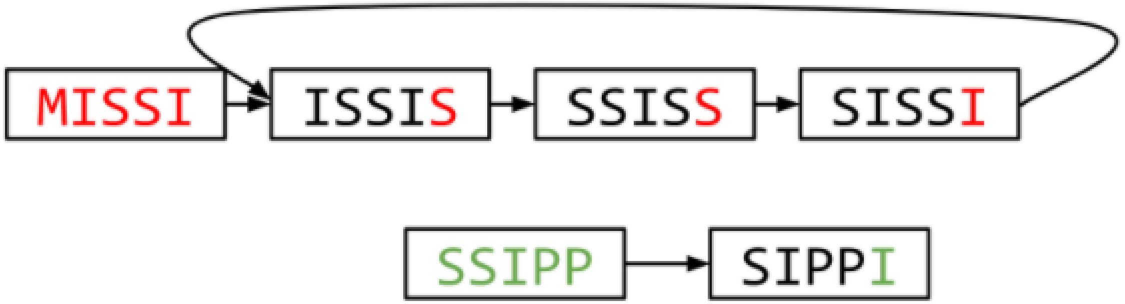
Different k

Example #2a:

MISSIS SSISSI SSIPPI

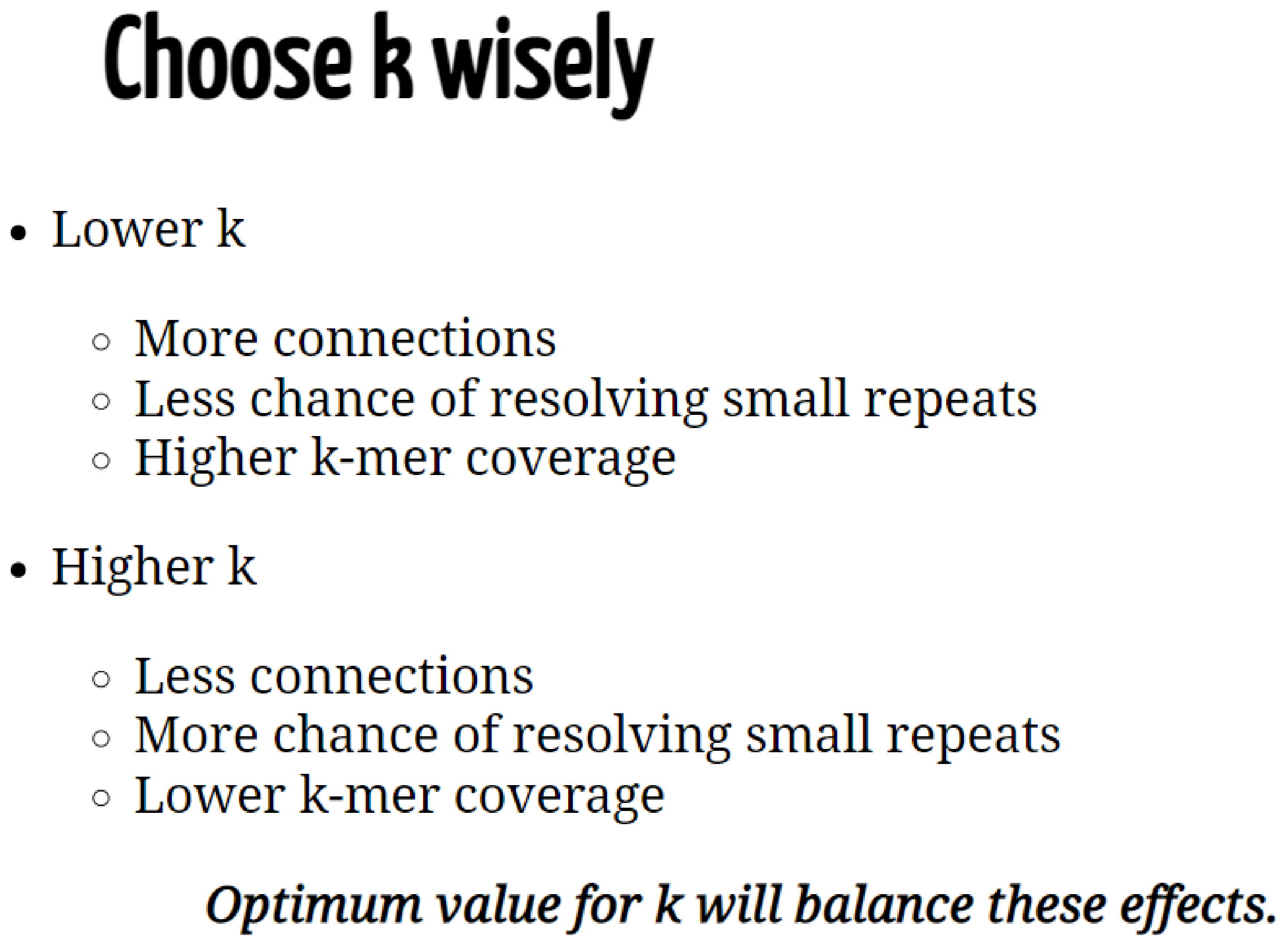
This time k = 5 k-mers:

MISSI ISSIS SSISS SISSI SSIPP SIPPI



MISSISSIS SSIPPI

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23



Choose k wisely

- Lower k
  - More connections
  - Less chance of resolving small repeats
  - Higher k-mer coverage
- Higher k
  - Less connections
  - More chance of resolving small repeats
  - Lower k-mer coverage

Optimum value for k will balance these effects.

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

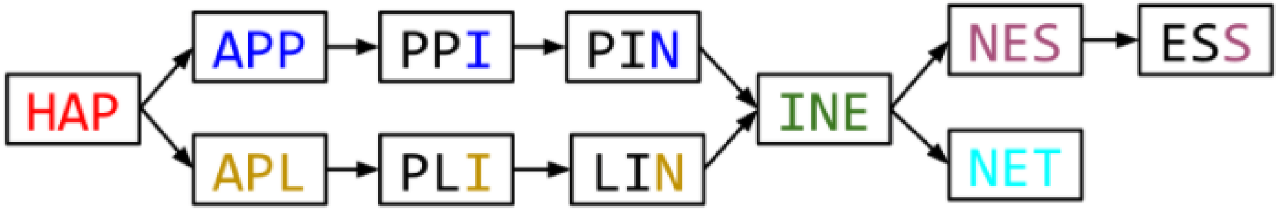
Read errors

Example #3:

HAPPI INESS APLIN PINET

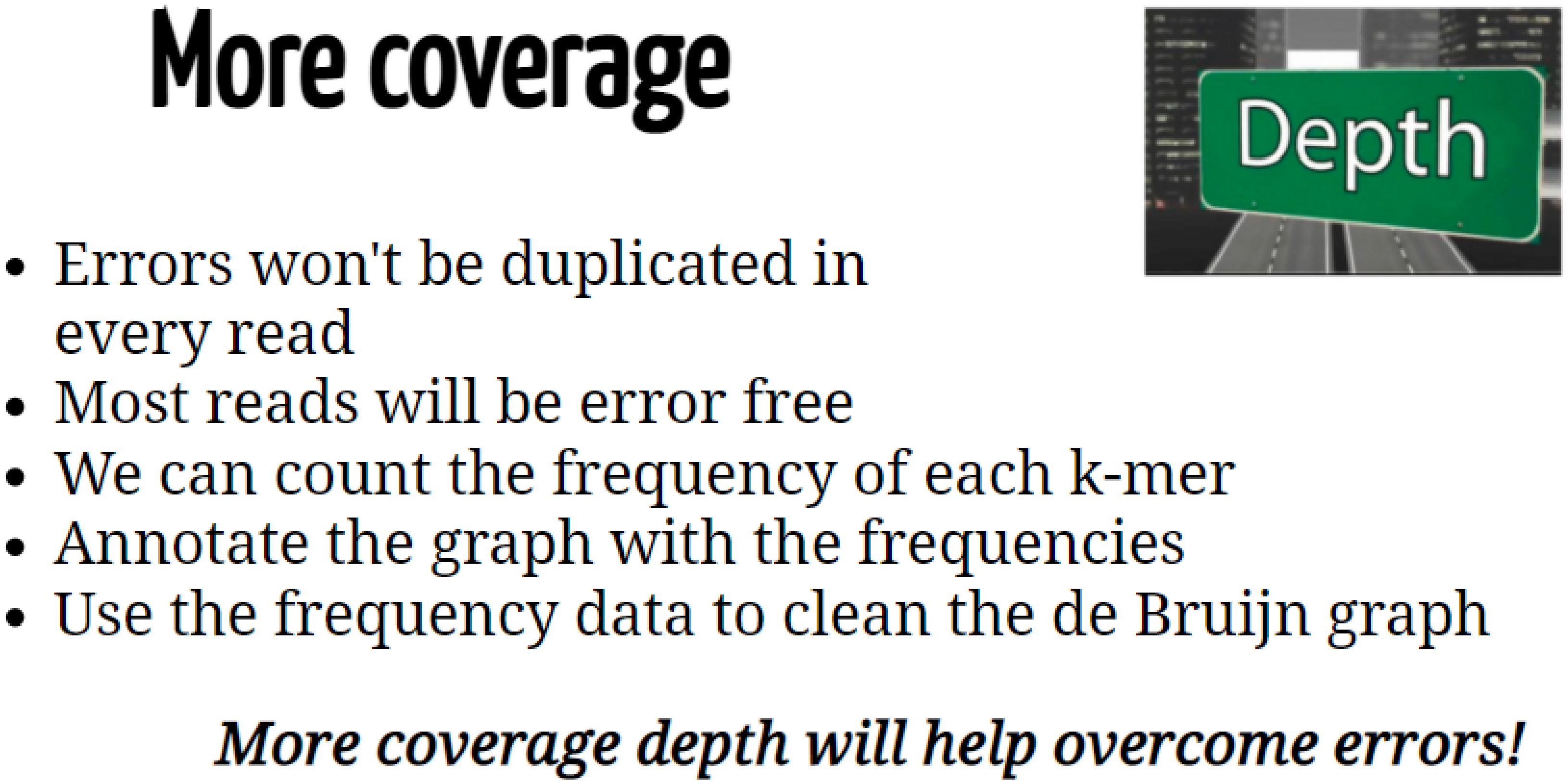
k = 3 k-mers:

HAP APP PPI INE NES ESS APL PLI LIN PIN NET



6 contigs: HAP APPIN APLIN INE NESS NET

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23



More coverage

- Errors won't be duplicated in every read
- Most reads will be error free
- We can count the frequency of each k-mer
- Annotate the graph with the frequencies
- Use the frequency data to clean the de Bruijn graph

More coverage depth will help overcome errors!

https://galaxyproject.github.io/training-material/topics/assembly/tutorials/debruijn-graph-assembly/slides.html#23

SPAdes

- de Bruijn graph assembler by Pavel Pevzner's group out of St. Petersburg

- Uses multiple k-mers to build the graph
    - Graph has connectivity **and** specificity
    - Usually use a low, medium and high k-mer size together.

- Performs error correction on the reads first

- Maps reads back to the contigs and scaffolds as a check

- Under active development

- Much slower than Velvet

- Should be used in preference to Velvet now.

Assembly: Scaffolding

- **From draft:**

  Order contigs (Nucmer, if there is reference it can be used to align and guide)

  Fill the GAPs (GapFiller, fill sequencing gap (not physical gap)

  Solve repeated sequence ambiguities (Expander)

  Resequence with different library:

  - Longer fragments and/or distance

- **Tools for assembly improvement**

  SSPACE (Scaffolding) REAPR (evaluate scaffolding, breaking incorrect scaffolds)

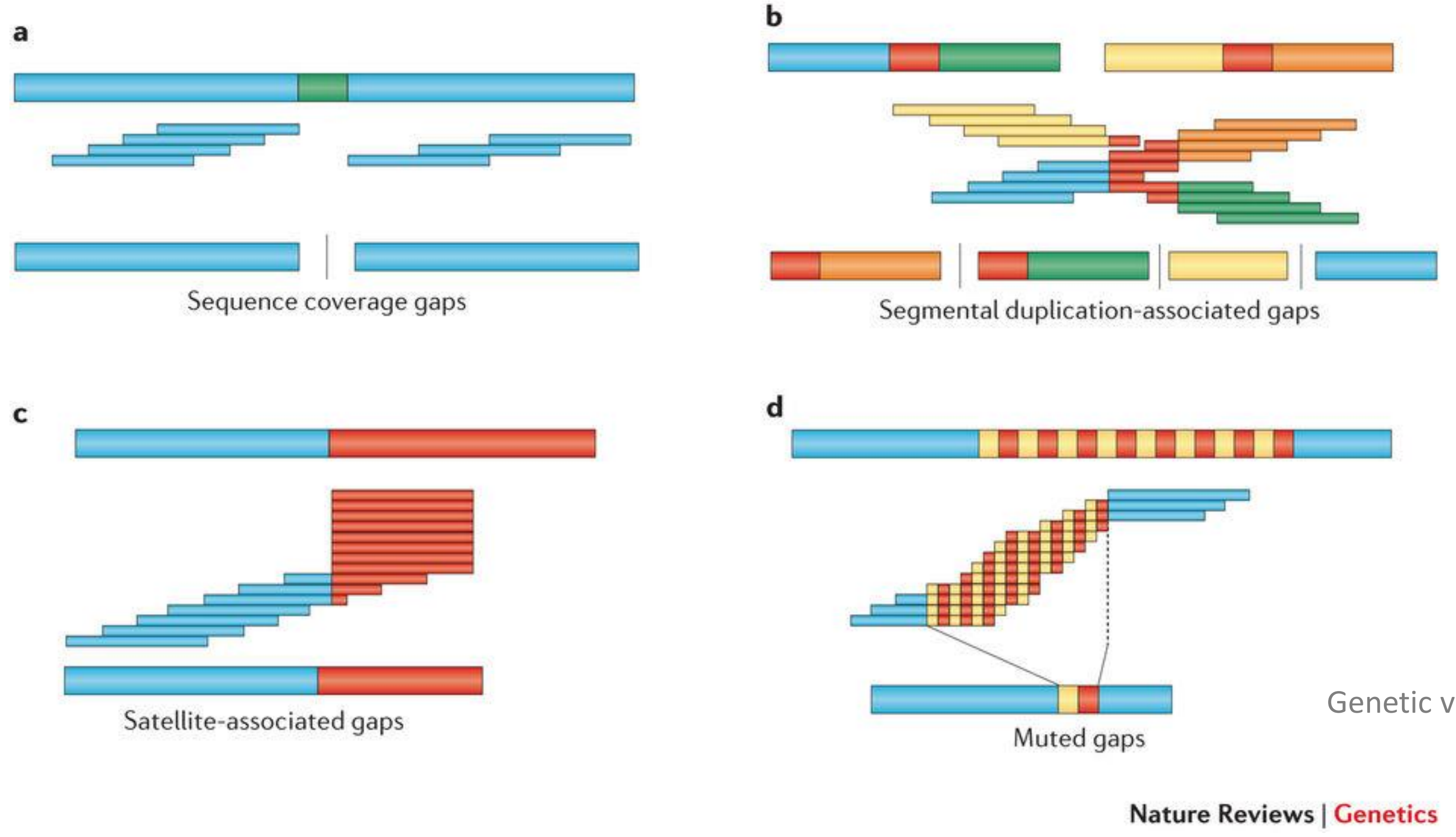
- **Assembly visualyzing**

  Artemis, ACT (compare two or more sequences), Icarus (Quast)

A move back to OLC

- New long read technologies
  - PacBio and MinIon
- Assemblers: HGap, CANU
  - Use overlap, layout consensus approach
- CANU can perform hybrid assemblies with long and short reads

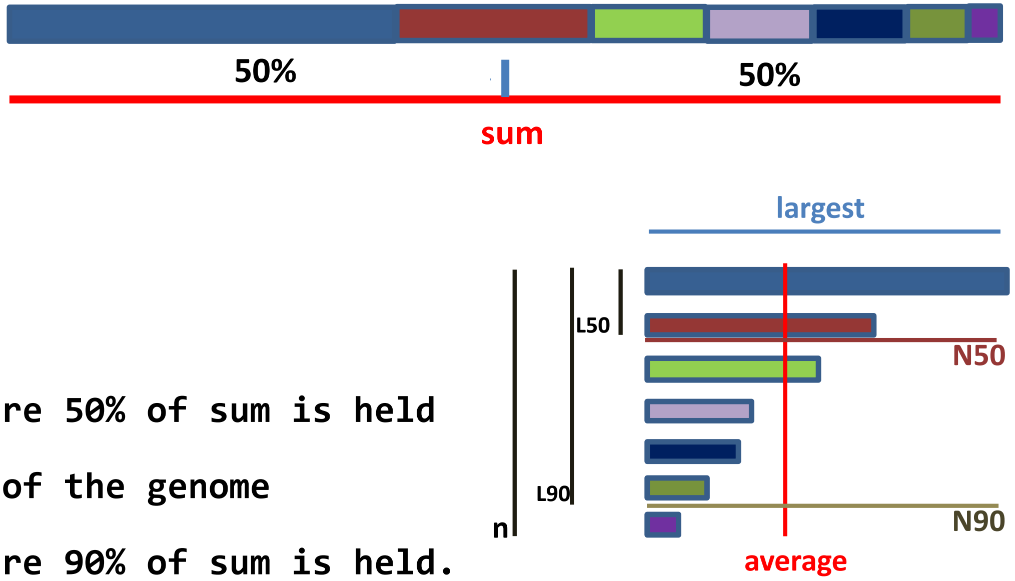
Ensamblado: Errores


a
Sequence coverage gaps
b
Segmental duplication-associated gaps
c
Satellite-associated gaps
d
Muted gaps
Nature Reviews | Genetics

- **A. Gaps – región del genoma sin secuenciar**
- **B. Duplicaciones de gran tamaño**
  - Quimeras
- **Regiones repetidas colapsadas**
  - **C. Terminales**
  - **D. Intersticiales**

Genetic variation and the de novo assembly of human genomes
Chaisson *et al*. 2015

Assembly: Metrics



- sum = total bases number
- n = contigs number
- average = average contig length
- largest = largest contig
- N50 = length of the shortest contig where 50% of sum is held
- L50 = number of contigs which have 50% of the genome
- N90 = length of the shortest contig where 90% of sum is held.
- L90 = number of contigs which have 90% of the genome

Assembly: Evaluation

- Software that evaluate differets algorithms & parameters
  iMetAMOS, *Koren et al., BMCBioinformatics 2014, 15:126*
  GAGE-B, *Magoc et al.,* Bioinformatics 2013,29(14):1718-25

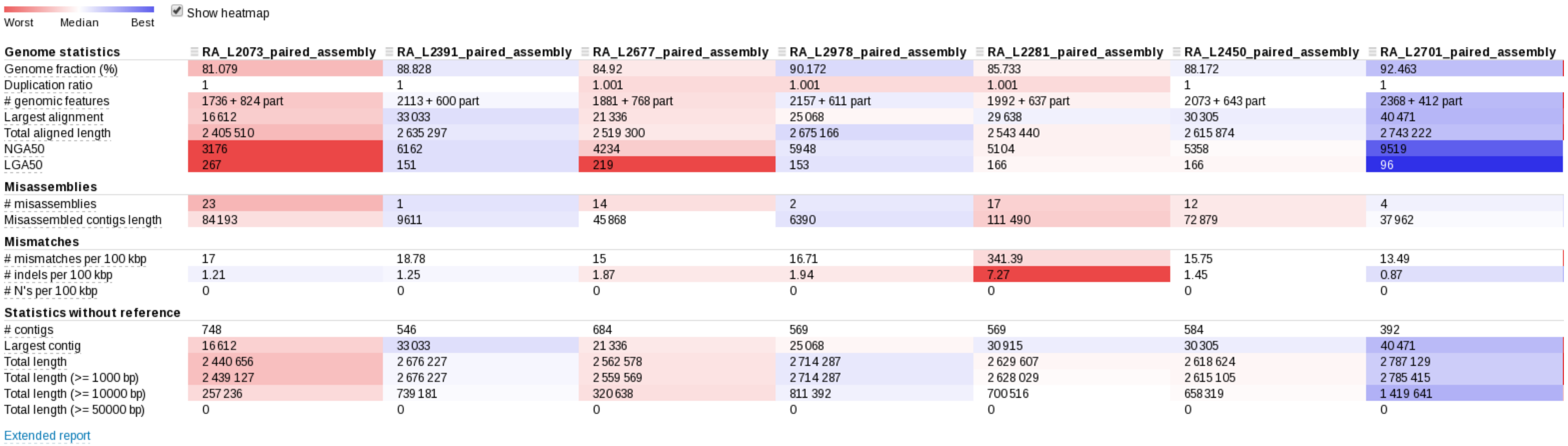
- **Graph evaluation**: Bandage, Wick R.R., Schultz M.B., Zobel J. & Holt K.E. (2015)

- **Assembly evaluation**: Quast, *Gurevich et al., Bioinformatics 2013, 29:8*

- **Metrics for a good assembly:**
  Large N50
  Sum closest to expected
  Low n
  Low L50

Assembly: Evaluation - Quast

- Assembly evaluation: Quast, *Gurevich et al., Bioinformatics 2013, 29:8*

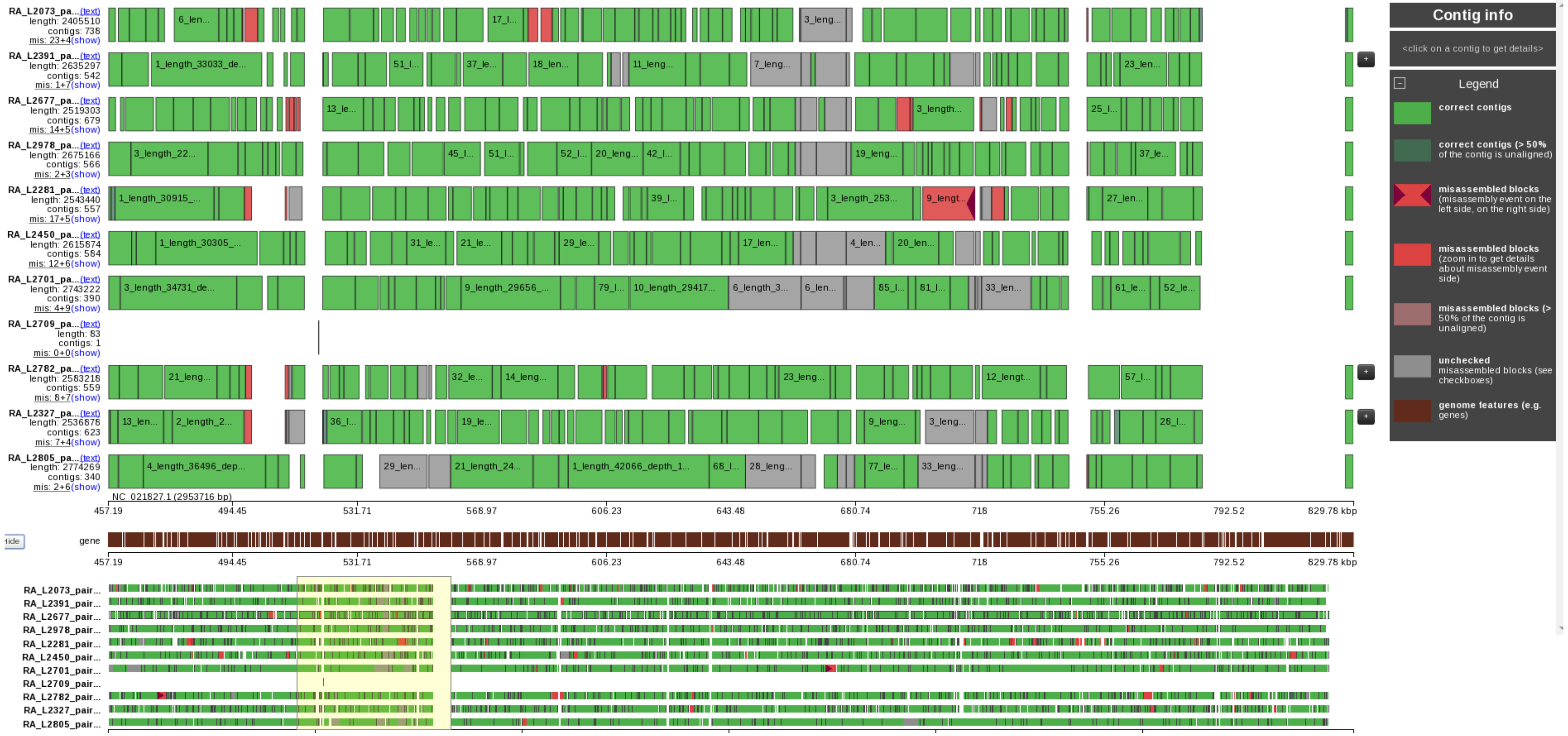


Genome statistics	RA_L2073_paired_assembly	RA_L2391_paired_assembly	RA_L2677_paired_assembly	RA_L2978_paired_assembly	RA_L2281_paired_assembly	RA_L2450_paired_assembly	RA_L2701_paired_assembly
Genome fraction (%)	81.079	88.828	84.92	90.172	85.733	88.172	92.463
Duplication ratio	1	1	1.001	1.001	1.001	1	1
# genomic features	1736 + 824 part	2113 + 600 part	1881 + 768 part	2157 + 611 part	1992 + 637 part	2073 + 643 part	2368 + 412 part
Largest alignment	16 612	33 033	21 336	25 068	29 638	30 305	40 471
Total aligned length	2 405 510	2 635 297	2 519 300	2 675 166	2 543 440	2 615 874	2 743 222
NGA50	3176	6162	4234	5948	5104	5358	9519
LGA50	267	151	219	153	166	166	96
Misassemblies							
# misassemblies	23	1	14	2	17	12	4
Misassembled contigs length	84 193	9611	45 868	6390	111 490	72 879	37 962
Mismatches							
# mismatches per 100 kbp	17	18.78	15	16.71	341.39	15.75	13.49
# indels per 100 kbp	1.21	1.25	1.87	1.94	7.27	1.45	0.87
# N's per 100 kbp	0	0	0	0	0	0	0
Statistics without reference							
# contigs	748	546	684	569	569	584	392
Largest contig	16 612	33 033	21 336	25 068	30 915	30 305	40 471
Total length	2 440 656	2 676 227	2 562 578	2 714 287	2 629 607	2 618 624	2 787 129
Total length (>= 1000 bp)	2 439 127	2 676 227	2 559 569	2 714 287	2 628 029	2 615 105	2 785 415
Total length (>= 10000 bp)	257 236	739 181	320 638	811 392	700 516	658 319	1 419 641
Total length (>= 50000 bp)	0	0	0	0	0	0	0

Extended report

Assembly: Evaluation - Quast

- Assembly evaluation: Quast, *Gurevich et al., Bioinformatics 2013, 29:8*



RA_L2073_pa...(text)
length: 2405510
contigs: 738
mis: 23+4(show)

RA_L2391_pa...(text)
length: 2635297
contigs: 542
mis: 1+7(show)

RA_L2677_pa...(text)
length: 2519303
contigs: 679
mis: 14+5(show)

RA_L2978_pa...(text)
length: 2675166
contigs: 566
mis: 2+3(show)

RA_L2281_pa...(text)
length: 2543440
contigs: 557
mis: 17+5(show)

RA_L2450_pa...(text)
length: 2615874
contigs: 584
mis: 12+6(show)

RA_L2701_pa...(text)
length: 2743222
contigs: 390
mis: 4+9(show)

RA_L2709_pa...(text)
length: 83
contigs: 1
mis: 0+0(show)

RA_L2782_pa...(text)
length: 2583218
contigs: 559
mis: 8+7(show)

RA_L2327_pa...(text)
length: 2536878
contigs: 623
mis: 7+4(show)

RA_L2805_pa...(text)
length: 2774269
contigs: 340
mis: 2+6(show)

Contig info

<click on a contig to get details>

Legend

🟩 **correct contigs**

🟩 **correct contigs (> 50%** of the contig is unaligned)

🟥 **misassembled blocks** (misassembly event on the left side, on the right side)

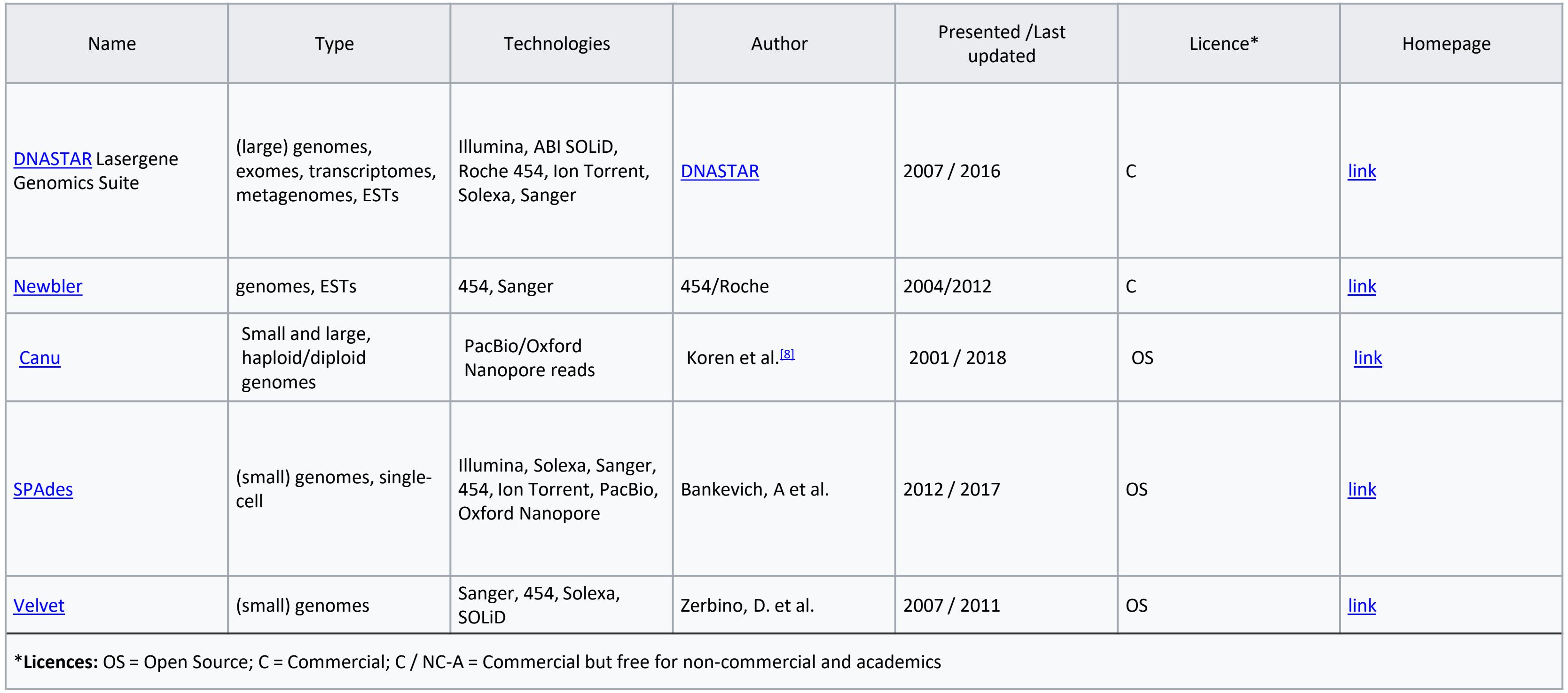
🟥 **misassembled blocks** (zoom in to get details about misassembly event side)

🟥 **misassembled blocks (>** 50% of the contig is unaligned)

⬜ **unchecked** misassembled blocks (see checkboxes)

🟫 **genome features (e.g.** genes)

Assembly: Assemblers

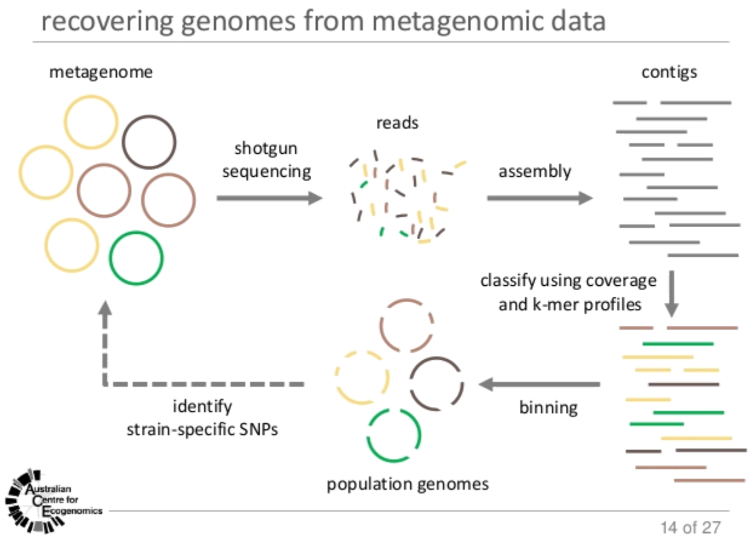


Name	Type	Technologies	Author	Presented /Last updated	Licence*	Homepage
DNASTAR Lasergene Genomics Suite	(large) genomes, exomes, transcriptomes, metagenomes, ESTs	Illumina, ABI SOLiD, Roche 454, Ion Torrent, Solexa, Sanger	DNASTAR	2007 / 2016	C	link
Newbler	genomes, ESTs	454, Sanger	454/Roche	2004/2012	C	link
Canu	Small and large, haploid/diploid genomes	PacBio/Oxford Nanopore reads	Koren et al.[8]	2001 / 2018	OS	link
SPAdes	(small) genomes, single-cell	Illumina, Solexa, Sanger, 454, Ion Torrent, PacBio, Oxford Nanopore	Bankevich, A et al.	2012 / 2017	OS	link
Velvet	(small) genomes	Sanger, 454, Solexa, SOLiD	Zerbino, D. et al.	2007 / 2011	OS	link

***Licences:** OS = Open Source; C = Commercial; C / NC-A = Commercial but free for non-commercial and academics

Assembly: Specials assemblers

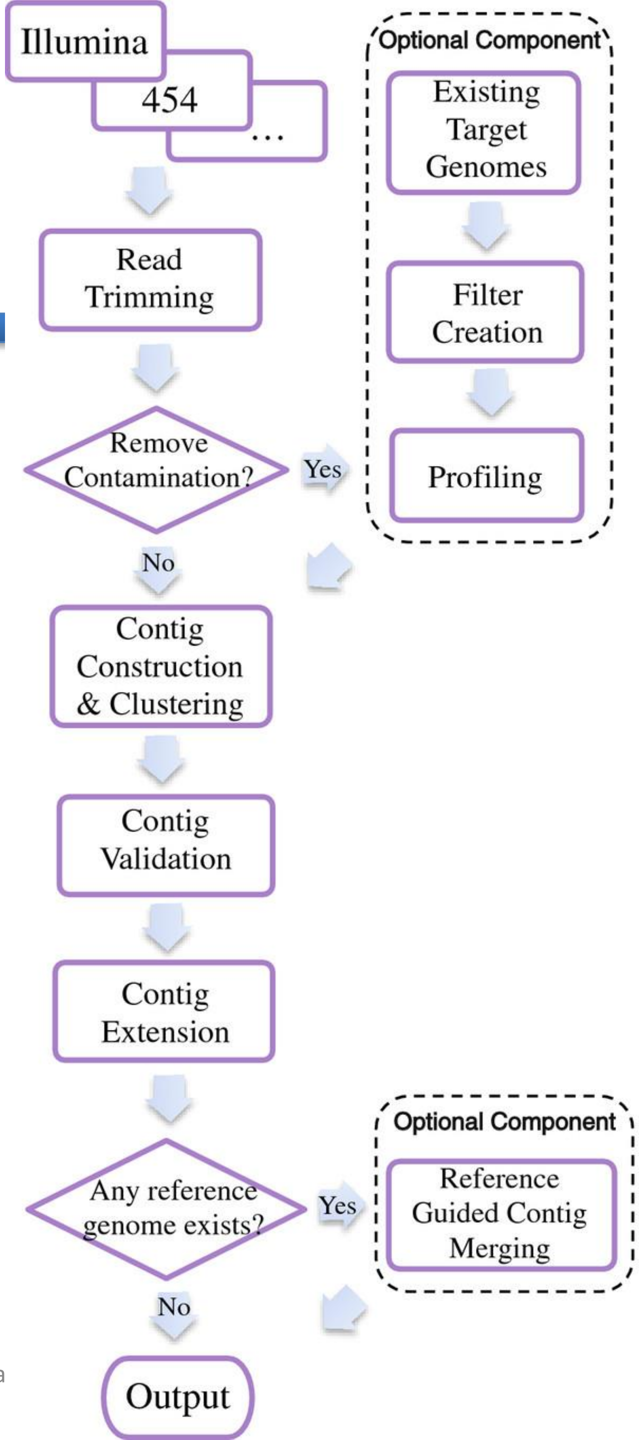
- **Diploid genomes**
- **Metagenomics**
- **Plasmids**
- **Transcriptome**
- **Virus**
  - VICUNA: population consensus genome assembly
  - IVA: assembler for RNA viruses



recovering genomes from metagenomic data
metagenome
contigs
reads
shotgun sequencing
assembly
classify using coverage and k-mer profiles
binning
identify strain-specific SNPs
population genomes
Australian Centre for Ecogenomics
14 of 27

VICUNA & IVA

- VICUNA is a de novo assembly program targeting populations with high mutation rates



Rnaviral Spades: CoronaSpades

https://github.com/ablab/spades#sec1.2

Coronavirus assembly mode for SPAdes assembler (also known as **coronaSPAdes**).

It allows to assemble full-length coronaviridae genomes from the transcriptomic and metatranscriptomic data. Algorithmically, coronaSPAdes is an rnaviralSPAdes that uses the set of HMMs from Pfam SARS-CoV-2 2.0 set as well as additional HMMs

HMM-guided mode: amino acid profile HMMs are aligned to the edges of assembly graph

Bioinformatics 2021 Dec 22;38(1):1-8. doi: 10.1093/bioinformatics/btab597

METAVIRALSPADES: assembly of viruses from metagenomic data

Bioinformatics, 36(14), 2020, 4126-4129
doi: 10.1093/bioinformatics/btaa490

METAVIRALSPADES tool for identifying viral genomes in metagenomic assembly graphs that is based on analyzing variations in the coverage depth between viruses and bacterial chromosomes

METAVIRALSPADES includes **VIRALASSEMBLY, VIRALVERIFY** and **VIRALCOMPLETE** modules
https://github.com/ablab/spades/tree/metaviral_publication,
https://github.com/ablab/viralVerify/
https://github.com/ablab/viralComplete/.

viralVerify classifies contigs (output of metaviralSPAdes or other assemblers) **as viral, non-viral or uncertain**, based on gene content. Also for non-viral contigs it can optionally provide plasmid/non-plasmid classification.
viralVerify predicts genes in the contig using Prodigal in the metagenomic mode, runs hmmsearch on the predicted proteins and classifies the contig as vrial or non-viral by applying the Naive Bayes classifier (NBC). For the set of predicted HMMs, viralVerify uses trained NBC to classify this set to be viral or chromosomal.

viralComplete is intended for completeness verification of novel viral contigs. It heavily relies on following assumptions:
1.Virus genome size is consistent across the viral family.
2.If a newly constructed viral contig is complete and belongs to a known family of viruses then its gene content should be similar to the gene content of a known virus.
We thus compute the “similarity” of a given contig (based on the Naive Bayesian Classifier) to each known virus from the RefSeq database, and check whether the most similar known virus have length similar to the contig length.

Haploflow: strain-resolved de novo assembly of viral genomes

Fritz et al. Genome Biology (2021) 22:212
https://doi.org/10.1186/s13059-021-02426-8

A deBruijn graph-based assembler for de novo genome assembly of viral strains from mixed sequence samples using a novel flow algorithm.

Haploflow reconstructs viral strain genomes from patient HCMV samples and SARS-CoV-2 wastewater samples identical to clinical isolates

Haploflow: strain-resolved de novo assembly of viral genomes

Fritz et al. Genome Biology (2021) 22:212
https://doi.org/10.1186/s13059-021-02426-8

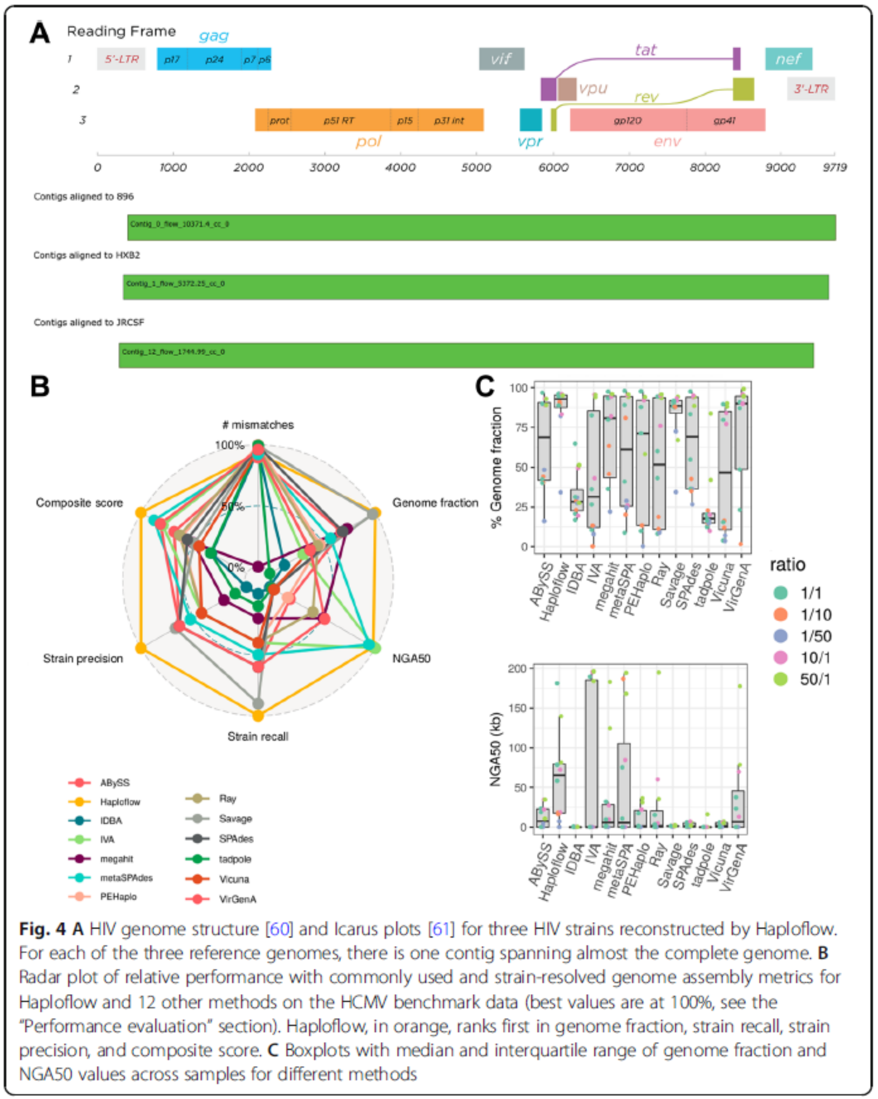


Fig. 4 **A** HIV genome structure [60] and Icarus plots [61] for three HIV strains reconstructed by Haploflow. For each of the three reference genomes, there is one contig spanning almost the complete genome. **B** Radar plot of relative performance with commonly used and strain-resolved genome assembly metrics for Haploflow and 12 other methods on the HCMV benchmark data (best values are at 100%, see the "Performance evaluation" section). Haploflow, in orange, ranks first in genome fraction, strain recall, strain precision, and composite score. **C** Boxplots with median and interquartile range of genome fraction and NGA50 values across samples for different methods

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

Metagenomic assemblers typically use de Bruijn graph (DBG)

CHALLENGES:

- Uneven sequencing coverage of organisms within the metagenome.
- The presence of conserved regions across different species.
- Repeat regions within genomes.
- The introduction of false k-mers by both closely related genomes at differing abundances and sequencing errors at high read coverage.
- Virome data is characterised by:
  - high proportions of repeat regions
  - hypervariable genomic regions associated with host interaction
  - high mutation rates which lead to increased metagenomic complexity and strain variation

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

DATASETS

- Simulated viromes dataset (572 genomes)
- Mock viral communities
- Human gut viromes spiked with a known exogenous bacteriophage
- Human virome data

AIMS

- Assembly efficacy and accuracy comparison
- Runtime and RAM usage
- Impact of sequencing coverage
- Genomic repeats

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

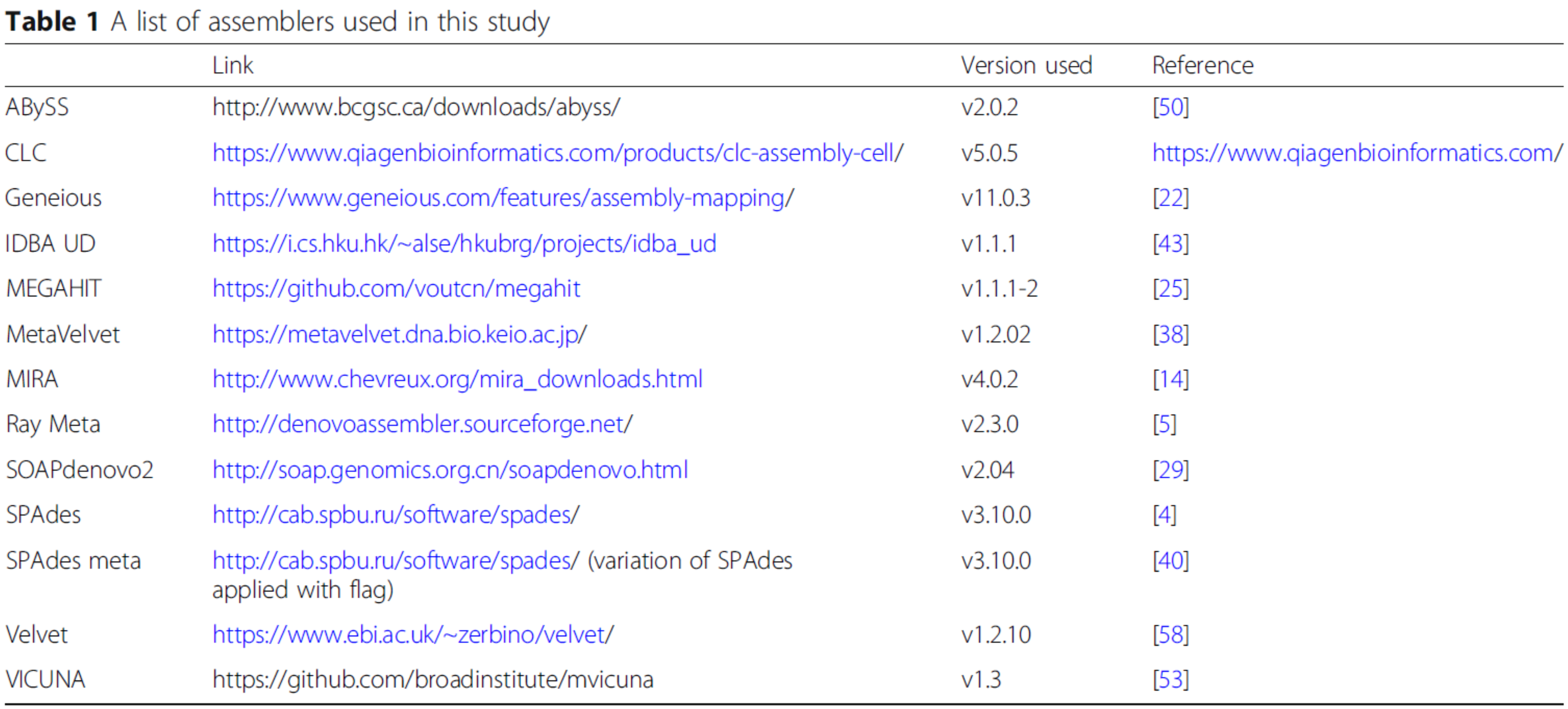


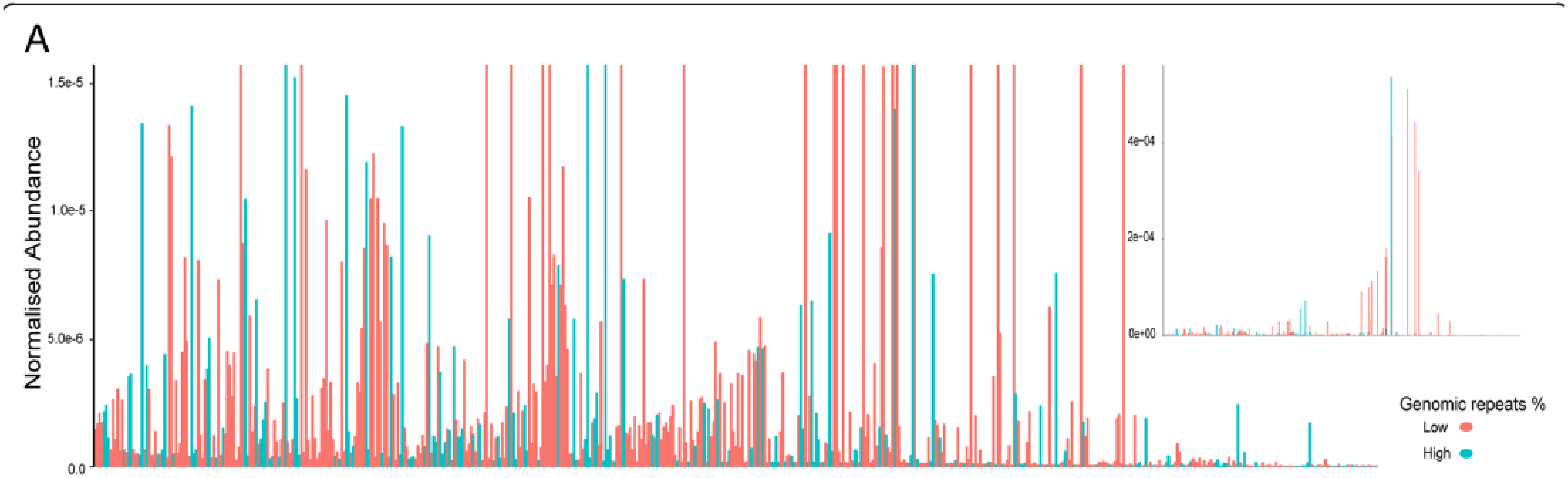
Table 1 A list of assemblers used in this study

	Link	Version used	Reference
ABySS	http://www.bcgsc.ca/downloads/abyss/	v2.0.2	[50]
CLC	https://www.qiagenbioinformatics.com/products/clc-assembly-cell/	v5.0.5	https://www.qiagenbioinformatics.com/
Geneious	https://www.geneious.com/features/assembly-mapping/	v11.0.3	[22]
IDBA UD	https://i.cs.hku.hk/~alse/hkubrg/projects/idba_ud	v1.1.1	[43]
MEGAHIT	https://github.com/voutcn/megahit	v1.1.1-2	[25]
MetaVelvet	https://metavelvet.dna.bio.keio.ac.jp/	v1.2.02	[38]
MIRA	http://www.chevreux.org/mira_downloads.html	v4.0.2	[14]
Ray Meta	http://denovoassembler.sourceforge.net/	v2.3.0	[5]
SOAPdenovo2	http://soap.genomics.org.cn/soapdenovo.html	v2.04	[29]
SPAdes	http://cab.spbu.ru/software/spades/	v3.10.0	[4]
SPAdes meta	http://cab.spbu.ru/software/spades/ (variation of SPAdes applied with flag)	v3.10.0	[40]
Velvet	https://www.ebi.ac.uk/~zerbino/velvet/	v1.2.10	[58]
VICUNA	https://github.com/broadinstitute/mvicuna	v1.3	[53]

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

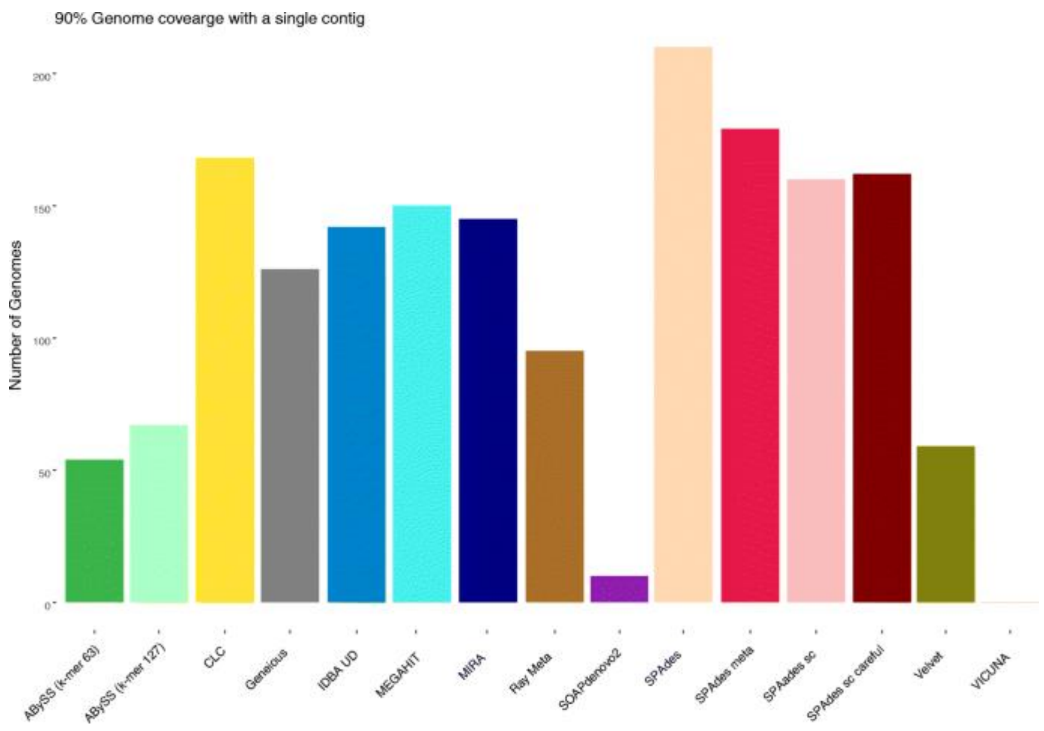
- Simulated viromes dataset (572 genomes)



Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- Simulated viromes dataset (572 genomes)



Number of contigs each assembler recovered to a minimum genome fraction of 90% in a single contig

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- Simulated viromes dataset (572 genomes)

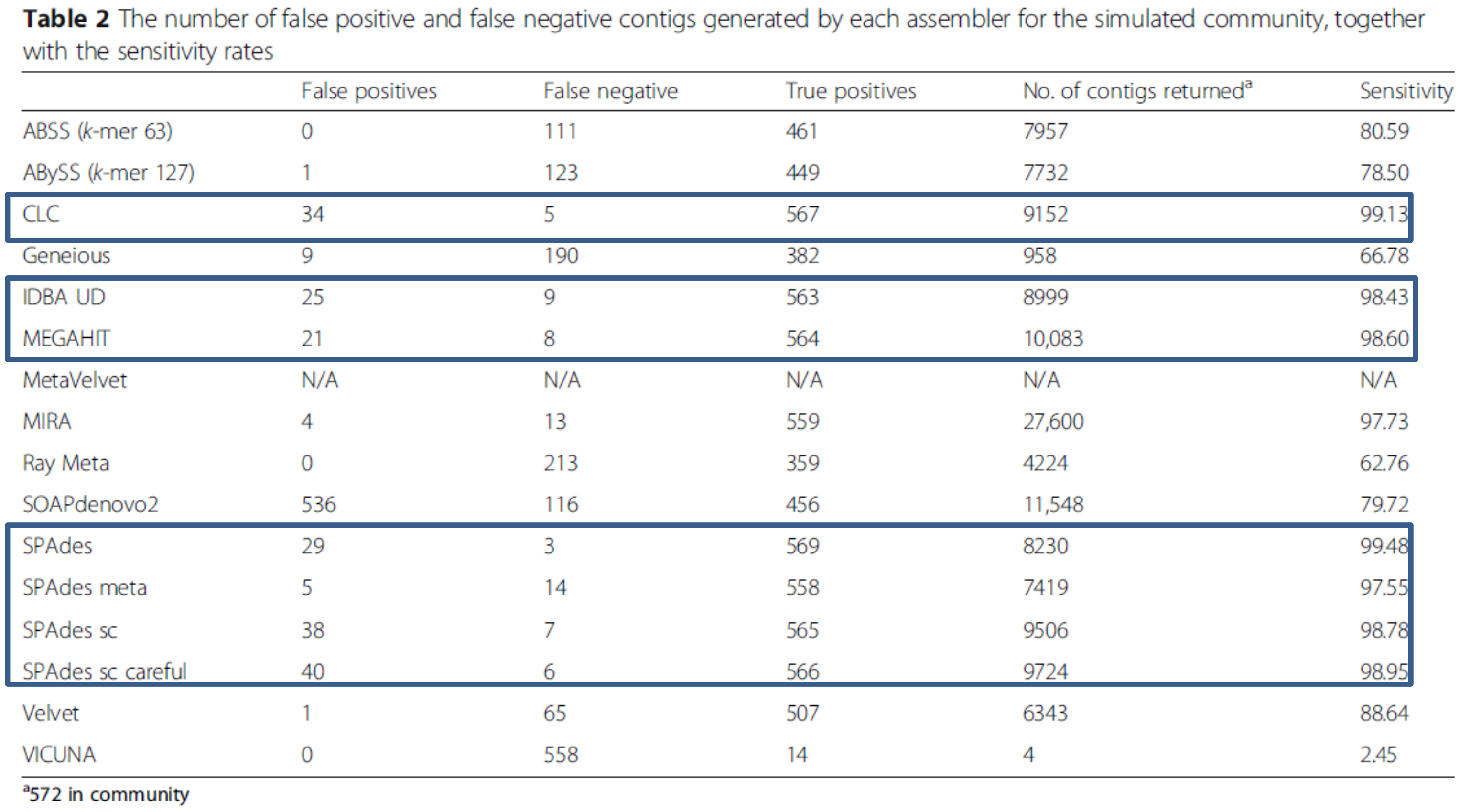


Table 2 The number of false positive and false negative contigs generated by each assembler for the simulated community, together with the sensitivity rates

	False positives	False negative	True positives	No. of contigs returned[a]	Sensitivity
ABSS (*k*-mer 63)	0	111	461	7957	80.59
ABySS (*k*-mer 127)	1	123	449	7732	78.50
CLC	34	5	567	9152	99.13
Geneious	9	190	382	958	66.78
IDBA UD	25	9	563	8999	98.43
MEGAHIT	21	8	564	10,083	98.60
MetaVelvet	N/A	N/A	N/A	N/A	N/A
MIRA	4	13	559	27,600	97.73
Ray Meta	0	213	359	4224	62.76
SOAPdenovo2	536	116	456	11,548	79.72
SPAdes	29	3	569	8230	99.48
SPAdes meta	5	14	558	7419	97.55
SPAdes sc	38	7	565	9506	98.78
SPAdes sc careful	40	6	566	9724	98.95
Velvet	1	65	507	6343	88.64
VICUNA	0	558	14	4	2.45

[a] 572 in community

false positive: no alignment to reference genomes

false negative: recovered genome fraction of 0%

Sensitivity: true positive / (false positive + false negative)

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- Mock community dataset - were used to investigate the impact **of high and low abundance ssDNA viruses on assembly performance**

Mock A (Table 3a) contained 12 viral genomes, 10 of which were at equal abundance (9.82% of the total community) and 2 ssDNA genomes (NC_001330 and NC_001422) at low abundance (0.92%).

Mock B (Table 3b) contained 12 viral genomes, 10 of which were at equal abundance (9.82% of the total community) and 2 ssDNA genomes (NC_001330 and NC_001422) at higher abundance (32.47%).

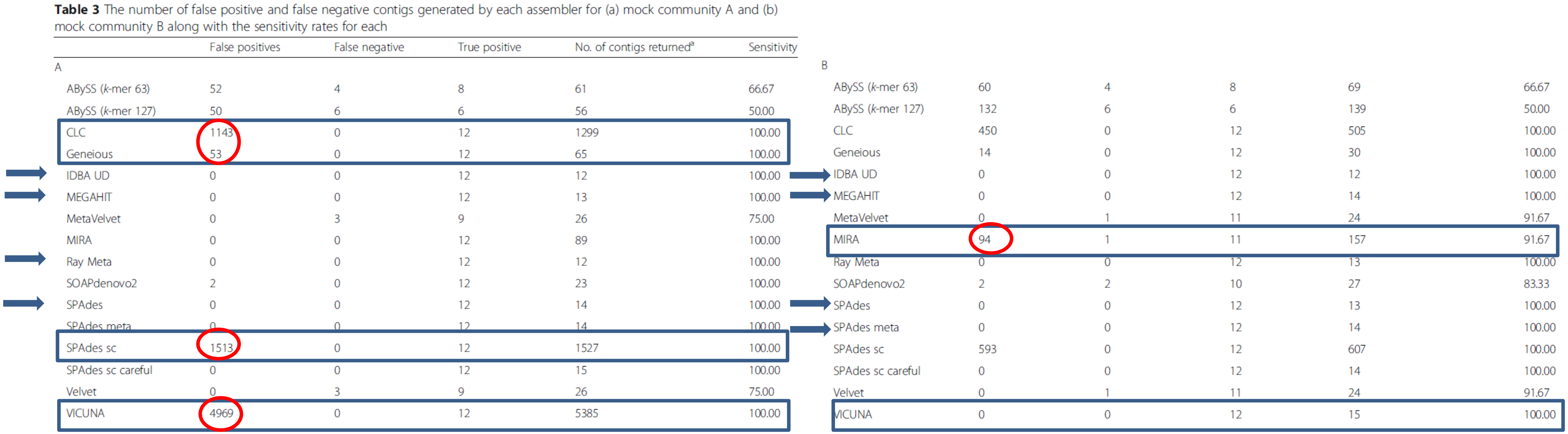


Table 3 The number of false positive and false negative contigs generated by each assembler for (a) mock community A and (b) mock community B along with the sensitivity rates for each

	False positives	False negative	True positive	No. of contigs returned[a]	Sensitivity
A					
ABySS (*k*-mer 63)	52	4	8	61	66.67
ABySS (*k*-mer 127)	50	6	6	56	50.00
CLC	1143	0	12	1299	100.00
Geneious	53	0	12	65	100.00
IDBA UD	0	0	12	12	100.00
MEGAHIT	0	0	12	13	100.00
MetaVelvet	0	3	9	26	75.00
MIRA	0	0	12	89	100.00
Ray Meta	0	0	12	12	100.00
SOAPdenovo2	2	0	12	23	100.00
SPAdes	0	0	12	14	100.00
SPAdes meta	0	0	12	14	100.00
SPAdes sc	1513	0	12	1527	100.00
SPAdes sc careful	0	0	12	15	100.00
Velvet	0	3	9	26	75.00
VICUNA	4969	0	12	5385	100.00
B					
ABySS (*k*-mer 63)	60	4	8	69	66.67
ABySS (*k*-mer 127)	132	6	6	139	50.00
CLC	450	0	12	505	100.00
Geneious	14	0	12	30	100.00
IDBA UD	0	0	12	12	100.00
MEGAHIT	0	0	12	14	100.00
MetaVelvet	0	1	11	24	91.67
MIRA	94	1	11	157	91.67
Ray Meta	0	0	12	13	100.00
SOAPdenovo2	2	2	10	27	83.33
SPAdes	0	0	12	13	100.00
SPAdes meta	0	0	12	14	100.00
SPAdes sc	593	0	12	607	100.00
SPAdes sc careful	0	0	12	14	100.00
Velvet	0	1	11	24	91.67
VICUNA	0	0	12	15	100.00

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- Human gut viromes spiked with a known exogenous bacteriophage

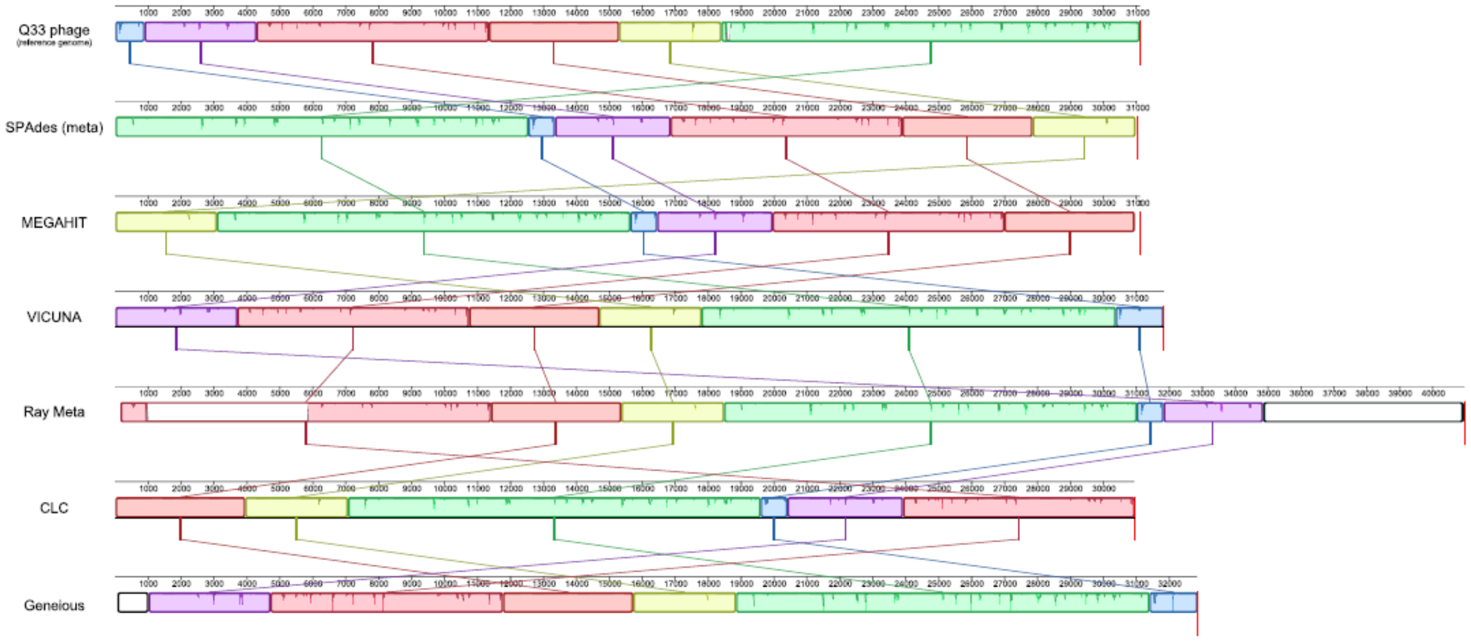
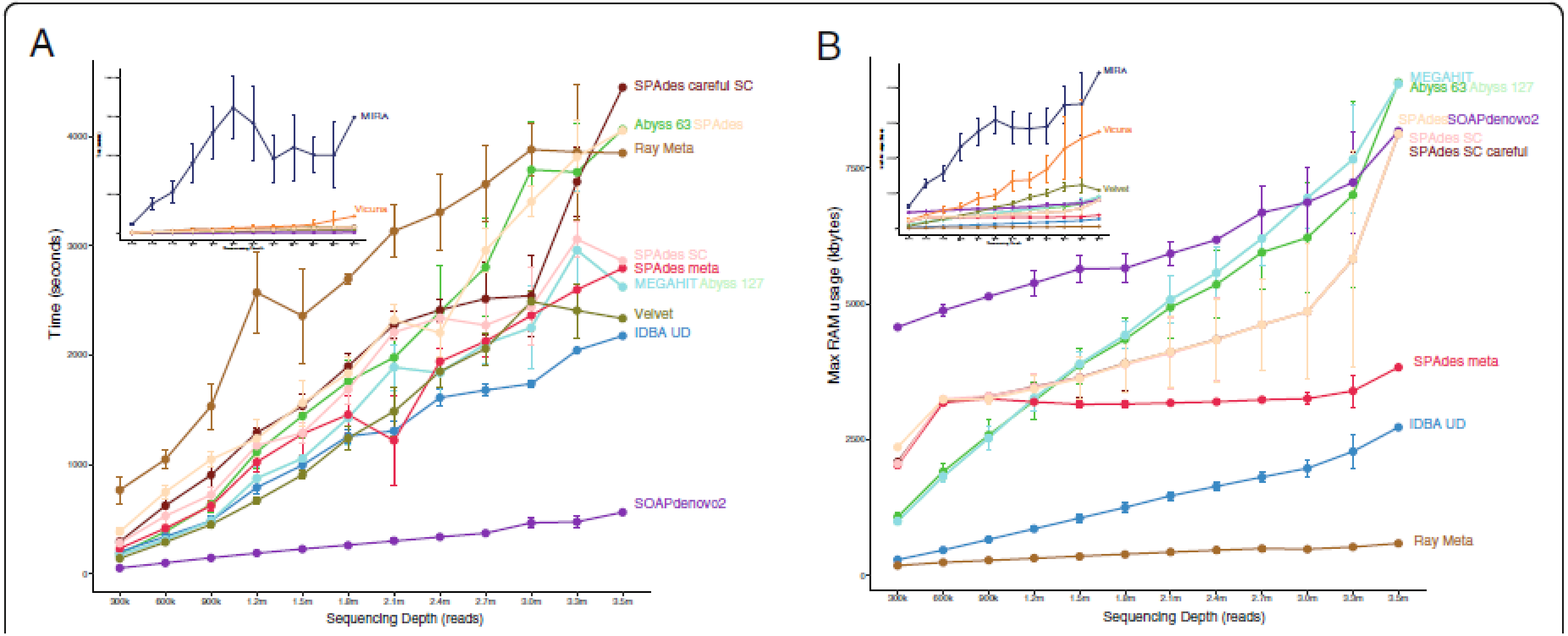


Fig. 3 Mauve output of the Q33 reference genome (top) along with of the six assemblers which recovered > 99% of the genome with a single contig. Assembly regions outside of locally collinear blocks which do not share homology to the reference genome are highlighted by a black outline. Reverse complement of assemblies in the opposite orientation to the reference were plotted for visualisation purposes (VICUNA, CLC, Geneious)

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- healthy human gut viromes and various sequencing depths



Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- healthy human gut viromes and various sequencing depths

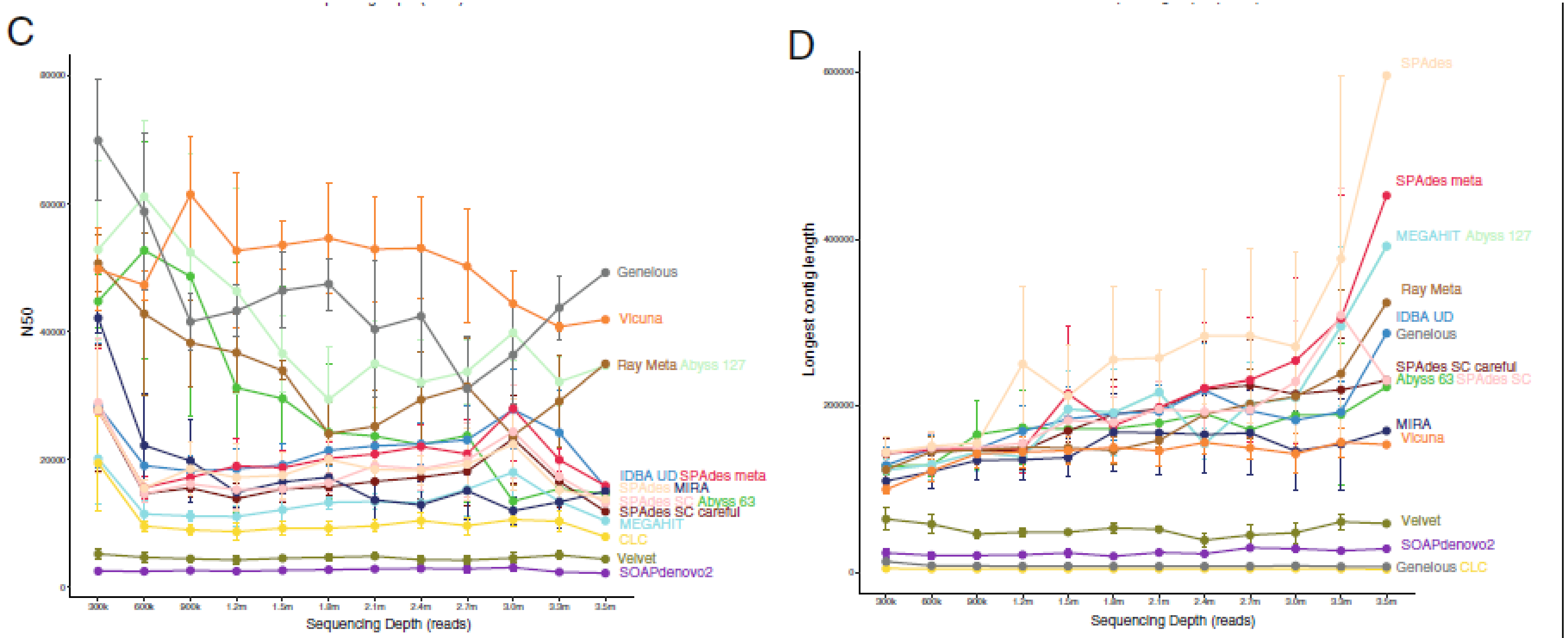


Fig. 4 a Time, measured in seconds, for each assembly to reach completion successfully for each read subset. **b** The maximum RAM, measured in MB, used for each assembly for each read subset. **c** Mean N50 length and **d** mean contig length for four samples for each assembly across the read subsets after filtering contigs less than 1000 bases. Points represent the mean time for the four samples while error bars are the standard error

Choice of assembly software has a critical impact on virome characterisation

Sutton et al. Microbiome (2019) 7:12
https://doi.org/10.1186/s40168-019-0626-5

- **CONCLUSIONS**

❖ Assemblers were assessed using four independent virome datasets, namely, simulated reads, two mock communities, viromes spiked with a known phage and human gut viromes.

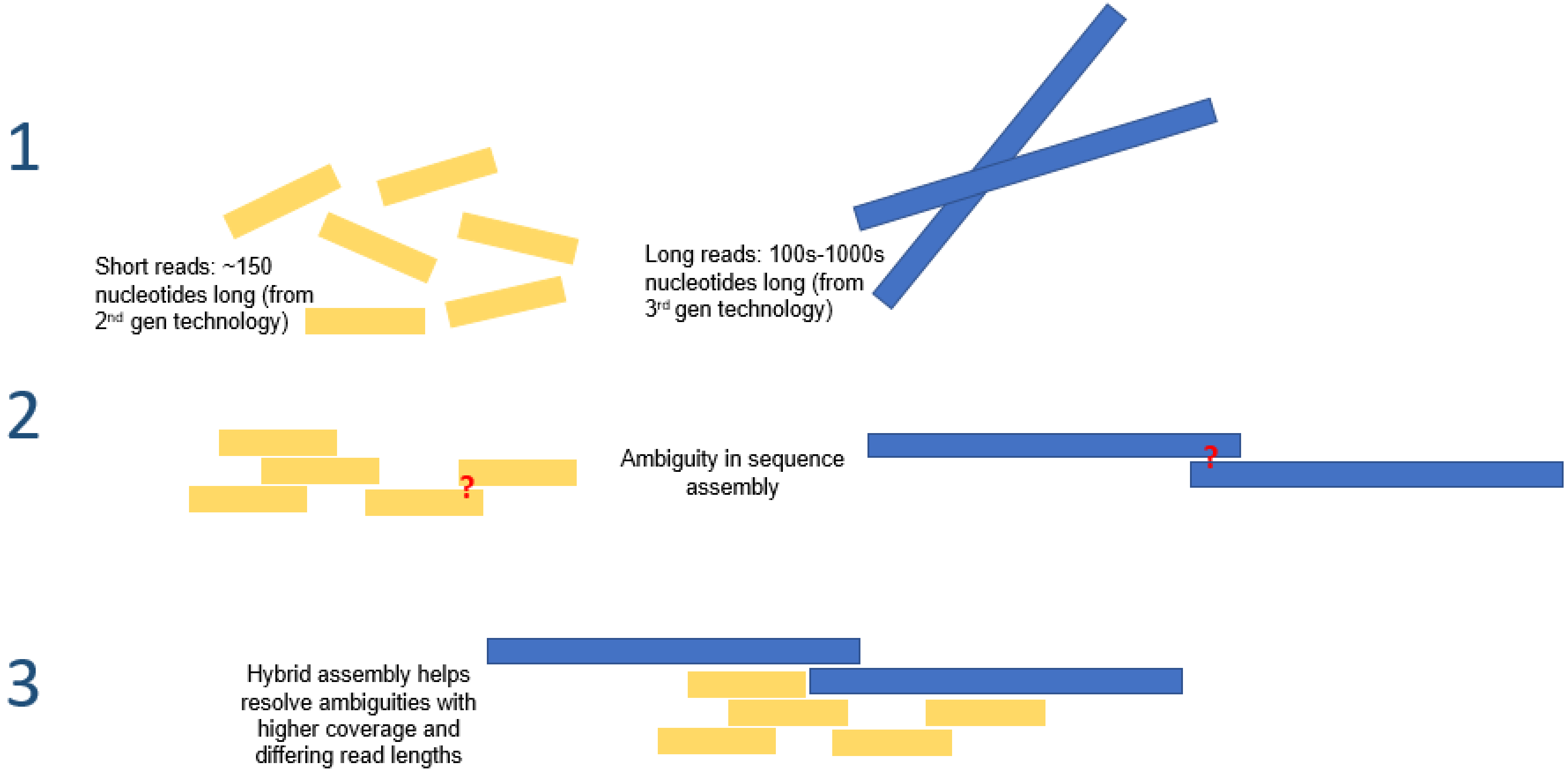
❖ Assembly performance varied significantly across all test datasets, with SPAdes (meta) performing consistently well.

❖ It was also found that while some assemblers addressed the challenges of virome data better than others, all assemblers had limitations

❖ Low read coverage and genomic repeats resulted in assemblies with poor genome recovery, high degrees of fragmentation and low-accuracy contigs across all assemblers.

Hybrid genome assembly – short and long reads

https://en.wikipedia.org/wiki/Hybrid_genome_assembly



1

Short reads: ~150 nucleotides long (from 2nd gen technology)

Long reads: 100s-1000s nucleotides long (from 3rd gen technology)

2

Ambiguity in sequence assembly

3

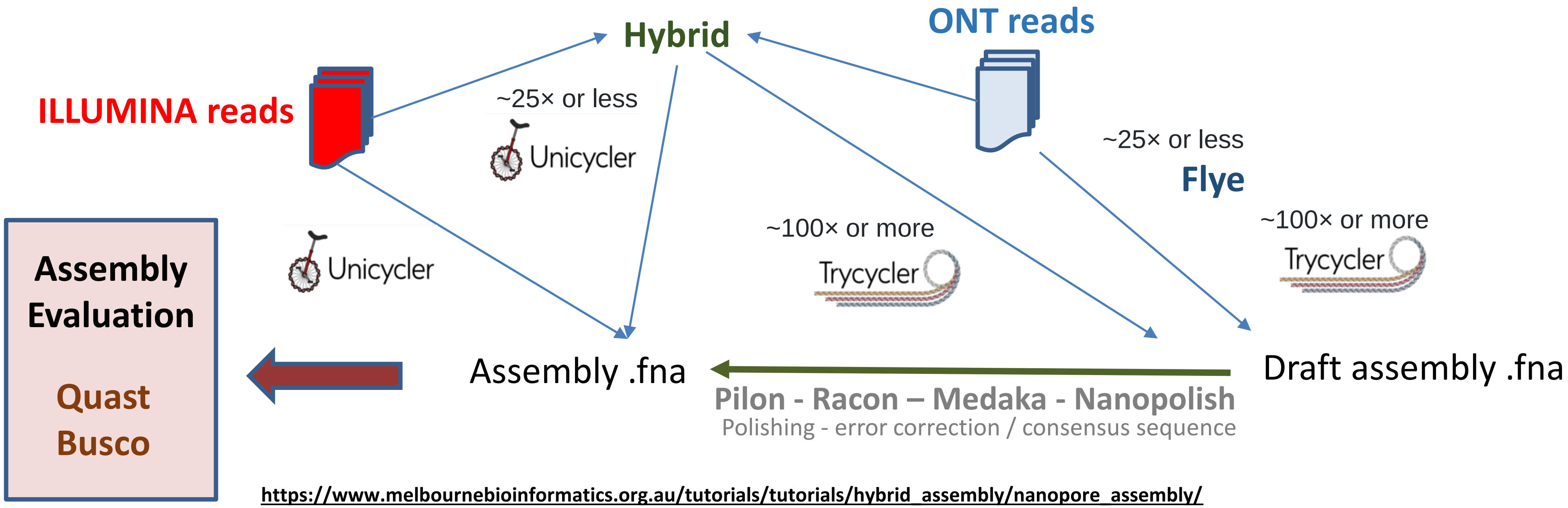
Hybrid assembly helps resolve ambiguities with higher coverage and differing read lengths

Hybrid genome assembly - nanopore and illumina

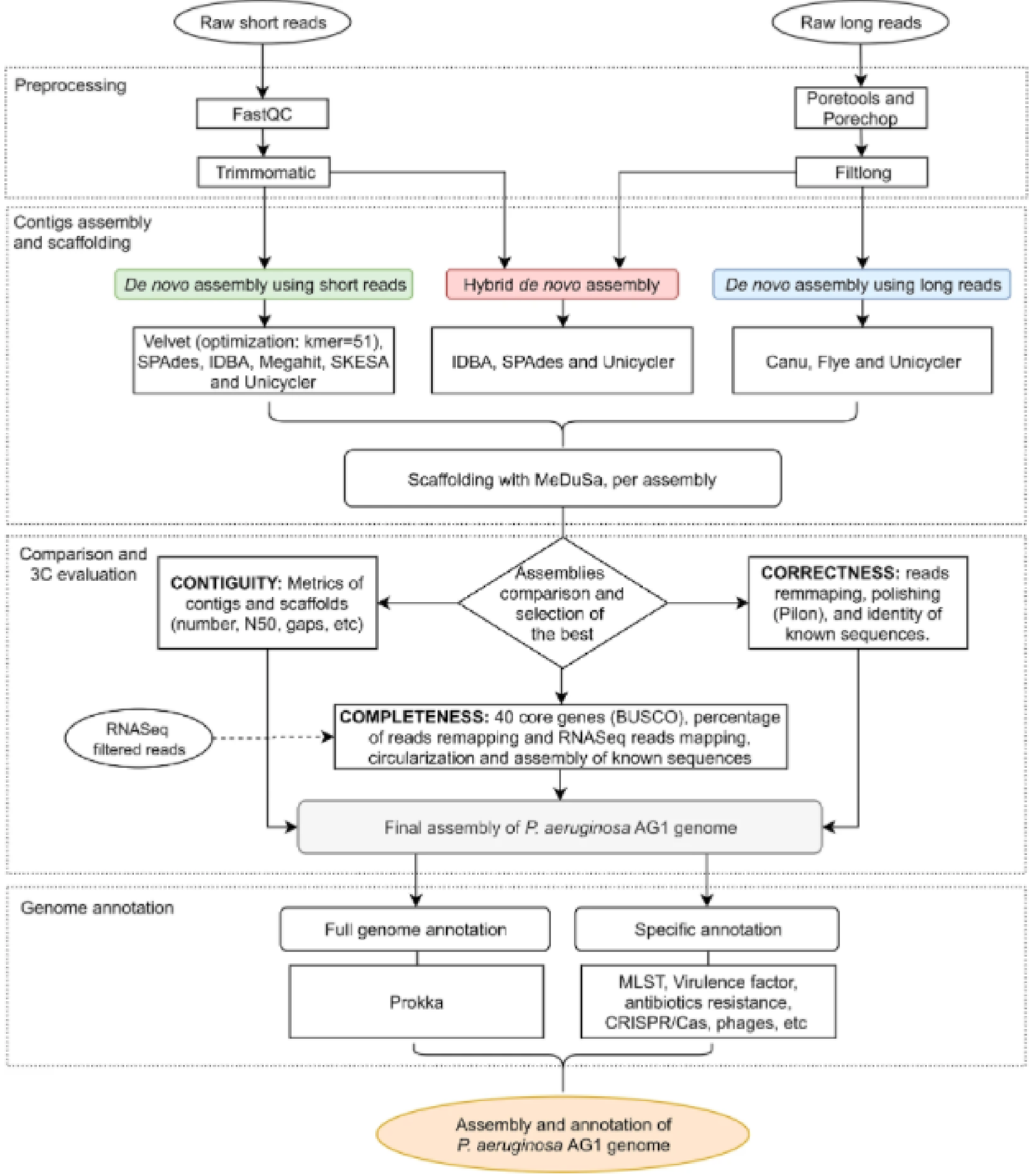
Short reads (ILLUMINA) + Long reads (ONT) → deNovo assembly (De novo assembly is the process of assembling a genome from scratch using only the sequenced reads as input - no reference genome is used.) → **high-quality assembly**

ONT: >40.000b, higher error rate – **genome structure**
ILLUMINA: 300b, lower error rate – **high base-level accuracy**

Higher COST



https://www.melbournebioinformatics.org.au/tutorials/tutorials/hybrid_assembly/nanopore_assembly/
https://github.com/rrwick/Unicycler
https://denbi-nanopore-training-course.readthedocs.io/en/latest/index.html



Raw short reads

Raw long reads

Preprocessing

FastQC

Trimmomatic

Poretools and Porechop

Filtlong

Contigs assembly and scaffolding

De novo assembly using short reads

Hybrid *de novo* assembly

De novo assembly using long reads

Velvet (optimization: kmer=51), SPAdes, IDBA, Megahit, SKESA and Unicycler

IDBA, SPAdes and Unicycler

Canu, Flye and Unicycler

Scaffolding with MeDuSa, per assembly

Comparison and 3C evaluation

CONTIGUITY: Metrics of contigs and scaffolds (number, N50, gaps, etc)

Assemblies comparison and selection of the best

CORRECTNESS: reads remmaping, polishing (Pilon), and identity of known sequences.

RNASeq filtered reads

COMPLETENESS: 40 core genes (BUSCO), percentage of reads remapping and RNASeq reads mapping, circularization and assembly of known sequences

Final assembly of *P. aeruginosa* AG1 genome

Genome annotation

Full genome annotation

Specific annotation

Prokka

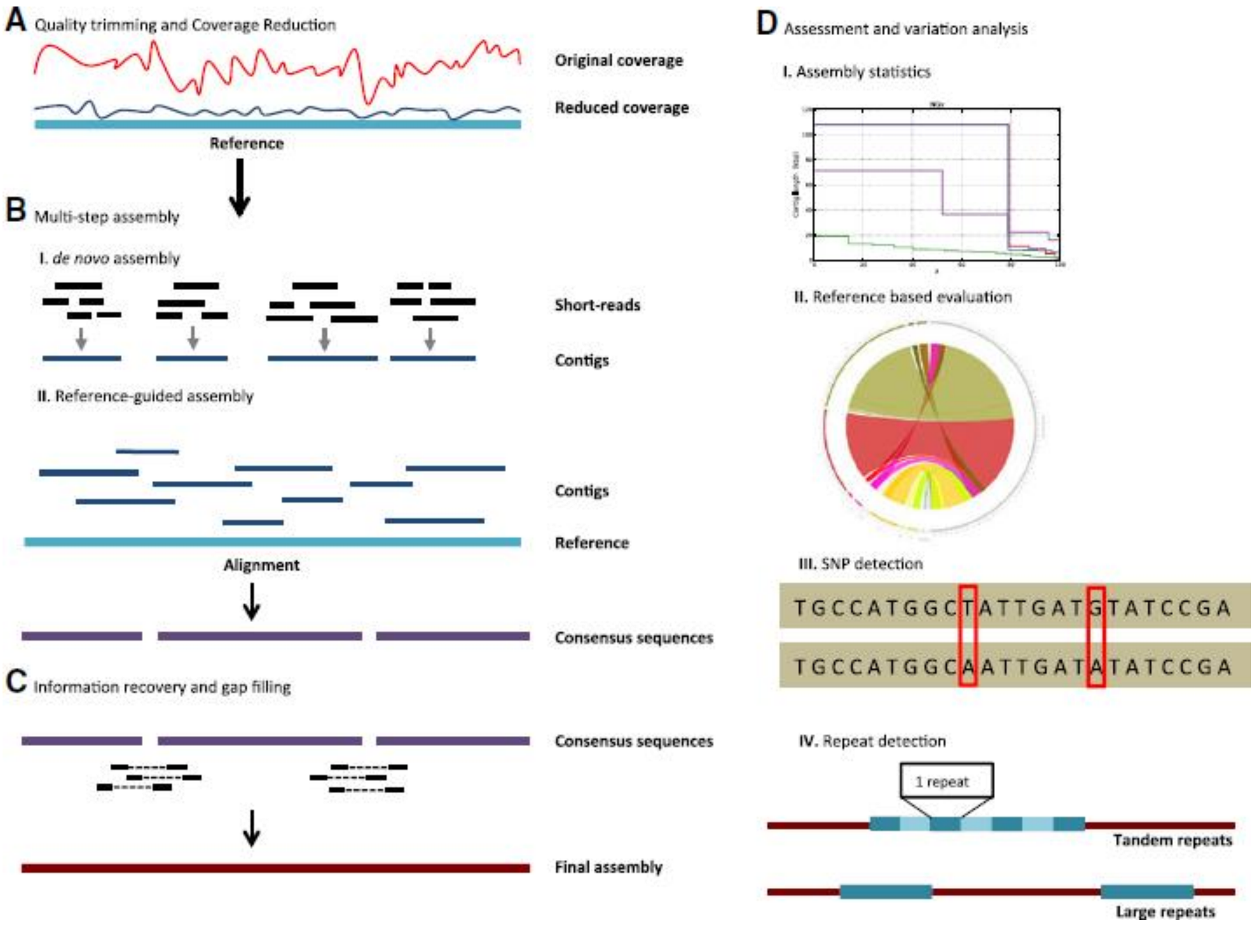
MLST, Virulence factor, antibiotics resistance, CRISPR/Cas, phages, etc

Assembly and annotation of *P. aeruginosa* AG1 genome

General bioinformatic pipeline to assemble, compare and annotate the *Pseudomonas aeruginosa* AG1 genome using short and long reads as well as hybrid approaches.

Molina-Mora et al., Scientific Reports 2020

VirAmp: a galaxy-based viral genome assembly pipeline



Gigascience, Volume 4, Issue 1, December 2015, s13742-015-0060-y, https://doi.org/10.1186/s13742-015-0060-y

The content of this slide may be subject to copyright: please see the slide notes for details.

Thanks for your attention!

Questions ?