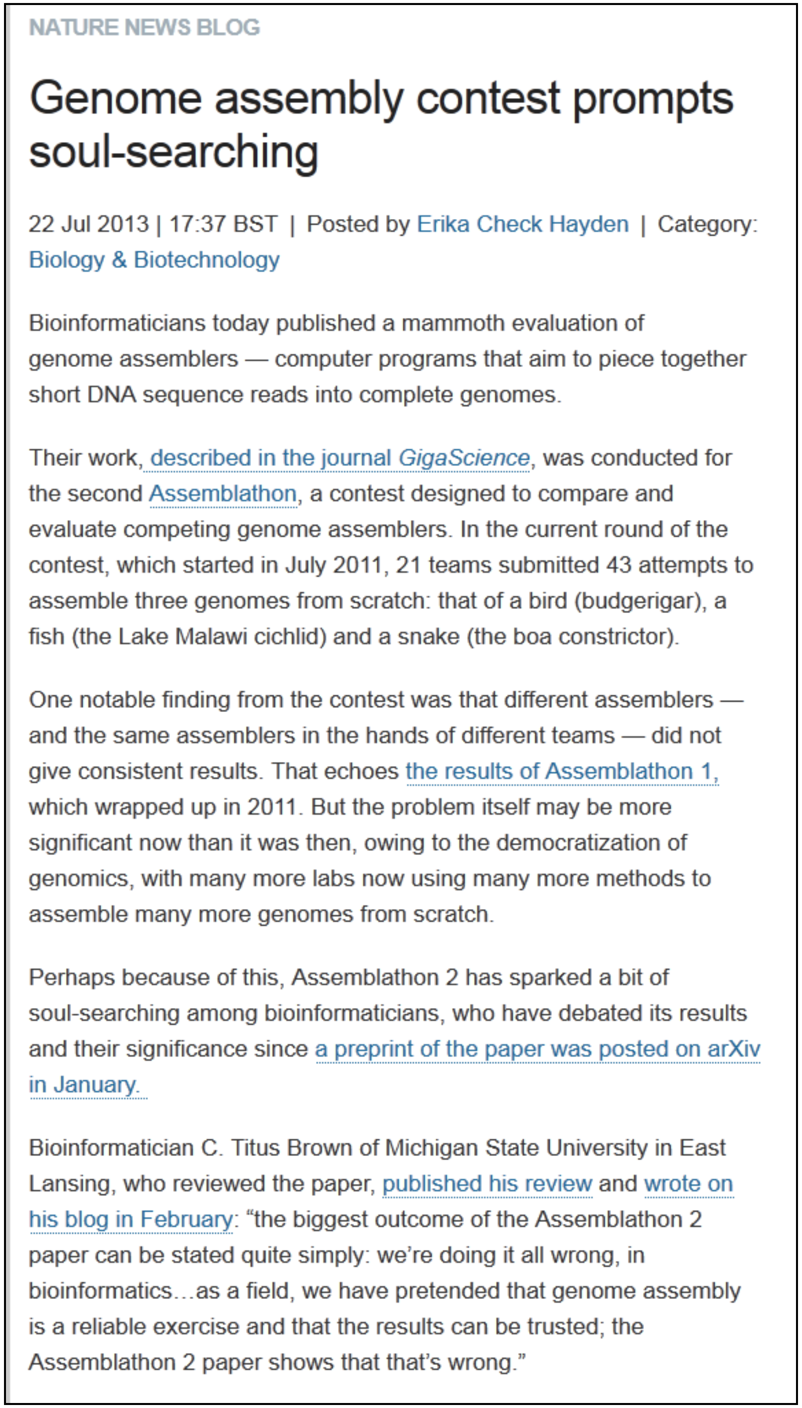


NATURE NEWS BLOG

Genome assembly contest prompts soul-searching

22 Jul 2013 | 17:37 BST | Posted by Erika Check Hayden | Category: Biology & Biotechnology

Bioinformaticians today published a mammoth evaluation of genome assemblers — computer programs that aim to piece together short DNA sequence reads into complete genomes.

Their work, described in the journal *GigaScience*, was conducted for the second Assemblathon, a contest designed to compare and evaluate competing genome assemblers. In the current round of the contest, which started in July 2011, 21 teams submitted 43 attempts to assemble three genomes from scratch: that of a bird (budgerigar), a fish (the Lake Malawi cichlid) and a snake (the boa constrictor).

One notable finding from the contest was that different assemblers — and the same assemblers in the hands of different teams — did not give consistent results. That echoes the results of Assemblathon 1, which wrapped up in 2011. But the problem itself may be more significant now than it was then, owing to the democratization of genomics, with many more labs now using many more methods to assemble many more genomes from scratch.

Perhaps because of this, Assemblathon 2 has sparked a bit of soul-searching among bioinformaticians, who have debated its results and their significance since a preprint of the paper was posted on arXiv in January.

Bioinformatician C. Titus Brown of Michigan State University in East Lansing, who reviewed the paper, published his review and wrote on his blog in February: “the biggest outcome of the Assemblathon 2 paper can be stated quite simply: we’re doing it all wrong, in bioinformatics…as a field, we have pretended that genome assembly is a reliable exercise and that the results can be trusted; the Assemblathon 2 paper shows that that’s wrong.”

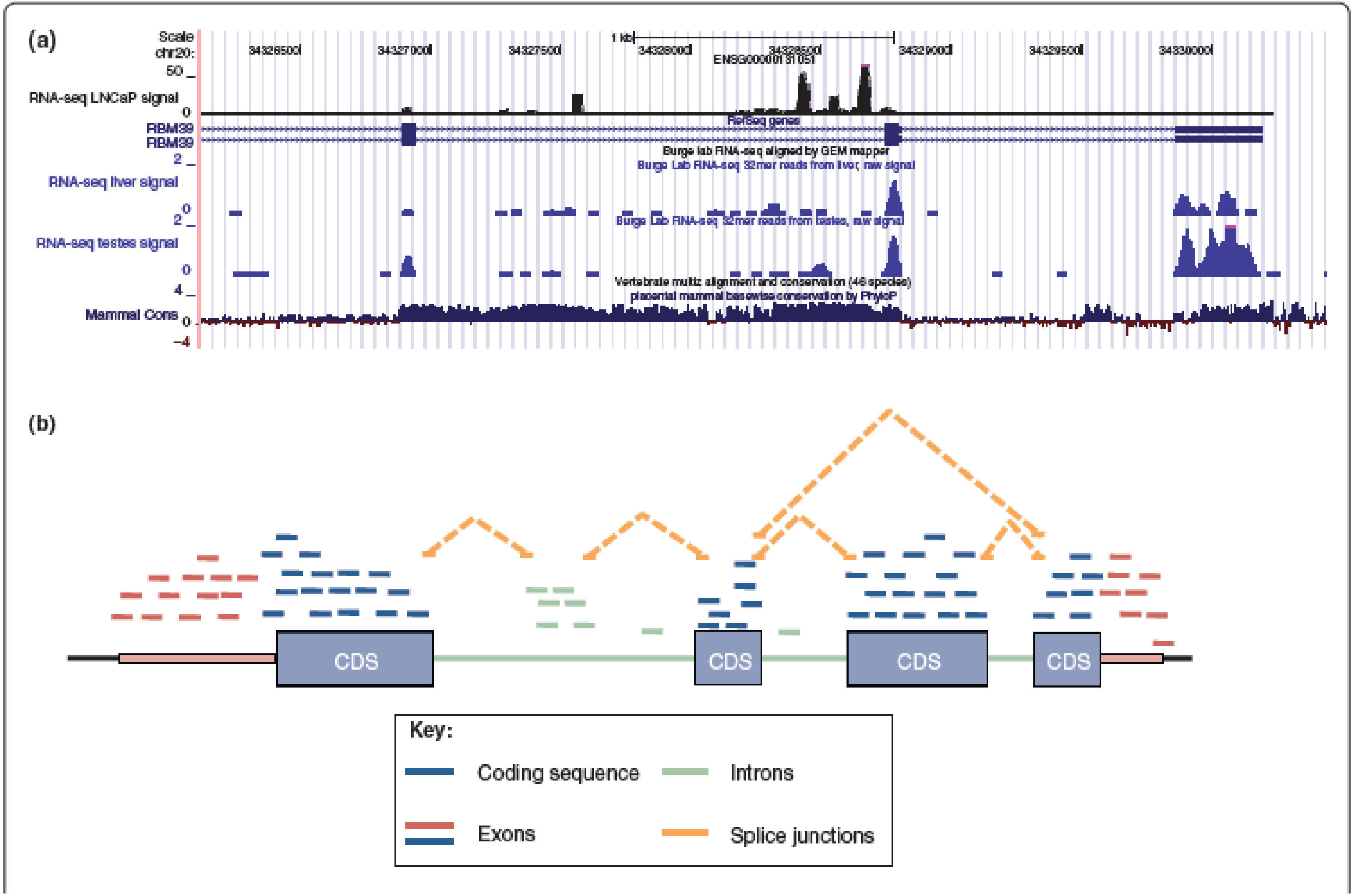
3 basic approaches:

read mapping

reference guided assembly

de novo assembly

Read Mapping RNA-Seq

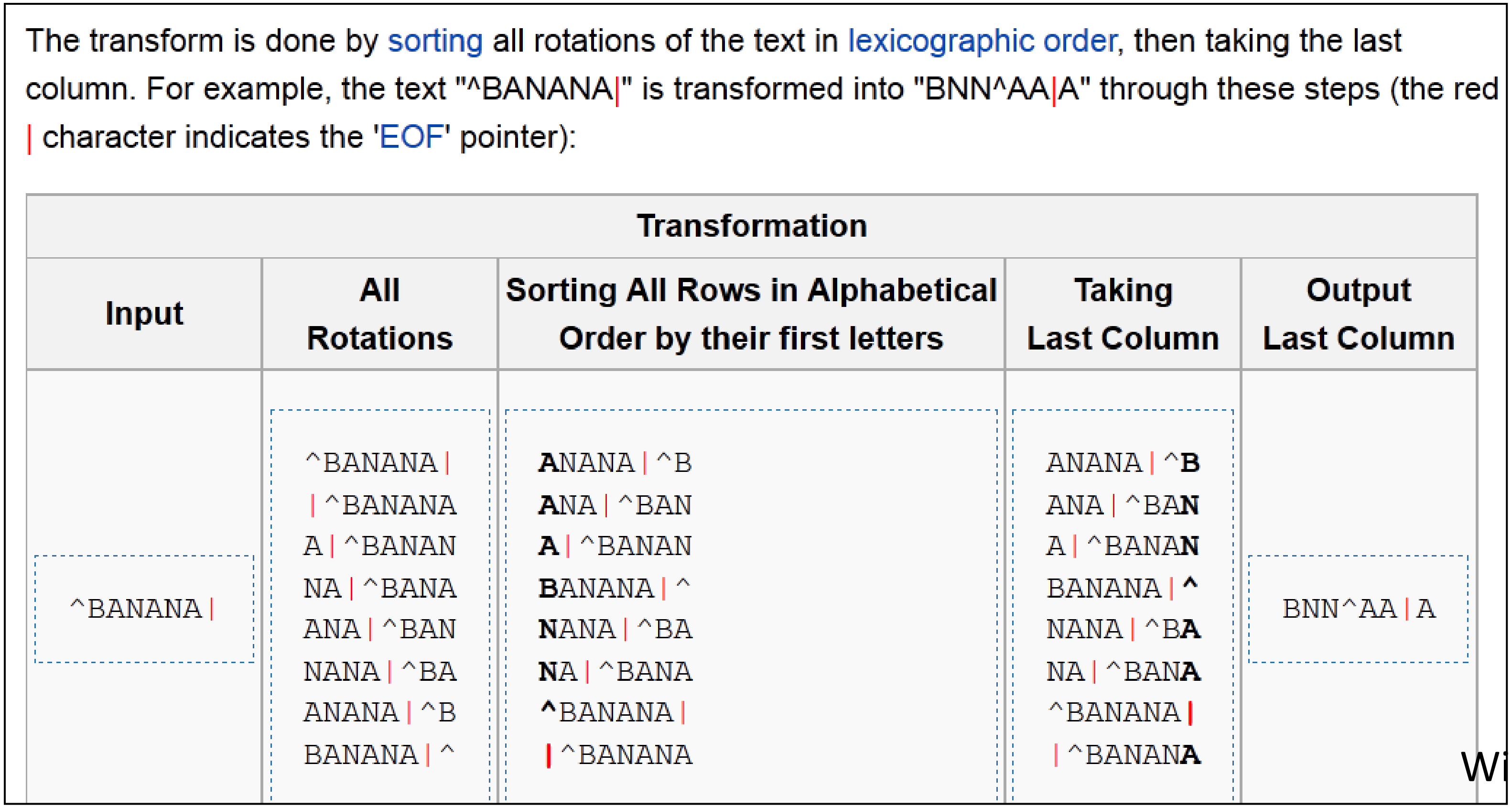


Oshlack et al. 2010. From RNA-seq reads to differential expression results . Genome Biology 11:220.

Read Mapping with the Burrows-Wheeler transform

examples

Bowtie2 (Langmead & Salzberg 2012)
BWA (Li & Durbin 2009)
SOAPAligner (Li et al 2009)



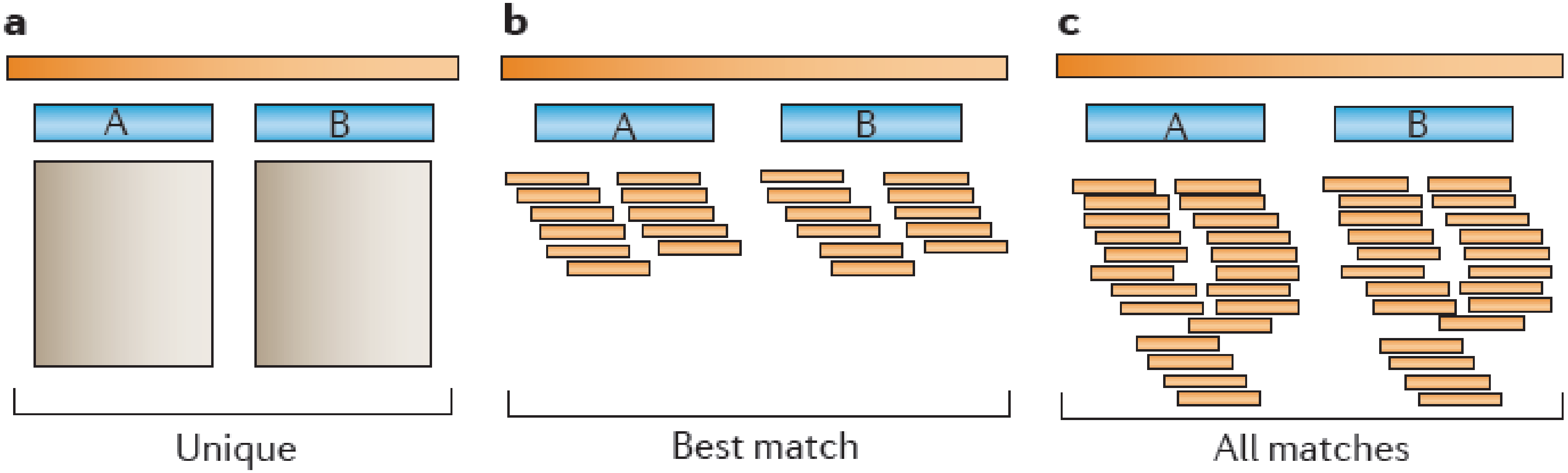
The transform is done by sorting all rotations of the text in lexicographic order, then taking the last column. For example, the text "^BANANA|" is transformed into "BNN^AA|A" through these steps (the red | character indicates the 'EOF' pointer):

Transformation				
Input	**All Rotations**	**Sorting All Rows in Alphabetical Order by their first letters**	**Taking Last Column**	**Output Last Column**
^BANANA|	^BANANA|	ANANA|^B	ANANA|^B	BNN^AA|A
	|^BANANA	ANA|^BAN	ANA|^BAN	
	A|^BANAN	A|^BANAN	A|^BANAN	
	NA|^BANA	BANANA|^	BANANA|^	
	ANA|^BAN	NANA|^BA	NANA|^BA	
	NANA|^BA	NA|^BANA	NA|^BANA	
	ANANA|^B	^BANANA|	^BANANA|	
	BANANA|^	|^BANANA	|^BANANA	

Wikipedia

A very efficient data compression method applied to the reference genome
Relies on a reversible sort, that functions as an index to sequences in the genome
Performance improves with larger data sets.

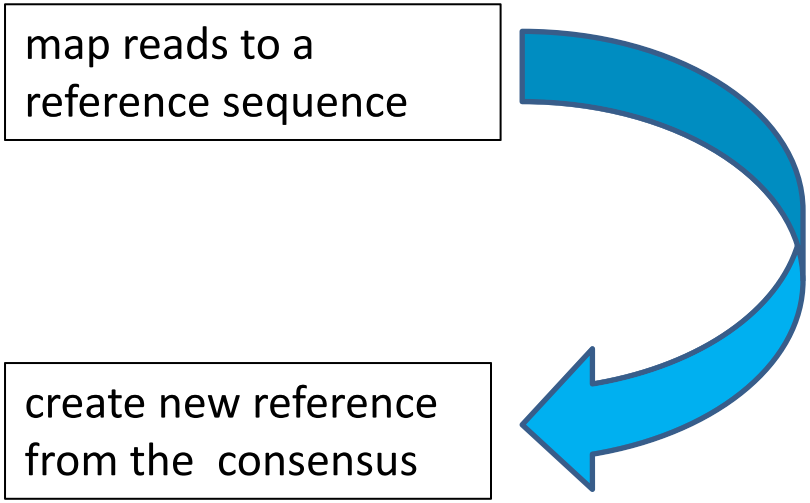
Multi-Reads Map to Identical or Similar Repeats



a. only report unique matches (read ignored)
b. randomly distribute repeat matches (1 per read)
c. report all repeat matches (many per read)

Treangen & Salzberg 2012. Repetitive DNA and next-generation sequencing: computational challenges and solutions Nature Review Genetics 13:36-46

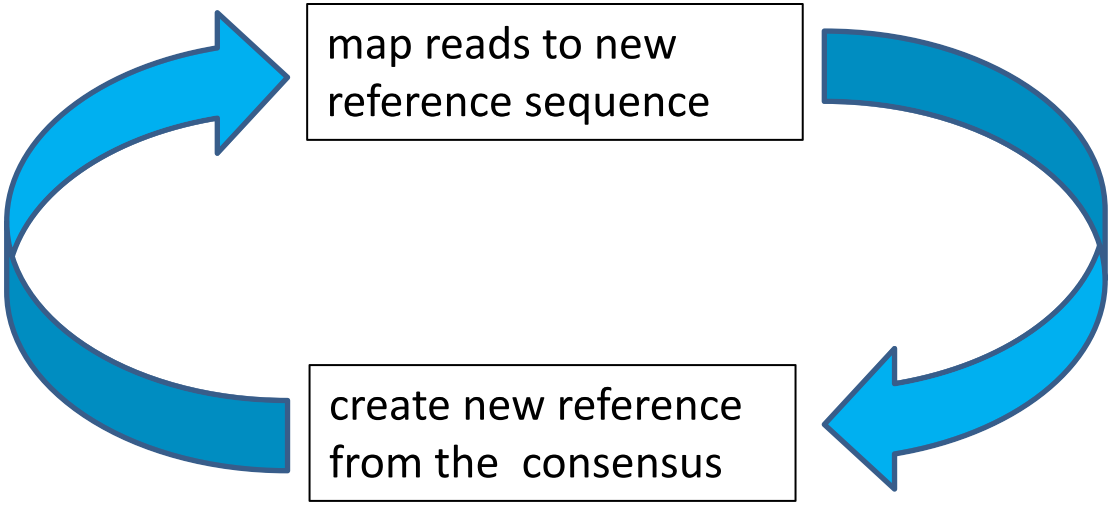
Reference Guided Assembly



map reads to a reference sequence

create new reference from the consensus

Reference Guided Assembly



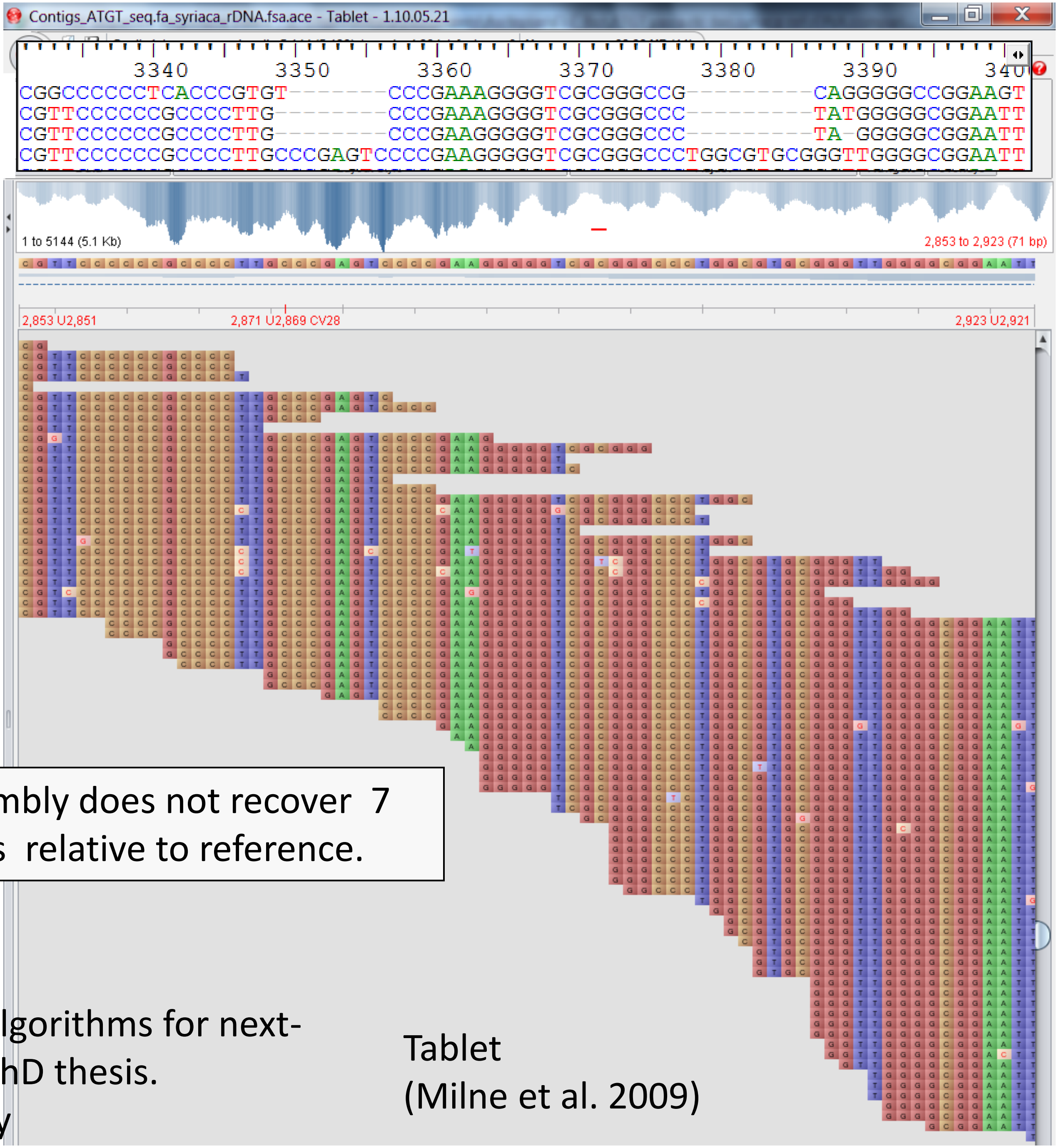
map reads to new reference sequence

create new reference from the consensus

Repeat until no further improvement

YASRA

yet another short read aligner



Contigs_ATGT_seq.fa_syriaca_rDNA.fsa.ace - Tablet - 1.10.05.21

```
                  3340      3350      3360      3370      3380      3390      340
reference      CGGCCCCCTCACCCGTGT-------CCCGAAAGGGGTCGCGGCCG---------CAGGGGGCCGGAAGT
other aligners CGTTCCCCCCGCCCCTTG-------CCCGAAAGGGGTCGCGGCCC---------TATGGGGGCGGAATT
               CGTTCCCCCCGCCCCTTG-------CCCGAAGGGGGTCGCGGCCC---------TA-GGGGGCGGAATT
YASRA          CGTTCCCCCCGCCCCTTGCCCGAGTCCCCGAAGGGGGTCGCGGCCCTGGCGTGCGGGTTGGGGCGGAATT
```

Non-iterative assembly does not recover 7 and 9 bp insertions relative to reference.

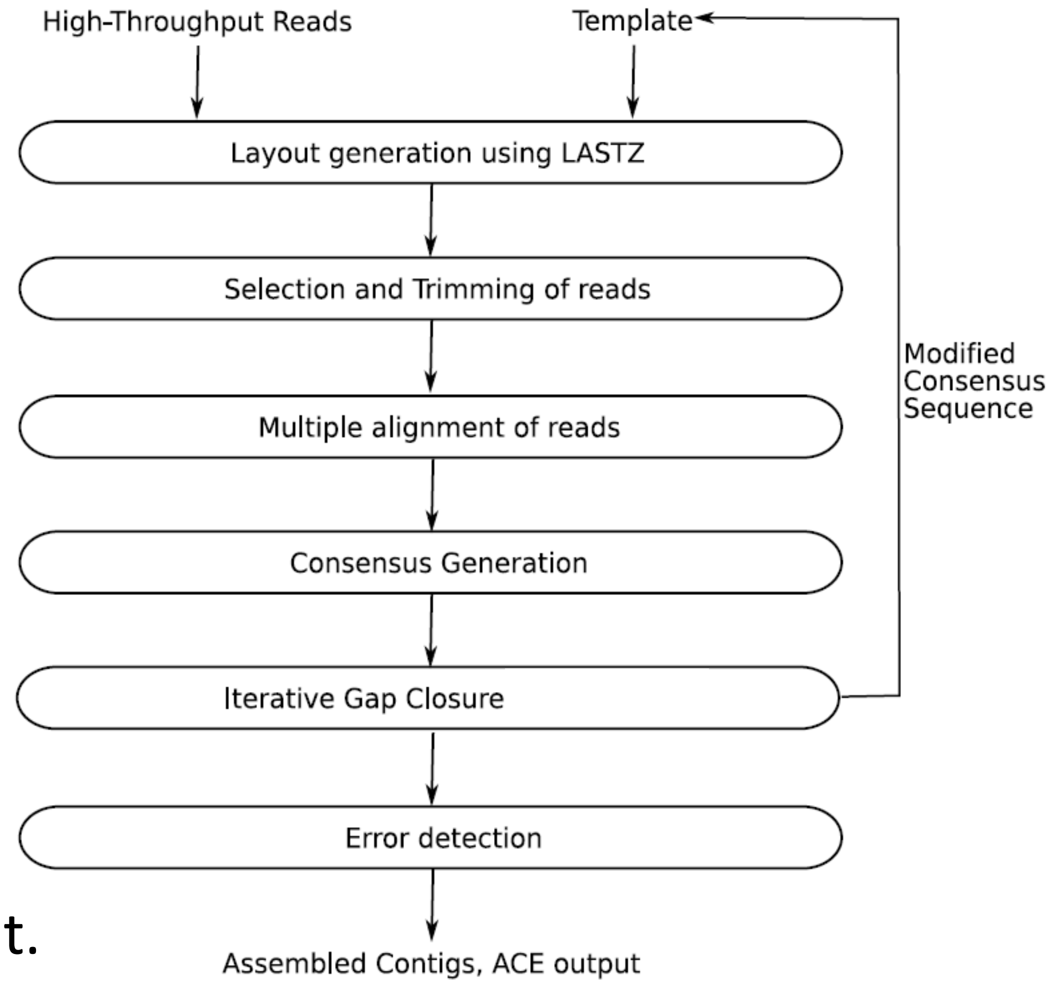
Ratan, A. (2009). Assembly algorithms for next-generation sequence data. PhD thesis. Pennsylvania State University

Tablet (Milne et al. 2009)

Reference Guided Assembly

YASRA (Ratan, 2009)

1. Reference can be 80-90% divergent.
2. Map reads to reference followed by de novo assembly of unmapped reads.
3. Closes gaps with overlap-layout consensus.
4. Creates a new reference.
5. Repeats the process until no additional improvement.

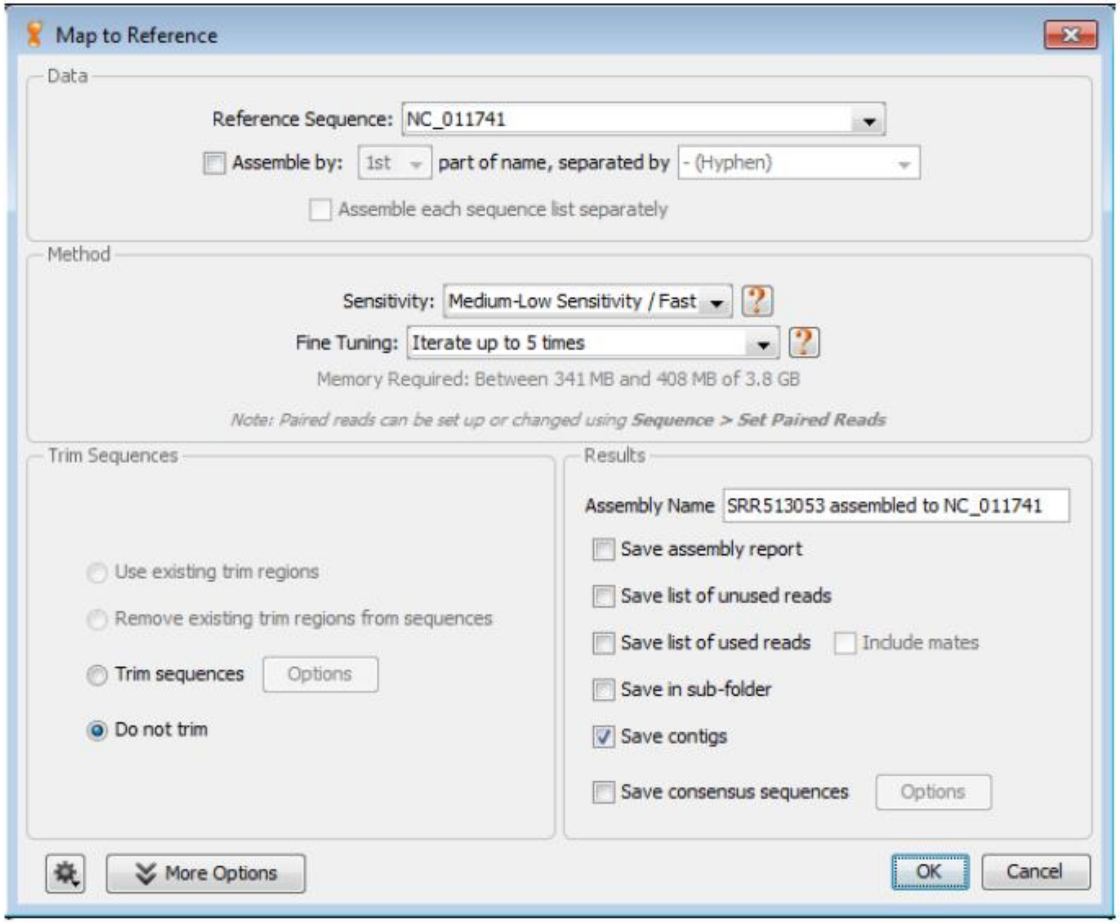

High-Throughput Reads
Template
Layout generation using LASTZ
Selection and Trimming of reads
Multiple alignment of reads
Consensus Generation
Iterative Gap Closure
Error detection
Assembled Contigs, ACE output
Modified Consensus Sequence

Reference Guided Assembly

The Geneious 6.0.3 Read Mapper

Authors

Developer: Matthew Kearse
Authors: Matthew Kearse, Shane Sturrock, Peter Meintjes



Map to Reference

Data

Reference Sequence: NC_011741

□ Assemble by: 1st part of name, separated by - (Hyphen)

□ Assemble each sequence list separately

Method

Sensitivity: Medium-Low Sensitivity / Fast

Fine Tuning: Iterate up to 5 times

Memory Required: Between 341 MB and 408 MB of 3.8 GB

Note: Paired reads can be set up or changed using ***Sequence > Set Paired Reads***

Trim Sequences

○ Use existing trim regions

○ Remove existing trim regions from sequences

○ Trim sequences Options

● Do not trim

Results

Assembly Name SRR513053 assembled to NC_011741

□ Save assembly report

□ Save list of unused reads

□ Save list of used reads □ Include mates

□ Save in sub-folder

☑ Save contigs

□ Save consensus sequences Options

Reference Guided Assembly

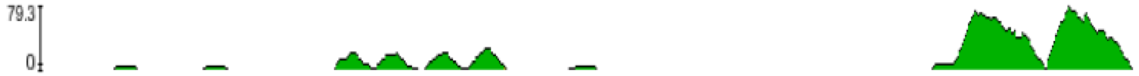
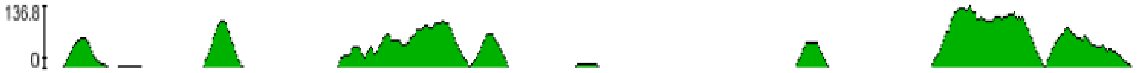
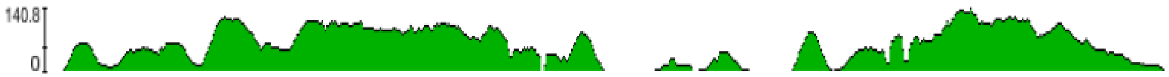
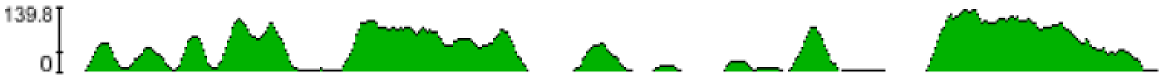
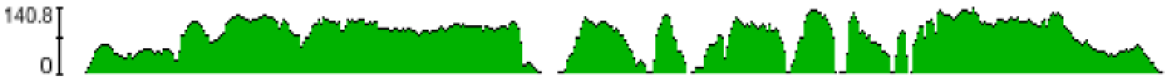
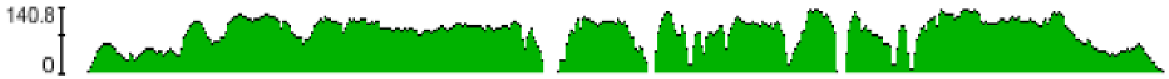
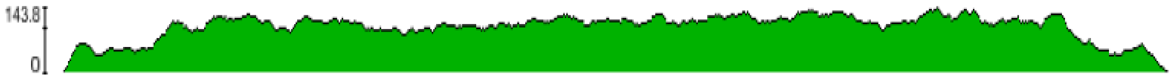
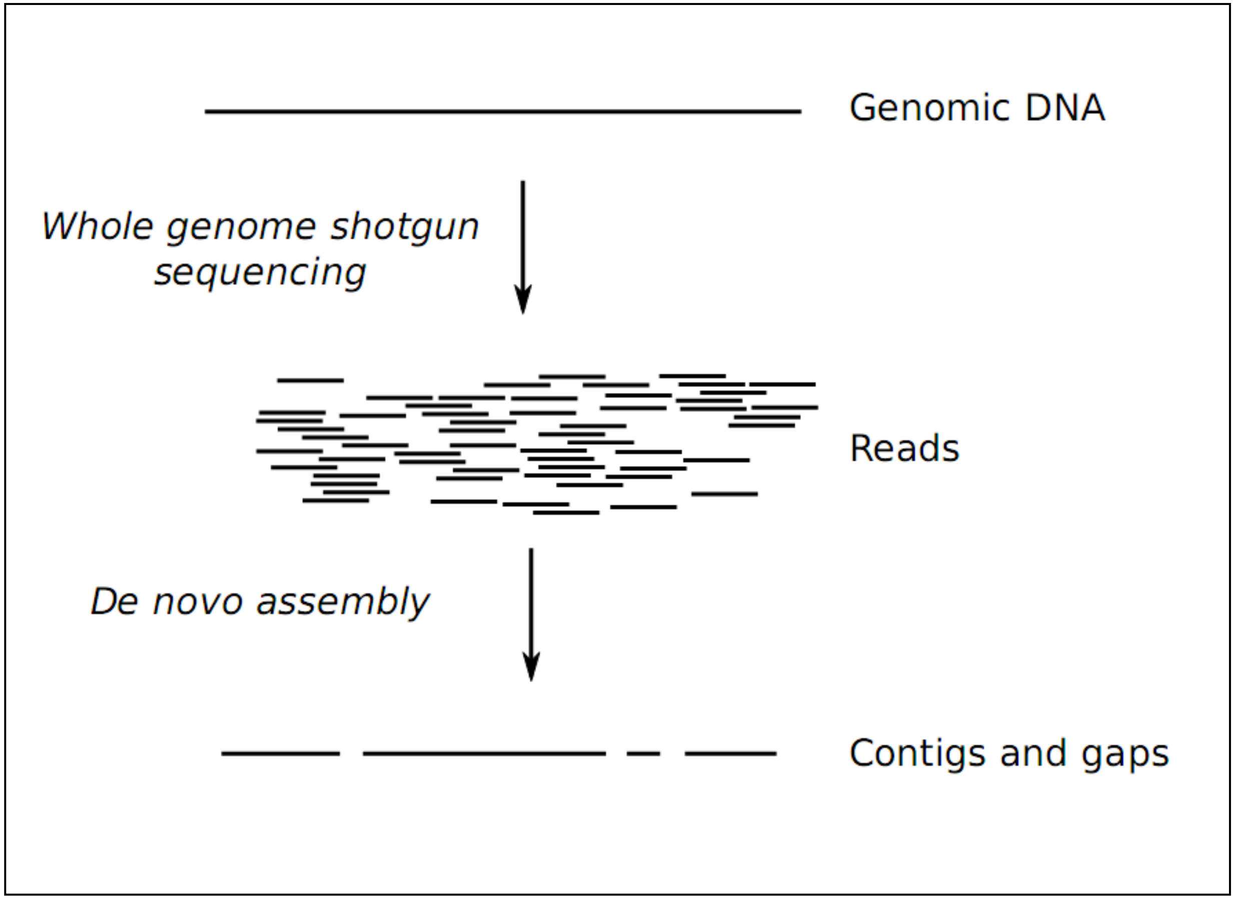
Algorithm	Coverage
Bowtie 1 (default)	
SOAP2 (default)	
BWA (default)	
Bowtie 2 (default)	
Bowtie 2 (very sensitive)	
SMALT (default)	
Geneious (default)	

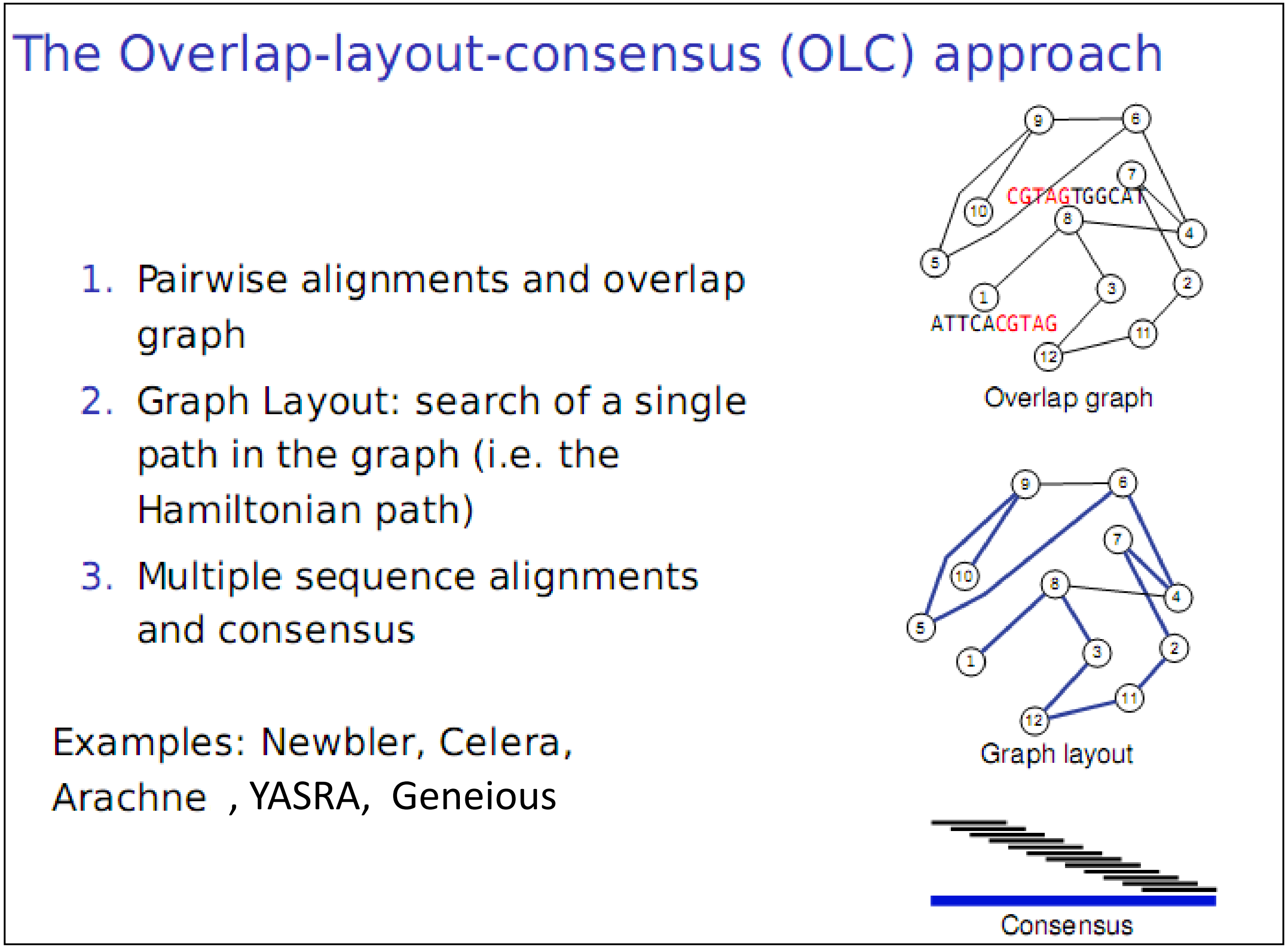
Figure 5: Graphical coverage plots in Geneious for each mapping algorithm

De Novo Assembly



Tristan Lefébure, Cornell University

De Novo Assembly



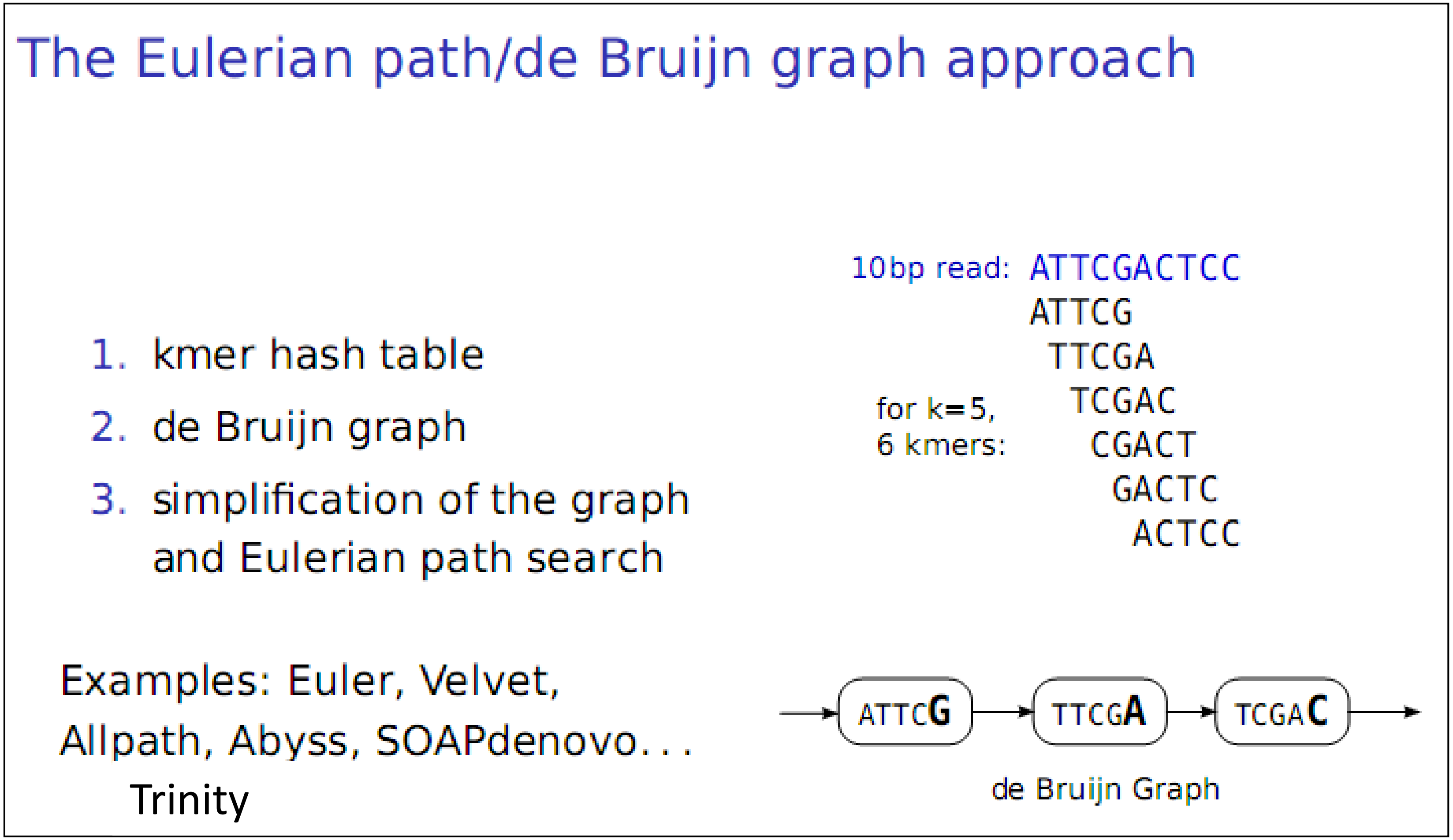
The Overlap-layout-consensus (OLC) approach

1. Pairwise alignments and overlap graph
2. Graph Layout: search of a single path in the graph (i.e. the Hamiltonian path)
3. Multiple sequence alignments and consensus

Examples: Newbler, Celera, Arachne , YASRA, Geneious

Tristan Lefébure, Cornell University

De Novo Assembly



The Eulerian path/de Bruijn graph approach

1. kmer hash table
2. de Bruijn graph
3. simplification of the graph and Eulerian path search

Examples: Euler, Velvet, Allpath, Abyss, SOAPdenovo. . .
Trinity

```
10bp read:  ATTCGACTCC
            ATTCG
             TTCGA
for k=5,      TCGAC
6 kmers:       CGACT
                GACTC
                 ACTCC
```

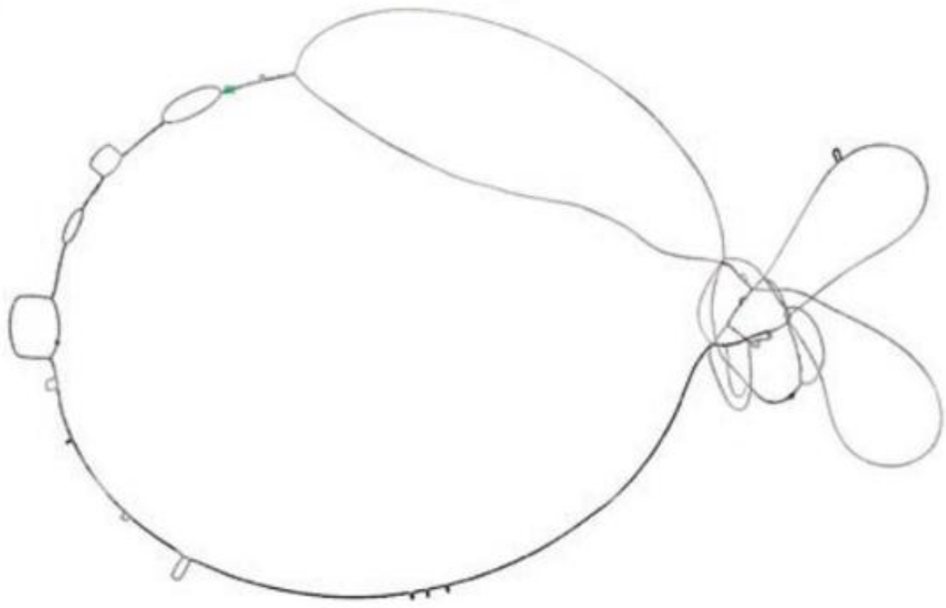
ATTCG
TTCGA
TCGAC

de Bruijn Graph

Tristan Lefébure, Cornell University

De Novo Assembly

de Bruijn Graph Approach

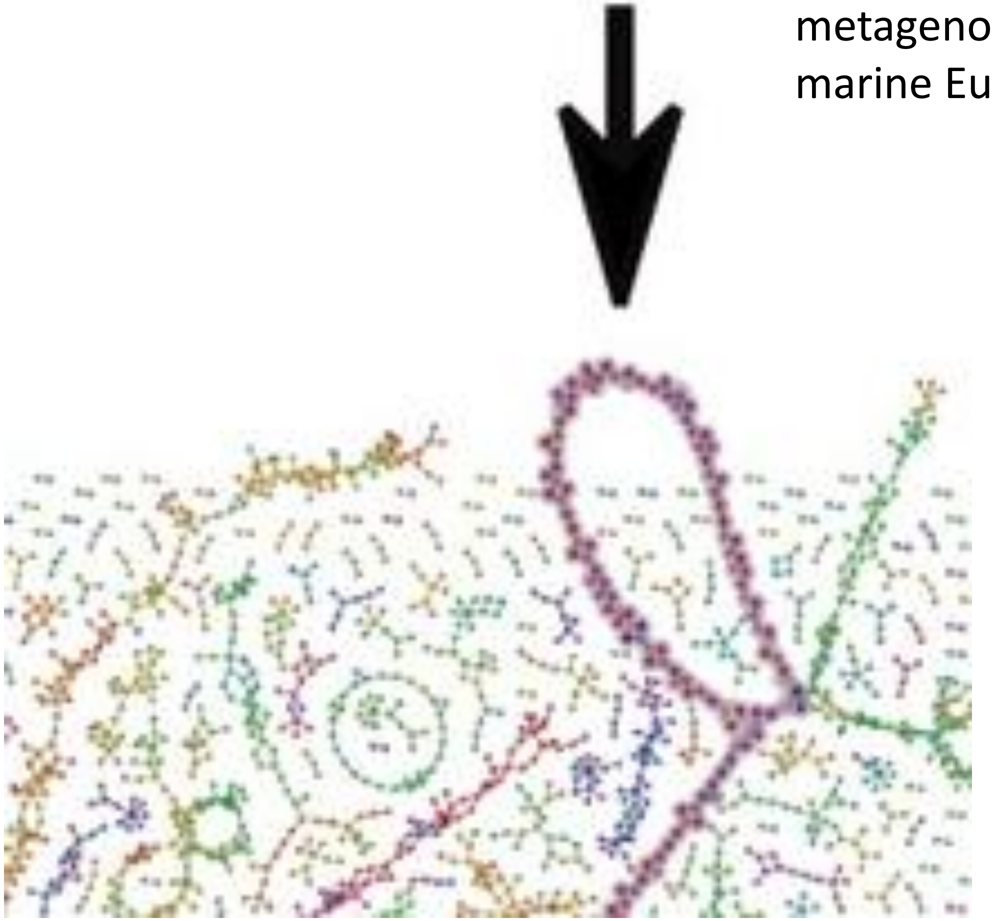


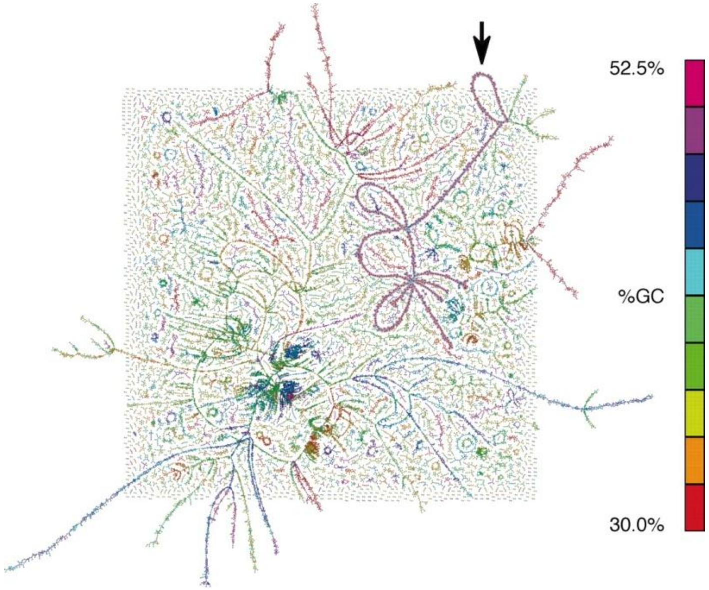
A full de Bruijn graph of two related plasmids. The de Bruijn graph was created with 30-bp *k*-mers. The open loops (bubbles) are regions that differ between the two plasmids, whereas the heavier lines indicate common regions.

Flicek & Birney. 2009. Sense from sequence reads: methods for alignment and assembly. Nature Methods 6: S6-S12.

De Novo Assembly

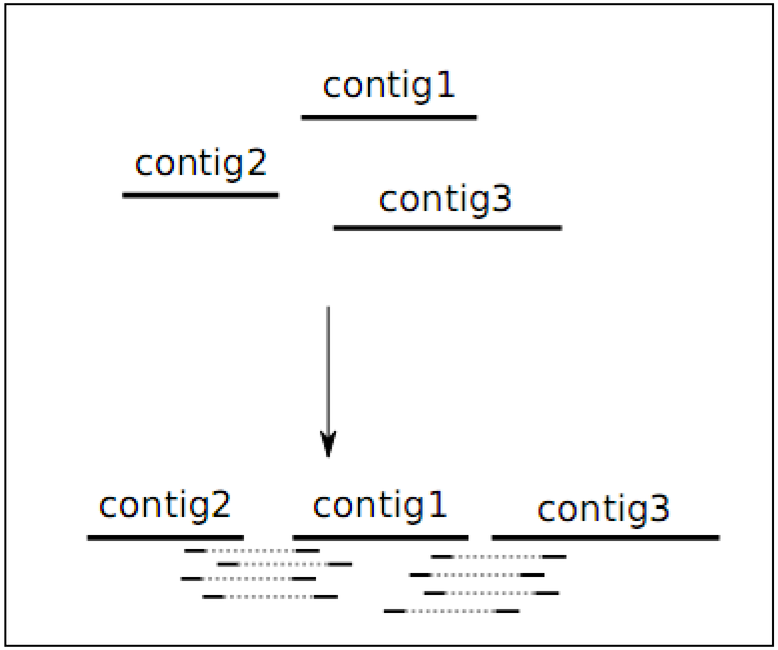
Iverson et al. 2012. Untangling genomes from metagenomes: revealing an uncultured class of marine Euryarchaeota." *Science* 335: 587-590.





Mate-pair connection graph illustrating the metagenome de novo assembly. Lines represent contigs with mate-pair connections scoring greater than 750 bits (n = 30,945). Long strands represent prokaryote genome sequences, and small circular strands show likely virus or plasmid sequences. The MG-II genome assembly is marked.

Strategies to Improve De Novo Assemblies



Tristan Lefébure, Cornell University

A. Informatics

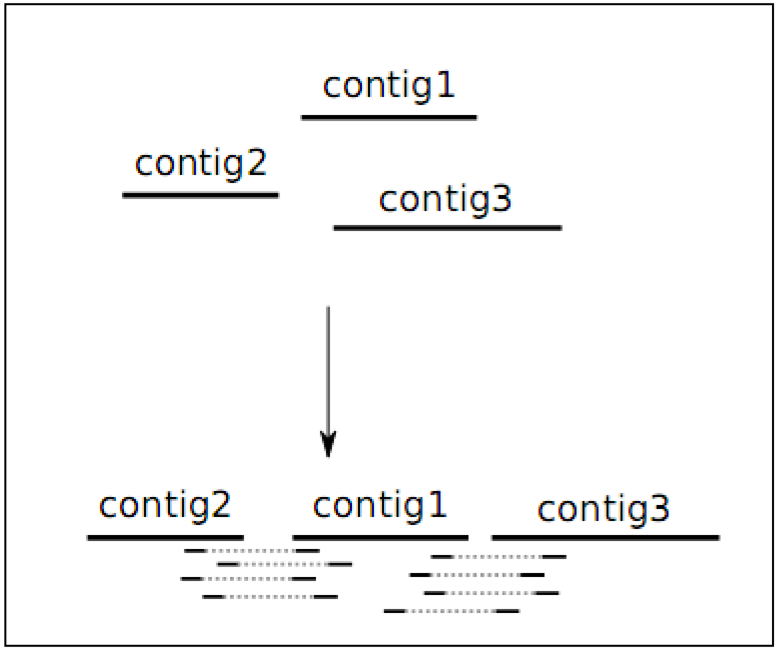
1. Remove low quality reads

2. Remove duplicate reads

3. Remove contaminating reads (adapters, other organisms, organelles)

4. Choose an appropriate k-mer (66%-95% of read length)

Strategies to Improve De Novo Assemblies



Tristan Lefébure, Cornell University

B. Library

1. Multiple paired end insert sizes (<1000 bp)
2. Mate-pairs (1 kbp – 20 kbp)
3. Fosmid ends (30-40 kbp)
4. BAC ends (up to 150 kbp)
5. longer reads

Strategies to Improve De Novo Assemblies

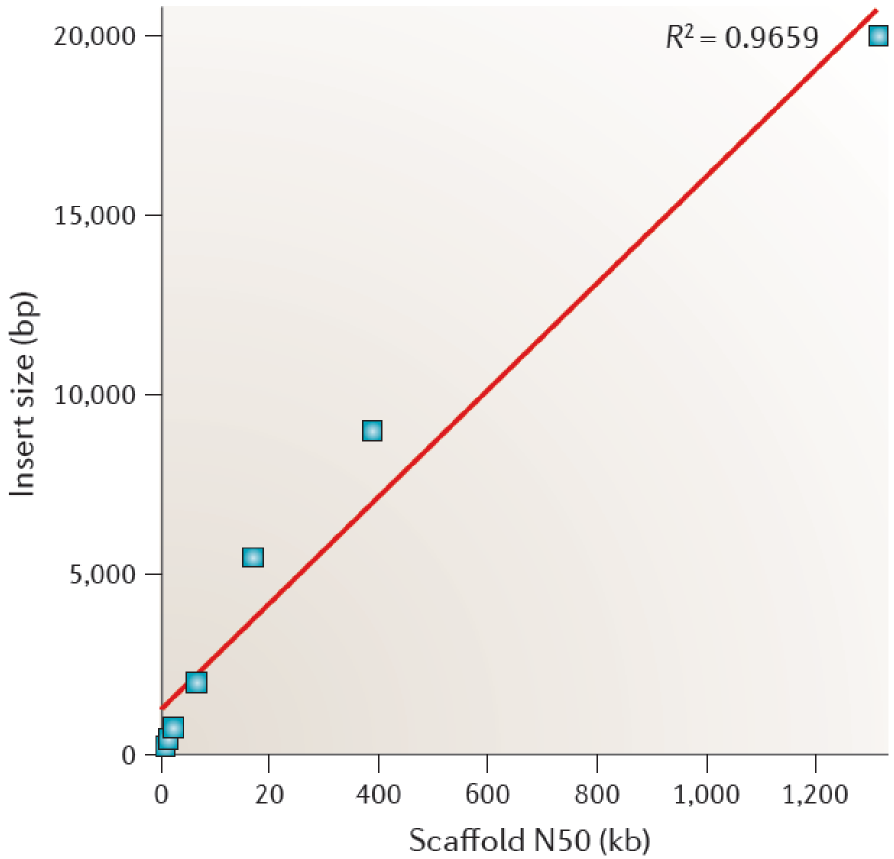
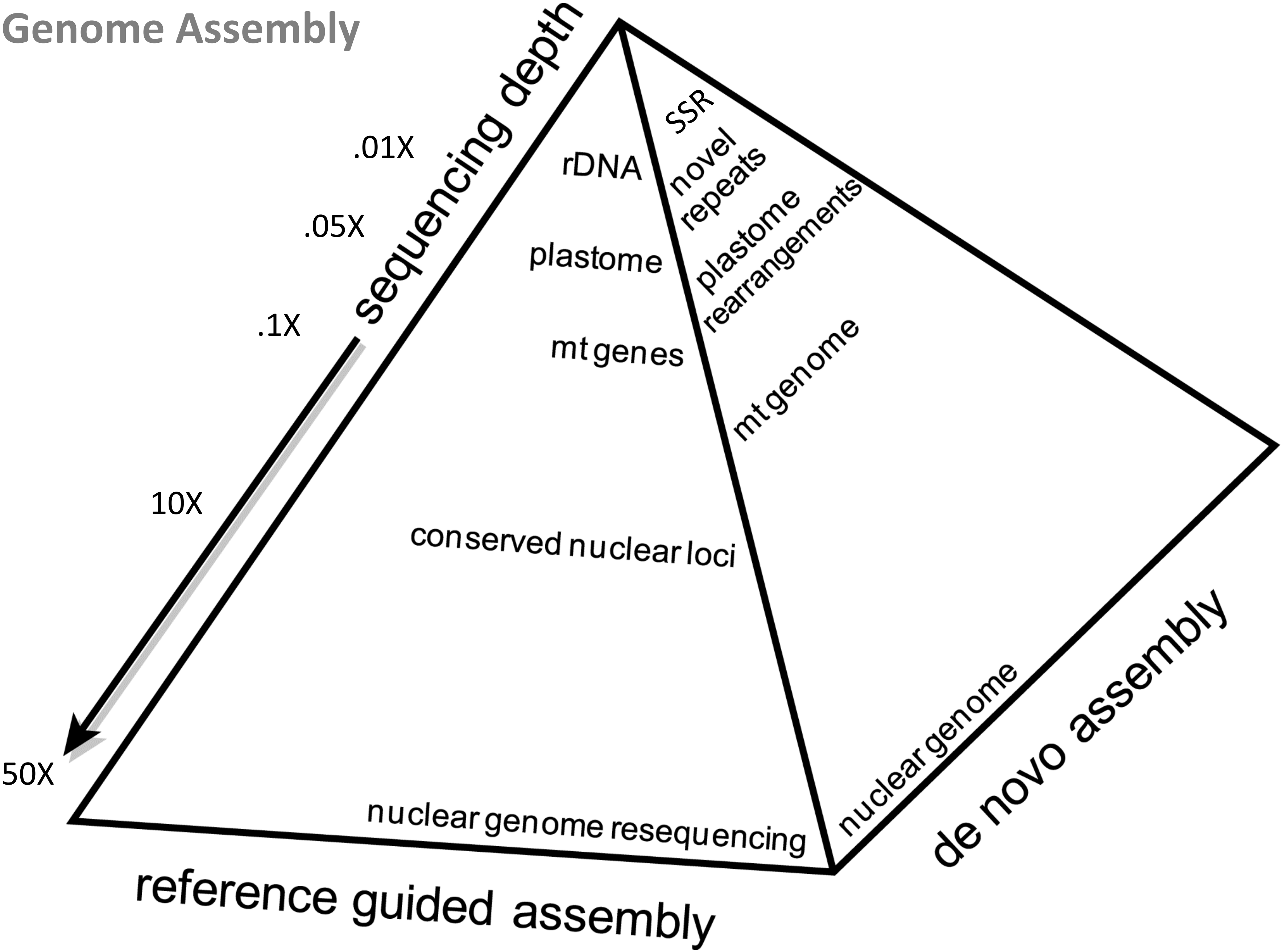


Figure 4 | **Longer paired-end libraries improved assembly contiguity in the repetitive potato genome.** Each point represents the scaffold N50 size of an assembly of the potato genome that was built using paired-end reads from inserts of a specific size and smaller. Successive points moving from left to right used all previous data plus one additional, longer paired-end library size, which is plotted on the *y* axis. With the addition of the final, 20 kb library, the scaffold N50 size reached 1.3 Mb. The data in this figure are taken from REF. 56.

Treangen & Salzberg 2012. data from Xu et al. 2011. Genome sequence and analysis of the tuber crop potato. Nature 475:189-(2011): 189-195.

Genome Assembly



Straub et al. 2012. American Journal of Botany 99:349-364.