Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation

Sergey Koren[1,5], Brian P. Walenz[1,5], Konstantin Berlin[2], Jason R. Miller[3], Nicholas H. Bergman[4] and Adam M. Phillippy[1]

[1] Genome Informatics Section, Computational and Statistical Genomics Branch, National Human Genome Research Institute, National Institutes of Health, Bethesda, Maryland 20892, USA;
[2] Invincea Incorporated, Fairfax, Virginia 22030, USA;
[3] J. Craig Venter Institute, Rockville, Maryland 20850, USA;
[4] National Biodefense Analysis and Countermeasures Center, Frederick, Maryland 21702, USA

Presented By
Anmol Sharma[1]

[1] MSc Student in Computing Science
Medical Image Analysis Laboratory
Simon Fraser University
Burnaby, BC Canada.

20 September 2018

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

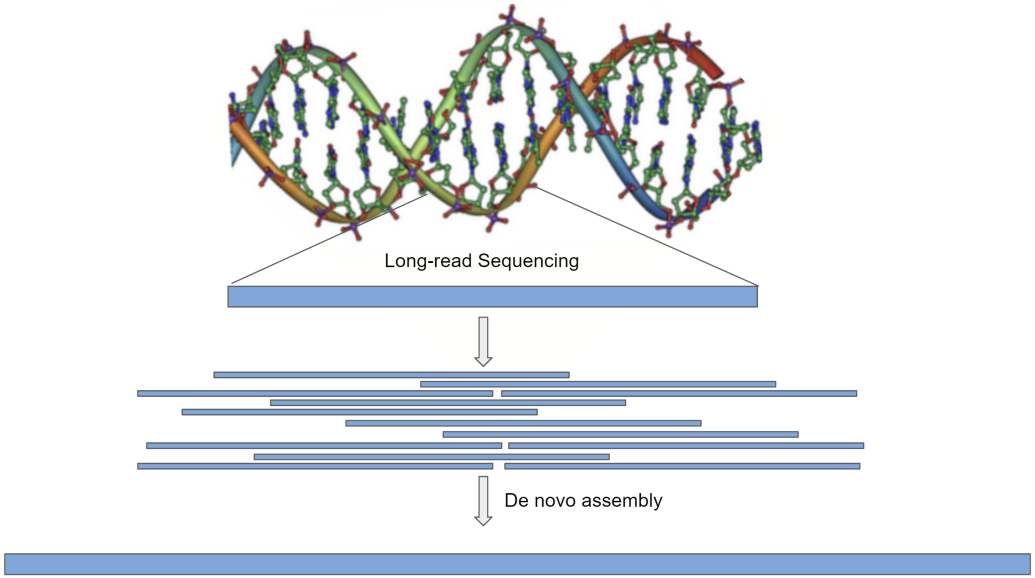
4 Results
- Performance and Comparison

5 Conclusion

Problem Statement

Overview

- *De novo* genome assembly from long-read, single molecule sequencing.
- Due to low accuracy of long-read sequencing technologies, efficient and accurate assembly of large repeats and closely related haplotypes remains challenging.
- **Question** : How do we accurately assemble the whole genome using noisy, inaccurate long-reads ?



Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Background

DNA Sequencing

- DNA is the basic building block of any organism which contains genetic “code” which makes all chemical activity in an organism possible.
- A code is analogous to a long sentence, and “bases” are the words that form that sentence. “Sequencing” a DNA refers to mapping out the exact sequence of bases as they are laid out in the DNA.
- Mapping the bases accurately is a required precursor for any downstream analysis such as in forensics, medical science and mutations.
- In recent years, advances in sequencing technology (like PacBio and Oxford Nanopore) have enabled us to sequence the DNA faster, accurately, and with relatively low cost.

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Background

Sequencing Technologies

- **PacBio** from Pacific Biosciences : The Sequel System is based on Single Molecule, Real-Time (SMRT).
  - Capable of reading between 10-20kbp per read.
  - Workflow of <1 day.
- **Oxford Nanopore** : Direct, real time sequencing technology.
  - Capable of reading upto 2mbp per read.
  - Available in handheld versions as well as industrial sized versions.

Background

Outstanding Issues

- Despite rapid advances in sequencing technology there are still outstanding issues.
- Average number of bases in the DNA varies tremendously between different organisms, ranging from 4 million (E. coli K12) to 3 billion (human genome).
- The current sequencing technologies are only capable of randomly reading small subsequences of DNA from random locations, that too with errors.
- This is akin to a large jigsaw puzzle, with multiple overlapping pieces which may not even be correct, and no reference image to facilitate assembly of jigsaw pieces.

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Canu Pipeline

Overview of the System

- Canu pipeline is modular assembly of three stages :
  - Correction
  - Trimming
  - Assembly
- Each stage can be run independently or in sequence.
- Canu supports both single computer node and compute clusters, with support for most major job schedulers (SLURM, LSF, SGE).
- Summary statistics are output as the job progresses.
- Each stage uses it's own indexed database, which disregards the previous input data once created.

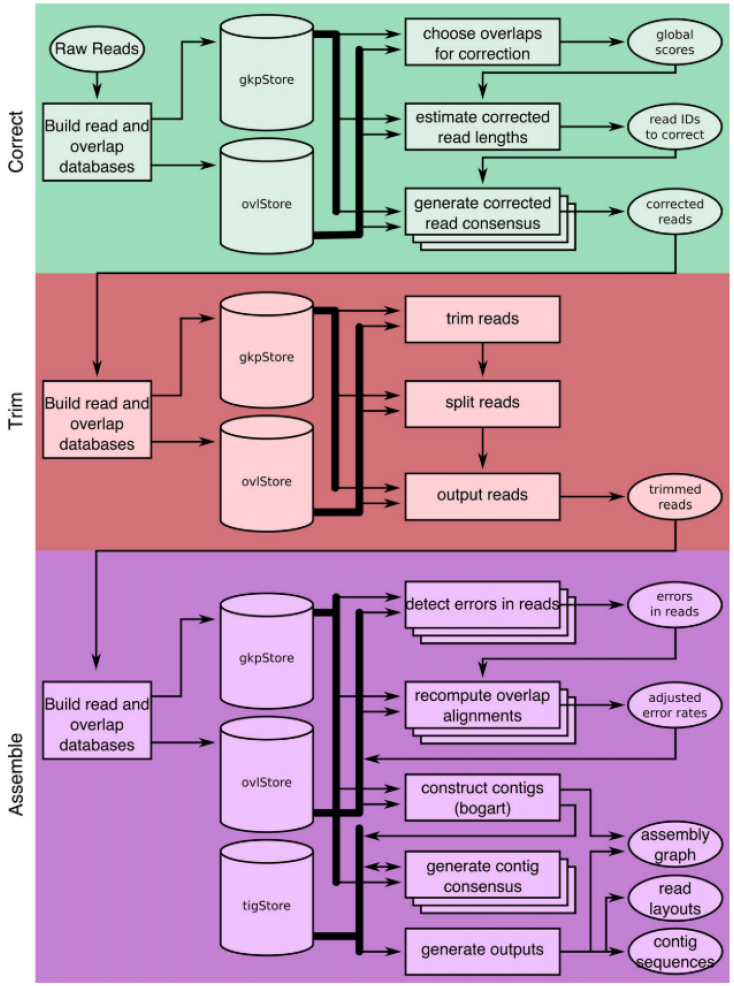


FIGURE – Canu whole pipeline

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Correction Stage

Overview

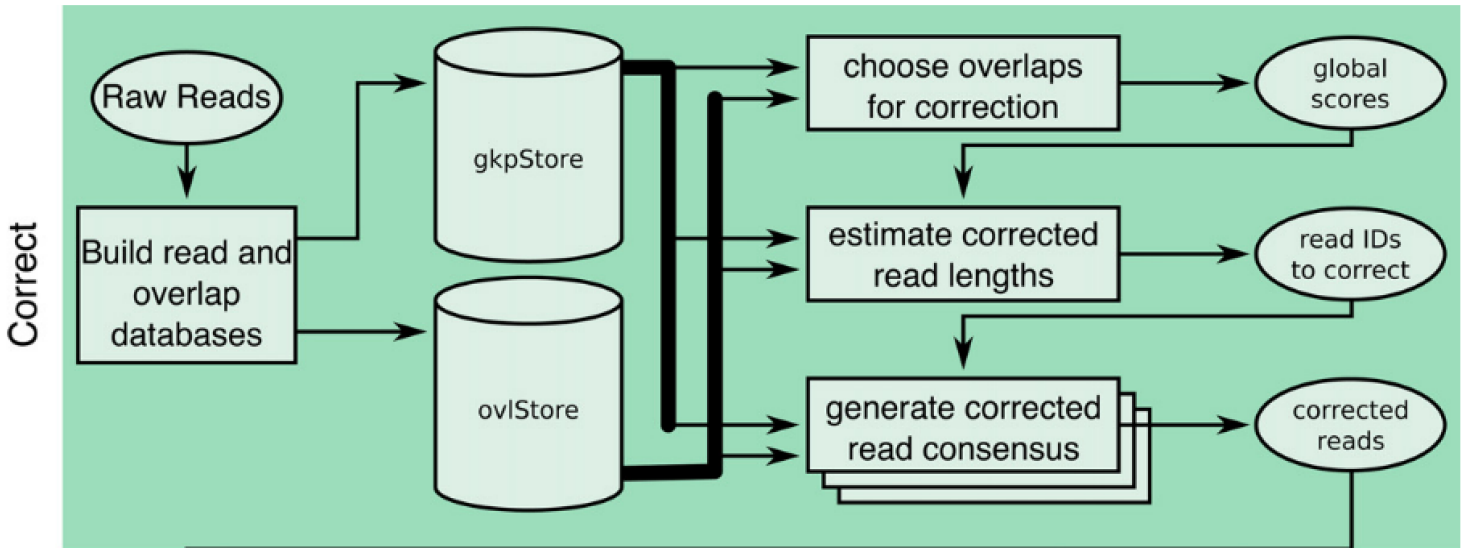


FIGURE – Correction Stage

Correction Stage

Identify Overlaps : MinHash Alignment Process (MHAP)

- MinHash Alignment Process (MHAP) was proposed in (Berlin 2015 [1]) for aligning overlapping noisy, long reads using probabilistic, locality-sensitive hashing.
- MHAP integrated with Celera Assembler was able to perform *reference grade* de novo assemblies of many genome sequences.
- It uses the idea of MinHash Sketches to perform rapid overlapping of noisy reads.

1. Berlin, Konstantin, et al. "Assembling large genomes with single-molecule sequencing and locality-sensitive hashing." Nature biotechnology 33.6 (2015) : 623.

Correction Stage

Identify Overlaps : MHAP Example

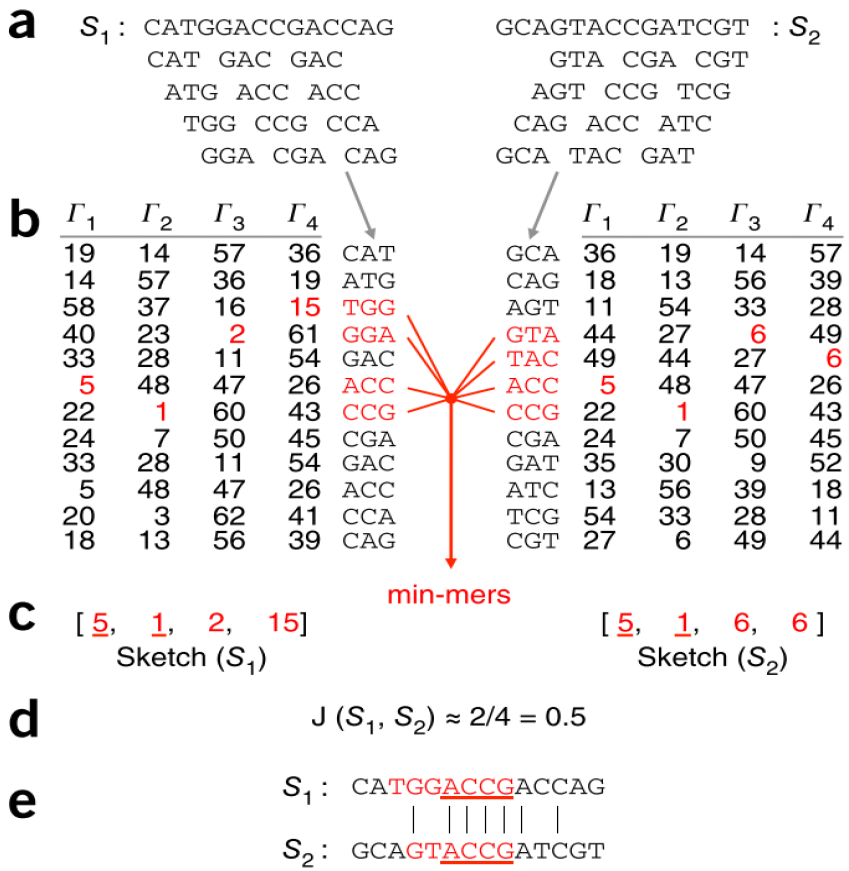
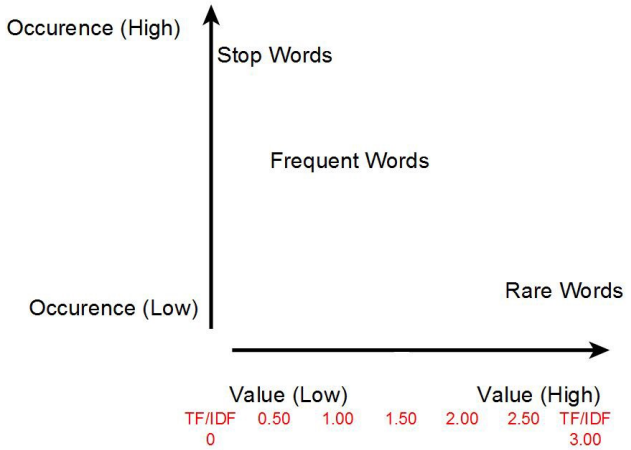


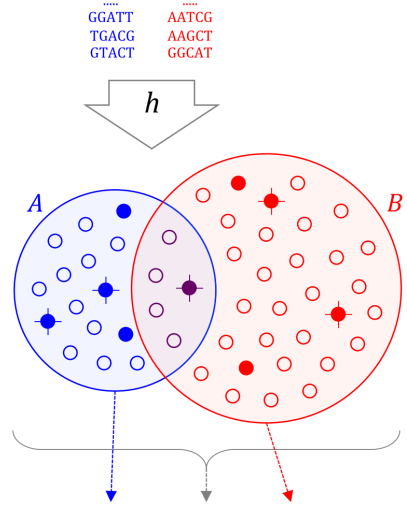
FIGURE – MHAP Example

Correction Stage

Identify Overlaps : MinHash Alignment Process (MHAP) with *tf-idf*

- *Canu* uses a modified version of vanilla MHAP.
- In *Canu*, MHAP uses a two-stage overlap filter.
  - STAGE 1 uses *tf-idf* weighting to prefer informative, non-repetitive *k*-mers which increases sensitivity to tru overlaps and reduces number of false, repetitive overlaps considered.
  - STAGE 2 MHAP uses “bottom-sketch" strategy similar to the one proposed in (Ondov 2016)[2]. Used to check the quality of overlap.





$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

2. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM. 2016. Mash : fast genome and metagenome distance esti- mation using MinHash. Genome Biol 17 : 132

Correction Stage

Read Correction

- Long read sequencing exhibit high error rate (5-15% depending on the technology).
- There is a need to to correct each of the individual reads using a consensus based approach.
- *Canu* uses all-vs-all overlap information to correct individual reads.
- It uses *two* filtering steps to determine which overlaps should be chosen to correct each individual read.
  - STAGE 1 (Global Filter) Each read chooses to provide correction evidence for *C* other reads.
  - STAGE 2 (Local Filter) Each read accepts/rejects the correction evidence supplied by other reads.
- This strategy was first proposed by (Koren et al. 2012) [3] as PBcR (PacBio Corrected Reads Pipeline), which performs hierarchical correction and assembly of single-molecule reads.
- Corrected reads are then generated using “falcon_sense" algorithm, originally proposed in (Chin et al. 2016) [4]

3. Koren, Sergey, et al. "Hybrid error correction and de novo assembly of single-molecule sequencing reads." Nature biotechnology 30.7 (2012) : 693.

4. Chin CS, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, Dunn C, O’Malley R, Figueroa-Balderas R, Morales-Cruz A, et al. 2016. Phased diploid genome assembly with single-molecule real-time se- quencing. Nat Methods 13 : 1050–1054.

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Trimming Stage

Overview

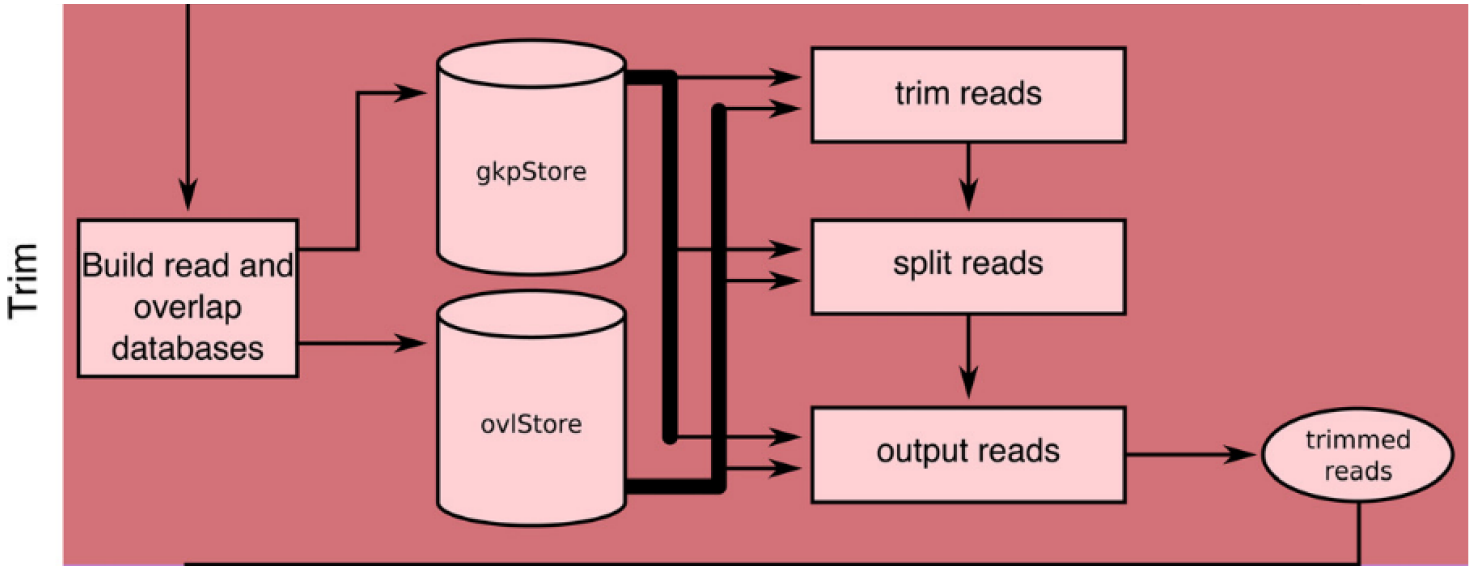


FIGURE – Trim Stage

Trimming Stage

Overlap Based Trimming (OBT)

- Once reads are corrected in the Correction stage, and new database of reads and overlaps is created once again in Trimming stage through overlapInCore, the reads are trimmed.
- Trimming is performed according to the algorithm first described in (Miller et al. 2008)[6].
- The reads are trimmed to the largest portion covered to at least depth C by overlaps of at most E error, and minimum length L.
- Parameters are technology dependent, and are chosen empirically.
- A second pass is made to detect hairpin adapters and chimeras in the reads. Once detected, the reads are trimmed to largest supported region.

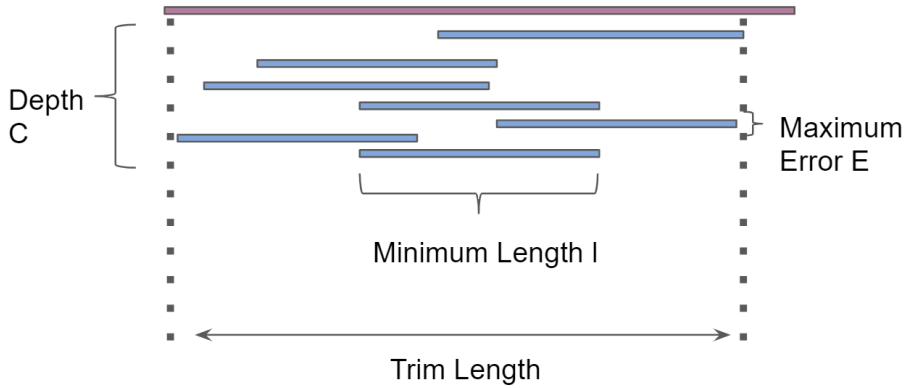


FIGURE – First pass to trim reads

5. Miller JR, Delcher AL, Koren S, Venter E, Walenz BP, Brownley A, Johnson J, Li K, Mobarry C, Sutton G. 2008. Aggressive assembly of pyrosequenc- ing reads with mates. Bioinformatics 24 : 2818–2824.

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Assembly Stage

Overview

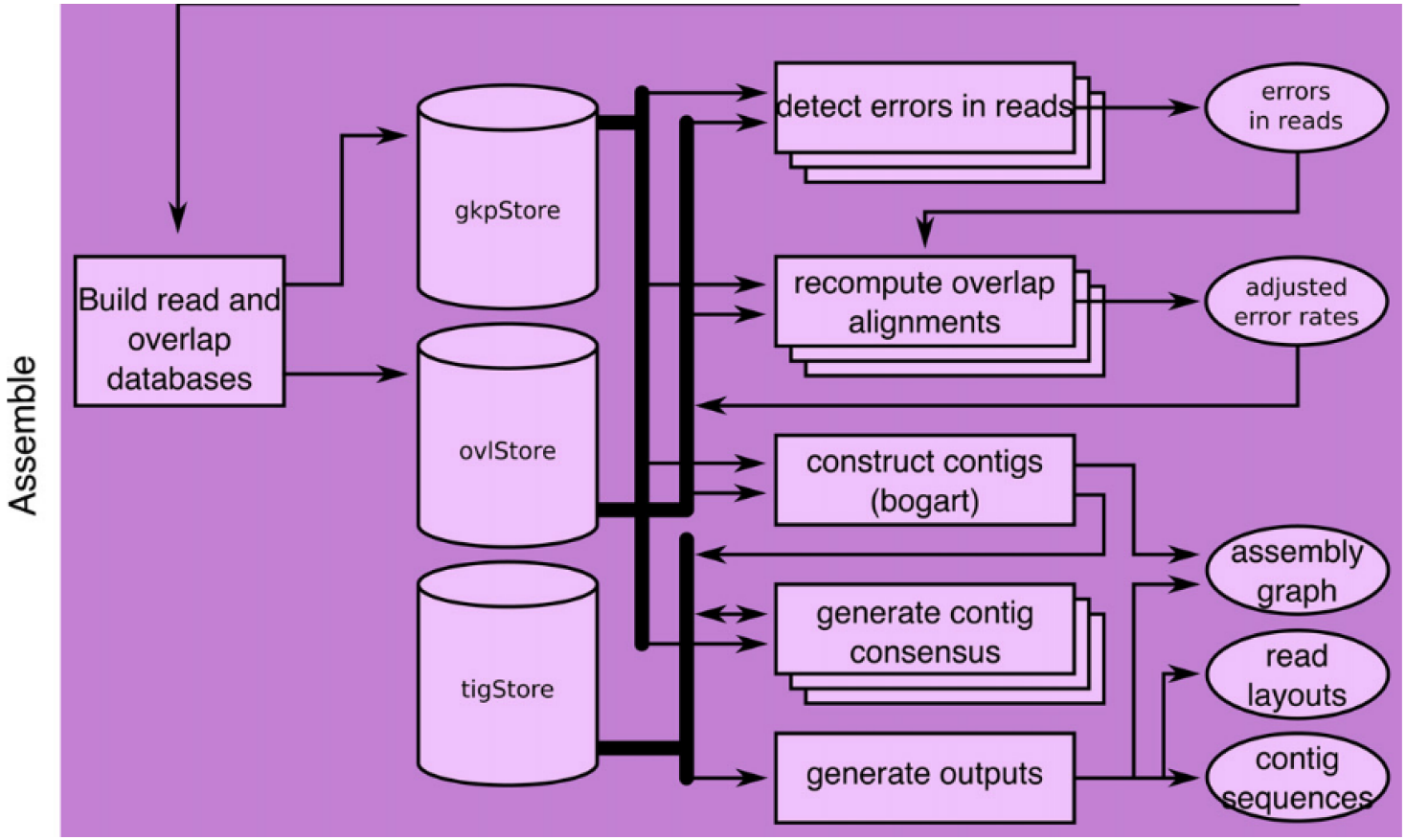


FIGURE – Assembly Stage

Assembly Stage

Overlap Error Adjustment

- After trimming and before graph construction, *Canu* recomputes overlaps and makes final attempt at detecting sequencing errors.
- The intuition is to improve separation between true sequencing differences (repeats and haplotypes) and false differences due to random error.
- Each read is corrected by the majority vote of it's overlapping reads.
- However the reads are **not changed**, since the overlaps have already been constructed.
- Reported error rates for each overlap is adjusted **had the changes in reads were made**.
- Two passes are made through the reads. The first pass finds the errors in reads, and the second pass temporarily applies the fixes in order to calculate reported error rates.
- This algorithm was first used in (Holt et al. 2002) [6].

6. Holt RA, Subramanian GM, Halpern A, Sutton GG, Charlab R, Nusskern DR, Wincker P, Clark AG, Ribeiro JC, Wides R, et al. 2002. The genome se- quence of the malaria mosquito Anopheles gambiae. Science 298 : 129–149

Assembly Stage

Graph Construction

- Graph construction uses the “best overlap graph" (BOG) strategy from (Miller et al. 2008)[a].
- Overlaps are defined as “containment".
  - If all bases in one read are aligned to another read, or *dovetail*, if involving only the ends of both reads.
- A “best overlap" is the longest dovetail overlap to a given read end.
- In original paper, all overlaps are picked up to a user specified cut-off threshold.
- In “BOGART" (Canu’s version of “best overlap graph"), the best overlaps are chosen after several filtering steps.
  - This removes high-error overlaps, potential chimeric reads, and reads whose overlaps indicate a possible sequence anomaly.
- BOGART results in a cleaner and more accurate graph construction.

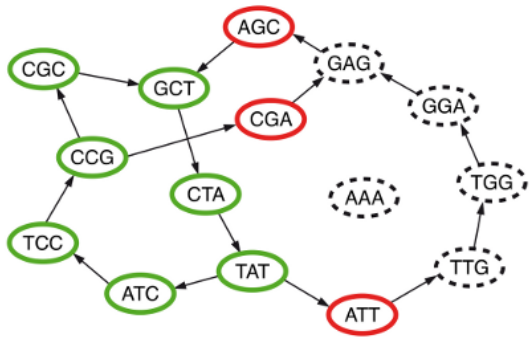


FIGURE – Overlap graph

7. Miller JR, Delcher AL, Koren S, Venter E, Walenz BP, Brownley A, Johnson J, Li K, Mobarry C, Sutton G. 2008. Aggressive assembly of pyrosequenc- ing reads with mates. Bioinformatics 24 : 2818–2824

Assembly Stage

Contig Consensus

- Contig consensus is performed using a slightly modified version of “pbdagcon" algorithm proposed in (Chin et al. 2013) [8].
- A “template sequence" is constructed for each contig by splicing reads together from approximate positions based on BOG.
- The template is accurate within individual reads, but may still contain indel errors at read boundaries.
- To correct this, all reads in contig are aligned to template sequence using Myers’ O(ND) algorithm [9], and added to a DAG.
- The DAG is used to call a consensus sequence as described in (Chin et al. 2013) [10].

8. Chin CS, Alexander DH, Marks P, Klammer AA, Drake J, Heiner C, Clum A, Copeland A, Huddleston J, Eichler EE, et al. 2013. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods 10 : 563–569
9. Myers EW. 1986. An O(ND) difference algorithm and its variations. Algorithmica 1 : 251–266
10. Chin CS, Alexander DH, Marks P, Klammer AA, Drake J, Heiner C, Clum A, Copeland A, Huddleston J, Eichler EE, et al. 2013. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods 10 : 563–569

Outline

1 Problem Statement
- Overview

2 Background
- DNA Sequencing
- Sequencing Technologies

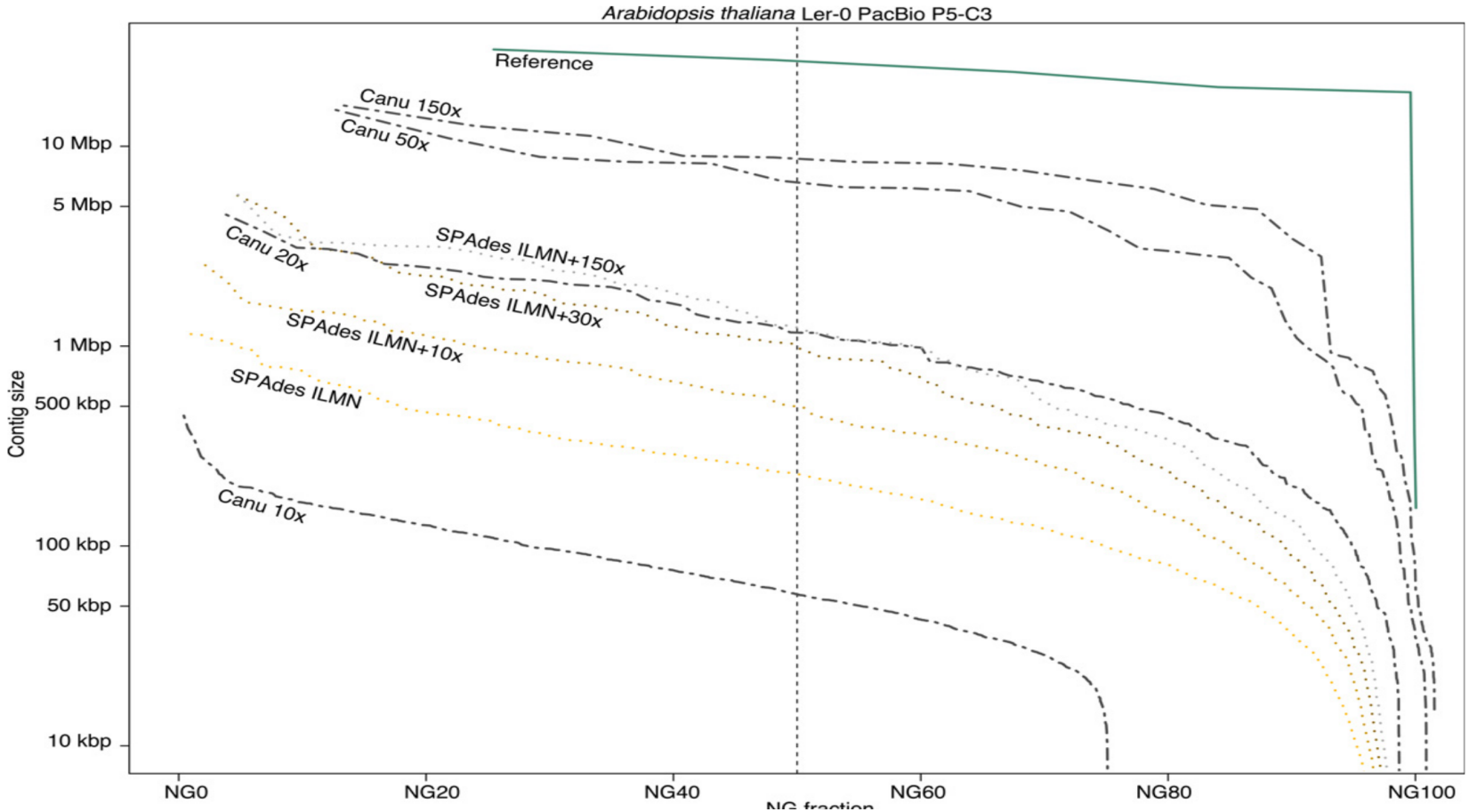
3 Canu Pipeline
- Overview
- Correction Stage
- Trimming Stage
- Assembly Stage

4 Results
- Performance and Comparison

5 Conclusion

Results

Performance of Canu as a function of coverage



Results

Canu on PacBio Data

Table 1. Canu is fastest for generating a high-quality polished assembly from PacBio data

Genome	Asm/Polish	Max (Mbp)	NG50 (Mbp)	% Ref	No. of breakpoints	Time (CPU h)	% Idy
Escherichia coli	Canu+Quiver	**4.68**	**4.68**	**100%**	**0**	12.25	**99.9999%**
	FALCON+Quiver	4.64	4.64	**100%**	2	25.14	99.9998%
	Miniasm+Quiver	4.64	4.64	99.99%	2	31.93	99.9998%
	SPAdes	4.64	4.64	**100%**	**0**	**4.09**	99.9972%
Drosophila melanogaster	Canu+Quiver	**25.78**	**21.31**	**97.47%**	1025	**1396.52**	99.9795%
	FALCON+Quiver	23.08	9.84	96.12%	1054	2305.92	**99.9813%**
	Miniasm+Quiver	15.85	5.84	96.51%	**752**	1484.33	**99.9813%**
Arabidopsis thaliana	Canu+Quiver	**15.95**	**8.31**	**82.94%**	220	**925.31**	**99.0710%**
	FALCON+Quiver	15.94	8.17	82.72%	222	1132.25	**99.0710%**
	Miniasm+Quiver	11.61	5.07	82.88%	**205**	976.43	**99.0710%**
Caenorhabditis elegans	Canu+Quiver	5.34	2.35	**99.70%**	139	410.07	**99.9745%**
	FALCON+Quiver	4.99	1.88	98.82%	**138**	**397.40**	99.9735%
	Miniasm+Quiver	**5.85**	**2.96**	99.44%	141	526.16	99.9706%
CHM1	Canu+Quiver	80.08	**21.95**	**86.84%**	1105	**22,749.71**	99.8081%
	FALCON+Quiver	52.34	9.46	86.58%	**1082**	68,789.00	**99.8086%**

Results

Canu on Oxford Nanopore Data

Table 2. Canu consistently assembles complete genomes from only Oxford Nanopore data

Genome	Asm/Polish	No. of contigs	Max (Mbp)	% Ref	No. of breakpoints	Time (CPU h)	% Idy
Escherichia. coli MAP005	Canu+Nanopolish	**(1)**	**4.64**	**99.98%**	2	376.87	**99.43%**
	FALCON+Nanopolish	105	0.42	23%	2	**106.2**	99.41%
	Miniasm+Nanopolish	3	3.40	99.96%	**0**	2344.02	99.36%
E. coli MAP006-1	Canu+Nanopolish	**(1)**	4.63	99.80%	**0**	**167.04**	**99.81%**
	FALCON+Nanopolish	**(1)**	4.63	99.86%	**0**	207.45	99.78%
	Miniasm+Nanopolish	**(1)**	**4.66**	**99.97%**	2	1801.02	99.72%
E. coli MAP006-2	Canu+Nanopolish	**(1)**	4.64	99.91%	**2**	**168.69**	**99.78%**
	FALCON+Nanopolish	**(1)**	4.64	**99.94%**	**2**	196.16	99.76%
	Miniasm+Nanopolish	**(1)**	**4.65**	99.70%	4	1482.95	99.69%
E. coli MAP006-PCR-1	Canu+Nanopolish	**(1)**	**4.64**	99.95%	**0**	**164.08**	**99.84%**
	FALCON+Nanopolish	**(1)**	4.63	99.80%	2	168.37	99.82%
	Miniasm+Nanopolish	3	2.15	**99.96%**	**0**	1338.28	99.77%
E. coli MAP006-PCR-2	Canu+Nanopolish	**(1)**	4.64	99.99%	2	**206.09**	**99.85%**
	FALCON+Nanopolish	**(1)**	4.64	**100.00%**	2	212.89	99.84%
	Miniasm+Nanopolish	**(1)**	**4.65**	99.98%	**0**	1669.83	99.81%
Bacillus anthracis	Canu+Nanopolish	**(2)**	5.20	**99.77%**	**0**	894.40	99.14%
	FALCON+Nanopolish	31	0.47	86.29%	**0**	**795.93**	**99.17%**
	Miniasm+Nanopolish	4	**5.22**	97.21%	**0**	5094.90	99.05%
Yersinia pestis	Canu+Nanopolish	**(4)**	4.67	**99.97%**	**11**	**254.25**	**99.76%**
	FALCON+Nanopolish	**(4)**	**4.68**	**99.97%**	12	295.01	99.72%
	Miniasm+Nanopolish	9	2.69	99.91%	**11**	2000.16	99.65%

Results

Canu vs Hybrid Methods

Table 3. Nanopore assemblies exceed hybrid methods in continuity and match their quality when polished with Illumina data

Genome	Asm/Polish	No. of contigs	Max (Mbp)	% Ref	No. of breakpoints	Time (CPU h)	% Idy
E. coli MAP005	Canu+Pilon	**(1)**	**4.65**	99.99%	2	10.98	99.9873%
	FALCON+Pilon	105	0.42	23.04%	2	4.36	99.9550%
	Miniasm+Pilon	3	3.40	90.62%	42	**3.15**	97.3878%
	SPAdes	**(1)**	4.64	**100.00%**	**0**	3.61	**99.9989%**
E. coli MAP006-1	Canu+Pilon	**(1)**	4.63	99.82%	**0**	5.89	**99.9995%**
	FALCON+Pilon	**(1)**	4.63	99.86%	**0**	7.3	99.9964%
	Miniasm+Pilon	**(1)**	**4.66**	96.97%	21	**3.14**	99.6118%
	SPAdes	**(1)**	4.64	**100.00%**	**0**	3.65	99.9965%
E. coli MAP006-2	Canu+Pilon	**(1)**	**4.64**	99.94%	2	3.92	**99.9987%**
	FALCON+Pilon	**(1)**	**4.64**	99.94%	2	3.93	99.9933%
	Miniasm+Pilon	**(1)**	**4.64**	97.98%	26	**2.73**	99.6336%
	SPAdes	**(1)**	**4.64**	**100.00%**	**0**	3.56	99.9965%
E. coli MAP006-PCR-1	Canu+Pilon	**(1)**	**4.64**	99.95%	**0**	4.15	**99.9993%**
	FALCON+Pilon	**(1)**	4.63	99.80%	2	3.55	99.9969%
	Miniasm+Pilon	3	2.16	98.41%	12	**2.15**	99.6734%
	SPAdes	2	3.95	**100.00%**	**0**	3.56	99.9965%
E. coli MAP006-PCR-2	Canu+Pilon	**(1)**	4.64	**100.00%**	2	6.16	**99.9992%**
	FALCON+Pilon	**(1)**	4.64	**100.00%**	2	9.22	99.9963%
	Miniasm+Pilon	**(1)**	**4.65**	98.57%	20	**2.69**	99.6734%
	SPAdes	**(1)**	4.64	**100.00%**	**0**	4.00	99.9965%
B. anthracis	Canu+Pilon	**(2)**	5.21	99.77%	1	65.01	99.8476%
	FALCON+Pilon	31	0.48	86.31%	**0**	14.95	99.8888%
	Miniasm+Pilon	4	**5.25**	79.36%	44	**4.9**	92.2732%
	SPAdes	6	4.13	**100.00%**	**0**	8.47	**99.9948%**
Y. pestis	Canu+Pilon	**(4)**	**4.66**	**99.83%**	23	17.92	99.8946%
	FALCON+Pilon	**(4)**	4.64	99.65%	26	10.63	99.8715%
	Miniasm+Pilon	9	2.70	93.76%	42	**8.68**	98.7866%
	SPAdes	29	0.37	95.99%	**15**	17.08	**99.9559%**

Conclusion

Canu

- Canu is a three stage sequence assembly system that brings together improved state-of-art methods.
- Canu assembles high-quality polished assembly from PacBio data, and highly continuous genomes from Oxford Nanopore data.
- When compared to methods which perform the same task, Canu retrieved the highest coverage of the reference sequence, and also dramatically reduced the computational time (CPU hours).
- Canu provides a robust and efficient method for sequence assembly from long read sequence data, and enables high quality downstream analysis of the sequences due to its redundant error correction steps that lead to accurate and complete assemblies of the genome.

Thank You !

Questions ?