 doi:10.20944/preprints202009.0207.v1

SMARTdenovo: A *de novo* Assembler Using Long Noisy Reads

Hailin Liu* Shigang Wu* Alun Li* Jue Ruan✉

Guangdong Laboratory for Lingnan Modern Agriculture, Genome Analysis Laboratory of the Ministry of Agriculture, Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences, Shenzhen, 518120, China

*These authors contributed equally to this work.
✉To whom correspondence may be addressed:
Jue Ruan: ruanjue@caas.cn; ORCID ID: https://orcid.org/0000-0003-3713-3192

Abstract:
Long-read single-molecule sequencing has revolutionized *de novo* genome assembly and enabled the automated reconstruction of reference-quality genomes. It also has been widely used to study structural variants, phase haplotypes and more. Here, we introduce the assembler— SMARTdenovo, which is an SMS assembler that follows the overlap-layout-consensus (OLC) paradigm. SMARTdenovo (RRID: SCR_017622) was designed to be a fast assembler that did not require highly accurate raw reads for error correction, unlike other, contemporaneous SMS assemblers. It has performed well for evaluating congeneric assemblers and has been successful for a variety of assembly projects. It is compatible with Canu for assembling high-quality genomes, and several of the assembly strategies in this program have been incorporated into subsequent popular assemblers. The assembler has been in use since 2015, and here we provide information on the development of SMARTdenovo and how to implement its algorithms into current projects.
Keywords: SMARTdenovo; single-molecule sequencing; genome assembly

 doi:10.20944/preprints202009.0207.v1

Introduction

The development of high-throughput sequencing provide the means to deliver fast, inexpensive, and accurate information for assembling whole genomes. As a result, there has been a rapid growth in the number of whole-genome sequencing projects [1-3]. Single-molecule sequencing (SMS) technologies, such as Pacific Biosciences (PacBio) and Oxford Nanopore, which generate sequencing reads of >20 kb in length, are now widely used in whole genome projects. These long reads are advantageous because they span polymorphic regions, repeats, and transposable elements, and because they provide long-range information for assemblies that are usually too complex to be resolved by short reads alone.

The huge demand for long-range DNA sequencing and mapping technologies has catalysed a renaissance of the development of high-quality SMS assemblers, such as PBcR[4, 5], Falcon[6], Canu[7], Miniasm[8], Ra[9], Wtdbg2[10], Flye[11], ABruijn[12]. In fact, highly available SMS assemblers have always been essential for improving the quality of genome assemblies.

SMARTdenovo is a long-read SMS assembler that follows the overlap-layout-consensus (OLC) paradigm. It assembles genomes following four steps: overlapping, trimming, layout, and consensus. The source code for SMARTdenovo was released in GitHub in 2015, and assessment by others has shown that it performs well when compared to other congeneric assemblers [13], and it has been widely used for generating highly accurate contigs in many genome assemblies [14-16]. For datasets from both PacBio and Nanopore, such as 20~30× 2D reads of different varieties of yeast, SMARTdenovo assembled the most accurate and highest contiguity sequences compared with other assemblers [17, 18]. SMARTdenovo was also used successfully for datasets from the wild tomato *Solanum pennellii* (~1.2 Gb) and *Sorghum bicolor* (~732 Mb) using Nanopore reads [16, 19], and for the long-read datasets for *Taraxacum kok-saghyz* (~1.04 Gb) and the woody plant, *Rhizophora apiculata* (~274 Mb) using PacBio RSII reads [20, 21]. Here we provide the details of how we developed SMARTdenovo and use cases to show its ability.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 September 2020 doi:10.20944/preprints202009.0207.v1

Methods

The SMARTdenovo algorithm. SMARTdenovo uses four steps for assembly: overlapping, trimming, layout, and consensus. We used homopolymer-compressed (HPC) k-mers for seed-indexing and identifying collinear seeds. HPCs have been widely adopted in Minimap2, Wtdbg2 and SHASTA[10, 22, 23]. We then trimmed low-quality regions and chimeric reads based on the overlapping reads. We applied the Best-Overlap-Graph [24] to generate the layout of the reads and the PBDAG-Con algorithm[25] to generate a consensus.

Overlapper. The algorithm for alignment follows a typical seed-chain-align procedure that is used by most full-genome aligners. In this step, we constructed each read by contracting homopolymer reads to a single base, called HPC strings. An HPC k-mer, a 1000-bp substring of an HPC string, were treated as a seed ("wtzmo -k 16 by default"). For HPC-based k-mer-indexing, we scanned all the reads and counted kmers in a hashtable with a 64-bit key to store a kmer and a 64-bit value to store its count. Kmers that were present more than 500 times were filtered out ("wtzmo -K 500"). A seed array was also created and filled with the "read_id" and the orientation of the remaining seeds. To manage the cost of memory for k-mer-indexing, we used two different parameters: 1) we kept only the index with smaller values between the k-mer and its reverse complement; 2) we randomly selected k-mers according to the hash code (one quarter was set as the default). All queried k-mers were indexed against the hashtable and seed array to identify candidate reads. We sorted the seeds by the "read_id" and "strand" and calculated the coverage length of the overlaps. If the coverage was longer than 300 bp (wtzmo -d 300), the candidate would be kept. The top 500 candidates were chosen for each query (wtzmo –A 500).

To refine the collinearity relationship between query and candidate, we further built a similar but shorter HPC-based index called a "z-mer" on queries for seed chaining. Each z-mer was recorded with its offset and strand. Pairs of matched seeds between candidate and query were obtained. However, because the high rate of INDELs made the distance between two nearby z-mers high variable, wtzmo was used to identify the synteny between query and candidate in using sliding windows (wtzmo -y 800) on query instead of using the whole overlapping regions. We first filtered z-mer-windows using a minimum match length of 200 bp. We then developed a scoring algorithm to filter excessive matches. The adjacent matched pairs of seeds (p_i, p_{i+1}) were scored based on the relationship of z-mers on candidates: $S_{i+1}=S_i+L_{i+1}-Distance_{i-(i+1)}$, S、L、D represented the sort of seeds, length of seeds, and the distance between the adjacent seeds, respectively. If S_{i+1} was larger than L_{i+1}, pairs of p_i, p_{i+1} were considered to be a syntenic block, and the block would be extended until S_j was smaller than L_j. At that point, the block ended, S_j recovered to L_j and the next round of syntenic block identification began. The z-mer-block that contained the highest value of S was the block chosen as the best syntenic overlap between the two windows. If the coverage was longer than 100 bp, the matched windows were retained. Seed-windows were also scored with the same method as above for searching the best colinear pairs. Finally, candidates with the best colinear window-block that covered more than 300 bp were retained for pairwise alignment.

To speed calculation time and reduce unnecessary computation, we developed a weighting algorithm that ranked a repetitive region negatively based on its depth in the alignment process. Wtzmo -q 100 was set so that the weights ranged from 1 to 0 if the depth ranged from 10 to 100. Large numbers of false candidates that contained similar repeats were eliminated with queries. The pairwise alignment procedure based on the collinearity relationship was split into four steps: 1) first, the bases of the z-mers were

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 September 2020 doi:10.20944/preprints202009.0207.v1

decompressed and gaps were added between the matched z-mers; 2) global alignment was conducted between two adjacent z-mers within a z-mer-window; 3) the two adjacent z-mer-windows were aligned using the banded global Smith-Waterman algorithm with a dynamic bandwidth that increased according to the length of the gaps (wtzmo –w 50 ~ 3200); 4) the two ends were extended by global alignment with the band width set at 800 bp.

Trimming. Wtclp trimmed or discarded reads to a maximize total length of the valid overlaps. It took one read as a reference, and tiled all reads containing overlaps. A functional model, "call_legal_overlaps_wtclp", calculated the length of valid overlaps. First, it clipped the ends that had high error rates. Then, it detected chimeras and trimmed them according to their depths. Structures that contained partially aligned reads were called "spurs". We counted the depth of reads crossing the "spurs" as "m" and counted the number of reads with partial alignments as "n". If the "m" of a read was less than half of the average depth, or if "n" was larger than the average depth, or if "n" was larger than "m/2", the read was considered chimeric and discarded. Otherwise, it was considered as a sequencing error and the maximum region of reads were retained. Errors in the structure were corrected based on the graph. If a single read connected two subgraphs, it was considered a chimera. We then used wtclp to check whether there was an alternative path formed by valid overlaps of tiled reads.

Layout. Wtlay was used to achieve the Best-Overlap-Graph (BOG) [24] to generate layout of reads following the OLC paradigm. In general, if an overlap was not end-to-end, leaving N (no greater than 100) bp unaligned, it would be treated as true (wtlay -w 100). Due to the high INDEL rate, wtlay identified the best overlap with an alignment score ≥0.95 instead of picking out the longest one (wtlay -r 0.95). Using this process, would keep bubbles from being merged, and instead find one appropriate path. The wtlay script also filtered out each unitig that shared more than 40% identity with another unitig to avoid islands. The output included uncorrected unitigs and all the parameters needed by the consensus caller.

Consensus. We used the wtcns command to implement the PBDAG-Con algorithm described in HGAP to generate consensus [25]. Because an alignment algorithm is integrated into wtcns, it did not require any other alignment tools. Wtcns took the layout file as input and output the consensus sequences in fasta format.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 September 2020 doi:10.20944/preprints202009.0207.v1

Results

Assembling the genome of the fruit fly and evaluating accuracy of the assembly

We benchmarked SMARTdenovo against SMS assemblers Flye, Canu, Ra and Wtdbg2 using the dataset of the fruit fly, *Drosophila melanogaster*, and calculated the accuracy of the assembly by aligning it with the reference genome. A total of 29.3 Gb PacBio reads from the NCBI database (SRX499318) were assembled with the command line "smartdenovo.pl -c 1 -t 16 reads.fa > wtasm.mak && make -f wtasm.mak". The length of the genome sequence was 146 Mb, with N50 to be 18.83 Mb. We also tested four other SMS assemblers (Canu, Flye, Ra, Wtdbg2) using this dataset, and SMARTdenovo was superior to Flye, Wtdbg2 and Ra in both total length and contig N50, but was inferior to Canu (Table 1).

Generally, the genome size of *D. melanogaster* is estimated to be 180 Mb. Of this, 60 Mb of the genome consists of centric heterochromatin making it intractable for assembly [26]. In comparison to the released reference[27], SMARTdenovo, Canu and Wtdbg2 were able to create longer assemblies: ~146.29 Mb, ~161.97 Mb, and ~144,07 Mb, respectively. SMARTdenovo and Canu performed better than the other three SMS assemblers, not only by creating longer assembly lengths, but also having higher coverage when aligned to the reference genome.

Assembling the genome of the wild tomato

We also compared the performance of SMARTdenovo with that of four other SMS assemblers: Flye, Canu, Ra and Wtdbg2 using the dataset for the wild tomato, *Solanum pennellii*. A total of 27.5 Gb Oxford nanopore reads were downloaded from EBI database (PRJEB19787). A k-mer analysis of this dataset indicated that this accession of *S. pennellii* (LYC1722) has a genome size between 1 and 1.2 Gb[19]. We assembled a 30-fold Nanopore dataset and achieved an assembly of 902.96 Mb with the N50 size 339 kb (Table 2). We also tried Flye, Ra and Wtdbg2 on this dataset: Flye obtained the longest genome sequence (1.27 Gb) and a higher N50 value (429 kb). The results of Ra and Wtdbg2 were unable to achieve the same sequence length as SMARTdenovo. When taking into accout the computation time, SMARTdenovo required 651 CPU hours, which was 70 hours faster than Flye (Table 2).

Table 1: Evaluation of long-read assemblies on the fruit fly genome (PacBio datasets)

	Canu	Flye	Ra	SMARTdenovo	Wtdbg2
Total length/bp	161,976,465	137,023,027	139,271,520	146,291,629	144,067,721
Count of contigs	633	970	424	242	815
Total_length (>=50000 bp)/Mb	151.43	131.37	135.14	143.15	133.36
Largest_contig/Mb	25.87	25.74	5.82	23.29	9.31
NG50/Mb	21.44	19.03	1.19	18.83	3.52
LG50	3	3	29	3	12
Misassembled_contigs	149	58	43	50	167
Total_aligned_length/Mb	135.74	123.45	126.5	127.9	127.55

Table 2: Comparison of different assemblers on the wild tomato genome (Nanopore datasets)

Assembler	Total length/bp	Count of contigs	N50/bp	hours per CPU	peek MEM/Gb
Flye	1,265,086,225	10,398	429,250	723.91	135.57
Ra	815,599,886	6,490	161,107	507.97	135.5
SMARTdenovoo	902,985,307	4,395	399,450	651.72	27.78
Wtdbg2	890,128,856	9,400	246,954	126.67	63.7

 doi:10.20944/preprints202009.0207.v1

Discussion

SMARTdenovo is an accurate and efficient SMS assembler compatible with data formats output of both PacBio and Oxford Nanopore technologies. It consists of several command line tools: wtzmo, to overlap reads; wtclp, to trim low-quality regions and chimeras: wtlay, to generate the assembly graph layout; and "wtcns" to calculate the consensus. Based on the results from tests on the wild tomato dataset, we found that SMARTdenovo was more memory-intensive than the other SMS assemblers, but its performance was comparable, and it was faster. SMARTdenovo has been successfully used to assemble data from various species such as plasmids [28], protists [15, 29], fungi [17, 18, 30, 31], microorganisms [32], as well as the complex plants [16, 19-21].

In addition to its solid performance, SMARTdenovo includes multiple algorithms that can be — and have been — useful for improving other programs. These algorithms have had a positive impact on the following popular SMS assemblers, such as MECAT, Minimap2, Shasta, and Flye. For example, in developing SMARTdenovo, we introduced the first algorithm to use HPC-based *k*-mers, and this has now been incorporated into many other assemblers[10, 22, 23]. SMARTdenovo has had a more extensive impact on our development of the assembler Wtdbg2, as it includes several of its algorithms for handling long reads, including those for indexing, seed chaining, trimming, consensus, and some of the data formation.

SMARTdenovo has several other algorithms that have not yet been taken advantage of for use in other programs, such as its weighting algorithm for handling repeat regions, which significantly improves both its speed and the accuracy of the alignment. At this point, no other long read assemblers have this feature.

SMARTdenovo has been available since 2015 on github, but its performance not only remains comparable with current assemblers, and but also has several advantages as described. Furthermore, given its excellent performance for use on corrected long sequence reads, it continues to be widely used in for genome assembly projects today[33-42].

Availability of supporting source code and requirements

- Project name: SMARTdenovo
- Project home page: https://github.com/ruanjue/smartdenovo
- Operating systems: 64-bit Linux
- Programming language: C 92.2%, C++ 4.5%, Perl 3.1%, other 0.2%
- Other requirements: None
- License: GNU GPL-3.0
- RRID: RRID:SCR_017622

A Code Ocean capsule is available to execute and this in the Code Ocean platform [43].

Acknowledgements

This study was supported by the Natural Science Foundation of China (grant nos. 31571353 and 31822029 to J.R.).

Figure legends

Figure 1: The algorithm for processing overlaps. a. k-mer indexing and identification of candidate reads based on overlap coverage. LR: long reads. b. seed chaining based on sliding windows. c. weighting algorithm that ranked the repetitive regions negatively according to their depth. d. pairwise alignment with four steps.

References

1. Metzker ML. Sequencing technologies - the next generation. *Nat Rev Genet* 2010, 11:31-46.
2. Huang SW, Li RQ, Zhang ZH, *et al*. The genome of the cucumber, Cucumis sativus L. *Nature Genetics* 2009, 41:1275-U1229.
3. Li RQ, Fan W, Tian G, *et al*. The sequence and de novo assembly of the giant panda genome. *Nature* 2010, 463:311-317.
4. Koren S, Schatz MC, Walenz BP, *et al*. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat Biotechnol* 2012, 30:693-700.
5. Berlin K, Koren S, Chin CS, *et al*. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* 2015, 33:623-630.
6. Chin CS, Peluso P, Sedlazeck FJ, *et al*. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat Methods* 2016, 13:1050-1054.
7. Koren S, Walenz BP, Berlin K, *et al*. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res* 2017, 27:722-736.
8. Li H. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* 2016, 32:2103-2110.
9. Ra:https://github.com/lbcb-sci/ra.
10. Ruan J, Li H. Fast and accurate long-read assembly with wtdbg2. *Nat Methods* 2020, 17:155-158.
11. Kolmogorov M, Yuan J, Lin Y, *et al*. Assembly of long, error-prone reads using repeat graphs. *Nat Biotechnol* 2019, 37:540-546.
12. Lin Y, Yuan J, Kolmogorov M, *et al*. Assembly of long error-prone reads using de Bruijn graphs. *PNAS* 2016, 113:E8396-E8405.
13. Istace B, Friedrich A, d'Agata L, *et al*. de novo assembly and population genomic survey of natural yeast isolates with the Oxford Nanopore MinION sequencer. *Gigascience* 2017, 6:1-13.
14. Xu SH, He ZW, Zhang Z, *et al*. The origin, diversification and adaptation of a major mangrove clade (Rhizophoreae) revealed by whole-genome sequencing. *National Science Review* 2017, 4:721-734.
15. Belkhelfa S, Labadie K, Cruaud C, *et al*. Complete Genome Sequence of the Facultative Methylotroph Methylobacterium extorquens TK 0001 Isolated from Soil in Poland. *Genome Announc* 2018, 6.
16. Deschamps S, Zhang Y, Llaca V, *et al*. A chromosome-scale assembly of the sorghum genome using nanopore sequencing and optical mapping. *Nature Communication* 2018, 9:1-10.
17. Fournier T, Gounot JS, Freel K, *et al*. High-Quality de Novo Genome Assembly of the Dekkera bruxellensis Yeast Using Nanopore MinION Sequencing. *G3-Genes Genomes Genetics* 2017, 7:3243-3250.
18. Istace B, Friedrich A, d'Agata L, *et al*. de novo assembly and population genomic survey of natural yeast isolates with the Oxford Nanopore MinION sequencer. *Gigascience* 2017, 6.
19. Schmidt MH-W, Vogel A, Denton AK, *et al*. De novo assembly of a new Solanum pennellii accession using nanopore sequencing. *Plant Cell* 2017, 29:2336-2348.
20. Xu SH, He ZW, Zhang Z, *et al*. The origin, diversification and adaptation of a major mangrove clade (Rhizophoreae) revealed by whole-genome sequencing. *National Science Review* 2017, 4:721-734.
21. Lin T, Xu X, Ruan J, *et al*. Genome analysis of Taraxacum kok-saghyz Rodin provides new insights into rubber biosynthesis. *National Science Review* 2018, 5:78-87.
22. Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 2018, 34:3094-3100.
23. Shafin K, Pesout T, Lorig-Roach R, *et al*. Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nat Biotechnol* 2020:715722.
24. Miller JR, Delcher AL, Koren S, *et al*. Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics* 2008, 24:2818-2824.
25. Chin CS, Alexander DH, Marks P, *et al*. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat Methods* 2013, 10:563-569.
26. Adams MD, Celniker SE, Holt RA, *et al*. The genome sequence of Drosophila melanogaster. *Science* 2000, 287:2185-2195.
27. Hoskins RA, Carlson JW, Wan KH, *et al*. The Release 6 reference sequence of the Drosophila melanogaster genome. *Genome Res* 2015, 25:445-458.

28. Fang H, Feng J, Xu Y, *et al*. Sequencing of pT5282-CTXM, p13190-KPC and p30860-NR, and comparative genomics analysis of IncX8 plasmids. *Int J Antimicrob Agents* 2018, 52:210-217.
29. Pollo SM, Reiling SJ, Wit J, *et al*. MinION re-sequencing of Giardia genomes and de novo assembly of a new Giardia isolate. bioRxiv 343541; doi: https://doi.org/10.1101/343541
30. Wang X, Peng J, Sun L, *et al*. Genome Sequencing Illustrates the Genetic Basis of the Pharmacological Properties of Gloeostereum incarnatum. *Genes* 2019, 10.
31. Sossah FL, Liu Z, Yang C, *et al*. Genome sequencing of Cladobotryum protrusum provides insights into the evolution and pathogenic mechanisms of the cobweb disease pathogen on cultivated mushroom. *Genes* 2019, 10:124.
32. Shin SC, Kim H, Lee JH, *et al*. Nanopore sequencing reads improve assembly and gene annotation of the Parochlus steinenii genome. *Sci Rep* 2019, 9:5095.
33. Perumal S, Koh CS, Jin L, *et al*. High contiguity long read assembly of Brassica nigra allows localization of active centromeres and provides insights into the ancestral Brassica genome. *BioRxiv* 2020.
34. Xu Z, Gao R, Pu X, *et al*. Comparative Genome Analysis of Scutellaria baicalensis and Scutellaria barbata Reveals the Evolution of Active Flavonoid Biosynthesis. *BioRxiv* 2020.
35. Dussert Y, Legrand L, Mazet ID, *et al*. Identification of the first oomycete mating-type locus sequence in the grapevine downy mildew pathogen, Plasmopara viticola. *Current Biology* 2020.
36. Adams TM, Armitage AD, Sobczyk MK, *et al*. Genomic investigation of the strawberry pathogen Phytophthora fragariae indicates pathogenicity is associated with transcriptional variation in three key races. *Frontiers in Microbiology* 2020, 11:490.
37. Zhang S, Xia Z, Zhang W, *et al*. Chromosome-Scale Genome Assembly Provides Insights into Speciation of Allotetraploid and Massive Biomass Accumulation of Elephant Grass (*Pennisetum purpureum Schum*.). *bioRxiv* 2020.02.28.970749; doi: https://doi.org/10.1101/2020.02.28.970749
38. Fang Y, Coelho MA, Shu H, *et al*. Long transposon-rich centromeres in an oomycete reveal divergence of centromere features in Stramenopila-Alveolata-Rhizaria lineages. *PLOS Genetics* 2020, 16:e1008646.
39. Takehana Y, Zahm M, Cabau C, *et al*. Genome Sequence of the Euryhaline Javafish Medaka, Oryzias javanicus: A Small Aquarium Fish Model for Studies on Adaptation to Salinity. *G3* 2020, 10:907-915.
40. Armitage AD, Cockerton HM, Sreenivasaprasad S, *et al*. Genomics Evolutionary History and Diagnostics of the Alternaria alternata Species Group Including Apple and Asian Pear Pathotypes. *Front Microbiol* 2019, 10:3124.
41. Feron R, Zahm M, Cabau C, *et al*. Characterization of a Y-specific duplication/insertion of the anti-Mullerian hormone type II receptor gene based on a chromosome-scale genome assembly of yellow perch, Perca flavescens. *Molecular Ecology Resources* 2020, 20:531-543.
42. Large CR, Hanson N, Tsouris A, *et al*. Genomic stability and adaptation of beer brewing yeasts during serial repitching in the brewery. *bioRxiv 2020.06.26.166157; doi: https://doi.org/10.1101/2020.06.26.166157*
43. Jue Ruan. SMARTdenovo: a de novo assembler using long noisy reads. Code Ocean 2020. https://doi.org/10.24433/CO.4665826.v1