Indian Journal of Agricultural Sciences **90** (10): 2000–5, October 2020/Article
https://doi.org/10.56093/ijas.v90i10.107981

hAssembler: A hybrid *de novo* genome assembly approach for large genomes

AMIT KAIRI[1], PRIYANKA GUHA MAJUMDAR[2] and ATMAKURI RAMAKRISHNA RAO[3]

ICAR-Indian Agricultural Statistics Research Institute, New Delhi 110 012, India

Received: 18 February 2020; Accepted: 22 September 2020

ABSTRACT

Genome assembly is a process where large contigs and scaffolds are constructed from raw reads generated by sequencing machines. Based on the size of the generated reads they can be primarily categorized into short reads and long reads. Modern genome assemblers follow De Bruijn Graph (DBG) approach for assembly of short reads, whereas Overlap Layout Consensus (OLC) approach for assembly of long reads. For *de novo* genome assembly, DBG based assemblers are very efficient at repeat resolution but are computation intensive and sensitive to sequencing errors. On the other hand, OLC based assemblers are intuitive and very time efficient but not efficient at resolving repeat regions. Here, we developed an hAssembler, which leverages the advantages of both DBG and OLC approaches and compared its performance with the existing hybrid assemblers. It uses both long reads and short reads and run OLC and DBG in parallel. By using both the long and short reads, the time complexity of hAssembler was reduced considerably. The results showed that hAssembler outperformed the existing hybrid assemblers in terms of time and performance (N50) while assembling the large genomes.

Key words: DBG, Hybrid genome assembly, Illumina, OLC, Long Reads, PacBio, Short reads

The process of alignment and merging of reads generated by sequencing machines to form a longer DNA sequence and eventually organizing all the fragments into one genome sequence is known as genome assembly. The need for reconstructing the reads lies in the fact that all the second generation as well as third generation sequencing platforms generate only a small fragment of DNA of the genome. The second-generation sequencer, namely, HiSeq generates short reads around 200 bp with 2% error rates (Au *et al.* 2012).

On the other hand, third generation sequencing technologies like Pacific Biosciences (PacBio) generates relatively longer reads (up to 10 kb) (Gordon *et al.* 2016) but with high error rates (Salmela *et al.* 2014). Broadly two genome assembly approaches, *viz.* reference genome assembly and *de novo* genome assembly, are followed to assemble the reads generated from the 2nd and 3rd generations of Next Generation Sequencing (NGS). To be specific, reference genome assembly is done by keeping a reference genome in the background and the contigs and scaffolds are generated (Pop *et al.* 2004). On the other hand, *de novo* genome assembly is done when no reference genome is available. Unlike reference assembly, *de novo* assembly approach is employed to assemble new and completely unknown species and hence is of more importance. Further, the approaches used for *de novo* genome assembly are broadly categorized into two approaches: Overlap Layout Consensus (OLC) and De Bruijn Graph (DBG).

Phrap (Green 1994) is one of the first OLC based assemblers, which was developed for very short whole-genome shotgun (WGS) sequencing reads. Other prominent assemblers that fall under such category include Celera assembler (Myers *et al.* 2000), ARACHNE (Batzoglou *et al.* 2002), Phusion (Mullikin and Ning 2003), RePS (Wang *et al.* 2002), PCAP (Huang *et al.* 2003), and Atlas (Havlak *et al.* 2004). The disadvantage in OLC based assembler was the inability to address the problem of repeat region on the genome due to length of the short reads. A new approach, namely, de Bruijn graph approach (Pevzner *et al.* 2001) was later on proposed to overcome the problem of repeat regions.The prominent DBG based assemblers are Velvet (Zerbino and Birney 2008), ALLPATHS (Butler *et al.* 2008), and EULER-SR (Chaisson and Pevzner 2008).

Longer reads (5-20 kb) (Ye *et al.* 2016) came into existence with the introduction of third generation sequencing technologies (Pacific Biosciences 2013; Oxford Nano Technologies 2014). Longer reads can be assembled by OLC approach with perfectly resolving most of the repeat regions. With the advent of long read sequences, now, OLC approach resolves the problem of repeat regions. However, these reads come with significantly higher error rates, to the tune of 15% in case of PacBio sequencing (Koren *et al.*

[1] Research Scholar (amit.kairi90@gmail.com); [2] Research Scholar (priyanka.814.ubkv@gmail.com); [3] Principal Scientist (ar.rao@icar.gov.in), Centre for Agricultural Bioinformatics (CABin), ICAR-IASRI, Pusa, New Delhi.

2012; Miclotte *et al.* 2016) and to the tune of 40% in case of Oxford Nanopore sequencing (Laver *et al.* 2015). These high error rates reduce the accuracy of the genome assembly.

On the other hand, the reads of "l" size are split into (l-k+1) reads of k-mer size in DBG approach. While applying on long read sequences, such splitting makes DBG approach computationally complex in terms of space and processing speed. Thus, hybrid assembly approach has been introduced, to resolve the problems prevalent in third-generation long reads as well as second generation short reads.

PacBioToCA (Au *et al.* 2012) is an error correction module in Celera Assembler originally designed to align short reads to PacBio reads and generate consensus sequences. SPAdes3.0 (Lapidus *et al.* 2013) is a short-read assembler that also allows long reads as an argument facilitating hybrid assembly approach. Whereas, Cerulean (Deshpande *et al.* 2013) starts with an assembly graph from ABySS and extends contigs by resolving bubbles in the graph using PacBio long reads. Lee *et al.* (2014) proposed a set of tools, named as ECTools, that uses contigs instead of short reads for correction and was successfully run on genomes <100 Mb. Lordec (Salmela *et al.* 2014) is essentially an error correction tool which uses highly accurate Illumina short-reads and develops succinct de Bruijn graph to map onto erroneous PacBio long reads to resolve the errors. Yet another error correction tool, Jabba (Miclotte *et al.* 2016), uses accurate Illumina short-reads to correct the PacBio long-reads by their novel pseudo alignment approach using maximal exact matches (MEM). Ye *et al.* (2016) developed *DBG2OLC*, which uses Illumina contigs as anchors to build an overlap graph with PacBio reads thereby allowing very fast assembly. The said hybrid assemblers mostly focused on error correction on long reads. However, there exists a scope to make improvement in the parallelization of assembly process. Moreover, parallelization essentially reduces the time complexity of assembly process, as contigs generation from short reads using De Bruijn Graph approach and contigs generation from long reads using Overlap Layout Approach, can be done in parallel using Message Passing Interface (MPI) processing.

Thus, the aim of the paper is to (i) develop an alternative algorithmic approach for parallelization of Hybrid *de novo* genome assembly, (ii) compare the performance of the proposed approach with the existing approaches in terms of time complexity and (iii) provide the developed hybrid assembler (hAssembler) to the users.

MATERIALS AND METHODS

Short and long read sequences

The short reads and long reads of *Arabidopsis thaliana, Bos taurus, Danio rerio* and *Oryza sativa* (IR8) were collected from Sequence Read Archive (SRA) database of National Centre for Biotechnology Information (NCBI) for the evaluation of the performance of hAssembler. The downloaded SRA IDs is given in Table 1.

Table 1 SRA IDs of short and long read sequences for four different organisms

Organism	Long read (PacBio)	Short read (Illumina)
Arabidopsis thaliana	SRR6325776	SRR7760270
Bos taurus	SRR5753568	SRR567261
Danio rerio	ERR1356691	ERR2886551
Oryza sativa (IR8)	SRR5045716	SRR869317

Source: National Centre for Biotechnology Information

A proposed algorithmic approach: hAssembler

A *de novo* genome assembly approach *hAssembler* has been developed by involving both OLC and DBG where contig libraries are generated from short reads using DBG approach and long reads using OLC approach. Both the short read and long read libraries are again mapped upon one another to generate hybrid scaffolds. Here, these repeat regions that could not be resolved using a short-read assembler were therefore, resolved by using the assembled contigs generated from the long reads using OLC approach. As well as errors in long-reads were resolved by the assembled contigs generated from DBG approach.

The algorithmic approach consists of 3 major steps:

Step 1: Generation of short read contigs using a distributed de Bruijn Graph approach.

Step 2: Generation of long read contigs using partial OLC approach.

Step 3: Generation of scaffolds using spaced suffix array.

The details of three steps of the proposed algorithm are as follows:

Step 1: Generation of short read contigs using a distributed de Bruijn Graph approach

In the proposed hAssembler, initially, Illumina reads were clustered based on number of CPUs available. Each cluster is used as input and De Bruijn Graph was run independently, and subsequently Eulerian path contigs were generated. The contigs generated from each cluster were brought together and OLC was applied once, thereby generating final contigs, which were used for further downstream hybrid assembly (Fig 1).

Step 2: Generation of long read contigs using partial OLC

OLC is one of the approaches of *de novo* genome assembly based on the overlaps between read sequences. It finds the best match between the suffix of one read and the prefix of another. Error may occur during sequencing of reads (Li *et al.* 2012). To overcome this problem, in general, fragments that do not share significantly long common substring are filtered out and multiple alignments from overlapping reads is performed. Here, overlap is performed to identify the potentially overlapping reads. Besides, layout is performed to merge to overlap reads to generate contigs.

Step 3: Generation of scaffolds using spaced suffix array

LAST (Kiełbasa *et al.* 2011) tool was used in the background to align short read contigs of step1 on long

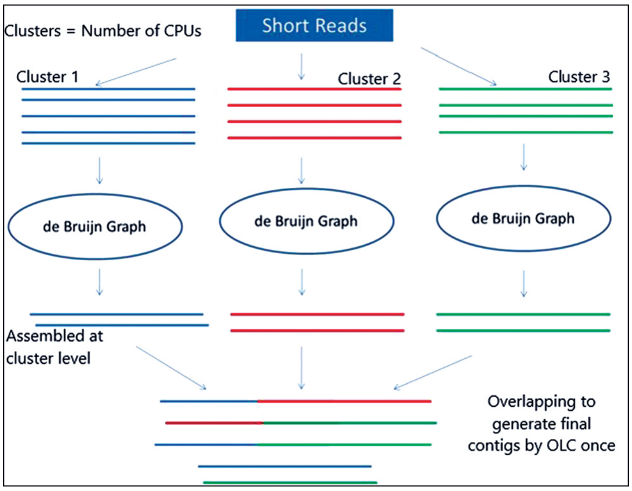
Clusters = Number of CPUs
Short Reads
Cluster 1
Cluster 2
Cluster 3
de Bruijn Graph
de Bruijn Graph
de Bruijn Graph
Assembled at cluster level
Overlapping to generate final contigs by OLC once

Fig 1 Generation of contigs from short reads using distributed de Bruijn graph. *Source:* Authors' diagram

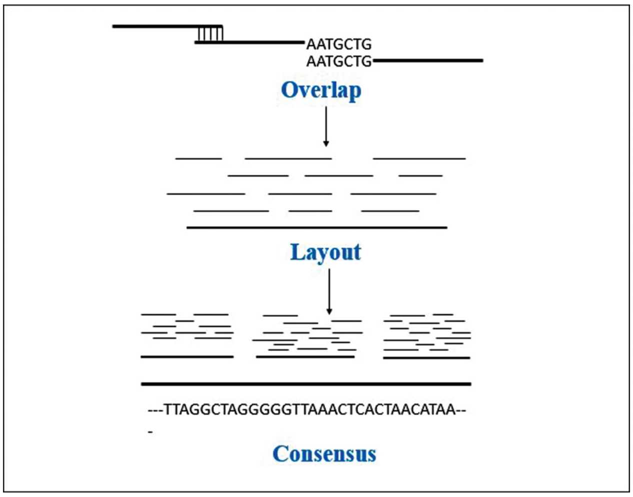

Fig 2 Generation of long-reads by partial OLC. *Source:* Authors' diagram

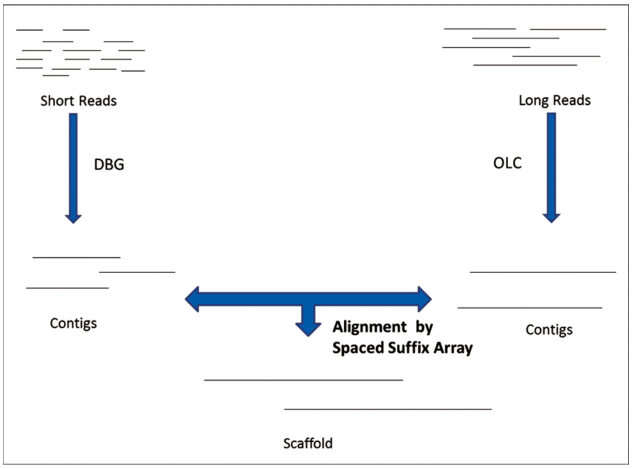

Fig 3 hAssembler flow diagram. *Source:* Authors' diagram

read contigs of step 2. LAST uses a spaced suffix array (or subset suffix array), analogous to spaced seeds (or subset seeds) and the details of suffix array are given in Supplementary Material 1. Using LAST tool, the final scaffolds are generated (Fig 3) to perform consensus of short read contigs (Step 1) on to long read contigs (Step 2) so as to generate hybrid scaffolds.

Bulges removal from short read based assembled contigs

In general, during the assembly process, when the DBG diverges in two paths (due to SNP, of repeats), bulges are generated creating ambiguities in the assembly process. hAssembler leverages the potential of hybrid assembly to replace the bulges occurred in short reads by the long reads during alignment process.

Error removal from long reads

On the other hand, ambiguities present in long reads can be replaced efficiently by short Illumina reads to generate accurate result. This is done in Step 3 where the errors present in long read contigs were removed by the short read contigs.

A pseudocode was developed to implement hAssembler consisting of the above mentioned 3 steps. The pseudocode can be shared with the users upon request through email from the authors. For easy application by the users, hAssembler is made available at http://cabgrid.res.in/cabin/hAssembler.

Assembly statistics

Most widely used two assembly statistics, namely, N50 and NG50 have been taken into consideration for the measurement of assembly statistics. N50 is a measure of the contig length or scaffold length containing a `typical' nucleotide. Specifically, it is the maximum length L such that 50% of all nucleotides lie in contigs (or scaffolds) of size at least L (Lander *et al*. 2001). On the other hand, NG50 is identical to N50, except that the length of the genome being assembled is estimated as being equal to the average of the length of the two haplotypes, α_1 and α_2 (Earl *et al.* 2011).

Assembly performance

Assembly performance indicates the percentage increase in the length of the reads before and after the assembly. Assembly performance can be calculated by using the following formula (Lee *et al.* 2014).

$$\begin{array}{l} \text{Assembly} \\ \text{Performance (\%)} \end{array} = \begin{array}{c} \text{(N50 from Assembly/N50 from} \\ \text{Chromosome Segments)} \end{array} \times 100\%$$

Computer system employed for the study

For the testing of hAssembler and to carry out the comparative study, Intel(R) Xeon(R) CPU E7-8860 v4 @ 2.20GHz system, multi-core CPU based SMP system was employed having Red Hat Enterprise Linux 7 operating system. This system consists of 288 CPUs having 18 cores per socket and 2 threads per core. The available memory of the system is ITB.

Table 2 Time taken by hAssembler for performing hybrid assembly of four eukaryotic genomes

Organism	Hybrid Assembly Long read (PacBio) + Short read (Illumina)	Time (user)
Arabidopsis thaliana	SRR6325776 + SRR7760270	15m32.373s
Bos taurus	SRR5753568 + SRR567261	6m15.915s
Danio rerio	ERR1356691 + ERR2886551	0m41.256s
Oryza sativa (IR8)	SRR5045716 + SRR869317	7m55.108s

Source: Authors' calculation

RESULTS AND DISCUSSION

The short reads and long reads collected from the public domain for the four species, *viz. Arabidopsis thaliana*, *Danio rerio*, *Bos taurus* and *Oryza sativa* (IR8) have been analyzed by the proposed hAssembler for whole genome assembly purpose. For the analysis purpose, as mentioned earlier, Intel(R) Xeon(R) CPU E7-8860 v4 @ 2.20GHz) with 1TB(1056499932 KB) memory has been used. The time taken by the hAssembler for performing genome assembly is listed in Table 2. Short reads were subjected to distributed DBG and the long reads were subjected to partial OLC defined under hybrid approach of hAssembler. The result was generated by aligning short reads and long reads using suffix array algorithm.

Table 3 Assembly statistics: N50 and NG50 for four different organisms

Organism	Est. genome size(bp)	N50	NG50
Arabidopsis thaliana	134634692	7474	9260
Bos taurus	2857605192	7980	34565
Danio rerio	1412464843	14955	17398
Oryza sativa (IR8)	466000000	10012	10000

Source: Authors' calculation

Assembly statistics

After performing hybrid assembly, a BioPython script was run to calculate the assembly statistics of the generated scaffolds. For this purpose, three assembly statistics, *viz.* N50 and NG50 defined in materials and methods were taken into consideration. The calculated statistics have been shown in Table 3. All the reads are showing high N50 as well as NG50 values exhibiting the quality of genome assembly.

Assembly performance

For the species: *Arabidopsis thaliana*, *Danio rerio*, *Bos taurus* and *Oryza sativa*-IR8, the assembly performances are calculated and given in Table 4 by using the respective formulae. It is observed that the assembly performance seems to be high after performing the hybrid assembly. Using the hAssembler, it can be observed that for the organism *Arabidopsis thaliana,* the assembly performance is significantly high followed by organism *Bos taurus, Danio rerio* and *Oryza sativa* (IR8).

To evaluate the time efficiency, hAssembler was compared with two prominent hybrid assemblers, *viz.* DBG2OLC and HybridSPAdes v 3.13.0. The time efficiency was calculated for the 4 species (*Arabidopsis thaliana*, *Danio rerio*, *Bos taurus* and *Oryza sativa*-IR8) using 3 hybrid assemblers with the same server and given in Table 5. In all the cases, hAssembler took lesser time than DBG2OLC and HybridSPAdes barring one exception.

Assembly statistics comparison (N50)

The scaffolds generated from DBG2OLC, HybridSPAdes and hAssembler were subjected to BioPython script for the

Table 4 Assembly performance on different organisms

Organism	SRA_ID (Long reads)	SRA_ID (Short reads)	N50 of long reads prior to assembly	N50 of short reads prior to assembly	N50 after hybrid assembly	% hybrid assembly performance over long reads prior to assembly	% hybrid assembly performance over short reads prior to assembly
Arabidopsis thaliana	SRR6325776	SRR7760270	3188	161	7474	234.44	4642.23
Bos taurus	SRR5753568	SRR567261	7812	75	7980	102.15	10640.00
Danio rerio	ERR1356691	ERR2886551	11726	100	14955	127.53	14955.00
Oryza sativa (IR8)	SRR5045716	SRR869317	10000	76	10012	100.12	13173.68

Source: Authors' calculation

Table 5 Time comparison of hAssembler with other hybrid assemblers

Organism	PacBio reads	Illumina reads	DBG2OLC	HybridSPAdes	hAssembler
Arabidopsis thaliana	SRR6325776	SRR7760270	17m18.81s	13m42.102s	15m52.235s
Bos taurus	SRR5753568	SRR567261	186m41.246s	181m43.894s	6m28.393s
Danio rerio	ERR1356691	ERR2886551	157m53.515s	36m13.302s	6m18.243s
Oryza sativa (IR8)	SRR5045716	SRR869317	191m12.358s	217m32.879s	8m13.159s

Source: Authors' calculation

Table 6 N50 values of hAssembler compared over different hybrid assemblers

Organism	DBG2OLC	HybridSPAdes	hAssembler
Arabidopsis thaliana	5574	934	7474
Bos taurus	9731	222	7980
Danio rerio	10093	3709	11012
Oryza sativa (IR8)	14190	---	14955

Source: Authors' calculation

purpose of computing assembly statistics. Subsequently a comparison among the N50 values over three hybrid assemblers was made. It can be observed from Table 6 that in all the cases hAssembler is giving higher N50 values than HybridSPAdes. While comparing with DBG2OLC, it can be observed that hAssembler generated higher N50 values in all the species except *Bos taurus.*

The combined analysis of short and long reads has become essential to overcome the demerits present in the application of methods/algorithms meant for assembling short and long reads sequences separately. To meet such requirement, an empirical approach involving hybrid assembly of short and long reads- hAssembler was proposed in the present study. The hAssembler to some extent has the advantage of correcting the errors present in long read assembly and filling up the gaps present in short read assembly. Besides, hAssembler consumes less time as compared to the existing hybrid assemblers as well as improves the assembly performance and N50 value. To be more specific, hAssembler integrates two major algorithms, *viz.* OLC and DBG to generate contigs and scaffolds. OLC based assemblers SSAKE (Warren *et al.* 2006), SHARCGS (Dohm *et al.* 2007), PE-Assembler (Ariyaratne *et al.* 2010) were designed earlier to assemble short read sequences (Illumina) but were unable to perform efficiently because of their inability to solve the problem of sequence repeat. On the other hand, Euler (Pevzner *et al.* 2001), ALLPATHS (Butler *et al.* 2008), Velvet (Zerbino and Birney 2008) AbySS (Simpson *et al.* 2008), SOAPdenovo (Li *et al.* 2010), PASHA (Liu *et al.* 2011), SPAdes (Bankevich *et al.* 2014), which are different variants of DBG based assemblers, could solve repeat problems to certain extent. But with the introduction of long reads with error rates the algorithmic complexity becomes significantly high while using DBG. FALCON (Chin *et al.* 2016) and Canu (Koren *et al.* 2017) that are assemblers specifically designed for long reads assembly applying OLC approach. But the inherent inaccuracies present in long reads makes the assembly eventually erroneous. These assemblers demand high coverage data which again make them less cost effective. Hybrid assemblers came resolve the problems present in long reads and short reads. PacBioToCA (Koren *et al.* 2012), LorDec (Salmela *et al.* 2014), ECTools (Lee *et al.* 2014), Jabba (Miclotte *et al.* 2016) are essentially different variants of long read (PacBio or Oxford Nanopore) error correction algorithm using short reads (Illumina). HybridSPAdes (Antipov *et al.* 2015) is another version of SPAdes which takes long reads into consideration during short read *de novo* assembly. Ye *et al.* *(*2016) developed DBG2OLC, which uses both the DBG and OLC approach to assemble genome. But for the assembly of short reads (by DBG approach) it requires another DBG based assembler and the resultant contigs are finally used for assembly purpose. However, the existing hybrid assemblers, *viz.*DBG2OLC (Ye *et al.* 2016) and Hybrid Spades (Antipov *et al.* 2015) consume more time and exhibit fair enough N50 values. Moreover, they do not consider distributed DBG, which is a concept of parallel computation. On the other hand, in the proposed assembly algorithm, short reads are given more importance while generating scaffolds because the error rates of short reads are much lower than that of long reads. Whereas, longer reads in hybrid assemble has been mainly used for gap fillings and anchoring purpose to increase the length of the scaffolds. Hence, the % assembly performance of hybrid assembly is much higher and observed to have a significant increase over N50 of short reads as compared to long reads (Table 4). Further, hAssembler is taking less time to assemble reads as compared to HybridSPAdes and DBG2OLC (Table 5). Also, hAssembler is showing high N50 value as compared to HybridSPAdes and DBG2OLC (Table 6). Further, hAssembler considers the distributed DBG and tackles the problems arising from Single Nucleotide Polymorphisms, sequencing errors and repeats in an efficient and timely manner. Thus, the assembly performance of hAssembler is found to be higher than other assemblers in most of the datasets considered under the study. The reason could be the poor quality of data in case of long reads of *Bos taurus* available in public domain. Thus, hAssembler is going to supplement the existing hybrid assembler and of high use to the application users of genomic research.

ACKNOWLEDGMENT

The study is part of the Ph D research of the first author. The first author acknowledges the Post Graduate School, ICAR-IARI, New Delhi and ICAR-IASRI, New Delhi, for providing the necessary facilities for the research study.

REFERENCES

Au K F, Underwood J G, Lee L and Wong W H. 2012. Improving PacBio long read accuracy by short read alignment. *PloS One* **7**(10): e46679.

Batzoglou S, Jaffe D B, Stanley K, Butler J, Gnerre S, Mauceli E and Lander E S. 2002. ARACHNE: a whole-genome shotgun assembler. *Genome Research* **12**(1): 177-189.

Butler J, MacCallum I, Kleber M, Shlyakhter I A, Belmonte M K, Lander E S and Jaffe D B. 2008. ALLPATHS: *de novo* assembly of whole-genome shotgun microreads. *Genome Research* **18**(5):810-820.

Compeau P E, Pevzner P A and Tesler G. 2011. How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology* **29**(11): 987.

Denisov G, Walenz B, Halpern A L, Miller J, Axelrod N, Levy S and Sutton G. 2008. Consensus generation and variant detection by Celera Assembler. *Bioinformatics* **24**(8): 1035-1040.

Deshpande V, Fung E D, Pham S and Bafna V. 2013. Cerulean: A hybrid assembly using high throughput short and long reads.

(In) *International workshop on algorithms in bioinformatics*, Springer, Berlin, Heidelberg, pp 349-363.

Earl D, Bradnam K, St John J, Darling A, Lin D, Fass J and Paten B. 2011. Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Research* **21**(12): 2224–2241. doi:10.1101/gr.126599.111

Gordon D, Huddleston J, Chaisson M J, Hill C M, Kronenberg Z N, Munson K M and Dunn C. 2016. Long-read sequence assembly of the gorilla genome. *Science* **352**(6281): aae0344.

Green P. 1994. Phrap. (http://www.genome.washington.edu/UWGC/analysistools/phrap.htm).

Huang X and Madan A. 1999. CAP3: A DNA sequence assembly program. *Genome Rresearch* **9**(9): 868-877.

Kiełbasa S M, Wan R, Sato K, Horton P and Frith M C. 2011. Adaptive seeds tame genomic sequence comparison. *Genome Research* **21**(3): 487-493.

Koren S, Schatz M C, Walenz B P, Martin J, Howard J T, Ganapathy G and Phillippy A M. 2012. Hybrid error correction and *de novo* assembly of single-molecule sequencing reads. *Nature Biotechnology* **30**(7): 693.

Lander E S, Linton L M, Birren B, Nusbaum C, Zody M C, Baldwin J, Devon K, Dewar K, Doyle M, FitzHugh W, Funke R *et al.* 2001. Initial sequencing and analysis of the human genome. *Nature* 409 (6822): 860-921.

Lapidus A, Antipov D, Bankevich A, Gurevich A, Korobeynikov A, Nurk S, Prjibelski A, Safonova Y, Vasilinetc I, Pevzner P A. 2013. New Frontiers of Genome Assembly with SPAdes 3.0. (poster).

Laver T, Harrison J, O'neill P A, Moore K, Farbos A, Paszkiewicz K, and Studholme D J. 2015. Assessing the performance of the Oxford nanopore technologies MinION. *Biomolecular Detection and Quantification* **3**: 1-8.

Lee H, Gurtowski J, Yoo S, Marcus S, McCombie W R and Schatz M. 2014. Error correction and assembly complexity of single molecule sequencing reads. *BioRxiv*: 006395.

Miclotte G, Heydari M, Demeester P, Rombauts S, Van de Peer Y, Audenaert P and Fostier J. 2016. Jabba: hybrid error correction for long sequencing reads. *Algorithms for Molecular Biology* **11**(1):10.

Mullikin J C and Ning Z. 2003. The phusion assembler. *Genome Rresearch* **13**(1): 81-90.

Paul H, Chen R, Durbin K J, Egan A, Ren Y, Song X, Weinstock G M and Gibbs R A. 2004. The Atlas genome assembly system. *Genome Research* **14**(4): 721-732.

Pop M, Phillippy A, Delcher A L and Salzberg S L. 2004. Comparative genome assembly. *Briefings in Bioinformatics* **5**(3): 237-248.

Salmela L and Rivals E. 2014. LoRDEC: accurate and efficient long read error correction. *Bioinformatics*, **30**(24): 3506-3514.

Wang J, Wong G K S, Ni P, Han Y, Huang X, Zhang J and Xu X. 2002. RePS: a sequence assembler that masks exact repeats identified from the shotgun data. *Genome Research* **12**(5): 824-831.

Ye C, Hill C M, Wu S, Ruan J, and Ma Z S. 2016. DBG2OLC: efficient assembly of large genomes using long erroneous reads of the third-generation sequencing technologies. *Scientific Reports* **6** : 31900.

Zerbino D and Birney E. 2008. Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome Research* **18**(5):821-829.