


PGcloser: Fast Parallel Gap-Closing Tool Using Long-Reads or Contigs to Fill Gaps in Genomes

Peng Lu¹ , Jingjing Jin¹, Zefeng Li¹, Yalong Xu¹, Dasha Hu², Jiajun Liu² and Peijian Cao¹

¹China Tobacco Gene Research Center, Zhengzhou Tobacco Research Institute of CNTC, Zhengzhou, China. ²Department of Computer Science and Technology, Sichuan University, Chengdu, China.

Evolutionary Bioinformatics
Volume 16: 1–8
© The Author(s) 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1176934320913859



ABSTRACT: Assembled draft genomes usually contain many gaps because of the length limit of next-generation sequencing. Although many gap-closing tools have been developed, most of them still attempt to fill gaps on the basis of next-generation sequencing reads (always < 200 bp). Hence, the gap-filling effect is inferior. Several tools that use long-reads to close gaps have recently been created. However, they require extensive runtimes, which may not be suitable for large genomes. We describe a gap-closing tool called PGcloser, which supports parallel mode and adopts long-reads/contigs to fill gaps in genome sequences. Three tests show that PGcloser is faster than other tools but exhibits similar accuracy. PGcloser is free open-source software that is available at <http://software.tobaccodb.org/software/pgcloser>.

KEYWORDS: Parallel, long-reads, gap-closing

RECEIVED: October 10, 2019. **ACCEPTED:** February 25, 2020.

TYPE: Parallel Computing in Evolutionary Bioinformatics—Original Research

FUNDING: The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Projects of Gap-Closing in the Tobacco Genome (902016CA0170), Tobacco Breeding Big Data (110201901024(SJ-03)), ENCODE of Tobacco Genome (110201601033(JY-07)), and the Young Elite Scientists Sponsorship Program by CAST (2016QNRC001).

DECLARATION OF CONFLICTING INTERESTS: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

CORRESPONDING AUTHOR: Peijian Cao, China Tobacco Gene Research Center, Zhengzhou Tobacco Research Institute of CNTC, Zhengzhou 450001, China. Email: peijiancao@163.com

Introduction

Genes are DNA molecular fragments with genetic effects. They are the code of life and record and transmit genetic information. Genetic research has explored biological inheritance and variation.¹ Genetic studies were based on gene sequences that are mainly obtained by genome sequencing. More than 800 kinds of bacteria and more than 100 kinds of eukaryotic organisms' genome sequences have been released after the introduction of the Sanger sequencing method in 1977. Thousands of genome sequences have been restored in gene databases, and a large number of species are still being sequenced.

The main sequencing technology that has been applied in recent years is called next-generation sequencing (NGS) or second-generation sequencing. Reads generated through this method have short sequences, high coverage rates, and paired-end information. Numerous algorithms have been introduced to assemble reads using the overlap between the sequences of fragments,² such as greedy extension algorithm, overlap-layout-consensus algorithm, and De Bruijn graph (DBG) algorithm. And a lot of software including Velvet,³ SOAPdenovo,⁴ AbySS⁵ are developed based on these algorithms. However, given the length limitations of next-generation sequencing reads and the high ratio of repeat sequences in genomes, some regions of genome sequences have not been assembled, all these factors make some regions difficult or impossible to assemble, leading to gaps and fragmented genome assemblies.⁶ Consequently, the final draft genomes contain many gaps.

The public dataset in National Center for Biotechnology Information (NCBI) (Table 1[June 2018]) showed that only a few species have complete genomes, and most of them are viruses and bacteria. The large genomes of certain plants and animals remain draft genomes at the contig/scaffold level. Thus, they may still contain many gaps.

The gaps in scaffolds may contain a considerable amount of useful biological information, such as important genes. Hence, filling the gaps may result in the acquisition of the unknown information, thereby improving the integrity of gene sequences. At present, gaps are filled using five major methods, as follows: (1) assembly by multiple software, such as Velvet³ and Edena⁷; (2) use of reference genomes from closely related species, such as software Velvet,³ Edena,⁷ and MUMer⁷; (3) assembly by multiple types of data, such as ALLPATHS-LG⁸ and SSPAKE⁹; (4) use of polymerase chain reaction amplification at the ends of gaps, such as ABACAS¹⁰; and (5) adoption of improved assembly methods based on DBG, such as GapCloser,¹¹ IMAGE,¹² and GapFiller.¹³

Third-generation sequencing technology, including PacBio's SMRT sequencing technology¹⁴ and Oxford's single-molecule nanopore sequencing technology,¹⁵ have recently been applied to biological genome sequencing. Compared with the read length of next-generation sequencing, that of third-generation sequencing is considerably longer, possibly exceeding 10 kb¹⁶; it is also hundreds of times longer than that of NGS. Several gap-closing tools have been developed using the long-reads instead of short NGS reads, such as LR_Gapcloser,⁶ GMcloser,¹⁷ PBjelly,¹⁸ and FGAP¹⁹. PBjelly and FGAP primarily focus on



Table 1. Genome assembly of various species.

GROUP	SPECIES NUMBER	ASSEMBLY LEVEL							
		SNC	RC (%)	SNS	RS (%)	SNCHR	RCHR (%)	SNCG	RCG (%)
Animal	195	30	16.20	117	59.78	48	24.02	0	0
Plant	70	8	14.04	40	56.14	20	26.32	2	3.51
Fungi	266	43	15.38	215	81.00	5	2.26	3	1.36
Bacteria	11362	4520	39.78	5742	49.41	211	2.18	889	8.64
Virus	4 663	0	0	1	0.02	25	0.37	4637	99.61

RC: ratio of contig assembly level in the group; RCG: ratio of complete genome assembly level in the group; RChr: ratio of chromosome assembly level in the group; RS: ratio of scaffold assembly level in the group; SNC: species number of contig assembly level; SNCG: species number of complete genome assembly level; SNChr: species number of chromosome assembly level; SNS: species number of scaffold assembly level.

assembly extension and not gap closing. GMcloser was developed to close gaps by measuring the likelihood ratios of true alignments. Its accuracy is 3-fold to 100-fold higher than that of other available tools that use NGS data. LR_Gapcloser utilizes long reads generated from TGS sequencing platforms. It closes gaps more rapidly with a lower error rate and a considerably lower memory usage than GMcloser. However, LR_Gapcloser and GMcloser have certain limitations. They cannot be run in multiple nodes. Thus, they are slow for large genomes. Moreover, they require the use of a large memory for large genomes and reads.

Here, we developed PGcloser to efficiently and rapidly close gaps in assemblies using long reads or contigs. Compared with the abovementioned gap-closing tools, PGcloser has advantages in runtime, average memory usage, and efficiency.

Methods

The main idea of PGcloser is to reduce the amount of computing data and increase running speed. Thus, we split a genome into small sub-files for parallel computation and then used the error-corrected and repeats-removed long reads or contigs to minimize the number of sequences in the reference reads. The pipeline for PGcloser is shown in Figure 1.

PGcloser is a Linux-based integrative analysis workflow. This tool contains the following steps:

1. It extracts the gaps for each scaffold in the genome file. Then, two ends of each gap are extracted as anchors for each scaffold.
2. The long reads are regarded as reference, and an index file is built for this reference. Then, anchor sequences are mapped to the long reads.
3. PGcloser analyzes the mapping results and obtains a specified mapping position for each anchor sequence. If two anchors map to the same sequence in the reference with reasonable distance, close this gap. If one or two anchors map to different sequences in the reference, then the corresponding gap is extended.

4. All gap-closing results are combined, and the final gap-closing genome is produced.

Step 1: the genome is split into small sub-files, and the gaps are extracted

PGcloser splits the genome file into sub-files by the number of threads provided by the user. Then, it extracts the gaps for each scaffold, and two ends of each gap are used as anchors. The corresponding parameters, including the minimum gap length, anchor length, and thread number, are provided by the user.

Step 2: the anchor sequence is aligned to the long reads

PGcloser regards the long reads/contigs as the reference and builds an index for it. Then, the anchor sequences generated in Step 1 are mapped to the index file. The mapping result is then stored in SAM/BAM format.

Step 3: the gaps are filled or extended

The mapping result generated in Step 2 can be divided into three categories, as follows:

1. Two anchors mapped to the same read. In this mapping process, we defined two filter criteria. First, two anchors map to the same read in the reference. And another is the ratio of the gap length in the reference sequence to the gap length in the genome (error ratio) is within a reasonable range (this parameter can be submitted by the user). By these two filter criteria, the anchors mapped to repetitive regions will be greatly reduced after filtering. For the last repetitive regions, we will keep all the possible mapping reads.
2. Two anchors mapped to two different reads or just one anchor mapped to one read. If two anchors map to different reads or just one anchor maps to one read in the

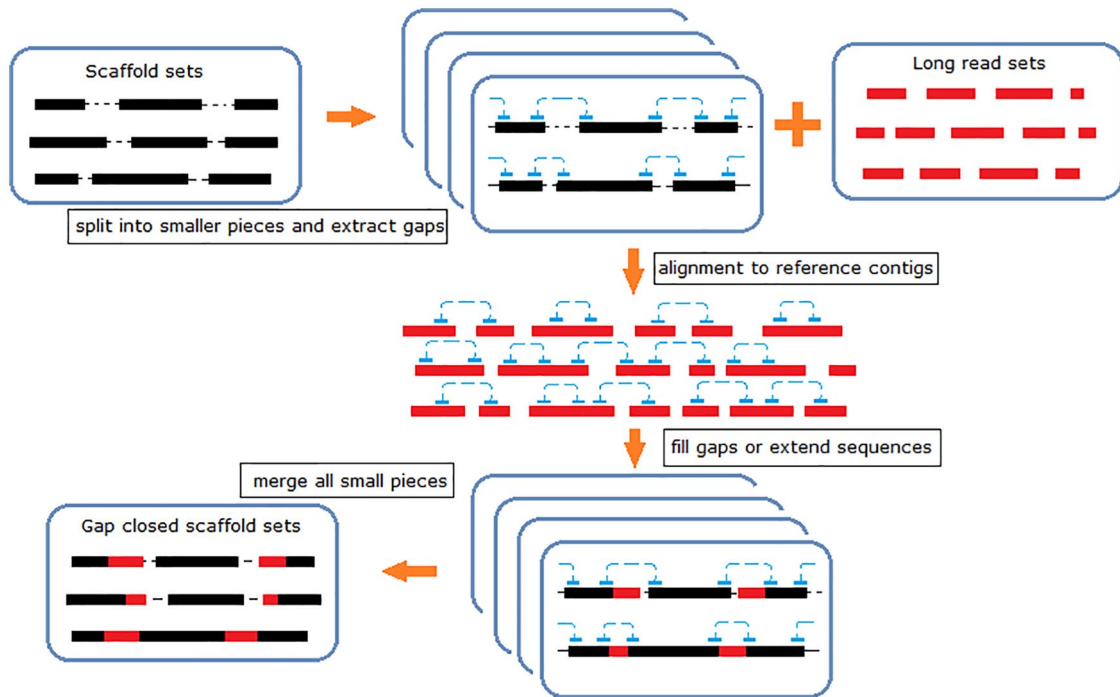


Figure 1. Pipeline of PGcloser.

reference, PGcloser will extend the corresponding gap related to these anchors.

- Two anchors are not mapped to any read. If two anchors do not map to any reference read, PGcloser will continue to keep this gap.

Step 4: all sub-files are merged

In this last step, PGcloser combines the processed gaps and forms the final result.

Implementation. In accordance with the abovementioned algorithm, we developed the gap-closing tool PGcloser, which contains the following modules: SplitFa, ExtrGap, BwtBuilt, CompGap, ClsGap, MergFa, and GetCls.

- SplitFa:** This module splits the genomic file into small sub-files in accordance with the number of threads and nodes provided by the user.
- ExtrGap:** This module extracts all the gaps of the sub-files in multiple threads and nodes. Then, it generates a fasta file of the anchor sequences.
- BwtBuilt:** This module constructs the index file for the input long reads set.
- CompGap:** This module aligns the anchor sequences of the gaps to the index file and generates SAM files for the next module.
- ClsGap:** This module extracts the specified position sequence for the anchor sequences and deals with gaps in accordance with their mapping result.
- MergFa:** This module merges the closed and extended gaps to form the final result.

- GetCls:** This module executes all the modules in PGcloser and outputs the GapClosed.fa file.

Result

Datasets

To test the performance of PGcloser, we selected three datasets with different genome sizes from *Arabidopsis thaliana*, *Oryza sativa*, and *Homo sapiens* (Table 2), more data description can be visited and downloaded from the website <http://software.tobacodb.org/software/pgcloser>.

Benchmark

We compared the performance, efficiency, and genome evaluation of PGcloser, LR_Gapcloser, GMcloser, GapFiller, and FGAP on a 48-core server with Intel(R) Xeon(R) Gold 6126 CPU @ 2.60 GHz and 512 GB RAM.

For performance, we compared speed and memory usage. For efficiency, we compared the gap-closing ratio and gap length reduction. For genome evaluation after gap-closing, we compared BUSCO and reads mapping rate.

With m as the gap number before gap-closing and n as the gap number after gap-closing, the close rate is defined as follows:

$$\text{Gap closing rate} = \frac{n - m}{n} * 100\%$$

With x as the gap length before gap-closing and y as the gap length after gap-closing, the gap length reduction is defined as follows:

$$\text{Gap length reduction} = \frac{x - y}{x} * 100\%$$

Table 2. Information for three datasets.

GENOME DATA DESCRIPTION				
SPECIES	GAP NUMBER	GAP LENGTH	GENOME SIZE	DESCRIPTION
<i>A. thaliana</i>	31155	6.3 Mb	96.5 Mb	N50:13138, L50:1874
<i>O. sativa</i>	30953	16.9 Mb	391.1 Mb	N50: 11163166, L50:5249
<i>H. sapiens</i>	220318	171.4 Mb	2615.0 Mb	N50: 23924, L50: 30971
CONTIG AND READS DATA DESCRIPTION				
SPECIES	FILE TYPE	FILE SIZE	DESCRIPTION	
<i>A. thaliana</i>	Contig	127.4 Mb	N50: 11163166, L50:5	
	Reads	3.4 Gb	PacBio, 35×, Corrected Reads	
	Reads	4.8 Gb	Oxford Nanopore, 50×	
	Reads	5.8 Gb	Illumina NextSeq 500, 55×, Paired-end	
<i>O. sativa</i>	Contig	418.9 Mb	N50: 2522746, L50:52	
	Reads	8.4 Gb	PacBio, 20×, Corrected Reads	
	Reads	12.4Gb	Oxford Nanopore, 30×	
	Reads	18.1 Gb	Illumina NovaSeq 6000, 45×, Paired-end	
<i>H. sapiens</i>	Contig	2.94 Gb	N50: 20609304, L50: 40	
	Reads	145 Gb	PacBio, 50×, Corrected Reads	
	Reads	186Gb	Oxford Nanopore, 60×	
	Reads	119.0 Gb	Illumina NovaSeq 6000, 30×, Paired-end	

Table 3. Gap-closing results in gene regions.

SPECIES	TOOLS	NO. OF CLOSED GAPS	CLOSED RATE	NO. OF ACCURATE CLOSED ^a GAPS	ACCURATE CLOSED RATE
<i>A. thaliana</i>	PGcloser	542	54.2%	518	95.57%
	LR_Gapcloser	511	51.1%	491	96.09%
	GMcloser	538	53.8%	519	96.47%
<i>O. sativa</i>	PGcloser	302	30.2%	286	94.71%
	LR_Gapcloser	411	41.1%	383	93.32%
	GMcloser	287	28.7%	271	94.42%
<i>H. sapiens</i> (Ch01)	PGcloser	672	67.2%	631	93.90%
	LR_Gapcloser	629	62.9%	587	94.34%
	GMcloser	437	43.7%	398	91.08%

^aAccurate closed—sequence difference before and after gap closing < 5%.

Artificial data test

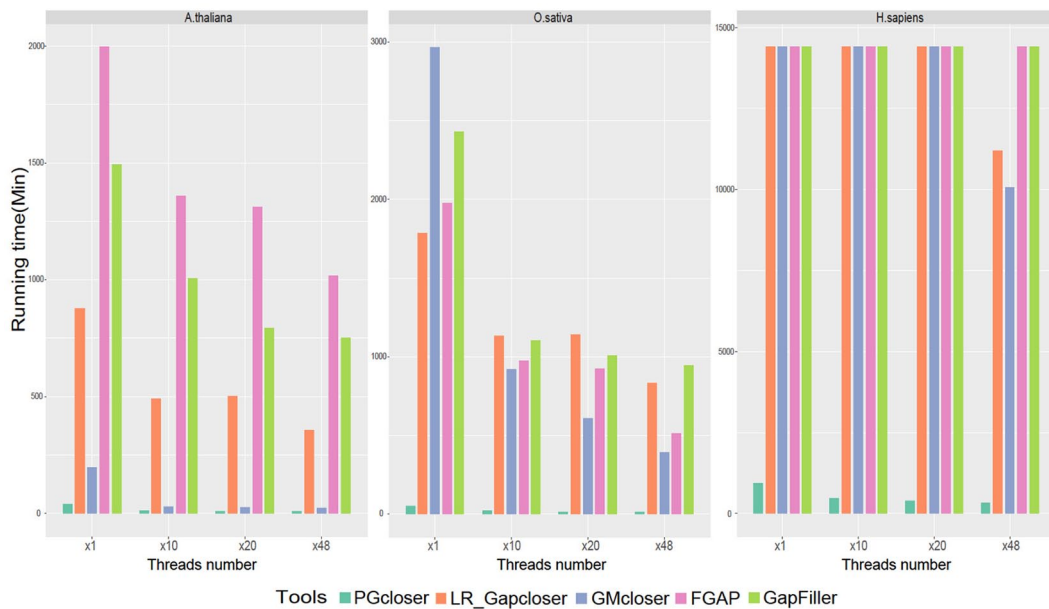
In order to initially evaluate the gap-closing quality of PGcloser, we have created 2000 artificial gaps with length from 100 to 300bp in three genomes. Half of them were inserted in gene

regions, and others were inserted into repetitive elements. Then, we compared the filling sequence of the gaps by the gap-closing tools with the original sequence. The results are as follows.

Tables 3 and 4 show the results of artificial gap-closing. For the closed gaps, the accuracy rate of PGcloser in genes is

Table 4. Gap-closing results in repetitive elements regions.

SPECIES	TOOLS	NO. OF CLOSED GAPS	CLOSED RATE	NO. OF ACCURATE CLOSED GAPS	ACCURATE CLOSED RATE
<i>A. thaliana</i>	PGcloser	227	22.7%	117	51.54%
	LR_Gapcloser	282	28.2%	169	59.92%
	GMcloser	203	20.3%	96	47.29%
<i>O. sativa</i>	PGcloser	166	16.6%	89	53.61%
	LR_Gapcloser	254	25.4%	108	42.52%
	GMcloser	171	17.1%	94	54.97%
<i>H. sapiens</i> (Ch01)	PGcloser	338	33.8%	243	71.89%
	LR_Gapcloser	469	46.9%	295	62.90%
	GMcloser	181	18.1%	159	87.85%

**Figure 2.** Running time (min).

around 95% and in repetitive elements is more than 50% on three test datasets, which are similar to other tools.

The results of the artificial gap test showed, that the quality of gap-closing using PGcloser is reliable.

Main content

Performance comparison. LR_Gapcloser, GMcloser, FGAP, and GapFiller tools cannot run in multiple nodes. Thus, we tested them with different thread numbers (from 1 to 48) in one node. We also performed an additional test using PGcloser in multiple nodes. The performance results between these tools are as follows:

The time consumption of LR_Gapcloser, GMcloser, GapFiller, and FGAP is too long. Hence, we stopped the processes after 10 days. GapFiller and FGAP failed to obtain

gap-closing results on the *H. sapiens* dataset with all threads. Figures 2 and 3 show that PGcloser outperforms the other tools in terms of running time and memory usage. Compared with the other tools, PGcloser achieved a 10-fold to 100-fold faster running time, especially in the multiple thread mode. Memory usage decreased by more than 90% using *A. thaliana* and *O. sativa*.

Efficiency comparison. The efficiency for gap closing is as follows. Table 5 shows the results for the three datasets. PGcloser has a similar gap-closing number compared with most of the other tools, including the relatively stable tools LR_Gapcloser and GMcloser. FGAP only has superior performance for *A. thaliana*, which has a small genome. As for the closing ratio for the gaps, PGcloser, GMcloser, and LR_Gapcloser outperform GapFiller and FGAP, which respectively can achieve 60% and 90% for *H. sapiens*, which has a large genome.

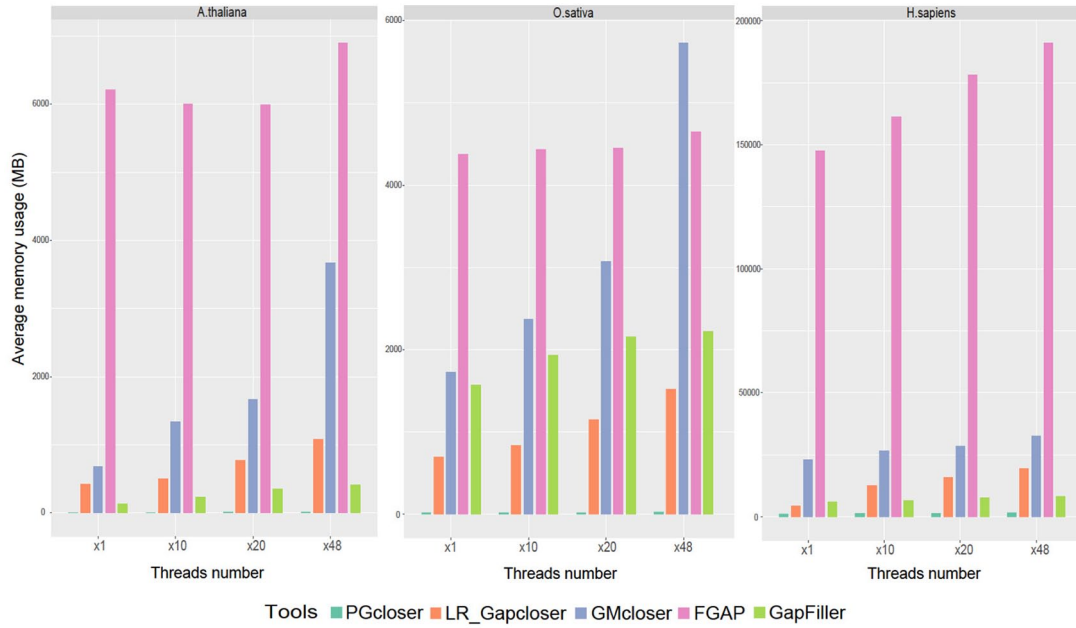


Figure 3. Average memory usage (Mb).

Table 5. Efficiency results.

SPECIES	TOOLS	CLOSED GAP NUMBER	GAP-CLOSING RATE	TOTAL GAP LENGTH AFTER CLOSING	GAP LENGTH REDUCTION
<i>A. thaliana</i>	PGcloser	9770	31.36%	2.52Mb	59.68%
	LR_Gapcloser	8530	27.56%	2.57 Mb	58.88%
	GMcloser	9846	31.60%	2.68 Mb	57.12%
	GapFiller	3486	10.12%	6.14 Mb	1.76%
	FGAP	19740	63.36%	5.34 Mb	18.40%
<i>O. sativa</i>	PGcloser	4048	13.08%	4.31 Mb	74.54%
	LR_Gapcloser	8483	27.23%	10.11 Mb	40.28%
	GMcloser	3959	12.79%	5.96 Mb	64.79%
	GapFiller	1118	3.61%	16.66 Mb	1.59%
	FGAP	4544	14.68%	15.88 Mb	6.20%
<i>H. sapiens</i>	PGcloser	142216	64.55%	11.29 Mb	93.41%
	LR_Gapcloser	174494	79.20%	13.76 Mb	91.97%
	GMcloser	6345	2.88%	159.59 Mb	6.86%
	GapFiller	N/A	N/A	N/A	N/A
	FGAP	N/A	N/A	N/A	N/A

N/A, no data.

Genome evaluation. We used two methods to evaluate the quality of genomes after gap-closing. One is showing BUSCOs before and after gap-closing by running BUSCO.²⁰ Another is showing the long reads mapping rates before and after gap-closing by running Minimap2.²¹ More details are as follows.

Table 6 shows the results of BUSCOs before and after genome gap-closing. The BUSCO result of PGcloser is similar to other tools on the three datasets.

Table 7 shows the results of reads mapping rate before and after gap-closing. The mapped rates had increased after gap-closing for *A. thaliana*. Although reads mapping rate is

Table 6. BUSCO results.

SPECIES	GENOME NAME	COMPLETE BUSCOS	FRAGMENTED BUSCOS	MISSING BUSCOS
<i>A. thaliana</i>	Original genome ^a	85.5%	2.7%	11.8%
	PGcloser	87.9%	2.7%	9.4%
	LR_Gapcloser	89.4%	2.0%	8.6%
	GMcloser	85.9%	2.4%	11.7%
	GapFiller	87.0%	2.0%	11.0%
	FGAP	85.9%	2.0%	12.1%
<i>O. sativa</i>	Original genome	85.5%	2.7%	11.8%
	PGcloser	90.2%	2.4%	7.4%
	LR_Gapcloser	90.2%	2.4%	7.4%
	GMcloser	90.2%	2.4%	7.4%
	GapFiller	89.0%	2.7%	8.3%
	FGAP	89.8%	2.7%	7.5%
<i>H. sapiens</i>	Original genome	56.5%	5.5%	38.0%
	PGcloser	58.1%	4.7%	37.2%
	LR_Gapcloser	57.7%	4.7%	37.6%
	GMcloser	56.5%	5.5%	38.0%
	GapFiller	N/A	N/A	N/A
	FGAP	N/A	N/A	N/A

^aOriginal genome, genome before gap-closing.

Table 7. Reads mapping rate results.

SPECIES	GENOME NAME	QC-PASSED READS	MAPPED READS	MAPPED RATE
<i>A. thaliana</i>	Original genome	743559	623602	83.87%
	PGcloser	836245	782972	93.63%
	LR_Gapcloser	1046186	1004096	95.98%
	GMcloser	743537	623580	83.88%
	GapFiller	746345	652712	87.45%
	FGAP	795832	721563	90.67%
<i>O. sativa</i>	Original genome	1332536	1332236	99.98%
	PGcloser	1359739	1359467	99.98%
	LR_Gapcloser	1384577	1384301	99.98%
	GMcloser	1313879	1313581	99.98%
	GapFiller	1333362	1333060	99.98%
	FGAP	1332271	1331971	99.98%
<i>H. sapiens</i>	Original genome	15309295	15297909	99.93%
	PGcloser	15323632	15312905	99.93%
	LR_Gapcloser	15374113	15363351	99.93%
	GMcloser	15191096	15179958	99.93%
	GapFiller	N/A	N/A	N/A
	FGAP	N/A	N/A	N/A

same for *O. sativa* and *H. sapiens*, the number of mapped reads had increased after gap-closing. The reads mapping result of PGcloser is similar to other tools on the three datasets.

The evaluation results of the above two methods showed that the quality of three genomes has improved after gap closing. And the quality of the genome after gap-closing by PGcloser is similar as the quality obtained with other tools.

Conclusion

We compared the gap-closure performance of PGcloser and four currently available tools: LR_Gapcloser,⁶ GMcloser,¹⁷ GapFiller,¹³ and FGAP.¹⁹ We ran each tool with 1, 10, 20, and 48 threads on the same machine. We estimated the performance of each tool using runtime and average memory usage, approximated the efficiency using the gap closing rate and gap length reduction, and then evaluated the quality of genome after gap-closing by BUSCOs and reads mapped rates.

The results of the three datasets showed that PGcloser reduced the running time and memory usage compared with the other tools. PGcloser was considerably faster and had similar or even better efficiency than the other tools. PGcloser showed a bigger advantage than the other approaches, especially for large genomes.

Author Contributions

PL, JJ, and PC designed the project. PL, ZL, and YX collected sample data used in this project. PL, JJ, DH, and JL designed the algorithm. PL and JJ wrote the program codes and performed the bioinformatic analyses. PL, JJ, and PC wrote the manuscript. PL and JJ contributed equally to the study.

Availability and Requirements

Project name: PGcloser

Project home page: <https://software.tobacodb.org/software/PGcloser>

Operating system: Linux

Cluster platform: LSF

Programming language: C++

Other requirements: bowtie2, bedtools2

License: GNU GPLv2

Any restrictions against use by non-academics: None

ORCID iD

Peng Lu  <https://orcid.org/0000-0002-9898-3951>

REFERENCES

1. Pop M. Genome assembly reborn: recent computational challenges. *Brief Bioinform.* 2009;10:354-366.
2. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. *Genomics.* 2010;95:315-327.
3. Zerbino DR, Birney E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 2008;18:821-829.
4. Li RQ, Zhu HM, Ruan J, et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.* 2010;20:265-272.
5. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJ, Birol I. ABySS: a parallel assembler for short read sequence data. *Genome Res.* 2009;19:1117-1123.
6. Xu G-C, Xu T-J, Zhu R, et al. LR_Gapcloser: a tiling path-based gap closer that uses long reads to complete genome assembly. *Gigascience.* 2019;8:giy157.
7. Kusuma WA, Ishida T, Akiyama U. A combined approach for de novo DNA sequence assembly of very short reads. *IPSP Trans Bioinform.* 2011;4:21-33.
8. Ribeiro FJ, Przybylski D, Yin SY, et al. Finished bacterial genomes from shotgun sequence data. *Genome Res.* 2012;22:2270-2277.
9. Diguistini S, Liao NY, Platt D, et al. De novo genome sequence assembly of a filamentous fungus using Sanger, 454 and Illumina sequence data. *Genome Biol.* 2009;10:R94.
10. Assefa S, Keane TM, Otto TD, Newbold C, Berriman M. ABACAS: algorithm-based automatic contiguation of assembled sequences. *Bioinformatics.* 2009;25:1968-1969.
11. Li R, Fan W, Tian G, et al. The sequence and de novo assembly of the giant panda genome. *Nature.* 2010;463:311-317.
12. Tsai IJ, Otto TD, Berriman M. Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. *Genome Biol.* 2010;11:R41.
13. Boetzer M, Pirovano W. Toward almost closed genomes with gapfiller. *Genome Biol.* 2012;13:R56.
14. Pacific Biosciences of California, Inc. <http://www.pacb.com/smrt-science/smrt-sequencing/>.
15. Clarke J, Wu HC, Jayasinghe L, Patel A, Reid S, Bayley H. Continuous base identification for single-molecule nanopore DNA sequencing. *Nat Nanotechnol.* 2009;4:265-270.
16. Rhoads A, Au K. PacBio sequencing and its applications. *Genomics Proteomics Bioinformatics.* 2015;13:278-289.
17. Kosugi S, Hirakawa H, Tabata S. GMcloser: closing gaps in assemblies accurately with a likelihood-based selection of contig or long-read alignments. *Bioinformatics.* 2015;31:3733-3741.
18. English AC, Richards S, Han Y, et al. Mind the gap: upgrading genomes with Pacific Biosciences RS long-read sequencing technology. *PLoS ONE.* 2012;7:e47768.
19. Piro VC, Faoro H, Weiss VA, et al. FGAP: an automated gap closing tool. *BMC Res Notes.* 2014;7:371.
20. Seppy M, Manni M, Zdobnov EM. BUSCO: assessing genome assembly and annotation completeness. *Methods Mol Biol.* 2019;1962:227-245.
21. Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics.* 2018;34:3094-3100.