Bioinformatics, 31(23), 2015, 3733–3741
doi: 10.1093/bioinformatics/btv465
Advance Access Publication Date: 10 August 2015
Original Paper

Genome analysis

GMcloser: closing gaps in assemblies accurately with a likelihood-based selection of contig or long-read alignments

Shunichi Kosugi*,†, Hideki Hirakawa and Satoshi Tabata

Department of Technology Development, Kazusa DNA Research Institute, Kisarazu, Chiba 292-0818, Japan

*To whom correspondence should be addressed.
†Present address: Center for Integrated Medical Sciences, RIKEN, Tsurumi-ku, Yokohama 230-0045, Japan

Associate Editor: Inanc Birol

Received on January 6, 2015; revised on July 31, 2015; accepted on August 4, 2015

Abstract

Motivation: Genome assemblies generated with next-generation sequencing (NGS) reads usually contain a number of gaps. Several tools have recently been developed to close the gaps in these assemblies with NGS reads. Although these gap-closing tools efficiently close the gaps, they entail a high rate of misassembly at gap-closing sites.

Results: We have found that the assembly error rates caused by these tools are 20–500-fold higher than the rate of errors introduced into contigs by *de novo* assemblers. We here describe GMcloser, a tool that accurately closes these gaps with a preassembled contig set or a long read set (i.e. error-corrected PacBio reads). GMcloser uses likelihood-based classifiers calculated from the alignment statistics between scaffolds, contigs and paired-end reads to correctly assign contigs or long reads to gap regions of scaffolds, thereby achieving accurate and efficient gap closure. We demonstrate with sequencing data from various organisms that the gap-closing accuracy of GMcloser is 3–100-fold higher than those of other available tools, with similar efficiency.

Availability and implementation: GMcloser and an accompanying tool (GMvalue) for evaluating the assembly and correcting misassemblies except SNPs and short indels in the assembly are available at https://sourceforge.net/projects/gmcloser/.

Contact: shunichi.kosugi@riken.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Next-generation sequencing (NGS) technologies allow the low-cost and high-speed construction of unknown genome sequences by *de novo* assembly. The assembly of reads produces contigs, but the extension of these contigs frequently stops at sites with repetitive regions, heterozygous alleles, sequencing errors or low read coverage, causing assembly fragmentation (Miller *et al.*, 2010; Treangen and Salzberg, 2012). Hence, for large eukaryotic genomes with highly repetitive and/or polymorphic sequences, the lengths of the contigs generated by short read assembly are generally short. Contig fragments are then further linked with paired-end (PE) or mate-pair (MP) reads, which are generated by sequencing both ends of DNA fragments with a range of sizes, to form scaffolds. Scaffolds contain many gaps, with a range of sizes, which are estimated from the fragment sizes of PE or MP reads.

Closing the gaps in scaffolds is an important finishing process in assembly construction. Because the gap regions correspond not only to repeat sequences, which are often found in intergenic regions, but also to gene-encoding regions, which could not be assembled because of low read coverage or heterologous sites, the effectiveness and accuracy of gap closing significantly affects downstream analyses, including gene annotations for the constructed assemblies. Connecting the contigs (non-gapped DNA segments split by gaps in scaffolds and hereafter referred to as ‘subcontig’) encompassing a gap in a scaffold by closing the gaps increases the length of the subcontigs and its capacity to include the encoded genes to be annotated.

Gap closing has traditionally been performed by manually or semi-automatically sequencing PCR-products generated with primers that encompass the gaps (Assefa *et al.*, 2009; Gordon *et al.*, 2001). However, recently developed gap-closing tools that use NGS reads, GapCloser (Li *et al.*, 2010a, b), IMAGE (Tsai *et al.*, 2010), GapFiller (Boetzer and Pirovano, 2012) and FinIS (Gao *et al.*, 2012), have replaced the laborious manual gap-closing process. The former three tools use a similar strategy to close gaps: (i) alignment of PE reads to the contig regions of the scaffolds; (ii) collection of unmapped mates when the other read of a pair is aligned to a region neighboring a gap; (iii) local assembly of collected unmapped reads based on de Bruijn graphs; and (iv) the iteration of processes (i–iii). IMAGE is usually only applicable to assemblies of small genomes (Boetzer and Pirovano, 2012). GapCloser is designed for use as a finishing process of a SOAPdenovo assembler, but can be used as an independent tool. GapFiller is unique in that it assembles only reads that are partially aligned to span a gap, and considers the consistency between the gap lengths and the lengths of the sequences used to fill the gaps to increase the accuracy. FinIS closes gaps by constructing local assembly graphs using reads that were not used for the assemblies and pre-built contig-graph information. FinIS can only be applied to assemblies from some specific assemblers (i.e. Velvet and SOAPdenovo).

Another tool to close the gaps in assemblies is PBJelly (English *et al.*, 2012), which use PacBio long reads instead of short NGS reads. PBJelly closes gaps and corrects errors in the closed sequences that are derived from PacBio reads by adopting consensus sequences of the multiple reads aligned to the gap regions. Additionally, assembly-merging tools, PCAP.REP (Huang *et al.*, 2006), GAM-NGS (Vicedomini *et al.*, 2013), GAA (Yao *et al.*, 2012), GARM (Soto-Jimenez *et al.*, 2013), CISA (Lin and Liao, 2013), FGAP (Piro *et al.*, 2014), MAIA (Nijkamp *et al.*, 2010), Minimus2 (Sommer *et al.*, 2007) and Mix (Soueidan *et al.*, 2013) can potentially close the gaps by merging the assemblies, although they are intended for assembly extension rather than gap closure.

In this study, we demonstrated that all the existing gap-closing tools cause a high rate of misassembly at gap-closing sites. To develop a gap-closing tool with a lower error rate, we adopted a strategy with which gaps are closed with a set of preassembled contigs or long reads. Our strategy involves the selection of the correct alignments between scaffolds and contigs (or long-reads) by measuring the likelihood ratios of the true alignments. We demonstrate that this likelihood-based method allows the efficient and accurate connection of contigs and is applicable to other technologies that involve the merging or assembly of contigs or long reads.

2 Methods

2.1 Sequence read sets and reference genomes used

Sequence reads and reference genome sequences were obtained from several websites, and simulated read sets were generated using a pIRS (Hu *et al.*, 2012), SimSeq (https://github.com/jstjohn/SimSeq), 454sim (Lysholm *et al.*, 2011) and PBSIM (Ono *et al.*, 2013) simulators (Supplementary Table S1). PacBio reads were error-corrected using the PBcR (pacBioToCA) component of the Celera assembler ver. 8.1 (Denisov *et al.*, 2008; Koren *et al.*, 2012) or proovread (Hackl *et al.*, 2014). A detailed explanation for this section is described in detail in Section S2.1 of Supplementary Methods.

2.2 Generation of genome assemblies

To generate contigs Illumina paired-end read sets were assembled using iterative de Bruijn graph-based assemblers, SOAPdenovo2 ver. 2.23 (Li *et al.*, 2010b; Luo *et al.*, 2012), IDBA-UD ver. 1.1.0 (Peng *et al.*, 2012) and ALLPATHS-LG ver. 45879 (Gnerre *et al.*, 2011) or an FMD-index-based assembler, Fermi ver. 1.1 (Li, 2012) and 454 reads were assembled using an overlap-layout-consensus-based assembler, Newbler ver. 2.7 (Margulies *et al.*, 2005). Scaffolds were generated with Illumina paired-end and mate-pair read sets using SOAPdenovo2, ALLPATHS-LG or SSPACE premium ver. 2.3 (Boetzer *et al.*, 2011; http://www.baseclear.com/landingpages/basetools-a-wide-range-of-bioinformatics-solutions/sspace-premium/; Supplementary Table S2). The *de novo* assembly was conducted with options –K = 45–55 –m = 70–90 –d = 3 –D = 3 –L = 200 –C = 2.5 for SOAPdenovo2 and with the default settings for other assemblers. Scaffolding of *Caenorhabditis elegans* Newbler contig set and *Arabidopsis* SOAP contig set was performed with SSPACE premium with options –m = 50 and –z = 500. To evaluate the performance of FinIS, Illumina reads were assembled with SOAPdenov ver. 1.0.5 (Li *et al.*, 2010b), and the resulting contigs were scaffolded with paired-end and mate-pair reads using Opera ver. 1.3 (Gao *et al.*, 2011). *Mycobacterium abscessus* assembly sets generated with Illumina HiSeq reads were obtained from the GAGE-B website (http://ccb.jhu.edu/gage_b). For all the contig sets obtained, contigs with <200 bp were filtered. To create error-free assemblies, we split the scaffolds at misscaffolded sites and corrected or deleted misassembled (sub)contigs using our program, GMvalue (which is included in the GMcloser package). Although GMvalue does not correct SNPs or short indels, it creates an assembly set with no misassembly, which is hereafter referred to as 'error-free' assembly. The statistics for the resulting error-free assembly sets are shown in Supplementary Table S2.

2.3 Statistical analysis of the alignments generated between contig sets and between contigs and PE reads

The error-free rice or *C.elegans* contig sets generated with SOAPdenovo2, FERMI and Newbler were reciprocally aligned with Nucmer, and end-to-end alignments were extracted. The aligned contig pairs were merged and the merged fragments were aligned to the respective reference genome. Based on the reference alignment data, alignments with ≥95% identity and ≥95% contig coverage were classified as 'true' alignments and the others as 'false' alignments. Finally, we statistically evaluated the frequencies of the lengths and the identities of the alignment overlap regions of the true and false alignment fractions (Supplementary Fig. S1A, B; Supplementary Table S4). We also conducted a statistical analysis of the PE-read alignments with the contigs. A set of experimentally obtained PE reads of rice (24 × coverage) was aligned to the assembled error-free contigs. We computed the rates of the aligned read pairs that supported the true or false contig–contig alignments in the total number of reads aligned to the contig–contig segments (Supplementary Fig. S1C; Supplementary Table S4). The likelihood ratios were calculated according to the following Equation (I).

$$L(T|a) = \frac{Ta/TA}{Fa/FA} \qquad \text{(I)}$$

where $L(T|a)$ and Ta represent the likelihood ratio and frequency of the true alignments, respectively, for a range '*a*' in an alignment factor (e.g. alignment identity); *Fa* indicates the frequency of false alignments for a range '*a*' in an alignment factor; and *TA* and *FA* indicate the total number of true and false alignments, respectively.

To calculate the Bayesian posterior probabilities from the likelihood ratios, we consider the posterior odds. Because the posterior odds are the likelihood ratios multiplied by the prior odds, the Bayesian posterior probability is given by the following Equations (II) and (III).

$$O(T|a) = L(T|a) \cdot TA/FA \tag{II}$$

$$P(T|a) = O(T|a)/(1 + O(T|a)) \tag{III}$$

where $O(T|a)$ and $P(T|a)$ represent the posterior odds and the Bayesian posterior probability of the true alignments for a range '*a*' in the alignment factor, respectively. Therefore, the Bayesian posterior probabilities were calculated according to the following Equation (IV).

$$P(T|a) = \frac{L(T|a) \cdot Ta/(TA - Ta)}{1 - L(T|a) \cdot (Ta/(TA - Ta))} \tag{IV}$$

A detailed explanation for this section is described in detail in Section S2.2 of Supplementary Methods.

2.4 Algorithm of GMcloser

GMcloser is executed in the following five steps: (i) alignment of a contig (or long read) set to the subcontigs in the scaffold; (ii) alignment of PE reads to both sets of contigs (or long reads) and subcontig; (iii) likelihood-based selection of the correct contig–subcontig alignment pairs using contig- and read-alignment statistics; (iv) filling and closure of the gaps with the selected and assigned contigs (or long reads); and (v) connection of subcontig pairs that encompass a gap.

2.4.1 Alignment of contigs (or long reads) to scaffolds

The scaffolds are split into subcontigs and the subcontigs are aligned with another preassembled contig (or long read) set using MUMer/Nucmer ver. 3.23 (Kurtz *et al.*, 2004). This alignment can be performed with another aligner, BLASTn, which is suitable for the alignment of assemblies of large genomes, by optional selection. After alignments separated by short indels are connected, the alignments covering the entire region or either terminus of the subcontigs (end-to-end) are stored.

2.4.2 Alignment of PE short reads

The first and second read sets of PE short reads with 300–800 bp fragment size and with 20–100× coverage are aligned separately to both the contig (or long read) and subcontig sets with Bowtie2 in the single-end mode. Short reads with more than one mismatch with a haploid genome or two mismatches with a heterozygous diploid genome are filtered using the filtering function of Coval (Kosugi *et al.*, 2013), and then the positions, directions and names of the aligned short reads are recorded for the two short read sets and for the two assembly sets.

2.4.3 Selection of the correct contig–subcontig alignments

To ascertain whether the contig–subcontig (or long read-subcontig) alignments are correct, a likelihood-based estimate is made using predetermined likelihood ratios for true test alignments. The likelihood ratios for the alignment overlap length, overlap identity and PE-read mapping rate were determined with true and false contig–contig alignment data for rice (Supplementary Fig. S1). The three likelihood ratios for each contig–subcontig alignment are multiplied together to compute an alignment score. When the calculated score for an alignment is equal to or larger than the threshold score, the alignment is deemed to be true. The threshold score will deviate from the standard score, depending especially on the quality and/or sequence length of the assembly sets and PE reads to be used. To set a threshold score to be optimized for the input data, we added a data-specific constant (*C*) to a standard threshold score. The data-specific constant is calculated by subtracting the mean score for the rice standard alignment data from the mean alignment score for the sample alignment data. The formulae used to calculate the alignment score (*Sa*) and threshold score (*THa*) are:

$$Sa = \log\Big(L(T|i) \cdot L(T|l) \cdot L(T|m)\Big)$$

$$THa = THs + C$$

$$C = \frac{\Sigma Sa}{n} - Sm$$

where $L(T|i)$, $L(T|l)$ and $L(T|m)$ are the likelihood ratios of true alignments corresponding to the predetermined ranges for alignment overlap identity, alignment overlap length and PE-read mapping rate, respectively; *THs* is the standard threshold score (i.e. 2.5); *Sm* is the mean alignment score for the rice alignment data; and *n* is the total number of alignments in the sample data. When a contig set is used as a query and multiple contigs are aligned around a single gap, GMcloser selects the single contig whose alignment overlap length is the longest. When a single contig is aligned around multiple gaps in different assemblies, GMcloser selects the single alignment with the longest alignment overlap length.

2.4.4 Filling gaps in scaffolds

Using the selected contig–subcontig alignment data, the gaps present in the scaffold dataset are filled and the scaffold termini are optionally extendable. For filled but not completely closed gaps, a pairwise alignment between the subcontigs that encompass the gap is performed using the YASS aligner (Noe and Kucherov, 2005), and the gap is closed only when the 3′-terminal segment of the upstream subcontig is aligned to the 5′-terminal segment of the downstream subcontig. If the pairwise alignment gives no significant result and the contig segment extended to fill the gap is longer than 1.5 times the gap length, the contig assigned to the gap is not used to fill the gap. We chose 1.5 times the gap length as the threshold is because the standard deviations of the fragment sizes of the PE reads and MP reads are generally ∼20% of their fragment sizes.

2.4.5 Connection of neighboring subcontigs

We observed that a significant number of connectable, neighboring subcontig pairs that encompass a gap were originally present in input scaffolds before the gap-closing treatment (Supplementary Table S3). The rate of connectable subcontig pairs in the total subcontig pairs varied from 0.2 to 46%, depending on the assemblers and read datasets used. Because the overlapping lengths between the subcontig pair are relatively short, the connection of the subcontig pair with simple filtering (e.g. ≥95% identity and an overlapping length of ≥20 bp) could lead to a high rate of gap-closing errors (i.e. false positive rate of 3–44%; Supplementary Table S3). To accurately connect the connectable subcontigs preexisting in scaffolds, GMcloser uses the three alignment factors (i.e. the alignment overlap length, alignment identity and mapping rate of the PE reads) to select the correct subcontig pairs. This function is optionally selectable.

When using long reads (e.g. error-corrected PacBio reads) as the query, the entire gap-closing process can be iteratively implemented in the long-read mode. GMcloser assumes that the coverage of a long read set is ≥ 2× and that of a contig query set is < 2×. When using a contig set with ≥ 2× coverage, the job can be run in the long-read mode. Three iteration processes can often achieve almost maximal gap-closing efficiency when using a query set with ~15× coverage.

2.5 Gap closing with gap-closing tools

For GMcloser, PCAP.REP (Huang *et al.*, 2006), GAM-NGS (Vicedomini *et al.*, 2013) and GAA (Yao *et al.*, 2012), gaps in error-free scaffolds were closed with an error-free contig set or an intact contig set. GMcloser was executed using an option that connects neighboring subcontigs (–c), unless otherwise stated, and PCAP.REP used an option (–m) specifying the minim identity of overlap option, which was set to 97. For GapCloser ver. 1.12 (Li *et al.*, 2010b) and GapFiller ver. 1.9 (Boetzer and Pirovano, 2012), gap closure was performed with PE read sets and the default options, except that the option specifying the minimum overlap length (–p for GapCloser and –m for GapFiller) was set to 31 unless otherwise stated. Gap closure with IMAGE ver. 2.33 (Tsai *et al.*, 2010) was performed with options –iteration = 1, –all_iteration = 10 and –kmer = 61 (or –kmer = 51 for the *Arabidopsis* data). Gap closure with FinIS ver. 0.3 (Gao *et al.*, 2012) was performed using a real scaffold set generated with Opera ver. 1.3 (Gao *et al.*, 2011) and the assembly data generated with SOAPdenovo ver. 1.0.5 (Li *et al.*, 2010b). For PBJelly ver. 14.9.9 (English *et al.*, 2012), gap closing was conducted with error-corrected or intact PacBio read set and the default options. The specified options for each tool achieved the most accurate gap closure while retaining ≥90% of the maximal gap-closing effectiveness for most assembly sets.

2.6 Evaluation of gap-closing results

The accuracy of gap-closed assemblies were evaluated using GMvalue and Quast (Gurevich *et al.*, 2013). In evaluating the accuracy of assemblies with GMvalue, we regarded as true only those exhibiting ≥99% contig coverage, ≥97% identity and ≥200 bp alignment length in the overlaps with the references, and permitted SNPs and short indels with ≤100 bp in the assemblies. The detailed explanation for assembly evaluation is described in Section S2.3 of Supplemental Methods.

3 Results

3.1 Current gap-closing tools display a high rate of misassembly in the gap-closing process

The currently available gap-closing tools, including IMAGE, GapCloser and GapFiller, close gaps in scaffolds using k-mer-based local assembly of PE reads that are aligned around the gap regions. *De novo* assemblers also use a similar k-mer-based (de Bruijn graph) strategy to assemble NGS reads. We compared the error (misassembly) rates of these gap-closing tools with those of *de novo* assemblers (Table 1). The same sets of PE reads from *C.elegans*, *Arabidopsis*, and rice were used to execute gap closers (IMAGE, GapCloser and GapFiller) and assemblers (SOAPdenovo2 [Li *et al.*, 2010b; Luo *et al.*, 2012] and FERMI [Li, 2012]). To accurately assess the misassembly rates of the gap closers, we generated error-free preassembled scaffold sets (see Supplementary Methods and Supplementary Table S2), and used for gap closing. To measure the misassembly rate for each tool, the assembled contigs or subcontigs that were split from the gap-closed assemblies were aligned to the corresponding reference genomes using MUMmer/Nucmer aligner (Kurtz *et al.*, 2004). Misassembly events were counted (allowing single-nucleotide polymorphisms [SNPs] and short insertions/deletions [indels]) with a method (GMvalue) related to that used for the GAGE assembly evaluation (Salzberg *et al.*, 2012) and the Quast tool (Gurevich *et al.*, 2013), because the available assembly evaluation tools cannot be applied to assemblies of large genomes and lack the options to specify misassembly definitions. The number of misassemblies per closed or assembled sequence length (Mb) for each tool and each species indicated that the gap-closing tools caused 25–500-fold higher rate of misassembly than the *de novo* assemblers (Table 1). This result demonstrates that the current k-mer-based gap-closing tools incorporate misassembled fragments into the assembly gap regions at a considerably high rate.

Table 1. Misassembly rate in assemblies generated with k-mer-based *de novo* assembly and gap-closing tools

Category	Tools	PE reads[c]	Total errors	Error rate[d]
Gap closer[a]	IMAGE	*C.elegans*	3055	729
		Arabidopsis	1829	653
		Rice	—[e]	—[e]
	GapCloser	*C.elegans*	725	165
		Arabidopsis	679	63
		Rice	992	31
	GapFiller	*C.elegans*	4124	1031
		Arabidopsis	770	99
		Rice	1420	87
De novo assembler[b]	SOAPdenovo2	*C.elegans*	379	4
		Arabidopsis	169	1.5
		Rice	50	0.16
	FERMI	*C.elegans*	462	4.4
		Arabidopsis	276	2.5
		Rice	1084	3.5

[a] Scaffold set used: SOAP scaffolds for rice and SSPACE scaffolds for the others.

[b] Contigs ≥ 500 bp were used for counting the misassemblies.

[c] PE-read sets for the indicated species, which are shown in Supplementary Table S1 and were used for the execution of the indicated tools.

[d] Misassembly events per total length (Mb) of the assembled contig sequences or gap-filled sequences.

[e] No output was obtained, probably because the input dataset was large.

3.2 Selection of correct contig–contig alignments with the likelihood ratios for the three alignment factors

To obtain alignment data for contigs, we prepared three rice contig sets generated with SOAPdenovo2, FERMI and Newbler assemblers. These three contig sets were reciprocally aligned between each other with Nucmer, and the resulting alignment data were categorized into 122,133 true and 2584 false alignments using GMvalue (see Supplementary Methods for detail). A statistical analysis of the contig alignment data showed that the likelihood ratios calculated for the alignment overlap length, alignment identity and mapping rate of the PE reads provide good classifiers for selecting the correct contig alignments (Supplementary Fig. S1; Supplementary Table S4). The likelihood ratios for the three factors in the rice data were similar to those for the *C.elegans* alignment data (Supplementary Fig. S2), and similar to the posterior probability based on Bayes' theorem (Supplementary Fig. S3). To make a combined alignment score from the three alignment factors, we multiplied likelihood ratios for the three alignment factors for each alignment. From the multiplied scores for each true and false alignment datasets, we calculated the mean log score for each true and false alignment dataset (true data, 4.5; false data, –5.2). A receiver operating characteristics (ROC) curve analysis of the likelihood-based selection using the rice contig alignment data showed that selection with several threshold scores accurately discriminated between true and false alignments (Supplementary Fig. S4; Supplementary Table S5). We specified a threshold score of 2.5, which accepted 87% of the true alignments and rejected 97% of the false alignments.

3.3 Development of the gap-closing tool, GMcloser

We developed the gap-closing tool, GMcloser, by introducing a strategy to select correct contig alignments, as described above. An outline of the gap-closing processes of GMcloser is shown schematically in Figure 1 and Supplementary Figure S5, and the detailed algorithm is described in Methods. First, the target scaffold sequences are split into subcontigs and the subcontig set is aligned with another set of query contigs or long reads using MUMmer/Nucmer or BLASTn, and both the end-to-end alignments and alignments that are completely covered by a query sequence are extracted. The target subcontig set and the query sequence set are aligned separately with PE short reads. The PE read sequences themselves are not used to fill the gaps, but are used only as evidence when selecting the contigs/long-reads to fill the gaps. From the extracted end-to-end sequence alignments, the potentially correct alignments are selected with a threshold score that is set based on the alignment likelihood ratios, as described in the previous sections. Using the selected alignment data and query sequence information assigned to the gap regions of the scaffolds, the scaffolds are reconstituted and the gaps are filled.

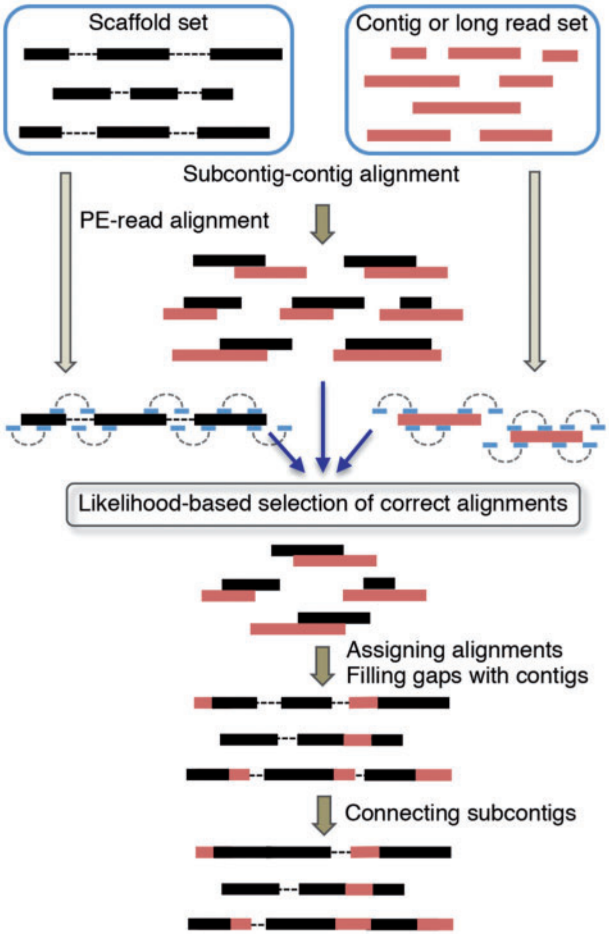
Scaffold set
Contig or long read set
Subcontig-contig alignment
PE-read alignment
Likelihood-based selection of correct alignments
Assigning alignments
Filling gaps with contigs
Connecting subcontigs

Fig. 1. Outline of the gap-closing process of GMcloser. Scaffold sequences in which gaps are to be closed are split into subcontigs, and the subcontigs are aligned with another set of preassembled contigs or long reads using Nucmer or BLAST to find their end-to-end alignments. Each set of subcontigs and contigs is aligned with PE short reads using Bowtie2. To select the correct subcontig–contig alignments, the predetermined likelihood ratios for the alignment overlap length, overlap identity and PE-read mapping rate are multiplied together, and the log values are assigned as the alignment scores for the individual alignments. Alignments with scores lower than the threshold score, which is calculated with a predetermined standard score and a data-specific constant, are discarded as incorrect alignments. The scaffolds are reconstituted and the gap regions in the scaffolds are filled with the assigned contigs (or long reads). Finally, the closure of any gaps that remain unclosed is attempted using a local pairwise alignment of a subcontig pair that encompass the gap. For a contig or long-read set with ≥ 2× coverage, the entire process can be iteratively implemented in the long-read mode

We compared the gap-closing performances of GMcloser and a simple merging method using a modified version of GMcloser omitting both the PE read alignment and the likelihood-based alignment selection steps. To define only errors occurring at the gap-closing step, we used error-free scaffold and contig sets of *C.elegans*, *Arabidopsis*, and rice, which are shown in Supplementary Figure S6. GMcloser closed the gaps in assemblies with 3–12-fold less misassemblies than the simple merging method, while their gap-closing effectiveness was similar.

3.4 Comparison of the performances of GMcloser and other tools using rice, *Rhodobacter sphaeroides, Saccharomyces cerevisiae, C.elegans, Arabidopsis,* sorghum and foxtail millet assemblies

We tested the performance of GMcloser using assemblies from rice, *Rhodobacter sphaeroides*, *Saccharomyces cerevisiae*, *C.elegans*, *Arabidopsis*, foxtail millet and sorghum. For each organism, we prepared error-free scaffolds and closed the gaps in the error-free scaffolds with GapCloser, GapFiller, IMAGE and GMcloser. We designated error-free assemblies as assemblies that are not different from the sequences of the corresponding reference genome. To evaluate the performance of GMcloser, we used two types of contig sets: an error-free contig set and a real contig set containing misassemblies. By using error-free contig sets, we can determine the number of misassemblies that are introduced by GMcloser itself. We computed the total lengths of the sequences that were newly added to the gap regions of the input scaffolds and the total numbers of gap-closing/assembly errors observed in the output gap-closed scaffolds. Because the input scaffolds lacked assembly errors, all the observed errors correspond to the errors generated by the individual gap-closing tools. Evaluation of the output assemblies showed that the gap-closing accuracy of GMcloser was considerably higher than that of the other tools, whereas its gap-closing effectiveness varied from 61 to 440% of the effectiveness of GapCloser or GapFiller (Fig. 2; Supplementary Table S6, also see Supplementary Fig. S7 for their ROC-like plots). In most cases, over 80% of misassembly events were observed in independent subcontigs. The increased errors (73% increase on average) in GMcloser assemblies generated with real contig sets were probably derived from assembly errors contained in the input contigs. Relatively high numbers of assembly errors observed for sorghum and foxtail millet are probably because the reference genomes were constructed using *de novo* assembly with short reads. The gaps were efficiently closed even when the same assembler was used for the constructions of the contig set and scaffold set (i.e. ALLPTHS-scaf/ALLPATHS-contig in Fig. 2A), indicating that the sequences that cover the gaps in a scaffold set are not

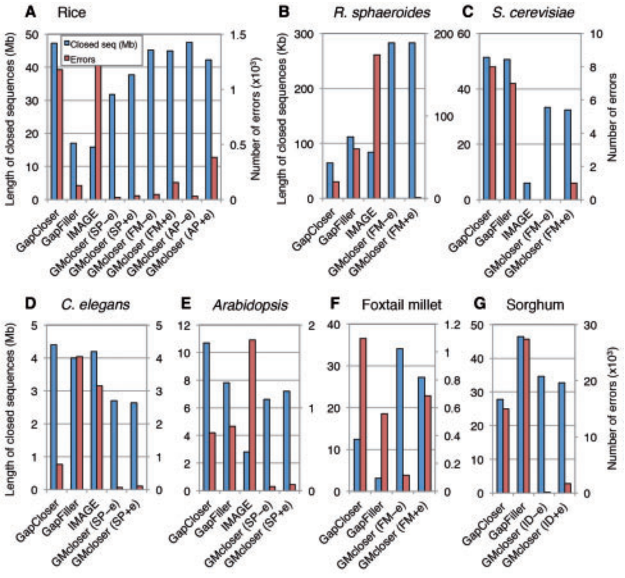

Fig. 2. Comparison of the performances of gap-closing tools using rice, *R.sphaeroides*, *S.cerevisiae*, *C.elegans*, *Arabidopsis*, sorghum and foxtail millet sequencing data. (**A**) Gap closing in rice scaffolds generated with ALLPATHS-LG. An error-free ALLPATHS scaffold set was used to evaluate the indicated gap-closing tools, including GMcloser, GapCloser, GapFiller and IMAGE. (**B**) Gap closing in *R.sphaeroides* scaffolds generated with SOAPdenovo. (**C**) Gap closing in *S.cerevisiae* scaffolds generated with SOAPdenovo. (**D**) Gap closing in *C.elegans* scaffolds generated with SSPACE. (**E**) Gap closing in *Arabidopsis* scaffolds generated with SSPACE. (**F**) Gap closing in foxtail millet scaffolds generated with ALLPATHS-LG. (**G**) Gap closing in sorghum scaffolds generated with SOAPdenovo2. For GMcloser, contig sets generated with real (error-containing) SOAPdenovo2 (SP + e), FERMI (FM + e), ALLPATHS-LG (AP + e) and IDBA (ID + e) contig sets, and their error-free contig sets (SP–e, FM–e, AP–e and ID–e, respectively) were used for gap closing. The total length (Kb for *R.sphaeroides* and *S.cerevisia* and Mb for the others) of the gaps filled with sequences and the total number of misassembled/mismerged events for each tool are shown with blue (left scale on the y-axis) and red bars (right scale), respectively

necessarily underrepresented in a contig set that is generated with the same assembler. Evaluation of the misassemblies in the gap-closed sequences with the Quast tool (Gurevich *et al.*, 2013) showed that the results of Quast correlated roughly with those of our evaluation method (GMvalue), although it is difficult to quantitatively compare between these results because the definition of 'local misassembly' differs between the two methods. The Quast program could not be applied to assemblies from relatively large genomes, including those of rice, foxtail millet and sorghum (Supplementary Table S6).

We tested the performance of another gap-closing tool, FinIS (Gao *et al.*, 2012). This tools is limited to assemblies generated with the Opera scaffolder (Gao *et al.*, 2011) and the Velvet or SOAPdenovo ver. 1 contig assembler. Because this tool is incompatible with user-generated scaffold sets including error-free scaffold sets, we generated another scaffold sets using Opera and SOAPdenovo ver. 1 to compare the performances of FinIS with those of the other tools. Our evaluation indicated that FinIS caused the highest number of misassemblies of all the gap-closing tools tested, despite its similar effectiveness (Supplementary Fig. S8).

We also tested the gap-closure performances of scaffold-compatible assembly-merging tools, PCAP.REP, GAM-NGS, GAA, GARM and Mix using preassembled error-free scaffold/contig sets. Although PCAP.REP and Mix exhibited a moderate level of the gap-closing effectiveness, they introduced a considerably higher number of misassemblies than the other tools (Supplementary Fig. S9). GAM-NGS and GAA closed the gaps in the tested scaffold sets negligibly (the results of GARM could not be obtained because of an error in the program; Supplementary Fig. S9).

Table 2. Runtime and memory consumption of gap-closing tools

Data	Tools	Thread[c]	Runtime (min)[d]	Max memory (Gb)
Yeast	GapCloser	4	2	1.3
		16	2	1.6
	GapFiller[a]	4	25	0.5
		16	15	0.8
	MAGE[a]	1	397	1.1
	GMcloser[b]	4	8 (31)	1.1
		16	5 (24)	1.2
C.elegans	GapCloser	4	40	15.3
		16	27	13.7
	GapFiller[a]	4	1797	0.6
		16	1113	2.2
	IMAGE[a]	1	15 510	14.8
	GMcloser[b]	4	174 (198)	19.7
		16	70 (86)	21.8
Rice	GapCloser	4	100	26.6
		16	67	26.6
	GapFiller[a]	4	3430	0.5
		16	3043	1.4
	IMAGE[a]	1	29 809	8.0
	GMcloser[b]	4	397 (944)	29.2
		16	197 (507)	39.8

[a] GapFiller and IMAGE were applied for 10 iterations, which were specified with the –i and –all_iteration options, respectively.

[b] Preassembled contig sets used for gap closing with GMcloser: real SOAP contig set for *C.elegans*: real FERMI contig set for yeast and rice.

[c] Number of threads to run.

[d] Values in parentheses indicate the runtime including the time spent in assembly of the input contig sets for GMcloser.

We measured the running time and memory consumption of each gap-closing tool (Table 2). The runtime of GMcloser was longer than that of GapCloser but shorter than those of GapFiller and IMAGE. The maximal memory of GMcloser was comparable to those of GapCloser, but larger than the other tools.

3.5 Sequential treatment of scaffolds with GMcloser closes more gaps

A scaffold set in which the gap had been closed with GMcloser was treated again with GMcloser with another preassembled contig set. The gap regions of the resulting scaffolds for the datasets from rice, *C.elegans* and *Arabidopsis* were further reduced relative to those in the input scaffold sets, although the assembly errors were slightly increased in proportion to the closed gap lengths (Supplementary Figs S10 and S11; Supplementary Table S7). An additional treatment with another contig set (i.e. AP contig sets) further reduced the gap regions of the assemblies (see SP:FM:AP–e and SP:FM:AP+e; Supplementary Fig. S10A). These results indicate that different assemblers generate contigs that cover different gaps in a scaffold set, in addition to contigs that cover the same gaps. Successive treatments of GMcloser-treated scaffolds with GapCloser increased the number of misassemblies, and the number of misassemblies and the lengths of closed gaps were close to those found when treated with GapCloser alone (Supplementary Fig. S12), indicating that GapCloser has a relatively constant error rate for similar datasets.

3.6 Gap closure with Pacific Biosciences RS reads

We attempted gap closure with Pacific Biosciences (PacBio) RS CLR reads from *R.sphaeroides*, *S.cerevisiae* and *C.elegans* using GMcloser. The real PacBio RS reads for *R.sphaeroides* and the W303 strain of *S.cerevisiae* were obtained from the public websites (Supplementary Table S1), and the simulated PacBio reads for *C.elegans* and the S288c strain of *S.cerevisiae* were generated with PBSIM (Ono *et al.*, 2013). The potential sequencing errors in these read sets were corrected with Illumina short reads using PBcR (pacBioToCA; Denisov *et al.*, 2008; Koren *et al.*, 2012) or proovread (only for *C.elegans* reads; Hackl *et al.*, 2014; Supplementary Table S8), and the resulting read sets had 3.5×, 7× and 14× coverage for each organism, respectively. Error-corrected PacBio reads of the yeast W303 strain were obtained from the corresponding website. For *R.sphaeroides* and *S.cerevisiae* W303, GMcloser and PacBio reads with 7× coverage closed 7.8-fold and 1.7-fold longer length of gaps in the assemblies than GapCloser, respectively, while the number of misassemblies was similar to that generated with GapCloser for *R.sphaeroides* and was only 7% of that of GapCloser for *S.cerevisiae* W303 (Fig. 3A, Supplementary Fig. S13, Supplementary Table S9). For *S.cerevisiae* S288c and *C.elegans*, although the lengths of the gaps closed with GMcloser were similar to those closed with GapCloser, the observed misassembly events were 18∼25% of those in observed for GapCloser (Fig. 3B and D, Supplementary Fig. S13, Supplementary Table S9). In all the assemblies tested, GMcloser with error-corrected PacBio reads closed the gaps more effectively than that with preassembled contig sets. Notably, sequential treatments of the assemblies with contig and error-corrected PacBio read sets allowed more efficient and accurate gap closing than the treatments with PacBio reads alone. We then compared the performances of GMcloser and PBJelly. Gap-closing tests using PBJelly with intact or error-corrected PacBio reads with 14- to 50-fold coverage showed that PBJelly introduced a 9- to 260-fold higher number of errors into the gap-closed assemblies than GMcloser, while the effectiveness of gap-closure was similar to GMcloser or GapCloser (Fig. 3, Supplementary Fig. S13, Supplementary Table S9). The gap-closed assemblies generated with PBJelly contained a number of assemblies containing incompletely error-corrected PacBio reads, many of which were measured as local-misassembly events, as represented in Supplementary Figure S14. Overall, these results indicate that GMcloser can close the gaps more accurately than other gap-closing tools. The efficient and accurate gap-closing ability of GMcloser with a low coverage of PacBio reads should improve the closing of gaps in assemblies.

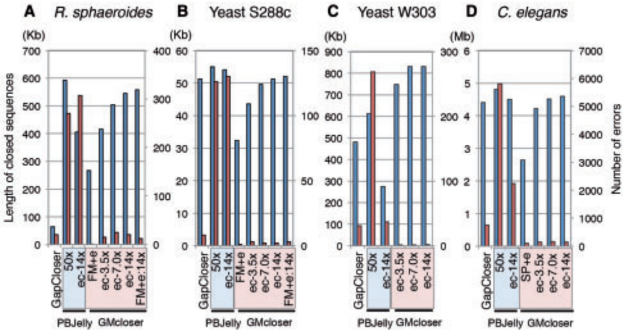

Fig. 3. Gap closure with PacBio RS reads. (**A**) Gap closing in *R.sphaeroides* SOAP scaffolds. Real PacBio RS reads were error-corrected with PBcR (pacBioToCA) and Illumina reads, and the error-corrected (ec-) PacBio reads with the indicated coverage were used for gap closure with GMcloser. Gap closing with PBJelly was performed with intact and error-corrected PacBio reads with the indicated coverage. (**B**) Gap closing in *S.cerevisiae* S288c SOAP scaffolds. (**C**) Gap closing in *S.cerevisiae* W303 SOAP scaffolds. (**D**) Gap closing in *C.elegans* SSPACE scaffolds. For *S.cerevisiae* S288c and *C.elegans*, simulated PacBio reads were generated with PBSIM, and error-correction was conducted using pacBioToCA and Proovread, respectively. Gap-closing results for GapCloser and GMcloser with a real contig set (FM + e or SP + e) are shown for comparison. Results of successive treatments using GMcloser with a real contig set and PacBio reads are also shown (FM + e:14×). The bars are as in Figure 2. The data are from the assembly results of GMcloser at iteration 3. The minimum alignment overlap identity to measure misassembly events with GMvalue was set to 92%, because the PacBio reads could be incompletely error-corrected

3.7 Statistical evaluation of the gap-closing results

For the gap-closed assemblies shown in the gap-closing results above, we measured how many numbers of subcontig termini or gaps in the scaffolds were extended or closed (Supplementary Table S10). The percentage of extended subcontig termini or closed gaps correlated with the total length of the closed gaps in many cases. In six out of the eight experiments for different organisms, the ability of GMcloser to extend the subcontigs was higher than or similar to those of the other tools (Supplementary Table S10).

We classified the misassemblies observed in the gap-closed assemblies into four classes, local misassembly, relocation (local misassembly with the break-point distance of ≥1 Kb), translocation and inversion. Local msassembly was the major class for all the tools, and GMcloser tended to produce fewer errors of translocation and inversion (Supplementary Table S11). This observation is supported by the Quast evaluation results (Supplementary Table S13), although the definition of 'local misassembly' differs between Quast and GMvalue. In GMvalue, local misassemblies are considered to occur when separate segments from a contig are aligned to an identical chromosome of the reference and their break points occur at a distance of >100 bp (corresponding to the maximum allowable size [i.e. 100 bp] of indels), whereas local misassemblies in Quast contain only the relocated contig segments with break points at a distance of <1000 bp. Because the minimum threshold value for the break-point distance and the alignment coverage/identity of the contigs are not stipulated in Quast, it is difficult to quantitatively compare the results for local misassembly in GMvalue and Quast. This local-misassembly event was observed more frequently when the maximum allowable size of indels (or the distance between the break points of a local misassembly) was lowered (Supplementary Fig. S15). However, the specified values for the maximum allowable size of indels/local misassemblies usually generated similar ratios for the number of misassemblies between the gap-closing tools, as shown in Supplementary Figure S15.

We also measured the N50 values and the corrected N50 values (i.e. N50 values for assemblies split at misassembly sites) for the gap-closed assemblies (Table 3). The results indicate that GMcloser produced higher values for the corrected scaffold N50s than the other tools. GMcloser also gave similar or higher values for the corrected subcontig N50s in many cases, but the corrected N50 metrics, especially for subcontigs, may not be useful for evaluating gap-closing performances because they do not effectively reflect the assembly error rates in an assembly set. Instead, they were almost constant, even when ∼50% of the sequences in an assembly set were split at randomly selected sites, as shown in Supplementary Table S12.

4 Discussion

Current gap-closing tools use a strategy that aligns PE-reads, followed by the k-mer-based local assembly of the reads aligned around the gap regions. Although this strategy efficiently closes gaps, it

Table 3. N50 values for gap-closed assemblies

Organisms	Input or output data	Number of errors[a]	Corrected N50 (Kb)	
			Scaffolds	Contigs[b]
Rhodobacter	SOAP-scaf input	0	154.6	2.32
	FM + e input	3 (0.5%)	–	14.0 (14.0)
	ec-PacBio reads (14x)	1210 (2.4%)	–	1.8 (1.8)
	GapCloser	20 (1%)	104.9	3.5 (3.5)
	GapFiller	60 (3.8%)	69.6	4.3 (4.3)
	IMAGE	174 (7.7%)	n.d.	2.6 (2.6)
	PBJelly (PacBio)	307 (17.2%)	42.4	3.8 (3.9)
	GMcloser (FM + e)	1 (0%)	156.4	5.4 (5.4)
	GMcloser (PacBio)	20 (2.4%)	137.3	13.8 (14.7)
S.cerevisiae S288c	SOAP-scaf input	0	30.5	20.6
	FM + e input	40 (7.3%)	–	42.8 (44.0)
	ec-PacBio reads (14x)	917 (0.6%)	–	1.3 (1.3)
	GapCloser	8 (0.7%)	30.1	21.5 (21.7)
	GapFiller	7 (0.9%)	30.3	28.6 (28.6)
	IMAGE	0 (0%)	n.d.	20.6 (20.6)
	PBJelly (PacBio)	130 (26.3%)	44.5	43.4 (46.2)
	GMcloser (FM + e)	1 (0.1%)	30.5	25.5 (25.5)
	GMcloser (PacBio)	2 (0.2%)	30.5	29.5 (29.5)
S.cerevisiae W303	SOAP-scaf input	0	472.0	30.4
	ec-PacBio reads (14x)	230 (1.4%)	–	13.8 (13.9)
	GapCloser	31 (4.7%)	336.1	56.3 (61.7)
	PBJelly (PacBio)	37 (3.5%)	458.8	32.6 (33.3)
	GMcloser (PacBio)	2 (1.0%)	462.8	212 (212)
C.elegans	SSPACE-scaf input	0	123.3	3.84
	SP + e input	379 (2.8%)	–	16.3 (16.4)
	FM + e input	496 (5.3%)	–	26.5 (27.0)
	PacBio reads (14x)	4618 (0.7%)	–	3.1 (3.1)
	GapCloser	761 (4.9%)	88,2	11.2 (11.7)
	GapFiller	4037 (27%)	32.4	11.7 (12.4)
	IMAGE	3151 (36%)	n.d.	25.3 (30.5)
	PBJelly (PacBio)	2238 (9.7%)	112.8	8.4 (8.8)
	GMcloser (SP + e)	106 (0.5%)	118.7	9.3 (9.4)
	GMcloser (FM + e)	129 (0.6%)	117.7	7.9 (8.0)
	GMcloser (SP:FM + e)	164 (0.9%)	115.3	11.3 (11.5)
	GMcloser (PacBio)	147 (2.1%)	116.8	40.2 (41.1)
Arabidopsis	SSPACE-scaf input	1	245.8	27.6
	SP + e input	170 (0.8%)	–	12.3 (12.3)
	FM + e input	287 (2.7%)	–	30.0 (30.1)
	GapCloser	699 (2.4%)	138.9	7.2 (7.4)
	GapFiller	774 (5.4%)	146.2	18.8 (19.2)
	IMAGE	1818 (4.3%)	n.d.	3.6 (3.7)
	GMcloser (SP + e)	73 (0.4%)	218.5	13.5 (13.5)
	GMcloser (FM + e)	94 (0.4%)	220.3	12.4 (12.5)
	GMcloser (SP:FM + e)	117 (0.8%)	209.2	19.4 (19.5)
Rice	AP-scaf input	0	313.4	13.0
	SP + e input	49 (0%)	–	11.6 (11.6)
	FM + e input	1059 (2.6%)	–	24.1 (24.4)
	AP + e input	1616 (8.6)	–	39.0 (44.4)
	GapCloser	1179 (7.3%)	209.0	87.3 (98.0)
	GapFiller	127 (0.4%)	285.0	16.9 (17.0)
	IMAGE	1372 (4.1%)	n.d.	16.6 (16.8)
	GMcloser (SP + e)	33 (0.1%)	305.5	35.4 (35.5)
	GMcloser (FM + e)	152 (0.9%)	283.2	44.7 (45.1)
	GMcloser (AP + e)	383 (2.2%)	251.1	44.9 (46.2)
	GMcloser (SP:FM + e)	150 (0.9%)	285.6	47.0 (47.6)
	GMcloser (SP:FM:AP + e)	326 (2.3%)	261.7	54.5 (55.7)
Foxtail millet	AP-scaf input	1	115.1	6.3
	FM + e input	29 171 (11%)	–	3.7 (3.8)
	GapCloser	1096 (4.0%)	114.0	7.5 (7.6)
	GapFiller	556 (2.1%)	114.9	6.7 (6.8)
	GMcloser (FM + e)	685 (2.4%)	85.9	8.7 (8.9)
Sorghum	SOAP-scaf input	33	11.8	6.3
	ID + e input	32 295 (19%)	–	3.8 (4.5)
	GapCloser	14 989 (4.4%)	10.8	4.0 (4.0)
	GapFiller	27 370 (11%)	9.8	7.9 (8.2)
	GMcloser (ID + e)	1710 (0.5%)	11.7	4.5 (4.5)

[a] Percentages of contigs/subcontigs with misassemblies are indicated in parentheses. For PacBio reads and PacBio reads-treated assemblies, the data were measured with 92% of the minimum sequence identity in GMvalue.

[b] Corrected N50 values for subcontig sequences in gap-closed scaffolds. Uncorrected N50 values are indicated in parentheses.

n.d., not determined.

introduces a large number of errors (Table 1). The high misassembly rates of the k-mer-based gap-closing tools probably arise because the gap-closing process involves assembly at genomic sites that are difficult to assemble with *de novo* assemblers. Because the current gap-closing tools use only PE reads that are mapped to gap-flanking regions for local assembly, misassembly could occur frequently at repeated regions in which reads can be incorrectly aligned. These hard-to-assemble sites include repetitive regions and regions of highly polymorphic sequences that are attributable to sequencing errors or heterozygosity. Nevertheless, assembly tools are likely to generate many contigs that cover gaps in the scaffolds, because the gaps present in scaffolds generated with an assembler can be efficiently closed with a contig set generated with the same assembler. Therefore, our gap-closing strategy, based on the alignment of a preassembled contig set, allows accurate and efficient gap closing. The assemblies in which the gaps were closed with GMcloser and with an intact contig set still contained a number of errors. However, most of these errors were derived from misassembly errors contained in the preassembled input contigs. Thus, the use of preassembled contig sets generated with accurate assemblers, such as SOAPdenovo2, should ensure accurate gap closing with GMcloser.

The high accuracy and high effectiveness of GMcloser for gap closing are also due to the likelihood-based selection of the correct alignment between a contig and a subcontig of a scaffold. This selection method uses likelihood ratios for three alignment factors: the alignment overlap length, the alignment overlap identity and the mapping rate of PE reads, which are determined statistically from true and false alignment datasets. Although likelihood-based algorithms or Bayesian probabilistic approaches have been adapted to estimate genotypes, short read alignments and assembly errors (Howison *et al.*, 2013), no likelihood-based method has been developed to select the correct alignment of DNA fragments such as contigs. Our selection method of contig alignment should be applicable to the development of tools involving the alignment of DNA sequence sets, such as assemblers of long reads or contigs.

After a successive treatment of different contig sets with GMcloser, a number of gap regions are still barely filled, and these gap regions seem to correspond to genomic regions with low read coverage. Sequencing reads generated with the PacBio RS platform have long read lengths (mean: 10 kb, max: 50 kb) and relatively constant coverage across the sequenced genome, although the sequencing error rate is high (∼15%; Ross *et al.*, 2013). The errors in PacBio RS reads can be effectively corrected with short reads (Au *et al.*, 2012; Koren *et al.*, 2012) or with a multiple-alignment strategy between the PacBio reads (Chin *et al.*, 2013). In this study, we have demonstrated that the gaps in assemblies can be efficiently and accurately closed with a low coverage of error-corrected PacBio reads using GMcloser, and that the gap-closing accuracy of GMcloser is better than that of another gap-closing tool with PacBio reads, PBJelly (English *et al.*, 2012). This feature of GMcloser should also offer economical benefits, especially for assemblies of large genomes. However, the accuracy of the gap closing with PacBio reads is lower than that with preassembled contig sets, and many of the misassemblies are attributable to incomplete error-correction of PacBio reads (Supplementary Tables S8 and S12). Thus, the future development of more complete methods for error-correction of PacBio reads would allow a more accurate gap closure with GMcloser.

Acknowledgements

We thank Prof. Takehiko Itoh for at Tokyo Institute of Technology for helpful suggestions for the design of this study.

Funding

This work was supported by KAKENHI from the Ministry of Education, Culture, Sports, Science and Technology of Japan [Grant-in-Aid for Scientific Research on Innovative Areas No.221S0002] and by JSPS KAKENHI Grant Number 26430200.

Conflict of Interest: none declared.

References

Assefa,S. *et al.* (2009) ABACAS: algorithm-based automatic contiguation of assembled sequences. *Bioinformatics*, **25**, 1968–1969.

Au,K.F. *et al.* (2012) Improving PacBio long read accuracy by short read alignment. *PLoS One*, **7**, e46679.

Boetzer,M. *et al.* (2011) Scaffolding pre-assembled contigs using SSPACE. *Bioinformatics*, **27**, 578–579.

Boetzer,M. and Pirovano,W. (2012) Toward almost closed genomes with GapFiller. *Genome Biol.*, **13**, R56.

Chin,C.S. *et al.* (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods*, **10**, 563–569.

Denisov,G. *et al.* (2008) Consensus generation and variant detection by Celera Assembler. *Bioinformatics*, **24**, 1035–1040.

English,A.C. *et al.* (2012) Mind the gap: upgrading genomes with Pacific Biosciences RS long-read sequencing technology. *PLoS One*, **7**, e47768.

Gao,S. *et al.* (2012) FinIS: improved in silico finishing using an exact quadratic programming formulation. *Lect. Notes Comput. Sci.*, **7534**, 314–325.

Gao,S. *et al.* (2011) Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. *J. Comput. Biol.*, **18**, 1681–1691.

Gnerre,S. *et al.* (2011) High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. USA*, **108**, 1513–1518.

Gordon,D. *et al.* (2001) Automated finishing with autofinish. *Genome Res.*, **11**, 614–625.

Gurevich,A. *et al.* (2013) QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**, 1072–1075.

Hackl,T. *et al.* (2014) proovread: large-scale high-accuracy PacBio correction through iterative short read consensus. *Bioinformatics*, **30**, 3004–3011.

Howison,M. *et al.* (2013) Toward a statistically explicit understanding of de novo sequence assembly. *Bioinformatics*, **29**, 2959–2963.

Hu,X. *et al.* (2012) pIRS: Profile-based Illumina pair-end reads simulator. *Bioinformatics*, **28**, 1533–1535.

Huang,X. *et al.* (2006) Application of a superword array in genome assembly. *Nucleic Acids Res.*, **34**, 201–205.

Koren,S. *et al.* (2012) Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol.*, **30**, 693–700.

Kosugi,S. *et al.* (2013) Coval: improving alignment quality and variant calling accuracy for next-generation sequencing data. *PLoS One*, **8**, e75402.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Li,H. (2012) Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics*, **28**, 1838–1844.

Li,R. *et al.* (2010a) The sequence and de novo assembly of the giant panda genome. *Nature*, **463**, 311–317.

Li,R. *et al.* (2010b) De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.*, **20**, 265–272.

Lin,S.H. and Liao,Y.C. (2013) CISA: contig integrator for sequence assembly of bacterial genomes. *PLoS One*, **8**, e60843.

Luo,R. *et al.* (2012) SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience*, **1**, 18.

Lysholm,F. *et al.* (2011) An efficient simulator of 454 data using configurable statistical models. *BMC Res. Notes*, **4**, 449.

Margulies,M. *et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.

Miller,J.R. *et al.* (2010) Assembly algorithms for next-generation sequencing data. *Genomics*, **95**, 315–327.

Nijkamp,J. *et al.* (2010) Integrating genome assemblies with MAIA. *Bioinformatics*, **26**, i433–439.

Noe,L. and Kucherov,G. (2005) YASS: enhancing the sensitivity of DNA similarity search. *Nucleic Acids Res.*, **33**, W540–W543.

Ono,Y. *et al.* (2013) PBSIM: PacBio reads simulator—toward accurate genome assembly. *Bioinformatics*, **29**, 119–121.

Peng,Y. *et al.* (2012) IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, **28**, 1420–1428.

Piro,V.C. *et al.* (2014) FGAP: an automated gap closing tool. *BMC Res. Notes*, **7**, 371.

Ross,M.G. *et al.* (2013) Characterizing and measuring bias in sequence data. *Genome Biol.*, **14**, R51.

Ross,M.G. *et al.* (2013) Characterizing and measuring bias in sequence data. *Genome Biol.*, **14**, R51.

Salzberg,S.L. *et al.* (2012) GAGE: a critical evaluation of genome assemblies and assembly algorithms. *Genome Res.*, **22**, 557–567.

Sommer,D.D. *et al.* (2007) Minimus: a fast, lightweight genome assembler. *BMC Bioinformatics*, **8**, 64.

Soto-Jimenez,L.M. *et al.* (2013) GARM: genome assembly, reconciliation and merging pipeline. *Curr. Top. Med. Chem.*, **14**, 418–424.

Soueidan,H. *et al.* (2013) Finishing bacterial genome assemblies with Mix. *BMC Bioinformatics*, **14**, S16.

Treangen,T.J. and Salzberg,S.L. (2012) Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet.*, **13**, 36–46.

Tsai,I.J. *et al.* (2010) Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps. *Genome Biol.*, **11**, R41.

Vicedomini,R. *et al.* (2013) GAM-NGS: genomic assemblies merger for next generation sequencing. *BMC Bioinformatics*, **14**, S6.

Yao,G. *et al.* (2012) Graph accordance of next-generation sequence assemblies. *Bioinformatics*, **28**, 13–16.