OXFORD (GIGA)n SCIENCE

GigaScience, 2022, **11**, 1–12
DOI: 10.1093/gigascience/giab100
TECH NOTE

DENTIST—using long reads for closing assembly gaps at high accuracy

Arne Ludwig [1,2], Martin Pippel [1,2], Gene Myers [1,2] and Michael Hiller [1,2,3,4,5,6,*]

[1]Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany
[2]Center for Systems Biology Dresden, Pfotenhauerstr. 108, 01307 Dresden, Germany
[3]Max Planck Institute for the Physics of Complex Systems, Nöthnitzer Str. 38, 01187 Dresden, Germany
[4]LOEWE Centre for Translational Biodiversity Genomics, Senckenberganlage 25, 60325 Frankfurt, Germany
[5]Senckenberg Research Institute, Senckenberganlage 25, 60325 Frankfurt, Germany
[6]Goethe-University, Faculty of Biosciences, Max-von-Laue-Str. 9, 60438 Frankfurt, Germany
*__Correspondence address.__ Michael Hiller, LOEWE Centre for Translational Biodiversity Genomics, Senckenberganlage 25, 60325 Frankfurt, Germany, Tel: +49 69 7542-1398; E-mail: Michael.Hiller@senckenberg.de

Abstract

Background: Long sequencing reads allow increasing contiguity and completeness of fragmented, short-read–based genome assemblies by closing assembly gaps, ideally at high accuracy. While several gap-closing methods have been developed, these methods often close an assembly gap with sequence that does not accurately represent the true sequence.

Findings: Here, we present DENTIST, a sensitive, highly accurate, and automated pipeline method to close gaps in short-read assemblies with long error-prone reads. DENTIST comprehensively determines repetitive assembly regions to identify reliable and unambiguous alignments of long reads to the correct loci, integrates a consensus sequence computation step to obtain a high base accuracy for the inserted sequence, and validates the accuracy of closed gaps. Unlike previous benchmarks, we generated test assemblies that have gaps at the exact positions where real short-read assemblies have gaps. Generating such realistic benchmarks for *Drosophila* (134 Mb genome), *Arabidopsis* (119 Mb), hummingbird (1 Gb), and human (3 Gb) and using simulated or real PacBio continuous long reads, we show that DENTIST consistently achieves a substantially higher accuracy compared to previous methods, while having a similar sensitivity.

Conclusion: DENTIST provides an accurate approach to improve the contiguity and completeness of fragmented assemblies with long reads. DENTIST's source code including a Snakemake workflow, conda package, and Docker container is available at https://github.com/a-ludi/dentist. All test assemblies as a resource for future benchmarking are at https://bds.mpi-cbg.de/hillerlab/DENTIST/.

Keywords: genome assembly, long sequencing reads, assembly gaps

Introduction

The quality of a genome assembly has an important effect on the quality of any downstream genomic analysis. High contiguity, completeness, and accuracy of genome assemblies are fundamental to (i) comprehensively annotating genes, their promoters, and other functional genomic elements; (ii) understanding genomic repeat content and organization; (iii) performing phylogenomic analysis; (iv) linking phenotypic differences to genomic differences; (v) mapping transcriptomic or epigenetic read data to an assembly; and ultimately using genomes for evolutionary and biomedical studies [1–3]. However, assembly of complex genomes is a challenging task because sequencing reads are much shorter than chromosomes and large parts of eukaryotic genomes consist of repetitive regions [4]. Facilitated by advances in short-read sequencing that drastically increased throughput and sharply decreased costs, genomes of numerous species have been sequenced with short-read technologies such as Illumina instruments, illustrated by the Zoonomia and B10K project, which generated new assemblies for 131 mammals and 267 birds, respectively [5, 6]. However, such reads are too short to span many repetitive regions, resulting in rather fragmented assemblies with short contigs (contiguous sequences) despite high read coverage. Scaffolding methods such as mate pair sequencing, chromosome conformation capture (Hi-C) read pairs, or optical maps can order and orient contigs into long, sometimes chromosome-scale scaffolds. Nevertheless, owing to a large number of assembly gaps, i.e., unknown genomic sequence between neighboring contigs, such assemblies remain inherently incomplete.

Pacific Biosciences (PacBio) or Oxford Nanopore Technologies instruments enable long sequencing reads, which can cover many kilobases and sometimes exceed 1 Mb in length. These reads are often longer than genomic repeats, which enables the assembly of highly complete and highly contiguous genomes, and are therefore increasingly used for *de novo* assembly [1–3, 7]. However, *de novo* long-read–based assembly requires a high coverage. A coverage of ∼60× is typically recommended for PacBio error-prone CLR (continuous long read) reads and ∼30–40× is typically recommended for the more expensive PacBio HiFi (high-fidelity) reads, which can be a significant cost factor. In addition to *de novo* assembly, a lower coverage of long reads can provide a more cost-efficient means to improve the contiguity (contig lengths) of existing short-read assemblies by bridging between neighboring con-

Received: September 21, 2021. **Revised:** December 7, 2021. **Accepted:** December 15, 2021

tigs and thus closing assembly gaps. Furthermore, gap filling is also one of the final steps in state-of-the-art pipelines to generate high-quality long-read assemblies [1]. To close gaps in existing fragmented assemblies with more limited long-read coverage, several methods have been developed in the past, exemplified by PBJelly [8], FinisherSC [9], PacBio GenomicConsensus (PacBio GC [10]) comprising the Arrow and Quiver algorithm implemented in the variantCaller tool, LR_gapcloser [11], and TGS-Gapcloser [12]. These methods align a set of given long reads to an input short-read assembly, determine which reads span assembly gaps, and close these gaps with the new sequence. However, as we demonstrate below, even for shorter and less repeat-rich genomes, these tools lack high accuracy when closing assembly gaps. Thus, while applying these tools improves assembly contiguity, they compromise the quality of the resulting assembly, which can impair downstream analyses.

Here, we developed DENTIST, a new, reliable and sensitive gap-closing method that unlike previous methods achieves a high level of accuracy. Using various simulated and real datasets for genomes ranging from small (*Drosophila*) to large and complex (human), and using realistic assembly gap loci, we demonstrate that DENTIST consistently achieves the highest accuracy compared to other state-of-the-art approaches, while having a similar or better runtime and memory consumption.

Results

Overview of DENTIST

DENTIST implements a full gap-closing pipeline (Fig. 1) and was developed with the main consideration of closing assembly gaps at a very high accuracy. Conceptually, given an assembly and a set of long reads, gap closing starts by aligning the reads to the assembly to identify those that reach into or span assembly gaps. The reads are then used to infer the DNA sequence that should be used to fill these gaps.

DENTIST differs from previous approaches in the following key aspects. First, a key step in accurately closing gaps is to align the long reads to the correct loci in the input assembly. This is not a trivial task because genomic repeats can lead to ambiguous read alignments and potentially assigning reads to the wrong genomic locus. Therefore, DENTIST integrates 4 approaches to identify repetitive regions in both the input assembly and the input long reads, and explicitly uses this repeat mask to identify reliable read alignments. Many of the remaining alignment ambiguities and conflicts are resolved using a scaffolding graph. Second, because long reads generally have a high base error rate, it is important to determine an accurate consensus sequence from the long reads that span or overlap an assembly gap before closing the gap. DENTIST uses a state-of-the-art reference-based consensus caller to generate high-quality consensus sequence for each closed assembly gap, maintaining a high base accuracy in the final assembly. Third, aiming at a high accuracy, DENTIST validates both the consensus sequence as well as the closed gap by aligning the input reads again to the gap-closed assembly, and does not perform questionable gap closures in favor of a correct result.

Test procedure

To assess the performance of DENTIST and compare it to existing methods, we followed previous strategies to generate a "ground truth scenario" where assembly gaps were inserted into a high-quality assembly and long reads were used to close the created assembly gaps. This general strategy allows one to compare sensitivity (number of closed gaps, increase in assembly contiguity) and, because the real gap sequence is known, the accuracy of the inserted sequence.

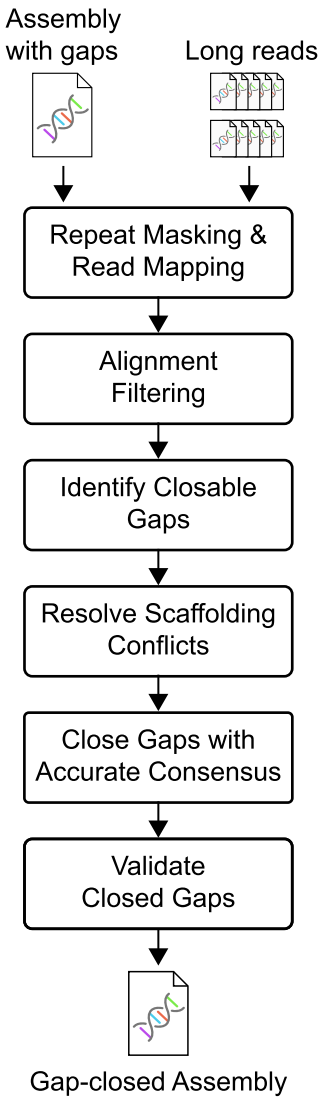

Figure 1: Overview of the assembly gap-closing pipeline implemented in DENTIST.

Unlike previous comparisons, we devised a ground truth scenario, where the assembly gap sizes and loci are as realistic as possible, which is important to assess real-life performance. Previous comparisons introduced assembly gaps at random locations [8] or replaced repeats by assembly gaps of the same length [11]. By excising repeats, the latter approach results in gap flanks having often unique, non-repetitive sequence that make accurate read alignment easier. Consequently, both approaches do not reflect the complexity of assembly gap sizes and locations in reality, and the fact that contig ends often reach into the unbridgeable repeat (as opposed to ending right before it). To overcome this, we devised a procedure that uses 2 real assemblies of the same species. A real high-quality assembly constitutes the "ground truth" and a real fragmented short-read assembly is used to obtain realistic assembly gap locations. Specifically, we aligned the fragmented assembly to the ground truth assembly and introduced gaps into the ground truth assembly at the exact position and size where gaps occurred in the short-read assembly (Fig. 2). Then we used

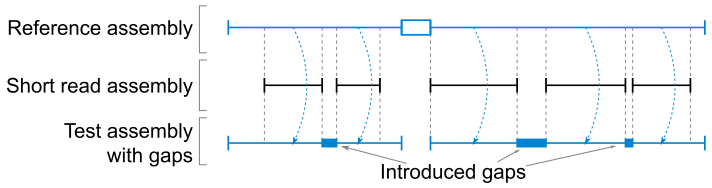

Figure 2: Realistic ground truth scenarios are created by "copying" gaps from a real short-read assembly to a real high-quality reference assembly. Horizontal lines represent contigs, and blue boxes represent genomic regions of the reference assembly, which were replaced with assembly gaps (N's). The white box in the reference assembly represents an assembly gap in the reference, which is not copied to the test assembly.

either long reads that were sampled (simulated) from the ground truth assembly or real PacBio reads of the same species (Supplementary Table S1) to close assembly gaps and evaluate sensitivity and correctness.

In the following comparison, we applied DENTIST and 5 other state-of-the-art methods, PBJelly [8], FinisherSC [9], PacBio GenomicConsensus [10], LR_gapcloser [11], and TGS-Gapcloser [12], to 4 species with various genome sizes and repeat content, representing the insect, plant, and vertebrate lineage: the fruit fly *Drosophila melanogaster* with a 134 Mb assembly where ∼10% of the ground truth assembly was covered by DENTIST's repeat masks and 1,340 gaps were introduced, the thale cress *Arabidopsis thaliana* (119 Mb, ∼11% repeats, 219 gaps), the hummingbird *Calypte anna* (1 GB, ∼2% repeats, 37,501 gaps), and human (3 Gb, ∼20% repeats, 5,846 gaps). Sources and details of all assemblies and reads, as well as statistics of introduced gaps, are listed in Supplementary Table S1. We were only able to test FinisherSC on the smallest genome (*Drosophila*) because this method failed on *Arabidopsis* with simulated reads, did not finish within 2 days on *Arabidopsis* with real PacBio reads, and required >1 TB of RAM for larger genomes. Because gap and assembly properties differ between the 4 species, we note that the following comparisons of different methods only provide a relative performance assessment on the same input dataset.

Simulated read data

We first simulated PacBio reads from each ground truth assembly with a typical PacBio error profile and a mean raw read length of 26.7 kb (Supplementary Table S1). For *Drosophila*, *Arabidopsis*, and hummingbird, the gap sequence inserted by DENTIST is on average ≥99.92% identical to ground truth sequence (Table 1, Fig. 3A). In comparison, the sequences inserted by LR_gapcloser, PBJelly, and TGS-Gapcloser had a lower mean identity to the ground truth sequence (∼87.7% for LR_gapcloser, 88.61–95.17% for PBJelly, 92.81–97.26% for TGS-Gapcloser). FinisherSC, which we could only run on the *Drosophila* dataset, achieved an identity of only 81.53% (Table 1). For the 3-Gb human genome, the mean sequence identity for DENTIST was 98.70%, whereas LR_gapcloser (88.01%), PBJelly (89.23%), and TGS-Gapcloser (93.63%) achieved lower mean identity values. In addition to a high average identity, DENTIST closed 46.1% (human) to 91.5% (*Drosophila*) of the gaps with a 100% accurate sequence, while other methods at best closed 17.6% (TGS-Gapcloser, 236 of 1,340 gaps for *Drosophila*) at 100% accuracy.

To account for the different sizes of assembly gaps, we also calculated the mean identity weighted by the size of the gap. This metric showed highly similar values (Table 1), with the exception of PBJelly and TGS-Gapcloser, where the weighted mean identity increased by 0.94–4.8% and 0.34–3.16%, respectively, indicating that larger gaps tended to be closed with a higher accuracy (discussed below). Nevertheless, the maximum weighted mean of PBJelly (96.11% for *Arabidopsis*) and TGS-Gapcloser (97.90% for hummingbird) is lower than the accuracy of DENTIST (always ≥99.72%). In summary, consistently for all assemblies, DENTIST achieves the highest accuracy.

Comparing sensitivity, we found that all methods closed a similar percentage of gaps for *Drosophila* (89.6–96.1%, except FinisherSC with 4.3%), *Arabidopsis* (92.2–98.6%), and hummingbird (90.4–98.7%, except PBJelly with 22.3%) (Table 1, Fig. 3A). TGS-Gapcloser closed the most gaps for *Drosophila* (96.1%) and *Arabidopsis* (98.6%), while DENTIST closed the most gaps for hummingbird (98.7%). For the larger human assembly, TGS-Gapcloser closed the most gaps (92.8%) followed by DENTIST (79.0%). To measure the increase in assembly contiguity, we computed the contig NG50 value, using the number of real bases (A, C, G, T) in the ground truth assembly as a fixed reference. This allows us also to compute the maximally possible contig NG50 that can be achieved by closing all introduced gaps in the ground truth assembly. It should be noted that contig NG50 increase is not a perfect measure of sensitivity because the NG50 increase heavily depends on the position of closed gaps and reflects only indirectly the number of closed gaps. For example, closing a single gap between 2 large contigs could result in a larger NG50 increase than closing several gaps between several small contigs. Indeed, this effect is also visible in our comparisons, exemplified by the hummingbird assembly where LR_gapcloser achieved a higher contig NG50 (14.6 vs 12.9 Mb) despite DENTIST closing 3,123 more gaps (Table 1). Despite these caveats, comparing contig NG50 increase, we found that LR_gapcloser achieved the highest contig NG50 for *Drosophila* and hummingbird, PBJelly achieved the highest contig NG50 for *Arabidopsis*, and DENTIST achieved the highest contig NG50 for human (Table 1). For all 4 species, DENTIST substantially increased the contig NG50 between 5-fold (*Arabidopsis*) and 263-fold (hummingbird). We note that PBJelly outputs an *Arabidopsis* assembly with a significantly higher contig NG50 value than what is achievable by correctly closing all introduced gaps (8.3 vs 7.1 Mb), which indicates that some gaps may be "overfilled." Because NG50 reduces contiguity to a single number, we also plotted which percent of the assembly is contained in contigs exceeding a certain size (Fig. 3C). These NG(x) plots showed that DENTIST, PBJelly, LR_gapcloser, and TGS-Gapcloser achieve similar improvements in contiguity for *Drosophila* and *Arabidopsis*, that DENTIST and LR_gapcloser achieve a higher contiguity for hummingbird, and that DENTIST achieves a higher contiguity for human. In summary, our comparisons using simulated reads show that the sensitivity of DENTIST is comparable to and sometimes better than other methods and that DENTIST excels in a very high accuracy.

Real PacBio long-read data

Because simulated reads cannot capture the full diversity of issues that are present in real PacBio reads (e.g., chimeras, low-quality regions, missed adapters) and further do not encompass the whole genome but were only sampled from the ground truth

Table 1: Comprehensive benchmarking of gap-closing methods

Species	Read type	Tool	ID (%)	wID (%)	No. gaps closed with sequence identity						Broken gaps	Existing gaps	Closable gaps	Contig NG50 (kb)		
					=100%	≥99%	≥95%	≥90%	≥70%	Total				Input	Output	Max
Drosophila	Simulated	DENTIST	**99.94**	**99.96**	**1,226**	**1,242**	**1,269**	**1,270**	**1,270**	1,270	0	1,340	1,340	238	8,811	21,486
		FinisherSC	81.53	80.76	0	0	3	16	54	58	1,172				265	
		LR_gapcloser	87.69	88.54	10	10	85	430	1,193	1,201	5				**13,484**	
		PBJelly	88.56	92.52	49	61	428	710	1,241	1,280	0				6,442	
		TGS-GapCloser	92.81	95.24	236	310	777	1,098	1,235	**1,288**	3				7,356	
	Real	DENTIST	**99.95**	**99.96**	**1,185**	**1,204**	1,220	1,221	1,222	1,222	0	1,292		188	2,837	13,637
		FinisherSC	78.44	85.97	0	0	0	6	18	21	1,183				198	
		LR_gapcloser	89.25	89.42	25	25	162	598	1,205	1,216	6				7,834	
		PacBio GC	99.53	98.54	1,114	1,158	**1,255**	**1,267**	**1,271**	**1,275**	0				6,422	
		PBJelly	91.58	94.47	80	85	546	917	1,245	1,261	0				3,906	
		TGS-GapCloser	95.16	38.98	370	448	928	1,202	1,243	1,272	3				**13,643**	
Arabidopsis	Simulated	DENTIST	**99.95**	**99.97**	**146**	**204**	**206**	**206**	206	206	0	219	217	1,131	6,092	7,139
		LR_gapcloser	87.75	87.60	3	3	6	17	210	210	0				6,092	
		PBJelly	95.18	96.09	1	2	176	190	201	202	0				**8,320**	
		TGS-GapCloser	97.26	97.60	7	113	187	201	**215**	**216**	0				6,092	
	Real	DENTIST	**99.40**	**99.93**	**126**	**181**	182	184	186	187	0	207		1,238	6,367	6,367
		LR_gapcloser	88.26	87.17	7	7	10	49	189	193	0				**6,368**	
		PacBio GC	95.16	94.88	31	40	133	156	171	175	0				5,653	
		PBJelly	94.64	95.32	1	1	117	179	186	188	0				4,294	
		TGS-GapCloser	96.57	89.26	10	140	**183**	**192**	**201**	**206**	0				6,367	
Hummingbird	Simulated	DENTIST	**99.92**	**99.94**	**32,586**	**36,380**	**36,933**	**37,007**	**37,024**	**37,029**	0	37,501	37,438	49	12,935	14,522
		LR_gapcloser	87.71	87.92	655	656	1,628	7,877	33,712	33,906	18				**14,552**	
		PBJelly	93.48	95.76	152	255	5,364	7,031	8,258	8,360	66				64	
		TGS-GapCloser	94.74	97.90	3,411	12,263	25,714	30,843	34,894	35,830	2				1,612	
	Real	DENTIST	**98.43**	**98.17**	**16,252**	**26,708**	**33,560**	**34,528**	35,200	35,531	0	37,501		49	2,230	14,522
		LR_gapcloser	88.26	88.15	994	998	2,663	13,986	34,178	34,621	19				**14,079**	
		PacBio GC	96.80	95.82	13,116	17,565	32,010	34,029	**35,330**	36,199	2				2,237	
		PBJelly	92.72	93.95	737	838	14,742	27,745	31,856	32,470	12				339	
		TGS-GapCloser	95.55	96.89	4,539	11,492	28,615	33,503	36,794	**37,445**	2				13,412	
Human	Simulated	DENTIST	**98.70**	**99.72**	**2,694**	**3,068**	**4,369**	**4,585**	4,616	4,617	0	5,846	5,823	1,700	**28,837**	50,761
		LR_gapcloser	88.01	87.31	118	122	411	1,429	4,482	4,520	145				20,998	
		PBJelly	89.23	94.04	93	114	1,440	2,478	3,818	3,990	37				6,842	
		TGS-GapCloser	93.63	95.95	463	1,471	3,365	4,456	**5,293**	**5,427**	3				15,258	
	Real	DENTIST	**96.64**	**98.00**	1,282	**1,768**	**3,465**	**4,085**	4,306	4,340	0	5,839		1,600	12,008	71,707
		LR_gapcloser	90.38	88.98	375	379	949	2,970	5,003	5,051	62				**23,718**	
		PacBio GC	92.79	91.39	**1,353**	1,580	3,019	3,736	4,345	4,633	0				19,698	
		PBJelly	89.19	91.90	202	208	1,312	2,651	3,792	4,021	19				6,271	
		TGS-GapCloser	91.31	92.18	234	868	2,825	4,072	**5,227**	**5,491**	5				23,263	

We define %ID as the mean across all sequence identities of closed gaps and w%ID as the mean weighted by the true gap length. Gaps are broken if the flanking contigs are not adjacent anymore after gap closing, indicating a misassembly. For simulated reads, we label a gap as closable if there are ≥3 simulated reads that span the gap and ≥500 bp of its flanks. The contig NG50 value of the test (input) and output assembly is given on the right; maximum contig NG50 refers the value if all introduced gaps would be closed. Best values in each category are in boldface.

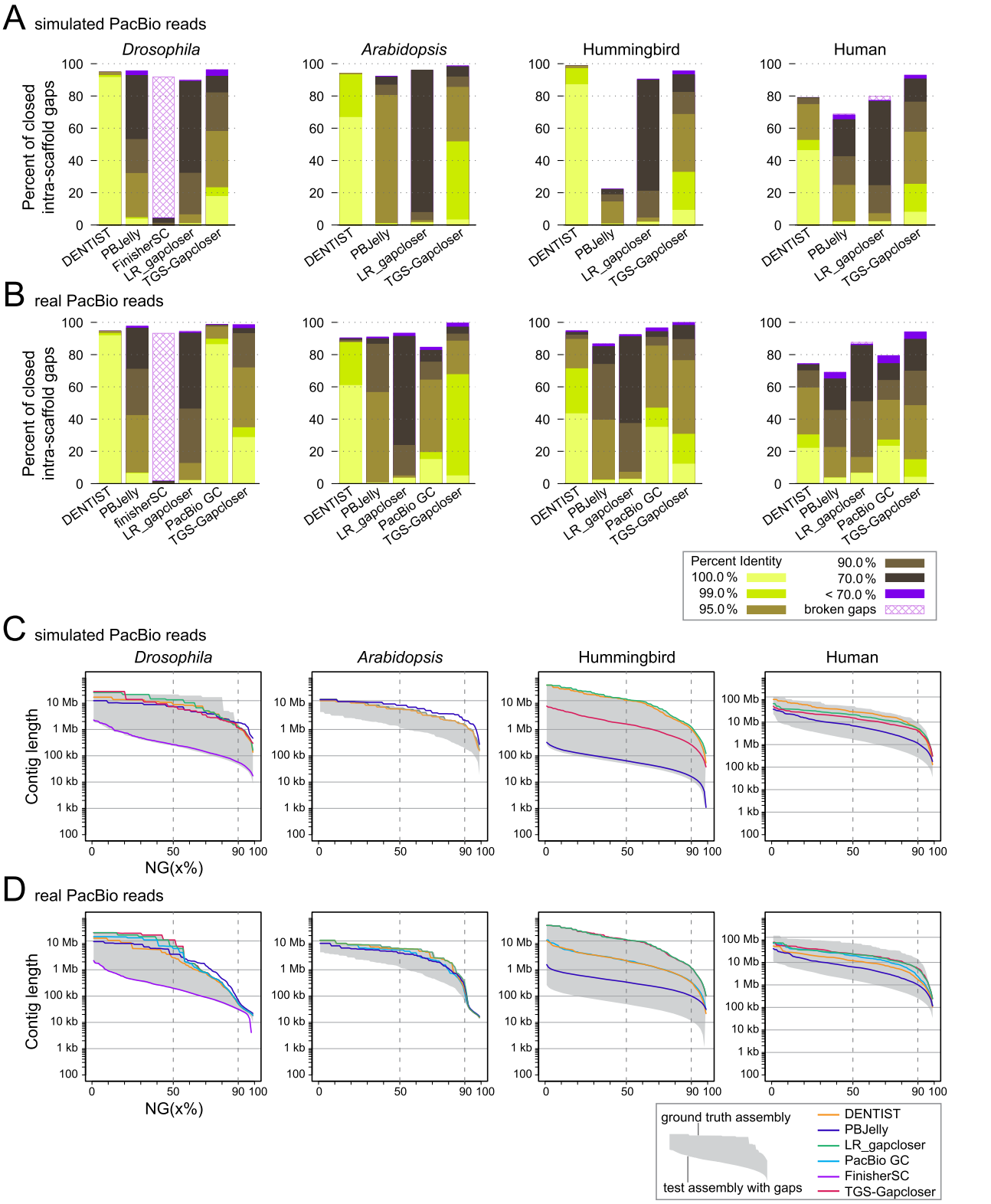

Figure 3: Comparison of accuracy and sensitivity of gap-closing methods. (A, B) Bar charts show the percent of gaps that are closed by different methods using simulated (A) or real (B) PacBio reads and a breakdown of how identical the inserted sequence is to the ground truth sequence. (C, D) NG(x) plots show which percent of the assembly consists of contigs of a certain minimum size (y-axis). The contig NG50 and NG90 values are indicated by vertical dashed lines. The grey area represents the NG(x) of the test assembly (contains introduced assembly gaps, lower bound) and the ground truth assembly (representing the maximally possible contiguity if all introduced gaps are closed, upper bound). For *Drosophila* (simulated and real reads) and *Arabidopsis* (simulated reads), PBJelly shows NG(x) values higher than the values of the ground truth assembly, indicating that some gaps are "overfilled." Owing to excessive memory consumption, we could run FinisherSC only on the *Drosophila* genome and this method produced many broken gaps (likely representing misassemblies because the input contigs are not adjacent anymore in the output assembly). PacBio GC requires real read data as input and is therefore only shown in panel D. For *Arabidopsis* and real PacBio reads (panel D, second column), the TGS-Gapcloser line overlaps the LR_gapcloser line and follows the upper bound.

contigs, we next evaluated the performance of DENTIST and other gap-closing methods on the same species but using real CLR PacBio reads (Table 1). Read length, gap, and ground truth assembly statistics are listed in Supplementary Table S1. We added the PacBio GC method to the tests, which requires additional sequencing metrics (such as pulse widths and inter-pulse durations) as input and can therefore only be applied to real read data. We note that in these tests, it will be harder to achieve a very high accuracy because real reads also contain single-nucleotide polymorhisms or larger haplotype variation and the ground truth assembly may contain base or assembly errors with respect to the real reads. Nevertheless, the accuracy of DENTIST is still very high, with a mean identity between inserted and ground truth sequence of 99.95% for *Drosophila*, 99.40% for *Arabidopsis*, 98.43% for the hummingbird, and 96.64% for human (Table 1, Fig. 3B). All other methods have a lower accuracy. The second best methods are PacBio GC for *Drosophila* (99.53%), hummingbird (96.80%), and human (92.79%) and TGS-Gapcloser for *Arabidopsis* (96.57%). A few examples of closed gaps are illustrated in Fig. 4.

For human, DENTIST's mean identity weighted by gap size is noticeably higher than the unweighted identity (98.00% vs 96.64%), indicating that larger gaps tend to be closed with a higher accuracy while some of the smaller gaps were closed less accurately. We confirmed this by plotting identity vs gap size (Fig. 5). Manual inspection showed that many of these less accurate cases comprise gaps <50 bp consisting of simple repeats such as homopolymer runs, for which long reads have a higher error rate [14], making it more difficult to compute an accurate consensus. Because very few incorrect bases in the consensus of a small gap have a large effect on the mean identity (e.g., 19 of 20 correct bases is an identity of only 95%), the weighted mean identity provides a better measure of overall sequence accuracy. Furthermore, a few base errors in a small gap may not be a serious problem because small gaps can easily be "polished" with Illumina reads in a downstream step (discussed below). A consistently higher weighted mean identity was also observed for PBJelly (Table 1, Fig. 5B). For TGS-Gapcloser, the weighted identity is significantly reduced for *Drosophila* (95.16% vs 38.98%) and *Arabidopsis* (96.57% vs 89.26%), suggesting that some large gaps were filled with a low accuracy. In summary, considering both the weighted and unweighted mean identity of the inserted sequence, DENTIST achieves the highest accuracy consistently for all 4 assemblies also for real PacBio CLR reads.

In terms of sensitivity, TGS-Gapcloser closed the most gaps for all assemblies except *Drosophila*, where PacBio GC closed 3 additional gaps. TGS-Gapcloser achieved the highest contig NG50 increases for *Drosophila* and LR_gapcloser for the other 3 assemblies (Table 1, Fig. 3D). DENTIST closed 74% (human) to ∼95% (*Drosophila* and hummingbird) of the gaps, achieving the maximally possible contig NG50 for *Arabidopsis* and otherwise increasing NG50 7.5-fold (human) to 45.3-fold (hummingbird). Thus, consistent with the simulated read data, DENTIST has a reasonable sensitivity and the highest accuracy also when using real reads.

Runtime and memory consumption

While a high accuracy and contiguity are certainly the most important objectives, speed and memory consumption of a gap-closing method determine how large the required computational resources must be. We used the human dataset with real PacBio reads to compare the total CPU time and the maximum memory usage (measured as maximum resident set size across all jobs) of the gap-closing tools. As shown in Table 2, TGS-Gapcloser is by far the fastest method (15.4 CPU hours), followed by DENTIST (255.5 CPU hours). TGS-Gapcloser required a slightly smaller amount of memory than DENTIST (24.6 vs 25.7 Gb). All other methods require significantly more memory and runtime. With this speed and memory consumption, DENTIST can finish a mammalian genome such as human within 1 day on a 16-core workstation and within a few hours on a compute cluster, giving it also a good turnaround time for testing.

Table 2: Comparison of runtime and maximum memory consumption on the human assembly with real PacBio reads

Tool	Total CPU Time (h)	Maximum memory consumption (GiB)
DENTIST	255.5	25.7
LR_gapcloser	289.4	96.8
PacBio GC	5,904.2	40.6
PBJelly	2.285.0	53.9
TGS-Gapcloser	**15.4**	**24.6**

For DENTIST, we did not count the runtime of local Snakemake rules, which consume very little resources. For PacBio GC, we excluded resources needed for the expensive pre- and post-processing of the assembly but included the essential calls of samtools faidx, pbindex, pbalign, and variantCaller. GiB = Gibibyte (binary gigabyte = 2^{30} bytes).

Read coverage analysis

Another relevant consideration is how much coverage in long reads needs to be generated to close assembly gaps and thus improve an existing fragmented assembly. A higher read coverage likely allows one to close more gaps but also increases the sequencing costs. To evaluate how DENTIST's performance is affected by coverage, we ran our method with varying coverage of simulated PacBio reads on the *D. melanogaster* assembly test case. As shown in Fig. 6, the number of closed gaps starts to plateau above a read coverage of ∼15×. Also, at >15× the majority of gaps are closed with an accuracy ≥99%. Higher coverages increase the percent of gaps closed with ≥99% accuracy because more reads facilitate the construction of a highly accurate consensus sequence. In summary, while assembly improvements are also possible with low coverages of 5× or 10×, a coverage between 15× and 20× seems to be a good trade-off between sequencing cost and power to accurately close most assembly gaps. Importantly, this coverage is substantially less than the recommended coverage of ∼60× that is typically required for *de novo* assembly, providing a cost-effective alternative.

Discussion

We have presented a novel method, DENTIST, that uses uncorrected, long sequencing reads to close gaps in fragmented assemblies. DENTIST was developed with the main goal of closing assembly gaps at a very high accuracy, for which it implements a repeat-aware read alignment step to map reads to the correct assembly loci, a consensus sequence step to obtain an accurate sequence to fill a gap, and a final validation step. Our tests using simulated and real PacBio long-read data show that our method is substantially more accurate than existing tools, while achieving good sensitivity. Furthermore, DENTIST is sufficiently fast and memory-efficient to close gaps in a reasonable amount of time also in larger assemblies such as human. Together, these features make DENTIST appropriate for the task of improving the quality

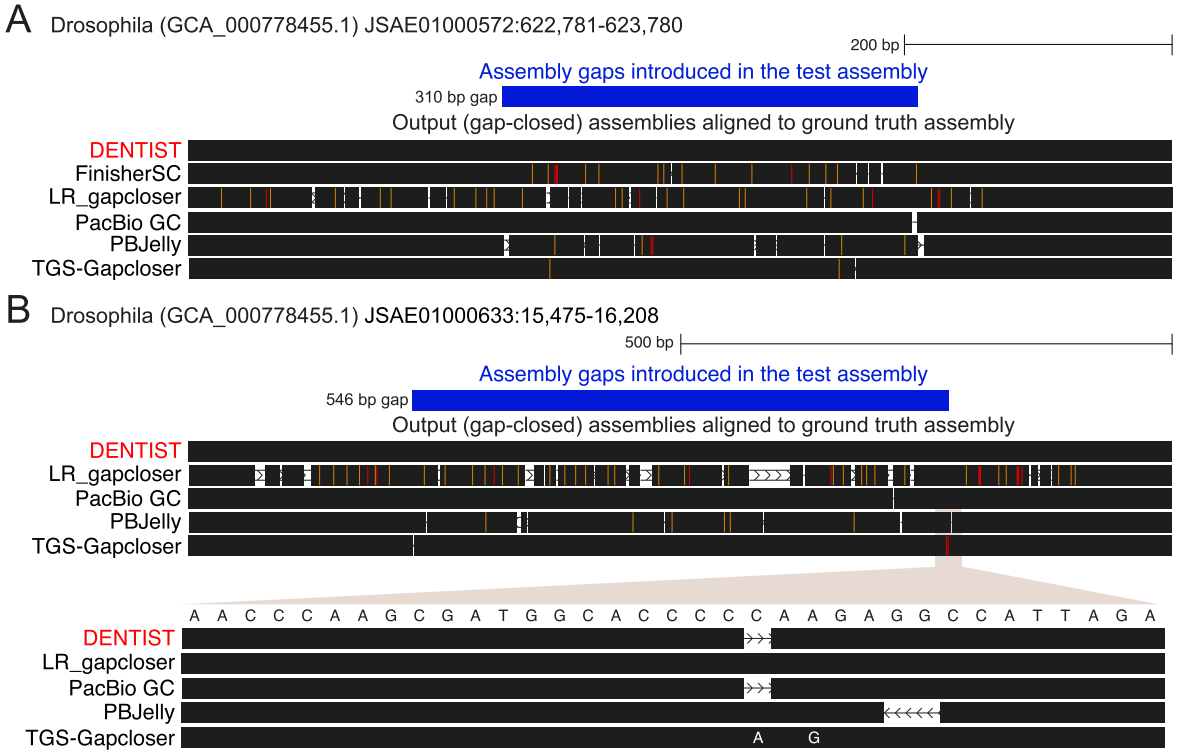

Figure 4: Examples of gaps in the *Drosophila* assembly that are closed with real PacBio reads. UCSC genome browser [13] visualizations show a part of the ground truth *Drosophila* assembly overlapping an assembly gap that we introduced in the test assembly (blue). The output assemblies produced by different gap-closing methods are aligned to the ground truth assembly, highlighting base differences in colour (deletions in white, insertions in orange, substitutions in red) and identical sequence parts in black. (A) While DENTIST closes the 310-bp gap with a sequence that is 100% identical to the ground truth, other methods introduce a few base errors (98.7% identity for TGS-Gapcloser and PacBio GC; lower for other methods). (B) DENTIST closes the 546-bp gap with a sequence that is 99.8% identical to the ground truth, while other methods have slightly lower identity values (PacBio GC 99.5%, TGS-Gapcloser 99.1%, PBJelly 96.4%, LR_gapcloser 84.2%). The inset shows that the single base error in the DENTIST output is a deletion of a "C" in a short homopolymer run of C's, and that PacBio GC makes the same mistake.

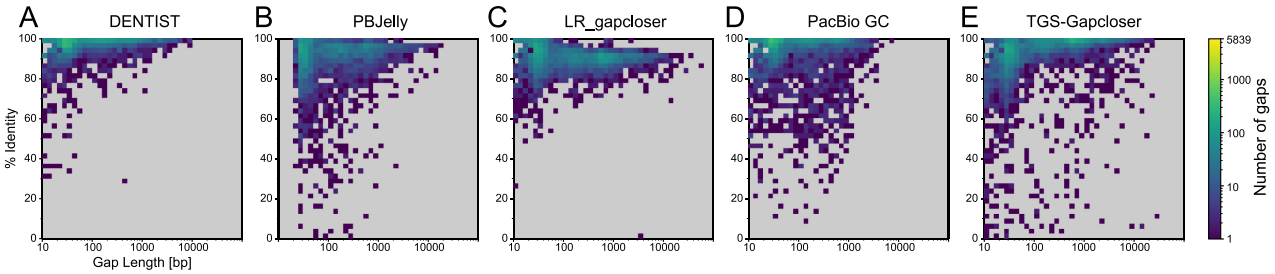

Figure 5: Comparison of base identity and gap size using the human assembly and real PacBio reads. (A) Gaps, which are closed at an accuracy of <90% by DENTIST, tend to be short. Many of these gaps contain simple repeats such as homopolymer runs, for which long reads have a higher error rate. In contrast, gaps closed at a lower accuracy by PBJelly (B), LR_gapcloser (C), PacBio GC (D), and TGS-Gapcloser (E) are more evenly distributed in size. For LR_gapcloser, there is a trend that the longer the gaps, the lower is the accuracy.

of hundreds of existing draft genomes with auxiliary long-read data.

The accuracy of a gap-closing method is mainly influenced by the accuracy of the read mappings and by the ability to determine an accurate consensus from the error-prone long sequencing reads. The latter aspect is generally challenging, and even with high read coverage it is difficult to reach a desired base accuracy of Q40 (99.99% base accuracy) [1]. Therefore, *de novo* genome assembly from long reads includes a final "polishing" step that maps shorter Illumina reads to the finalized assembly to correct remaining base errors [1, 2]. While DENTIST already achieves a high base accuracy of the sequence inserted into gaps, this accuracy is lower than Q40. Therefore, we recommend polishing the gap-closed assembly using Illumina reads after applying DENTIST. Notably, by achieving a base accuracy of 99% or higher, DENTIST facilitates the mapping of shorter Illumina reads, which would be more difficult with the lower accuracies produced by other methods.

To make it easy for users to run DENTIST, we used Snakemake [15] to automate the entire workflow (Fig. 1). This pipeline is built in a modularized manner and is therefore customizable. Furthermore, to enable easy application on a compute cluster without the necessity of complicated software installation steps, we provide DENTIST and all required tools in a Docker container envi-

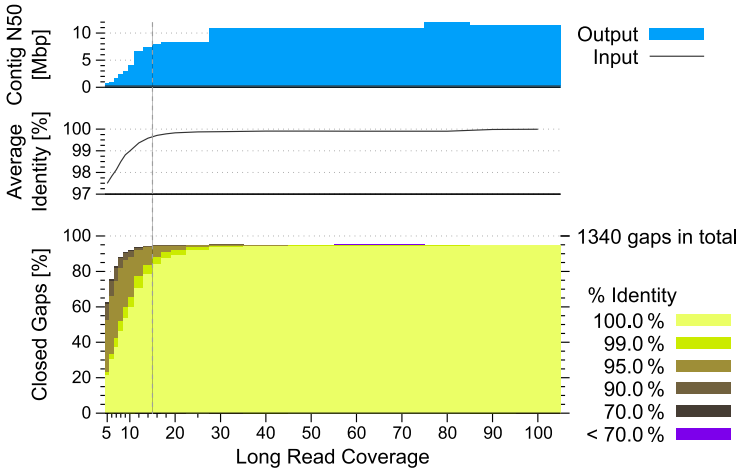

Figure 6: Coverage analysis by applying DENTIST to *Drosophila* with varying coverage of simulated PacBio CLR reads. The figure compares contig NG50 (top), average identity of the closed gaps (middle), and the number of closed gaps together with a breakdown of their sequence identity (bottom). The number of closed gaps starts to plateau above a read coverage of ∼15× (indicated by a vertical dashed line), which is significantly less than the recommended coverage of ∼60× required for *de novo* genome assembly.

ronment that can be easily used with Snakemake's Singularity integration [16]. A conda package is also available. The full source code is available at [17].

Methods

DENTIST parameters

The gap-closing pipeline implemented in DENTIST requires a number of parameters that we empirically optimized with the goal of achieving a very high accuracy on the different test assemblies. All default parameter settings and how they can be adjusted by users via command line parameters are listed in Supplementary Listing S1 and Supplementary Note S1).

Repeat masking and read mapping

Before aligning reads, DENTIST produces 4 types of repeat masks. First, it starts by masking low-complexity regions in the given assembly using DBdust [18, 19] to improve the sensitivity of the daligner alignment algorithm [20]. Second, tandem repeats are identified with datander and TANmask of the DAMASKER suite [21, 22]. Third, to identify other repetitive regions, DENTIST performs a self-alignment of the given assembly using daligner [20] and masks regions covered by ≥4 alignments to other genomic regions (adjustable via DENTIST's parameter –max-coverage-self). Fourth, DENTIST creates another repeat annotation by analysing the coverage of read alignments and marks assembly regions as repetitive that are covered by more reads than expected from the global read coverage (summed length of all long reads divided by genome size). To this end, DENTIST uses the first 3 repeat annotations as a soft mask and aligns all input long reads to the assembly using damapper [23, 24], which outputs chains of local alignments arising from read artefacts such as poor-quality regions or larger haplotype variations. All genomic regions covered by more than C_{max} alignments are considered repetitive. The threshold C_{max} is either given by the user via –max-coverage-reads or calculated from the global read coverage C (provided via –read-coverage) such that the probability of observing more than C_{max} alignments in a unique (non-repetitive) genomic region is very small (Supplementary Table S2). This probability is calculated under the assumption that the reads are sampled uniformly across the genome, implying a Poisson distribution of the number of sampled reads at any position in the genome (probability to observe k reads at any position is $C^k\left(e^{-C}/k!\right)$). Genomic regions with a read coverage higher than C_{max} comprise the first part of the fourth (read alignment-based) repeat annotation. To further increase the sensitivity, DENTIST searches for smaller repeat-induced local alignments. To this end, we define an alignment as "proper" if there are ≤100 bp (adjustable via –proper-alignment-allowance) of unaligned sequence on either end of the read. All other alignments, where only a smaller substring of the read aligns, are called "improper." Improper alignments are often indicative of repetitive regions. For example, an interspersed repeat inside a long read will result in improper alignments to other genomic loci, where a similar repeat copy exists. Therefore, DENTIST considers genomic regions where the number of improper read alignments is higher than a threshold to be repetitive. By default, this threshold equals half the global read coverage C (adjustable via –max-improper-coverage-reads). These genomic regions comprise the second part of the fourth repeat annotation.

These 4 repeat annotations are homogenized by transferring the annotated repeat regions to the reads using the read alignment and back again to the assembly. These homogenized, final repeat annotations are used in the next step of the pipeline.

Alignment filtering

To extract candidate reads that could close assembly gaps from the entire set of read alignments, it is crucial to remove potentially unreliable and irrelevant alignments. A read alignment is categorized as reliable if (i) it is proper (defined above); (ii) it is strongly anchored, i.e., ≥500 bp (adjustable via –min-anchor-length) of the aligned reference sequence are non-repetitive according to the homogenized repeat masks; and (iii) after filtering improper alignments, every region of the aligned read must align at most to 1 assembly region. Finally, (iv) alignments that are fully contained in a reference contig are removed because they are irrelevant for gap closing.

Identifying closable gaps

DENTIST identifies closable gaps by creating a so-called "scaffold graph" that connects input contigs based on the filtered align-

ments of the long reads. For every contig in the assembly, the graph has 4 nodes v^i_{pre}, v^i_{begin}, v^i_{end}, and v^i_{post} representing virtual locations relative to the contig. As illustrated in Supplementary Fig. S1, edges in this graph are interpreted as follows:

- $\{v^i_{begin}, v^i_{end}\}$ represents contig i,
- $\{v^i_{end}, v^j_{begin}\}$ for $i \neq j$ represent either reads that span contig i and j and thus also span the assembly gap between them, or intra-scaffold gaps (gaps between adjacent contigs) in the input assembly, and
- $\{v^i_{pre}, v^i_{begin}\}$ and $\{v^i_{end}, v^i_{post}\}$ represent reads extending the beginning or end of a contig, respectively.

Initially, the scaffold graph is populated with the contig and intra-scaffold gap edges. Then, for every read, edges of the types described above are inferred from its alignments. Note that for long reads that align to >2 consecutive contigs, we do not add the transitive edges (e.g., connecting contig i to $i + 2$). A list of the reads and corresponding alignments is kept for every edge such that they can be retrieved for gap closing.

Resolving scaffolding conflicts

The raw scaffold graph likely still contains some artefacts that need to be cleaned up. Scaffolding conflicts show up as begin nodes with >1 incoming edge or end nodes with >1 outgoing edge, meaning that there is >1 way to connect the respective contig.

For assemblies with small contigs, a common conflict is small cycles resulting from reads that are long enough to completely cover 1 or more contigs and reach into both neighboring contigs. Depending on the local accuracy of the read, the repeat content of the intermediate contigs, and the minimum alignment length, local alignments to intermediate, small contigs may not be reliably detected, resulting in an edge that connects the neighboring contigs but skips the intermediate one(s). However, for other reads with higher accuracy, local alignments to these small intermediate contigs can be detected, resulting in a small cycle in the scaffolding graph. To avoid mistaking these cycles for scaffolding conflicts, DENTIST searches for small cycles (up to 3 intermediate contigs) in the scaffold graph, identifies the "skipping" edge, and aligns the reads from that edge to the intermediate (skipped) contigs with increased alignment sensitivity. If this additional step detects an alignment between the skipping read and the intermediate contig(s), the new alignments will be used to correct the graph; otherwise the read will be discarded, avoiding scaffolding conflicts. This procedure effectively resolves many branching points in the scaffold graph while preserving valuable sequence information.

The remaining scaffolding conflicts are resolved by a crude yet effective heuristic: if there are ≥2 spanning edges at the begin or end node of a contig, then the spanning edge with the highest number of reads will be kept, if this edge is supported by ≥3 times (adjustable via –best-pile-up-margin) as many reads as for the other edges; otherwise all edges in question are discarded. Additionally, to use the given scaffolding information (contig order), edges that are supported by an intra-scaffold gap in the input assembly are given a higher weight by multiplying their read number by a bonus factor (default 6, adjustable via –existing-gap-bonus).

After removing scaffolding conflicts from the graph, DENTIST discards contig-spanning edges with <3 reads because no accurate consensus sequence for the gap can be determined from 1 or 2 reads. Also, more spanning reads give higher confidence about the correctness of the join. The minimum number of spanning reads can be increased via the parameter –min-spanning-reads to achieve a higher confidence. DENTIST also provides an "expert option" to allow gap closing with solitary reads, in which case the raw read sequence would be inserted. For valid contig-spanning edges, DENTIST puts the reads representing the spanning and extending edges into a single pile-up to gather as much sequence information as possible for the consensus procedure.

Closing the gaps

To compute a consensus sequence for a gap, reads assigned to spanning or extending edges are cropped such that all read alignments begin and end at the same position in the reference assembly. The cropping position is chosen such that all reads still overlap the flanking contigs as much as possible to allow validation of the consensus. Cropping reduces artefacts in the subsequent pairwise alignment of the involved reads and allows easy identification of false alignments.

The cropped reads are pairwise aligned to each other using daligner. These read-to-read alignments often contain several local alignments because long reads can contain regions of poor quality or larger indels and because complex gaps can result in local repeat-induced alignments. Before computing a consensus sequence, these local alignments must be filtered and "chained." To this end, we implemented a chaining algorithm that works directly with the alignment output produced by daligner. This algorithm is applied to every read-to-read alignment and reduces the alignment-chaining problem to the shortest paths problem on a directed acyclic graph with node and edge weights. Every local alignment is represented by a node with a negative (beneficial) weight corresponding to the mean number of base pairs covered by the local alignment of the involved reads. To determine edge weights, we define $gap_A(x,y)$ and $gap_B(x,y)$ as the distance between 2 ordered local alignments x and y on long reads A and B, respectively, and gapSizeDiff(x,y) as the absolute difference between $gap_A(x,y)$ and $gap_B(x,y)$. Thus, $gap_A(x,y)$ represents the number of unaligning bases in read A between x and y, which is 0 if both local alignments are adjacent, and gapSizeDiff(x,y) represents the difference in the number of unaligned bases in both reads. Two nodes in the graph are connected by an edge if the respective local alignments x and y are "chainable"; i.e., (i) the first alignment begins strictly before the second alignment and both occur in the same orientation; (ii) gapSizeDiff(x,y) <1,000 bp (adjustable via parameter –max-indel); (iii) $gap_A(x,y)$ and $gap_B(x,y)$ are smaller than 10,000 bp (adjustable via –max-chain-gap); and (iv) the relative overlap between the alignments, determined as the length of the overlapping region divided by the length of the smaller alignment, is ≤0.3 (adjustable via –max-relative-overlap). The edge then gets a positive weight, penalizing the difference between the number of unaligning bases and to a smaller extent the length of the unaligning region between both local alignments as $w_e(x,y) = \text{gapSizeDiff}(x,y) + 0.1 \cdot \max\{|gap_A(x,y)|, |gap_B(x,y)|\}$. Maximal shortest paths in this graph constitute candidates for alignment chains. From all candidate chains, the best scoring chain(s) are selected.

After chaining, intrinsic quality values (QVs) are derived from the alignment chains using DAScover and DASqv [25, 26]. The read with the lowest number of bad QVs is chosen as the reference read, where a QV is defined as bad if it belongs to the 8% worst QVs in the pile-up (adjustable via –bad-fraction). The cleaned up alignment is used as input for daccord [27], which uses a local de Bruijn graph–based approach to com-

pute an accurate consensus based on the selected reference read.

Subsequently, the consensus sequence will be aligned to the flanking contigs to validate its correctness and find the exact insert sites such that the contigs are not modified when closing the gap with the consensus sequence. DENTIST allows gaps to be closed in 3 different modes. In the default mode, DENTIST closes only intra-scaffold gaps that are provided in the input assembly. In the second mode, DENTIST additionally closes gaps between different scaffolds if they are spanned by sufficient long-read evidence (minimum of 3 spanning reads by default). In this mode, both intra- and inter-scaffold gaps are closed. In the third mode, DENTIST uses the existing scaffolding information (if available) only for conflict resolution and freely scaffolds the given contigs using the long reads. The second and third mode enable DENTIST to also improve contig-only assemblies. Note that the selection of candidates for gap closing is only based on the evidence derived from the input read data and does not depend on the chosen mode.

Validation of closed gaps

Aiming at a high accuracy, DENTIST performs a final validation step by mapping the input reads to the gap-closed assembly. For each closed gap, DENTIST analyses the genomic region 1,000 bp (adjustable via –region-context) upstream and downstream of the former gap. A closed gap is validated if there are ≥3 (adjustable via –min-spanning-reads) unchained read alignments spanning this region and the minimum "continuous alignment coverage" exceeds a user-given threshold, definable via the mandatory parameter –min-coverage-reads (alternatively, if the user provides the global long-read coverage via –read-coverage and the ploidy via –ploidy, DENTIST will set the threshold to 50% of the long-read coverage expected to be sequenced from a haploid locus). The continuous alignment coverage with window size w (default 500 bp; adjustable via –weak-coverage-window) at position x is defined as the number of local alignments (unchained and potentially improper) that completely cover the window $[x, x + w)$. The minimum continuous alignment coverage is then obtained by sliding the window across all positions in the genomic region defined above. For closed gaps that are not validated, DENTIST outputs the original gap (NNN…) sequence.

Evaluating the gap-closing accuracy with realistic assembly gap sizes and loci

To assess the accuracy of DENTIST and compare it with other methods, we devised a realistic ground truth scenario, where the true sequence of assembly gaps is considered to be known and where assembly gaps occur at realistic genomic loci and at realistic sizes. This is important because repetitive genomic regions are the main reason for assembly gaps; thus random assembly gap placement or replacing repeats with gaps will not create a realistic setting. Each test scenario comprises (i) a high-quality reference assembly, which we consider as the ground truth; (ii) a test assembly that contains assembly gaps and is input to the gap-closing method; and (iii) a set of long reads that either were simulated using the reference assembly or are real PacBio reads (Supplementary Table S1). Subsequently, we derived performance statistics from the output assembly of a gap closer using the same evaluation strategy.

To generate a test assembly with realistic gap sizes and loci, we aligned the reference assembly to a more fragmented, short-read assembly of the same species and copied gaps from the short-read assembly into the respective position of the reference assembly, as depicted in Fig. 2. Briefly, we aligned both assemblies using lastz [28] and constructed liftOver chains [29]. Then, we obtained the 500 bp upstream and downstream flanks of each gap in the short-read assembly and used liftOver [30] (parameter -minMatch=0.8) to map these flanks to the reference assembly. Before introducing assembly gaps into the reference, we disassembled the reference assembly into its contigs to avoid having gaps without a known ground truth sequence. For a pair of flanks that belong to the same gap and were mapped adjacently to a reference contig, we replaced the real sequence (ground truth) between both flanks with N's with "dentist build-partial-assembly" if the gap size is ≥10 bp. We required that the reference assembly itself (which is a real assembly with gaps on its own) not already have a gap within 3 kb of an introduced gap. It should be noted that the introduced gap size in our tests is identical to the size of the ground truth gap sequence. This gives gap closers that take gap size into account an advantage (e.g., PBJelly, LR_gapcloser, and PacBio GC) but not DENTIST, which does not use the estimated size of gaps because in reality they are often imprecise or even have a pseudo-size not related to the real gap size at all.

Because the true sequence in gaps is known, it is possible to automatically assess the performance of the gap-closing tools. To this end, we first identify the original contigs of the test assembly in a gap-closed assembly by searching for exact and unique matches. Duplicated contigs that already have >1 exact match within the test assembly are excluded. We noticed that some gap-closing tools do not only fill in gaps but also modify the input contigs. We handled this unexpected behaviour in 2 ways. First, before searching for exact matches, we remove a given number of base pairs from the flanks of the contig of the test assembly, thus allowing the gap closer to modify contig flanks. This number of base pairs is controlled by –crop-ambiguous (default 100 bp) when searching for duplicate contigs and by –crop-alignment when searching for contigs in the gap-closed assembly. Second, after identifying exact matches, we optionally conduct a second search for the remaining contigs, allowing for up to 1.5% mismatches but requiring that a contig fully and uniquely align to some part of the result assembly. Consequently, original contigs that are substantially modified by a gap-closing method may not be detected, which highlights an unwanted behaviour of the respective method.

After determining the locations of the test contigs, each gap is evaluated according to the following rules. A gap is considered "closed" if the locations of both contigs surrounding the gap in the test assembly are mapped to a single contig in the gap-closed assembly. In this case, the sequence identity between the known gap sequence and the inserted gap sequence is calculated. A gap is considered "unclosed" if the locations of both contigs surrounding the gap in the test assembly are known and the contigs are mapped to different but adjacent contigs in the gap-closed assembly; i.e., a gap remains. A gap is considered "unknown" if none of the flanking contigs are found or ≥1 flanking contigs cannot be located uniquely. These gaps are excluded from further analysis because we cannot accurately determine the inserted gap sequence. If neither of the above cases is true, the gap is said to be "broken." This is the case if both flanking contigs could be located but are not adjacent anymore, i.e., the contigs are misassembled. This procedure makes it posssible to obtain an accurate assessment of the performance of each gap-closing tool in an automated fashion.

Long reads used in our tests were either real PacBio reads (Supplementary Table S1) or reads that were sampled (simulated) from

the reference assembly adding a typical PacBio base and indel error profile using "simulator -m25000 -s125000 -e.13 -rSEED" [19]. These parameters provide a length and error rate distribution that match the distributions of current CLR reads. The seed used for simulating reads is listed in Supplementary Table S1. For real PacBio reads, we provided all subreads from all wells to the gap-closing tools.

Running gap-closing methods

All tools were called with the default or recommended parameters, except for parameters that do not influence the output such as the number of parallel threads. Supplementary Table S3 lists the exact parameters used for each tool. For finisherSC.py, we used parameters "–fast True –large True" to make the *Drosophila* application feasible in terms of runtime and memory requirements. For PacBio GC, we ran variantCaller with parameter "–algorithm=best", which automatically selects Arrow or Quiver as the most appropriate method. To prevent PacBio GC from modifying bases in the contigs outside of gaps, which hampers the exact identification of the inserted sequences, we restricted its application to the exact gap locations using the "–referenceWindowsFile" parameter. Furthermore, we distributed the computation on our compute cluster by (i) computing the read alignments for the whole assembly, (ii) splitting the assembly into blocks of ∼200 Mb and dividing the read alignments accordingly, (iii) applying PacBio GC to each block separately, and (iv) merging the unmodified contigs with the processed gaps into the output assembly. This complete workflow can be found at [31, 32].

The results were evaluated using "dentist check-results" with parameter "–crop-ambiguous=300" to ignore 300 bp from the contig flanks when searching for duplicate contigs. To evaluate PBJelly, parameter "–crop-alignment=100" was used in addition to ignore 100 bp from the contig flanks in the contig identification step. To evaluate LR_gapcloser, which may modify the original contig flanks, we additionally specified parameters "–crop-alignment=300" and "–recover-imperfect-contigs" to enable contig identification with ≤1.5% mismatches. The increase in contiguity of the gap-closed assembly was measured by the percentage of closed gaps and the increase in contig NG50. The correctness of closed gap sequences was measured as the sequence identity between the (known) ground truth and the inserted sequence for each gap. These values are presented in 3 forms: as a distribution binned in 6 intervals: [0, 0.7), [0.7, 0.9), [0.9, 0.95), [0.95, 0.99), [0.99, 1.0), and {1.0}; as the arithmetic mean over all the sequence identities; and as the weighted arithmetic mean using the true gap sizes as weights.

For tests with simulated reads, we also used "dentist find-closable-gaps" to determine which gaps are closable because the true origin of each read is known. We defined a gap as "closable" if and only if ≥3 reads span the gap and 500 bp on either side.

Data Availability

All data underlying this article, including the reference and test assemblies with introduced gaps and their true sequence as valuable data for future method comparisons, are available via our institutional server [31]. Supporting data and an archival copy of the code are also available via the *GigaScience* repository GigaDB [33].

Availability of Supporting Source Code and Requirements

Project name: DENTIST
Project home page: https://github.com/a-ludi/dentist
Operating system: Linux
Programming language: D
Other requirements: Snakemake 5.32.1 or higher
License: MIT
RRID:SCR_021856
biotools: dentist

Additional Files

Supplementary Figure S1. Visualization of a scaffold graph for two exemplary contigs.
Supplementary Table S1. List of data sets and corresponding sources.
Supplementary Table S2. Examples of DENTIST's default masking threshold C_{max} in dependence of the read coverage C.
Supplementary Table S3. Parameters used for the gap closers.
Supplementary Listing S1. List of command line options for DENTIST.
Supplementary Note S1. Guide on configuration of DENTIST.

Abbreviations

bp: base pairs; CLR: continuous long read; CPU: central processing unit; Gb: gigabase pairs; kb: kilobase pairs; Mb: megabase pairs; PacBio: Pacific Biosciences; UCSC: University of California Santa Cruz.

Competing Interests

The authors state that they have no competing interests.

Funding

This work was supported by the Max Planck Society, the Federal Ministry of Education and Research (grant 01IS18026C), and the LOEWE-Centre for Translational Biodiversity Genomics (TBG) funded by the Hessen State Ministry of Higher Education, Research and the Arts (HMWK).

Authors' Contributions

A.L. implemented the software and analysed the data. M.P. tested the software. M.H. and G.M. conceived and supervised this study. M.H. and A.L. wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgment

We thank the Computer Service Facilities of the MPI-CBG for their support.

References

1. Rhie, A, McCarthy, SA, Fedrigo, O, *et al.* Towards complete and error-free genome assemblies of all vertebrate species. *Nature* 2021;**592**(7856):737–46.

2. Jebb, D, Huang, Z, Pippel, M, *et al.* Six reference-quality genomes reveal evolution of bat adaptations. *Nature* 2020;**583**(7817):578–84.
3. Warren, WC, Harris, RA, Haukness, M, *et al.* Sequence diversity analyses of an improved rhesus macaque genome enhance its biomedical utility. *Science* 2020;**370**(6523):doi:10.1126/science.abc6617.
4. Rice, ES, Green, RE. New approaches for genome assembly and scaffolding. *Annu Rev Anim Biosci* 2019;**7**(1):17–40.
5. Zoonomia Consortium. A comparative genomics multitool for scientific discovery and conservation. *Nature* 2020;**587**(7833):240–5.
6. Feng, S, Stiller, J, Deng, Y, *et al.* Dense sampling of bird diversity increases power of comparative genomics. *Nature* 2020;**587**(7833):252–7.
7. Miga, KH, Koren, S, Rhie, A, *et al.* Telomere-to-telomere assembly of a complete human X chromosome. *Nature* 2020;**585**(7823):79–84.
8. 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**(11):e47768.
9. Lam, KK, LaButti, K, Khalak, A, *et al.* FinisherSC: a repeat-aware tool for upgrading de novo assembly using long reads. *Bioinformatics* 2015;**31**(19):3207–9.
10. Pacific Biosciences. GenomicConsensus: PacBio® variant and consensus caller. https://github.com/PacificBiosciences/GenomicConsensus. Accessed 12 December 2021.
11. Xu, GC, Xu, TJ, Zhu, R, *et al.* LR_Gapcloser: a tiling path-based gap closer that uses long reads to complete genome assembly. *Gigascience* 2019;**8**(1):doi:10.1093/gigascience/giy157.
12. Xu, M, Guo, L, Gu, S, *et al.* TGS-GapCloser: A fast and accurate gap closer for large genomes with low coverage of error-prone long reads. *Gigascience* 2020;**9**(9):doi:10.1093/gigascience/giaa094.
13. Lee, CM, Barber, GP, Casper, J, *et al.* UCSC Genome Browser enters 20th year. *Nucleic Acids Res* 2020;**48**(D1):D756–D61.
14. Wenger, AM, Peluso, P, Rowell, WJ, *et al.* Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat Biotechnol* 2019;**37**(10):1155–62.
15. Koster, J, Rahmann, S. Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics* 2012;**28**(19):2520–2.
16. Kurtzer, GM, Sochat, V, Bauer, MW. Singularity: Scientific containers for mobility of compute. *PLoS One* 2017;**12**(5):e0177459.
17. Ludwig, A, Pippel, M, Myers, G *et al.* DENTIST: Close assembly gaps using long-reads at high accuracy. https://github.com/a-ludi/dentist. Accessed 11 January 2022.
18. Myers, G. DAZZ_DB: The Dazzler Data Base. https://github.com/thegenemyers/DAZZ_DB. Accessed 04 December 2020.
19. Myers, G. DAZZ_DB Commands https://dazzlerblog.wordpress.com/command-guides/dazz_db-command-guide. Accessed 22 October 2022.
20. Myers, G. Efficient local alignment discovery amongst noisy long reads. In: D Brown, B Morgenstern, eds. *Algorithms in Bioinformatics. WABI 2014*. Berlin, Heidelberg: Springer; 2014:52–67.
21. Myers, G. DAMASKER: Module to determine where repeats are and make soft-masks of said. https://github.com/thegenemyers/DAMASKER. Accessed 13 October 2020.
22. Myers, G. DAMASKER commands. https://dazzlerblog.wordpress.com/command-guides/damasker-commands. Accessed 11 January 2022.
23. Myers, G. DAMAPPER: Long read to reference genome mapping tool. https://github.com/thegenemyers/DAMAPPER. Accessed 07 December 2020.
24. Myers, G . Mapping Your Reads: damapper. https://dazzlerblog.wordpress.com/2016/07/31/damapper-mapping-your-reads. Accessed 11 January 2022 .
25. Myers, G. DASCRUBBER: Alignment-based Scrubbing pipeline. https://github.com/thegenemyers/DASCRUBBER. Accessed 30 September 2020.
26. Myers, G. DASCRUBBER Commands. https://dazzlerblog.wordpress.com/command-guides/dascrubber-command-guide. Accessed 11 January 2022 .
27. Tischler, G, Myers, EW. Non hybrid long read consensus using local de Bruijn graph assembly. *bioRxiv* 2017:doi:10.1101/106252.
28. Harris, RS. *Improved pairwise alignment of genomic DNA. PhD Thesis*. The Pennsylvania State University, 2007.
29. Kent, WJ, Baertsch, R, Hinrichs, A, *et al.* Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proc Natl Acad Sci U S A* 2003;**100**(20):11484–9.
30. Hinrichs, AS, Karolchik, D, Baertsch, R, *et al.* The UCSC Genome Browser Database: update 2006. *Nucleic Acids Res* 2006;**34**(Database issue):D590–8.
31. Ludwig, A, Pippel, M, Myers, G, *et al.* DENTIST: Test Data. https://bds.mpi-cbg.de/hillerlab/DENTIST/. Accessed 11 January 2022.
32. Ludwig, A, Pippel, M, Myers, G *et al.* Snakemake workflow used for PacBio® GenomicConsensus. https://bds.mpi-cbg.de/hillerlab/DENTIST/source/arrow/Snakefile. Accessed 11 January 2022.
33. Ludwig, A, Pippel, M, Myers, G, *et al.* Supporting data for "DENTIST—using long reads for closing assembly gaps at high accuracy." *GigaScience Database* 2022. http://dx.doi.org/10.5524/100968.