Briefings in Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

A comprehensive evaluation of long read error correction methods

Haowen Zhang[1], **Chirag Jain**[1] **and Srinivas Aluru**[1,2,*]

[1]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA and
[2]Institute for Data Engineering and Science, Georgia Institute of Technology, Atlanta, GA 30332, USA.

[*]To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Third-generation sequencing technologies can sequence long reads, which is advancing the frontiers of genomics research. However, their high error rates prohibit accurate and efficient downstream analysis. This difficulty has motivated the development of many long read error correction tools, which tackle this problem through sampling redundancy and/or leveraging accurate short reads of the same biological samples. Existing studies to asses these tools use simulated data sets, and are not sufficiently comprehensive in the range of software covered or diversity of evaluation measures used.
Results: In this paper, we present a categorization and review of long read error correction methods, and provide a comprehensive evaluation of the corresponding long read error correction tools. Leveraging recent real sequencing data, we establish benchmark data sets and set up evaluation criteria for a comparative assessment which includes quality of error correction as well as run-time and memory usage. We study how trimming and long read sequencing depth affect error correction in terms of length distribution and genome coverage post-correction, and the impact of error correction performance on an important application of long reads, genome assembly. We provide guidelines for practitioners for choosing among the available error correction tools and identify directions for future research.
Availability: The source code is available at https://github.com/haowenz/LRECE.
Contact: aluru@cc.gatech.edu
Key words: long read; error correction; benchmark; evaluation

1 Introduction

Third-generation sequencing technologies produce long reads with average length of 10 Kbp or more that are orders of magnitudes longer than the short reads available through second-generation sequencing technologies (typically a few hundred bp). In fact, the longest read length reported to date is $>$ 1 million bp (Sedlazeck *et al.*, 2018). Longer lengths are attractive because they enable disambiguation of repetitive regions in a genome or a set of genomes. The impact of this valuable long-range information has already been demonstrated for *de novo* genome assembly (Loman *et al.*, 2015; Chin *et al.*, 2016; Jain *et al.*, 2018), novel variant detection (Sedlazeck *et al.*, 2017; Chaisson *et al.*, 2015), RNA-seq analysis (Gordon *et al.*, 2015), and epigenetics (Rand *et al.*, 2017; Simpson *et al.*, 2017).

The benefit of longer read lengths, however, comes with the major challenge of handling high error rates. Currently, there are two widely used third-generation sequencing platforms – Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT). Both sequencing platforms are similar in terms of their high error rates (ranging from 10-20%) with most errors occurring due to insertions or deletions (indels); however the error distribution varies (Carneiro *et al.*, 2012; Jain *et al.*, 2015, 2018). Pacbio sequencing errors appear to be randomly distributed over the sequence (Korlach and Biosciences, 2013). For ONT on the other hand, the error profile has been reported to be biased. For example, A to T and T to A substitutions are less frequent than other substitutions, and indels tend to occur in homopolymer regions (Jain *et al.*, 2015; Ashton *et al.*, 2015). These error characteristics pose a challenge for long read data analyses, particularly for detecting correct read overlaps during genome assembly and variants at single base pair resolution, thus motivating the development of error correction methods.

Error correction algorithms are designed to identify and fix or remove sequencing errors, thereby benefiting resequencing or *de novo* sequencing analysis. In addition, the algorithms should be computationally efficient to handle increasing volumes of sequencing data, particularly in the case of large, complex genomes. Numerous error correction methodologies and software have been developed for short reads; we refer readers to Yang *et al.* (2012) and Alic *et al.* (2016) for a thorough review. Given the distinct characteristics of long reads, i.e., significantly higher error rates and lengths, specialized algorithms are needed to correct them. Till date, several error correction tools for long reads have been developed including PacBioToCA (Koren *et al.*, 2012), LSC (Au *et al.*, 2012), ECTools (Lee *et al.*, 2014), LoRDEC (Salmela and Rivals, 2014), proovread (Hackl *et al.*, 2014), NaS (Madoui *et al.*, 2015), Nanocorr (Goodwin *et al.*, 2015), Jabba (Miclotte *et al.*, 2016), CoLoRMap (Haghshenas *et al.*, 2016), LoRMA (Salmela *et al.*, 2016), HALC (Bao and Lan, 2017), FLAS (Bao *et al.*, 2017), FMLRC (Wang *et al.*, 2018), HG-CoLoR (Morisse *et al.*, 2018) and Hercules (Firtina *et al.*, 2018).

In addition, error correction modules have been developed as part of long read *de novo* assembly pipelines, such as Canu (Koren *et al.*, 2017) and HGAP (Chin *et al.*, 2013). In the assembly pipeline, correction helps by increasing alignment identities of overlapping reads, which facilitates overlap detection and improves assembly. Many long error correction tools require and make use of highly accurate short reads to correct long reads (accordingly referred to as hybrid methods). Others, referred to as non-hybrid methods, perform self-correction of long reads using overlap information among them.

A few review studies have showcased comparisons among rapidly evolving error correction algorithms to assess state-of-the-art. Laehnemann *et al.* (2015) provide an introduction to error rates/profiles and a methodology overview of some correction tools for various short and long read sequencing platforms, although no benchmark is included. A review and benchmark for hybrid methods is also available (Mahmoud *et al.*, 2017). However, the study only used simulated reads and focused more on speed rather than correction accuracy. Besides, it does not include non-hybrid methods in the assessment. More recently, LRCstats (La *et al.*, 2017) was developed for evaluation of long read error correction software; however, it is restricted to benchmarking with simulated reads. Furthermore, it does not provide a comprehensive evaluation of many of the current state-of-the-art correction software.

While benchmarking with simulated reads is useful, it fails to convey performance in real-world scenarios. Besides the base-level errors (i.e., indels and substitutions), real sequencing data sets also contain larger structural errors, e.g., chimeras (Fichot and Norman, 2013). However, state-of-the-art simulators such as SimLoRD (Stöcker *et al.*, 2016) only generate reads with base-level errors rather than structural errors. Furthermore, Miclotte *et al.* (2016) consistently observed worse performance when using real reads instead of simulated reads, suggesting that simulation may fail to match the characteristics of actual error distribution. Therefore, benchmarking with real datasets is important.

In this study, we establish benchmark datasets, present an evaluation methodology suitable to long reads, and carry out comprehensive evaluation of the quality and computational resource requirements of state-of-the-art long read correction software. We also study the effect of trimming and different sequencing depths on correction quality. To understand impact of error correction on downstream analysis, we perform assembly using corrected reads generated by various tools and assess quality of the resulting assemblies.

2 Overview of long read error correction methods

2.1 Hybrid methods

Hybrid methods take advantage of high accuracy of short reads (error rates often $< 1\%$) for correcting errors in long reads. An obvious requirement is that the same biological sample be sequenced using both short read and long read technologies. Based on how these methods make use of short reads, we further divide them into two categories: *alignment-based* and *assembly-based*. The first category includes Hercules, CoLoRMap, Nanocorr, Nas, proovread, LSC and PacBioToCA, whereas HG-CoLoR, HALC, Jabba, LoRDEC, and ECTools are in the latter. The ideas underlying the methods are summarized below.

2.1.1 Short-read-alignment-based methods

As a first step, these methods align short reads to long reads using a variety of aligners, e.g. BLAST (Altschul *et al.*, 1990), Novoalign (`http://www.novocraft.com/products/novoalign/`). As long reads are usually error-prone, some alignments can be missed or biased. Thus, most of the tools in this category utilize various approaches to increase accuracy of alignments. Drawing upon the alignments, these methods use distinct approaches to generate corrected reads.

PacBioToCA: Consensus sequences for long reads are generated by multiple sequence alignment of short reads using AMOS consensus module (Pop *et al.*, 2004).

LSC: Short reads and long reads are compressed using homopolymer compression (HC) transformation prior to alignment. Then error correction is performed at HC points, mismatches and indels by temporarily decompressing the aligned short reads and then generating consensus sequences. Finally, the corrected sequences are decompressed.

proovread: Similar to PacBioToCA and LSC, short reads are mapped to long reads and then the resulting alignments are used to call consensus. But its alignment parameters are carefully selected and adapted to the PacBio sequencing error profile. To further improve correction, the phred quality score and Shannon entropy value are calculated at each nucleotide for quality control and chimera detection, respectively. To reduce run time, an iterative correction strategy is employed. Three pre-correction steps are performed using increasing subsamples of short reads. In each step, the long read regions are masked to reduce alignment search space once they are corrected and covered by a sufficient number of short read alignments. In the final step, all short reads are mapped to the unmasked regions to make corrections.

NaS: Like the other tools in this category, it first aligns short reads with long reads. However, only the stringently aligned short reads are found and kept as seed-reads. Then instead of calling consensus, similar short reads are retrieved with these seed-reads. Micro-assemblies of these short reads are performed to generate contigs, which are regarded as corrected reads. In other words, the long reads are only used as template to select seed-reads.

Nanocorr: It follows the same general approach as PacBioToCA and LSC, by aligning short reads to long reads and then calling consensus. But before the consensus step, a dynamic programming algorithm is utilized to select an optimal set of short read alignments that span each long read.

CoLoRMap: CoLoRMap does not directly call consensus. Instead, for each long read region, it runs a shortest path algorithm to construct a sequence of overlapping short reads aligned to that region with minimum edit distance. Subsequently, the region is corrected by the constructed sequence. In addition, for each uncovered region (called gap) on long reads, any unmapped reads with corresponding mapped mates are retrieved and assembled locally to fill the gap.

***Hercules**:* It first aligns short reads to long reads. Then unlike other tools, Hercules uses a machine learning-based algorithm. It creates a profile Hidden Markov Model (pHMM) template for each long read and then learns posterior transition and emission probabilities. Finally, the pHMM is decoded to get the corrected reads.

2.1.2 Short-read-assembly-based methods

These methods first perform assembly with short reads, e.g., generate contigs using an existent assembler, or only build the de Bruijn graph (DBG) based on them. Then the long reads are aligned to the assemblies, i.e., contigs/unitigs or a path in the DBG, and corrected. Algorithms for different tools in this category are summarized below.

***ECTools**:* First, unitigs are generated from short reads using any available assembler and aligned to long reads. Afterwards, the alignments are filtered to select a set of unitigs which provide the best cover for each long read. Finally, differences in bases between each long read and its corresponding unitigs are identified and corrected.

***LoRDEC**:* Unlike ECTools which generates assemblies, LoRDEC only builds a DBG of short reads. Subsequently, it traverses paths in the DBG to correct erroneous regions within each long read. The regions are replaced by the respective optimal paths which are regarded as the corrected sequence.

***Jabba**:* It adopts a similar strategy as in LoRDEC, and builds a DBG of short reads followed by aligning long reads to the graph to correct them. The improvement is that Jabba employs a seed-and-extend strategy using maximal exact matches (MEMs) as seeds to accelerate the alignment.

***HALC**:* Similar to ECTools, short reads are used to generate contigs as the first step. Unlike other methods which try to avoid ambiguous alignments (Koren *et al.*, 2012; Yang *et al.*, 2010), HALC aligns long reads to the contigs with a relatively low identity requirement, thus allowing long reads to align with their similar repeats which might not be their true genomic origin. Then long reads and contigs are split according to the alignments so that every aligned region on read has its corresponding aligned contig region. A contig graph is constructed with the aligned contig regions as vertices. A weighted edge is added between two vertices if there are adjacent aligned long read regions supporting it. The more regions support the edge, the lower is the weight assigned to it. Each long read is corrected by the path with minimum total weight in the graph. Furthermore, the corrected long read regions are refined by running LoRDEC, if they are aligned to similar repeats.

***FMLRC**:* This software uses a DBG-based correction strategy similar to LoRDEC. However, the key difference in the algorithm is that it makes two passes of correction using DBGs with different k-mer sizes. The first pass does the majority of correction, while the second pass with a longer k-mer size corrects repetitive regions in the long reads. Note that a straightforward implementation of a DBG does not support dynamic adjustment of k-mer size. As a result, FMLRC uses FM-index to implicitly represent DBGs of arbitrary length k-mers.

***HG-CoLoR**:* Similar to FMLRC, it avoids using a fixed k-mer size for the de Bruijn graph. Accordingly, it relies on a variable-order de Bruijn graph structure (Kowalski *et al.*, 2015). It also uses a seed-and-extend approach to align long reads to the graph. However, the seeds are found by aligning short reads to long reads rather than directly selecting them from the long reads.

2.2 Non-hybrid methods

These methods perform self-correction with long reads alone. They all contain a step to generate consensus sequences using pairwise alignment/overlap information. However, the respective methods vary in how they find the overlaps and generate consensus sequences. The details are as follows.

***FLAS**:* It takes all-to-all long read overlaps computed using MECAT (Xiao *et al.*, 2017) as input, and clusters the reads that are aligned with each other. In case of ambiguous instances, i.e., the clusters that share the same reads, FLAS evaluates the overlaps by computing alignments using sensitive alignment parameters either to augment the clusters or discard the incorrect overlaps. The refined alignments are then used to correct the reads. To achieve better accuracy, it also corrects errors in the uncorrected regions of the long reads. Accordingly, it constructs a string graph using the corrected regions of long reads, and aligns the uncorrected ones to the graph for further correction.

***LoRMA**:* By gradually increasing the k-mer size, LoRMA iteratively constructs DBGs using k-mers from long reads exceeding a specified frequency threshold, and runs LoRDEC to correct errors based on the respective DBGs. After that, a set of reads similar to each read termed *friends* are selected using the final DBG, which should be more accurate due to several rounds of corrections. Then, each read is corrected by the consensus sequence generated by its friends.

***Canu error correction module**:* As a first step during the correction process, Canu computes all-versus-all overlap information among the reads using a modified version of MHAP (Berlin *et al.*, 2015). It uses a filtering mechanism during the correction to favor true overlaps over the false ones that occur due to repetitive segments in genomes. The filtering heuristic ensures that each read contributes to correction of no more than D other reads, where D is the expected sequencing depth. Finally, a consensus sequence is generated for each read using its best set of overlaps.

3 Materials and Methods

We selected data sets from recent publicly accessible genome sequencing experiments. For benchmarking the different programs, our experiments used genome sequences from multiple species and different sequencing platforms with recent chemistry, e.g., R9/R7 for ONT or P6-C4/P5-C3 for PacBio. We describe our evaluation criteria and use it for a comprehensive assessment of the correction methods/software.

3.1 Benchmark data sets

Our benchmark includes resequencing data from three reference genomes – *Escherichia coli* K-12 MG1655 (*E. coli*), *Saccharomyces cerevisiae* S288C (yeast), and *Drosophila melanogaster* ISO1 (fruit fly). The biggest hurdle when benchmarking with real data is the absence of ground truth (i.e., perfectly corrected reads). However, the availability of reference genomes of these strains enables us to evaluate the output of correction software in a reliable manner using the reference. Essentially, differences in a corrected read with respect to the reference imply uncorrected errors. A summary of the selected read data sets is listed in Table 1. We leveraged publicly available high coverage read data sets of the selected genomes available from all three platforms – Illumina (for short reads), Pacbio, and ONT. In addition, some of these samples were sequenced using multiple protocols, yielding reads of varying quality. This enabled us to do a thorough comparison among error correction software across various error rates and error profiles.

Table 1. Details of the benchmark data sets

Data set	Sequencing specification	Sequencing NCBI accession	Sequencing depth[a]	Read length (bp)[b]	Number of reads	Reference genome	Genome length (bp)	Reference NCBI accession
D1-I	Illumina Miseq	-[c]	373x	2×151	2×5 729 470	E. coli K-12 MG1655	4 641 652	NC_000913.3
D1-P	Pacbio P6C4	-[d]	161x	13 982	87 217			
D1-O1	MinION R7.3 1D	PRJEB7385[e]	53x	8631	44 540			
D1-O2	MinION R7.3 2D	PRJEB7385[e]	29x	9356	22 270			
D2-I	Illumina Miseq	ERR1938683	81x	2×150	2×3 318 467	S. cerevisiae S288c	12 157 105	PRJNA128
D2-P	Pacbio P6C4	PRJEB7245	120x	8656	239 408			
D2-O1	MinION R9&R7.3 pass 2D	-[f]	31x	11 693	42 325			
D2-O2	MinION R9&R7.3 all 2D	-[f]	61x	11 075	90 791			
D3-I	Illumina Nextseq	SRX3676782	44x	2×151	2×20 619 401	D. melanogaster ISO1	143 726 002	PRJNA164
D3-P	Pacbio P5C3	SRX499318	204x	15 132	6 864 972			
D3-O	MinION R9.5 1D	SRX3676783	32x	11 934	663 784			

[a] Sequencing depth is estimated using the sequencing data and reference genome size.
[b] N50 is reported for PacBio or ONT reads, since their lengths vary.
[c] Downloaded from Illumina at ftp://webdata:webdata@ussd-ftp.illumina.com/Data/SequencingRuns/MG1655/MiSeq_Ecoli_MG1655_110721_PF_R1.fastq.gz and ftp://webdata:webdata@ussd-ftp.illumina.com/Data/SequencingRuns/MG1655/MiSeq_Ecoli_MG1655_110721_PF_R2.fastq.gz.
[d] Downloaded from https://github.com/PacificBiosciences/DevNet/wiki/E.-coli-Bacterial-Assembly.
[e] For PRJEB7385, only ERX708228, ERX708229, ERX708230 and ERX708231 are included.
[f] Pass and all 2D sequencing data are available from EBI Bio-Studies with accession number S-BSST17.

Dataset D1-O1 is a recent MinION sequencing of *E. coli* genome (Loman *et al.*, 2015). Its 2D reads were also extracted from the raw reads using poretools (Loman and Quinlan, 2014), and was included into the benchmark as D1-O2. Note that raw reads are more erroneous than the 2D reads, which enabled the evaluation of the tools across different error rates. Giordano *et al.* (2017) recently released a bundle of PacBio, MinION, and Miseq sequencing data of the yeast genome. For the same purpose, *pass* 2D reads and the combination of *pass* and *fail* 2D reads of the MinION data were downloaded and regarded as two separate data sets in our benchmark (D2-O1 and D2-O2).

To conduct performance evaluation under different sequencing depths, yeast sequencing reads (D2-P and D2-O1) were subsampled randomly using Seqtk (`https://github.com/lh3/seqtk`). Subsamples with average depth of 10x and 20x were generated for MinION reads. In addition, 10x, 20x, 30x, 60x and 90x PacBio read subsamples were generated from D2-P. Details of these subsamples are available in Supplementary Table 1.

3.2 Evaluation methodology

Our evaluation method takes uncorrected reads, corrected reads, and a reference genome as input. Both the uncorrected and corrected reads were filtered using a user defined length (default 500). Reads which were too short to include in downstream analysis were dropped during the filtration. Filtered reads were aligned to the reference genome using Minimap2 (Li, 2018). Majority of reads align to a single position in the reference. Fraction of base pairs with ambiguous or split read mappings is found to be insignificant (Supplementary Table 2). This can be attributed to two reasons. First, the reads were sequenced from the same reference genome to which they are aligned. Second, as the reads are long (>500 bp), majority of the base pairs map uniquely to the reference. As a result, we retain only the primary alignment for a read with multiple mappings or split alignments.

In an ideal scenario, an error correction software should take each erroneous long read and produce the error-free version of it, preserving each read and its full length. To assess how close to the ideal one can get, measures such as error rate post-correction or percentage of errors removed (termed *gain*; see Yang *et al.* (2012)) can be utilized. However, long read error correction programs do not operate in this fashion. They may completely discard some reads or choose to split an input read into multiple reads when the high error rate cannot be reckoned with. In addition, short read assembly based error correction programs use long read alignments to de Bruijn graphs, and produce sequences corresponding to the aligned de Bruijn graph paths as output reads instead. Though original reads may not be fully preserved, all that matters for effective use of error correction software is that its output consists of sufficient number of high quality long reads that reflect adequate read lengths, sequencing depth, and coverage of the genome. Accordingly, our evaluation methodology reflects such assessment.

We measure the number of reads and total bases output by each error correction software, along with the number of aligned reads and total number of aligned bases extracted from alignment statistics, because they together reveal the effectiveness of correction. Besides, statistics which convey read length distribution such as maximum length and N50 were calculated to assess effect of the correction process on read lengths. Fraction of the genome covered by output reads is also reported to assess if there are regions of the genome that lost coverage or suffered significant deterioration in coverage depth post-correction. Any significant drop on these metrics can be a potential sign of information loss during the correction. Finally, alignment identity is calculated by the number of matched bases divided by the alignment length, averaged over all reads. Tools which achieve maximum alignment identity with minimum loss of information are desirable.

As part of this study, we provide an evaluation tool to automatically generate the evaluation statistics mentioned above. Besides, we provide a wrapper script which can run state-of-the-art error correction software on a grid engine given any input data from user. Using the scripts, two types of evaluations can be conducted. Users can either evaluate the performance on a list of tools with their own data to find a suitable tool for their studies, or they can run any correction tool with the benchmark data and compare it with other state-of-the-art tools.

4 Experimental results and discussion

4.1 Experimental setup

All tests were run on the Swarm cluster located at Georgia Institute of Technology. Each compute node in the cluster has dual Intel Xeon CPU E5-2680 v4 (2.40GHz) processors equipped with a total of 28 cores and 256GB main memory. The cluster is set up using 64-bit Red Hat Linux kernel version 2.6.32.

4.2 Evaluated software

We evaluated 15 long read error correction programs in this study: Hercules, HG-CoLoR, FMLRC, HALC, CoLoRMap, Jabba, Nanocorr, proovread, LoRDEC, ECTools, LSC, PacBioToCA, FLAS, LoRMA and the error correction module in Canu. NaS was not included in the evaluation because it requires Newbler assembler which is no longer available from 454. The command line parameters were chosen based on user documentations of each software (Supplementary note section “Versions and configurations”). The tools were configured to run exclusively on a single compute node and allowed to leverage all the 28 cores if multi-threading is supported. A cutoff on wall time was set to three days.

4.3 Performance on benchmark data sets

We evaluated the quality and computational resource requirements of each software on our benchmark data sets (Table 1). Results for the three different datasets are shown in Tables 2, 3 and 4, respectively. Because multiple factors are at play when considering accuracy, it is important to consider their collective influence in assessing quality of error correction. In what follows, we present a detailed and comparative discussion on correction accuracy, runtime and memory-usage. In addition, to guide error correction software users and future developers, we provide further insights into the strengths and limitations of various approaches that underpin the software. This includes evaluating their resilience to handle various sequencing depths, studying the effect of discarding or trimming input reads to gain higher accuracy, and impact on genome assembly.

4.3.1 Correction quality

We measure quality using the number of reads and total bases output in comparison with the input, the resulting alignment identity, fraction of the genome covered and read length distribution including maximum size and N50 length. From Tables 2, 3 and 4, we gather that the best performing hybrid methods (e.g., FMLRC) are capable of correcting reads to achieve base-level accuracy in the high 90’s. For the *E. coli* and yeast data sets, many of these programs achieve alignment identity $> 99\%$. A crucial aspect to consider here is whether the high accuracy is achieved while preserving input read depth and N50. Few tools (e.g. Jabba, proovread) seem to attain high alignment identity at the cost of producing shorter reads and reduced depths because they choose to either discard uncorrected reads or trim the uncorrected regions. This may have a negative impact on downstream analyses. This trade-off is further discussed later in Section 4.3.4.

Among the hybrid methods, a key observation is that short-read-assembly-based methods tend to show better performance than short-read-alignment-based methods. We provide the following explanation. Given that long reads are error-prone, short read alignment to long reads is more likely to be wrong (or ambiguous) than long read alignment to graph structures built using short reads. Errors in long reads can cause false positives in identifying the true positions where the respective short reads should align, which causes false correction later. For example, during the correction of D3-P, the alignment identity of corrected reads generated by CoLoRMap in fact decreased when compared to the uncorrected reads. The reason is that CoLoRMap uses BWA-mem (Li, 2013) to map short reads, which is designed to report best mapping. However, due to the high error rates, the best mapping is not necessarily the true mapping. Large volume of erroneous long reads in D3-P can lead to many false alignments, which affected the correction process. On the other hand, long read lengths make it possible to have higher confidence when aligning them to paths in the graph. Therefore, in most of the experiments, assembly-based methods were able to produce reasonable correction.

Non-hybrid correction is more challenging as it relies solely on overlaps between erroneous long reads, yet the tools in this category yield competitive accuracy in many cases. However, non-hybrid methods may significantly reduce read count and/or read lengths, and completely fail when the original long reads are highly erroneous. For example, neither Canu nor LoRMA was able to correct D1-O1 where average input identity is only 63.46%. FLAS also discarded most of the reads.

4.3.2 Runtime and memory usage

Scalability of the correction tools is an important aspect to consider in their evaluation. Slow speed or high memory usage makes it difficult to apply them to correct large data sets. Our results show that hybrid methods, in particular assembly-based methods, are much faster than the rest. For instance, PacBioToCA and LSC failed to generate corrected reads in three days for D1-P, while most of the assembly-based tools finished the same job in less than one hour. Nanocorr, ECTools and LSC were unable to finish the correction of D2-O2 in three days, which was finished by FMLRC or LoRDEC in 30 minutes. Although proovread can complete the corrections of D2-P, D2-O1 and D2-O2, the run-time was 49.3, 17.5 and 29.3 times longer, respectively, than run-time needed by FMLRC. Moreover, assembly-based methods, e.g., LoRDEC and FMLRC, used less memory in most of the experiments. Therefore, in terms of computational performance, users should give priority to assembly-based methods over short-read alignment-based methods.

Among the non-hybrid methods, LoRMA’s memory usage was generally the highest among all the tools, and was slower than assembly-based methods. However, Canu showed superior scalability. Owing to a fast long read overlap detection algorithm using MinHash (Berlin *et al.*, 2015), Canu was able to compute long read overlaps and used them to correct the reads in reasonable time, which is comparable to most of the hybrid methods. The memory footprint of Canu was also lower than many hybrid-methods. However, Canu did not finish the correction of D3-P in three days probably because this data set is too large to compute pairwise overlaps. FLAS showed performance comparable to Canu as FLAS also leverages the fast overlap computation method in MECAT (Xiao *et al.*, 2017).

4.3.3 Effect of long read sequencing depth on error correction

Requiring high sequencing coverage for effective error correction can impact both cost and time consumed during sequencing and analysis. The relative cost per base pair using third-generation sequencing is still several folds higher when compared to the latest Illumina sequencers (Sedlazeck *et al.*, 2018). Accordingly, we study how varying long read sequencing depth affects correction quality, while keeping the short read data set fixed. We conducted this experiment using data sets D2-P and D2-O1 with various depth levels obtained using random sub-sampling. The output behavior of the correction tools is shown in Supplementary Tables 3-7.

For corrected reads generated by hybrid methods, no significant change on the metrics was observed except those generated by CoLoRMap. The alignment identity of its corrected reads increased with decreased sequencing depth. This observation is consistent with the experimental results reported by its authors. Similarly, CoLoRMap did not perform well on large data sets such as D3-P and D3-O as large data sets increase the risk of false positive alignments (discussed previously in Section 4.3.1).

Table 2. Experimental results for E. coli data sets

Data set	Method	# Reads	# Bases (Mbp)	# Aligned reads	# Aligned bases (Mbp)	Maximum length (bp)	N50 (bp)	Genome fraction (%)	Alignment identity (%)	CPU time (hh:mm:ss)	Wall time (hh:mm:ss)	Memory usage (GB)
D1-P	Original	85 460	748.0	82 886	688.0	44 113	13 990	100.000	86.8763	-	-	-
	FLAS	69 327	632.3	68 786	621.2	40 117	13 212	100.000	99.5959	09:47:50	00:56:45	4.9
	LoRMA	330 811	623.3	330 715	623.0	22 499	2441	100.000	99.6814	45:24:49	02:10:36	67.2
	Canu	9283	168.1	9193	166.7	39 693	20 391	100.000	99.6970	07:47:33	00:27:14	6.0
D1-P	Hercules	-	-	-	-	-	-	-	-	-	>72:00:00	-
+	HG-CoLoR	-	-	-	-	-	-	-	-	-	-	-
D1-I	FMLRC	85 260	706.5	83 320	669.9	44 084	13 364	100.000	99.6983	03:05:06	00:30:07	9.8
	HALC	85 256	711.1	84 030	661.7	44 117	13 399	100.000	99.4374	60:41:59	16:02:32	30.2
	CoLoRMap	85 674	730.7	83 765	678.6	44 113	13 641	100.000	95.2930	31:35:16	02:53:33	34.9
	Jabba	77 508	620.2	77 508	619.7	41 342	12 557	99.258	99.9624	02:05:09	00:12:01	37.0
	Nanocorr	73 368	504.9	73 316	493.1	41 079	10 796	100.000	98.3257	1862:59:19	70:57:19	15.1
	proovread	222 354	559.2	222 337	558.7	33 359	4087	100.000	99.9615	68:32:55	14:14:44	53.9
	LoRDEC	85 324	716.9	83 507	665.9	44 311	13 491	100.000	98.4149	15:03:42	00:40:05	2.0
	ECTools	55 687	577.4	55 687	575.7	39 772	13 583	100.000	99.8592	11:25:22	00:29:49	8.2
	LSC	-	-	-	-	-	-	-	-	-	>72:00:00	-
	PacBioToCA	-	-	-	-	-	-	-	-	-	>72:00:00	-
D1-O1	Original	38 919	245.7	21 663	105.1	43 624	8664	100.000	63.4565	-	-	-
	FLAS	404	2.4	397	2.1	22 733	6364	21.155	64.9588	00:08:51	00:03:43	1.7
	LoRMA	-	-	-	-	-	-	-	-	-	-	-
	Canu	-	-	-	-	-	-	-	-	-	-	-
D1-O1	Hercules	38 919	245.7	21 696	105.2	43 624	8666	100.000	63.4856	19:21:43	00:45:27	16.1
+	HG-CoLoR	37 264	262.9	37 258	236.8	73 992	9424	100.000	99.5605	46:04:31	02:21:57	36.9
D1-I	FMLRC	38 909	258.3	31 066	222.9	46 350	9163	100.000	98.9815	03:05:16	00:28:23	9.6
	HALC	39 108	252.6	35 694	139.7	43 714	8874	100.000	85.3297	12:19:51	01:54:56	20.9
	CoLoRMap	39 018	250.8	30 721	151.9	44 638	8836	100.000	77.4829	26:13:04	01:12:00	10.2
	Jabba	17 139	121.9	17 139	121.8	38 395	8807	97.808	99.9780	02:12:28	00:11:52	37.0
	Nanocorr	1605	1.5	1605	1.5	17 174	903	25.729	92.4350	767:46:55	27:37:41	10.5
	proovread	78 172	74.0	78 172	74.0	15 210	950	99.978	99.9165	17:37:12	03:59:56	15.7
	LoRDEC	38 948	251.0	31 604	147.9	44 553	8853	100.000	78.5298	06:17:23	00:19:12	1.8
	ECTools	1488	10.7	1488	10.7	33 223	8038	83.794	99.7331	05:17:17	00:12:59	8.5
	LSC	158	0.7	138	0.5	14 850	6583	10.801	67.0331	41:42:43	01:50:06	3.0
	PacBioToCA	47	0.0	47	0.0	1250	585	0.596	99.6878	02:45:13	01:11:11	3.2
D1-O2	Original	19 534	132.6	19 387	123.6	47 133	9387	100.000	79.9361	-	-	-
	FLAS	15 929	101.7	15 929	100.7	40 893	7714	99.828	90.5239	00:29:56	00:04:35	1.6
	LoRMA	1671	1.4	1671	1.4	2095	936	2.515	97.9661	00:52:13	00:03:44	63.7
	Canu	17 162	121.2	17 162	120.9	44 503	8919	99.862	93.3223	02:16:32	00:11:37	3.0
D1-O2	Hercules	19 522	133.8	19 386	124.8	47 447	9462	100.000	86.8682	130:03:16	05:22:31	8.9
+	HG-CoLoR	19 481	133.9	19 481	131.2	51 724	9462	100.000	99.5425	33:13:57	02:15:03	79.6
D1-I	FMLRC	19 478	133.4	19 417	133.0	46 399	9432	100.000	99.9380	00:54:14	00:23:33	9.7
	HALC	19 518	133.7	19 508	130.3	46 405	9441	100.000	99.7931	08:56:07	01:33:45	12.9
	CoLoRMap	20 084	135.7	20 047	129.3	47 187	9504	100.000	97.0861	15:01:21	00:57:28	10.3
	Jabba	19 455	124.2	19 455	124.1	42 474	9028	98.816	99.9562	02:11:03	00:11:37	37.0
	Nanocorr	18 822	125.2	18 822	121.7	39 244	9107	100.000	96.9823	426:12:04	15:42:56	14.6
	proovread	32 459	125.0	32 459	124.9	40 936	6052	99.978	99.9679	10:33:29	01:50:39	15.4
	LoRDEC	19 514	134.0	19 473	125.4	47 077	9468	100.000	98.4746	03:05:48	00:13:07	1.8
	ECTools	13 698	116.5	13 698	116.5	43 446	9427	100.000	99.8295	04:04:21	00:10:11	8.2
	LSC	17 369	117.6	17 369	113.7	46 990	8873	100.000	88.3439	44:21:47	02:30:34	48.6
	PacBioToCA	-	-	-	-	-	-	-	-	-	>72:00:00	-

Note: LoRMA and Canu failed to produce any corrected reads for D1-O1. HG-CoLoR reported an error when correcting D1-P. The corrected reads generated by PacBioToCA was less than 0.05 million bases for D1-O1.

On the other hand, the performance of non-hybrid methods deteriorated significantly when sequencing depth was decreased. As non-hybrid methods leverage overlap information to correct errors, they require sufficient long read coverage to make true correction. The genome fraction covered by corrected reads produced by LoRMA decreased from 99.59% to 82.97% when sequencing depth dropped from 90x to 60x, and further decreased to 9.61%, 5.39% and 3.78% for 30x, 20x and 10x respectively, implying loss of many long reads after correction. The alignment identities were still greater than 99% using all subsamples because LoRMA trimmed the uncorrected regions. For corrected reads generate by Canu, no significant change on genome fraction was observed. But the alignment identity dropped from above 99% to 97.03% and 95.63% for 20x and 10x sequencing depths, respectively. FLAS showed similar performance but

Table 3. Experimental results for yeast data sets

Data set	Method	# Reads	# Bases (Mbp)	# Aligned reads	# Aligned bases (Mbp)	Maximum length (bp)	N50 (bp)	Genome fraction (%)	Alignment identity (%)	CPU time (hh:mm:ss)	Wall time (hh:mm:ss)	Memory usage (GB)
D2-P	Original	239 408	1462.7	235 620	1332.6	35 196	8656	99.976	87.2637	-	-	-
	FLAS	173 187	1093.2	173 046	1078.8	30 046	8132	99.976	99.5777	11:46:31	01:15:40	7.9
	LoRMA	650 467	1142.0	650 333	1141.4	18 127	2323	99.951	99.7583	172:24:38	07:03:03	72.9
	Canu	38 228	453.2	38 172	446.7	28 748	12 021	99.975	99.5864	15:18:34	00:50:12	6.5
D2-P	Hercules	239 389	1460.3	235 630	1330.4	35 196	8644	99.976	87.6711	87:53:55	03:18:41	247.8
+	HG-CoLoR	-	-	-	-	-	-	-	-	-	-	-
D2-I	FMLRC	238 706	1380.8	236 883	1311.0	33 658	8185	99.977	99.3889	07:52:17	00:28:55	5.5
	HALC	238 787	1395.4	238 097	1287.6	34 785	8270	99.976	99.0796	52:12:11	09:45:10	29.0
	CoLoRMap	239 309	1429.6	237 135	1321.3	34 850	8409	99.976	96.3912	18:44:48	03:07:34	37.3
	Jabba	202 980	1087.2	202 879	1086.6	30 141	7847	95.627	99.9832	00:38:30	00:04:57	21.4
	Nanocorr	-	-	-	-	-	-	-	-	-	>72:00:00	-
	proovread	230 754	376.3	230 649	376.0	26 168	2331	43.503	99.9251	184:02:07	23:45:37	47.9
	LoRDEC	238 847	1405.0	237 278	1297.1	34 896	8326	99.978	97.9568	01:10:03	00:57:17	1.9
	ECTools	130 863	946.9	130 832	943.1	28 749	8412	99.810	99.7712	938:25:28	58:25:00	4.3
	LSC	-	-	-	-	-	-	-	-	-	>72:00:00	-
	PacBioToCA	298 309	975.8	298 304	975.0	28 422	5403	98.564	99.9530	117:02:13	10:40:55	26.2
D2-O1	Original	41 626	382.4	39 742	364.9	56 477	11 696	99.976	87.3194	-	-	-
	FLAS	33 435	314.7	32 875	311.2	56 593	11 312	99.570	96.8164	01:51:06	00:14:32	2.8
	LoRMA	222 611	263.9	221 363	261.6	21 444	1344	90.443	98.5186	47:06:17	02:06:38	65.5
	Canu	34 990	337.1	34 474	331.9	56 946	11 820	99.520	97.4439	7:22:12	00:25:35	5.4
D2-O1	Hercules	-	-	-	-	-	-	-	-	-	>72:00:00	-
+	HG-CoLoR	-	-	-	-	-	-	-	-	-	-	-
D2-I	FMLRC	41 615	390.6	40 276	379.7	58 193	11 969	99.976	99.7439	00:48:38	00:14:34	5.5
	HALC	41 628	391.1	40 705	375.4	58 196	11 980	99.975	99.2888	11:12:25	01:10:50	5.4
	CoLoRMap	41 717	392.1	39 866	376.8	58 557	11 973	99.976	97.2642	08:58:49	01:03:37	18.9
	Jabba	37 205	294.2	37 168	294.0	47 266	10 901	94.892	99.9800	00:40:00	00:05:15	21.4
	Nanocorr	38 996	366.3	38 972	361.7	41 499	11 715	99.975	99.0147	1140:40:36	46:23:26	37.5
	proovread	57 639	172.9	57 611	172.7	35 406	4993	43.482	99.8816	33:05:54	04:14:36	22.9
	LoRDEC	41 626	390.9	39 850	373.7	58 306	11 967	99.758	98.6564	05:29:16	00:15:26	1.7
	ECTools	-	-	-	-	-	-	-	-	-	>72:00:00	-
	LSC	-	-	-	-	-	-	-	-	-	>72:00:00	-
	PacBioToCA	68 254	299.3	68 222	299.1	41 948	7597	99.957	99.8905	48:30:07	04:09:13	16.9
D2-O2	Original	89 273	736.9	69 406	583.7	245 845	11 079	99.976	82.9910	-	-	-
	FLAS	55 821	496.8	55 781	491.2	56 481	10 802	99.759	95.0698	04:09:38	00:31:04	4.6
	LoRMA	373 984	459.6	350 297	435.9	18 555	1416	96.659	98.3725	161:46:36	06:34:53	67.8
	Canu	53 727	473.4	48 366	451.9	56 767	11 314	99.716	96.8725	10:37:59	00:36:21	7.3
D2-O2	Hercules	-	-	-	-	-	-	-	-	-	>72:00:00	-
+	HG-CoLoR	-	-	-	-	-	-	-	-	-	-	-
D2-I	FMLRC	89 268	752.7	73 782	648.7	245 845	11 385	99.976	99.4606	03:31:40	00:18:46	5.4
	HALC	89 293	755.0	78 351	619.8	245 822	11 394	99.976	98.7719	19:01:03	02:06:27	7.4
	CoLoRMap	89 392	753.4	70 147	607.1	245 845	11 346	99.976	94.6220	11:12:47	01:34:47	24.5
	Jabba	63 033	489.7	62 980	489.5	47 266	10 684	95.022	99.9789	00:27:45	00:05:24	21.4
	Nanocorr	-	-	-	-	-	-	-	-	-	>72:00:00	-
	proovread	110 399	219.2	110 328	219.0	35 406	3179	43.512	99.8810	56:07:12	09:04:48	27.7
	LoRDEC	89 284	753.5	71 098	605.5	245 831	11 370	99.976	97.2638	10:42:30	00:25:35	2.0
	ECTools	-	-	-	-	-	-	-	-	-	>72:00:00	-
	LSC	-	-	-	-	-	-	-	-	-	>72:00:00	-
	PacBioToCA	132 633	406.4	132 550	406.1	41 948	5891	99.969	99.8719	67:48:32	05:42:13	19.2

Note: HG-CoLoR could not finish these three tests and reported errors.

genome fraction for 10x sequencing depth was only 90.204%, which is lower than the 99.919% achieved by Canu.

4.3.4 Effect of discarding reads during correction

Many correction tools opt for discarding input reads or regions within reads that they fail to correct. As a result, the reported alignment identity is high (>99%), but much fewer number of bases survive after correction. This effect is more pronounced in corrected reads generated by Jabba, proovread, ECTools, PacBioToCA and LoRMA. They either trim uncorrected regions at sequence ends, or even in the middle, to avoid errors in the final output which eventually yields high alignment identity. However, aggressive trimming also makes the correction lossy and may influence downstream analysis because long range information is lost if the reads are shortened or broken into smaller pieces. Therefore, users should be conservative in trimming and turn it off when necessary. One good practice is to keep the uncorrected regions and let downstream

Table 4. Experimental results for fruit fly data sets

Data set	Method	# Reads	# Bases (Mbp)	# Aligned reads	# Aligned bases (Mbp)	Maximum length (bp)	N50 (bp)	Genome fraction (%)	Alignment identity (%)	CPU time (hh:mm:ss)	Wall time (hh:mm:ss)	Memory usage (GB)
D3-P	Original	5 366 088	28 797.8	1 839 681	16 543.5	74 735	15 374	99.191	85.2734	-	-	-
	FLAS	1 435 682	14 585.2	1 428 018	13 574.1	43 556	13 550	98.915	98.8363	271:44:27	36:30:42	53.1
	LoRMA	-	-	-	-	-	-	-	-	-	-	-
	Canu	-	-	-	-	-	-	-	-	-	>72:00:00	-
D3-P	Hercules	-	-	-	-	-	-	-	-	-	-	-
+	HG-CoLoR	-	-	-	-	-	-	-	-	-	-	-
D3-I	FMLRC	5 246 485	27 354.6	2 477 890	16 543.5	74 735	14 554	99.191	96.5284	327:37:22	13:49:04	31.2
	HALC	4 451 474	21 997.5	3 434 779	12 793.3	74 735	14 349	99.178	96.8863	770:35:46	55:58:24	73
	CoLoRMap	5 366 107	28 891.6	1 841 822	14 976.8	74 735	15 442	99.189	83.2580	495:11:17	64:52:25	189.4
	Jabba	35 549	239.8	35 505	239.1	37 729	10 461	65.616	99.9615	656:05:15	24:33:41	175.8
	Nanocorr	-	-	-	-	-	-	-	-	-	>72:00:00	-
	proovread	-	-	-	-	-	-	-	-	-	>72:00:00	-
	LoRDEC	5 363 998	28 354.1	2 056 812	15 636.9	74 719	15 078	99.200	92.2954	1011:52:27	36:19:18	5.9
	ECTools	-	-	-	-	-	-	-	-	-	>72:00:00	-
	LSC	-	-	-	-	-	-	-	-	-	>72:00:00	-
	PacBioToCA	-	-	-	-	-	-	-	-	-	>72:00:00	-
D3-O	Original	642 255	4609.5	554 083	3857.9	446 050	11 956	98.719	83.5921	-	-	-
	FLAS	423 097	3507.6	422 206	3402.6	64 365	11 517	97.588	95.3301	23:04:50	03:12:50	10.8
	LoRMA	703 097	615.5	682 288	592.3	32 644	865	30.338	98.1230	666:37:35	25:52:14	92.8
	Canu	430 082	3415.6	421 475	3220.2	254 967	12 090	97.592	96.3739	88:51:10	04:36:20	20.2
D3-O	Hercules	642 287	4612.8	554 630	3859.4	449 799	11 966	98.713	83.9340	398:10:17	17:32:36	247.7
+	HG-CoLoR	-	-	-	-	-	-	-	-	-	-	-
D3-I	FMLRC	641 945	4647.2	578 290	3978.2	444 605	12 088	98.592	97.6010	47:45:17	03:06:05	31.2
	HALC	643 002	4668.5	611 191	3955.7	451 284	12 115	98.616	97.6634	126:30:01	05:43:37	42.4
	CoLoRMap	649 041	4692.1	565 881	3963.8	442 948	12 050	98.715	94.3361	160:00:22	16:07:18	57.3
	Jabba	494 546	2878.2	494 430	2876.3	72 501	9305	83.166	99.9745	175:19:34	06:56:29	136.8
	Nanocorr	-	-	-	-	-	-	-	-	-	>72:00:00	-
	proovread	-	-	-	-	-	-	-	-	-	>72:00:00	-
	LoRDEC	642 882	4655.9	567 878	3921.1	447 726	12 079	98.691	94.0382	152:05:32	05:38:05	5.7
	ECTools	-	-	-	-	-	-	-	-	-	>72:00:00	-
	LSC	-	-	-	-	-	-	-	-	-	>72:00:00	-
	PacBioToCA	-	-	-	-	-	-	-	-	-	>72:00:00	-

Note: LoRMA and HG-CoLoR could not finish these two tests and reported segmentation fault. Hercules could not finish the correction of D3-P and reported segmentation fault.

analysis tools perform the trimming, e.g. overlap-based trimming after read correction in Canu.

A direct implication of discarding or trimming reads is the change of read length distribution. Figure 1 and 2 show the original and corrected read length distributions. Among all the tools, Hercules, FMLRC, HALC, CoLoRMAP and LoRDEC can maintain a similar read length distribution after correction whereas Nanocorr, Jabba, ECTools and prooveread lost many long reads after correction due to their trimming step. Nanocorr drops a long read when there is no short read aligning to it. This procedure can remove many error-prone long reads, which leads to a higher alignment identity after correction. However, the fraction of discarded reads in many cases is found to be significant. For example, a mere 1.5 million bp cumulative length of sequences survived out of 245.7 million bp data set, after correction of D1-O1. ECTools also generated only 10.7 million corrected bases using this data set. Canu changed the read length distribution significantly after correction although due to a different reason (Figure 1). Canu estimates the read length after correction and tries to keep the longest 40x reads for subsequent assembly. FLAS kept most of the reads with short length while losing many reads with long length.

Few hybrid-methods managed to generate enough corrected reads with relatively higher alignment identity. Notably, FMLRC and HG-CoLoR substantially outperformed other tools using D1-O1 by producing high alignment identity of 98.98% and 99.56% respectively and maintaining long read lengths (Table 2, Figure 2). Notably, HG-CoLoR generated one extremely long read of length 73,992 bp which is substantially longer than the longest read (43,624 bp) in D1-O1, perhaps due to the use of assembly DBG during the correction process.

4.3.5 Effect of error correction on genome assembly

We examine the effect of error correction on genome assembly, and evaluate if quality of error correction correlates well with the quality of genome assembly performed using corrected reads. To do so, we conducted an experiment to compute genome assembly using corrected PacBio and ONT 2D reads of *E. coli*, i.e., corrected reads for D1-P and D1-O2. Assembly was computed using Canu and its quality was assessed using QUAST (Gurevich *et al.*, 2013); results are shown in Table 5.

Considering the assemblies generated using corrected PacBio reads (D1-P), NGA50 score of >3 million bases was obtained when using reads generated by FLAS, Canu, FMLRC, Nanocorr, LoRDEC or ECTools. Surprisingly, the highest NGA50 was obtained when using corrected reads generated by LoRDEC, but the alignment identity of its corrected reads was lower than most of the tools. Similarly, the highest NGA50 was achieved using corrected reads generated by Canu for D1-O2, but the alignment identity of the corrected reads was 93.32%. Therefore,

Table 5. Results of genome assembly computed using corrected reads of D1-P and D1-O2

Method	# contigs	NGA50 (bp)	Largest contigs (bp)	Total length (bp)	Genome fraction (%)	# misassemblies	# mismatches	# indels (<=5bp)	# indels (>5bp)	Indel length (bp)
Using corrected reads of D1-P										
FLAS	2	3 996 362	4 681 650	4 689 583	99.998	4	4	162	0	167
LoRMA	14	696 878	2 501 146	4 663 900	99.938	4	75	4181	6	4295
Canu	1	3 976 437	4 670 120	4 670 120	99.998	4	7	92	0	95
FMLRC	9	3 821 409	4 657 352	4 831 908	99.998	8	1	4	0	5
HALC	25	2 947 777	4 682 714	5 388 722	99.983	8	541	35	8	257
CoLoRMap	86	1 217 587	1 448 649	5 700 143	99.998	4	42	3	7	478
Jabba	58	138 874	398 327	4 623 296	97.273	1	172	32	3	167
Nanocorr	18	3 095 077	4 646 253	4 931 697	99.998	5	65	34	2	157
proovread	17	695 218	2 446 937	4 693 737	99.855	5	69	17	0	20
LoRDEC	2	3 996 441	4 681 757	4 703 690	99.998	4	66	18	2	55
ECTools	19	3 548 731	4 657 296	5 154 324	99.974	4	592	80	2	188
Using corrected reads of D1-O2										
FLAS	14	409 405	960 036	4 447 245	84.860	6	1948	93 990	2531	177 057
LoRMA	4	n/a	3895	7817	0.170	0	15	100	0	133
Canu	1	3 881 246	4 532 581	4 532 581	99.995	4	2834	66 126	284	97 847
Hercules	29	243 628	581 562	4 610 960	98.786	4	2905	6217	269	13 135
HG-CoLoR	13	646 911	1 124 557	4 685 955	99.009	4	167	19	11	310
FMLRC	2	2 510 453	4 636 115	4 661 890	99.859	4	49	23	1	51
HALC	15	663 256	2 201 303	4 797 115	99.503	4	554	75	5	196
CoLoRMap	3	1 135 017	3 739 474	4 642 333	99.726	4	203	115	0	186
Jabba	57	105 474	311 624	4 460 218	95.838	0	117	22	5	179
Nanocorr	2	3 146 849	3 187 382	4 628 016	99.681	4	85	53	1	96
proovread	2	1 453 125	3 325 887	4 642 017	99.987	2	56	18	0	26
LoRDEC	31	495 790	761 345	4 854 139	98.870	4	1302	359	15	680
ECTools	9	895 512	1 311 398	4 646 949	98.558	4	859	366	5	679
LSC	20	422 885	843 894	4 621 853	99.293	4	4219	4591	170	9314

Note: the tools failed to generate corrected reads for any of the two data sets are excluded.

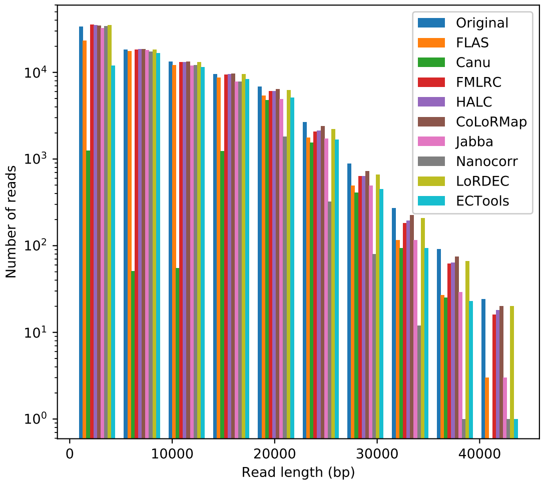

Fig. 1. Corrected read length distribution for D1-P.

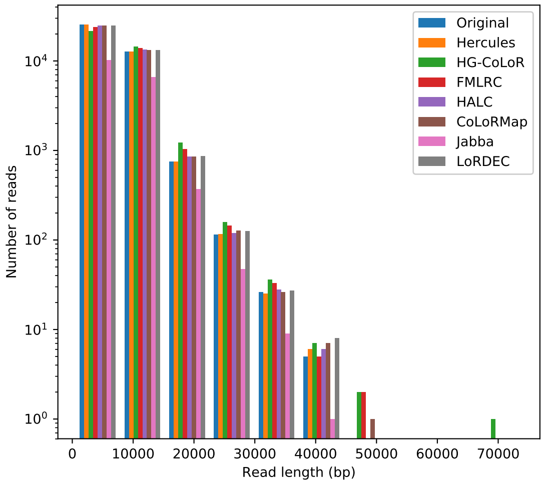

Fig. 2. Corrected read length distribution for D1-O1.

higher alignment identity does not necessarily translate to a better NGA50, i.e., a more continuous assembly.

We also examined the frequency of mismatches and indels in the assemblies. For both data sets D1-P and D1-O2, corrected reads generated by HALC and ECTools produced assemblies with $>$ 500 mismatches, significantly higher than the other tools. However, alignment identity of their corrected reads was either competitive with, or superior to, what is produced by other tools. Notably, both HALC and ECTools use assembled contigs from short reads to do error correction. Mis-assemblies of short reads, especially in repetitive and low-complexity regions, may cause false corrections, which leads to errors during assembly (Wang *et al.*, 2018). Corrected reads produced by FMLRC achieved the least number of errors in assembly. Meanwhile, its alignment identity was also the highest among the methods which avoid trimming. Therefore, higher alignment identity of corrected reads can lead to, but not guarantee, fewer errors in genome assemblies.

Non-hybrid methods such as LoRMA, Canu produced more indels than mismatches in their assemblies while most of the hybrid methods showed the opposite behavior. To further investigate, we visualized the alignments of corrected reads generated by Canu and FMLRC for D1-O2 in Supplementary Figures 1,2, and 3. More indels were observed in the alignments of corrected reads generated by Canu than FMLRC. Moreover, for D1-O2, indels mostly occurred in homoploymers which is consistent with ONT sequencing error profile. These observations suggest that self-correction methods are not good at handling indels when compared to hybrid methods.

5 Conclusions and Future Directions

In this work, we established benchmark data sets and evaluation methods for comprehensive assessment of long read error correction software. Our results suggest that hybrid methods aided by short accurate reads can achieve better correction quality, especially when handling low coverage-depth long reads, compared with non-hybrid methods. Within the hybrid methods, assembly-based methods are superior to alignment-based methods in terms of scalability to large data sets. Besides, better performance on correction such as preserving higher proportion of input bases and better alignment identity may lead to, but cannot guarantee, better results on downstream applications such as genome assembly. The tools with superior correction performance should be further tested in the context of applications of interest, to determine which are best suited for the application of interest.

Users can also select tools according to our experimental results for their specific expectations. When speed is a concern, assembly-based hybrid methods are preferred as long as short reads are available. Besides, hybrid methods are less sensitive to low sequencing depth than non-hybrid methods. Thus, users are recommended to choose hybrid methods when sequencing depth is relatively low. In cases where indel errors may cause a serious negative impact on downstream analyses, hybrid methods should be preferred over non-hybrid ones.

FMLRC outperformed other hybrid methods in almost all the experiments. For non-hybrid methods, Canu and FLAS showed better performance over LoRMA. Hence, these three are recommended as default when users want to avoid laborious tests on all the error correction tools.

For future work, better self-correction algorithms are expected to avoid hybrid sequencing, thus reducing experimental labor on short read sequencing preparation. In addition, most of the correction algorithms run for days to correct errors in the sequencing of even moderately large and complex genomes like the fruit fly, and become a bottleneck in sequencing data analysis. Therefore, more efficient or parallel correction algorithms should be developed to ease the computational burden. Furthermore, none of the hybrid tools makes use of paired-end information in their correction, except CoLoRMap. But the use of paired-end reads in CoLoRMap did not improve correction performance significantly according to previous studies. Paired-end reads have already been used to resolve repeats and remove entanglements in de Bruijn graphs (Bankevich *et al.*, 2012). Since many error correction tools build de Bruijn graphs to correct long reads, the paired-end information may also be able to improve error correction.

Most of the published error correction tools focus on correction of long DNA reads sequenced from a single genome, which also served as the motivation for our review. Long read sequencing is increasingly gaining traction for transcriptomics and metagenomics applications. It is not clear whether the existing tools can be leveraged or extended to work effectively in such scenarios, and is an active area of research (de Lima *et al.*, 2018).

Funding

This work is supported in part by the National Science Foundation under CCF-1718479.

Key Points

- Despite the high error rate of long reads, the state-of-the-art correction tools achieve high correction accuracy and throughput.
- The best hybrid methods show better performance than non-hybrid methods in terms of correction quality and computing resource usage.
- Few correction tools discard reads, which practitioners are supposed to be careful with.
- Evaluation of long read error correction should be conducted while checking its effect on downstream analysis, since better correction quality does not always imply better accuracy of downstream analysis.

Biographical Note

Haowen Zhang and **Chirag Jain** are PhD students in School of Computational Science and Engineering at Georgia Institute of Technology. **Srinivas Aluru, PhD,** is a Professor in School of Computational Science and Engineering and Co-Executive Director of Institute for Data Engineering and Science at Georgia Institute of Technology. He is a Fellow of AAAS and IEEE.

References

Alic, A. S., Ruzafa, D., Dopazo, J., and Blanquer, I. (2016). Objective review of de novo stand-alone error correction methods for NGS data. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, **6**(2), 111–146.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, **215**(3), 403–410.

Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S., Wain, J., and O'grady, J. (2015). Minion nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature biotechnology*, **33**(3), 296.

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

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., *et al.* (2012). Spades: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology*, **19**(5), 455–477.

Bao, E. and Lan, L. (2017). Halc: High throughput algorithm for long read error correction. *BMC bioinformatics*, **18**(1), 204.

Bao, E., Xie, F., Song, C., and Song, D. (2017). Hals: Fast and high throughput algorithm for pacbio long read self-correction.

Berlin, K., Koren, S., Chin, C.-S., Drake, J. P., Landolin, J. M., and Phillippy, A. M. (2015). Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology*, **33**(6), 623.

Carneiro, M. O., Russ, C., Ross, M. G., Gabriel, S. B., Nusbaum, C., and DePristo, M. A. (2012). Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC genomics*, **13**(1), 375.

Chaisson, M. J., Huddleston, J., Dennis, M. Y., Sudmant, P. H., Malig, M., Hormozdiari, F., Antonacci, F., Surti, U., Sandstrom, R., Boitano, M., *et al.* (2015). Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, **517**(7536), 608.

Chin, C.-S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., Clum, A., Copeland, A., Huddleston, J., Eichler, E. E., *et al.* (2013). Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature methods*, **10**(6), 563.

Chin, C.-S., Peluso, P., Sedlazeck, F. J., Nattestad, M., Concepcion, G. T., Clum, A., Dunn, C., O'Malley, R., Figueroa-Balderas, R., Morales-Cruz, A., *et al.* (2016). Phased diploid genome assembly with single-molecule real-time sequencing. *Nature methods*, **13**(12), 1050.

de Lima, L. I. S., Marchet, C., Caboche, S., Da Silva, C., Istace, B., Aury, J.-M., Touzet, H., and Chikhi, R. (2018). Comparative assessment of long-read error-correction software applied to RNA-sequencing data. *bioRxiv*, page 476622.

Fichot, E. B. and Norman, R. S. (2013). Microbial phylogenetic profiling with the pacific biosciences sequencing platform. *Microbiome*, **1**(1), 10.

Firtina, C., Bar-Joseph, Z., Alkan, C., and Cicek, A. (2018). Hercules: a profile hmm-based hybrid error correction algorithm for long reads. *Nucleic Acids Research*, page gky724.

Giordano, F., Aigrain, L., Quail, M. A., Coupland, P., Bonfield, J. K., Davies, R. M., Tischler, G., Jackson, D. K., Keane, T. M., Li, J., *et al.* (2017). De novo yeast genome assemblies from MinION, PacBio and MiSeq platforms. *Scientific reports*, **7**(1), 3935.

Goodwin, S., Gurtowski, J., Ethe-Sayers, S., Deshpande, P., Schatz, M. C., and McCombie, W. R. (2015). Oxford nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. *Genome research*, **25**(11), 1750–1756.

Gordon, S. P., Tseng, E., Salamov, A., Zhang, J., Meng, X., Zhao, Z., Kang, D., Underwood, J., Grigoriev, I. V., Figueroa, M., *et al.* (2015). Widespread polycistronic transcripts in fungi revealed by single-molecule mRNA sequencing. *PloS one*, **10**(7), e0132628.

Gurevich, A., Saveliev, V., Vyahhi, N., and Tesler, G. (2013). Quast: quality assessment tool for genome assemblies. *Bioinformatics*, **29**(8), 1072–1075.

Hackl, T., Hedrich, R., Schultz, J., and Förster, F. (2014). proovread: large-scale high-accuracy pacbio correction through iterative short read consensus. *Bioinformatics*, **30**(21), 3004–3011.

Haghshenas, E., Hach, F., Sahinalp, S. C., and Chauve, C. (2016). Colormap: Correcting long reads by mapping short reads. *Bioinformatics*, **32**(17), i545–i551.

Jain, M., Fiddes, I. T., Miga, K. H., Olsen, H. E., Paten, B., and Akeson, M. (2015). Improved data analysis for the minion nanopore sequencer. *Nature methods*, **12**(4), 351.

Jain, M., Koren, S., Miga, K. H., Quick, J., Rand, A. C., Sasani, T. A., Tyson, J. R., Beggs, A. D., Dilthey, A. T., Fiddes, I. T., *et al.* (2018). Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature biotechnology*, **36**(4), 338.

Koren, S., Schatz, M. C., Walenz, B. P., Martin, J., Howard, J. T., Ganapathy, G., Wang, Z., Rasko, D. A., McCombie, W. R., Jarvis, E. D., *et al.* (2012). Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature biotechnology*, **30**(7), 693.

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H., and Phillippy, A. M. (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research*, **27**(5), 722–736.

Korlach, J. and Biosciences, P. (2013). Understanding accuracy in SMRT® sequencing.

Kowalski, T., Grabowski, S., and Deorowicz, S. (2015). Indexing arbitrary-length k-mers in sequencing reads. *PloS one*, **10**(7), e0133198.

La, S., Haghshenas, E., and Chauve, C. (2017). Lrcstats, a tool for evaluating long reads correction methods. *Bioinformatics*, **33**(22), 3652–3654.

Laehnemann, D., Borkhardt, A., and McHardy, A. C. (2015). Denoising DNA deep sequencing data high-throughput sequencing errors and their correction. *Briefings in bioinformatics*, **17**(1), 154–179.

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

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **1**, 7.

Loman, N. J. and Quinlan, A. R. (2014). Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, **30**(23), 3399–3401.

Loman, N. J., Quick, J., and Simpson, J. T. (2015). A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature methods*, **12**(8), 733.

Madoui, M.-A., Engelen, S., Cruaud, C., Belser, C., Bertrand, L., Alberti, A., Lemainque, A., Wincker, P., and Aury, J.-M. (2015). Genome assembly using nanopore-guided long and error-free DNA reads. *BMC genomics*, **16**(1), 327.

Mahmoud, M., Zywicki, M., Twardowski, T., and Karlowski, W. M. (2017). Efficiency of pacbio long read correction by 2nd generation illumina sequencing. *Genomics*.

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

Morisse, P., Lecroq, T., Lefebvre, A., and Berger, B. (2018). Hybrid correction of highly noisy long reads using a variable-order de bruijn graph. *Bioinformatics*.

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

Rand, A. C., Jain, M., Eizenga, J. M., Musselman-Brown, A., Olsen, H. E., Akeson, M., and Paten, B. (2017). Mapping DNA methylation with high-throughput nanopore sequencing. *Nature methods*, **14**(4), 411.

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

Salmela, L., Walve, R., Rivals, E., and Ukkonen, E. (2016). Accurate self-correction of errors in long reads using de bruijn graphs. *Bioinformatics*, **33**(6), 799–806.

Sedlazeck, F. J., Rescheneder, P., Smolka, M., Fang, H., Nattestad, M., von Haeseler, A., and Schatz, M. C. (2017). Accurate detection of complex structural variations using single molecule sequencing. *Preprint at https://www. biorxiv. org/content/arly/2017/07/28/169557*.

Sedlazeck, F. J., Lee, H., Darby, C. A., and Schatz, M. C. (2018). Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics*, page 1.

Simpson, J. T., Workman, R. E., Zuzarte, P., David, M., Dursi, L., and Timp, W. (2017). Detecting DNA cytosine methylation using nanopore sequencing. *Nature methods*, **14**(4), 407.

Stöcker, B. K., Köster, J., and Rahmann, S. (2016). Simlord: simulation of long read data. *Bioinformatics*, **32**(17), 2704–2706.

Wang, J. R., Holt, J., McMillan, L., and Jones, C. D. (2018). Fmlrc: Hybrid long read error correction using an fm-index. *BMC bioinformatics*, **19**(1), 50.

Xiao, C.-L., Chen, Y., Xie, S.-Q., Chen, K.-N., Wang, Y., Han, Y., Luo, F., and Xie, Z. (2017). Mecat: fast mapping, error correction, and de novo assembly for single-molecule sequencing reads. *nature methods*, **14**(11), 1072.

Yang, X., Dorman, K. S., and Aluru, S. (2010). Reptile: representative tiling for short read error correction. *Bioinformatics*, **26**(20), 2526–2533.

Yang, X., Chockalingam, S. P., and Aluru, S. (2012). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, **14**(1), 56–66.