6

Comprehensive Evaluation of Error-Correction Methodologies for Genome Sequencing Data

Yun Heo • Gowthami Manikandan • Anand Ramachandran • Deming Chen

Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Author for correspondence: Deming Chen, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA.
Email: dchen@illinois.edu

Doi: https://doi.org/10.36255/exonpublications.bioinformatics.2021.ch6

Abstract: Next generation sequencing (NGS) technologies like Illumina and third generation sequencing (TGS) technologies like PacBio and Oxford Nanopore Technology use different techniques for sequencing and provide reads of different lengths and error profiles. Many tools exist for error correction of such sequencing data, improving the quality of downstream analyses. In this chapter, we evaluate the performance of 23 error-correction tools, providing insight into their strengths and weaknesses. This is accomplished through a set of algorithms we have developed and implemented as SPECTACLE, a Software Package for Error Correction Tool Assessment on nuCLEic acid sequences, and a dataset for NGS and TGS reads that we compiled emphasizing challenging scenarios for error correction tools. This chapter provides the reader an understanding of available tools, including advice on selecting appropriate tools for different circumstances. It also provides insights regarding aspects of sequencing data to be addressed to improve tool accuracy.

Keywords: error analysis; error correction; error correction evaluation; next generation sequencing; third generation sequencing

In: *Bioinformatics*. Nakaya HI (Editor). Exon Publications, Brisbane, Australia.
ISBN: 978-0-6450017-1-6; Doi: https://doi.org/10.36255/exonpublications.bioinformatics.2021

INTRODUCTION

Rapid improvements in recent years have given rise to next-generation sequencing (NGS) technologies that provide low-cost, high-throughput sequencing data. However, sequencing data is error-prone and errors in NGS reads degrade the quality of downstream analyses. Many methods for correcting errors in reads have been developed (1–16) which can improve downstream analyses (17–19). NGS is also applied to transcriptomic analysis (20). RNA sequencing data also has sequencing errors, and error-correction needs to account for different factors compared to DNA sequencing reads, such as non-uniform expression levels and alternative splicing. As a result, separate error-correction methods are developed (21) for RNA sequencing data.

Several third-generation sequencing (TGS) technologies have been developed, providing reads tens of thousands of bases long. Pacific Biosciences' Single-molecule real-time (SMRT) sequencing and Oxford Nanopore (ONT) sequencing are popular TGS methods. TGS reads can have relatively high error rates. SMRT Continuous Long Read (CLR) technology, emphasizing the longest read lengths, has over 10% error rate (22). ONT's MinION reads can have over 35% error rate (23). The dominant TGS errors are indels that are rare in Illumina reads. Consequently, error-correction methods for such PacBio (24–27) and ONT (28, 29) reads have been developed.

Despite many error-correction methods, only a few studies exist that are dedicated to the evaluation of the accuracy of these methods. This is due to the difficulty involved in discerning how many errors were corrected and how many were newly generated in the error-correction process. While checking if substitution errors have been corrected is straight-forward, it is not simple to evaluate how well indel-type errors are corrected. Tools may also trim reads which cannot be corrected, causing additional complications. Comparing corrected and uncorrected read alignments to the source genome is not the right solution as multiple best alignments can exist (6). Heterozygous sites are another difficulty.

There exist a few methods to compare error-correction methods for NGS reads. Error Correction Evaluation Toolkit (ECET) (30) consists of two software packages, one of which evaluates Illumina reads and the other, 454 or Ion Torrent reads. The packages are written for the dominant error types of the corresponding sequencing platform – substitutions for Illumina, and indels for 454 and IonTorrent (31, 32). Another evaluation work by Molnar *et al.* (33) calculates how many error-free reads or *k*-mers cover each base in a genome, and how many bases in a reference sequence are covered by error-free reads or *k*-mers, then checks how the two numbers are changed by error correction. Fiona (6) aligns both a read and its corrected version to a reference sequence to compare the two edit distances.

While these methods enable error-correction evaluation, they have limitations. ECET requires Illumina error-correction tools to explicitly annotate trimmed bases, and even then, additional processing is required. ECET's evaluation of indel errors in 454 or Ion Torrent reads may be inaccurate for trimmed reads (34). Molnar *et al*'s methods (33) may not be applicable to TGS reads where error-free *k*-mers of sufficient length may be difficult to find due to high error-rate of TGS, and shorter *k*-mers can obfuscate the results due to repetitions. Fiona's routines,

dependent on alignment, may make false evaluations where multiple best alignments exist, each implying a different error-correction result (34).

In addition to these methods, evaluations are presented when individual error-correction tools are introduced. For Illumina tools, quality of error correction is evaluated using counts of uncorrected errors and corrected errors and expressed in the form of metrics such as sensitivity, precision, and gain. The error counts used for calculating these metrics are obtained by mapping the reads to the reference (30). Such mapping-based methods can give inaccurate evaluation results due to the existence of multiple best alignments. Literature reporting new TGS error-correction tools on the other hand, do not typically report error count metrics such as sensitivity or gain. Instead, performance is measured based on improvements in downstream assembly and alignment results. While such results give a good picture of how much improvement is made by error correction, there can be variations in such measurements based on the specific assembly or alignment tools used to obtain these metrics.

This chapter addresses limitations of existing error-correction evaluation methods and introduces a new algorithm called SPECTACLE (Software Package for Error Correction Tool Assessment on nuCLEic acid sequences). SPECTACLE allows uniform and standardized error-correction evaluations across different sequencing technologies. We introduce new metrics for error-correction tool study that provide insights regarding design limitations of a tool, giving pointers on how the tool may be improved. Using these methods, we perform a comprehensive analysis of many error correction tools and report error count statistics, alignment and assembly statistics, as well as additional metrics for understanding tool behavior. Specifically, the following features and contributions are covered in this chapter:

(i) A new error-correction tool evaluation algorithm that works across sequencing technologies, error models and error rates. It works for both DNA and RNA sequencing data, and for NGS and TGS reads.
(ii) Methods to design input reads for error-correction evaluation that highlight the challenges in error correction such as heterozygosity, coverage variation, and repeats.
(iii) Evaluation of other error-correction tools for NGS and TGS reads using these methods. From SPECTACLE, we report error-correction statistics like sensitivity, precision, percentage similarity, NG50 length, supporting read coverage, alignment quality of corrected reads, point-sensitivity etc. for both NGS and TGS reads.

In the following sections, we discuss the error-correction algorithm, the data preparation methods, and the evaluation results.

EVALUATION STEPS

Figure 1 shows the SPECTACLE flows for evaluating error-correction tools with DNA simulated reads and DNA real reads. Each flow consists of two steps. In the first step, the locations of errors in input reads are determined, and in the next

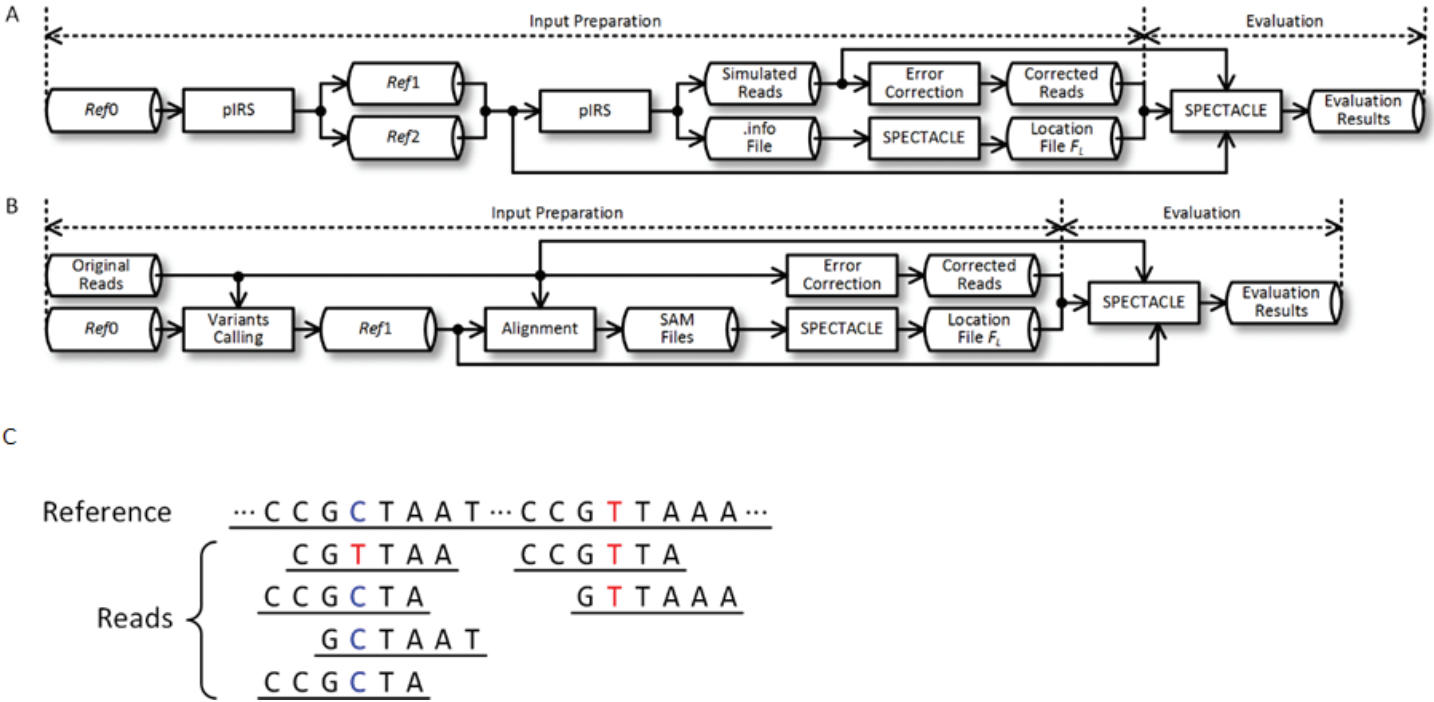

Figure 1. SPECTACLE flows for evaluating error-correction. A. Evaluation flow for simulated reads. **B.** Evaluation flow for real reads. **C.** Supporting reads and supporting read coverage. Supporting reads are the reads that include a specific position of the genome with a specific base at the position. On the left, there is a read CGTTAA with an erroneous base T, and three more correct reads are also sampled there. In this example, the number of supporting reads (supporting read coverage) for T at that position of the reference sequence is 1, while supporting read coverage for C is 3. However, there is another similar sequence in the genome (due to repeats) and the reads sampled at the right region could be supporting read for T on the left, which makes it hard to correct the error. Differential supporting read coverage is defined as (supporting read coverage of correct base) - (supporting read coverage for the erroneous base).

step this information is used to evaluate the output of an error-correction tool. A similar flow for RNA error-correction tools is also available in SPECTACLE (34).

Preparing input data

SPECTACLE supports both simulated reads and real reads to utilize their unique strengths. With simulated reads, we can determine the exact locations of errors in the reads. Moreover, reads can be generated from two different reference sequences in order to simulate diploid genomes.

The biggest advantage of using real reads is that no assumptions or modeling artifacts affect the data. Therefore, real reads can have some interesting properties that may not be accurately modeled in simulated reads. On the other hand, there can be ambiguities in finding error locations in real reads. To find error locations in real reads, the reads need to be aligned to a reference sequence, and this can cause some problems. As explained before, a read can have multiple equivalent alignments to the reference, and determining the correct alignment is sometimes impossible. In the case of highly repetitive genomes, ambiguous alignments occur frequently. Second, reads and a reference sequence might come from different samples, and the differences between them (variants) may be recognized as errors in this step without careful analysis. Third, the evaluation results will depend on the accuracy of the alignment tool.

SPECTACLE can work with the output reads from any read simulator that gives error location information in a Sequence Alignment/Map (SAM) format. In our study we used pIRS (35) exclusively for generating simulated Illumina reads. Error correction becomes challenging when there are heterozygous sites and read coverage variations (19, 36), and pIRS can be used to simulate both. Figure 1A depicts the evaluation flow for simulated reads. First, two reference sequences *Ref*1 and *Ref*2 that represent a pair of chromosomes in a diploid genome are generated by adding different variant sets to the input reference sequence *Ref*0. Once the two sequences are created, reads are generated from *Ref*1 and *Ref*2. The maximum ploidy level that SPECTACLE supports is two. After the reads are generated, the locations of errors in the reads should be written in an error location file F_L. F_L contains: (i), the positions where reads originate in the genome; (ii), the locations of substitutions, insertions, and deletions in each read, and (iii), reference sequence from which each read was sampled (*Ref*1 or *Ref*2). When pIRS generates reads, it also produces a file containing the error locations (.info file) and the .info file is converted into F_L.

To simulate PacBio reads, we used PBSIM (37). PBSIM generates a Mutation Annotation Format (MAF) file for indicating error locations, and the file is converted to F_L. Since PacBio does not use amplification techniques, coverage variation due to different GC-content values was not considered in generating the simulated reads for PacBio. Also, PacBio reads need only be generated from a single reference sequence, unlike Illumina reads. This is because the error rate in the reads is much higher than the frequency of heterozygous sites, and we do not expect the evaluation results to be altered appreciably by simulating heterozygous sites.

Figure 1B shows the evaluation flow for real reads. As mentioned before, if input reads and a reference sequence *Ref*0 do not come from the same sample, there can be variants between them; the variants should not be recognized later in

the flow as sequencing errors. To overcome this problem, a new reference sequence, *Ref*1, is generated by calling the variants and applying them to *Ref*0. In our evaluation, BWA (38) and SAMtools (39) were used for variant calling. The variants are added to *Ref*0 using VCFtools (40), the input reads are aligned to *Ref*1, and the alignment results in the SAM file are converted to F_L. Among the substitution errors in F_L, the errors falsely created due to heterozygous variants are removed by comparing F_L with the variant calling result.

Procedures for evaluating error-correction accuracy

Let R_C be the corrected version of a read R. To evaluate the accuracy of R_C, we should find corrected errors and newly added errors in R_C. SPECTACLE first takes the segment G_R from a reference sequence where read R was sampled. Then, R_C is aligned to G_R to find the errors in R_C. For Illumina reads, we implemented a modified version of the Gotoh algorithm (41) for handling trimmed bases and for extracting all the alignments with the best alignment score (34).

There can be a set of alignments, ALN_{BEST}, having the highest alignment score for a read R_C, but each alignment could imply different numbers of corrected and newly introduced errors. SPECTACLE introduces criteria that rank each of the alignments in ALN_{BEST} based on the error-correction accuracy in each case. Specifically, SPECTACLE calculates a penalty score based on newly introduced errors for each alignment in ALN_{BEST}, utilizing the scores used in the alignment step. Then, the alignment, aln_{BEST}, from ALN_{BEST} that has the least penalty is chosen. SPECTACLE makes the choice using the following equation, where $ERR(aln)$ and $ERR(R)$ are the sets of errors in an alignment aln and R and $ERR(aln)\backslash ERR(R)$ is the set of errors in aln but not in R.

$$aln_{BEST} = \underset{aln \in ALN_{BEST}}{\operatorname{argmax}} \sum_{err \in (ERR(aln) \backslash ERR(R))} penalty(err)$$

We can compute from aln_{BEST} how many errors in $ERR(R)$ are corrected and how many errors are newly added during correction. Since aln_{BEST} is computed through enumeration, the routine runs fast enough for NGS reads but not for long, high error-rate TGS reads for which ALN_{BEST} can be large.

Hence for TGS reads, we implemented a simplified dynamic programming version of this algorithm combining the alignment step with the step enumerating aln_{BEST} for faster execution. This allows us to compute the same error-correction metrics as for NGS reads, albeit with a more limited alignment scoring option. We also introduce later in the chapter metrics that are tailored for TGS reads.

In order to classify the bases in input reads, we introduce a notation consisting of a triplet, each character of which is either Y or N. The first character indicates whether the base in the original read is correct (Y) or not (N), the second character indicates whether the base has been modified by an error correction tool (Y) or not (N), and the third one indicates whether the base in the corrected read at that position is correct (Y) or not (N). For example, NYY describes a base that is erroneous in R, modified by an error correction tool, and error-free in R_C. All the bases should fall into one of the five categories: NNN, NYN, NYY, YNY, and YYN (YYY,

YNN, and NNY are inconsistent). Let η_x be the number of bases in a corrected read set of type x, where $x \in$ {NNN, NYN, NYY, YNY, YYN}. Then SPECTACLE calculates the following error-correction metrics.

$$Sensitivity = \frac{\eta_{NYY}}{\left(\eta_{NYY} + \eta_{NYN} + \eta_{NNN}\right)}$$

$$Gain = \frac{\eta_{NYY} - \eta_{YYN} - \eta_{NYN}}{\left(\eta_{NYY} + \eta_{NYN} + \eta_{NNN}\right)}$$

$$Specificity = \frac{\eta_{YNY}}{\left(\eta_{YYN} + \eta_{YNY}\right)}$$

$$Precision = \frac{\eta_{NYY}}{\left(\eta_{NYY} + \eta_{YYN} + \eta_{NYN}\right)}$$

$$F - score = 2\eta_{NYY}/(\eta_{NYY} + \eta_{YYN} + 2\eta_{NYN} + \eta_{NNN})$$

SPECTACLE can calculate and report the percentage similarity of reads for error-correction evaluation. This feature is mainly intended for long TGS reads. Percentage similarity of a read set, S_R, is defined using the following equation, where N_{RM}, N_{RMM}, N_{RI}, and N_{RD} are the number of matched bases, the number of mismatched bases, the number of inserted bases, and the number of deleted bases in the alignment result of R, respectively:

$$\text{Percentage Similarity} = \sum_{R \in S_R} \frac{N_{RM}}{N_{RM} + N_{RMM} + N_{RI} + N_{RD}}$$

SPECTACLE calculates percentage similarity both for input reads and for their error-correction results and shows how this number is improved after error correction. Most TGS error-correction methods trim uncorrected regions in reads. After this process, R_C could be split into multiple pieces and become much shorter than R. To capture this effect, SPECTACLE also reports read coverage that indicates how much data is retained after trimming and NG50 (17) that shows how long the average read length is.

SPECTACLE can report other detailed analyses such as supporting read coverage which helps users understand the characteristics of an error-correction tool in depth. Figure 1C explains supporting read, supporting read coverage, and differential supporting read coverage. An error in a read becomes difficult to correct if the corresponding correct base has low supporting read coverage, since error-correction tools recognize bases with low supporting read coverage as errors. Low differential supporting read coverage, which implies that both correct and erroneous bases have similar support, also makes error

correction harder. SPECTACLE gives the percentage of corrected bases against supporting read coverage for correct bases, and differential supporting read coverage. This helps in evaluating how sensitive an error correction tool is to variations in read coverage.

SPECTACLE collects the percentage of corrected bases in each position of reads (point sensitivity). Based on this, users can judge whether an error-correction tool can correct errors in a specific region of reads or not. This report can allow SPECTACLE users to discern how the output of an error-correction tool can be polished further, how multiple error-correction algorithms can be combined, and how an error-correction algorithm can be improved.

SPECTACLE also reports measurements that provide an idea about how good the corrected reads are in the context of downstream analyses. One potential method is to count the number of corrected reads that can be aligned to a reference sequence without mismatches or indels. However, this result can be misleading when reads are aligned to wrong parts of a reference sequence. To avoid this, SPECTACLE has the capability to compare the aligned locations of reads with F_L. If insertions or deletions in a read are corrected, the aligned position of the read can be shifted. SPECTACLE determines the largest possible amount of shift in the aligned positions for each read using the number of insertions and deletions, and then reports the number of reads aligned correctly within this predicted range.

The average number of times each base in the reference sequence is covered by error-free reads (i.e. error-free read coverage) and the fraction of a reference sequence that is covered by error-free reads (i.e. chromosome coverage) are important metrics that indicate the quality of a read set (33). SPECTACLE collects the two numbers using the exact alignment result described above.

EXPERIMENTS

We evaluated 17 Illumina read error-correction tools, four PacBio and two ONT read error-correction methods using SPECTACLE. All the experiments were done on a cluster, each computing node of which had two six-core Intel Xeon X5650 processors and 24 GB of memory. In the following sections, we include only selected results that highlight the strengths and weaknesses of the tools. The remaining results, software versions, and software command line options are available in our extended manuscript (34).

Preparing Illumina read sets

I1, I2, and I3 are *E. coli* bacterium genomes that have different GC-content values. I4 is the mouse chromosome Y known as a highly repetitive genome (42). I5 is human chromosome 1, the largest genome sequence used in our experiments. To evaluate the results for real reads, we downloaded I6 from the Illumina website (http://www.illumina.com/systems/miseq/scientific_data.ilmn [accessed on 27 March 2015]). The reads from this dataset have been sequenced from the exact same strain as I2 using the Illumina MiSeq sequencer and down-sampled to 40X.

Preparing PacBio read sets

The PacBio error-correction tools evaluated in this study require, in addition to PacBio reads, Illumina reads as well, since the PacBio error-correction tools use Illumina short reads to detect and correct errors. To evaluate the effect of Illumina read coverage on the accuracy of error correction for PacBio reads, we prepared four different Illumina read sets with different read coverage values corresponding to each set of PacBio reads.

We prepared two PacBio read sets named P1, and P2. Accompanying each PacBio read set are Illumina read sets with coverages in the range 10X-40X in increments of 10X. In the sequel these are suffixed -10X, -20X, -30X, and -40X. 40X-EF is an error-free version of the 40X Illumina read set and was used to evaluate the effects of sequencing errors in Illumina reads on error correction for PacBio reads.

P1 is *E. coli* K12 M1665 strain. Both the PacBio reads and the Illumina reads are real reads. The PacBio reads were downloaded from Pacific Biosciences DevNet (https://github.com/PacificBiosciences/DevNet/wiki/E%20coli%20K12%20MG1655%20Hybrid%20Assembly [accessed on 29 March 2015]). Illumina read sets were generated from SRR922409. P2 is the first 10 Mbp region of human chromosome 19, which was used for evaluating the scalability of the PacBio error correction tools. The PacBio CLR reads and the Illumina reads for P2 were simulated using PBSIM and pIRS, respectively.

Preparing ONT read sets

ONT error-correction tools also use short Illumina reads for error correction, similar to methods for PacBio error correction. We prepared two ONT read sets: O1 and O2, with accompanying Illumina reads of coverages 10X, 20X, and 30X. Both ONT read sets are real reads. O1 is *E. coli K12 M1665* strain. The raw reads were downloaded from GigaDB (http://gigadb.org/dataset/view/id/100102/token/S30Hp9ZurcARyhov [accessed on 6 March 2017]). O2 is *Saccharomyces cerevisiae W303* strain downloaded from the NCBI Sequence Read Archive (http://www.ncbi.nlm.nih.gov/sra [accessed on 9 Nov 2016]). Illumina reads for both these datasets were downloaded from Illumina BaseSpace (SRR567755). In addition, we simulated error-free versions of the 30X Illumina reads using pIRS.

Additional details regarding our datasets are provided in our extended manuscript (34).

Running Illumina read error-correction tools

The input read sets were corrected using the 17 error-correction tools. Among these, the stand-alone error correction tools are BFC (1), BLESS (2), Blue (3), Coral (4), HiTEC (7), Fiona (6), Lighter (8), Musket (9), Quake (10), QuorUM (11), RACER (12), Reptile (13), Trowel (14) and ECHO (5). The remaining three tools are parts of DNA assemblers, ALLPATHS-LG (43), SGA (44), and SOAPdenovo (45).

For each error-correction method, we applied successive numbers to the key parameters of the tools, and generated multiple corrected output read sets

corresponding to each parameter. The output read sets were assessed using SPECTACLE and the one that had the highest gain for substitutions, insertions, and deletions was chosen. The maximum *k*-mer length for Quake was limited to 18 beyond which the memory capacity of our server was exhausted.

ALLPATHS-LG, BFC, BLESS, Blue, Musket, Quake, QuorUM, RACER, Reptile, SGA, and SOAPec succeeded in generating outputs for all the input read sets. Coral, HiTEC, Fiona, and Trowel failed to correct errors in large genomes because of insufficient memory. ECHO had not finished after 70 hours for the I4 and I5 read sets. Lighter finished correcting all the read sets but it made no correction for the read sets with 10X coverage.

Running TGS read error-correction tools

PacBio read error-correction tools LoRDEC (24), LSC (46), PBcR (26), and Proovread (27) were evaluated using P1 and P2. No parameter tuning was needed for LSC, PBcR, and Proovread. For LoRDEC, we generated multiple output sets by applying successive values for *k*-mer length and solid *k*-mer occurrence threshold and chose the result that gave the highest percentage similarity. We could not assess LSC using P2 because it had not finished running after 70 hours. For ONT, we evaluated two error correction technologies NanoCorr (29) and NaS (28) using O1 and O2. Default parameters were used for these two error-correction methods.

Accuracy of Illumina error-correction tools

Sensitivity and gain for substitution type errors for Illumina experiments are summarized in Table 1. For I1, I2, and I3, ALLPATHS-LG, BLESS, Lighter, Musket, Quake, QuorUM, and SGA generated outputs with gain above 0.95. For the highly repetitive genome I4, only BLESS and Quake obtained gain above 0.8. For I5-40X, the largest input genome, ALLPATHS-LG, BFC, BLESS, Lighter, Musket, Quake, QuorUM, and SGA showed gain above 0.9. Other than BFC, these are the same tools that worked well for I1-I3. For I6, most tools performed similarly to I2, both of which were generated from *B. cereus*. However, Coral, Quake, Reptile, SOAPec, and Trowel showed a degradation of above 0.1 for the gain value in I6 when compared with I2.

Differences in sensitivity and gain are measures of the number of false corrections made by each tool. In general, BFC, BLESS, Quake, SGA, and SOAPec generated fewer false corrections than the others.

Table 1 shows variation in accuracy with coverage for different versions of I5. Only BLESS, Musket, and Quake had gain over 0.85 for all the read sets. Lighter showed good results for 20-40 X reads, but it could not correct the errors in I5-10X. BFC, BLESS, Musket, Quake, SGA, and SOAPec made a small number of false corrections for low coverage read sets. Gain was saturated in most tools at 30X coverage.

The percentage of corrected bases as a function of supporting read coverage for I5-40X is shown in Figure 2A. ALLPATHS-LG, Quake, and QuorUM corrected more errors than others when supporting read coverage of correct bases was close to 1. Even though ALLPATHS-LG and QuorUM are capable of correcting errors

TABLE 1 Sensitivity and Gain for Illumina error-correction

	I1-40X		I2-40X		I3-40X		I4-40X		I5-40X		I6		I5-10X		I5-20X		I5-30X		I5-40X	
Software	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain	Sens.	Gain
ALLPATHS-LG	0.998	0.983	0.998	0.984	0.99	0.966	0.851	0.69	0.969	0.904	0.96	0.958	0.911	0.811	0.964	0.886	0.968	0.897	0.969	0.904
BFC	0.964	0.964	0.96	0.959	0.948	0.94	0.777	0.711	0.934	0.92	0.981	0.979	0.81	0.749	0.919	0.891	0.929	0.912	0.934	0.92
BLESS	0.998	0.997	0.998	0.998	0.99	0.983	0.905	0.855	0.975	0.964	0.979	0.977	0.931	0.898	0.961	0.946	0.975	0.96	0.975	0.964
Blue	0.998	0.961	0.998	0.97	0.981	0.883	0.85	0.52	0.896	0.819	0.982	0.903	0.848	0.69	0.894	0.809	0.896	0.818	0.896	0.819
Coral	0.979	0.913	0.987	0.934	N/A	N/A	N/A	N/A	N/A	N/A	0.817	0.806	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ECHO	0.831	0.784	0.949	0.9	0.856	0.803	N/A	N/A	N/A	N/A	0.831	0.822	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Fiona	0.998	0.973	0.998	0.98	0.984	0.902	0.677	0.237	N/A	N/A	0.97	0.967	0.942	0.837	N/A	N/A	N/A	N/A	N/A	N/A
HiTEC	0.997	0.982	0.997	0.993	N/A	N/A	N/A	N/A	N/A	N/A	0.965	0.959	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Lighter	0.995	0.992	0.996	0.995	0.974	0.966	0.656	0.586	0.939	0.913	0.973	0.971	N/A	N/A	0.918	0.867	0.938	0.907	0.939	0.913
Musket	0.996	0.995	0.996	0.995	0.973	0.964	0.773	0.698	0.909	0.886	0.958	0.955	0.889	0.86	0.905	0.882	0.907	0.885	0.909	0.886
Quake	0.988	0.988	0.99	0.99	0.973	0.97	0.856	0.83	0.92	0.913	0.738	0.736	0.908	0.896	0.917	0.91	0.92	0.912	0.92	0.913
QuorUM	0.999	0.997	0.999	0.998	0.981	0.969	0.779	0.709	0.951	0.925	0.982	0.977	0.894	0.81	0.952	0.907	0.952	0.922	0.951	0.925
RACER	0.996	0.913	0.997	0.968	0.961	0.708	0.587	–0.1	0.902	0.114	0.967	0.946	0.819	–2.29	0.898	–0.16	0.902	0.052	0.902	0.114
Reptile	0.958	0.933	0.968	0.96	0.926	0.824	0.672	0.562	0.878	0.76	0.852	0.831	0.805	0.612	0.869	0.728	0.876	0.754	0.878	0.76
SGA	0.996	0.996	0.996	0.996	0.975	0.968	0.738	0.673	0.959	0.939	0.947	0.944	0.852	0.803	0.941	0.917	0.955	0.936	0.959	0.939
SOAPec	0.671	0.67	0.664	0.664	0.65	0.648	0.478	0.446	0.624	0.614	0.539	0.538	0.585	0.545	0.622	0.609	0.624	0.613	0.624	0.614
Trowel	0.817	0.814	0.836	0.833	0.835	0.818	0.599	0.469	N/A	N/A	0.677	0.675	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Sens., Sensitivity

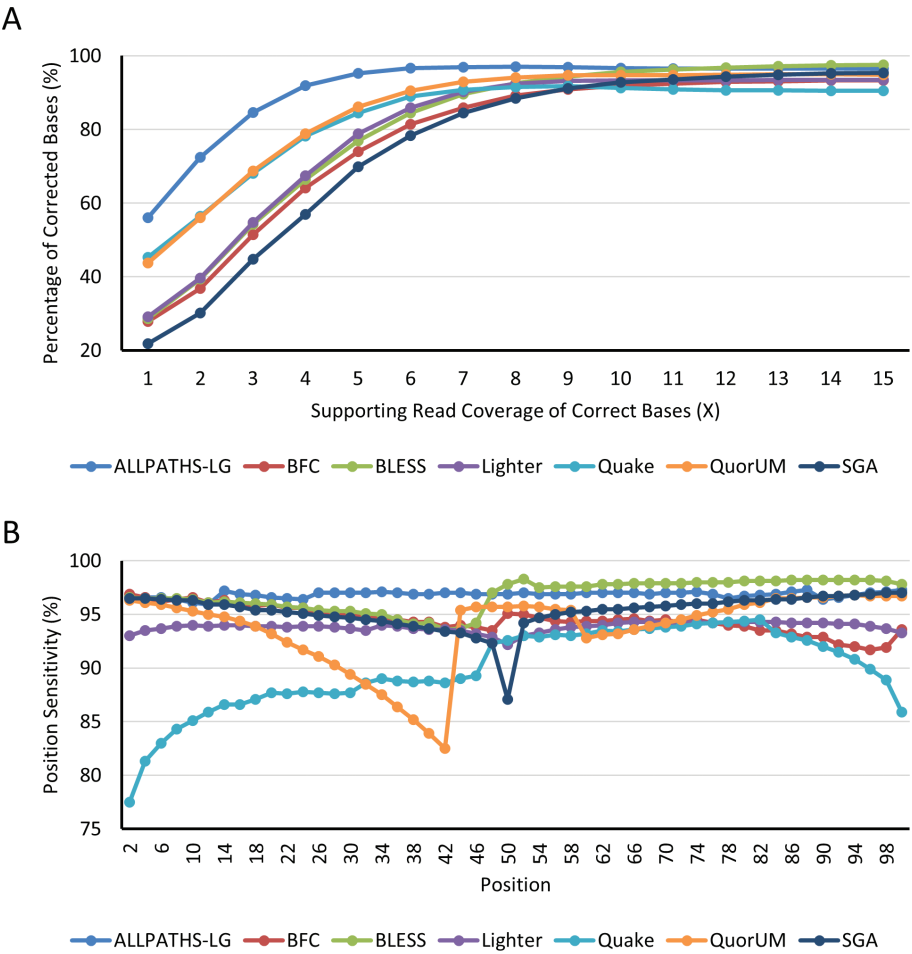

Figure 2. Corrected errors. A. The percentage of corrected errors in I5-40x for various supporting read coverage of correct bases. **B.** Point sensitivity of I5-40X.

with low supporting read coverage, gain for I5-10X (a low-coverage read set) of the tools in Table 1 was not as impressive, because they also generated false positives. The effect of differential supporting read coverage on error correction was significant only when read coverage was low (34).

Figure 2B shows percentage of errors corrected at different locations of reads. ALLPATHS-LG, BFC, BLESS, and Lighter correct errors relatively uniformly across read positions, while the plots for QuorUM and SGA have deep valley points. Also, Quake could only correct a relatively small number of errors at both ends of reads compared to the others. A similar analysis for insertions and deletions is presented in the extended manuscript (34).

Alignment results for Illumina error-correction tools

Reads were aligned using the paired-end alignment feature of Bowtie (47) without allowing any mismatches or indels (Table 2). I1-I5 have two reference sequences and corrected read sets were aligned to the reference sequence from which they originated. Tools that showed high sensitivity also had more reads aligned correctly to the reference sequences. In almost all the cases, the ratio of correctly aligned reads to the total number of aligned reads was over 99 percent except for I4. For I4, only the corrected reads from BLESS, Lighter, and Racer showed the accuracy of over 99 percent.

Accuracy of PacBio error-correction tools

Due to the higher error rates of TGS reads, error correction outputs can have many uncorrected bases. Therefore, most TGS error-correction tools generate two types of reads: (i), trimmed reads that only contain corrected regions in input reads; and (ii), untrimmed reads that include both corrected and uncorrected regions in input reads.

For PacBio, PBcR only produced trimmed reads, LSC and Proovread generated both trimmed reads and untrimmed reads, and they were assessed separately. For LoRDEC, trimmed reads were generated from the untrimmed reads using lordec-trim-split that is included in the LoRDEC package. For MinION reads, both NanoCorr and NaS produced trimmed reads.

Percent similarity of the input reads was 76.6% before error correction, and all the output results were better than this number (Figure 3A). Tools (except LSC) showed percentage similarity of over 95% for the trimmed reads. For the untrimmed reads, LoRDEC and Proovread generated more accurate reads than LSC. Except for the case of untrimmed LoRDEC reads, read coverage of Illumina reads had almost no impact on percentage similarity.

Figure 3B and Figure 3C show read coverage and NG50 of the outputs of the compared tools. The two charts had similar shapes. Both values were high where percentage similarity in Figure 3A was low. The trimmed LoRDEC reads and the PBcR outputs were improved a lot by increasing Illumina read coverage. The trimmed reads from Proovread were also improved but the values were saturated at 30X coverage.

Percentage similarity, read coverage, and NG50 are compared for P2-40X and P2-40X-EF in Figure 3D-F. Percentage similarity, read coverage, and NG50 of the input PacBio reads were 79.4%, 20X, and 12,095 bp, respectively. Trimmed outputs of Proovread and LoRDEC showed high percentage similarity. Percentage similarity and read coverage were similar for trimmed and untrimmed outputs of Proovread, while trimming reduced NG50. For LoRDEC, trimming eliminated too many bases, consequently significantly degrading read coverage and NG50. Also, it can be seen that error-free Illumina reads did not have meaningful impacts.

PacBio error-correction tools seem to have lower sensitivity and gain (considering substitutions and indels) compared to tools for Illumina (Figures 3G-H). Gain and sensitivity generally improve upon trimming. For example, the sensitivity of trimmed reads of LORDEC, Proovread and LSC are significantly higher than that of untrimmed versions. For LORDEC trimmed reads, though sensitivity increases with higher Illumina coverage, gain remains largely unchanged

TABLE 2 **Alignment results of 40 X Illumina read sets**

	I1-40X		I1-40X		I3-40X		I4-40X		I5-40X		I6	
Software	Aligned	Correct	Aligned	Correct	Aligned	Correct	Aligned	Correct	Aligned	Correct	Aligned	Correct
Original	52.52	100.00	50.86	100.00	51.16	99.99	51.26	99.54	51.12	99.98	81.07	100.00
ALLPATHS-LG	99.07	99.98	99.07	99.97	98.51	99.93	88.76	97.52	96.88	99.91	98.68	99.99
BFC	98.40	**100.00**	98.23	**100.00**	97.41	99.98	89.33	98.14	96.65	**99.98**	98.39	**100.00**
BLESS	99.83	**100.00**	99.85	99.99	**99.23**	99.98	**92.80**	99.08	**98.59**	**99.98**	98.65	**100.00**
Blue	99.64	99.90	99.68	99.92	96.07	99.66	84.35	92.67	93.08	99.73	**98.78**	99.94
Coral	92.13	98.72	92.52	98.52	79.26	97.84	51.26	**99.54**	N/A	N/A	95.96	99.57
ECHO	87.46	99.99	93.33	99.99	88.52	99.94	N/A	N/A	N/A	N/A	94.98	**100.00**
Fiona	98.28	99.96	98.65	99.94	95.28	99.76	70.46	94.31	N/A	N/A	98.17	99.99
HiTEC	98.78	99.99	99.30	99.99	N/A	N/A	N/A	N/A	N/A	N/A	97.83	**100.00**
Lighter	99.30	**100.00**	99.47	**100.00**	98.13	**99.99**	79.71	99.33	96.13	**99.98**	98.22	**100.00**
Musket	99.49	**100.00**	99.50	**100.00**	97.87	99.98	84.32	98.33	93.86	**99.98**	97.79	**100.00**
Quake	99.57	**100.00**	99.58	**100.00**	98.41	**99.99**	88.17	98.76	94.71	**99.98**	95.82	**100.00**
QuorUM	**99.88**	**100.00**	**99.90**	**100.00**	98.78	99.98	86.54	98.74	97.29	**99.98**	98.64	99.99
RACER	98.51	99.96	99.29	99.96	96.40	99.94	74.16	99.24	92.95	99.95	98.36	99.99
Reptile	97.77	99.99	98.25	99.97	92.00	99.86	79.47	97.22	89.65	99.92	96.69	99.99
SGA	99.57	**100.00**	99.60	**100.00**	98.53	**99.99**	86.72	98.87	97.61	**99.98**	97.95	**100.00**

Aligned: the percentage of aligned reads to the total number of reads; Correct: the percentage of reads that were aligned to correct positions to the number of aligned reads; Original: pre-correction results.

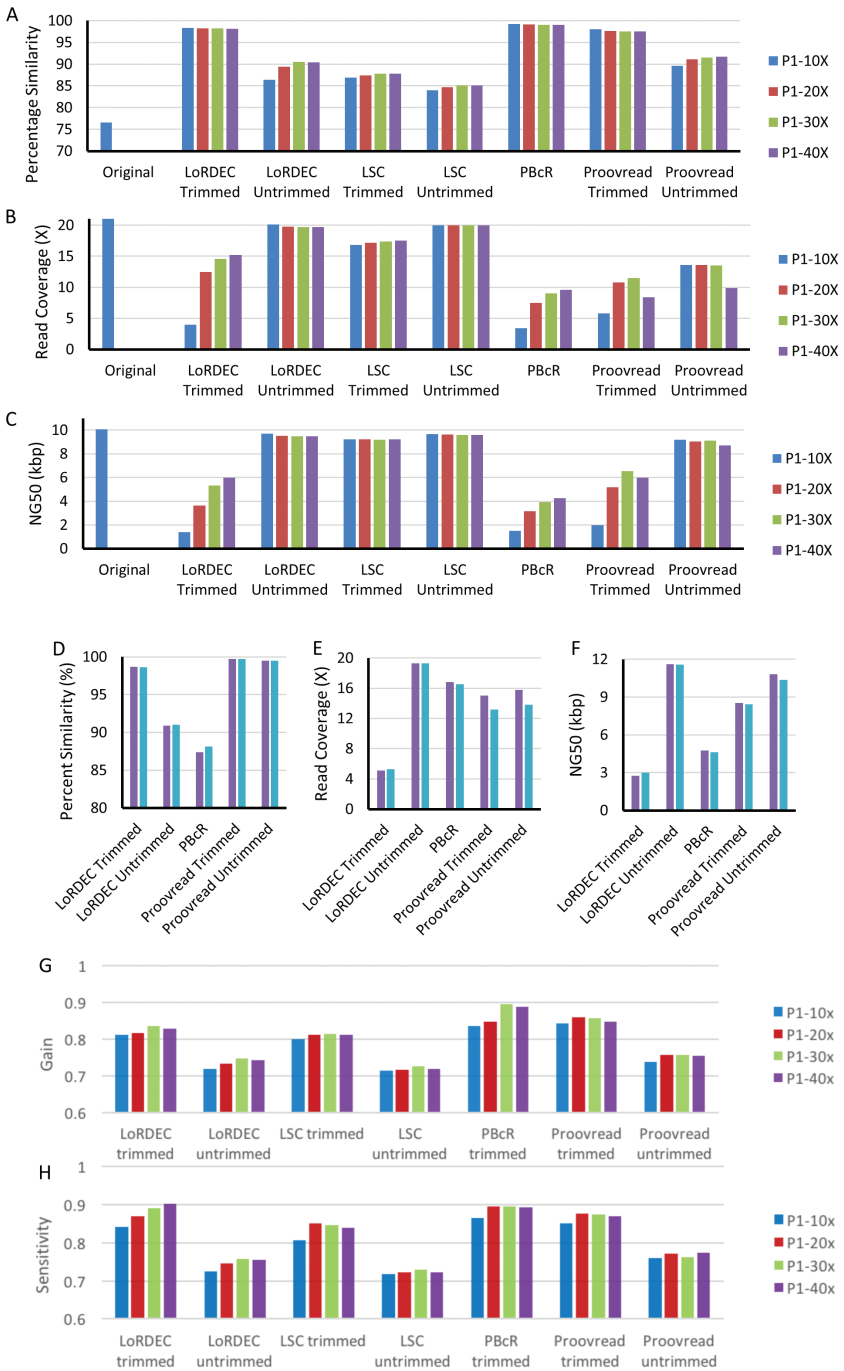

Figure 3. Evaluation results. A-C and **G-H**. Pacbio evaluation results for P1. **D-F.** Results for P2 (lighter shade is 40X-EF).

indicating that at higher Illumina coverage, more errors are corrected, but also there are more false corrections.

Accuracy of ONT read error-correction tools

Figure 4A shows percentage similarity for ONT error-correction tools. For O1, the input read similarity was 57.3%. Both tools significantly improved this number and the values did not significantly change with coverage of the Illumina datasets. Figure 4B shows NG50. Both tool outputs have a lower NG50 length than the input reads and NaS reads have a noticeably lower NG50 length compared to NanoCorr for the O2 dataset. Using error-free Illumina reads did not bring in a noticeable improvement in error correction. Figures 4C and 4D summarize the sensitivity and gain for the two ONT datasets. These results include both indel and substitution errors. It may be noted that NaS presents slightly higher sensitivity and gain compared to NanoCorr.

Software availability

SPECTACLE software and supplementary information available at https://github.com/gowthami19m/SPECTACLE.

CONCLUSION

Among the Illumina error-correction methods that were evaluated, ALLPATHS-LG, BFC, BLESS, Lighter, Quake, QuorUM, and SGA generated accurate results for over 30X read coverage. BLESS and Quake outperformed the others for reads with 10-20X read coverage, and it is expected that ALLPATHS-LG would work best for the reads with under 10X read coverage. For repetitive genomes, BLESS and Quake are recommended.

There was no apparent winner among PacBio tools that could generate both accurate and long reads. While trimming improved error-correction significantly, it reduced the NG50 length and read coverage appreciably. Proovread may be recommended in cases where the accuracy of corrected reads is more important than their length. If a large read set must be corrected in a short time, LoRDEC might be a good choice (34). Tools evaluated for ONT reads provided outputs with good percentage similarity and had comparable gain and sensitivity with respect to the PacBio tools. NanoCorr had longer NG50 length for one of the datasets.

In most cases, we tuned error-correction tool parameters independently and chose the best results. However, in a real situation where the locations of errors are not known in advance, it would not be possible to find the best parameters this way. While in many cases, there are parameter recommendations or defaults, these are encouraged to be made universal.

We believe that SPECTACLE will also be compatible with new sequencing technologies and some of its potential is evident from the fact that it can work with NGS and TGS reads with varied characteristics, providing a

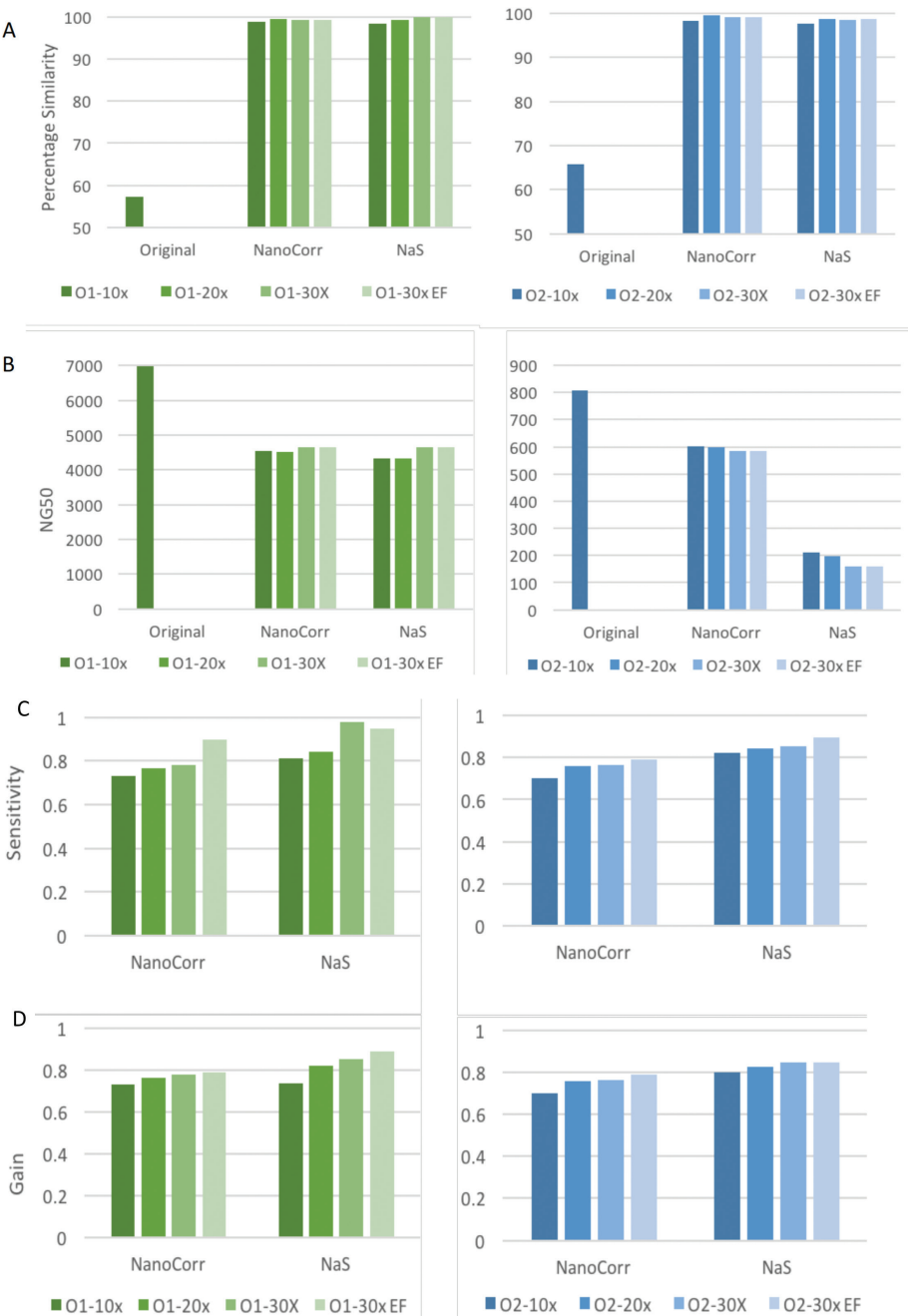

Figure 4. ONT evaluation results A. Percentage Similarity. **B.** Error correction NG50. **C-D.** Sensitivity and Gain.

comprehensive set of evaluation metrics. The fundamental strength of the tool is that the underlying evaluation algorithms are not tied to specific read lengths or error models.

Acknowledgement: The project was funded by the Samsung PhD Fellowship and NSF Grant CNS 1624790. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Conflict of Interest: The authors declare no potential conflicts of interest with respect to research, authorship and/or publication of this chapter.

Copyright and Permission Statement: The authors confirm that the materials included in this chapter do not violate copyright laws. Where relevant, appropriate permissions have been obtained from the original copyright holder(s), and all original sources have been appropriately acknowledged or referenced.

REFERENCES

1. Li H. BFC: correcting Illumina sequencing errors. Bioinformatics. 2015:btv290. https://doi.org/10.1093/bioinformatics/btv290
2. Heo Y, Wu X-L, Chen D, Ma J, Hwu W-M. BLESS: Bloom filter-based error correction solution for high-throughput sequencing reads. Bioinformatics. 2014;30(10):1354–62. https://doi.org/10.1093/bioinformatics/btu030
3. Greenfield P, Duesing K, Papanicolaou A, Bauer DC. Blue: correcting sequencing errors using consensus and context. Bioinformatics. 2014:btu368. https://doi.org/10.1093/bioinformatics/btu368
4. Salmela L, Schröder J. Correcting errors in short reads by multiple alignments. Bioinformatics. 2011;27(11):1455–61. https://doi.org/10.1093/bioinformatics/btr170
5. Kao W-C, Chan A, Song Y. ECHO: A reference-free short-read error correction algorithm. Genome Res. 2011;21(7):1181–92. https://doi.org/10.1101/gr.111351.110
6. Schulz MH, Weese D, Holtgrewe M, Dimitrova V, Niu S, Reinert K, et al. Fiona: a parallel and automatic strategy for read error correction. Bioinformatics. 2014;30(17):i356–63. https://doi.org/10.1093/bioinformatics/btu440
7. Ilie L, Fazayeli F, Ilie S. HiTEC: accurate error correction in high-throughput sequencing data. Bioinformatics. 2011;27(3):295–302. https://doi.org/10.1093/bioinformatics/btq653
8. Song L, Florea L, Langmead B. Lighter: fast and memory-efficient sequencing error correction without counting. Genome Biol. 2014;15(11):509. https://doi.org/10.1186/s13059-014-0509-9
9. Liu Y, Schröder J, Schmidt B. Musket: a multistage k-mer spectrum-based error corrector for Illumina sequence data. Bioinformatics. 2013;29(3):308–15. https://doi.org/10.1093/bioinformatics/bts690
10. Kelley D, Schatz M, Salzberg S. Quake: quality-aware detection and correction of sequencing errors. Genome Biol. 2010;11(11):R116. https://doi.org/10.1186/gb-2010-11-11-r116
11. Marçais G, Yorke JA, Zimin A. QuorUM: An Error Corrector for Illumina Reads. PLoS ONE 2015;10(6):e0130821. https://doi.org/10.1371/journal.pone.0130821
12. Ilie L, Molnar M. RACER: Rapid and accurate correction of errors in reads. Bioinformatics. 2013;29(19):2490–3. https://doi.org/10.1093/bioinformatics/btt407
13. Yang X, Dorman K, Aluru S. Reptile: representative tiling for short read error correction. Bioinformatics. 2010;26(20):2526–33. https://doi.org/10.1093/bioinformatics/btq468
14. Lim E-C, Müller J, Hagmann J, Henz SR, Kim S-T, Weigel D. Trowel: a fast and accurate error correction module for Illumina sequencing reads. Bioinformatics. 2014;30(22):3264–5. https://doi.org/10.1093/bioinformatics/btu513

15. Wirawan A, Harris RS, Liu Y, Schmidt B, Schröder J. HECTOR: a parallel multistage homopolymer spectrum based error corrector for 454 sequencing data. BMC bioinformatics. 2014;15(1):131. https://doi.org/10.1186/1471-2105-15-131
16. Schröder J, Schröder H, Puglisi SJ, Sinha R, Schmidt B. SHREC: a short-read error correction method. Bioinformatics. 2009;25(17):2157–63. https://doi.org/10.1093/bioinformatics/btp379
17. Salzberg S, Phillippy A, Zimin A, Puiu D, Magoc T, Koren S, et al. GAGE: A critical evaluation of genome assemblies and assembly algorithms. Genome Res. 2012;22(3):557–67. https://doi.org/10.1101/gr.131383.111
18. MacManes MD, Eisen MB. Improving transcriptome assembly through error correction of high-throughput sequence reads. PeerJ. 2013;1:e113. https://doi.org/10.7717/peerj.113
19. Fujimoto MS, Bodily PM, Okuda N, Clement MJ, Snell Q. Effects of error-correction of heterozygous next-generation sequencing data. BMC Bioinformatics. 2014;15(Suppl 7):S3. https://doi.org/10.1186/1471-2105-15-S7-S3
20. Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. Nat. Rev. Genet. 2009;10(1):57–63. https://doi.org/10.1038/nrg2484
21. Le H-S, Schulz M, McCauley B, Hinman V, Bar-Joseph Z. Probabilistic error correction for RNA sequencing. Nucleic Acids Res. 2013;41(10):e109-e. https://doi.org/10.1093/nar/gkt215
22. Quail M, Smith M, Coupland P, Otto T, Harris S, Connor T, et al. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics. 2012;13(1):341. https://doi.org/10.1186/1471-2164-13-341
23. Laver T, Harrison J, O'Neill PA, Moore K, Farbos A, Paszkiewicz K, et al. Assessing the performance of the Oxford Nanopore Technologies MinION. Biomol Detect Quantif. 2015;3:1–8. https://doi.org/10.1016/j.bdq.2015.02.001
24. Salmela L, Rivals E. LoRDEC: accurate and efficient long read error correction. Bioinformatics. 2014:btu538. https://doi.org/10.1093/bioinformatics/btu538
25. Au KF, Underwood JG, Lee L, Wong WH. Improving PacBio long read accuracy by short read alignment. PLoS One. 2012;7(10):e46679. https://doi.org/10.1371/journal.pone.0046679
26. Koren S, Schatz M, Walenz B, Martin J, Howard J, Ganapathy G, et al. Hybrid error correction and de novo assembly of single-molecule sequencing reads. Nature Biotechnol. 2012;30(7):693–700. https://doi.org/10.1038/nbt.2280
27. Hackl T, Hedrich R, Schultz J, Förster F. proovread: large-scale high-accuracy PacBio correction through iterative short read consensus. Bioinformatics. 2014;30(21):3004–11. https://doi.org/10.1093/bioinformatics/btu392
28. Madoui MA, Engelen S, Cruaud C, Belser C, Bertrand L, Alberti A, et al. Genome assembly using Nanopore-guided long and error-free DNA reads. Bmc Genomics. 2015;16. https://doi.org/10.1186/s12864-015-1519-z
29. Goodwin S, Gurtowski J, Ethe-Sayers S, Deshpande P, Schatz MC, McCombie WR. Oxford Nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. Genome Res. 2015;25(11):1750–6. https://doi.org/10.1101/gr.191395.115
30. Yang X, Chockalingam SP, Aluru S. A survey of error-correction methods for next-generation sequencing. Brief Bioinform. 2013;14(1):56–66. https://doi.org/10.1093/bib/bbs015
31. Luo C, Tsementzi D, Kyrpides N, Read T, Konstantinidis KT. Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. PloS One. 2012;7(2):e30087. https://doi.org/10.1371/journal.pone.0030087
32. Bragg L, Stone G, Butler M, Hugenholtz P, Tyson G. Shining a Light on Dark Sequencing: Characterising Errors in Ion Torrent PGM Data. PLoS Comput Biol. 2013;9(4):e1003031. https://doi.org/10.1371/journal.pcbi.1003031
33. Molnar M, Ilie L. Correcting Illumina data. Brief Bioinform. 2014. https://doi.org/10.1093/bib/bbu029
34. Heo Y, Manikandan G, Ramachandran A, Chen D. Comprehensive assessment of error correction methods for high-throughput sequencing data. arXiv 2020; arXiv:2007.05121 [q-bio.GN] https://arxiv.org/abs/2007.05121v1
35. Hu X, Yuan J, Shi Y, Lu J, Liu B, Li Z, et al. pIRS: Profile-based Illumina pair-end reads simulator. Bioinformatics. 2012;28(11):1533–5. https://doi.org/10.1093/bioinformatics/bts187

36. Medvedev P, Scott E, Kakaradov B, Pevzner P. Error correction of high-throughput sequencing datasets with non-uniform coverage. Bioinformatics. 2011;27(13):i137-i41. https://doi.org/10.1093/bioinformatics/btr208
37. Ono Y, Asai K, Hamada M. PBSIM: PacBio reads simulator-toward accurate genome assembly. Bioinformatics. 2013;29(1):119–21. https://doi.org/10.1093/bioinformatics/bts649
38. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics. 2009;25(14):1754–60. https://doi.org/10.1093/bioinformatics/btp324
39. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078–9. https://doi.org/10.1093/bioinformatics/btp352
40. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, et al. The variant call format and VCFtools. Bioinformatics. 2011;27(15):2156–8. https://doi.org/10.1093/bioinformatics/btr330
41. Gotoh O. An improved algorithm for matching biological sequences. J. Mol. Biol.1982;162(3):705–8. https://doi.org/10.1016/0022-2836(82)90398-9
42. Haubold B, Wiehe T. How repetitive are genomes? BMC Bioinformatics. 2006;7(1):541. https://doi.org/10.1186/1471-2105-7-541
43. Gnerre S, MacCallum I, Przybylski D, Ribeiro F, Burton J, Walker B, et al. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. PNAS. 2011;108(4):1513–8. https://doi.org/10.1073/pnas.1017351108
44. Simpson J, Durbin R. Efficient de novo assembly of large genomes using compressed data structures. Genome Res. 2011;22(3):gr.126953.111–556. https://doi.org/10.1101/gr.126953.111
45. Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. GigaScience. 2012;1(1):18. https://doi.org/10.1186/2047-217X-1-18
46. Emde A-K, Schulz M, Weese D, Sun R, Vingron M, Kalscheuer V, et al. Detecting genomic indel variants with exact breakpoints in single- and paired-end sequencing data using SplazerS. Bioinformatics. 2012;28(5):619–27. https://doi.org/10.1093/bioinformatics/bts019
47. Langmead B, Trapnell C, Pop M, Salzberg S. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10(3):R25–10. https://doi.org/10.1186/gb-2009-10-3-r25