COMPREHENSIVE EVALUATION OF ERROR CORRECTION METHODS FOR HIGH-THROUGHPUT SEQUENCING DATA

BY

GOWTHAMI JAYASHRI MANIKANDAN

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Deming Chen
Abstract

The advent of DNA and RNA sequencing has significantly revolutionized the study of genomics and molecular biology. Development of high-throughput sequencing technologies have brought about a quick and cheaper way to sequence genomes. Different technologies use different underlying methods for sequencing and are prone to different error rates. Though many tools exist for error correction in high-throughput sequencing data, no standard technology-independent method is available yet to evaluate the accuracy and effectiveness of these error correction tools. In order to supply a standard way to evaluate error correction methods for DNA and RNA sequencing, this thesis presents a Software Package for Error Correction Tool Assessment on nuCLEic acid sequences (SPECTACLE). SPECTACLE can evaluate corrected DNA and RNA reads from many underlying sequencing technologies and differentiate heterozygous alleles from sequencing errors. The work provides some key insights on many factors that stress the challenges in error correction by compiling high-throughput sequencing read sets from technologies like Illumina, PacBio and ONT. The performances of 23 different error correction tools have been analyzed using SPECTACLE and the compiled datasets. This thesis also provides unique and helpful insights into the strengths and weaknesses of various error correction tools and aims to establish a standard platform for evaluating error correction tools in the future.
Acknowledgments

I would first like to thank my thesis advisor Prof. Deming Chen from the ECE Department of the University of Illinois at Urbana-Champaign. He provided me with valuable feedback and suggestions regarding research and writing. He gave me an opportunity to work on this project and steered me in the right direction whenever I had any questions or difficulties.

I would also like to thank my colleagues Dr. Yun Heo and Anand Ramachandran for establishing the initial platform for SPECTACLE and for being actively involved in the validation survey for this project. This project was a collaborative effort and the initial platform and flow for SPECTACLE was developed by Dr. Yun Heo. This thesis work involves updates to SPECTACLE’s underlying algorithm for evaluation of long read sequences. This work also provides added support for evaluating Oxford Nanopore reads using SPECTACLE. I would like to acknowledge their contribution to this work in terms of figures used for illustration and other algorithmic features. Without their participation and input, the work would not have been comprehensive and effective.

Finally, I must express my gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not be possible without them.
Contents

1. Introduction ........................................................................................................................................... 1
2. Literature Review and Summary of Thesis Contributions ................................................................. 3
3. Implementation .................................................................................................................................... 6
   3.1 Preparing Input Data ......................................................................................................................... 6
   3.2 Evaluating Accuracy of Corrected Reads ......................................................................................... 8
   3.3 Calculation of Metrics Evaluated .................................................................................................... 13
4. Results .................................................................................................................................................. 17
   4.1 Data Preparation ............................................................................................................................ 17
      4.1.1 Preparing Illumina Read Sets .................................................................................................... 17
      4.1.2 Preparing PacBio Read Sets .................................................................................................... 18
      4.1.3 Preparing ONT Read Sets ........................................................................................................ 20
   4.2 Running Error Correction Tools .................................................................................................... 21
      4.2.1 Illumina Tools ............................................................................................................................ 21
      4.2.2 TGS (PacBio and ONT) Tools .................................................................................................... 22
   4.3 Evaluation Results for Illumina Error Correction Tools ................................................................. 22
      4.3.1 Accuracy of Illumina Error Correction Tools ........................................................................... 22
      4.3.2 Alignment Results for Illumina Error Correction Tools ............................................................ 25
      4.3.3 Effect of Using Different Alignment Tools on the Evaluation of Real Reads ......................... 26
   4.4 Evaluation Results of TGS Error Correction Tools ......................................................................... 26
      4.4.1 Accuracy of PacBio Error Correction Tools .............................................................................. 27
      4.4.2 Accuracy of ONT Error Correction Tools .................................................................................. 30
5. Conclusion ............................................................................................................................................. 32
References ................................................................................................................................................. 34
1. Introduction

Rapid improvements in next-generation sequencing (NGS) technologies have allowed us to generate a huge amount of sequencing data at a low cost. However, the quality of the data has not improved at the same pace as the throughput of the NGS technologies. For example, one of the widely used Illumina sequencing machines, HiSeq X Ten, can produce 1.8 tera base pairs (bp) in each run, but only about 75 percent of the bases are guaranteed to have Phred scores of over 30 [1].

Errors in NGS reads degrade the quality of downstream analyses, and correcting errors has been shown to improve the quality of these analyses [2]-[4]. Many standalone methods for correcting errors in DNA reads have been developed [5]-[20]. Besides, some DNA assemblers have their own error correction modules, which can be used as standalone error correction tools [21]-[23].

NGS is also used for transcriptome analysis [24]. RNA sequencing data also has sequencing errors, which makes RNA error correction an important problem to address. Error correction methods for DNA reads may not work well for RNA sequencing data because of non-uniform expression levels and alternative splicing. To solve this problem, Le et al. [25] developed a new error correction tool for RNA sequencing data.

Recently, several third-generation sequencing (TGS) technologies have also been developed. TGS sequences do not require any amplification and are based on single-molecule assembly and alignment [26]. Single-molecule real-time (SMRT) sequencing technology from Pacific Biosciences and Oxford Nanopore (ONT) sequencing are widely used TGS technologies. Even though sequencing systems that use the SMRT sequencing technology can generate reads up to tens of thousands of base pairs long, they have about 11 percent error rate and the errors are evenly distributed in reads [27]. Similarly, ONT’s MinION reads have an error rate of 38.2 percent [28]. Also, the dominant error types of these
technologies are insertions and deletions that are rare in Illumina reads. Due to these characteristics, dedicated error correction methods for PacBio reads [29]-[32] and Oxford Nanopore reads [33]-[36] have been developed.
2. Literature Review and Summary of Thesis Contributions

Despite such a large number of error correction methods, only a few studies have been carried out on the evaluation of the accuracy of these methods. Such scarcity is mainly 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 can be easily done by measuring the Hamming distance between a reference sequence and a corrected read, it is not so simple to evaluate how accurately errors are corrected when insertions and deletions also exist as errors. The evaluation becomes more complex when reads are trimmed during error correction. Aligning a read to the source genome does not always solve this problem since multiple best alignments can exist [10]. Heterozygosity also makes the evaluation hard. In a diploid genome, the same locus in a pair of chromosomes could have different alleles. Therefore, if reads from heterozygous genomes are compared with one reference sequence, one of the chromosome alleles that is different from the reference will be recognized as a sequencing error.

Only a handful of research works have been carried out to quantitatively evaluate how exactly errors in NGS reads have been corrected. Error Correction Evaluation Toolkit (ECET) [37] is an error correction evaluation platform that consists of two software packages, one of which evaluates Illumina reads and the other, 454 or Ion Torrent reads. The reason for having two separate algorithms for dealing with the different technologies is that the dominant error models of 454 and Ion Torrent reads are insertions or deletions in homopolymers while most errors in Illumina reads are substitutions [38], [39].

Another evaluation work by Molnar et al. [40] determines the correctness of reads or k-mers in the outputs from Illumina error correction tools instead of directly checking the correctness of bases. Their method calculates (1) how many error-free reads or k-mers cover each base in a genome and (2) 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.

Another evaluation methodology, compute_gain is a program that is a part of an error correction tool package Fiona [10]. It aligns both a read and its corrected version to a reference sequence, and calculates the difference in edit distance between the two alignments. Ambiguities in alignments are resolved by placing gaps at the leftmost or rightmost possible position.

Even though the three methods opened up ways of evaluating the outputs from error correction methods, all of them have limitations. The software package for Illumina reads in ECET can only work with the tools that explicitly specify the number of bases trimmed from both ends of reads. Even when this information is available, separate programs for each error correction tool are needed to extract the number of trimmed bases, because the tools output the number in different ways.

Even though the software reported in [40] can be applied to the outputs from any Illumina error correction method, it may not be applicable to other sequencing technologies. Since PacBio reads, for example, have a high error rate and the errors are evenly distributed in the reads, it is hard to get error-free k-mers of sufficient length. If short k-mers are used by this tool for the evaluation of PacBio reads, the specificity of the evaluation would be low because it is likely that the same or similar k-mers exist in other parts of the genome sequence as well.

The evaluation results of compute_gain, like that of ECET, could be inaccurate in some cases. Since the alignment scores used in compute_gain were designed to evaluate edit distance, a read could be aligned to a reference sequence in totally different ways before and after error correction, which makes it possible for the evaluation result to be inaccurate.
Individual error correction tools also use several metrics and methodologies to compare their performance with similar error correctors. However, in many cases their evaluation metrics are not comprehensive and could be skewed. For example, most TGS error correction tools do not report point sensitivity or gain of the error correction flow. They evaluate their performance based on improvements in assembly and alignment results. Various tools use different underlying assembly or alignment platforms and metrics which makes it difficult to compare and interpret the results.

Addressing the limitations regarding the lack of a universal and comprehensive error correction evaluation flows, a Software Package for Error Correction Tool Assessment on nuCLEic acid sequences (SPECTACLE) has been developed, and error correction methods for Illumina, SMRT and ONT reads, which are the most popular NGS and TGS technologies, have been evaluated. The key contributions of this thesis can be summarized as follows:

(1) Development of a new error correction tool evaluation algorithm that is independent of underlying error models. The algorithm can comprehensively and quantitatively evaluate any error correction tool for NGS and TGS reads.

(2) Design of ONT (MinION) read sets that highlight the challenges in error correction such as heterozygosity, coverage variation, and repeats. These reads can be used as standard inputs for the evaluation of error correction tools.

(3) Comparison of 23 state-of-the-art error correction tools for NGS and TGS reads. The tool reports many statistics pertaining to error correction like sensitivity, gain, precision, F1 score, percentage similarity of reads, NG50 length, supporting read coverage, alignment quality of corrected reads, performance of the tool with respect to read position etc. This will give the users systematic evaluations of strengths and weaknesses of the tools and indicate potential ways for their further improvement.
3. Implementation

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 step this information is used to evaluate the output of an error correction tool. The two steps will be explained in detail in Section 3.1.

3.1 Preparing Input Data

SPECTACLE supports using both simulated reads and real reads to utilize their unique strengths. With simulated reads, the exact locations of errors in the reads can be determined. Moreover, reads can be generated from multiple reference sequences with some differences in order to check whether an error correction tool is able to differentiate heterozygosity from sequencing errors. However, if a read simulator cannot exactly model real reads, using such reads could produce misleading results.

The biggest advantage of using real reads is that no assumptions or modeling artifacts exist behind the sequencing 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. In order to find the error locations in real reads, the reads need to be aligned to a reference sequence, and this can cause some problems. First, it is possible that a read can be aligned to multiple similar locations in a reference sequence (or to the same location in different ways), and
determining the correct alignment is sometimes impossible. In the case of highly repetitive genomes, ambiguous alignments occur frequently, raising the chances of inaccurate evaluation results. Second, reads and a reference sequence might come from different samples, and the differences between them (i.e., variants) will also be recognized as errors. Third, the evaluation results depend on the accuracy of the alignment tool.

Even though SPECTACLE can work with the output reads from any read simulator that gives error location information in a Sequence Alignment/Map (SAM) format, pIRS [41] was used for generating simulated Illumina reads. Error correction becomes challenging when there are heterozygosity and read coverage variations [4], [42], and pIRS can produce reads that stress these characteristics. First, pIRS can generate reads using a diploid genome, and consequently the reads have both sequencing errors and heterozygosity. Second, pIRS can change read coverage depth of a specific genomic region according to the GC-content of the region.

Figure 1A depicts the evaluation flow for simulated reads. First, two reference sequences Ref1 and Ref2 that represent a pair of chromosomes in a diploid genome are generated by adding different variant sets to the input reference sequence Ref0. Once the two sequences are created, reads are generated from Ref1 and Ref2. 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 (1) the positions where reads originate in the genome, (2) the locations of substitutions, insertions, and deletions in each read, and (3) reference sequences from which each read was sampled (i.e. Ref1 or Ref2). When pIRS generates reads, it also produces a file containing the error locations (i.e. .info file) and .info file is converted into $F_L$.

In order to simulate PacBio reads, PBSIM [43] was used. PBSIM generates a Mutation Annotation Format (MAF) file for indicating error locations, and the file is converted to $F_L$. Because these TGS
technologies do not use amplification that causes higher error rates in regions in the genome that have higher G and C base content, the coverage variation due to different GC-content values was not considered in generating the simulated reads for PacBio. These TGS reads are generated from a single reference sequence because their error rate is much higher than the frequency of heterozygous sites and the evaluation results are not expected to be altered appreciably by adding heterozygous points [39]. However, real datasets that have heterozygous sites have been used for our evaluation in order to capture the behavior of error correction tools at heterozygous sites.

Figure 1B shows the evaluation flow for real reads. If input reads and a reference sequence Ref0 do not come from the same sample, there can be variants between them; the variants would be recognized later in the flow as sequencing errors. To overcome this problem, a new reference sequence, Ref1, is generated by calling the variants and applying them to Ref0. For SPECTACLE evaluations, BWA [44] and SAMtools [45] were used for variant calling. The variants are added to Ref0 using VCFtools [46], the input reads are aligned to Ref1, and the alignment results in the SAM file are converted to FL. Among the substitution errors in FL, the errors generated by heterozygous alleles are removed by comparing FL with the variant calling result.

3.2 Evaluating Accuracy of Corrected Reads

Let RC be the corrected version of a read R. In order to evaluate the accuracy of RC, the number of corrected errors and newly added errors in RC need to be determined. SPECTACLE first takes the segment GR from a reference sequence where read R was sampled. The length of GR, LGR, can be calculated as LGR = LR + <Number of Deletions in R>. The number of deletions in R and the start index of GR can be found in FL. In order to get the index where R really ends, LGR should be calculated as LGR = LR + <Number of Deletions in R> - <Number of Insertions in R> (RC1 in Figure 2). However, more bases in the reference sequences are needed if an error correction tool corrects the insertions in R and the tool tries
to keep the length of corrected reads the same. $B_{R:L}$ and $B_{R:R}$, extra bases beside $G_R$ in the reference sequence (Figure 2), should also be recorded because (1) correcting insertions in $R$ can elongate it either to the left or to the right ($RC_2$ and $RC_3$ in Figure 2), (2) error correction tools may introduce deletions, and (3) some error correction tools can make $RC$ longer than $R$ by introducing insertions. The default length of $B_{R:L}$ and $B_{R:R}$ is 10.

![Figure 2 An example of taking part of a reference sequence that is compared with a corrected read and aligning reads into it](image)

Then, $R_C$ is aligned to $G_R$ to find the errors in $R_C$ (errors missed, or introduced by a tool). A modified version of the Gotoh algorithm [47] is used for handling trimmed bases and for extracting all the alignment with the best alignment score.

There can be a set of alignments $ALN_{BEST}$ having the same highest alignment score for a read $R_C$, but each alignment would imply different numbers of corrected and newly introduced errors. In this case, SPECTACLE calculates the penalty of the newly introduced errors in $R_C$ of each alignment utilizing the scores used in the alignment step. Then, the alignment $aln_{BEST}$ from $ALN_{BEST}$ that has the least penalty is chosen. This way all possible best alignments are evaluated to produce an evaluation result that is not biased towards alignment methods and scoring schemes. 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 $ERR(aln)\setminus ERR(R)$ is the set of errors in $aln$ but not in $R$. 
After \( \text{aln}_{\text{BEST}} \) is chosen, it can be determined as to which errors in \( \text{ERR}(R) \) are corrected and how many errors are newly added during correction.

However, it takes a long time to apply the above algorithm to a large number of TGS reads. Due to their long lengths and high repetition and error rates, the above enumeration step for choosing \( \text{aln}_{\text{BEST}} \) from \( \text{ALN}_{\text{BEST}} \) takes a lot of time and memory, as there might be a large number of best alignments to be enumerated in order to get the alignment with the least penalty. In order to evaluate long TGS reads that have high error rates in a reasonable amount of time, SPECTACLE uses a less complex dynamic programming approach to determine the best alignment score with the least number of new errors and the largest number of corrected errors. The algorithm picks the best alignment between \( R_C \) and \( G_R \) using only one alignment matrix and simplified gains and penalty scores. A penalty of \(-1\) is assigned for gaps and mismatches by default. A gain of \(+1\) is assigned for matches by default. These numbers can be changed based on the scoring scheme required for the alignment. The algorithm has two major dynamic programming steps.

First, evaluate the minimum edit-distance alignment between \( R_C \) and \( G_R \) that minimizes the number of new errors in \( R_C \) with respect to \( G_R \). This can be evaluated using the dynamic programming algorithm defined in Equation (1).

\[
\text{Edit}[i,j] = \begin{cases} 
0, & i = 0 \text{ and } j = 0 \\
0, & 1 \leq i \leq L_{G_R} \text{ and } j = 0 \\
-j, & i = 0 \text{ and } 1 \leq j \leq L_{R_C} \\
\max \left\{ \begin{array}{c} \text{Edit}[i-1,j-1] + SC(G_R[i], R_C[j]) \\ \text{Edit}[i-1,j] - 1 \\ \text{Edit}[i,j-1] - 1 \end{array} \right\}, & 1 \leq i \leq L_{G_R} \text{ and } 1 \leq j \leq L_{R_C} 
\end{cases}
\]

\[
SC(i,j) = \begin{cases} 
+1, & i = j \\
-1, & i \neq j 
\end{cases}
\]  

(1)
The first column of $Edit[i,j]$ has been initialized to 0 so that the gaps in $G_R$ are not penalized. In order to not penalize the final gaps in $G_R$, trace back of the matrix to obtain the best alignment is done starting from the maximum element in the last column of $Edit[i,j]$. Then, from Equation (2), $PredecessorSet_{Edit}(i,j)$ is computed, which defines the previous point in the alignment matrix through which the best alignment for $R_C$ and $G_R$ passes.

$$
\begin{align*}
PredecessorSet_{Edit}(i,j) & \leftarrow \emptyset \\
PredecessorSet_{Edit}(i,j) & \leftarrow PredecessorSet_{Edit}(i,j) \cup (i-1,j-1), \text{ if } Edit[i-1,j-1] + SC(G_R[i], R_C[j]) = Edit[i,j] \\
PredecessorSet_{Edit}(i,j) & \leftarrow PredecessorSet_{Edit}(i,j) \cup (i-1,j), \text{ if } Edit[i-1,j] - 1 = Edit[i,j] \\
PredecessorSet_{Edit}(i,j) & \leftarrow PredecessorSet_{Edit}(i,j) \cup (i,j-1), \text{ if } Edit[i,j-1] - 1 = Edit[i,j]
\end{align*}
$$

(2)

Using $PredecessorSet_{Edit}(i,j)$, the alignment score that introduces the least number of new errors in the read can be computed from the recursion in Equation (3).

$$
MinNewErrors(i,j) = \min_{(a,b) \in PredecessorSet_{Edit}(i,j)} (MinNewErrors(a,b) + error_{i,j}(a,b))
$$

(3)

where $error_{i,j}(a,b)$ is defined as shown in Equation (4).

$$
error_{i,j}(a,b) = \begin{cases} 
1, \text{ if } a = i-1, b = j-1, & \text{if } (G_R[i] \neq R_C[j]) \text{ and } (G_R[align_{G_R,R}[i]] = R[i] \text{ or } align_{G_R,R}[i] = -) \\
0, \text{ otherwise }
\end{cases}
$$

$$
= \begin{cases} 
1, \text{ if } a = i-1, b = j, & \text{if } align_{R_C,R}[i] \neq - \\
0, \text{ otherwise }
\end{cases}
$$

$$
= \begin{cases} 
1, \text{ if } a = i, b = j-1, & \text{if } align_{R_C,R}[i] \neq - \\
0, \text{ otherwise }
\end{cases}
$$

(4)
Here, $align_{GR,R}$ is obtained from the alignment of the original read $R$ to the reference region $G_R$. It is defined as follows:

$$align_{GR,R}[i] = \begin{cases} j & ; \text{implies base in } R[i] \text{ aligns to reference } G_R[j] \\ - & ; \text{implies base } R[i] \text{ aligns to a gap in } G_R \text{ (insertion in } R) \end{cases}$$ (5)

Similarly,

$$align_{R,GR}[i] = \begin{cases} j & ; \text{implies base in } R[i] \text{ aligns to reference } G_R[j] \\ - & ; \text{implies base } R[i] \text{ aligns to a gap in } G_R \text{ (insertion in } R) \end{cases}$$ (6)

The second step is to find the reads with the maximum number of corrected errors among the corrected reads with the minimum edit distance and minimum number of new errors (computed from Equation (3)). $PredecessorSet_{MinErr}(i,j)$ is computed as follows:

$$PredecessorSet_{MinErr}(i,j) \leftarrow \emptyset$$

$$PredecessorSet_{MinErr}(i,j) \leftarrow PredecessorSet_{MinErr}(i,j) \cup (i-1,j-1)$$

, if $MinNewErrors(i,j) = MinNewErrors(i-1,j-1) + error_{i,j}(i-1,j-1)$

$$PredecessorSet_{MinErr}(i,j) \leftarrow PredecessorSet_{MinErr}(i,j) \cup (i-1,j)$$

, if $MinNewErrors(i,j) = MinNewErrors(i-1,j) + error_{i,j}(i-1,j)$

$$PredecessorSet_{MinErr}(i,j) \leftarrow PredecessorSet_{MinErr}(i,j) \cup (i,j-1)$$

, if $MinNewErrors(i,j) = MinNewErrors(i,j-1) + error_{i,j}(i,j-1)$

(7)

The best read alignment with the minimum number of new errors that also has the maximum number of corrected errors can be obtained using the following recursion:

$$MaxCorrectedErrors(i,j) = \max_{(a,b) \in PredecessorSet_{MinErr}(i,j)} \left( MaxCorrectedErrors(a,b) + corrected_{i,j}(a,b) \right)$$ (8)

where,

$$corrected_{i,j}(a,b) = \begin{cases} 1, & \text{if } a = i-1, b = j-1, \\ 0, & \text{otherwise} \end{cases}$$

$$corrected_{i,j}(a,b) = \begin{cases} 1, & \text{if } R[i] \neq G_R[align_{GR,R}[i]] \\ 0, & \text{otherwise} \end{cases}$$ (9)
MinNewErrors(\text{L}_{RC}, \text{L}_{GR}) gives the number of new errors and MaxCorrectedErrors(\text{L}_{RC}, \text{L}_{GR}) gives the number of corrected errors in the best alignment. Since the identification of the best alignment with minimum new errors and maximum corrected errors are done using dynamic programming recursions, this algorithm takes only 37.3\% of the time taken by the previous algorithm (that enumerates all the alignments with the highest score to find the best one in terms of the number of errors corrected) to evaluate the Illumina I1 (40x) dataset.

### 3.3 Calculation of Metrics Evaluated

In order to classify the bases in input reads, a notation consisting of a triplet is used, each character of which should be 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 \text{R}, modified by an error correction tool, and error-free in \text{RC}. All the bases should fall into one of the five categories: NNN, NYN, NYY, YNY, and YYN because YYY, YNN, and NNY are logically impossible. Using these triplets, the accuracy metrics that are summarized in Table 1 are calculated.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>( \text{sum}(\text{NYY}) / (\text{sum}(\text{NYY}) + \text{sum}(\text{NYN}) + \text{sum}(\text{NNN})) )</td>
</tr>
<tr>
<td>Gain</td>
<td>( (\text{sum}(\text{NYY}) - \text{sum}(\text{YNN}) - \text{sum}(\text{NYN})) / (\text{sum}(\text{NYY}) + \text{sum}(\text{NYN}) + \text{sum}(\text{NNN})) )</td>
</tr>
<tr>
<td>Specificity</td>
<td>( \text{sum}(\text{YNY}) / (\text{sum}(\text{YNN}) + \text{sum}(\text{YNY}) + \text{sum}(\text{NYN})) )</td>
</tr>
<tr>
<td>Precision</td>
<td>( \text{sum}(\text{NYY}) / (\text{sum}(\text{NYY}) + \text{sum}(\text{YNN}) + \text{sum}(\text{NYN})) )</td>
</tr>
<tr>
<td>F-score</td>
<td>( 2 \times \text{sum}(\text{NYY}) / (\text{sum}(\text{NYY}) + \text{sum}(\text{YNN}) + 2\times\text{sum}(\text{NYN}) + \text{sum}(\text{NNN})) )</td>
</tr>
</tbody>
</table>

SPECTACLE also can calculate and report the percentage similarity of reads for error correction evaluation. Percentage similarity of a read set \( S_R \) is defined using the follow equation, where \( N_{RRM}, N_{RMM} \).
$N_{R_M}$ and $N_{R_D}$ 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:

$$\text{Percentage Similarity} = \sum_{R \in SR} \frac{N_{R_M}}{N_{R_M} + N_{R_M} + N_{R_I} + N_{R_D}}$$

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 $R$. Therefore, SPECTACLE also reports read coverage that indicates how long total read length (after trimming) is and NG50 [2] that shows how long the average read length is.

![Reference and Reads](image)

**Figure 3 Supporting reads and supporting read coverage**

In addition to these metrics, SPECTACLE can report other detailed analyses such as those related to supporting read coverage which help users understand the characteristics of an error correction tool in depth. Figure 3 explains a supporting read, supporting read coverage, and differential supporting read coverage. Supporting reads for base X include a specific position of a reference sequence with a specific base at that position. In the left side of Figure 2, there is a read CGTTA with an erroneous base T, and three other reads with correct base C. In this example, the number of supporting reads (i.e. supporting read coverage) for T at that position of the reference sequence is 1, while the supporting read coverage for C is 3. However, there is another similar sequence in the reference (i.e. a repeat) and the reads sample from that region could also constitute supporting reads for the region on the left side, which makes it harder to correct errors. Differential supporting read coverage for an erroneous base can be
defined as (supporting read coverage for correct base) – (supporting read coverage for erroneous base). An error in a read becomes difficult to correct if the corresponding correct base has low supporting read coverage. This is because most error correction tools recognize bases with low supporting read coverage as errors. Low differential supporting read coverage also makes error correction harder, because then both a correct base and an erroneous base have a similar number of supporting reads. SPECTACLE gives the percentage of corrected bases against supporting read coverage for correct bases, and the percentage of corrected bases against differential supporting read coverage. This metric helps in evaluating how sensitive an error correction tool is to variations in read coverage.

SPECTACLE also collects the percentage of corrected bases in each position of reads (i.e., 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 further.

There are some indirect measurements that provide an idea about how good the corrected reads are in the context of downstream analyses. One of the most intuitive ways to evaluate these 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 positions in a reference sequence. In order to avoid this, SPECTACLE has the capability to compare the aligned locations of reads in a SAM format with \( F \). 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 [40]. SPECTACLE collects the two numbers using the exact alignment result described above.
4. Results

SPECTACLE has been used to evaluate 17 Illumina read error correction tools, 4 PacBio and 2 ONT read error correction methods. 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.

4.1 Data Preparation

4.1.1 Preparing Illumina Read Sets

<table>
<thead>
<tr>
<th>ID</th>
<th>Species</th>
<th>Accession Number</th>
<th>( G_i ) (Mbp)</th>
<th>GC (%)</th>
<th>Length</th>
<th>Cov. (X)</th>
<th>Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1-10X</td>
<td><em>R. sphaeroids</em></td>
<td>NC_007488.1, NC_007489.1, NC_007490.1, NC_007493.1</td>
<td>4.6</td>
<td>68.8</td>
<td>100</td>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>I1-20X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1-30X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1-40X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2-10X</td>
<td><em>B. cereus</em> ATCC 10987</td>
<td>NC_003909.8, NC_005707.1</td>
<td>5.4</td>
<td>35.5</td>
<td>100</td>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>I2-20X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2-30X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2-40X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3-10X</td>
<td><em>O. sativa Chr. 5</em></td>
<td>NC_008398.2</td>
<td>29.9</td>
<td>44.0</td>
<td>100</td>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>I3-20X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3-30X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I3-40X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I4-10X</td>
<td><em>Mouse Chr. Y</em></td>
<td>NC_000087.7</td>
<td>88.1</td>
<td>38.9</td>
<td>100</td>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>I4-20X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I4-30X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I4-40X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5-10X</td>
<td><em>Human Chr. 1</em></td>
<td>NC_00001.11</td>
<td>230.5</td>
<td>41.7</td>
<td>100</td>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>I5-20X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5-30X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5-40X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I6</td>
<td><em>B. cereus</em> ATCC 10987</td>
<td>NC_003909.8, NC_005707.1</td>
<td>5.4</td>
<td>35.5</td>
<td>100</td>
<td>40</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 2 Details of Illumina Read Sets

As explained in Chapter 1, coverage variation, heterozygosity, and repeats complicate error correction, and all the three factors were considered when input reads were prepared for evaluation. The Illumina read sets prepared are described in Table 2. Five different genomes I1-I5 were used to
generate simulated read sets. In Table 2, $G_L$ denotes genome length without Ns (unknown bases); $GC$ Avg. denotes the average of GC content; $GC$ Std. denotes the standard deviation of GC content; $Cov$ denotes read coverage and $Error$ Rate is calculated as \((<\text{total number of substitutions}> + <\text{total number of inserted bases}> + <\text{total number of deleted bases}>) \div <\text{total number of bases in reads}>\).

Even though high coverage read sets are popular, correcting errors in low coverage reads is still important. For example, cancer genome samples could be the mixture of cancer genomes and normal genomes, and the portion of one of the genomes could be very low [48]. Error correction tools for such genomes should have the capability to correct errors in low coverage reads. Therefore, read sets having both high and low coverage values are considered, and the coverage of them is indicated using the postfixes -10X, -20X, -30X, and -40X. Coverage ranges from 10x to 40x have been picked to be consistent with base datasets used in other works reporting and validating error correction methods.

$I_1$, $I_2$, and $I_3$ are $E.\ Coli$ bacterium genomes that have different GC-content values. $I_4$ is the mouse chromosome Y known as a highly repetitive genome [49]. $I_5$ is human chromosome 1, the largest genome sequence used in our experiments.

To evaluate the results for real reads, $I_6$ was downloaded from the Illumina website (http://www.illumina.com/systems/miseq/scientific_data.ilmn). The reads from this dataset have been sequenced from the exact same strain as $I_2$ using the Illumina MiSeq sequencer. Because the coverage of the reads is over 2,500 X, the reads were down-sampled to 40 X.

4.1.2 Preparing PacBio Read Sets

The read sets used for evaluating PacBio error correction tools are shown in Table 3. The PacBio error correction tools evaluated in this study require, in addition to PacBio reads, Illumina reads that are much more accurate than the PacBio reads as most PacBio error correction tools use high-quality Illumina short reads to detect and correct errors. These Illumina reads are described in the "Illumina"
column of Table 3. In order to evaluate the effect of Illumina read coverage on the accuracy of error correction for PacBio reads, four different Illumina read sets with different read coverage values (suffixed -10X, -20X, -30X, and -40X) have been prepared. The 40X-EF is an error-free version of 40X and the read set was used to evaluate the effects of sequencing errors in Illumina reads on error correction for PacBio reads.

Table 3 Details of PacBio Read Sets

<table>
<thead>
<tr>
<th>ID</th>
<th>Reference</th>
<th>PacBio</th>
<th>Illumina</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Species</td>
<td>Accession Number</td>
<td>$G_1$ (Mbp)</td>
</tr>
<tr>
<td>P1-10X</td>
<td>E. coli</td>
<td>NC_000913.3</td>
<td>4.6</td>
</tr>
<tr>
<td>P1-20X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1-30X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1-40X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2-10X</td>
<td>Human Chr19 10 Mbp</td>
<td>NC_000019.10</td>
<td>10.0</td>
</tr>
<tr>
<td>P2-20X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2-30X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2-40X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2-40X-EF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P1 is E. coli K12 M1665 strain, and 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). Four Illumina read sets with different read coverage values were generated by taking a different number of reads 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. SPECTACLE was tried to run with the entire human chromosome 19. However, only LoRDEC could be finished within 70 hours, which is the maximum allocated runtime in our cluster; as a result, only a portion of the chromosome was used. The PacBio reads and the Illumina reads for P2 were simulated using PBSIM and pIRS, respectively.
PacBio reads with lengths shorter than 500 bp were filtered out because short PacBio reads have a lot of errors (due to the high error rate of TGS reads) compared to the corresponding short Illumina reads (that are used for error correction of TGS reads). As a result, the Illumina reads do not have a significant benefit on the correction of such short PacBio reads, so such reads are pruned away to get unbiased evaluation results.

### 4.1.3 Preparing ONT Read Sets

**Table 4 Details of ONT (MinION) Read Sets**

<table>
<thead>
<tr>
<th>ID</th>
<th>Reference</th>
<th>MinION reads</th>
<th>Illumina reads</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Species</td>
<td>Accession Number</td>
<td>G, (Mbp)</td>
</tr>
<tr>
<td>O1-10X</td>
<td><em>E. coli</em></td>
<td>NC_000913.3</td>
<td>4.6</td>
</tr>
<tr>
<td>O1-20X</td>
<td></td>
<td>SRR922409</td>
<td>97</td>
</tr>
<tr>
<td>O1-30X</td>
<td></td>
<td>SRR922409</td>
<td>97</td>
</tr>
<tr>
<td>O1-30x-EF</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>O2-10X</td>
<td><em>Saccharomyces cerevisiae</em></td>
<td>SRP055987</td>
<td>7.5</td>
</tr>
<tr>
<td>O2-20X</td>
<td>W303</td>
<td>SRR567755</td>
<td>250</td>
</tr>
<tr>
<td>O2-30X</td>
<td></td>
<td>SRR567755</td>
<td>250</td>
</tr>
<tr>
<td>O2-30X-EF</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 shows the details of the ONT datasets that have been used for the evaluation of ONT error correction tools. ONT is a relatively newer technology, ONT read simulation and error correction techniques are currently being explored. There are fewer ONT datasets that are publicly available for testing and evaluation. ONT error correction tools also use short Illumina reads of high quality for error correction. The details of the Illumina reads used have also been mentioned in Table 4. Both O1 and O2 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). O2 is *Saccharomyces cerevisiae W303* strain downloaded from the NCBI Sequence Read Archive (http://www.ncbi.nlm.nih.gov/sra). The real Illumina reads for both these datasets were downloaded from Illumina BaseSpace. The error-free
versions of the corresponding Illumina reads (O1-30X-EF and O2-30X-EF) were simulated using pIRS.

Similar to PacBio reads, ONT reads shorter than 500 bp were filtered out.

4.2 Running Error Correction Tools

4.2.1 Illumina Tools

The input read sets were corrected using the 17 error correction tools that had shown good accuracy in the previous evaluations or had been newly published at the time of running the evaluations. Among these, the standalone error correction tools are BFC [5], BLESS [6], Blue [7], Coral [8], ECHO [9], HiTEC [11], Fiona, Lighter [12], Musket [13], Quake [14], QuorUM [15], RACER [16], Reptile [17], and Trowel [18]. The remaining three tools are parts of DNA assemblers, ALLPATHS-LG [21], SGA [22], and SOAPdenovo [23].

For each error correction method, successive numbers were applied to the key parameters of the tools, and multiple corrected output read sets were generated 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 10 X coverage.
4.2.2 TGS (PacBio and ONT) Tools

Widely used PacBio read error correction tools LoRDEC [29], LSC [30], PBcR [31], and Proovread [32] were evaluated using P1 and P2. No parameter tuning was needed for LSC, PBcR, and Proovread. For LoRDEC, multiple output sets were generated by applying successive values for k-mer length and solid k-mer occurrence threshold, and result that gave the highest percentage similarity was chosen. LSC could not be assessed using P2 because it had not finished after 70 hours.

Since ONT is a relatively newer technology, ONT read error correction technologies are just being explored and studied in detail. Two of the most recent ONT read error correction technologies NaS [33] and NanoCorr [34] were evaluated using O1 and O2.

4.3 Evaluation Results for Illumina Error Correction Tools

4.3.1 Accuracy of Illumina Error Correction Tools

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLPATHS-LG</td>
<td>0.998</td>
<td>0.983</td>
<td>0.998</td>
<td>0.984</td>
<td><strong>0.990</strong></td>
<td>0.966</td>
<td>0.851</td>
<td>0.690</td>
<td>0.969</td>
<td>0.904</td>
<td>0.960</td>
<td>0.958</td>
</tr>
<tr>
<td>BFC</td>
<td>0.964</td>
<td>0.964</td>
<td>0.960</td>
<td>0.959</td>
<td>0.948</td>
<td>0.940</td>
<td>0.777</td>
<td>0.711</td>
<td>0.934</td>
<td>0.920</td>
<td>0.981</td>
<td>0.979</td>
</tr>
<tr>
<td>BLESS</td>
<td>0.998</td>
<td>0.997</td>
<td>0.998</td>
<td>0.998</td>
<td><strong>0.990</strong></td>
<td>0.983</td>
<td><strong>0.905</strong></td>
<td>0.855</td>
<td><strong>0.975</strong></td>
<td>0.964</td>
<td>0.979</td>
<td>0.977</td>
</tr>
<tr>
<td>Blue</td>
<td>0.998</td>
<td>0.981</td>
<td>0.998</td>
<td>0.970</td>
<td>0.981</td>
<td>0.883</td>
<td>0.850</td>
<td>0.520</td>
<td>0.896</td>
<td>0.819</td>
<td>0.982</td>
<td>0.903</td>
</tr>
<tr>
<td>Coral</td>
<td>0.979</td>
<td>0.913</td>
<td>0.987</td>
<td>0.934</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.817</td>
<td>0.806</td>
</tr>
<tr>
<td>ECHO</td>
<td>0.831</td>
<td>0.784</td>
<td>0.949</td>
<td>0.900</td>
<td>0.856</td>
<td>0.803</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.831</td>
<td>0.822</td>
</tr>
<tr>
<td>Fiona</td>
<td>0.998</td>
<td>0.973</td>
<td>0.998</td>
<td>0.980</td>
<td>0.984</td>
<td>0.902</td>
<td>0.677</td>
<td>0.237</td>
<td>N/A</td>
<td>N/A</td>
<td>0.970</td>
<td>0.967</td>
</tr>
<tr>
<td>HITEC</td>
<td>0.997</td>
<td>0.982</td>
<td>0.997</td>
<td>0.993</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.965</td>
<td>0.959</td>
</tr>
<tr>
<td>Lighter</td>
<td>0.995</td>
<td>0.992</td>
<td>0.996</td>
<td>0.995</td>
<td>0.974</td>
<td>0.966</td>
<td>0.656</td>
<td>0.586</td>
<td>0.939</td>
<td>0.913</td>
<td>0.973</td>
<td>0.971</td>
</tr>
<tr>
<td>Musket</td>
<td>0.996</td>
<td>0.995</td>
<td>0.996</td>
<td>0.995</td>
<td>0.973</td>
<td>0.964</td>
<td>0.773</td>
<td>0.698</td>
<td>0.909</td>
<td>0.886</td>
<td>0.958</td>
<td>0.955</td>
</tr>
<tr>
<td>Quake</td>
<td>0.988</td>
<td>0.988</td>
<td>0.990</td>
<td>0.990</td>
<td>0.973</td>
<td>0.970</td>
<td>0.856</td>
<td>0.830</td>
<td>0.920</td>
<td>0.913</td>
<td>0.738</td>
<td>0.736</td>
</tr>
<tr>
<td>QuorUM</td>
<td><strong>0.999</strong></td>
<td><strong>0.997</strong></td>
<td><strong>0.999</strong></td>
<td><strong>0.998</strong></td>
<td><strong>0.981</strong></td>
<td><strong>0.969</strong></td>
<td><strong>0.779</strong></td>
<td><strong>0.709</strong></td>
<td><strong>0.951</strong></td>
<td><strong>0.925</strong></td>
<td><strong>0.982</strong></td>
<td><strong>0.977</strong></td>
</tr>
<tr>
<td>RACER</td>
<td>0.996</td>
<td>0.913</td>
<td>0.997</td>
<td>0.968</td>
<td>0.961</td>
<td>0.708</td>
<td>0.587</td>
<td>-0.097</td>
<td>0.902</td>
<td>0.114</td>
<td>0.967</td>
<td>0.946</td>
</tr>
<tr>
<td>Reptile</td>
<td>0.958</td>
<td>0.933</td>
<td>0.968</td>
<td>0.960</td>
<td>0.926</td>
<td>0.824</td>
<td>0.672</td>
<td>0.562</td>
<td>0.878</td>
<td>0.760</td>
<td>0.852</td>
<td>0.831</td>
</tr>
<tr>
<td>SGA</td>
<td>0.996</td>
<td>0.996</td>
<td>0.996</td>
<td>0.996</td>
<td>0.975</td>
<td>0.968</td>
<td>0.738</td>
<td>0.673</td>
<td>0.959</td>
<td>0.939</td>
<td>0.947</td>
<td>0.944</td>
</tr>
<tr>
<td>SOAPec</td>
<td>0.671</td>
<td>0.670</td>
<td>0.664</td>
<td>0.664</td>
<td>0.650</td>
<td>0.648</td>
<td>0.478</td>
<td>0.446</td>
<td>0.624</td>
<td>0.614</td>
<td>0.539</td>
<td>0.538</td>
</tr>
<tr>
<td>Trowel</td>
<td>0.817</td>
<td>0.814</td>
<td>0.836</td>
<td>0.833</td>
<td>0.835</td>
<td>0.818</td>
<td>0.599</td>
<td>0.469</td>
<td>N/A</td>
<td>N/A</td>
<td>0.677</td>
<td>0.675</td>
</tr>
</tbody>
</table>
Sensitivity and gain for substitution errors for the 40 X input read sets are summarized in Table 5, where, the numbers in bold indicate superior tool performances. For all the bacterium genomes 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, BLESS and Quake outperformed the others, and only these two tools obtained gain above 0.8. For I5, 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. In the evaluation using I6, most tools showed similar performance as they did for I2 since both I2 and I6 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.

The difference between sensitivity and gain shows how many false corrections were made by each tool. In general, BFC, BLESS, Quake, SGA, and SOAPec generated fewer false corrections than the others.

Table 6 Sensitivity and Gain of Substitution Errors for I5 Read Sets With Different Coverage Values

<table>
<thead>
<tr>
<th>Software</th>
<th>I5-10X</th>
<th>I5-20X</th>
<th>I5-30X</th>
<th>I5-40X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sensitivity</td>
<td>Gain</td>
<td>Sensitivity</td>
<td>Gain</td>
</tr>
<tr>
<td>ALLPATHS-LG</td>
<td>0.911</td>
<td>0.811</td>
<td><strong>0.964</strong></td>
<td>0.886</td>
</tr>
<tr>
<td>BFC</td>
<td>0.810</td>
<td>0.749</td>
<td>0.919</td>
<td>0.891</td>
</tr>
<tr>
<td>BLESS</td>
<td>0.931</td>
<td><strong>0.898</strong></td>
<td>0.961</td>
<td><strong>0.946</strong></td>
</tr>
<tr>
<td>Blue</td>
<td>0.848</td>
<td>0.690</td>
<td>0.894</td>
<td>0.809</td>
</tr>
<tr>
<td>Fiona</td>
<td><strong>0.942</strong></td>
<td>0.837</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Lighter</td>
<td>N/A</td>
<td>N/A</td>
<td>0.918</td>
<td>0.867</td>
</tr>
<tr>
<td>Musket</td>
<td>0.889</td>
<td>0.860</td>
<td>0.905</td>
<td>0.882</td>
</tr>
<tr>
<td>Quake</td>
<td>0.908</td>
<td>0.896</td>
<td>0.917</td>
<td>0.910</td>
</tr>
<tr>
<td>QuorUM</td>
<td>0.894</td>
<td>0.810</td>
<td>0.952</td>
<td>0.907</td>
</tr>
<tr>
<td>RACER</td>
<td>0.819</td>
<td>-2.287</td>
<td>0.898</td>
<td>-0.164</td>
</tr>
<tr>
<td>Reptile</td>
<td>0.805</td>
<td>0.612</td>
<td>0.869</td>
<td>0.728</td>
</tr>
<tr>
<td>SGA</td>
<td>0.852</td>
<td>0.803</td>
<td>0.941</td>
<td>0.917</td>
</tr>
<tr>
<td>SOAPec</td>
<td>0.585</td>
<td>0.545</td>
<td>0.622</td>
<td>0.609</td>
</tr>
</tbody>
</table>

Table 6 shows the variation in gain with different read coverage values for I5, where, the numbers in bold indicate superior tool performances. Only BLESS, Musket, and Quake generated gain above 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 when read coverage became 30X.

![Figure 4 Percentage of Corrected Errors in I5-40X for Various Supporting Read Coverage of Correct Bases](image)

**Figure 4 Percentage of Corrected Errors in I5-40X for Various Supporting Read Coverage of Correct Bases**

The percentage of corrected bases as a function of supporting read coverage for I5-40X is shown in Figure 4. ALLPATHS-LG, Quake, and QuorUM corrected more errors than the others when supporting read coverage of correct bases was close to 1. Even though ALLPATHS-LG and QuorUM have the capability to correct errors with low supporting read coverage, gain for I5-10X of the tools in Table 6 was
not as impressive as this result. This is because they also generated many false positives for this input set.

As shown in Figure 5, tools can correct different percentages of errors in different locations in reads. The plots for ALLPATHS-LG, BFC, BLESS, and Lighter show relatively flat lines, which means that they corrected almost the same proportion of errors in all the positions in reads. On the other hand, plots for QuorUM and SGA have deep valley points, and the positions of these regions with little correction match with the k-mer length used with these tools for generating the respective outputs. In addition, Quake could only correct a relatively small number of errors at both ends of reads compared to the others.

### 4.3.2 Alignment Results for Illumina Error Correction Tools

#### Table 7 Alignment Statistics of Corrected Illumina Datasets

<table>
<thead>
<tr>
<th>Software</th>
<th>I1-40X</th>
<th>I1-40X</th>
<th>I3-40X</th>
<th>I4-40X</th>
<th>I5-40X</th>
<th>I6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aligned</td>
<td>Correct</td>
<td>Aligned</td>
<td>Correct</td>
<td>Aligned</td>
<td>Correct</td>
</tr>
<tr>
<td>Original</td>
<td>52.52</td>
<td>100.00</td>
<td>50.86</td>
<td>100.00</td>
<td>51.16</td>
<td>99.99</td>
</tr>
<tr>
<td>ALLPATHS-LG</td>
<td>99.07</td>
<td>99.98</td>
<td>99.07</td>
<td>99.97</td>
<td>98.51</td>
<td>99.93</td>
</tr>
<tr>
<td>BFC</td>
<td>98.40</td>
<td>100.00</td>
<td>98.23</td>
<td>100.00</td>
<td>97.41</td>
<td>99.98</td>
</tr>
<tr>
<td>BLESS</td>
<td>99.83</td>
<td>100.00</td>
<td>99.85</td>
<td>99.99</td>
<td>99.23</td>
<td>99.98</td>
</tr>
<tr>
<td>Blue</td>
<td>99.64</td>
<td>99.90</td>
<td>99.68</td>
<td>99.92</td>
<td>96.07</td>
<td>99.66</td>
</tr>
<tr>
<td>Coral</td>
<td>92.13</td>
<td>98.72</td>
<td>92.52</td>
<td>98.52</td>
<td>79.26</td>
<td>97.84</td>
</tr>
<tr>
<td>ECHO</td>
<td>87.46</td>
<td>99.99</td>
<td>93.33</td>
<td>99.99</td>
<td>88.52</td>
<td>99.94</td>
</tr>
<tr>
<td>Fiona</td>
<td>98.28</td>
<td>99.96</td>
<td>98.65</td>
<td>99.94</td>
<td>95.28</td>
<td>99.76</td>
</tr>
<tr>
<td>HiTEC</td>
<td>98.78</td>
<td>99.99</td>
<td>99.30</td>
<td>99.99</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Lighter</td>
<td>99.30</td>
<td>100.00</td>
<td>99.47</td>
<td>100.00</td>
<td>98.13</td>
<td>99.99</td>
</tr>
<tr>
<td>Musket</td>
<td>99.49</td>
<td>100.00</td>
<td>99.50</td>
<td>100.00</td>
<td>97.87</td>
<td>99.98</td>
</tr>
<tr>
<td>Quake</td>
<td>99.57</td>
<td>100.00</td>
<td>99.58</td>
<td>100.00</td>
<td>98.41</td>
<td>99.99</td>
</tr>
<tr>
<td>QuorUM</td>
<td>99.88</td>
<td>100.00</td>
<td>99.90</td>
<td>100.00</td>
<td>98.78</td>
<td>99.98</td>
</tr>
<tr>
<td>RACER</td>
<td>98.51</td>
<td>99.96</td>
<td>99.29</td>
<td>99.96</td>
<td>96.40</td>
<td>99.94</td>
</tr>
<tr>
<td>Reptile</td>
<td>97.77</td>
<td>99.99</td>
<td>98.25</td>
<td>99.97</td>
<td>92.00</td>
<td>99.86</td>
</tr>
<tr>
<td>SGA</td>
<td>99.57</td>
<td>100.00</td>
<td>99.60</td>
<td>100.00</td>
<td>98.53</td>
<td>99.99</td>
</tr>
</tbody>
</table>

Table 7 shows how many corrected reads can be exactly aligned to the reference sequences. In Table 7, *Aligned* indicates the percentage of aligned reads to the total number of reads; *Correct* indicates the ratio of reads that were aligned to correct positions as a percentage of the total number of aligned reads; *Original* indicates pre-correction results. The numbers in bold indicate superior tool
performances. Reads were aligned using the paired-end alignment feature of Bowtie [50] without allowing any mismatches or indels. The genomes I1-I5 have two reference sequences, and corrected read sets were aligned to the reference sequence from which they originated. The alignment results are well matched with the results in Table 5, and the 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 with the exception of I4. For I4, only the corrected reads from BLESS, Lighter, and Racer showed the accuracy of over 99 percent.

4.3.3 Effect of Using Different Alignment Tools on the Evaluation of Real Reads

For real reads, the errors corrected by an error correction tool were compared against mismatches and indels obtained in aligning the reads to a reference sequence. Therefore, the numbers and the locations of errors could vary according to alignment tools. Two $F_I$ files from I6 were generated using BWA [44] and Bowtie 2 [51] with default options, and the two files were compared. While BWA found 473,090 substitution errors in D6, Bowtie 2 found 632,705. About 97 percent of substitutions in the BWA set were also found in the Bowtie 2 set, which means that Bowtie 2 is more aggressive than BWA and that it enables tools to flag more errors in reads.

When the error correction results were evaluated using the $F_I$ file from Bowtie 2, sensitivity and gain dropped by up to 8 percent compared to the results with the $F_I$ file from BWA because some of the new errors found by Bowtie 2 were not corrected in the error correction tools.

4.4 Evaluation Results of TGS Error Correction Tools

Due to the high error rate of TGS reads, error correction outputs could have many uncorrected bases. Therefore, most TGS error correction tools generate two types of reads: (1) trimmed reads that
only contain corrected regions in input reads and (2) untrimmed reads that include both corrected and uncorrected regions in input reads.

For PacBio reads, PBcR only produces trimmed reads, LSC and Proovread generate 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 produce untrimmed reads.

4.4.1 Accuracy of PacBio Error Correction Tools

Figure 6 Percentage Similarity, Read Coverage and NG50 of PacBio Read Error Correction Methods for P1

In Figure 6A, percentage similarity of the outputs from PacBio read error correction methods for P1 are compared. Percent similarity of the input reads was 76.6 percent before error correction, and all the
output results were better than this number. Among the four tools, three tools except LSC showed percent similarity over 95 percent 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 6B and Figure 6C 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 6A 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 30 X coverage.

![Figure 6A, 6B, 6C](image)

*Figure 7 Percentage Similarity, Read Coverage and NG50 of PacBio Read Correction Methods for P2*

Percentage similarity, read coverage, and NG50 are compared for P2-40X and P2-40X-EF which is the error-free version of P2-40X in Figure 7. Percentage similarity, read coverage and NG50 of the input PacBio reads before error correction were 79.4 percent, 20X and 12,095 bp, respectively. Both, trimmed Proovread reads and trimmed LoRDEC reads showed high percentage similarity. Percentage similarity and read coverage of the untrimmed Proovread reads were almost the same compared to those of the trimmed Proovread reads. However, NG50 of trimmed Proovread reads was shorter than that of
untrimmed Proovread reads. LoRDEC generated the trimmed reads with high percent similarity but it removed too many bases and read coverage and NG50 of the read set became much lower than those of the original input reads.

For all these cases, P2-40-EF did not make a meaningful difference when it was compared with P2-40. This means sequencing errors in Illumina reads are not important when Illumina read coverage is about 40 X.

**Figure 8 Sensitivity and Gain of PacBio Read Error Correction Methods for P1**

Figure 8 shows the sensitivity and gain results for the different PacBio error correction tools. Compared to Illumina sequences of the same genome (I1 and P1 reads of *E. Coli*), PacBio error correction tools have lower sensitivity and gain because the error rates of PacBio reads are higher than that of Illumina reads. Untrimmed reads have still lower values of sensitivity and gain as they also return
the uncorrected portions of the reads. Though all the tools are comparable in terms of sensitivity and gain, PBeR and Proovread perform the best.

4.4.2 Accuracy of ONT Error Correction Tools

Figure 9A shows percentage read similarity values for ONT datasets. For O1, the original read similarity was 57.3 percent, which is lower compared to the corresponding PacBio reads (for example, P1 and O1 from the E.Coli genome) because of the higher error rate of ONT reads. Both the error correction tools significantly improved the percentage similarity of reads. And, the values were comparable for different coverage values. Figure 9B shows the NG50 values for O1 and O2 datasets. NaS reads have a slightly lower (better) NG50 length compared to NanoCorr for the O2 dataset. Similar to

Figure 9 Percentage Similarity and NG50 of MinION Read Error Correction Methods for O1 and O2
PacBio, the use of error-free Illumina reads did not show a great improvement in error correction compared to the use of erroneous Illumina reads.

Figure 10 Sensitivity and Gain of MinION Error Correction Methods for O1 and O2

Figure 10 summarizes the sensitivity and gain results for the two ONT datasets. Sensitivity and gain patterns are similar to that of the PacBio reads. NaS performs slightly better out of the two in terms of accuracy metrics. Higher the coverage of the complementary Illumina reads, higher is the gain and sensitivity of the tool, which is intuitive because, higher the number of supporting Illumina reads available to correct errors, the better the tool performs.
5. Conclusion

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

Among the evaluated PacBio error correction tools, there was no apparent winner that could generate both accurate and long reads. Proovread could be recommended in cases where the accuracy of corrected reads is more important than their length. If long read length is more important or a large read set should be corrected in a short time, LoRDEC might be a good choice.

Though some tools have recommendations for choosing input parameters, the parameters were tried to be tuned independently based on the results for fair comparison. 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. Therefore, it is recommended to developers of error correction tools that as many parameters as possible should be automatically determined or clear guidelines for determining them should be given to users.

Several TGS platforms (like MinION from ONT) are just being explored and studied in detail. As of yet, only a few datasets are publicly available for comprehensive testing and evaluation. Read simulation and error correction methodologies for reads from such nascent sequencing platforms are still being explored. It is believed that SPECTACLE will still be compatible to such new methodologies and provide comprehensive evaluation and characterization for new error correction tools to come, as illustrated by the MinION sequence error correction evaluation results presented in this work.
Even though the work presents a comprehensive analysis of the accuracy of most of the state-of-the-art error correction methods for the NGS and TGS technologies, the study can be further extended to evaluate TGS reads from larger genomes using more powerful computational resources. It is expected that repeats in a genome would affect the quality of error correction in TGS reads. However, repeats cause a significant problem only when genome length is sufficiently long.

It is also desirable to study how sequencing errors degrade the quality of downstream analyses. A detailed understanding of the mechanism will yield useful insights into how to correct errors that are detrimental to a specific application and how to make applications less sensitive to sequencing errors, and SPECTACLE can help in categorizing such errors.
References


