RACER: Rapid and accurate correction of errors in reads

Lucian Ilie* and Michael Molnar
Department of Computer Science, University of Western Ontario, N6A 5B7 London, ON, Canada

ABSTRACT
Motivation: High-throughput next-generation sequencing technologies enable increasingly fast and affordable sequencing of genomes and transcriptomes, with a broad range of applications. The quality of the sequencing data is crucial for all applications. A significant portion of the data produced contains errors, and ever more efficient error correction programs are needed.

Results: We propose RACER (Rapid and Accurate Correction of Errors in Reads), a new software program for correcting errors in sequencing data. RACER has better error-correcting performance than existing programs, is faster and requires less memory. To support our claims, we performed extensive comparison with the existing leading programs on a variety of real datasets.

Availability: RACER is freely available for non-commercial use at www.csd.uwo.ca/~ilie/RACER/.

Contact: ilie@csd.uwo.ca

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on February 20, 2013; revised on April 19, 2013; accepted on July 8, 2013

1 INTRODUCTION
The automated Sanger sequencing method (Sanger et al., 1977) has revolutionized biological research by unveiling the sequence of the DNA molecule, most prominently that of the human genome. Limitations of the method created the need for improved sequencing technologies. Great demand caused the discovery of several so-called next-generation sequencing (NGS) technologies, such as Illumina/Solexa, Roche/454, Life/PG’s SOLiD, Helicos BioSciences’ HeliScope, Pacific Biosciences and Life’s Ion Torrent; the survey of Metzker (2010) gives detailed descriptions.

These high-throughput technologies produce huge amounts of data at decreasing costs, thus enabling an ever increasing number of applications, including de novo genome assembly, genome resequencing, cancer mutation discovery, metagenomics, DNA–protein interaction discovery and so forth. The way has been opened for ambitious projects, such as the 1000 Genomes Project (Siva, 2008), the first project to sequence the genomes of a large number of people to provide a comprehensive resource on human genetic variation, and the Genome 10K Project (Haussler et al., 2009), aiming at discovering the genomes of 10000 vertebrate species.

A large number of bioinformatics programs are essential in analyzing the NGS data. Although their variety is impressive, they all need high-quality data. Nearly half of the reads produced by Illumina, the currently dominant technology, contain errors. Among other problems, they make genome assembly much more difficult, and genome assemblers, such as Euler-SR (Chaisson and Pevzner, 2008), ALLPATHS (Butler et al., 2008), ABYSS (Simpson et al., 2009), SOPAdenovo (Li et al., 2010) or SGA (Simpson and Durbin, 2012), include their own correcting mechanisms.

The importance of the problem led to the design of many stand-alone error-correcting programs, such as Euler (Chaisson et al., 2004), SHREC (Schröder et al., 2009), Reptile (Yang et al., 2010), Quake (Kelley et al., 2010), CUDA (Shi et al., 2010), HSHREC (Salmela, 2010), SOAP (Li et al., 2010), HiTEC (Ilie et al., 2011), Coral (Salmela and Schröder, 2011), Hammer (Medvedev et al., 2011), ECHO (Kao et al., 2011), PSAEC (Zhao et al., 2011b) and MyHybrid (Zhao et al., 2011a); see also the survey of Yang et al. (2013).

The errors present in sequencing data consist of substitutions, indels, and try to correct the entire read with minimum edit distance such that all k-mers have counts above a threshold; CUDA, Quake, Reptile and Hammer are included here. Finally, Coral, ECHO and MyHybrid are based on multiple sequence alignments. The reader is referred to the survey of Yang et al. (2013) for details.

We propose a new program, RACER (Rapid and Accurate Correction of Errors in Reads), that belongs to the first category. Although SHREC and HSHREC use suffix trees and HiTEC uses the more efficient suffix arrays, RACER uses completely different, more efficient data structures. This way, whereas both SHREC and, to a lesser extent, HiTEC, have high space
Table 1. The datasets used for evaluation, sorted increasingly by total number of base pairs

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accession number</th>
<th>Reference genome</th>
<th>Genome length</th>
<th>Read length</th>
<th>Number of reads</th>
<th>Number of base pairs</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lacococcus lactis</td>
<td>SRX088759</td>
<td>NC_013656.1</td>
<td>2598144</td>
<td>36</td>
<td>4370050</td>
<td>157321800</td>
<td>60.55</td>
</tr>
<tr>
<td>Treponema pallidum</td>
<td>SRR361468</td>
<td>CP002376.1</td>
<td>1139417</td>
<td>35</td>
<td>7133663</td>
<td>249678205</td>
<td>219.13</td>
</tr>
<tr>
<td>E.coli 75a</td>
<td>SRR365636</td>
<td>NC_000913.2</td>
<td>4639675</td>
<td>75</td>
<td>3454048</td>
<td>259053600</td>
<td>55.83</td>
</tr>
<tr>
<td>Bacillus subtilis</td>
<td>DRR000852</td>
<td>NC_000964.3</td>
<td>4215606</td>
<td>75</td>
<td>3519504</td>
<td>263962800</td>
<td>62.62</td>
</tr>
<tr>
<td>E.coli 75b</td>
<td>SRR365532</td>
<td>NC_000913.2</td>
<td>4639675</td>
<td>75</td>
<td>4341061</td>
<td>325579575</td>
<td>70.17</td>
</tr>
<tr>
<td>Pseudomonas aeruginosa</td>
<td>SRR396641</td>
<td>NC_002516.2</td>
<td>6264404</td>
<td>36</td>
<td>9306557</td>
<td>335036052</td>
<td>53.48</td>
</tr>
<tr>
<td>E.coli 47</td>
<td>SRX022918</td>
<td>NC_000913.1</td>
<td>4771872</td>
<td>47</td>
<td>14408630</td>
<td>677205610</td>
<td>141.92</td>
</tr>
<tr>
<td>Leptospira interrogans L</td>
<td>SRR355563</td>
<td>NC_004342.2</td>
<td>4338762</td>
<td>100</td>
<td>7066162</td>
<td>706616200</td>
<td>162.86</td>
</tr>
<tr>
<td>Leptospira interrogans C</td>
<td>SRR397962</td>
<td>NC_005823.1</td>
<td>4277185</td>
<td>100</td>
<td>7127250</td>
<td>712725000</td>
<td>166.63</td>
</tr>
<tr>
<td>E.coli 36</td>
<td>SRX000429</td>
<td>NC_000913.1</td>
<td>4771872</td>
<td>36</td>
<td>20816448</td>
<td>749392128</td>
<td>157.04</td>
</tr>
<tr>
<td>Haemophilus influenzae</td>
<td>SRX065202</td>
<td>NC_000907.1</td>
<td>1830138</td>
<td>42</td>
<td>23935272</td>
<td>1005281424</td>
<td>549.29</td>
</tr>
<tr>
<td>S.aureus</td>
<td>SRX022866</td>
<td>NC_003923.1</td>
<td>2901156</td>
<td>76</td>
<td>25551716</td>
<td>1941930416</td>
<td>669.36</td>
</tr>
<tr>
<td>S.cerevisiae</td>
<td>SRX100885</td>
<td>PRJNA128</td>
<td>12416363</td>
<td>76</td>
<td>52061664</td>
<td>395686464</td>
<td>318.67</td>
</tr>
<tr>
<td>C.elegans</td>
<td>SRR065390</td>
<td>wormbase.org</td>
<td>102291899</td>
<td>100</td>
<td>67617092</td>
<td>676170920</td>
<td>66.10</td>
</tr>
<tr>
<td>D.melanogaster</td>
<td>SRX006151</td>
<td>flybase.org</td>
<td>120220296</td>
<td>457595</td>
<td>10548652</td>
<td>6903898304</td>
<td>57.43</td>
</tr>
</tbody>
</table>

Note: All datasets and reference genome sequences are obtained from National Center for Biotechnology Information except C.elegans from www.wormbase.org and D.melanogaster from flybase.org.

consumption, RACER is space efficient. It is also faster and more effective at correcting errors than the existing programs. We have performed extensive comparison with the leading programs on a variety of real datasets. RACER is freely available for non-commercial use at www.csd.uwo.ca/~iile/RACER/.

2 RESULTS

2.1 Datasets

We have performed extensive testing on a wide variety of datasets with difference read length, genome size and coverage. Comparison has been performed exclusively on real datasets because, in our experience, simulated datasets do not offer a good indication of real-life performance. In addition, they are easy to correct and the correcting programs often exhibit unrealistically high performance.

Fifteen real datasets were considered, some of which have been previously used either for correcting purposes or assembly: Escherichia coli 36, E.coli 47 (the numbers following the name represent read length and are used to distinguish them), Staphylococcus aureus, Saccharomyces cerevisiae and Drosophila melanogaster have been used in the survey of Yang et al. (2013), and Caenorhabditis elegans was used for comparing SGA with other genome assemblers by Simpson and Durbin (2012). The remaining datasets are new. The accession numbers and details of the datasets, together with the corresponding information concerning the reference genomes, are given in Table 1. Full organism names are provided in Supplementary Table S4.

2.2 Evaluation

We have compared RACER with the programs tested by Yang et al. (2013), namely, HiTEC, SHREC, Reptile, Quake, Coral, SOAP and ECHO. (We have tested SHREC instead of HSHREC, as we consider only substitution errors.) Of these, SOAP and ECHO were unable to run most of the datasets because of high space or time requirements and are left out of the comparison.

All programs tested were evaluated on the raw datasets in Table 1 for their ability to correct errors as well as the time and space required. Error-correcting performance was evaluated as the percentage of errors corrected. Our evaluation method has the advantage of avoiding the interference of mapping or assembl- ing programs and is described in detail in Section 3.2 of Methods.

To have a uniform comparison, we have normalized the time and space values by dividing them by the total number of base pairs in the dataset. The actual time and space values are given in the Supplementary Material.

We have run all programs on the same Hewlett-Packard computer with 24 cores AMD Opteron at 2.1 GHz and 98 GB of random access memory running Linux Red Hat, CentOS 5.5. Because HiTEC and Reptile do not run in parallel, we performed the testing in both serial and parallel modes, to include all programs in the comparison. We have used 24 cores in parallel mode.

The results are presented in Table 2, which contains the error-correcting performance in the top part in percentage of errors corrected, the time in the middle part in seconds per megabyte of input base pairs and space in the bottom part in megabytes required per megabyte of input base pairs. Some programs could not run all datasets and the reasons are indicated by letters explained in the caption of the table. The last row in each part represents the average of the entries for those datasets for which all programs could run. To improve the visualization of the tables, we have used color intensities with darker color representing better performance.
other programs. Quake is again second best and uses >50% more space than RACER in parallel mode.

RACER and Reptile were the only programs able to run all datasets. Quake came second in time and space, except serial space where it is third, but its error-correcting performance was second last in serial and last in parallel, about half of the performance of RACER. This is in part because Quake trims reads when unable to correct. The evaluation counts the number of base pairs in all correct reads. Reptile came third in time and second in space (parallel mode) but its accuracy was last. SHREC required clearly the highest space and for that reason it was able to run the fewest datasets on our machine. On the other hand, it came consistently in third place for correcting performance. Coral was the slowest in serial mode but had the best speedup among all programs, and its parallel performance was close to that of Quake and SHREC. The 15× speedup came at the cost of increasing the space 12 times. RACER’s speedup was 11× with a small increase in space.

### 3 METHODS

#### 3.1 RACER

RACER belongs to the class of $k$-mer counting programs. It uses 2-bit encoding of nucleotides and random replacement of the unknown positions. Each $k$-mer is represented as a 64-bit integer. The $k$-mers are stored in a hash table. For each $k$-mers, count, which is assumed correct if the count is higher than $t$ and erroneous otherwise. The approximate size of the sequenced genome is required as input, from which RACER automatically computes the $k$-mer size $k$ and threshold $t$ used in the correction part. The combination of parameters used has been experimentally determined. Like HiTEC, the thresholds are varied to achieve higher correcting performance.

A significant space advantage over SHREC and HiTEC is given by the avoidance of full text indexes, such as suffix trees and suffix arrays. The direct comparison shows a significant difference.

### 3.2 Evaluation details

To obtain the most accurate evaluation, we have avoided the use of mapping programs or processing of corrected data for other applications, such as genome assembly. Mapping programs are used for evaluation to know where the errors are and how they should be corrected. However, many reads cannot be mapped uniquely and even more cannot be mapped at all. Such reads are discarded and, therefore, the obtained datasets are significantly different from the original. In particular, all programs have artificially increased performance because the reads that cannot be mapped at all. This reason creates a speed advantage because the $k$-mers and counters, once computed, can be used multiple times for correction.

RACER corrects reads of varying length, from both fasta and fastq data. It has been implemented in C++ and OpenMP and no other software is necessary to run it.

### 2.3 Comparison

As far as error-correcting performance is concerned, only HiTEC comes close to RACER. However, HiTEC requires the second highest space and, therefore, cannot run the largest datasets. For D._melanogaster, HiTEC did not attempt to run it because of different read lengths. Note also HiTEC’s poor performance on some of the datasets that were not included in the average. Direct comparison shows a significant difference.

Concerning time, RACER was the fastest in both serial and parallel modes, twice faster than second-placed Quake in serial and about one order of magnitude faster than all the others in parallel mode. RACER is also more memory efficient than the
performance was computed as \( (e_b - e_a)/e_a \). Because we have \( e_b = TP + FN, e_a = FP + FN \) (Ilie et al., 2011), we have that \( \frac{e_b}{e_a} = \frac{TP - FP}{TP + FP} \), which is the same as the formula used by Yang et al. (2013). By considering whole-read correction, as explained above, instead of individual-based correction, similar performance is expected; however, the problems associated with read mapping are avoided.

Nevertheless, because in some articles the comparison is performed on mapped datasets, we provide, for completeness, such a comparison in the Supplementary Material where the datasets have been mapped using BWA (Li and Durbin, 2009) with high error rate, so that a high number of reads are kept and bias is minimized. The comparison on the mapped datasets is similar with the one on the raw datasets except that, as expected, the percentage of errors corrected increases.

All programs have been run according to the instructions given by the authors in the corresponding articles, Web sites or readme files. We have not tuned the parameters of any of the programs to improve performance, as this would be unrealistic. The commands used to run all programs are given in the Supplementary Material.

4 DISCUSSION

Extensive testing shows that RACER is superior to the existing programs in all aspects: error-correcting performance, time and space. It corrects about three-quarters of the errors in <2 min for a bacterial dataset and 30–40 min for a larger organism such as a worm or fly. RACER does not require any additional software to run.

5 CONCLUSION

We have presented a new tool, RACER, for correcting errors in NGS data. The current version of RACER targets Illumina data, thus correcting substitution error. Future versions will be able to handle indel errors to enable correction of data from other sequencing platforms as well as mixed data.

ACKNOWLEDGEMENTS

Performance evaluation has been performed using the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) and Compute/Calcul Canada.

Funding: The Natural Sciences and Engineering Research Council of Canada (NSERC) (in parts to L.I.).

Conflict of Interest: none declared.

REFERENCES