In Search of Perfect Reads

Soumitra Pal*
*Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Powai, Mumbai, 400076, India
Email: mitra@cse.iitb.ac.in

Srinivas Aluru†*
†School of Computational Science and Engineering
College of Computing, Georgia Institute of Technology
266 Ferst Drive, Atlanta, GA 30332, USA
Email: aluru@cc.gatech.edu

Abstract—Continued advances in next generation short-read sequencing technologies are increasing through and read lengths, while driving down the error rates, for example within 1% for Illumina HiSeq reads. Moreover, the errors are not uniformly distributed in all reads, and a large percentage of reads are indeed error-free. Ability to predict such perfect reads can have significant impact on run-time complexity of applications.

In this paper, we present a simple and fast k-spectrum analysis based method to identify error-free reads. Our experiments show that if around 80% of the reads in a dataset are perfect, then our method retains almost 99.9% of them with more than 90% precision rate. Though filtering out reads identified as erroneous by our method reduces the coverage by about 7% on an average, coverage pattern across genome remains similar. The filtration process can be customized at several levels of stringency depending upon the downstream application need.

Keywords: Next generation sequencing, error correction

I. INTRODUCTION

High-throughput short read sequencing technologies have become the mainstay of genomic research. Critical attention is paid to read quality as it affects the quality and performance of sequencing applications. For example, read quality directly impacts accuracy in mapping to a reference genome. In de novo genome sequencing, apart from accuracy of the generated contigs, read quality affects contig lengths. Error-free reads can also improve algorithmic performance, as alignments can be replaced with much faster exact matching.

In applications such as resequencing and de novo sequencing in which a single genome is sampled at high coverage, the infrequent occurrence of errors in reads, and the apparent lack of affinity of errors to any fixed genomic location, provide a way to detect and correct erroneous bases in reads. If the reads covering a specific genomic position can be identified and properly positioned relative to their locations of genomic occurrence, this layout can be used to infer the true base by majority vote and correct the others. Several error correction algorithms have been developed, using k-spectrum [5], [10], suffix trees [2], [6], or multiple sequence alignments [3], [7] to identify overlapping reads. For a detailed survey of error correction methods, see [8], [9].

Most error correction methods are designed for Illumina sequencers, which are predominantly used. Such high-end sequencers have error rates well within 1%, and a large percentage of reads are indeed claimed to be free of errors. Taking advantage of this, in this paper we propose a different approach: rather than base-level error correction, we seek to identify reads that are error-free (or perfect). If such predictions can be made with high accuracy, it opens the door to simplifying algorithms for downstream applications, not to mention improvements in quality.

II. METHODS

Our algorithm is based on analyzing the k-spectrum of a given data set, which constitutes the collection of all its kmers. Define a kmer to be valid if it is present in the genome being sequenced, and invalid otherwise. A read is perfect if it does not contain any invalid kmer.

In the absence of the reference genome, the validity of a kmer can be estimated from its frequency in the input reads and the quality scores of the constituent bases. As errors are infrequent, with sufficient coverage, a valid kmer should occur at significantly larger frequency than invalid kmers. Thus, in the first phase, we estimate frequency f(T) of a kmer T by counting each instance of T with all bases having quality at least Q_E (stands for Excellent Quality). We also construct a graph to link kmers within short Hamming distance.

In the second phase, each read is checked for potential errors by decomposing it into overlapping kmers such that the overlap between two consecutive kmers is k/2, half their length. If there are insufficient base pairs for such an overlap towards the end of a read, the last kmer is chosen to be the suffix of the read of length k. If all of these kmers are estimated to be valid, the algorithm outputs the read as perfect. The algorithm relies on five different rules based on properties P1…P5 below to estimate if a kmer T is valid:

P1: f(T) ≥ C_E
P2: f(T) ≥ C_G and each base pair in T has quality ≥ Q_G
P3: f(T) ≥ C_G and T does not have a neighbor T' in Hamming graph with f(T') ≥ C_G
P4: f(T) ≥ C_G and T does not have a neighbor T' in Hamming graph with f(T') ≥ f(T) × F_H
P5: f(T) ≥ C_G and all neighbors T' of T in the Hamming graph have property: all the base pairs of T where T differs from T' have quality score ≥ Q_G

where the parameters are to be set appropriately: C_E (Excellent Count), C_G (Good Count), Q_G (Good Quality), F_H (High-cardinality Factor), C_E > C_G and Q_E > Q_G.

T is estimated to be valid by Rule i if T satisfies any one of the properties P1, P2, …., P_i. Thus, the rules are in decreasing order of stringency. In the most stringent case (P1), the algorithm treats T as valid only if f(T) attains a threshold C_E. In P2, f(T) is allowed to be above a lower threshold C_G but each base in T must have quality score above Q_G. In less stringent P3 and P4, T with comparatively lower f(T) is allowed as there are no high cardinality Hamming neighbors and T might be from a low coverage region. In P5, T has strong quality scores at all the positions in which it differs from its neighbors, and hence it is highly likely to be valid. Our algorithm can be customized by selecting a suitable Rule. For...
very few perfect reads from the original dataset, and can do
will be operating on data that has over 90% perfect reads, miss
algorithm, instead of the raw datasets. Doing so, the applications
coverage and lower percentage of perfect reads.

performance on S1 can be attributed to the comparatively lower
perfect reads, is over 90% in all cases except S1. The lower
and S3). Precision, the ratio of true perfect reads to predicted
datasets indicates to what extent our algorithm succeeded in
weeding out reads that contain at least one error. Except for
made, we define the following:

example, if the objective is to retain most of the perfect reads
despite the risk of increasing false positives, then Rule 5 should be
used. On the other hand, if the objective is to minimize false
positives, Rule 1 should be used.

III. RESULTS

We applied our algorithm to 6 datasets from the NCBI short
read archive, the details of which are given in Table I. Dataset
S1 is from HiSeq 2500 platform and the rest are from HiSeq
2000. To evaluate our method, we aligned each dataset using
the BWA aligner [4] with default parameters. A read is taken
to be error-free if it is perfectly aligned by BWA without any
substitution, insertion, or deletion.

We used the following default values of parameters: \(k=24\),
\(C_G=1\), \(C_E=8\), \(Q_G=45\). We chose \(Q_E=71\) for S3 and \(Q_E=73\)
for the remaining datasets. To assess the quality of predictions
made, we define the following: \(TP\) for number of perfect reads
classified by our algorithm as perfect, \(FN\) for perfect reads
classified as erroneous, \(FP\) for erroneous reads classified as perfect,
and \(TN\) for erroneous reads classified as erroneous. We use the standard measures of specificity \(S_p = TN/(TN + FP)\),
sensitivity \(S_n = TP/(TP + FN)\), and precision \(P_r = TP/(TP + FN)\).

The results of our experiments using Rule 2, which tests
for conformance with at least one of properties P1 and P2, are
presented in Table II. Except for dataset S2, Rule 2 achieves
near 100% sensitivity, indicating this rule correctly classifies
an overwhelming majority of perfect reads, and misclassifies
near 100% sensitivity, indicating this rule correctly classifies
for erroneous reads classified as erroneous.

Thanks to the continuous technological improvements in
high-throughput DNA sequencing, reads of dominant sequenc-
ing platforms such as the Illumina HiSeq are sporting high
coverage and accuracy. This has now reached an extent where
we can envision just filtering out reads with errors, thus
making error correction less important. Our algorithm is a first
attempt at this new paradigm. Our experiments demonstrate
that development of such algorithms shows great promise.

ACKNOWLEDGMENTS

This work is supported in part by NSF under ATD-1120597
and CCF-1360593, and a DST Swarnajayanti Fellowship from the
Government of India.

REFERENCES

[1] F. García-Alcalde, K. Okonechnikov, J. Carbonell, L.M. Cruz, S. Götz,
S. Tarazona, J. Dopazo, T.F. Meyer, and A. Conesa. Qualimap: Eval-
uating Next-generation Sequencing Alignment Data. Bioinformatics,
28(20):2678–2679, 2012.

[2] L. Ilie, F. Fazayeli, and S. Ilić. HiTEC: Accurate Error Correction
in High-throughput Sequencing Data. Bioinformatics, 27(3):295–302,
2011.

[3] W.C. Kao, A.H. Chan, and Y.S. Song. ECHO: A Reference-free Short-
read Error Correction Algorithm. Genome Research, 21(7):1181, 2011.

[4] H. Li and R. Durbin. Fast and Accurate Short Read Alignment
with Burrows–Wheeler Transform. Bioinformatics, 25(14):1754–1760, 2009.

[5] P. Medvedev, E. Scott, B. Kakaradov, and P. Pevzner. Error Correction
of High-throughput Sequencing Datasets with Uniform Coverage.
Bioinformatics, 27(13):i137–i141, 2011.

[6] L. Salmela. Correction of Sequencing Errors in a Mixed Set of Reads.
Bioinformatics, 26(10):1284–1290, 2010.

[7] L. Salmela and J. Schröder. Correcting Errors in Short Reads by
Multiple Alignments. Bioinformatics, 27(11):i1455–i1461, 2011.

[8] M. Tahir, M. Sardaraz, A.A. Ikram, and H. Bajwa. Review of Genome
Sequence Short Read Error Correction Algorithms. American Journal of
Bioinformatics Research, 3(1):1–9, 2013.

[9] X. Yang, S.P. Chockalingam, and S. Aluru. A Survey of Error
Correction Methods for Next-generation Sequencing. Briefings in
Bioinformatics, 14(1):56–66, 2013.

[10] X. Yang, K.S. Dorman, and S. Aluru. Reptile: Representative Tiling for
Short Read Error Correction. Bioinformatics, 26(20):2526–2533, 2010.

TABLE I: Sequence Datasets

| Set | SRA Accession | Reference Genome | Strain | Size (Mb) | Read Cov. Size of Reads | Number of Reads | Perfect Reads (%) | Error Rate |
|-----|---------------|-----------------|--------|----------|------------------------|----------------|------------------|-----------|
| S1  | SRR790669 D. Miranda MSH22 136.73 90 43 | 16480814 | 85855982 | 52 | 0.71 |
| S2  | SRR467456 E. coli O157:H7 5.59 101 | 20461422 | 13294580 | 065 | 0.72 |
| S3  | ERR036168 P. Falciparum 3D7 23.27 75 | 11242870 | 74845636 | 67 | 0.43 |
| S4  | ERR142615 B. Pertussis ST24 4.12 75 | 1000 54996906 | 44437437 | 81 | 0.24 |
| S5  | ERR142617 P. Falciparum 3D7 23.27 75 | 160 49783806 | 40472551 | 81 | 0.18 |
| S6  | SRR057777 S. Cerevisiae S288c 12.16 76 | 362 57886340 | 51612399 | 89 | 0.13 |

TABLE II: Results Using Default Parameters and Rule 2

| Set | TP | FN | FP | TN | \(P_r\) | \(S_p\) | \(S_n\) |
|-----|----|----|----|----|-------|-------|-------|
| S1  | 3323396 | 786 | 25789056 | 6588766 | 0.563 | 0.204 | 1.000 |
| S2  | 10890703 | 2403877 | 781985 | 6384877 | 0.633 | 0.391 | 0.819 |
| S3  | 74699298 | 146338 | 4348595 | 33224039 | 0.884 | 0.998 |
| S4  | 44392266 | 45171 | 3873795 | 6685674 | 0.920 | 0.633 | 0.999 |
| S5  | 40420872 | 51679 | 4422332 | 6685674 | 0.920 | 0.633 | 0.999 |
| S6  | 51587979 | 33420 | 1681374 | 4592567 | 0.968 | 0.732 | 0.999 |

TABLE III: Results on Dataset S6 Using Different Rules

| Rule | TP | FN | FP | TN | \(P_r\) | \(S_p\) | \(S_n\) |
|------|----|----|----|----|-------|-------|-------|
| Rule 1 | 51341705 | 270694 | 1354288 | 491563 | 0.974 | 0.784 | 0.995 |
| Rule 2 | 51578979 | 33420 | 1681374 | 4592567 | 0.968 | 0.732 | 0.999 |
| Rule 3 | 51579033 | 33866 | 1681628 | 4592313 | 0.968 | 0.732 | 0.999 |
| Rule 4 | 51605360 | 7039 | 1684518 | 4589423 | 0.968 | 0.732 | 1.000 |
| Rule 5 | 51605719 | 6680 | 1690728 | 4583214 | 0.968 | 0.732 | 1.000 |