A New Method for DNA Sequencing Error Verification and Correction via an On-Disk Index Tree *

Yarong Gu
The University of Michigan, Dearborn, USA
yarongg@umich.edu

Xianying Liu
The University of Michigan, Dearborn, USA
xianying@umich.edu

Qiang Zhu
The University of Michigan, Dearborn, USA
qzhu@umich.edu

Youchao Dong
The University of Michigan, Dearborn, USA
youchaod@umich.edu

C. Titus Brown
Michigan State University, East Lansing, USA
ctb@msu.edu

Sakti Pramanik
Michigan State University, East Lansing, USA
pramanik@msu.edu

ABSTRACT

Existing sequencing error correction techniques demand large expensive memory space. In this work, we introduce a new disk-based sequencing error correction method to solve the problem. The key idea is to utilize a special on-disk index structure, called the BoND-tree, to store and access a large set of k-mers and their associated metadata on disk. With the BoND-tree, a set of special box queries to retrieve the relevant k-mers and their counts are efficiently processed. A comprehensive voting mechanism is adopted to determine and correct an erroneous base in a genome sequence. Experiments demonstrate that the proposed method is quite promising in verifying and correcting sequencing errors in terms of accuracy and scalability.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences – Biology and Genetics

General Terms

Algorithms

Keywords

Bioinformatics, sequencing error correction, disk index tree

1. INTRODUCTION

DNA sequencing is increasingly serving studies on numerous biological problems. However, current sequencers have quite high random per-base error rates. Error correction has emerged as one of the dominant practical problems in sequence analysis. Error correction and other sequence analysis applications have made counting large amounts of k-mers a paramount need for research in bioinformatics.

In order to deal with the computational challenges for large k-mer datasets, a number of efficient k-mer counting methods have been proposed in the literature. Most of them use a large in-memory structure such as hash table, Bloom filter, suffix tree, and sorted bin set to store the k-mers. These in-memory structures provide efficient random access to k-mers in memory. However, these methods usually have a high demand on computing resources. For example, Jellyfish [3], which is a popular hash based counting method, was run on a computer with 32 cores and 256GB RAM. Such high-end computing equipment is still not common for most biology laboratories today. Quake [2], which is a widely-used error correction tool/method, utilizes Jellyfish to perform the counting job during its error correction process. Beyond a plain k-mer counting, Quake also takes into account the quality scores of base calls when distinguishing untrusted k-mers (i.e., with low-abundance) from trusted ones.

One way to reduce the expensive memory requirement for an error correction method is to develop a new technique utilizing relatively cheap disk space. To tackle these challenges, we present a novel disk-based sequencing error correction method in this work.

2. THE METHOD

Our sequencing error correction method first loads the overlapping k-mers obtained from the sequencing reads for a target genome sequence along with their relevant metadata (e.g., ids of the sequencing reads that contain the k-mer, which could not be stored in a conventional memory-based method due to expensive memory cost) into a so-called BoND-tree [1] on disk. The BoND-tree is a recent index technique that was specially designed for supporting efficient processing of box queries in a non-ordered discrete data space (NDDS) [4, 5] with datasets like k-mers on disk. It has a balanced hierarchical indexing tree structure (see Fig. 1). Each (leaf or non-leaf) node consists of a set of entries and occupies one disk block. Each entry in a non-leaf node consists of a pointer pointing to a subtree and the minimum bounding box (mbb) for all the k-mers stored in the subtree. Each entry in a leaf node consists of a k-mer and a pointer pointing to relevant metadata. Special strategies making use of the characteristics of the underlying NDDS

*Research supported by the US National Science Foundation under Grants #IIS-1320078 and #IIS-1319909.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).
BCB'15, September 9–12, 2015, Atlanta, GA, USA.
ACM 978-1-4503-3853-0/15/09.
http://dx.doi.org/10.1145/2808719.2811429.

for k-mers are adopted to build the tree so that box queries can be processed efficiently.

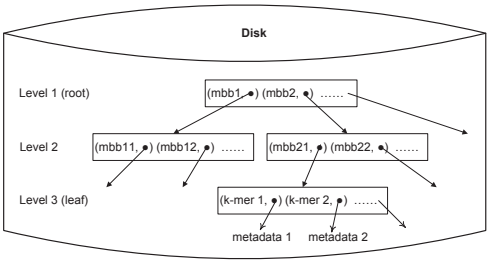

Figure 1: The BoND-tree Structure.

Given a sequencing read (e.g., read 3 in Fig. 2) and a suspicious error position in the read (e.g., the position having t in read 3), the possibly erroneous base (i.e., t) at the position can be verified and corrected by a vast majority voting approach described as follows. We can choose a suspicious k-mer (e.g., β in Fig. 2) that covers the suspicious error position and replace the possibly erroneous base (i.e., t) at the position by a set/box $X = \{a, t, c, g\}$ to form a box query to be executed on the BoND-tree. Since there usually is a high coverage (e.g., about 20 times) with sequencing reads at the suspicious error position, most of the reads typically contain the correct base at the position. Using a vast majority voting rule, we can discover if the possibly erroneous base is indeed an error and, if so, what the correct base (e.g., a in Fig. 2) is. Once an erroneous base has been verified, the error correction is just a matter of replacing the erroneous base by the correct one.

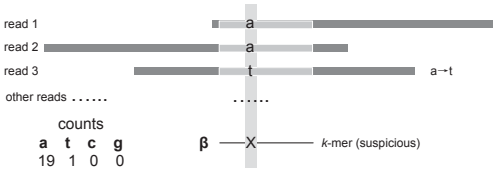

Figure 2: Voting in Case of Solo Occurence of a Suspicious k-mer.

However, the above simple voting strategy cannot handle the situation when a suspicious k-mer β that we use to convert into a box query for a suspicious error position happens to also appear as a trusted (without error) k-mer at another location of the target genome sequence. To solve this problem, we consider all the shifted k-mers $\beta_1, \beta_2, ..., \beta_k$ that cover the suspicious error position. For each β_i $(1 \leq i \leq k)$, we form a box query by replacing the possibly erroneous base at the suspicious error position by a set/box $X = \{a, t, c, g\}$. After these k (shifted) box queries are performed on the BoND-tree, k counts for each base can be obtained. The minimal count for each base is then found. If the maximum among the four minimal counts is close to the coverage number, its corresponding base is determined to be the correct one at the suspicious position by our improved vast majority voting strategy. The chance for all k shifted k-mers to be repeated at two locations is small.

Our sequencing error correction method also adopts strategies to handle situations in which multiple errors occur in one read or the vast majority voting mechanism fails to give a determination.

3. EXPERIMENTAL RESULTS

To examine the performance of our method, we conducted experiments using genome data collected from *E. coli* 536 (GenBank: NC008253, 5.5 M). Simulated 36 bp reads with an error rate at 0.5% were used. Our method was implemented in the C++ programming language. All the experiments were conducted on a Dell PC with a 3.2 GHz Intel Core i7-4790 CPU, 12 GB RAM, 5 TB Hard Drive, and Linux 3.16.0 OS.

Table 1 shows the accuracy and efficiency of our method when k ranges from 12 to 16 and the coverage is 15 (times). The results demonstrate that, in general, the accuracy of our method is increasingly better as k increases. $k = 15$ is the optimal length determined by Quake [2] for the same dataset. For those error positions that Quake tried to correct, it corrected 99.8% of them, which is comparable to the observed accuracy of our method. On the other hand, the experimental results demonstrate that our method can yield a quite high accuracy even when k is relatively small (e.g., $k = 12$). Utilizing this advantage could help us further reduce the space requirement by storing shorter k-mers, leading to an enhanced scalability for our method. Furthermore, we can see that the error correcting time for our method is relatively small, which indicates that the BoND-tree can help achieve high efficiency for a disk based sequencing error correction method. On the other hand, the creation of a BoND-tree took significant amount of time, which was bound to the efficiency of the index building algorithm given in [1]. Strategies such as bulk loading and streaming loading can be adopted to improve the index creation efficiency.

Table 1: Correction Accuracy and Time for Simulated 36 bp *E. coli* with Coverage=15

k	Total errors	Correc -tions	Accuracy (%)	Create idx (min)	Correct err (min)
12	812170	784286	96.57%	254	41
13	812840	806516	99.22%	274	27
14	813143	810879	99.72%	287	24
15	813865	812837	99.87%	290	27
16	813055	812408	99.92%	301	25

4. CONCLUSIONS

In this work, a new disk based sequencing error correction method utilizing an on-disk index structure to efficiently perform a set of carefully designed box queries on the k-mer dataset is presented. Experiments demonstrate that the method is quite promising in terms of accuracy and efficiency besides the scalability benefit obtained from using inexpensive disk space. Furthermore, the method can provides high accuracy even when the k-mer length is small, resulting in further enhanced scalability.

5. REFERENCES

[1] C. Chen, A. Watve, S. Pramanik and Q. Zhu. The BoND-tree: An efficient indexing method for box queries in nonordered discrete data spaces. *IEEE TKDE*, 25(11):2629–2643, 2013.

[2] D. R. Kelley, M. C. Schatz, S. L. Salzberg, et al. Quake: quality-aware detection and correction of sequencing errors. *Genome Biol*, 11(11):R116, 2010.

[3] G. Marçais and C. Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.

[4] G. Qian, Q. Zhu, Q. Xue, and S. Pramanik. Dynamic indexing for multidimensional non-ordered discrete data spaces using a data-partitioning approach. *ACM TODS*, 31(2):439-484, 2006.

[5] G. Qian, Q. Zhu, Q. Xue and S. Pramanik. A space-partitioning-based indexing method for multidimensional non-ordered discrete data spaces. *ACM TOIS*, 23(1):79-110, 2006.