Bioinformatics, 31(10), 2015, 1577–1583
doi: 10.1093/bioinformatics/btv024
Advance Access Publication Date: 20 January 2015
Original Paper

OXFORD

Sequence analysis

Ultrafast SNP analysis using the Burrows–Wheeler transform of short-read data

Kouichi Kimura* and Asako Koike

[1]Biosystems Research Department, Central Research Laboratory, Hitachi, Ltd., 1-280 Higashi-Koigakubo, Kokubunji, Tokyo 185-8601, Japan

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on September 11, 2014; revised on December 24, 2014; accepted on January 12, 2015

Abstract

Motivation: Sequence-variation analysis is conventionally performed on mapping results that are highly redundant and occasionally contain undesirable heuristic biases. A straightforward approach to single-nucleotide polymorphism (SNP) analysis, using the Burrows–Wheeler transform (BWT) of short-read data, is proposed.

Results: The BWT makes it possible to simultaneously process collections of read fragments of the same sequences; accordingly, SNPs were found from the BWT much faster than from the mapping results. It took only a few minutes to find SNPs from the BWT (with a supplementary data, *fragment depth of coverage* [FDC]) using a desktop workstation in the case of human exome or transcriptome sequencing data and 20 min using a dual-CPU server in the case of human genome sequencing data. The SNPs found with the proposed method almost agreed with those found by a time-consuming state-of-the-art tool, except for the cases in which the use of fragments of reads led to sensitivity loss or sequencing depth was not sufficient. These exceptions were predictable in advance on the basis of *minimum length for uniqueness* (MLU) and FDC defined on the reference genome. Moreover, BWT and FDC were computed in less time than it took to get the mapping results, provided that the data were large enough.

Availability and implementation: A proof-of-concept binary code for a Linux platform is available on request to the corresponding author.

Contact: kouichi.kimura.hh@hitachi.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Since the advent of so-called next-generation DNA sequencers (NGSs), which rapidly and cost-effectively generate billions of short reads, large-scale analysis of sequence data of a few-hundred giga base pairs (Gbp), requiring a large computational resources, is not uncommon anymore. The first step to extract biologically meaningful information from sequence data often involves analysis of the variation (mutation) of that data in comparison with a reference genome sequence data. Billions of short reads are first mapped onto the reference genome, and unambiguous and recurrent mismatches between the short reads and the reference genome are identified as candidate mutations (DePristo *et al.*, 2011). This line of approach is hereafter referred to as *the mapping-based approach*. Although it is the most appreciated and most commonly used approach, it has the following basic weak points. (i) The computation of mapping is highly redundant because of large sequencing depth (typically ranging from 30× to 100×). (ii) Some mutations can be lost by mapping tools because such tools use certain heuristics of their own to resolve mapping ambiguities. (iii) It is not easy to switch from one reference genome to another after the computation of mapping has been completed.

To address these weak points, an alternative solution, *the dictionary-based approach*, is proposed. The short-read data are converted into *a dictionary of reads*, so that numbers of occurrences of any sequence in the short-read data are immediately obtained. Then,

© The Author 2015. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

mutations can be inferred on the basis of these numbers by means of querying genomic subsequences with and without the mutations (Fig. 1). The dictionary can be implemented efficiently by means of the Burrows–Wheeler transform (BWT) (Burrows and Wheeler, 1994) [a.k.a. FM index (Ferragina and Manzini, 2000)] because it is simple and particularly suitable for DNA sequences.

Although the proposed approach appears to be too naïve, it has the following potential advantages. (i) Redundancy due to deep sequencing coverage is efficiently managed by the dictionary of reads. (ii) The dictionary of reads does not suffer information loss or heuristic bias because it is essentially constructed by means of sorting the data in alphabetical order. (iii) It is easy to switch from one reference genome to another one after the dictionary of reads is constructed. However, the following issues of the proposed approach remain to be addressed.

1. The construction of BWT for large data (more than 100 Gbp) is very time-consuming even with the fastest known algorithm, BCRext (Bauer *et al.*, 2011; Cox *et al.*, 2012).
2. A large number of occurrences in the short-read data of a genomic subsequence with a candidate mutation do not necessarily imply the existence of the mutation because some of them might be derived from different genomic regions with similar subsequences.
3. To get useful information from the dictionary of reads, it is necessary to prepare effective queries that are likely to contain candidate mutations. Namely, it is necessary to locate genomic positions with a significant chance of finding mutations.

To address these issues, the following algorithm and concepts are introduced in this study.

1. BWT/WT, a modified and parallelized BCRext algorithm for computing the BWT of reads.
2. *The minimum length for uniqueness* (MLU), a simple criteria for evaluating the uniqueness of the subsequence.
3. *The fragment depth of coverage* (FDC), an estimate of sequencing depth of coverage on the basis of exact matching of read fragments at single-base resolution.

2 Dictionary-based approach

2.1 Notations

For any DNA sequence, $A = a_0 a_1 \cdots a_{L-1}$, $A[i,j]$ denotes the subsequence $a_i a_{i+1} \cdots a_j$ for $0 \le i \le j < L$, and $\overline{A}$ denotes the reverse complement. Let $C_0, C_1, \ldots, C_{I-1}$ be the DNA sequences of chromosomes (or contigs) in the reference genome, where I is the number of them. Let $G^+ = C_0\$C_1\$\cdots C_{I-1}\$$ and $G^- = \overline{C}_{I-1}\$\cdots\$\overline{C}_1\$\overline{C}_0\$$ denote the concatenations on the positive and negative strand in mutually reverse order, where \$ denotes a sentinel (punctuation) symbol. The whole-genome sequence on both strands is represented by $G^* = G^+G^-$, and the BWT, denoted by $T(G^*)$, is used as the dictionary of the reference genome, where the alphabetical order $\$ < A < C < G < T < N$ is assumed and \$ does not to match any other symbols, including itself. Let L be the total length of the genome on a single strand including the sentinels; namely, $L = |G^+| = |G^-|$. For a genomic coordinate, $0 \le x < L$, the reference base at x on the positive (negative) strand is given by $G^*[x]$ ($G^*[\overline{x}]$), where $\overline{x} = 2L - 1 - x$. Similarly, a genomic subsequence on the positive (negative) strand of length equal to ℓ with the left (right) end at x is given by $G^+(x,\ell) = G^*[x, x+\ell-1]$ ($G^-(x,\ell) = G^*[\overline{x}, \overline{x}+\ell-1]$).

Let $r_1, r_2, \ldots, r_J$ be the DNA sequences of the short reads, and $R = r_1\$r_2\$\cdots\$r_J\$$ be the concatenation, where J denotes the number of reads. The BWT of R, denoted by $T(R)$, is used as the dictionary of the reads.

For any DNA sequence, w, the number of occurrences of w in the reference genome and that in the short-read data, denoted by $N_{G^*}(w)$ and $N_R(w)$, are immediately computed from $T(G^*)$ and $T(R)$, respectively, by using rank functions (González *et al.*, 2005) [a.k.a. LF mappings (Ferragina and Manzini, 2000)].

2.2 MLU

The MLU at x in the positive (negative) direction, denoted by $\lambda^+(x)$ ($\lambda^-(x)$), is defined as the minimum length of the subsequence with the left (right) end at x, such that the subsequence appears only once in both strands of the genome. Namely,

$$\lambda^{\pm}(x) = \min\{\ell \,|\, N_{G^*}(G^{\pm}(x,\ell)) = 1\}. \quad (1)$$

MLU is closely related to the suffix array (SA) and the longest common prefix (LCP) array (Manber and Myers, 1990) and is thereby computed efficiently (see Section 3).

MLU varies with position on the genome and mostly takes a moderate value, except in the case of repetitive or duplicated regions. For example, MLU is 40 or less (more than 100) in 88.2% (3.9%) of the reference human genome, hg19, excluding long runs of N with more than 500 Kbp.

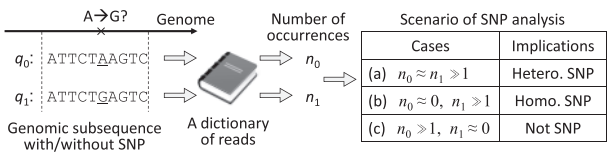
2.3 FDC

When a genomic subsequence with the left end at x is taken as a query, namely, $w = G^+(x,\ell)$, the number of occurrences of w in short-read data, $N_R(w)$, reflects the sequencing depth at x, provided that the length ℓ is properly chosen. If ℓ is less than $\lambda^+(x)$, the number is clearly overestimated because some of the occurrences come from different positions. If ℓ is equal to $\lambda^+(x)$, the number is expected to be a proper estimation of the sequencing depth because of the uniqueness condition. However, it is in fact prone to be affected by occasional contributions from reads [with sequencing errors or single-nucleotide polymorphisms (SNPs)] derived from different genomic regions with similar sequences. Therefore, ℓ somewhat larger than MLU should be taken. On the other hand, if ℓ is too large, the number is underestimated because of the finite read length.

The FDC at x in the positive (negative) direction, denoted by $d^+(x)$ ($d^-(x)$), is defined as the number of occurrences of w in the short-read data when the length is chosen, such that $\ell = \lambda^+(x) + \alpha$ ($\ell = \lambda^-(x) + \alpha$) for a small positive constant, α. Namely,

$$d^{\pm}(x) = N_R(G(x, \ell^{\pm}(x))),\ \ell^{\pm}(x) = \lambda^{\pm}(x) + \alpha. \quad (2)$$

As is clear from the above definitions, FDC has single-base resolution and is sensitive to direction (Fig. 2a). For example, FDCs in



Scenario of SNP analysis	
Cases	Implications
(a) $n_0 \approx n_1 \gg 1$	Hetero. SNP
(b) $n_0 \approx 0,\ n_1 \gg 1$	Homo. SNP
(c) $n_0 \gg 1,\ n_1 \approx 0$	Not SNP

Fig. 1. Basic concept of dictionary-based SNP analysis. Given a SNP candidate on the genome, appropriate genomic subsequences with and without the SNP, namely, q_1 and q_0, are chosen as queries. The numbers of occurrences of the queries in short read-data, n_1 and n_0, are immediately obtained from the dictionary of short reads. The candidate is evaluated on the basis of n_1 and n_0 as follows. (**a**) If both n_0 and n_1 are sufficiently large and almost equal, the candidate is likely to be a heterozygous SNP. (**b**) If n_0 is sufficiently small, and n_1 is sufficiently large, the candidate is likely to be a homozygous SNP. (**c**) Conversely, if n_0 is sufficiently large, and n_1 is sufficiently small, the candidate is likely to be false

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

both directions suddenly drop in the vicinity of a SNP in a characteristic pattern (Fig. 2b).

Although FDC is an approximation of sequencing depths, it is progressively underestimated as MLU increases because of the finite read length. In particular, when MLU is greater than the read length, FDC is zero and useless. Otherwise, on the assumption that the sequencing errors are randomly distributed according to a Poisson distribution with an average frequency of r_E per base, the underestimation factor is given by $(1-\ell^{\pm}(x)/\ell_R)e^{-r_E\ell^{\pm}(x)}$, where ℓ_R denotes read length.

2.4 Overall scheme

The reference genome sequence and the short-read data are separately transformed into dictionaries (BWTs). The MLU is computed from the dictionary of the reference genome sequence alone. The FDC is computed from the dictionary of the short-read data and the MLU. These precomputations are necessary for genetic-variation analysis downstream (Fig. 3a). In contrast, as for the conventional mapping-based approach, the reference genome sequence is formatted into a convenient form, which is sometimes implemented by means of BWT (Li and Durbin, 2009). The short reads are mapped onto the reference genome using the formatted data, and the results are sorted and indexed according to the positions on the genome (Li *et al.*, 2009). These precomputations are necessary for downstream analysis (Fig. 3b).

In contrast to the mapping-based approach (by which reads are treated individually), the dictionary-based approach is expected to be efficient in the downstream-analysis phase because collections of read fragments with the same sequences can be processed simultaneously.

3 Methods

3.1 Calculation of BWTs

Contigs separated by long runs of N (500 Kbp or more), $C_1, C_2, \ldots, C_I$, are extracted from the reference genome sequence, and the concatenated genome sequence (on both strands), G^*, is obtained. In the case of the reference human genome, hg19, $I = 47$ contigs and G^* of length $2L \simeq 5.75 \times 10^9$ are thus obtained. The BWT and SA of G^* are calculated using the induced-sorting algorithm (Nong *et al.*, 2011). The SA-IS code presented in Nong *et al.* (2011) is modified, so that it can treat data larger than 4 Gbp and cope with multiple occurrences of sentinels.

Large short-read data of more than 100 Gbp is beyond the scope of the induced-sorting algorithm. The BWT of short reads is incrementally calculated from smaller partial BWTs in a cache-oblivious manner, which basically follows the BCRext algorithm (Bauer *et al.*, 2011; Cox *et al.*, 2012). The kth partial BWT is defined as the BWT of k-suffixes of reads, where the k-suffix is a suffix of length k (if the read length is larger than k) or the entire read otherwise. Both of the partial BWTs and the remaining prefix data are compactly encoded into wavelet trees (Grossi *et al.*, 2003), resulting in a memory requirement of about $0.6N$ GB for data of N Gbp. The incremental calculations are executed in parallel according to the first bases of the suffixes, thus accelerating the calculation three to four times. The modified BCRext algorithm is hereafter referred to as 'BWT/WT'.

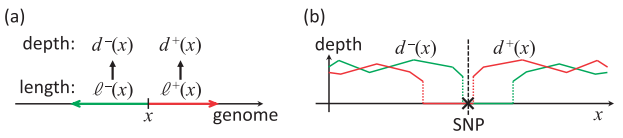


Fig. 2. FDC has base-level resolution and is sensitive to direction. (**a**) The FDCs at x in the positive and negative directions, $d^+(x)$ and $d^-(x)$, are defined as the number of occurrences in the short-read data of genomic fragments on the positive and negative strands starting at x with length $\ell^+(x)$ and $\ell^-(x)$, which are chosen to be larger than MLU: $\ell^{\pm}(x) = \lambda^{\pm}(x) + \alpha$ for a small positive constant α. (**b**) The FDC in the positive (negative) direction drops at a SNP position and in its left (right) vicinity because of the difference between the reference genome and short reads at the SNP position. The widths of drops are determined by the MLU and α. The depths of drops are halved when the SNP is heterozygous

3.2 Calculation of MLU

The SA of both strands of the reference genome,

$$\mathrm{SA}_{G^*} = (\mathrm{SA}_{G^*}[0], \mathrm{SA}_{G^*}[1], \ldots, \mathrm{SA}_{G^*}[2L-1]), \quad (3)$$

is a permutation of $0, 1, \ldots, 2L-1$, such that the sequence of suffixes,

$$(\, G^*[\mathrm{SA}_{G^*}[i], 2L-1]\,)_{i=0,1,\ldots,2L-1},$$

is sorted alphabetically (Manber and Myers, 1990). The LCP array of the genome,

$$\mathrm{LCP}_{G^*} = (\mathrm{LCP}_{G^*}[0], \mathrm{LCP}_{G^*}[1], \ldots, \mathrm{LCP}_{G^*}[2L-1]) \quad (4)$$

is an array of integers, where $\mathrm{LCP}_{G^*}[i]$ is the length of the LCP of suffixes $G^*[j, 2L]$ with $j = \mathrm{SA}_{G^*}[i]$ and $j = \mathrm{SA}_{G^*}[i-1]$ (Manber and Myers, 1990). The LCP can be efficiently calculated from the SA and its relatives (Kärkkäinen *et al.*, 2009). Then, MLU is given by

$$\lambda^+(x) = \max\,(\mathrm{LCP}_{G^*}[\mathrm{SA}_{G^*}^{-1}[x]], \mathrm{LCP}_{G^*}[\mathrm{SA}_{G^*}^{-1}[x]+1]) + 1, \quad (5)$$

$$\lambda^-(x) = \max\,(\mathrm{LCP}_{G^*}[\mathrm{SA}_{G^*}^{-1}[\overline{x}]], \mathrm{LCP}_{G^*}[\mathrm{SA}_{G^*}^{-1}[\overline{x}]+1]) + 1 \quad (6)$$

for $0 \le x < L$, where $\mathrm{SA}_{G^*}^{-1}$ denotes the inverse suffix array, i.e. the inverse permutation of the SA.

All of the values of the MLU are compactly encoded into a bit array as follows. Since $G^*(x+1, \lambda^+(x+1))$ occurs exactly once in both strands of the genome, its leftward one-base extension, $G^*(x, \lambda^+(x+1)+1)$, occurs at most once in both strands. This fact implies that

$$\lambda^+(x) \le \lambda^+(x+1) + 1 \quad (0 \le x < L), \quad (7)$$

and hence $2x + \lambda^+(x)$ is strictly increasing with x. Similarly,

$$\lambda^-(x) \le \lambda^-(x-1) + 1 \quad (0 \le x < L), \quad (8)$$

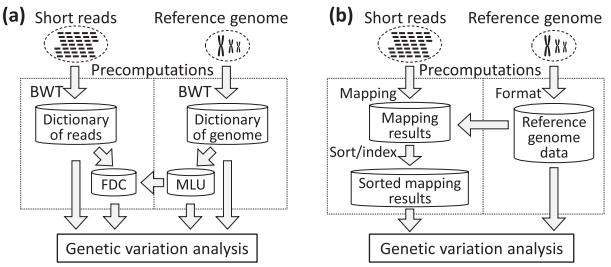


Fig. 3. Comparison of the overall schemes. (**a**) Proposed approach. The reference genome sequence is transformed into the dictionary (BWT), and MLU is calculated. The short-read data are transformed into the dictionary (BWT), and FDC is calculated from the BWT and MLU. The calculation results are used in the genetic-variation analysis downstream. (**b**) Conventional approach. The reference genome sequence is formatted into a convenient form (sometimes BWT), and the short reads are mapped onto the reference genome. The mapping results are sorted and indexed and used in downstream analysis

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

and hence $2\overline{x} + \lambda^-(x)$ is strictly decreasing with $\overline{x}$. On the basis of these implications, the MLU values are compactly encoded into bit array M of length $4L$ as follows:

$$M[y] = \begin{cases} 1 & (y = 2x + \lambda^+(x) \quad \text{for some } 0 \le x < L), \\ 1 & (y = 2\overline{x} + \lambda^-(x) \quad \text{for some } 0 \le x < L), \\ 0 & (\text{otherwise}). \end{cases} \tag{9}$$

Conversely, they are immediately decoded from M as follows:

$$\lambda^+(x) = \text{select}_M(x) - 2x, \ \ \lambda^-(x) = \text{select}_M(\overline{x}) - 2\overline{x} \tag{10}$$

for $0 \le x < L$, where $\text{select}_M(x)$ (i.e., the select function on M) gives the index of the xth occurrence of a set bit ('1') in M (González *et al.*, 2005). The select function is efficiently calculated using the hierarchical binary strings (HBS) (Kimura *et al.*, 2009).

3.3 Calculation of FDC

The SA of the short reads (only in the direct strand),

$$\text{SA}_R = (\text{SA}_R[0], \text{SA}_R[1], \ldots, \text{SA}_R[N-1]), \tag{11}$$

is a permutation of $0, 1, \ldots, N-1$, such that the sequence of suffixes,

$$(R[\text{SA}_R[i], N-1])_{i=0,1,\ldots,N-1},$$

is sorted alphabetically, where N is the total length of the short-read data including sentinels, namely, $N = |R|$.

For any DNA sequence, w, a collection of all occurrences of w in the short-read data is represented by the *SA interval* of w (Li and Durbin, 2009), $[\underline{I}_R(w), \overline{I}_R(w)]$, which is defined by

$$\underline{I}_R(w) = \min\{0 \le i < N \mid w \text{ is a prefix of } R[\text{SA}_R[i], N-1]\}, \tag{12}$$

$$\overline{I}_R(w) = \max\{0 \le i < N \mid w \text{ is a prefix of } R[\text{SA}_R[i], N-1]\}. \tag{13}$$

The initial value for the empty sequence ($w = \varepsilon$) is given by $[\underline{I}(\varepsilon), \overline{I}(\varepsilon)] = [0, N-1]$. It is then recursively calculated as follows.

$$\underline{I}_R(aw) = C(a) + \text{rank}_{T(R)}(a, \overline{I}_R(w) - 1), \tag{14}$$

$$\overline{I}_R(aw) = C(a) + \text{rank}_{T(R)}(a, \overline{I}_R(w) + 1) \tag{15}$$

for $a = A, C, G, T$ and N, where $C(a)$ is the number of bases in R that are lexicographically smaller than a, and $\text{rank}_{T(R)}(a, i)$ is the number of occurrences of a in $T(R)[0, i-1]$. The rank function is efficiently computed using the HBS.

Then, FDCs are given by the lengths of the SA intervals as follows:

$$d^\pm(x) = \overline{I}_R(G^\pm(x, \ell^\pm(x))) - \underline{I}_R(G^\pm(x, \ell^\pm(x))) + 1, \tag{16}$$

for $0 \le x < L$. Therefore, FDCs are calculated using Equations (14–16).

Moreover, the calculation is accelerated according to an idea similar to (Kärkkäinen *et al.*, 2009). It is common for adjacent positions, x and $x \pm 1$, to have the subsequences, $G^\pm(x, \ell)$ and $G^\pm(x \pm 1, \ell)$, in Equation (1), such that they have the same end position, namely $x \pm \lambda^\pm(x) = x \pm 1 \pm \lambda^\pm(x \pm 1)$. Then, $d^\pm(x \pm 1)$ is *reducible* in the sense that it is immediately given by Equation (16) with the known values of $\underline{I}_R$ and $\overline{I}_R$ that are obtained during the calculation of $d^\pm(x)$ using Equations (14–16).

3.4 Search for SNP candidates

Two methods for searching for SNP candidates, namely *the drop-scan method* and *the step-scan method*, are proposed in the following. As for the drop-scan method, SNPs are searched for only around significant drops in the precomputed FDC (since they are unlikely to be found elsewhere). As for the step-scan method, the whole genome is exhaustively scanned by a sliding window of a fixed size, and reads with exactly matching subsequences around the window on either side are collected, and their extensions into the window are examined to find any SNPs therein. The latter method does not require the precomputed FDC.

3.4.1 Drop-scan method

A simple criterion for genomic coordinate x to be a significant leftward (rightward) drop in the positive (negative) direction is

$$d^+(x) < (1-r)d^+(x+1) \quad \left(d^-(x) < (1-r)d^+(x-1)\right) \tag{17}$$

where $0 < r < 1$ is a small constant, referred to as *drop ratio*. However, random fluctuations of the FDC may sometimes satisfy the criteria; besides, reads that are derived from different homologous genomic regions and altered by SNPs or sequencing errors may affect the criteria. The criteria are therefore made more stringent by the additional following procedures (Fig. 4).

1. Take $s = G^\pm(x \pm 1, \ell^\pm(x \pm 1))$, a seed (of a sufficient length) adjacent on the right (left) of x, and collect all of its occurrences in the forward (reverse) reads.
2. Collect all possible leftward (rightward) extensions beyond x in a sufficient length, $\ell^\mp(x \mp 1) + 1$.
3. Align the extensions with the reference genome using a fast dynamic programming (DP) algorithm (Kimura *et al.*, 2012).
4. Select valid extensions that have at most n_e mismatches or small indels (insertions or deletions), where n_e is a positive constant integer.
5. Find any mismatches or small indels that are repeatedly observed in the alignments of the valid extensions in at least n_m cases and with a relative frequency of at least r, where n_m is a positive constant integer.
6. Filter out any mismatches or small indels that are not found consistently from both strands.

3.4.2 Step-scan method

SNP candidates are located by using a sliding window of fixed length W along the genome in the following steps (Fig. 5).

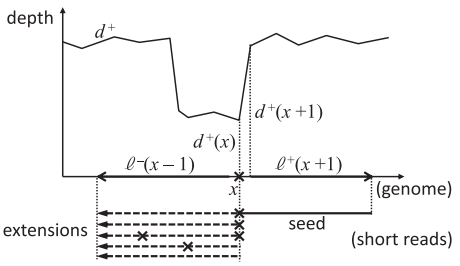


Fig. 4. Drop-scan method. A leftward drop of the FDC at x in the positive direction, such that $d^+(x) < (1-r)d^+(x+1)$ is located by scanning the whole genome, where $0 < r < 1$ is the drop ratio. For such x, a seed is taken as a genomic subsequence on the positive strand started at $x+1$ and with length equal to $\ell^+(x+1)$. The occurrences of the seed in the short read-data and all possible leftward extensions are collected by using the dictionary of short reads. The extensions (including x and beyond) are aligned with the reference genome sequence, and repeatedly observed mismatches or small indels are extracted. Likewise, rightward drops of the FDC in the negative direction are considered. The mismatches and indels consistently extracted from both directions are then obtained as SNP candidates

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

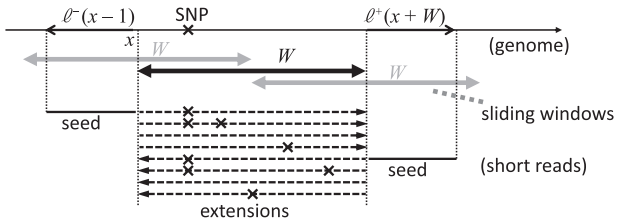


Fig. 5. Step-scan method. The whole-genome region is scanned by a sliding window of length W at every $W/2$ bp position. Two adjacent seeds on the right and left sides, starting, respectively, at $x+W$ and $x-1$, are taken on the positive and negative strand. Their lengths are, respectively, equal to $\ell^+(x+W)$ and $\ell^-(x-1)$. The occurrences of the right-hand (left-hand) seed in the short-read data and all possible leftward (rightward) extensions into the window are collected by using the dictionary of short reads. The extensions are aligned with the reference genome sequence, and consistent mismatches or small indels are extracted as SNP candidates

1. Take a window of length W with the left end at $x = kW/2$ for $k = 0, 1, 2, \ldots, \lfloor 2L/W \rfloor$.
2. Take $s = G^+(x+W, \ell^+(x+W))$ $(s = G^-(x-1, \ell^-(x-1))$, a seed (of a sufficient length) adjacent on the right (left) side of the window and collect all occurrences of s in the forward (reverse) reads.
3. Collect all possible leftward (rightward) extensions into the window.
4. Align the extensions with the reference genome using a fast DP algorithm.
5. Select valid extensions that have at most n_e mismatches or small indels.
6. Find any mismatches or small indels that are repeatedly observed in the alignments of the valid extensions in at least n_m cases and with a relative frequency of at least r.
7. Filter out any mismatches or small indels that are not found consistently from both strands.

4 Results and discussion

The dictionary-based methods were implemented in C++ and Perl for proof-of-concept experiments. The test data included actual biological data downloaded from public websites and simulation data (Table 1).

4.1 Precomputation time

The BWT and MLU of both strands of the reference human genome sequence (hg19) were calculated once, and the calculation results were stored and reused. The total computation time was 2 h and 16 min by a single core of an Intel Xeon CPU (X7560, 2.3 GHz); and the maximum memory usage was 71.3 GB.

The precomputation for short-read data was performed in parallel; 10 threads were used for the exome and transcriptome sequencing data, and 24 threads were used for the whole-genome sequencing data. In the case of transcriptome and genome sequencing data, the computation time was much shorter than that required by the conventional mapping-based method (Table 2).

Although the computation time of BWT/WT was almost linear in relation to the data size, the computation time of FDC was mostly dominated by the length of the reference genome and was not much affected by the data size owing to the nature of the dictionary. Thus, the precomputation time for the smaller data size was largely occupied by the latter time.

Table 1. Test data for experiments

Data	Source	Type	Size	Read length
S1	(simulation)	Exome	10.0 Gbp	100 bp
E1	SRX043462	Exome	6.7 Gbp	120 bp
E2	SRX253902	Exome	12.4 Gbp	100 bp
E3	SRX506949	Exome	16.8 Gbp	100 bp
E4	SRX097050	Exome	22.9 Gbp	101 bp
T1	SRX472980	Transcriptome	6.3 Gbp	100 bp
T2	SRX588484	Transcriptome	14.3 Gbp	100 bp
T3	SRX105217	Transcriptome	20.3 Gbp	99 bp
G1	ERX009609	Whole genome	135.3 Gbp	100 bp
G2	ERX069715	Whole genome	137.1 Gbp	100 bp
G3	ERX168840	Whole genome	163.4 Gbp	100 bp

Actual biological data were downloaded from the NCBI Sequence Read Archive. They were paired-ended reads of human samples obtained by Illumina Genome Analyzer II, IIx and HiSeq 2000. The simulation data were generated from the human reference genome sequence (hg19) around NCBI RefSeq coding exons with randomly introduced homozygous and heterozygous SNPs (0.1%) and sequencing errors (1%), each of which consists of single-base substitutions (98%), insertions (1%) and deletions (1%); the insertion lengths of the paired reads were assumed to be distributed normally with 300-bp mean and 20-bp standard deviation.

Table 2. Comparison of precomputation time for short-read data

Data	Dictionary-based method			Mapping based method
	BWT/WT	FDC	Total	
S1	31 m 22 s	1 h 06 m 49 s	1 h 38 m 11 s	41 m 14 s
E1	22 m 50 s	1 h 05 m 53 s	1 h 28 m 43 s	51 m 50 s
E2	33 m 25 s	1 h 12 m 29 s	1 h 45 m 54 s	1 h 57 m 38 s
E3	44 m 38 s	1 h 14 m 28 s	1 h 59 m 06 s	2 h 02 m 45 s
E4	1 h 11 m 37 s	1 h 14 m 22 s	2 h 25 m 59 s	2 h 53 m 05 s
T1	16 m 24 s	1 h 03 m 45 s	1 h 20 m 09 s	4 h 21 m 01 s
T2	32 m 30 s	1 h 10 m 50 s	1 h 43 m 19 s	10 h 32 m 51 s
T3	48 m 41 s	1 h 22 m 25 s	2 h 11 m 07 s	19 h 59 m 55 s
G1	11 h 46 m 19 s	1 h 16 m 37 s	13 h 02 m 56 s	32 h 02 m 33 s
G2	11 h 58 m 22 s	1 h 14 m 54 s	13 h 13 m 15 s	34 h 07 m 40 s
G3	13 h 42 m 13 s	1 h 18 m 16 s	15 h 00 m 29 s	56 h 57 m 17 s

BWT/WT and FDC were computed as described in the text. As for the mapping-based method, BWA 0.7.8 mem (Li and Durbin, 2009) was used for the exome and genome data, and TopHat 2.0.7 (Kim *et al.*, 2013) was used for the transcriptome data; conversion into BAM files, sorting, and merging was done by SAMtools 0.1.19 (Li *et al.*, 2009). A Linux workstation with a single CPU (Intel Core i7-4930 K, 3.4 GHz) and 64-GB memory was used with 10 threads in parallel for the exome and transcriptome data; and a Linux server with double CPUs (Intel Xeon X7560, 2.3 GHz) and 256-GB memory was used with 24 threads in parallel for the genome data.

4.2 Example of FDC plots

Although FDC plots, as illustrated in Figure 2b, are informative and meaningful, they are usually degraded by noises induced by randomly matched sequences. However, as parameter α increases, the noise reduces rapidly, while the signal degrades slowly. Thus, $\alpha = 3$ was chosen tentatively so as to make the plots clear and sensitive. In the case of the exome data, the signals were apparently localized around captured regions and clearly dropped at SNPs. An example of actual biological data (E1) is shown in Figure 6. Similarly, in the case of transcriptome data, the signals were apparently localized inside exons; besides, considerable amounts of signals and noises, seemingly to reflect complex alternative splicing and other miscellaneous transcriptional activities, were also observed. In the case of the

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

whole-genome data, copy-number variations and loss of heterogeneity were also observed as expected.

4.3 Search for SNP candidates

Parameters $n_e = 8$, $n_m = 2$, $r = 0.2$ and $W = 40$ were chosen experimentally using simulation data (S1), so as to trade-off computation time and sensitivity. The sensitivities of finding homozygous and heterozygous single-base substitutions were 94.3% and 93.6% by the drop-scan method and 94.5% and 94.3% by the step-scan method. They were slightly smaller than 95.8% and 95.4% attained by a conventional mapping-based method, GATK (DePristo *et al.*, 2011). The primary reason for slightly lower sensitivity of the two proposed methods is thought to be the fact that the methods do not use full lengths of reads (with paired-end information); instead, they only use fragments of reads.

It is expected that the sensitivity is prone to decrease as MLU increases. In fact, the sensitivity of the drop-scan method for the simulated exome data was 99.2% when MLU was at most 40 (93.3% of the cases), while it decreased to 26.9% when MLU was >40 (6.7% of the cases).

Table 3. Comparison of times for searching for SNPs

Data	Dictionary-based method		Mapping-based method (GATK, Picard)
	Drop-scan	Step-scan	
S1	1 m 14 s	17 m 10 s	3 h 11 m 08 s
E1	1 m 19 s	15 m 56 s	1 h 40 m 24 s
E2	2 m 18 s	21 m 30 s	2 h 31 m 37 s
E3	2 m 33 s	24 m 56 s	3 h 13 m 19 s
E4	6 m 21 s	25 m 34 s	12 h 28 m 20 s
T1	2 m 23 s	16 m 22 s	2 h 46 m 05 s
T2	2 m 31 s	20 m 39 s	9 h 35 m 13 s
T3	9 m 51 s	32 m 48 s	2 h 58 m 13 s
G1	19 m 41 s	2 h 23 m 57 s	39 h 34 m 19 s
G2	20 m 05 s	2 h 24 m 17 s	41 h 10 m 19 s
G3	14 m 24 s	2 h 13 m 22 s	44 h 01 m 33 s

Precomputed FDC was used for the drop-scan method but not for the step-scan method. GenomeAnalysisTK 2.1-8 (DePristo *et al.*, 2011) and Picard tools 1.77 (http://picard.sourceforge.net/) were used. Each data were computed in parallel with the same number of threads by the same computer as for Table 2.

Times for searching for SNPs in the whole human genome by different methods are compared in Table 3. As expected, the drop-scan method is much faster than the step-scan method, and the latter is much faster than the conventional mapping-based method.

The agreement of the results given by the proposed and conventional methods is assessed as follows. It is known that GATK is one of the most-sensitive mapping-based tools and that the ratio of agreement of results given by different tools is not generally high (Pabinger *et al.*, 2014). Therefore, the result given by the proposed method (the drop-scan method) and that given by GATK confidently (after Q-filtering, where the quality value was not <300) in less repetitive (with M-filtering, where the MLU was at most 40 bp) and deeply covered (with D-filtering, where FDCs around the drops were at least 10) regions were compared (Table 4). As indicated by the relative sensitivity and specificity of the proposed method on the assumption that the results given by GATK were correct, high ratios of agreement were obtained.

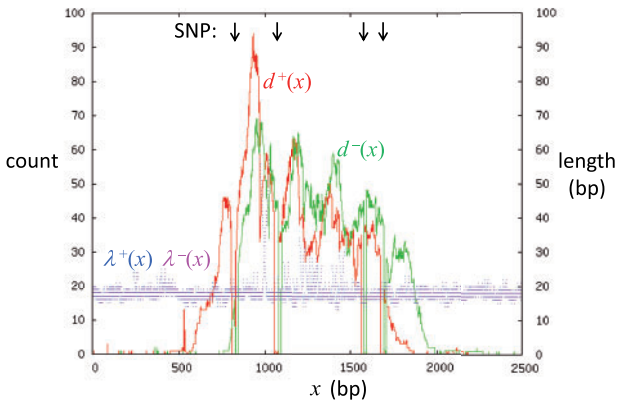


Fig. 6. MLU and FDC plots in a captured region of exome sequencing data (E1). The MLUs on the positive and negative strand, $\lambda^+(x)$ and $\lambda^-(x)$, along the right-hand ordinate (in length), and the FDCs on the positive and negative strand, $d^+(x)$ and $d^-(x)$, along the left-hand ordinate (in count), are plotted against the relative coordinate in a genomic region (chr17:3 100 001-3 102 500 in hg19). The SNP candidates, located at sharp drops of FDCs, are indicated by downward arrows

Table 4. Agreement of SNPs found by the drop-scan method and the mapping-based method

Data	Mapping-based					Dictionary-based		Relative accuracy of [a] assuming that [b] is true			
	All	Q-filtered	Q/M-filtered	(ratio)	Q/M/D-filtered[b]	All	D-filtered[a]	TP	FP	Sensitivity	Specificity
S1	133 361	106 148	100 977	(95.1%)	79 506	112 303	79 186	79 069	117	99.5%	99.9%
E1	237 562	75 692	68 727	(90.8%)	21 885	67 820	20 689	20 457	232	93.5%	98.9%
E2	442 788	156 336	135 716	(86.8%)	58 027	110 219	51 561	50 585	976	87.2%	98.1%
E3	414 171	103 431	96 204	(93.0%)	41 015	87 588	36 925	36 576	349	89.2%	99.1%
E4	222 419	67 441	61 609	(91.4%)	33 053	85 298	28 234	27 708	526	83.8%	98.1%
T1	112 071	26 742	24 196	(90.5%)	7891	54 324	7894	7099	795	90.0%	89.9%
T2	175 395	44 654	39 570	(88.6%)	18 529	73 207	16 829	15 848	981	85.5%	94.2%
T3	579 699	112 103	103 636	(92.4%)	32 402	275 562	35 005	30 547	4458	94.3%	87.3%
G1	5 043 802	4 522 189	3 817 262	(84.4%)	2 869 000	3 939 619	2 844 998	2 686 923	158 075	93.7%	94.4%
G2	5 102 468	4 608 488	3 884 808	(84.3%)	2 989 489	3 988 886	2 961 803	2 799 135	162 668	93.6%	94.5%
G3	4 289 853	3 915 819	3 264 571	(83.4%)	2 773 595	3 305 833	2 726 082	2 592 566	133 516	93.5%	95.1%

Numbers indicate the number of SNP candidates (only for single-base substitutions) found by each method and those filtered under specified conditions. Q-filter removed those with a quality value <300; M-filter removed those with MLU > 40 and D-filter removed those with FDC around the drops <10. The mapping-based results were obtained by GATK 2.1-8, and the dictionary-based results were obtained by the drop-scan method. The Q-filtering eliminates the SNP candidates to which GATK does not give high confidence. The M-filtering eliminates those in regions where the drop-scan method is known to be insensitive. The ratios of the numbers in the third and fourth columns are given in the fifth column; they are larger in the exome and transcriptome data than in the genome data (see text). The D-filtering eliminates those in regions where the sequencing coverage is not deep enough. [a]Final results given by the dictionary-based method. [b]Final results given by the mapping-based method, tentatively assumed to be correct. (TP: true positives, FP: false positives).

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

The M-filtering eliminates the SNP candidates in regions where the dictionary-based methods are known to be insensitive. The reduction ratios (in the fifth column in Table 4) are generally larger for the exome and transcriptome data and smaller in the genome data in comparison to 88.2%, which is the proportion of genomic regions where MLU is 40 or less. This result seems to be reasonable because the former data mostly consist of reads from gene regions where MLU generally takes smaller values and because the latter data also contain many reads from repetitive regions where MLU is very large and mutation rate (including SNPs) is relatively high. Thus, the ratio of useful SNPs that are predicted to be undetectable by the drop-scan method is estimated to be around 10%.

Similar results to those given in Table 4 were also obtained by the step-scan method (Supplementary Information, Supplementary Table S1). The step-scan method generally gives higher relative accuracy than the drop-scan method for exome and genome sequencing data but not for transcriptome data. The primary reason for this lower accuracy in the latter case is thought to be the fact that the sliding windows often cross the exon boundaries, making it impossible to start effective searches from either side of the windows.

5 Conclusion and future works

In contrast to the conventional mapping-based approach, a dictionary-based approach to sequence analysis is proposed. It is expected to be efficient because the dictionary (BWT) of short-read data makes it possible to simultaneously process collections of read fragments with the same sequences. In particular, SNPs were found from the dictionary much faster than from the mapping results. It was experimentally shown that it took only a few minutes to find SNPs from the BWT and FDC using a desktop workstation in the case of human exome or transcriptome sequencing data and 20 min using a double-CPU server in the case of human genome sequencing data.

However, the use of read fragments (instead of full-lengths of reads with paired-end information) sometimes leads to sensitivity loss. Such cases are predictable in advance on the basis of MLU and are estimated to generally occupy about 10% of the cases; therefore, the proposed approach should be taken only in the majority of other cases.

The SNPs obtained by the proposed methods mostly agreed with those obtained by a time-consuming state-of-the-art tool, except for the cases in which loss of sensitivity was predicted in advance on the basis of MLU or sequencing depth was estimated to be low on the basis of FDC.

The dictionary of short-read data was computed in less time than it took to map them onto a reference genome and to sort the mapping results along the genome, provided that the data was large enough. It was free from heuristic bias or information loss, unlike the mapping results. Since it does not depend on any particular reference genome sequence, the dictionary-based approach will be advantageous when multiple reference sequences are available.

Although this study focuses exclusively on SNP analysis, it is clear that the proposed approach is generally applicable to many other kinds of sequence analysis. In particular, straightforward and promising applications include:

1. Alternative splicing analysis of transcriptome data
    a. Sensitive detection of different combinations of exon junctions by means of consulting the dictionary of reads.
    b. Detection of novel alternative exons from FDC plots.
2. Structural variation analysis of genome data
    a. Sensitive detection of split reads (across break points associated with deletions or translocations) by means of consulting the dictionary of reads.
    b. Identification of insertions as extensions (by repeatedly performing LF mappings on the dictionary of reads) from partially mapped read fragments.
    c. Copy number variation analysis on the basis of FDC plots.

MLU will be useful to avoid false detections in many of these applications.

Conflict of Interest: none declared.

References

Bauer,M.J. *et al.* (2011). Lightweight BWT construction for very large string collections. In: Giancarlo,R. and Manzini,G. (eds) *Combinatorial Pattern Matching,* volume 6661 of *Lecture Notes in Computer Science*. Springer, Berlin, pp. 219–231.

Burrows,M. and Wheeler,D.J. (1994). A block-sorting loss-less data compression algorithm. *SRC Res. Rep.*, **124**.

Cox,A.J. *et al.* (2012). Large-scale compression of genomic sequence databases with the Burrows-Wheeler transform. *Bioinformatics*, **28**, 1415–1419.

DePristo,M.A. *et al.* (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.*, **43**, 491–498.

Ferragina,P. and Manzini,G. (2000). Opportunistic data structures with applications. In: *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, 2000, IEEE, pp. 390–398.

González,R. *et al.* (2005). Practical implementation of rank and select queries. In: *Proceedings of the 4th International Workshop on Experimental and Efficient Algorithms (WEA'05)*, Santorini, 2005, pp. 27–38.

Grossi,R. *et al.* (2003). High-order entropy-compressed text indexes. In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, Baltimore, 2003, pp. 841–850. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Kärkkäinen,J. *et al.* (2009). Permuted longest-common-prefix array. In: Kucherov,G. and Ukkonen,E. (eds.) *Combinatorial Pattern Matching*, volume 5577 of *Lecture Notes in Computer Science*. Springer, Berlin, pp. 181–192.

Kim,D. *et al.* (2013). Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.*, **14**, 1–13.

Kimura,K. *et al.* (2009). Computation of rank and select functions on hierarchical binary string and its application to genome mapping problems for short-read DNA sequences. *J. Comput. Biol.*, **16**, 1601–1613.

Kimura,K. *et al.* (2012). A bit-parallel dynamic programming algorithm suitable for DNA sequence alignment. *J. Bioinform. Comput. Biol.*, **10**, 1250002.

Li,H. and Durbin,R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li,H. *et al.* (2009). The sequence alignment/map format and samtools. *Bioinformatics*, **25**, 2078–2079.

Manber,U. and Myers,G. (1990). Suffix arrays: a new method for on-line string searches. In: *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '90, pp. 319–327. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Nong,G. *et al.* (2011). Two efficient algorithms for linear time suffix array construction. *IEEE Trans. Comput.*, **60**, 1471–1484.

Pabinger,S. *et al.* (2014). A survey of tools for variant analysis of next-generation genome sequencing data. *Brief. Bioinform.*, **15**, 256–278.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016