BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 23 2011, pages 3228–3234
doi:10.1093/bioinformatics/btr563

Genome analysis

Advance Access publication October 12, 2011

SVseq: an approach for detecting exact breakpoints of deletions with low-coverage sequence data

Jin Zhang* and Yufeng Wu

Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA

Associate Editor: Alex Bateman

ABSTRACT

Motivation: Structural variation (SV), such as deletion, is an important type of genetic variation and may be associated with diseases. While there are many existing methods for detecting SVs, finding deletions is still challenging with low-coverage short sequence reads. Existing deletion finding methods for sequence reads either use the so-called split reads mapping for detecting deletions with exact breakpoints, or rely on discordant insert sizes to estimate approximate positions of deletions. Neither is completely satisfactory with low-coverage sequence reads.

Results: We present SVseq, an efficient two-stage approach, which combines the split reads mapping and discordant insert size analysis. The first stage is split reads mapping based on the Burrows–Wheeler transform (BWT), which finds *candidate* deletions. Our split reads mapping method allows mismatches and small indels, thus deletions near other small variations can be discovered and reads with sequencing errors can be utilized. The second stage filters the false positives by analyzing discordant insert sizes. SVseq is more accurate than an alternative approach when applying on simulated data and empirical data, and is also much faster.

Availability: The program SVseq can be downloaded at http://www.engr.uconn.edu/~jiz08001/

Contact: jinzhang@engr.uconn.edu

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

Received on May 3, 2011; revised on September 15, 2011; accepted on October 6, 2011

1 INTRODUCTION

A deletion is a structural variation (SV) in which a segment of DNA is missing comparing with a reference genome. We focus on developing computational methods for finding deletions in this article. Previously, comparative genomic hybridization with whole-genome tiling arrays (Sebat *et al*., 2004) was the primary method for characterizing structural variations. Now high-throughput sequencing (HTS) technologies (such as the Roche 454 FLX, Illumina Genome Analyzer, and ABI SOLiD) become more available. These HTS technologies have been applied in generating huge amount of sequence data and newer methods for finding SVs are mainly developed for sequence data. For example, the pilot study of the 1000 genomes project uses multiple methods to discover SVs from whole-genome sequence data collected by different technologies from hundreds of individuals. As more individuals are being sequenced at low coverage for the purpose of finding variations at a population level, it is important to develop SV detection algorithms that are efficient for processing large amount of sequence reads and accurate given low-coverage short sequence reads. These algorithms should be able to handle sequencing errors that are hard to avoid due to technological limitations. Refer to Hormozdiari *et al*. (2009); Lee *et al*. (2010); Medvedev *et al*. (2009); Ye *et al*. (2009) for discussions on the latest methods for discovering SVs using sequence data.

In this article, we develop a new approach called SVseq for finding deletions with exact breakpoints from low-coverage next-generation sequencing data. Different from methods for estimating approximate positions of beak points, SVseq finds the breakpoints of deletions in resolution of 1 bp, which is similar to the program Pindel (Ye *et al*., 2009). Mapping the breakpoints of deletions to nucleotide resolution facilitates the analysis of the origin and functional impact of the deletions (Mills *et al*., 2011). Different from Pindel, SVseq takes a two-stage approach for discovering deletions. The first step applies an enhanced split reads mapping approach to identify *candidate* deletion sites from sequence reads. The second step uses mapped paired-end reads *spanning* candidate deletions as supports to filter false positives. In some sense, SVseq exploits more information (i.e. both the split reads and discordant paired-end reads) in the sequence data than existing approaches, which often use only one source of information. Better utilizing the given data is the key to achieve higher accuracy when dealing with low-coverage data.

2 BACKGROUND

In this article, we are mainly concerned with paired-end reads that are mapped to a reference genome by reads mapping tools. Several reads mapping tools, such as Bowtie (Langmead *et al*., 2009) and BWA (Li and Durbin, 2009), use the Burrows–Wheeler transform (BWT). Mapping with BWT requires less memory and is also efficient (Ferragina and Manzini, 2000). Note that the accuracy of reads mapping can be affected by sequencing errors and genetic polymorphisms (such as single nucleotide polymorphisms and small indels). Moreover, repeats in the genome cause further uncertainty in reads mapping.

Unmapped reads can be caused by the presence of SVs. In the case of deletions, for example, when a read contains breakpoints of a deletion site, the read will contain two portions: one from the region prior to the deletion site and one from the region following the deletion site. The read may be unmappable as a whole read on the reference genome. But if mapped as a split read, it may reveal where a deletion occurs. Figure 1 gives an example of a mapped split read that is from a deletion site.

*To whom correspondence should be addressed.

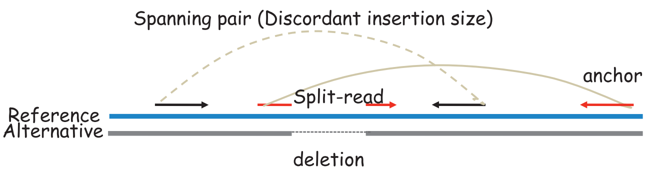


Fig. 1. Our SVseq approach maps a split read with an anchor. The direction of the anchor of this read is on the reverse strand, so that the split read is on the forward strand. The two parts of the previously unmapped read are mapped on two different positions, which may indicate that there is a deletion. The discordant distance of a spanning paired-end read supports the deletion, since its abnormally large insert size can be explained by the presence of the deletion.

Sometimes the two ends of a paired-end read is mapped in the right orientation and order, but the insert size is discordant with the library size. This can be an indication of the existence of a SV (e.g. deletion). There are many methods that detect SVs by analyzing the insert sizes of discordant pairs, such as PEMer (Korbel *et al*., 2009), BreakDancer (Chen *et al*., 2009) and VariationHunter (Alkan *et al*., 2009). Also see Suzanne *et al*. (2009) and Lee *et al*. (2010). A drawback of these methods is that only approximate positions of the breakpoints of the SVs can be found.

In practice, for the purpose of reducing sequencing cost, many existing sequence data has low-coverage. It is suggested that more variations can be found with more samples of individuals at low-coverage (The 1000 Genomes Project Consortium, 2010). With even more low-coverage sequence data being generated, new efficient algorithms that handle low-coverage sequence data are clearly needed. A natural approach for finding population deletion polymorphisms with population sequence data is to combine sequence reads from all individuals from a population and then call SVs from the combined data (Lee *et al*., 2010). Even using pooled data, the problem of finding deletions can still be difficult, not only because of the low frequency of some deletions in populations, but also due to other aspects such as sequencing errors, non-uniform reads distribution, repeats and other genetic polymorphisms.

Regarding to finding exact breakpoints, the program Pindel (Ye *et al*., 2009) is currently one of the best performing methods. According to the release of The 1000 Genomes Project Consortium (2010) and Mills *et al*. (2011), Pindel has been used in the 1000 genomes project and is credited to finding many deletions with exact breakpoints using low-coverage sequence data. However, as shown in the deletion calling results in the releases, Pindel appears to miss some deletions with exact breakpoints (which were found by other technologies).

3 METHODS

3.1 Split reads mapping based on BWT

SVseq maps both portions of a split read on a reference genome using a reads mapping algorithm based on BWT. When mapping a whole read without errors in it, the BWT mapping method described in Ferragina and Manzini (2000) can be used. Starting from one end of a sequence, this algorithm maps one charactor of the sequence in one step. The range of hits on the BWT of the sequence's mapped portion is updated until the whole read is mapped. Handling of mismatches and small indels has been introduced into BWT mapping by Bowtie (Langmead *et al*., 2009) and BWA (Li and Durbin, 2009). SVseq adopts the same method of BWA for inexact mapping. To find the two breakpoints of the a deletion using one split read, both portions of the split read need to be mapped on the reference. Currently, existing mapping tools do not map both portions of a split read.

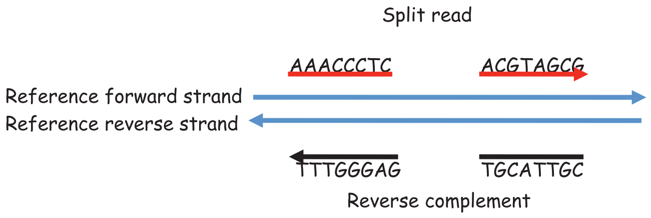


Fig. 2. Split reads mapping using BWT. Suppose the read in red color is from forward strand. Mapping it on the BWT of the forward strand reversely starts from GC to AA. Mapping from the other end is the same as mapping the reverse complement of the read on the other strand of BWT. So instead mapping from AA to CG on the forward strand, we map the reverse complement on the reverse strand from TT to GC.

The breakpoint on a split read is not clear until all ways of splitting the read are examined and the pairs of portions are mapped. A naive way for this purpose is to break the read into two portions, take each portion as a new read and run a reads mapping algorithm $2(n-2m-1)$ times, where m is the minimum allowed length of a portion. A faster approach is to store the mappings of the portions as the algorithm proceeds. From one end of a read, after the portion of length l is mapped and the mappings are stored, we then proceed to map the portion of $l+1$ bps. The same procedure is processed from the other end of the read. But since the BWT of a sequence has its derection, mapping on the BWT can only be proceeded from one end of the read. That is, the portion of length l can be mapped on the BWT, but the other portion of length $n-l$ cannot be mapped using the same BWT. SVseq is able to map from both ends of a read by mapping the read from one end on the BWT of the reference and mapping the reverse complement of the read from the other end on the BWT of the reverse strand of the reference (Fig. 2). The algorithm of SVseq only runs the BWT mapping algorithm two times for a split read this way. During the process of mapping, the length of the currently mapped portion of a read, the range of the mappings on BWT and the mapping scores are kept when the length of the portion is longer than m. The procedure stops at the portion of $n-m$. After the read is mapped from both directions, if the two portions of the read meet each other (or the length of the two portions sums up to the length of the whole read) and are mapped with the least errors (mismatches and indels), the coordinates of the portions on the reference are computed. If the positions are in the right order on the reference, we report two breakpoints on the reference of the mapped split read.

SVseq searches split reads within a certain distance from the anchor. This distance threshold is equal to the maximum allowed deletion size plus the allowed insert size. For this purpose, the whole-genome is divided into segments of equal length and the BWTs for these local segments are built. The position of the anchor in the reference genome decides which local BWT the split reads mapping uses. To make sure an anchor and the split read are in the same local segment, consecutive local segments overlap with their neighbor segments by some predetermined length. If the split portions of a read can be mapped at more than one pair of positions on the reference or the reads can have more than one split breakpoints (see Fig. 3 for details), our method only keeps the split reads mappings with the highest mapping quality. If there are still ties in the mapping quality then all these positions on the reference with proper order are collected for the use of the next steps of the algorithm.

3.2 Finding candidate deletions from split reads

When both portions of a split read are mapped, the mappings correspond to the breakpoints of a potential deletion. Note that some mapped split

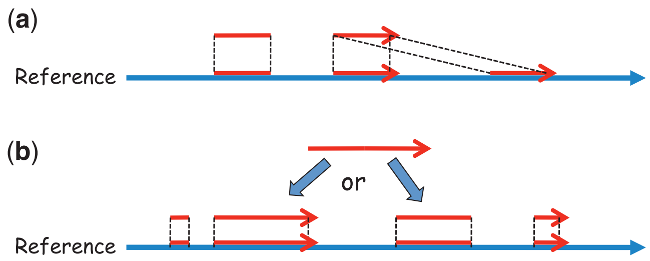


Fig. 3. Due to repeats and errors, a read can be mapped at more than one pair of positions or the reads can have more than one split breakpoints.

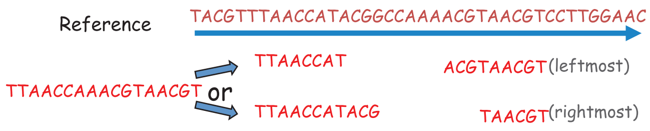


Fig. 4. Leftmost deletion breakpoints. Due to the occurrence of sequence ‘ACG’ at the two breakpoints of the deletion, if a split read from the alternative genome crosses the breakpoints of the deletion, splitting the read is not fully determined. Here, we only give the leftmost and rightmost way to break the read into two parts. SVseq uses the leftmost breakpoints to represent a deletion.

reads may contain noise. Deletions revealed by split reads become *candidate* deletions only if they are supported by a set of split reads. When working with population sequence reads, a candidate deletion can be supported by split reads from different individuals. In order to cluster the split reads that support the same candidate, the breakpoints of the split reads are converted into the leftmost breakpoints. Figure 4 shows an example of a deletion where the breakpoints can be shifted due to sequence similarity near the breakpoints. If there are no sequencing errors in the reads, the leftmost breakpoints of the split reads should be the same if they support the same deletion. The program Pindel (Ye *et al*., 2009) uses a cutoff value C to determine deletions. SVseq uses the cutoff value to call candidate deletions. That is, if a candidate is supported by at least C split reads, then we report it as a candidate deletion.

3.3 Calling deletions

Since candidate deletions found in the first stage may contain false positives, they are called as deletions when there are further evidences. Paired-end reads *spanning* candidate deletions are used in this step. We say a paired-end read spanning a candidate deletion if its two ends are mapped to different sides of the candidate. The insert size of each spanning paired-end read is examined to test whether it supports the candidate. When there is a deletion, a spanning pair is mapped on the reference genome with insert size extended by the length of the deletion. If such discordant insert size and the length of the candidate deletion match well with each other, then it is a strong signal that indicates the candidate is a true deletion. By matching well, we mean that the difference between a discordant insert size and the length of the candidate deletion is not significantly different from the library insert size. If the insert size of a pair minus the length of the candidate deletion is within 3 standard deviations of library insert size, we say the pair supports the candidate deletion. When a candidate deletion is supported by at least one spanning paired-end read, we report a deletion.

4 RESULTS

To evaluate the accuracy and efficiency of SVseq, we test our method on both simulated data and real data and compare with the program Pindel. There are two versions of Pindel:

- Pindel v_0.1.0: called Pindel v1 in this article. It only allows perfect matches in mapping split reads (Ye *et al*., 2009).
- Pindel v_0.2.0: called Pindel v2 in this article. This version has been released but not yet described in a paper. It allows mismatches and small indels in mapping split reads.

Pindel v1 has a default cutoff value $C=2$ and Pindel v2 has a default cutoff value $C=3$. We compare with Pindel v1 in finding the deletions up to 1 Mbps. But for Pindel v2, since its running time is significantly longer than Pindel v1 when finding larger SVs, we only compare with it in finding deletions up to 8092 bp. In this article, we focus on deletions that are at least 50 bps, although the algorithm of SVseq can be used to find smaller deletions. We note that there are existing methods, e.g. the program Dindel (Albers *et al*., 2011), which are dedicated for finding smaller indels. We only compare with Pindel in finding deletions, while Pindel can also find insertions, inversions and tandem duplications. Assuming paired-end reads are properly mapped, the reads picked out for the first step of SVseq are the ones with the anchors mapped as a whole read but the reads themselves could not. The reads are stored in the FASTQ format. The positions and strands of the anchors are provided in another file to SVseq. (There is a demo on the SVseq’s website). In the second step, SVseq take the BAM files as input for spanning pairs. For both steps, SVseq takes the reference in FASTA format as input.

4.1 Simulation

We generate simulated sequence reads based on human chromosome 15, which is 100, 338, 915 bp in length. In the simulation, true deletions are from the file union.2010_06.deletions.genotypes.vcf .gz on the ftp site of the 1000 genomes project release paper (Mills *et al*., 2011). Only the deletions with exact breakpoints are used and only the deletions for the 45 individuals in the CEU population are used in the simulation. There are 132 such deletions (127 of which with length 8092 bp or less). These deletions are added to the 90 genomes of the 45 individuals according to the genotype information given by the vcf file. Since the haplotypes of the deletions are not inferred in the file, for the heterogeneous deletions we arbitrarily place one such deletion to one of the two haplotypes of an individual. Since the deletions are usually far apart from each other, this may not have big effects on the accuracy of the simulation. A tool called *wgsim* (https://github.com/lh3/wgsim) is used with the ‘-h’ option to generate paired-end reads from the two copies of genomes of an individual. Single-nucleotide polymorphisms and small indels on each genomes are simulated using the default parameters. Two types of data are generated, one with read length 50 and ‘outer distance’ 200 and the other with read length 100 and ‘outer distance’ 500. For the length 50 data, coverage 1.6×, 3.2× and 4.8× are used and for the length 100 data, coverage 3.2×, 4.2× and 6.4× are used. All data are generated with base error rate 2%. BWA is used to map these simulated paired-end reads to the human reference genome. The pairs are picked out as input with one end uniquely mapped as a whole read but the other end not mapped. For example, at 4.8× coverage, for the data with read length 50, about 216 million pairs of reads are generated, and after mapping with BWA, about 208 million pairs are mapped in the right order on chromosome 15. These pairs are used to test discordant insert sizes. About 5.8 million paired-end

Table 1. Comparison of SVseq and Pindel v1 and v2 using simulated reads of lengths 50 and 100 on chromosome 15 with 132 deletions, where 127 of them are less than 8092 bp. The maximum size of deletion events is 1 Mbps when comparing with Pindel v1 and 8092 bp when comparing with Pindel v2. The cutoff value is 2 for Pindel v1, and 3 for Pindel v2. SVseq uses cutoff value 2 for the data with read length 50, and 3 for the data with read length 100. X stands for Coverage, M for Method, TP for True Positive, P v1 for Pindel v1 and P v2 for Pindel v2.

Data	X	M	Findings	TP	M	Findings	TP
50	1.6×	SVseq	74	74	SVseq	72	72
		P v1	57	56	P v2	54	53
	3.2×	SVseq	95	94	SVseq	92	91
		P v1	76	74	P v2	68	66
	4.8×	SVseq	102	100	SVseq	100	96
		P v1	83	81	P v2	81	78
100	3.2×	SVseq	111	108	SVseq	108	105
		P v1	63	62	P v2	84	83
	4.2×	SVseq	117	109	SVseq	114	106
		P v1	69	68	P v2	87	86
	6.4×	SVseq	128	120	SVseq	124	116
		P v1	85	84	P v2	104	101

reads have only one read mapped on chromosome 15 and the other read not mapped. Others reads are either not mapped or wrongly mapped.

Accuracy: The results of our method and the two versions of the program Pindel using read length 50 and 100 and different coverages are given in Table 1. The requirement of perfect mapping of Pindel v1 leads to utilizing less reads in finding deletions especially when using length 100 data. This is because at the same error rate, a longer read is more likely to contain errors. With high accuracy, SVseq is able to find more deletions than Pindel v1 and v2 provided with the same data in this simulation.

Missed deletions: Using longer reads of length 100, even with the highest coverage 6.4× in the simulation, there are still 12 deletions that are missed by SVseq (missed also by Pindel). We now take a closer look at these missed deletions and investigate why SVseq fails to find them. Low coverage is the major cause that these deletions cannot be found. Another reason is that the length of the read is short so that it is hard to map reads correctly especially when there are errors in them. For example, with the data of read length 100 and coverage 6.4×, 9 of the deletions do not have enough split reads crossing the breakpoints. The other three deletions have the coverage, but some of the split reads are not mapped on the right positions on the reference. For the data with read length 50 and coverage 4.8×, SVseq misses 32 deletions. Twenty of these deletions only have one or two individuals having the SV as heterozygous. Due to low coverage, 16 of the 20 deletions do not have enough split reads crossing the breakpoints. The other 4 deletions have 2 to 3 reads covering them, but SVseq is not able to map all of them correctly. For the remaining 12 deletions, 5 of them have higher frequency but do not have enough split reads due to low coverage and uneven distribution of reads. The remaining seven deletions are missed because some reads are wrongly mapped or are not mapped.

Split reads used: Split reads make only a very small portion of the input reads. Most of the unmapped reads are just the ones with higher error rate. SVseq utilizes more split reads that cross at deletions breakpoints than Pindel does. For example, for the data of read length 100 with coverage 6.4×, SVseq utilizes more than 5500 split reads in finding 120 correct deletions while Pindel v2 utilizes about 4800 reads in finding 101 correct deletions. Most reads Pindel utilizes are also utilized by SVseq, but there are also some reads utilized only by Pindel. For example, when simulating with the 4.2× data of read length 100 bp, comparing with the findings of SVseq, Pindel v2 finds one deletion that is missed by SVseq.

4.2 Real data

To evaluate the accuracy of SVseq with real data, SVseq is tested using the 1000 genomes project pilot 1 low-coverage data and pilot 2 high-coverage data. The results are compared with those of Pindel (Ye *et al.*, 2009) and the releases of the 1000 genomes project.

Data used in the tests:

- Low-coverage data: the pilot 1 200908 SLX BAM files from Illumina reads mapped by MAQ (Li *et al.*, 2008). The datasets of 45 individuals are used in this article. The individuals are the ones in the genotyping study of The 1000 Genomes Project Consortium (2010).
- High-coverage data: the Pilot 2 2009_07 SLX BAM files from Illumina reads mapped by MAQ (Li *et al.*, 2008). Only the datasets of the individual NA12878 are used.

Benchmarks: The 1000 genomes project has released the called SVs in Mills *et al.* (2011) based on whole-genome DNA sequencing data from 185 (179 low-coverage and 6 high-coverage) human genomes from several populations. The released SVs are called by multiple methods from multiple institutes. Extensive experimental validations have been carried out on the called deletions. The called deletions and the validation information for the low-coverage data are summerized in the Supplementary Table 3 in Mills *et al.* (2011). Most methods in this table do not call deletions with exact breakpoints (except Pindel and a method using 454 data). The called deletions and the validation information for the high-coverage trio data are summarized in the Supplementary Table 4 in Mills *et al.* (2011). Targeted breakpoint assembly is carried out and the breakpoints of deletions can be found in the Supplementary Table 18 in Mills *et al.* (2011). There are also deletions with exact breakpoints contained in the genotype vcf file union.2010_06.deletions.genotypes.vcf.gz from the ftp site of Mills *et al.* (2011). In this article, these validated and assembled deletions are used as benchmarks to evaluate the accuracy of SVseq.

The deletions in Supplementary Tables 3 and 4 in Mills *et al.* (2011) have estimated confidence intervals for both breakpoints of a deletion. In this article, for a deletion called by SVseq or Pindel, if the two breakpoints fall into the two confidence intervals given by a validated deletion, then the called deletion is counted as a true deletion. For the deletions in the Supplementary Table 18 of Mills *et al.* (2011) and in the vcf file (with exact breakpoints), the breakpoints and the length of the called deletions by SVseq or Pindel should be the same with those in the benchmarks. If there are similar sequences near the breakpoints, the called breakpoints are shifted to compare with the deletions in the benchmarks.

For the tests using low-coverage data, the following groups of benchmarks are used:

- Benchmark 1: the validated deletions in the Supplementary Table 3 of Mills *et al*. (2011), which include validated deletions from multiple institutes using different types of computational approaches: read pair (RP), read pair and read depth (PD), read depth (RD) and split read (SR). The deletions called by the Program Pindel are not used.
- Benchmark 2: the assembled deletions for the low-coverage dataset in the Supplementary Table 18 of Mills *et al*. (2011) and the deletions with exact breakpoints of the CEU population from the vcf file.

For the tests using high-coverage data, the following groups of benchmarks are usd:

- Benchmark 3: the validated deletions of the individual NA12878 in the Supplementary Table 4 of Mills *et al*. (2011), which include validated deletions from multiple institutes using different types of computational approaches: read pair (RP), read depth (RD), split read (SR) and assembly (AS).
- Benchmark 4: the assembled deletions for the trio dataset in the Supplementary Table 18 of Mills *et al*. (2011) and the deletions of the individual NA12878 with exact breakpoints from the vcf file.

It should be noted that some of the deletions in the benchmarks could not be found by SVseq and Pindel in the tests in this article because these deletions do not appear in the 45 individuals studied in this article. On the other hand, there may be true deletions found by SVseq that are not in the benchmarks. As a result, false positives may be overestimated.

Comparison with Pindel using low-coverage data: Figure 5 shows the total numbers of deletions found by SVseq and Pindel v1 and v2 using the low-coverage dataset of this article. SVseq and Pindel v1 are tested with cutoff value 2 and maximum event size 1 Mb. Given combined Benchmarks 1 and 2, SVseq finds ~14% more deletions than Pindel v1. If only Benchmark 2 is used, SVseq finds 5% more deletions than Pindel v1. Only using the Illumina reads from 45 individuals of the CEU population, SVseq finds thousands of deletions, a large number of the called deletions are supported by the validated deletions of the 1000 genomes project. What is more, the deletions found by SVseq all have breakpoints on 1 bp resolution, and SVseq uses only low-coverage sequence data to obtain these breakpoints. Using combined Benchmarks 1 and 2, more deletions called by SVseq and Pindel are supported than using just Benchmark 2. This may suggest that there are true deletions in the unsupported deletions but they are not in the benchmarks.

SVseq tolerates errors in mapping and allows mismatches and small indels. Thus, our method may be able to find more deletions with noisy data than Pindel v1. For example, in Figure 6, deletion P1_M_061510_1_922 is reported by the 1000 genomes project and is found by SVseq (also by Pindel v2), but *not* by Pindel v1. One possible reason is that Pindel v1 requires exact matches when mapping split reads. When there are sequencing errors, such restriction may fail to find a deletion.

Pindel v2 has added the ability of mapping split reads with mismatches and small indels. It is slower than Pindel v1 in finding larger deletions, and it also has a higher default cutoff value as 3. We run Pindel v2 using the recommended parameters to search for deletions up to 8092 bp. For comparison, SVseq is tested with cutoff values 2 and 3. Results are shown in Figure 5. Given combined Benchmarks 1 and 2 and using cutoff value 3, SVseq finds >13% of deletions than Pindel v2. If only given Benchmark 2, SVseq finds >12% of deletions than Pindel v2. When using cutoff value 2, the percentages are 28% and 25%, respectively. Using cutoff value 3, the total number of deletions found by SVseq is 12% less than Pindel v2, while using cutoff value 2, the total number of deletions found by SVseq is 27% more than Pindel v2. Using combined Benchmarks 1 and 2 and viewing the unsupported deletions called by SVseq and Pindel v2 as false positives, the accuracy of SVseq is >13% higher than Pindel v2 (in the setting of cutoff value 3 and maximum event size 8,092 bps). Figure 7 shows the findings of SVseq and Pindel v2 on a chromosome basis. Using Benchmark 2 and cutoff 3, SVseq finds more true positives than Pindel v2 on 22 chromosomes.

Running time: Comparing with Pindel v1, SVseq is generally faster. Pindel v1 does not allow inexact mapping while our method

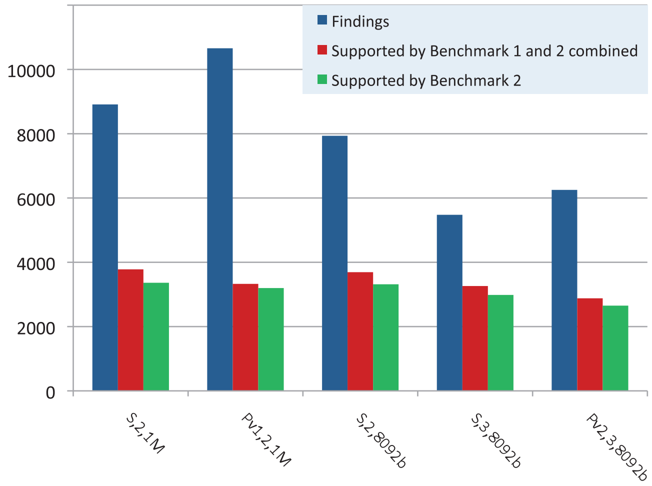
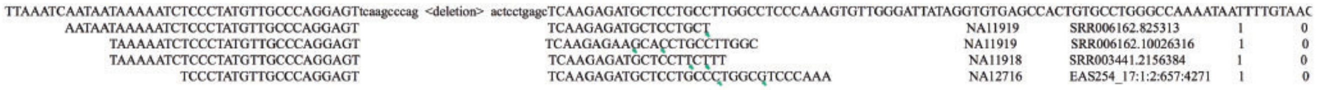


Fig. 5. Results of using low-coverage population data from the 1000 genomes project pilot 1 low-coverage data. Numbers of deletions found by SVseq, Pindel v1 and v2 using different parameters and numbers of deletions that are supported by the benchmarks are plotted in columns. S stands for SVseq. Pv1 stands for Pindel v1 and Pv2 stands for Pindel v2. Cutoff values are 2 or 3. Maximum event sizes are 1 Mbs or 8092 bp.



```
TTAAATCAATAATAAAAATCTCCCTATGTTGCCCAGGAGTtcaagcccag <deletion> actcctgagcTCAAGAGATGCTCCTGCCTTGGCCTCCCAAAGTGTTGGGATTATAGGTGTGAGCCACTGTGCCTGGGCCAAAATAATTTTGTAAC
        AATAATAAAAATCTCCCTATGTTGCCCAGGAGT                                TCAAGAGATGCTCCTGCT                         NA11919    SRR006162.825313      1    0
           TAAAAATCTCCCTATGTTGCCCAGGAGT                                  TCAAGAGAAGCACCTGCCTTGGC                    NA11919    SRR006162.10026316    1    0
           TAAAAATCTCCCTATGTTGCCCAGGAGT                                  TCAAGAGATGCTCCTTCTTT                       NA11918    SRR003441.2156384     1    0
                TCCCTATGTTGCCCAGGAGT                                     TCAAGAGATGCTCCTGCCCTGGCGTCCCAAA            NA12716    EAS254_17:1:2:657:4271 1   0
```

Fig. 6. Deletion P1_M_061510_1_922 is found by SVseq and Pindel v2, but not Pindel v1. Note that all four mapped split reads have one or more sequencing errors (pointed by arrows).

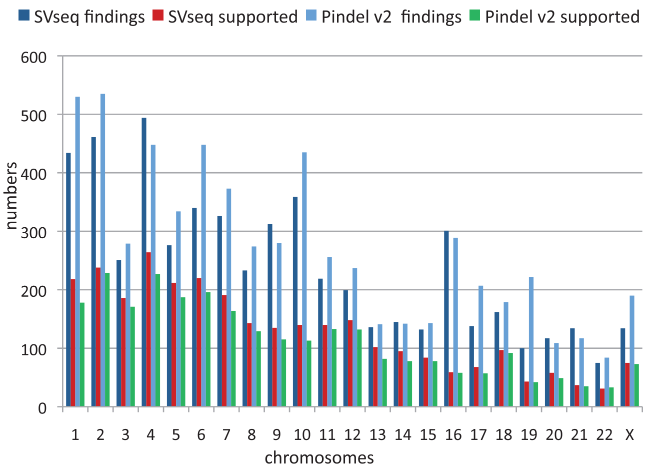


Fig. 7. Chromosome view of comparison of SVseq (cutoff 3) and Pindel v2 in finding deletions up to 8092 bp with low-coverage data using Benchmark 2. The horizontal axis is the name of chromosomes, and the vertical axis is the numbers of deletions.

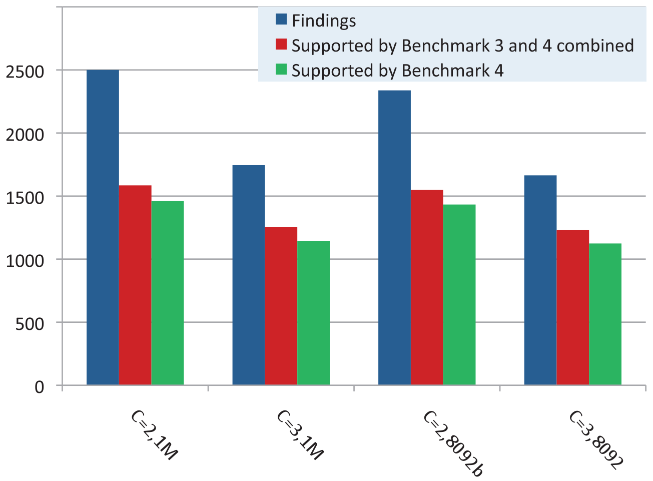


Fig. 8. Results of using high-coverage sequence data from the 1000 genomes project pilot 2 trio data. Numbers of deletions of individual NA12878 found by SVseq using different parameters and number of deletions that are supported by the benchmarks are plotted in columns. The cutoff value are 2 or 3 and the maximum even sizes are 1 Mb or 8092 bp.

allows inexact mapping (i.e. SVseq considers a larger search space). Nonetheless our method is about three times faster than Pindel v1 for processing the same amount of sequence reads. Pindel v2 allows mismatches and runs with multi-threads. But Pindel v2 is still slow even with multi-threads. When using Pindel v2 to find SVs on chromosome 1 and setting parameter of Maximum Event Size Index to 9 (corresponding to 2 071 552 bp), we run it with 20 threads on our server that has 24 cores. Pindel v2 did not finish after more than 30 h on the server. When set the parameter to 8 (corresponding to 517 888 bp) with 20 threads, Pindel v2 runs 8 h on our server (Note there is time spent trying to find other SVs). SVseq handles the same amount of data in about 3.5 h with one thread on the same machine and finds deletions up to 1 Mbps. Note that efficiency is important since the sequence data size is very large and is growing rapidly.

Result with high-coverage data: SVseq can also be used to find deletions with high-coverage data of a single individual. Figure 8 shows the numbers of deletions of the individual NA12878 found by SVseq under different parameters settings, and the numbers of deletions supported by these benchmarks. It can be seen that SVeq finds a significant number of deletions that are in the benchmarks. Using cutoff value 2 and maximum event size 1 Mb, SVseq finds 2500 deletions and 1585 (∼63%) of them are supported by the validated deletions of the 1000 genomes project. Using cutoff value 3 and maximum event size 8092 bp, SVseq finds 1664 deletions, and 1230 (about 74%) of them are supported by the validated deletions from the 1000 genomes project. In the Supplementary Table 4 of Mills *et al.* (2011), the program Pindel finds 1531 deletions for the individual NA12878 and 1253 of which are validated. The deletions finds by Pindel in the table are all less than 50 000 bp in length. Using high-coverage data, the number of validated deletions found by Pindel is similar to what is found by SVseq.

5 DISCUSSION AND CONCLUSION

Our results show that our new approach SVseq outperforms the program Pindel in terms of the number of deletions found, accuracy and running time in discovering larger deletions using low-coverage short sequence data. Higher accuracy is due to the combination of split reads mapping and discordant insert size analysis. Our enhanced split reads mapping algorithm is able to find more true positives, but more false positives may be introduced. Increasing cutoff values may reduce false positives but it may also reduce the number of true deletions found. Thus, there is a trade off in choosing the cutoff values for the split reads threshold. Discordant insert size analysis plays an important role in controlling the false positives caused by noises in performing split reads mapping. If there are no spanning pairs that match well with the size of a candidate then the candidate is likely to be false positive. In principle, we can also impose a threshold for the number of spanning paired-end reads when calling a deletion from a candidate. Our current implementation only requires a single matching spanning paired-end read for a called deletion. Our results suggest this works reasonably well.

The key challenge in finding deletions from real sequence reads is handling noise. As described before, there are many sources of noise when working with real sequence reads. As discussed in Section 3.2, DNA sequences around the breakpoints are sometimes similar, which makes split reads mapping harder to locate the exact breakpoints. Other factors such as sequencing errors and repeats also can reduce the number of deletions that can be found. In general, using more information from sequence reads tends to improve the accuracy. Finding deletions becomes more accurate with longer sequence reads and higher coverage and higher quality. There is a trade off between accuracy and cost. As the releases of the 1000 genomes project shows, most deletion finding methods for low-coverage data do not give exact breakpoints, while SVseq is able to detect exact breakpoints using low-coverage data. Although our method improves existing approaches in finding exact breakpoints of deletions from low-coverage data, we note that the deletion finding problem remains a challenge. New algorithmic development and advances in sequencing technologies (e.g. longer reads with lower error rate) are likely to improve accuracy and allow accurate and efficient finding of more structural variations in individuals and populations.

Funding: National Science Foundation (grant IIS-0953563); Tests are run on workstations supported by National Science Foundation (grant IIS-0916948).

Conflict of Interest: none declared.

REFERENCES

Albers,C. *et al*. (2011) Dindel: accurate indel calls from short-read data. *Genome Res.*, **21**, 961–973.

Alkan,C. *et al*. (2009) Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.*, **41**, 1061–1067.

Chen,k. *et al*. (2009) BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat. Methods*, **6**, 677–681.

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

Hormozdiari,F. *et al*. (2009) Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res.*, **19**, 1270–1278.

Korbel,J.O. *et al*. (2009) PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol.*, **10**, R23.

Langmead,B. *et al*. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Lee,S. *et al*. (2010) MoGUL: detecting common insertions and deletions in a population. In *Proceedings of the Annual International Conference on Computational Biology (RECOMB 2010)*, **6044**, pp. 356–368.

Li,H. *et al*. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.*, **18**, 1851–1858.

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

Medvedev,P. *et al*. (2009) Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods*, **6**, S13–S20.

Mills,R.E. *et al*. (2011) Mapping copy number variation by population-scale genome sequencing. *Nature*, **470**, 59–65.

Sebat,J. *et al*. (2004) Large-scale copy number polymorphism in the human genome. *Science*, **305**, 525–528.

Suzanne,S. *et al*. (2009) A geometric approach for classification and comparison of structural variants. *Bioinformatics* **25**, i222–i230.

The 1000 Genomes Project Consortium. (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.

Ye,K. *et al*. (2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**, 2865–2871.