Bioinformatics Advance Access published July 31, 2012

PRISM: Pair read informed split read mapping for base-pair level detection of insertion, deletion and structural variants

Yue Jiang[1,2*], Yadong Wang[1*] and Michael Brudno[2,3*]

[1]Center for Biomedical Informatics, School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang, 150001, China

[2]Department of Computer Science and The Donnelly Centre, University of Toronto, Toronto, Canada

[3]Centre for Computational Medicine, Hospital for Sick Children, Toronto, Canada

Associate Editor: Dr. Alex Bateman

ABSTRACT

Motivation: The development of high-throughput sequencing (HTS) technologies has enabled novel methods for detecting Structural Variants (SVs). Current methods are typically based on depth of coverage or pair-end mapping clusters. However, most of these only report an approximate location for each SV, rather than exact breakpoints.

Results: We have developed PRISM, a method that identifies structural variants and their precise breakpoints from whole-genome resequencing data. PRISM uses a split alignment approach informed by the mapping of paired-end reads, hence enabling breakpoint identification of multiple SV types, including arbitrary sized inversions, deletions, and tandem duplications. Comparisons to previous datasets and simulation experiments illustrate PRISM's high sensitivity, while PCR validations of PRISM results, including previously uncharacterized variants, indicate an overall precision of ~90%.

Availability: PRISM is freely available at http://compbio.cs.toronto.edu/prism.

1 INTRODUCTION

The development of high-throughput sequencing (HTS) technologies has enabled novel methods for detecting Structural Variants (SVs). Current methods, which are usually based on depth of coverage (Abyzov *et al.* 2011), pair-end mapping clusters (Chen *et al.* 2009), or a combination of these (Medvedev *et al.* 2010), have been successful in quantifying structural variation in individual genomes and populations (Mills *et al.* 2011). However, most of these only report an approximate location for each SV, rather than exact breakpoints. Accurate mapping of split reads (reads that span across a breakpoint of a SV) is evidence of the linking relationship of the segments that are adjacent in the sequenced genome (donor) but not in the reference human genome. Split-read based methods, such as Pindel (Ye *et al.* 2009), Splitread (Karakoc *et al.* 2011), and SVseq (Zhang *et al.* 2011), while able to identify these breakpoints, have been limited in their ability to identify large-scale "structural" variants. These tools take the approach of aligning the split read only in the immediate vicinity of the read's pair, and thus limit the maximum discoverable variant size. Pindel and Splitread take a user-specified maximum indel size as a parameter, while SVseq uses a fixed constant. The programs' running times and accuracies typically heavily depend on this parameter, as alignment of a small portion of the read to a large segment of the human genome is computationally intensive and error-prone. Two recent approaches (Nord *et al.* 2011, Zhang *et al.* 2012) combine the split mapping signature with additional information: the depth of coverage and discordant paired ends, respectively. However the Nord *et al.* method can only be used with capture data rather than whole genomes, and the Zhang *et al.* method (SVseq2) only handles indels, and not other structural variants. The recent CREST method (Wang *et al.* 2011) takes the alternative approach of assembling the unaligned clipped ends of reads, and mapping these to the genome. This approach, however, has limited sensitivity if the breakpoint is repetitive, or in a low complexity region, and hence more difficult to assemble.

We have developed Pair-Read Informed Split Mapping (PRISM), a method that makes use of discordant pair-end clusters to direct split read mapping. The use of discordant clusters allows us to substantially reduce the search-space for split-mapping, leading to faster runtimes and significantly higher accuracy at identifying large deletion variants. PRISM also utilizes a modification of the Needleman-Wunsch (NW) algorithm for the base-level alignment of the split read to enable high accuracy in the presence of SNPs, small indels, and sequencing errors proximal to the SV.

2 METHODS

Overview: The PRISM method relies on a thorough analysis of hanging read pairs (where only one of the two mates has a reliable mapping). PRISM uses clusters of discordant read pairs (pairs which have abnormal mapping distance, orientation, or order) as signals for the possible existence of large deletions, inversions and tandem duplications (Fig. 1A). If the unmapped read could not be mapped because it spans the breakpoint of an SV, it should be split-mappable to the two sides of the breakpoint (Fig. 1B). A detailed description of the discordant pairs and split reads can be found in the review (Medvedev *et al.* 2009). PRISM has two different strategies to select database and query sequences for SVs with and without discordant clusters. First, in order to identify small indels, each unmapped read of a hanging pair is aligned to the region approximately one insert size away from the location of the mapped one, while allowing for an unpenalized insertion or deletion in the alignment. Such indels may have been missed by the initial align-

[*]To whom correspondence should be addressed.

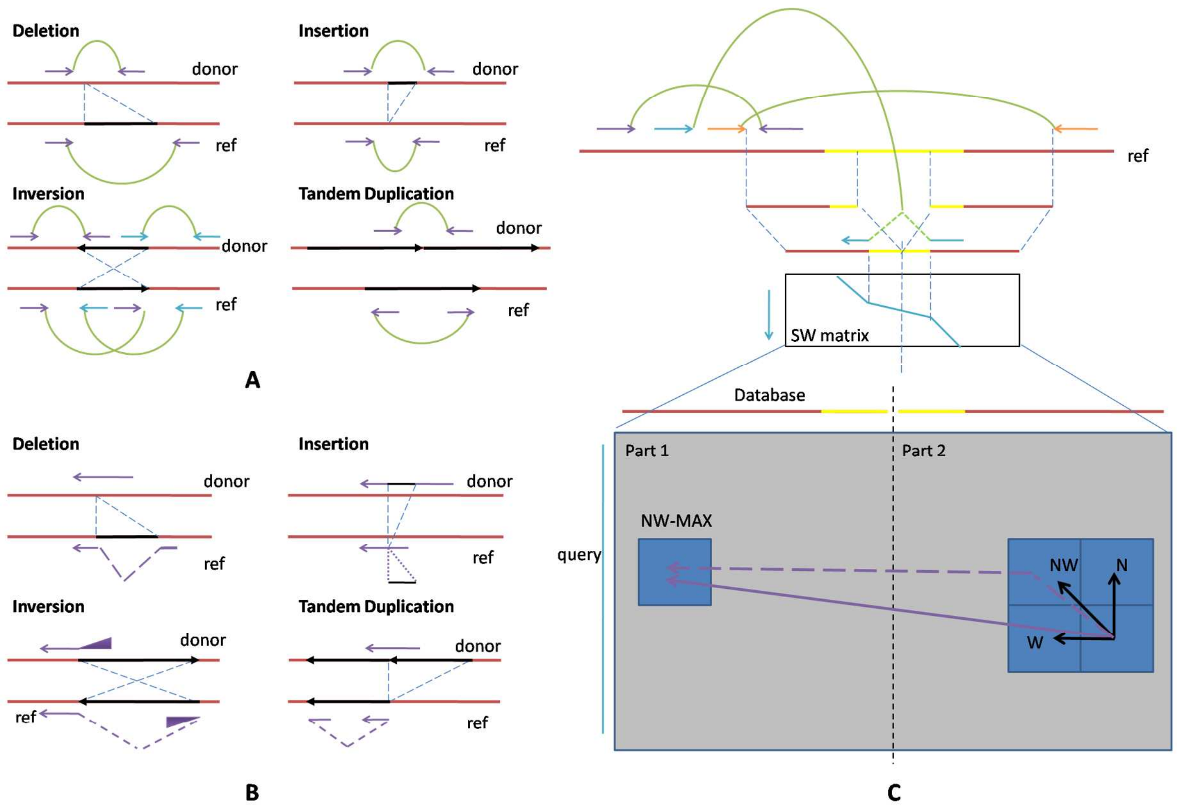

Fig. 1. **A**: Discordant pairs caused by different types of SVs. **B**: The corresponding split mapping signatures in reads that span across the SV breakpoints. **C**: The process of split mapping for a cluster supported deletion. The presence of the deleted segment (yellow line) in the reference (red line) generates a group of discordant pairs (in orange). The breakpoints of the deletion are likely to fall close to the two end points of the cluster. PRISM picks the two regions surrounding the two end points, and uses them for split-mapping of all hanging reads (in blue) in proximity of the cluster. To map the read, we modify the standard Needleman-Wunsch algorithm matrix to allow for a large, unpenalized gap that spans the breakpoint between the two regions.

ment due to a large gap penalty, or insufficient sensitivity of the mapping program. If the hanging pair is located near a cluster of discordant pairs, PRISM attempts to align the unmapped read partially to the sequence near the mapped read, and partially to the segment around the distal side of the cluster. The two segments are selected for full dynamic programming alignment, where an arbitrary "jump" is allowed between the two segments, so that the read aligns partially to one side of the breakpoint, and partially to the other. Fig. 1C demonstrates this for the case of a cluster supporting a large deletion. In the following sections we present a thorough description of PRISM.

2.1 Definitions

In this section we will define terms useful for presenting the PRISM method. All of these terms are illustrated in Fig. 2.

1 μ and σ are the mean and the standard deviation of the insert size, respectively.

2 Mapped pair: given a read pair *P* which consists of reads *r1* and *r2*, if *r1* and *r2* are both mapped *P* is a mapped pair.

3 Discordant pair: given a mapped pair *P* which consists of reads *r1* and *r2*, if a) the mapping distance between *r1* and *r2* is larger than $\mu + 3\sigma$ (we take the value $\mu + 3\sigma$ as a threshold for a discordant pair, though this can be set by the user) or b) the mapping orientation of *r1* or *r2* is different from sequencing orientation, read pair *P* is a discordant pair.

4 Concordant pair: if a mapped pair *P* is not a discordant pair, *P* is a concordant pair.

5 Anchor read and Hanging read: given a read pair *P* that consists of reads *r1* and *r2*, if *r1* is mapped, and *r2* is not mapped, mapped with one or more indels, or has an unaligned section (soft-clipping), *r1* is an anchor read and *r2* is a hanging read.

6 Hanging pair: if read pair *P* consists of an anchor read and a hanging read, *P* is a hanging pair.

7 Neighbor: given pair *P* consists of *r1* and *r2* and *Q* consists of *s1* and *s2*. Assume without loss of generality that *r1* is mapped to *pos1* and *s1* is mapped to *pos2*. If $|pos1 - pos2| \leqslant 6\sigma$, *r1* and *s1* are neighbor reads. *P* and *Q* are neighbor pairs.

8 Concordant region: given a hanging pair *P* which consists of reads *r1* and *r2*, where *r1* is the anchor read mapped to *pos1* and assume *pos2* is *pos1*+insert size. Given an interval *d* (the value depends on in what region the breakpoint of the SV is expected to be, usually about 3σ), the region [*pos2-d*, *pos2+d*] is a concordant region for *P*. We call the region [*pos2-d*, *pos2+d*] a concordant region when there is no ambiguity.

9 Discordant region: given a discordant pair *P* which consists of read *r1* and *r2*, *r1* is mapped to *pos1* and *r2* is mapped to *pos2*. Given another read pair *Q* consists of read *s1* and *s2*, *s1* is an anchor read mapped to *pos3* and *s2* is a hanging read. If

r1 is a neighbor read of *s1*, given an interval *d* (the value depends on in what region the breakpoint of the SV is expected to be, usually smaller than μ), we call region [*pos2-d*, *pos2+d*] a discordant region for *Q* from *P*. We simply call region [*pos2-d*, *pos2+d*] a discordant region when there is no ambiguity. In practice, we use clusters of discordant pairs to identify discordant regions: each cluster has two feet corresponding to the two ends of a pair.

2.2 PRISM Workflow

Running PRISM consists of 5 stages: mapping reads, preprocessing mapping result files, clustering discordant pairs, split mapping and calling SVs. PRISM typically deals with one chromosome in each execution, though it is also possible to combine the chromosomes in order to detect translocation events (see Supplementary section 4.5).

Stage 1: Mapping reads We map the reads using BWA (version 0.5.9rc1; Li *et al.* 2009a) with default settings. A series of SAM (Li *et al.* 2009b) files are generated and processed in the subsequent stages.

Stage 2: Preprocessing We identify discordant and hanging pairs from the SAM file. The discordant pairs are clustered in Stage 3. The hanging pairs are sorted by the positions of the anchor read, and are further utilized in Stage 4.

Stage 3: Clustering We identify all discordant pairs, and cluster these with the paired read clustering tool used in CNVer (Medvedev *et al.* 2010). This program utilizes a greedy algorithm that clusters together pairs with similar mapping distance and orientation. The details of this algorithm can be found in the original paper. The generated discordant clusters are used together with hanging pairs in the next stage.

Stage 4: Split Mapping This stage is the core of PRISM. PRISM scans the hanging pairs generated in Stage 2, trying several split mappings. For each hanging read PRISM tries to align it in the concordant region, while allowing for one insertion or deletion with a fixed penalty (regardless its size). Additionally, if there are discordant clusters located within the concordant region, the hanging read is also aligned in a way that allows part of it to map to the concordant region, and the other part to the discordant region. PRISM uses a modified Needleman-Wunsch algorithm for split mapping, described in section 2.3. PRISM generates read alignments in the SAM format. If the variant is an inversion or a duplication PRISM modifies the original read sequence so that it can map linearly to the genome, and stores the modification in an additional SAM field (see description of "ST Field" in README file for details).

Stage 5: Filtration & Calling SVs After the alignment, PRISM calls the SV loci from the SAM files. PRISM filters the initial list of variants based on the number of supporting reads and the alignment score. Users can set these thresholds to trade-off between sensitivity and specificity. In this manuscript we require each variant to be supported by at least 5 reads, of which at least one has a score equal to 870 (for a 100bp read the maximum score is 1000) for analysis of simulation data and of the NA18507 genome. Because of the lower coverage in the NA12878 dataset we also allow variants to be supported by only 2 reads, as long as one of them has a minimum score of 925.

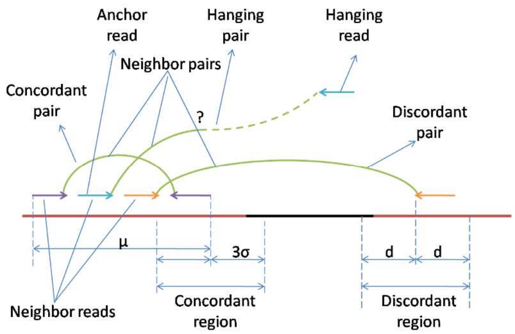

Fig.2. An illustration of the definitions. The red line is the reference with a deletion (black line). Purple arrows are two reads of a concordant pair whose mapping distance is between μ - 3σ and μ + 3σ. Orange arrows are two reads of a discordant pair whose mapping distance is larger than μ + 3σ. The left blue arrow is an anchored read which is aligned to the reference and the right one is a hanging read which is not aligned. The left purple, blue and orange arrows are neighbor reads. The pairs in these three colors are neighbor pairs. The concordant and discordant regions are also shown.

2.3 Algorithm for split mapping

2.3.1 Modified Needleman-Wunsch algorithm

To perform alignment of the split-mapped reads to the reference genome we use a modified Needleman-Wunsch (NW) algorithm. For deletions, the query is the read sequence and the database are two segments of the reference, where we expect the beginning and the end to map (the selection of these two segments, called *region1* and *region2*, is described in *2.3.2*). Note that the two regions may be the same. We build two dynamic programming matrices for read×*region1* (*matrix1*) and read×*region2* (*matrix2*). Each cell in *matrix1* is computed as in the traditional NW (with affine gap penalties). For *matrix2* we calculate one additional recurrence, which uses the max score of all cells of the immediately previous line in *matrix1* (Fig. 1C). We use this score to allow the alignment to "jump" from *matrix2* to *matrix1* by introducing one large gap that corresponds to the split in the read. The penalty for this gap is a constant not related to its length. For insertions the algorithm is similar, except we align a single region of the genome to two copies of the read. The algorithm for alignment is identical, except that the "jump" is now between two copies of the read.

The modified NW matrix for deletions is built using the following recurrences (also see illustration in Supplementary Figure 1).

$$M(0,j) = I_{db}(0,j) = I_{qr}(0,j) = 0 \qquad 0 \le j \le m_1 + m_2 + 1$$

$$I_{qr}(i,0) = I_{qr}(i,m_1+1) = -gap_{open} - i \times gap_{ext} \qquad 1 \le i \le l$$

$$I_{db}(i,0) = M(i,0) = I_{db}(i,m_1+1) = M(i,m_1+1) = -\infty 1 \le i \le l$$

$$M(i,j) = \max\begin{cases} M(i-1,j-1) + w(qr[i],db[j]) & \text{a} \\ I_{db}(i-1,j-1) + w(qr[i],db[j]) & \text{b} \\ I_{qr}(i-1,j-1) + w(qr[i],db[j]) & \text{c} \\ M(i-1,j_{max}) + w(qr[i],db[j]) - jump_{qr} & \text{d} \end{cases}$$

$$\text{for a, b and c: } 1 \le i \le l, 1 \le j \le m_1 + m_2 + 1, j \ne m_1 + 1$$
$$\text{for d: } 1 \le i \le l,\ m_1 + 1 < j \le m_1 + m_2 + 1$$

$$I_{db}(i,j) = \max\begin{cases} M(i,j-1) - gap_{open} \\ I_{db}(i,j-1) - gap_{ext} \end{cases} \qquad \begin{matrix} 1 \le i \le l, j \ne m_1 + 1, \\ 1 \le j \le m_1 + m_2 + 1 \end{matrix}$$

$$I_{db}(i,j) = \max\begin{cases} M(i-1,j) - gap_{open} \\ I_{qr}(i-1,j) - gap_{ext} \end{cases} \qquad \begin{matrix} 1 \le i \le l, j \ne m_1 + 1, \\ 1 \le j \le m_1 + m_2 + 1 \end{matrix}$$

where:

db is built by two fragments of reference sequence as described in section *2.3.2*. The lengths of these two fragments are m_1 and m_2.

qr is the read sequence. *l* is the length of *qr*.

$M(i, j)$, $I_{qr}(i, j)$ and $I_{db}(i, j)$ are the maximum similarity scores given that $qr[i]$ is aligned to $db[j]$ (match/mismatch), $qr[i]$ to a gap (insertion), or $db[j]$ to a gap (deletion), respectively.

gap_{open} is the penalty for opening a gap in the alignment.

gap_{ext} is the penalty for extending a gap in the alignment.

$w(a, b)$ is a scoring function which is positive when $a = b$ and negative otherwise.

$jump_{qr}$ is penalty for a jump from *matrix2* to *matrix1* (from cell Cur to cell NW-MAX, see Fig. 1C).

$j_{\max}$ is the *db* index of cell NW-MAX, i.e. if cell Cur's coordinate is (i, j) cell NW-MAX's should be $(i\text{-}1, j_{\max})$ (see Fig. 1C).

Note that the jump score is not calculated for I_{qr} and I_{db} because a deletion cannot be followed directly by an insertion, while a deletion followed directly by a jump can be included in the jump. The same algorithm is used for split mapping of inversions and tandem duplications as deletions, as illustrated in section *2.3.2*. The algorithm for insertions is similar, except that the read, rather than the reference segment is duplicated, and is presented in the Supplement (section 2). Finally, to optimize the performance of the dynamic programming step we utilize an anchored alignment approach, which is also presented in the Supplement (section 3).

2.3.2 Selection of query and database for modified Needleman-Wunsch matrix For SVs of different type and size PRISM has different strategies to select query and database sequences for the modified Needleman-Wunsch algorithm. The database can consist of the sequence in the concordant region, the discordant region, or their reverse-complements. The query is the read sequence or the doubled read sequence (for insertions).

Split mapping within the concordant region For each anchor read the hanging read may be unmapped due to spanning across a small indel that is not big enough to generate discordant pairs. In this case the indel will be within the concordant region. This case is illustrated in Fig. 3A (for deletions) and 3B (for insertions). In the deletion case the two regions to which the read is mapped are identical. Similarly, in the case of an insertion the read is duplicated and aligned to a single copy of the genomic region. We attempt both the insertion and deletion split alignments for each hanging read.

Split mapping between the concordant and the discordant regions For each hanging pair, if there are neighbor reads that are members of a discordant cluster, the hanging read may be unmapped because it spans across the breakpoints of a large deletion, an inversion, or a duplication (depending on the type of the cluster). The cluster allows us to identify the discordant region, and the difference between the split mappings of these three types of SVs is only in the selection of the discordant region, as the alignment only needs to allow for a large "jump" from the concordant to the discordant region. Fig. 3C-F illustrate the selection of the databases and queries for these SVs.

2.4 Simulation Dataset

To estimate both the sensitivity and specificity of PRISM on a realistic dataset with a known ground truth we implanted known human indels (Mills *et al*. 2006) into chromosome 1 of the human genome and simulated 100bp paired reads with Gaussian-distributed inserts (500bp mean, 30bp s.d.). The sequencing error model followed real illumina data (1% overall error rate).

3 RESULTS

To evaluate our method, we conducted simulation experiments and analyzed two genomes recently sequenced by Illumina.

3.1 Simulation Results

We ran PRISM, Pindel, SVseq, Splitread, BreakDancer (Chen *et al.* 2009), CNVnator (Abyzov *et al.* 2011) and CREST on our simulation dataset, with the results summarized in Supplementary Table 1. PRISM achieved recall ≥ 95% for small variants (insertions and deletions 1-10bp and 11-50bp), with precision ≥ 94%. For medium-size deletions (51-100bp) the recall was 63% with precision 77%. For deletions of 101-1000bp PRISM achieved recall of 80% at 95% precision, while for deletions >1kb the recall was 80% and precision was 95%. Pindel and SVseq performed the best of the remaining tools, followed by Splitread and CREST. CREST is designed for somatic, rather than germline variants, so we decided to exclude it from the comparisons on real genomes. Splitread and SVseq were also not run on whole genomes due to inappropriateness of the method for our experiments, or due to technical issues, as discussed in the Supplement.

3.2 Results on Real Data

To evaluate PRISM on real data we used two recent whole-genome paired-end datasets generated Illumina Genome Analyzer II: Yoruban HapMap individual NA18507 (NCBI SRA id: SRA010896) with 47x sequencing coverage, insert size 500bp, duplicate reads percent 7.4% and the CEU HapMap individual NA12878 (SRA id: ERX012406) sequenced at 15x coverage, 300bp insert size, duplicate reads percent 3.6%. We used the hg18 reference genome. The characterization of these genomes in multiple previous studies (e.g. Mills *et al.* 2011; Kidd *et al.* 2008; McCarroll *et al.* 2007; McKernan *et al*. 2009) allows us to compare PRISM results to both predicted and validated variants, as well as to identify novel variants which could not be discovered with previous methods. We compared PRISM with the latest (unpublished) version of Pindel (2.0) that is able to identify tandem duplications and inversions, as well as small insertions and large deletions. In Table 1 we present a comparison to variants that have been detected using Sanger sequencing, PCR or array studies, as well as the study (McKernan *et al.* 2009) for inversions. Comparison to additional high-throughput sequencing-based analyses is in Supplementary Table 3. To evaluate the various tools we consider a previously reported set of variants as a gold standard, and compute precision – the fraction of true positives over all predictions, and recall – the fraction of true positives over all variants in the dataset. Note that because the gold standard datasets are not complete we do not expect the precision to be close to 1 for most datasets. In particular the datasets (Kidd *et al.* 2008; McCarroll *et al.* 2007), which did not utilize high-coverage whole-genome sequencing, are likely missing true variants. Similarly, only a fraction of all variants are identifiable through a split-read approach. A thorough discussion of the causes of false-negatives is presented in the Supplement.

For the NA18507 genome PRISM detected 784319 indels of 1-100bp, of which 145944 (19%) were previously identified in Kidd *et al.* (2008). PRISM was able to identify 65% of the variants known from the study (Table 1). This compares favorably with Pindel, which identified fewer total variants, of which a smaller percentage were previously known. We also compared PRISM with GATK (McKenna *et al.* 2010), a standard method for identi-

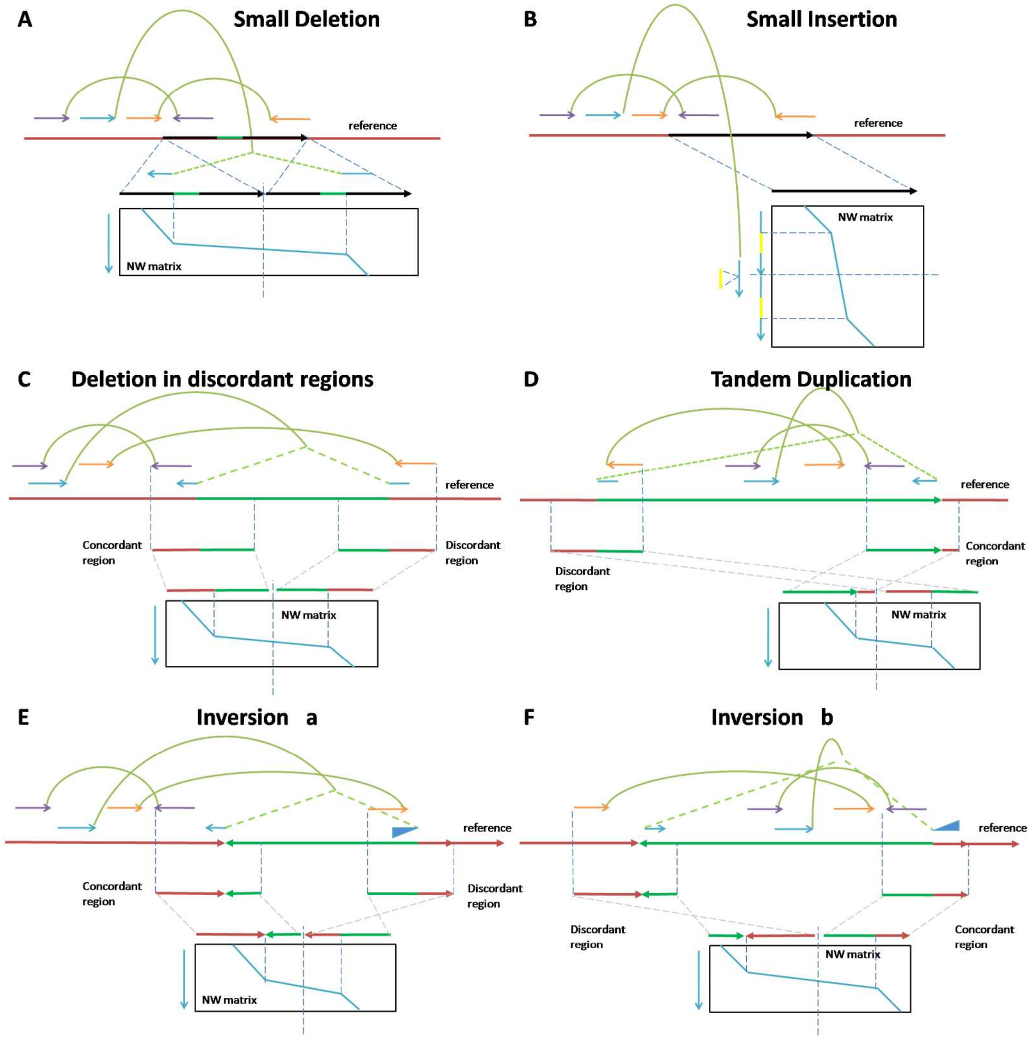

Fig. 3. Illustration of alignment of small indels (not supported by clusters), long deletions, inversions, and duplications (supported by clusters).**A**: An illustration of alignment with a small deletion (gap in the read), within the concordant region. The concordant region (black arrow) of the anchor read (blue arrow) with a deletion (green line) is duplicated to be the database of modified NW algorithm. The query is the hanging read. **B**: An illustration of alignment with a small insertion (gap in the reference) within the concordant region. The concordant region (black arrow) of the anchor read (blue arrow) is the database of modified NW algorithm. The hanging read with the insertion (yellow line) is duplicated to be the query. The split mappings we expect to find are shown in both sub-figures (blue line). In C-F purple arrows are concordant pairs and orange arrows are discordant pairs. Hanging pairs are in blue. All these SVs can share the same split mapping algorithm. The only difference is the selection of database sequence. **C**: Deletion (green line). The database is the connection of the concordant region and the discordant region. **D**: Duplication (green arrow). The database is the connection of the discordant region and the concordant region. **E**: Inversion (green arrow), the anchor read is outside of the inversion. The database is the connection of the concordant region and the reverse complement of the discordant region. **F**: Inversion (green arrow), the anchor read is inside of the inversion. The database is the connection of the reverse complement of the discordant region and the concordant region.

fying small indels, and found that the two methods were comparable for indels 1-20bp, PRISM showed better sensitivity at detecting larger variants: GATK did not detect any indels larger than 50bp. We also analyzed the CEU NA12878 genome, where we identified an overall smaller number of indels (592373 1-100bp, 2042 101-5000bp, 160 >5kb). PRISM showed a significant advantage at identifying larger variants due to its use of paired-end analysis to identify likely locations of breakpoints. Comparing to several datasets of validated deletions (Mills *et al.* 2011) PRISM achieved significantly higher recall and somewhat lower precision for variants 50-200bp, and higher precision and recall for deletions >200bp (Fig. 4D and Supplementary Table 6). When comparing to large deletions (>5kb) identified via Sanger sequencing (Kidd *et al.* 2008) in the NA18507 individual the same trend held: despite reporting 82% fewer variants than Pindel, PRISM identified a higher number of known deletions, thus achieving better precision and recall. The results of PRISM were nearly on par with BreakDancer, a tool which only utilizes pair-end data, and cannot identify the precise breakpoints of variants.

We also used the NA18507 genome to analyze performance on inversions and duplications. For these variants breakpoints are often in repetitive regions, making them especially difficult to identify with the split-read approach. PRISM identified 172 inversion breakpoints in this genome. Of these 36 (21%) were among the 83 inversion variants previously identified with a paired-end approach (McKernan *et al.* 2009). We also found 407 split reads

Table 1. Comparison of PRISM, Pindel, GATK and BreakDancer (BreakD.) on the NA18507 dataset. Known SVs catalogued in several studies ([1] Kidd *et al.* 2008. [2] McKernan *et al.* 2009. [3] McCarroll *et al.* 2007.) are separated into several groups: small indels[1] (1-20, 21-50, 51-100bp), large deletions[1] (>5000bp), duplications[2] and inversions[3]. The Total line indicates the number of variants of a given type identified by each method, as well as the number present in each dataset. Comparison for deletions[3] (> 100bp) is in Supplementary Table 3.

	Indels 1-20bp			Indels 21-50bp			Indels 51-100bp	
Known	223196[1]			2727[1]			186[1]	
SV Caller	Pindel	GATK	PRISM	Pindel	GATK	PRISM	Pindel	PRISM
Observed	669781	781066	772242	11266	2735	10361	1387	1716
Found	124871	145055	144851	995	423	1026	49	67
Recall	55.9%	64.9%	64.9%	36.5%	15.5%	37.6%	26.3%	36.0%
Precision	18.6%	18.6%	18.8%	8.8%	15.5%	9.9%	3.5%	3.9%
	Deletions >5000bp			Inversions			Duplications	
Known	151[1]			83[2]			26[3]	
SV Caller	Pindel	BreakD.	PRISM	Pindel	BreakD.	PRISM	Pindel	PRISM
Observed	1997	362	351	193	343	172	427	407
Found	42	55	45	33	54	36	3	7
Recall	27.8%	36.4%	29.8%	39.7%	65.1%	43.4%	11.54%	26.9%
Precision	2.1%	15.2%	12.8%	17.1%	15.7%	20.9%	0.7%	1.7%

displaying the duplication signature, and compared these with the copy counts previously reported by McCarroll *et al.* (2007), who used array-CGH to identify genomic regions with a significant difference in intensity of 270 individuals (90 Yoruban and 180 Eurasian). Following the method of Medvedev *et al.* (2010) we identified 26 genomic regions in which NA18507 likely has higher copy counts than the reference. We found 7/407 of our duplication calls agree with these 26 regions, compared to 3/427 duplications reported by Pindel.

Finally, to directly measure PRISM's false positive rate and to validate the novel variants identified by our method, we conducted PCR experiments on 58 variants. These were chosen from various size ranges, with approximately 10 variants selected from each of the following categories: Insertions 10-50bp, Deletions 10-50, 50-100, 100-1000, 1000-5000 and >5000bp. To allow for the evaluation of the overall accuracy of PRISM we randomly selected 5 variants from each category. To characterize PRISM's ability to identify novel genomic variants inaccessible to previous tools we attempted to select 5 additional variants from within each size set which did not intersect with previously known indels or structural variants (from the Database of Genomic Variants; http://projects.tcag.ca/variation/) and which were not reported by Pindel. The novel variants were also visually inspected (see Fig. 4A-C) to exclude regions with multiple overlapping events. Overall, in 2/58 cases the primers did not work, and in 4/58 neither of the two alleles matched in size either the reference genome, or the predicted variant, likely indicating mis-priming or a mis-assembly in the reference genome. Of the remaining experiments, 47/52 (90%) validated PRISM predictions, while 2/52 (4%) indicated no variation. The final 3/52 (6%) revealed a variant at the genomic region that was different from the one predicted by PRISM (See Supplementary Table 9). Of the randomly chosen variants 30/33 (91%) were validated, with 2/33 (6%) not validating, and 1/33 (3%) showing alternate variants. Similarly, 17/19 (89%) selected novel variants validated, with the remaining two regions showing alternate genomic variation. We also indirectly measured PRISM's accuracy by analyzing the fraction and length of indel variants within coding regions of the genome. The fractions of such indels reported by the three tools are very close (373/784319=0.05% for PRISM, 364/682434=0.05% for Pindel and 373/783801=0.05% for GATK), while a higher fraction of PRISM coding indels had lengths that are a multiple of three (155/373=41.6% for PRISM, 143/364=39.3% for Pindel and 144/373=38.6% for GATK).

4 CONCLUSION

PRISM has several advantages over previous methods for detection of structural variants from high-throughput sequencing data, including combining paired read analysis and split-mapping for detection of the exact breakpoints for variants of arbitrary size. The sensitive alignment algorithm used in PRISM accurately identifies breakpoints even proximal to other variants or sequencing errors. One such example is shown in Fig. 4, where a large deletion predicted by PRISM (and validated by PCR) is immediately followed by a small deletion, making the alignment of reads that span the deletion especially challenging. The PCR validations confirm not only the overall high accuracy of PRISM results, but also its ability to identify novel variants inaccessible to previous methods.

ACKNOWLEDGEMENTS

We are grateful to The Centre for Applied Genomics (Toronto) for assisting with the wet lab validation experiments.

Funding: This work was supported by a CIHR grant to MB and a China Scholarship Council Fellowship to YJ.

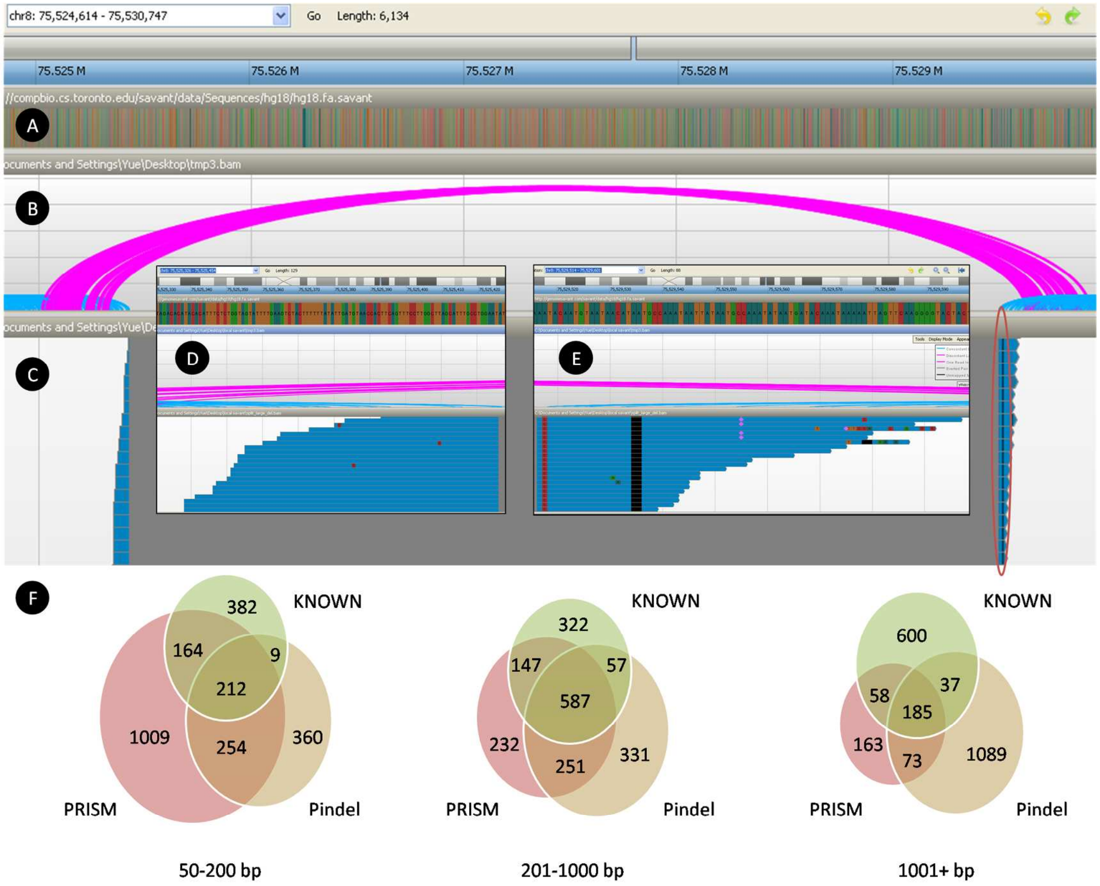

Fig. 4. Visualization of a PRISM-predicted variant in the Savant Genome Browser (Fiume *et al.* 2010) and a comparison of deletions for the NA12878 individual. The variant is the 4090bp deletion on chromosome 8 validated by PCR. The three tracks show **A**: The reference genome; **B**: The aligned read pairs, visualized as arcs. The height of the arc is proportional to the distance between the reads, with blue arcs indicating concordant pairs, while purple arcs are discordant pairs indicating the presence of a deletion; **C**: the PRISM split mapping track. Most of the reads in track C contain the same deletion (grey region), which is consistent with the discordant pairs in track B (purple arcs). Furthermore, there is a second small deletion spanned by most of the reads supporting this long deletion (black column within the red oval in track C). This additional deletion makes aligning these reads especially challenging. **D** and **E**: Zoom in views of both sides of the deletion. **F**: Venn Diagrams illustrating the concordance of PRISM and Pindel deletion calls of various lengths in the NA12878 individual with variants annotated at nucleotide resolution by the Yale group (YL_SR) based on 454 read data and validated with PCR (Mills *et al.* 2011). Two calls are considered to overlap if they have exactly the same size and their locations deviate by < 100bp.

REFERENCES

Abyzov A. *et al.* (2011) CNVnator: an approach to discover, genotype and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res.*, **21**, 974-984.

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

Fiume M. *et al.* (2010) Savant: genome browser for high-throughput sequencing data. *Bioinformatics*. **26**(16), 1938-1944.

Kidd, J. M. *et al.* (2008) Mapping and sequencing of structural variation from eight human genomes. *Nature*, **453**, 56–64.

Karakoc E. *et al.* (2011) Detection of structural variants and indels within exome data. *Nat. Methods*, **9**(2), 176-8.

Levy S. *et al.* (2007) The diploid genome sequence of an individual human(2007) *PLoS Biol.*, **5**(10), e254.

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

Li, H. *et al.* (2009b) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

McCarroll, S.A. *et al.* (2007) Copy-number variation and association studies of human disease. *Nat. Genet.*, **40**, 1166–74.

McKernan, K.J. *et al.* (2009) Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Res.*, **19**, 1527-1541.

McKenna, A. *et al*. (2010) The Genome Analysis Toolkit: a Map Reduce framework for analyzing next-generation DNA sequencing data. *Genome Res*., **20**, 1297-1303.

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

Medvedev, P. *et al.* (2010) Detecting copy number variation with mated short reads. *Genome Res.*, **20**, 1613 - 1622.

Mills, R.E. *et al.* (2006) An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Res*, **16**, 1182–1190.

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

Nord, A. *et al.* (2011) Accurate and exact CNV identification from targeted high-throughput sequence data. *BMC Genomics* **12**(1), 184.

Wang, J. *et al.* (2011) CREST maps somatic structural variation in cancer genomes with base-pair resolution. *Nat. Methods*, **8**(8), 652-4.

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.

Zhang J. *et al.* (2011) SVseq: an approach for detecting exact breakpoints of deletions with low-coverage sequence data. *Bioinformatics*, **27**(23), 3228-3234.

Zhang, J. *et al.* (2012) An improved approach for accurate and efficient calling of structural variations with low-coverage sequence data. *BMC Bioinformatics.* **13**(Suppl 6), S6.