SOAPindel: Efficient identification of indels from short paired reads

Shengting Li,[1,2] Ruiqiang Li,[1,7] Heng Li,[3] Jianliang Lu,[1] Yingrui Li,[1] Lars Bolund,[1,4] Mikkel H. Schierup,[2,8] and Jun Wang[1,5,6,8]

[1]BGI Shenzhen, Shenzhen 518000, China; [2]Bioinformatics Research Centre, Aarhus University, DK 8000 Aarhus C, Denmark; [3]Broad Institute, Cambridge, Massachusetts 02142, USA; [4]Human Genetics, Aarhus University, DK 8000 Aarhus C, Denmark; [5]The Novo Nordisk Foundation Center for Basic Metabolic Research, University of Copenhagen, 2200 Copenhagen, Denmark; [6]Department of Biology, University of Copenhagen, 2200 Copenhagen, Denmark

We present a new approach to indel calling that explicitly exploits that indel differences between a reference and a sequenced sample make the mapping of reads less efficient. We assign all unmapped reads with a mapped partner to their expected genomic positions and then perform extensive de novo assembly on the regions with many unmapped reads to resolve homozygous, heterozygous, and complex indels by exhaustive traversal of the de Bruijn graph. The method is implemented in the software SOAPindel and provides a list of candidate indels with quality scores. We compare SOAPindel to Dindel, Pindel, and GATK on simulated data and find similar or better performance for short indels (<10 bp) and higher sensitivity and specificity for long indels. A validation experiment suggests that SOAPindel has a false-positive rate of ~10% for long indels (>5 bp), while still providing many more candidate indels than other approaches.

[Supplemental material is available for this article.]

Calling indels from the mapping of short paired-end sequences to a reference genome is much more challenging than SNP calling because the indel by itself interferes with accurate mapping and therefore indels up to a few base pairs in size are allowed in the most popular mapping approaches (Li et al. 2008; Li and Durbin 2009; Li et al. 2009). The most powerful indel calling approach would be to perform de novo assembly of each genome and identify indels by alignment of genomes. However, this is computationally daunting and requires very high sequencing coverage. Therefore, local approaches offer more promise. Recent approaches exploit the paired-end information to perform local realignment of poorly mapped pairs, thus allowing for longer indels (Ye et al. 2009; Homer and Nelson 2010; McKenna et al. 2010; Albers et al. 2011). One such approach, Dindel, maps reads to a set of candidate haplotypes obtained from mapping or from external information. It uses a probabilistic framework that naturally integrates various sources of sequencing errors and was found to have high specificity for identification of indels of sizes up to half the read length (Albers et al. 2011). Deletions longer than that can be called using split read approaches such as implemented in Pindel (Ye et al. 2009). Long insertions remain problematic because short reads will not span them and a certain amount of de novo assembly is required.

Our approach, implemented in SOAPindel, performs full local de novo assembly of regions where reads appear to map poorly as indicated by an excess of paired-end reads where only one of the mates maps. The idea is to collect all unmapped reads at their expected genomic positions, then perform a local assembly of the regions with a high density of such reads and finally align these assemblies to the reference. A related idea has recently been published by Carnevali et al. (2012), but their approach is designed for a different sequencing method, and software is not available for comparison.

While conceptually simple, our approach is sensitive to various sources of errors, e.g., false mate pairs, sequencing errors, nonunique mapping, and repetitive sequences. We deal with these complexities by examining all the paths in an extended de Bruijn graph (Zerbino and Birney 2008) and choose those that anchor at some points on the reference genome sequence. In this way, we can detect heterozygous indels as two different paths in the de Bruijn graph and, in principle, call multiallelic indels in polyploid samples or pools of individuals. Unlike, e.g., Pindel, the approach treats insertions and deletions in the same way and has no constraint on indel length other than that determined by the local assembly.

We explore the specificity and the sensitivity of SOAPindel by extensive simulations based on the human genome and by indel calling on one of the high-coverage samples of the 1000 Genomes Project. We estimate a low false-positive rate of the de novo indel calls by direct Sanger resequencing as well as from simulated reads data based on the Venter genome and the chimpanzee genome, mapped against the reference genome. We benchmark SOAPindel against Dindel, Pindel, and GATK, and it shows similar or better specificity and sensitivity for short indels and much higher sensitivity for long indels.

Results

The SOAPindel algorithm is outlined in Figure 1 (for details, see Methods). The performance of SOAPindel on indels simulated from the Venter-hg19 alignments is compared with Dindel, GATK, and Pindel in Figure 2. Three quality thresholds were used for SOAPindel for illustration. Lowering the quality score to Q1, the false-negative rate is very low, but this is at the expense of a very high number of false-positive indels. For Q10, SOAPindel has an acceptable false-positive rate similar to Dindel, while still being

[7]Present address: Biodynamic Optical Imaging Center, and College of Life Sciences, Peking University, Beijing 100871, China.
[8]Corresponding authors
E-mail mheide@birc.au.dk
E-mail wangj@genomics.cn

Article published online before print. Article, supplemental material, and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.132480.111.

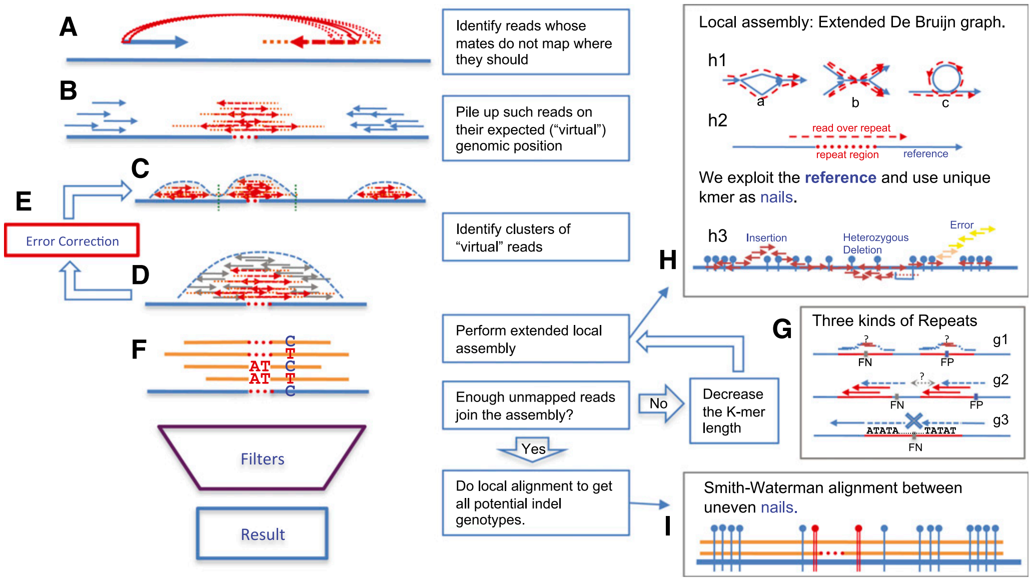
A
Identify reads whose mates do not map where they should
B
Pile up such reads on their expected ("virtual") genomic position
C
E
Error Correction
D
Identify clusters of "virtual" reads
F
AT
AT
C
T
C
T
C
Filters
Result
Perform extended local assembly
Enough unmapped reads join the assembly?
No
Yes
Decrease the K-mer length
Do local alignment to get all potential indel genotypes.
Local assembly: Extended De Bruijn graph.
h1
a
b
c
h2
read over repeat
repeat region
reference
We exploit the reference and use unique kmer as nails.
h3
Insertion
Heterozygous Deletion
Error
H
G
Three kinds of Repeats
g1
FN
FP
g2
FN
FP
g3
ATATA
TATAT
FN
Smith-Waterman alignment between uneven nails.
I

Figure 1. Outline of the SOAPindel algorithm. (*A*) Identify nonmapping mates. (*B*) Pile up unmapped reads. (*C*) Identify clusters of unmapped reads. (*D*) Cut out clusters and add mapped reads. (*E*) Error correct clusters. (*F*) Find candidate indels by de novo assembly. (*G*) Treatment of repeats. (*H*) Extended local assembly with nails. (*I*) Local alignment between nails. For details, see Methods.

more sensitive than Dindel and GATK. We recommend the use of Q10 based on a false-positive rate on simulated data of about the 1% expected, while missing only ~20% of real indels (see Supplemental Fig. S2). Pindel is more sensitive but has a very high false-positive rate. These findings are true over the complete spectrum of indel sizes, but the greatest difference is being found for indels >30 bp in size. There is little difference in the pattern when indels are divided into insertions and deletions relative to hg19 (see Supplemental Fig. S3). The cause of the very high false-negative rate is investigated in more detail in the Supplemental Material, where it is shown that low complexity sequence and high density of variation in close proximity contribute to the lack of sensitivity (Supplemental Fig. S4).

In Figure 3, the effects of indel length, sequence coverage, and read length on the sensitivity and specificity of the different methods are summarized and divided into insertions and deletions. Sequence coverage is a major determinant of the sensitivity, whereas read length is mainly important for the size limit of indel detection for the alignment-based approaches, but not for SOAPindel. For indels of size 100, Pindel can call deletions but not insertions, and SOAPindel is the only approach that maintains both high specificity and sensitivity in this case. False-negative rates for heterozygous and homozygous indels separately are shown in Supplemental Figure S5. Heterozygotes are more difficult to detect than homozygotes, because detecting them involves assembly of two different paths, which is particularly difficult when the sequencing depth is low.

Indel calling results using SOAPindel (Q10) on short read data from the NA18507 individual are shown for the whole genome in Figure 4A. A total of 1,018,647 Q10 indels were called. More deletions than insertions were found, which may partly be due to the fact that hg19 is assembled from many individuals with a bias toward using the longer variant of indels among these individuals in the assembly. Two notable peaks in deletion sizes are seen, and we manually investigated the peak around 170 bp. There are 322 Q10 indels around 170–172 bp, and 296 (92%) of them are around the centromere and share sequence similarity with the alpha satellite. Figure 4B compares indel calling on chromosome 7 with other methods (using their default settings). For all size classes, SOAPindel reports more indels, while GATK and Dindel find no indels, and Pindel finds no insertions larger than 19 bp in this data set, which is based on 40-bp read length.

We used different approaches to investigate the false-positive rate of SOAPindel in this data set. First, we attempted experimental validation of 30 Q10 indel positions (size >5 bp) detected by SOAPindel in the same two ENCODE regions, ENm010 (hg19: chr7:26730761–27230760) and ENm013 (hg19:chr7:89428340–90542763), previously used to validate Dindel. Twenty-six of these were confirmed, including a 93-bp insertion (Table 1). Thus, the specificity is ~86% for indels >5 bp. (For further details, see Supplemental Table S1.) Second, we used comparison to the alignment of chimpanzee (panTro3) with hg19 for validation. An indel vari-

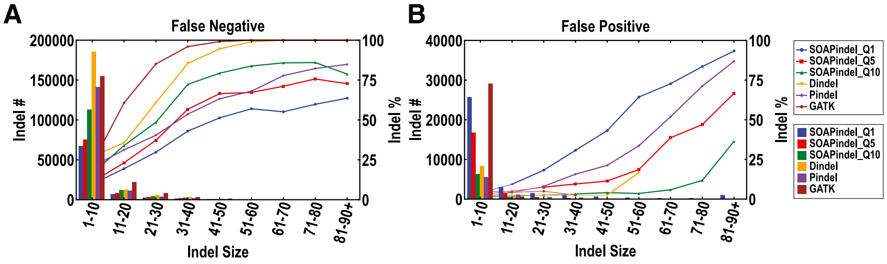

Figure 2. False-negative (FN) (*A*) and false-positive (FP) (*B*) rates of indels called by SOAPindel, Dindel, Pindel, and GATK on data simulated based on hg19. The read length is 100 bp, and the coverage is 20×, and SNP and indel variation are from the empirical differences of the Venter and hg19 genomes. The numbers of FP and FN indels refer to histograms, whereas the lines correspond to the percentage of indels being either FN or FP (secondary *y*-axis).

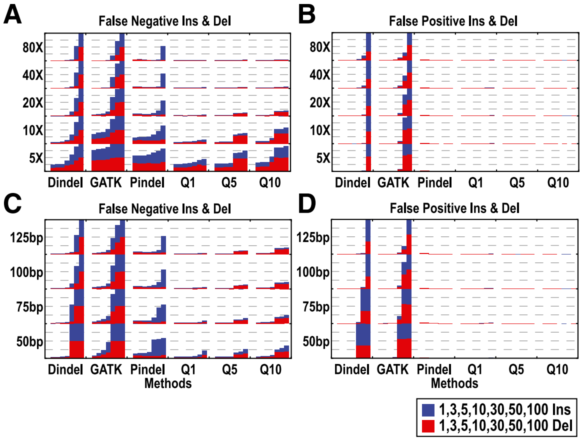

Figure 3. The effect of indel size, coverage, and read length on the false-negative and false-positive rates of SOAPindel, Dindel, Pindel, and GATK on simulated data based on hg19, chromosome 22. (*A*,*B*) FN and FP percentage for read length of 100 bp and coverage of 5/10/20/40/80×, with fixed indel sizes of 1/3/5/10/30/50/100. Every column is split into two parts: % of insertion and % of deletion. (*C*,*D*) FN and FP percentage for read lengths of 50/75/100/125 bp and coverage of 20×, with the same fixed indel sizes and also split into insertion and deletion.

ant found in NA18507 was considered to be validated if the chimpanzee sequence against hg19 had an indel of the same size that could be slid into the exact same position without increasing the number of mismatches. This can only validate at most 50% of the real indels because the chimpanzee is expected to be similar to the hg19 variant in half of all cases. Furthermore, the alignment of hg19 and panTro3 is incomplete, in particular, in repetitive and indel-rich regions. About 20% of the indels (Q10) called by SOAPindel are confirmed by this approach (Table 2). The confirmation rate is almost independent of size (Supplemental Fig. S6) up to 100 bp, which is the threshold for the human–chimpanzee alignment to be broken. Table 2 also shows a comparison with the indel calls of Dindel for chromosome 7. SOAPindel finds more indels that can be confirmed by the human–chimpanzee alignment, suggesting that SOAPindel has a higher sensitivity than Dindel. However, the specificity for short indels seems lower since a smaller percentage of SOAPindel calls than Dindel calls are validated by the alignment. The exception is for indels >10 bp, where SOAPindel calls many more indels than Dindel and has a confirmation rate of 23% compared with 7% for Dindel (Table 2). Finally, Table 3 shows the distribution of indel calls over introns, exons, and intergenic regions. Coding indels shows an enrichment of in-frame indels; 50.0% of all indel calls (and 82% of all indel calls >1 bp) are multiples of three. Mills et al. (2011) and Mullaney et al. (2010) reported that in-frame indels should constitute 50%–60% of all indels. Our result is in the lower range of these numbers, suggesting some false-positive indel calls for very short indels, in particular.

Discussion

SOAPindel explicitly performs extensive local de novo assembly in regions where indels are expected, using both reads that map to these regions and unmapped reads that should map to the regions based on the mapping position of the mate. Recently, Carnevali et al. (2012) published an indel calling approach that also uses local reassembly in an iterative fashion for the Complete Genomics sequencing approach, but it is not clear how this would work on Illumina sequence data, and software is not available for comparison to SOAPindel predictions.

Local assembly as compared with realignment, which is used by widely used indel detection tools such as GATK and Dindel, is expected to be most powerful for long insertions and deletions, and this is what we have found using both simulated data and analyses of short read data from one human genome.

All indel detection methods based on reference sequence mapping share the property that it is easier to find long deletions than long insertions. This is because the length of the read can be used for mapping deletions, whereas only part of the read (read length minus size of insert) can be used for mapping insertions. SOAPindel suffers less from this asymmetry in insertion and deletion calling than alignment-based approaches since it combines local assembly with mapping, and on simulated data the asymmetry is very low (Fig. 3). However, insertions always need more evidence than deletions to enable their detection.

SOAPindel have no preset limitations on indel size, but certain conditions can affect the max indel size that can be detected. For sufficiently long deletions, the mapped reads will be drawn apart. Hence, the cluster of unmapped reads will be split into two clusters (Fig. 5A). The threshold deletion length limitation for cluster splitting is around (insert size × 1.2 − read length) × 2. Cluster splitting does not imply that we cannot detect this deletion. The cutting border [the default is max<read_len,50>] could be extended by the length of the deletion to be detected. Thus, for detection of a 1000-bp size deletion, one should set the parameter "-ext 1000." However, extending the border will increase the misassembly rate. For long insertions, the main problem is that both paired reads can be within the insertions and not be mapped (Fig. 5B). When the size of the insertion is larger than (insert size × 1.2 − read length) × 2, we cannot detect the insertion from single end mapped reads alone.

SOAPindel does not exploit reads mapping to multiple positions and does not make use of candidate indel information provided by some alignment tools. It is also limited in the handling of certain types of repetitive sequences (Fig. 1G). Even though all alignment tools share this problem, they can find candidate positions if a unique piece of sequence (several base pairs are enough) exists around the indel. So Dindel can resolve many of the cases illustrated in Figure 1G1 and 1G2 if provided with proper candidates by alignment tools. For these reasons, we will include candidate indel information in the next release of SOAPindel (version 2.0).

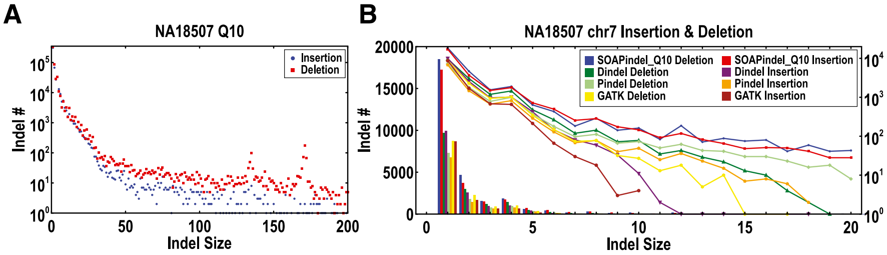

Figure 4. SOAPindel predictions (Q10) on NA18507. (*A*) Distribution of insertion and deletion. (*B*) Comparison of the detected insertion and deletion between SOAPindel, Dindel, Pindel, and GATK on chromosome 7. The result is shown as a histogram of the number of indels (*B*: *left y*-axis) and log ratio lines (*B*: *right y*-axis).

Table 1. Validation results from 30 long indels (>5 bp) called in NA18507 in the ENCODE regions ENm010 and ENm013

	Confirmed				Unconfirmed	
	Number	Longest	Het number	Hom number	Number	Longest
Deletion	16	38 bp	8	8	2	11 bp
Insertion	10	93 bp	4	6	2	11 bp
Total	26		12	14	4	

The present version of SOAPindel is programmed in Perl and thus runs on most platforms. The SOAPindel pipeline contains two major parts: (1) preparing the clusters; (2) assembly and alignment. Running times and memory requirements are different for these two components. For part one, both the running time and memory consuming are nearly linear in coverage and do not easily benefit from increasing the number of CPUs. For the complete NA18507 data, a maximum of 5 GB of memory and ~30 h of CPU time is needed. Part two is easily parallelized. Every cluster can be assembled and aligned separately. There are a total of 12.8M clusters for NA18507. Each cluster uses ~0.15 sec and ~10 MB of memory. The running time of SOAPindel is compared with the three other methods in Supplemental Figure S7.

Methods

The SOAPindel algorithm

The pipeline is outlined in Figure 1. First, we identify all sets of reads where only one mate maps, allowing for no gaps and at most three mismatches. If both reads map at a distance that is more than 40% different from the expected distance, they are treated as two pairs of single-mapped reads. Reads with multiple mapping positions are ignored during this initial screening with a small loss in sensitivity but a large gain in specificity. Each unmapped read collected is given an approximate candidate genomic position using the orientation and expected distance from the mapped mate (Fig. 1B). The underlying idea is that the density of such virtually mapped reads is higher around real indels. Note that since we do not allow for gaps during mapping, many reads that other approaches map with gaps are included in this set of reads.

Next, clusters of virtual reads are identified as regions with high coverage of reads separated by regions of low coverage (the coverage threshold is a parameter set by the user; we use the value 2 when average sequencing depth <40×, and to 1/20 of the depth for higher coverage in order to reduce the computational time) (Fig. 1C). If a cluster is too long, it will be cut into small fixed-length (default 300-bp) pieces with overlaps (default 100 bp). The genomic region around each cluster is then cut out of the alignment with a proper small extension as border and treated as a separate entity. Reads that initially mapped to the region are then added before local assembly (Fig. 1D).

When the coverage is high, the cumulative error rate is high due to sequencing errors, mapping errors, and false mate pairs. We perform one round of error correction by calling all SNPs in the region and then remove reads from the set that contain one or more mismatches that are not among the identified SNPs (Fig. 1E).

As in most other de novo assembly algorithms (Li et al. 2010), we use a *K*-dimensional De Bruijn graph of four symbols (A|T|G|C) to store possible paths among *K*-mer vertices for each candidate region. Since regions are small, all reads and repeat information can be kept in memory so we can resolve some of the hard problems in genome-wide de novo assembly (Fig. 1H1). (The assembly procedure is shown in the Supplemental Material and Supplemental Fig. S1.) Heterozygotes in a diploid genome cause bubbles in the graph. Whole genome assembly typically only traces one of the two paths (the one that is most strongly supported), but we trace both (Fig. 1H1a). Likewise, short repeats whose length $\geq K$ will cause forks on the graph (Fig. 1H1b), and tandem repeats with length $\geq \max\langle K, L\rangle + L$ (L is the pattern length; ATATAT's pattern is AT) will cause loops (Fig. 1H1c). Whole genome assembly will trace only one path or break the path, but we can trace either path as long as the repeat length is shorter than the read length.

With low coverage, the path may be broken because the default *K*-mer length is too long. In order not to lose specificity by generally lowering *K*-mer length, we use a dynamic approach of decreasing *K*-mer length where the path is broken. For candidate homozygous indels, we search for unused reads with gradually shorter *K*-mers until a path is formed or the lower bound on *K*-mer length has been reached. For candidate heterozygous indels (cases where one path with many unmapped reads is broken), we completely redo the assembly for gradually decreasing *K*-mer lengths.

During assembly we exploit that all unique *K*-mers on the reference should be collinear with the local assembly, and we can use them as markers to guide the direction, remove errors, and control the length of the local assembly.

Finally, we align all contigs (assembly results) to the reference and call the genotype of any indels present. The number of alignable contigs is twice the number of heterozygous indels plus heterozygous SNPs, since heterozygosity in the sample of both

Table 2. Validating indels by the chimpanzee genome

	Chromosome 7						All chromosomes		
	Chimp SOAPindel			Chimp Dindel			Chimp SOAPindel		
	Confirmed	Unconfirmed	% Confirmed	Confirmed	Unconfirmed	% Confirmed	Confirmed	Unconfirmed	% Confirmed
Total	11,276	41,057	21.5%	8802	21,590	29.0%	198,455	737,156	21.2%
Het Del	2744	15,653	14.9%	1638	9004	15.4%	50,034	279,469	15.2%
Het Ins	3146	13,633	18.7%	2227	6190	26.5%	57,427	246,761	18.9%
Hom Del	1987	6452	23.5%	1705	3481	32.9%	35,516	114,162	23.7%
Hom Ins	3399	5319	39.0%	3232	2915	52.6%	55,478	96,764	36.4%
≤5 bp	10,552	37,611	21.9%	8527	20,746	29.1%	185115	675,826	21.5%
>5 bp	724	3446	17.4%	275	836	24.8%	13340	61,330	17.9%
>10 bp	325	1412	18.7%	10	144	6.5%	5710	24,630	18.8%

The number of indels called by SOAPindel (Q10) and Dindel from the read data of NA18507 that can/cannot be supported by the chimpanzee/human alignment (pantro3 vs hg19).

Table 3. The genomic distribution of NA18507 indels

	Coding (32M)	Noncoding (1.2G)	Intergenic (1.9G)	All (3.1G)
Total indels	0.0017%	0.0354%	0.0274%	0.0302%
Chimp confirmed	0.0003%	0.0073%	0.0060%	0.0064%

The density (indel rate/bp) of indels in different functional parts of the genome for all indels (total of 1,018,647 Q10 indels), and the indels confirmed by the human–chimpanzee alignment (total of 200,923 indels).

indels and SNPs causes bifurcations in the graph. Since the time complexity of pairwise alignment is $O(n^2)$, we save significant computational time by exploiting the unique K-mers again to shorten the length of the region to be aligned. For each contig, we compare its K-mer set with the K-mer set of the reference to identify all nail pairs (a nail is a K-mer that is unique on both contig and reference; a nail pair is a pair of adjacent nails). The presence of one or more nail pairs with different distances on the contig and the reference is sufficient for calling a candidate indel. A Smith-Waterman alignment between uneven nail pairs defines the exact position and sequence of candidate indels (Fig. 1I) (for more details, see the Supplemental Material). This constitutes the full set of candidates for further processing.

Quality scores of candidate indels

Several filters have been designed for increasing specificity, and all can be modified by the user and are described in the Supplemental Material. The most important of these are also integrated in a quality score. SOAPindel uses a method similar to Phred (Ewing and Green 1998) to assign Q-values to all candidate indels.

We use five parameters that are particularly effective at discriminating between true and false indels:

1. Coverage
2. Indel size
3. Number of neighboring SNPs and indels
4. Position of the second different base pair (Indels surrounded by low-complexity sequence are more likely to be false positives.)
5. Distance to the edge on assembled contig (Indels near the edge are more likely to be false positives.)

We generated training data to build up the parameter threshold lookup table by simulating data based on the alignments between the Venter genome and hg19. Running SOAPindel on this simulation data set produces the training data and the lookup table for assigning Q-values to candidate indels. Retraining should therefore be done if using this quality score for calling indels in genomes of other species.

Complex repeats

Three different categories of repeats can currently not be handled by SOAPindel: (1) Repeats with length >(max read length) and distance >(max cluster length) (default 300) (Fig. 1G1). (2) Repeats with length >(max read length) and distance <(max cluster length) will confuse assembly (Fig. 1G2). (3) Palindrome repeats with length >(max read length) will break the assembly (Fig. 1G3).

Simulations

We simulated data based on the Venter genome (assembled from long reads; see http://huref.jcvi.org/) and mapped simulated read pairs against hg19 in order to evaluate the performance of the different indel callers. Paired reads were simulated from the observed differences between the two genomes, sampling either from the same genome or from different genomes to allow for a diploid individual to show a homozygous and heterozygous SNP/indel, respectively. The ratio of homozygotes to heterozygotes was fixed at 2:3, and paired-end reads were 100-bp reads with 500-bp insert size, and the coverage was 20×. We introduced random sequencing errors corresponding to a read quality q32 for the first 75 bp, q31 for the next 15 bp, and q30 (0.1% error) for the last 10 bp. We did not introduce sequencing errors causing indels in the reads, since indel errors are much rarer than wrong bases in Illumina sequencing (Minoche et al. 2011).

In a different set of simulations specifically designed to test FN and FP as a function of indel size, we generated artificial data based on human chromosome 22. We introduced random SNPs at a density of 1 in 1000. Insertion or deletion events were randomly placed at 0.025% of the positions as heterozygous or homozygous at a ratio of 3:2. Data sets were simulated with the following combinations of parameters: Indel sizes (1, 3, 5, 10, 30, 50, 100); read lengths: (50 bp, 75 bp, 100 bp, 125 bp); and coverages: (5×, 10×, 20×, 40×, and 80×) with insert size = 500 bp in all cases. Sequencing errors in reads were introduced as above.

Mapping of the simulated reads to the hg19 was done using SOAP2 (Li et al. 2009) or BWA (Li and Durbin 2009) (SOAPindel supports both formats).

Benchmarking

We benchmarked SOAPindel against Dindel (version 1.01), Pindel (version 0.2.4d), and GATK (version 1.4-9) on all simulated data and for analyses of real data. In all cases, we used default parameter settings. For SOAPindel, we investigated three quality thresholds (Q1, Q5, and Q10), whereas for the other programs, we used default settings. The evaluation measures were sensitivity and specificity as a function of indel size, sequencing depth, and read length. To determine whether a candidate indel is a true positive, we required the candidate indel to match a real indel of identical size, but allowing a small tolerance in the position of the indel (depending on the complexity of the adjacent sequence).

Validation

Validation was based on analysis of the sample NA18507 (http://ftp.ncbi.nih.gov/1000genomes/ftp/data/NA18507/sequence_read/).

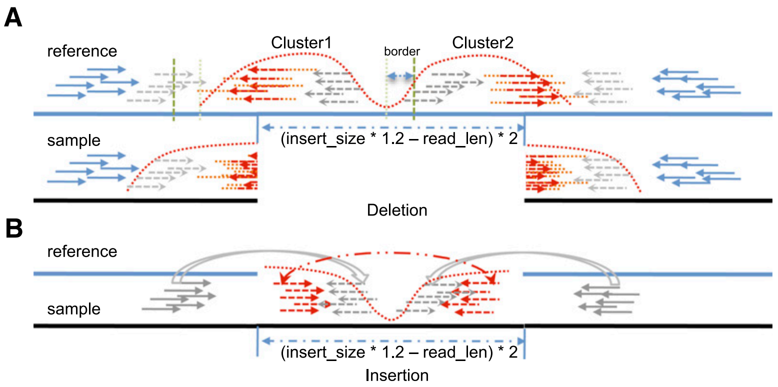

Figure 5. Illustration of some theoretical detection limits for SOAPindel. (*A*) Deletion size threshold. (*B*) Insertion size threshold.

Two types of validation were pursued. First, as a direct validation, we randomly chose 30 insertions and deletions with Q10 or above and longer than 5 bp for Sanger sequencing after PCR amplification. Second, we compared the detected indels to the alignment between chimpanzee genome and hg19 in order to determine whether a new indel call matched the chimpanzee sequence. We note that this can validate at most 50% of real indels since the chimpanzee is equally likely to match the new indel call and the reference genome.

Data access

The SOAPindel program is written in Perl and can be obtained from http://soap.genomics.org.cn/soapindel.html or https://sourceforge.net/projects/soapindel.

Acknowledgments

We thank Min Jian, Ye Yin, Feng Wang, and Jia Lu for their contributions to PCR-based indel validation; Freddy B. Christiansen for extensive comments on previous versions of the manuscript; Cornelis Arnout Albers for advice on Dindel comparisons; and three anonymous reviewers for comments that greatly improved the manuscript. We acknowledge funding from EU FP7 Marie Curie IAPP "NextGene," BGI-Shenzhen, LuCamp (Lundbeck Foundation Center), the Danish Research Councils, the National Basic Research Program of China (973 program no. 2011CB809201, 2011CB809202, 2011CB809203), the National Natural Science Foundation of China (30890032, 31161130357), Shenzhen Key Laboratory of Transomics Biotechnologies (CXB201108250096A), and Shenzhen Key Laboratory of Gene Bank for National Life Science.

References

Albers CA, Lunter G, Macarthur DG, McVean G, Ouwehand WH, Durbin R. 2011. Dindel: Accurate indel calls from short-read data. *Genome Res* **21:** 961–973.

Carnevali P, Baccash J, Halpern AL, Nazarenko I, Nilsen GB, Pant KP, Ebert JC, Brownley A, Morenzoni M, Karpinchyk V, et al. 2012. Computational techniques for human genome resequencing using mated gapped reads. *J Comput Biol* **19:** 279–292.

Ewing B, Green P. 1998. Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* **8:** 186–194.

Homer N, Nelson SF. 2010. Improved variant discovery through local re-alignment of short-read next-generation sequencing data using SRMA. *Genome Biol* **11:** R99. doi: 10.1186/gb-2010-11-10-r99.

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

Li H, Ruan J, Durbin R. 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res* **18:** 1851–1858.

Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J. 2009. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics* **25:** 1966–1967.

Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Li Y, Li S, Shan G, Kristiansen K, et al. 2010. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res* **20:** 265–272.

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* **20:** 1297–1303.

Mills RE, Pittard WS, Mullaney JM, Farooq U, Creasy TH, Mahurkar AA, Kemeza DM, Strassler DS, Ponting CP, Webber C, et al. 2011. Natural genetic variation caused by small insertions and deletions in the human genome. *Genome Res* **21:** 830–839.

Minoche AE, Dohm JC, Himmelbauer H. 2011. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. *Genome Biol* **12:** R112. doi: 10.1186/gb-2011-12-11-r112.

Mullaney JM, Mills RE, Pittard WS, Devine SE. 2010. Small insertions and deletions (INDELs) in human genomes. *Hum Mol Genet* **19:** R131–R136.

Ye K, Schulz MH, Long Q, Apweiler R, Ning Z. 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.

Zerbino DR, Birney E. 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* **18:** 821–829.

Received September 29, 2011; accepted in revised form September 10, 2012.