BRIEFINGS IN BIOINFORMATICS. VOL 14. NO 1. 46–55
Advance Access published on 24 March 2012
doi:10.1093/bib/bbs013

Analysis of insertion–deletion from deep-sequencing data: software evaluation for optimal detection

Joseph A. Neuman, Ofer Isakov* and Noam Shomron*

Submitted: 11th November 2011; Received (in revised form): 22nd February 2012

Abstract

Insertion and deletion (indel) mutations, the most common type of structural variance in the human genome, affect a multitude of human traits and diseases. New sequencing technologies, such as deep sequencing, allow massive throughput of sequence data and greatly contribute to the field of disease causing mutation detection, in general, and indel detection, specifically. In order to infer indel presence (indel calling), the deep-sequencing data have to undergo comprehensive computational analysis. Selecting which indel calling software to use can often skew the results and inherent tool limitations may affect downstream analysis. In order to better understand these inter-software differences, we evaluated the performance of several indel calling software for short indel (1–10 nt) detection. We compared the software's sensitivity and predictive values in the presence of varying parameters such as read depth (coverage), read length, indel size and frequency. We pinpoint several key features that assist successful experimental design and appropriate tool selection. Our study may also serve as a basis for future evaluation of additional indel calling methods.

Keywords: *mutation; insertion; deletion; indel; indel-calling; deep sequencing; next generation sequencing; software evaluation*

INTRODUCTION

Variation profiling, the detection and localization of an individual's single nucleotide polymorphisms (SNPs), copy number variations (CNV) and structural variants, is becoming a feasible endeavor owing to the unprecedented rate of human genome sequencing and the rapid evolution of computational variance analysis. Indels (insertions and deletions) analysis is a growing field in structural variants assessment. Indels are the second most common type of polymorphism and the most common structural variant [1]. Their presence contributes to the pathogenesis of disease [2], gene expression and functionality [3], viral disease forms identification [4] and they can be used as genetic markers in natural populations [5]. Common indel detection methods include microarrays [6], real-time polymerase chain reaction (PCR) [7], denaturing high-performance liquid chromatography (DHPLC) [8] and high-throughput sequencing (Deep sequencing or next-generation sequencing) [9].

Detection of indels through deep sequencing is becoming more common due to the decrease in cost, increase in efficiency and sensitivity improvements demonstrated by the various sequencing platforms and analysis tools [10, 11]. Indels occur in an estimated rate that is 8-fold lower than SNPs [12]. This rate, which varies extensively between sequenced individuals, can partially be attributed to performance variability within mapping and

Corresponding author. Noam Shomron, Department of Cell and Developmental Biology, Sackler Faculty of Medicine, Tel Aviv University, PO Box 39040, Tel Aviv 69978, Israel. Tel.: +972-3-6406594; Fax: +972-3-6407432; E-mail: nshomron@post.tau.ac.il
*These authors contributed equally to this work.

Joseph A. Neuman is an MD student, holds a B.Sc. degree in Applied Mathematics from Bar-Ilan University. He works at Shomron's laboratory on high-throughput data management and software evaluation and design.

Ofer Isakov is an MD/PhD student at Shomron's laboratory. He uses bioinformatics and advanced high-throughput technologies to elucidate genes and pathways involved in genetic diseases.

Noam Shomron heads a research laboratory at the Sackler Faculty of Medicine at Tel Aviv University. His team combines experimental and computational biology in order to bring genomic information to clinic reality.

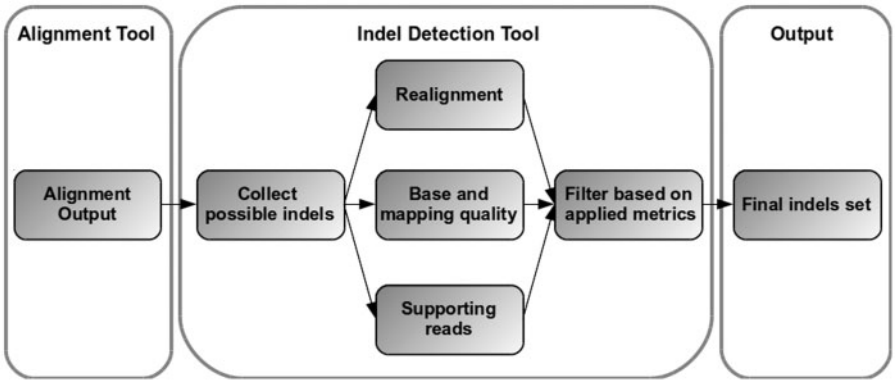

Figure 1: Basic indel calling workflow. The initial step is alignment against a reference genome in which all possible indels are detected. The following step, performed by the indel calling tools, is the collection of these possible indels, calculating various metrics, depending on the specific tool, that either support or oppose the presence of each indel. An optimal combination of both alignment and indel calling tools should result in an accurate set of confidently called indels.

detection tools [5]. Sequence reads covering indels are generally more difficult to map since their correct alignment either involves complex gapped alignment or paired-end sequencing inference [13]. The key computational software tools required during deep-sequencing indel detection analysis are alignment and indel detection tools that interpret the alignment results in order to infer the presence of an indel. This analysis process varies between different methods and is based on per base quality, mapping property, number of supporting reads, realignment around potential indels, known variation data and various other probabilistic matrices. An effective combination of the two will produce the optimal detection pipeline that will result in accurate and reliable variation calling (Figure 1). The effect of different alignment tools on detection efficiency has been studied and accounted for [11], recommending the use of single-end reads gapped alignment enabled mapping tools such as BWA [13] and Novoalign [14]. However, these studies did not address the effects and implications of the software chosen to detect the indels; therefore, further knowledge on the effects of these tools is still required.

The variety of available indel identification software is rapidly increasing with better performance, sensitivity and specificity as the main objectives. Indel identification becomes more complex when detection is made using single-end reads shorter than <100 nt since they lack insert length variance (the gap between sequences in paired-end reads) that facilitates indel detection [15]. We set out to compare four common indel detection software – VarScan [16], Dindel [17], SAMtools mpileup [18] and the Genome Analysis Toolkit (GATK) [19]. Using simulated sample data, we compared the detection software sensitivity and predictive values while changing initial parameters such as read depth (coverage), read length, indel size and frequency. We implemented these indel tools on real experimental data in order to demonstrate concurrence to our simulations. In general, our study pinpoints several key features that assist successful experimental design and appropriate tool selection. Our study may also serve as a basis for future evaluation of additional indel calling methods.

METHODS

The simulated data

In order to evaluate how well the different software can detect indels, we simulated several genomic regions that contain indels and SNPs in variable frequencies. In order to construct the simulation data, we extracted a section of 10 M base pairs in length from human chromosome 16 (between 10 000 001 and 20 000 000; build GRCh37/hg19) to be used as a reference sequence. Using a specialized software (inGAP [20]), SNPs were inserted at a rate coinciding with observed human genome SNP rate of 1:1000 bases [21]. Indels were inserted to the simulated read data according to the specific comparison analysis (specified below). We then created a set of simulated deep sequencing single-end reads data using the same software, each read data according to the specific variable

examined (available upon request). As indel lengths tested in our study (1–10 nt) are lower than the simulated read lengths (36–72 nt), we expect the various tools to demonstrate high performance even without the advantages of paired-end reads. We therefore chose to focus our analysis on the tools' performance using simulated single-end data. Base qualities were assigned using read qualities retrieved from publicly available Illumina GAII whole-genome sequencing run, retrieved from the Short Read Archive (http://www.ncbi.nlm.nih.gov/sra; study accession number: SRP002535) in order to better simulate the known inherent decrease in base calling quality. Excluding the experiment testing the specific variable effect, all of the simulated read data were set to an indel frequency of 1:10 000, which is similar to the average indel rate observed in human DNA [22]. The average coverage was set to 100× with a read length of 72 nt in order to simulate optimal conditions in which only our tested parameter is expected to influence performance.

Alignment

We aligned the simulated sequence reads against our reference using BWA (v0.5.9), an open-source, popular, publicly available tool that utilizes backward search with Burrows–Wheeler transform (BWT), supports gapped alignment and was shown to be suitable and optimal for both SNP and indel calling [11]. The alignment was performed using the same arguments for all comparison types, allowing gap extensions of up to 10 bases:

bwa aln -e 10 <reference sequence> <reads>

Indel detection

Detected indels were recorded as true positives when the exact correct position and allele sequence were identified. We also considered indels that were in challenging positions (e.g. in a homopolymer sequence) that result in the same exact post-indel consensus sequence as our reference indel as true positives. In the event where the software detects the wrong allele at the correct position, the indel was considered a false positive. Our chosen indel detection software for comparison were as follows: (i) Dindel (v1.01), (ii) VarScan (v2.2.2), (iii) GATK (v1.0.5083) and (iv) SAMtools mpileup (v0.1.16).

Dindel [17] is a software developed by the Wellcome Trust Sanger Institute in the UK, which utilizes a Bayesian approach for small (<50 nt) indel calling by basically realigning the sequence reads against a variety of candidate haplotypes for which prior probabilities have been assigned. Dindel takes into account each read's reference similarity and mapping quality. As Dindel sequencing error indel model was trained using Illumina data, it should only be used for indel analysis of Illumina sequencing data. We ran the software using its default arguments, with a maximal number of candidate haplotypes of eight, which according to the developers provides high sensitivity while maintaining low false discovery rates.

The command workflow:

```
dindel –analysis getCIGARindels –reads.bam
<reads.bam> –outputFile <outFile.dindel_output>
–ref <reference.fasta> makeWindows.py
–inputVarFile  <outFile.dindel_output.variants.txt>
–windowFilePrefix <winFile.realign_windows>
–numWindowsPerFile 1000
```

Then, for each file created,

```
dindel –analysis indels –doDiploid –reads.bam
<reads.bam> –ref <reference.fasta> –varFile
<outFile.realign_windows.$i.txt> –libFile
<outFile.dindel_output.libraries.txt> –outputFile
<outFile.dindel_stage2_output_windows.$i>
  mergeOutputDiploid.py –inputFiles
<outFile.dindel_stage2_outputfiles.txt> –outputFile
<outFile.variantCalls.VCF> –ref <reference.fasta>
```

VarScan [16] is an open-source tool that utilizes a SAMtools (v0.1.16) [18] generated pileup file for scoring and sorting the sequence alignments in order to isolate reads that map uniquely to one location in the reference sequence. Unmapped and ambiguous reads are discarded from downstream analysis and the uniquely mapped reads are scanned for variants (mismatches and indels). Alleles are then defined by applying criteria set, including read coverage, *P*-value, variant frequency, base quality and more. We performed our analysis with VarScan's default parameters (unfiltered VarScan) and again with more rigorous filtering (filtered VarScan), analyzing indels with a minimal coverage of 20× reads and a minimal allele frequency of 0.35:

```
java -jar VarScan.v2.2.2.jar pileup2indel
<pileup.file> > <outFile.varscan-indel.output>
java -jar VarScan.v2.2.2.jar pileup2indel
<pileup.file> –min-reads2 20 –min-var-freq 0.35 >
<outFile.varscan-indel.output>
```

GATK [19] is a collection of biodata analysis tools developed by the Broad Institute at MA, USA, which allows variant calling with its Unified

Genotyper (UG) tool. Specifically, UG's indel calling process is based on Dindel, though as we demonstrate (see below) the tools perform differently under a variety of conditions. We performed our analysis using UG's default arguments, requiring a minimal indel phred-scaled Qscore of 30, with a minimal standard emission Qscore of 30. As per the tool's recommended workflow, we first realign the reads around indels before we call variants.

Realignment:

```
java -jar GenomeAnalysisTK.jar
-T RealignerTargetCreator
-I <sorted.bam>
-R <reference.fasta>
-o <intervals.intervals>

java -Xmx4g -jar GenomeAnalysisTK.jar
-I <sorted.bam>
-R <reference.fasta>
-T IndelRealigner
-targetIntervals <intervals.intervals>
-o <sorted.realigned.bam>
```

Indel calling:

```
java -jar GenomeAnalysisTK.jar
-T UnifiedGenotyper
-glm DINDEL
-R <reference.fasta>
-I <realigned.sorted.bam>
-l INFO
-o <output.vcf >
-stand_call_conf 30.0
-stand_emit_conf 30.0
```

SAMtools mpileup [18] is based on a Bayesian model for indel calling, mpileup is probably one of the most common tools for variation detection due to its simple use and various features such as underlying read mismatch consideration, local realignment and base quality assessment. We performed the analysis using the default indel detection parameters, with a small increase in the coverage threshold (-D 200).

```
samtools view -bS <input.sam> > <aln.bam>
samtools sort <aln.bam> <aln.sorted>
samtools index <aln.sorted.bam>
samtools mpileup -C50 -uf <referece.fasta>
   <aln.sorted.bam> > <aln.mpileup>
bcftools view -bvcg <aln.mpileup> >
   <alnvar.raw.bcf>
bcftools view <alnvar.raw.bcf>| vcfutils.pl
   varFilter -D200 > <aln.var.flt.vcf>
```

Statistical analysis

The correlation between the different test parameters (e.g. indel frequency, read length, indel size and read depth) and their effect on positive predictive value (PPV) and sensitivity was tested using Pearson's correlation tests.

RESULTS

Effect of indel frequency

Indel frequency has been shown to vary across the genome with lower rate in conserved and functional regions and an increased rate in hot spots for genetic variation [22, 23], with an approximate average rate of 0.7–1.9 indels in 10 kb of DNA [22]. We compared the indel calling software's performance with indel frequency ranges of 1:10 –100:10 kb while maintaining a constant coverage of 100× and read length of 72 nt. We found that increased indel frequency up to 10:10 kb did not affect the software's performance and sensitivity decreased only for extremely high indel frequency levels (>10:10 kb) with filtered VarScan demonstrating the highest decrease in sensitivity (20%). GATK presented a mild decrease in PPV (0.994–0.979; $R^2 = 0.97$; $P < 0.05$) as frequency increased.

Effect of read length

Read length varies between platforms and within each platform [24]. Using Illumina's most common range of single-end read length (36–72 nt), we set out to test the effect on the performance of the indel calling software while maintaining constant coverage of 100× and conclude the optimal number of sequencing cycles for the detection process. We found that although the coverage was maintained, raising the read length greatly improved performance, increasing sensitivity across all software and PPV for GATK and Dindel ($P < 0.005$). Indel calling using read length of 36 nt resulted in a sensitivity range of 0.64 (mpileup) to 0.76 (VarScan unfiltered). This decrease in indel detection sensitivity is mainly due to poor detection of insertions 4–7 nt long and inability to detect longer (>7) insertions all together (Figure 2). Raising the read length to 54 nt resulted in a 40% increase in sensitivity reaching an average of 0.91. In order to test whether this inability to detect long insertions is due to failure to align covering reads, we compared the average coverage for each type and length of variation. We observed a notable decrease in the number of

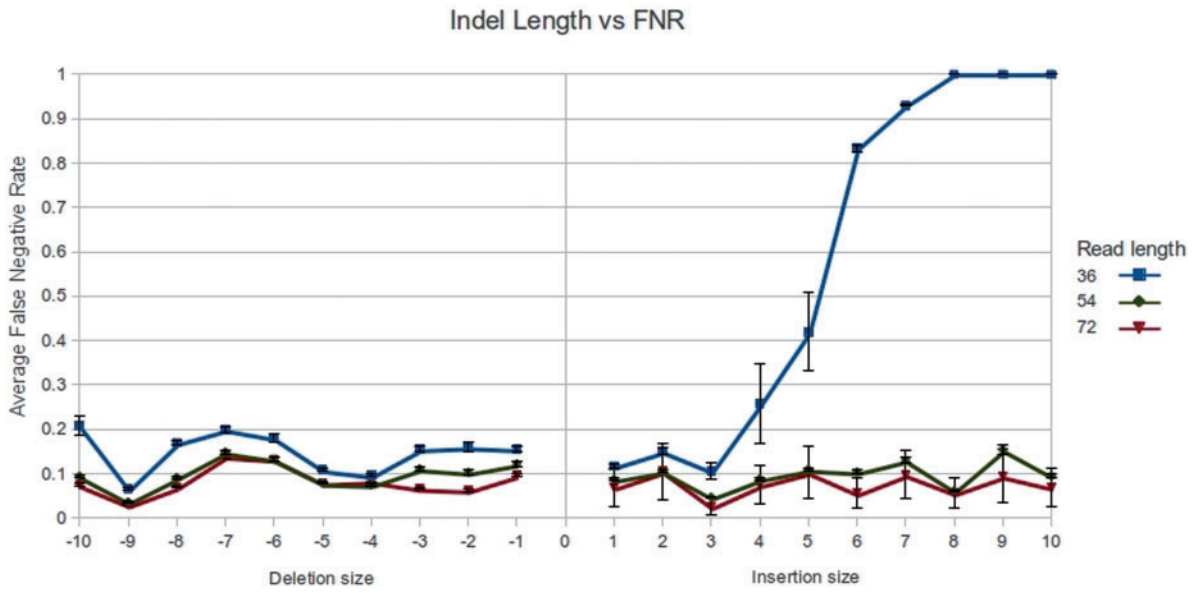

Figure 2: Indel length versus false negative rate (FNR), describing the average FNR across all software as indel length increases. It is noticeable that insertions and deletions are detected in a similar rate across lengths, with the exception of increased rate of missed insertions of >3 nt long, when read length is 36 nt.

covering reads as indel length increased for both variant types (98.6–28.3; 95.7–44.1 for insertions and deletions, respectively). This reduced mapping of reads covering longer insertions results in decreased detection capability for insertions longer than >3 nt when utilizing short (≤36 nt) reads.

Effect of indel size

Previous studies show that the majority of naturally occurring short indels (<60 nt) are 1–10 nt long [1, 25]. We set out to compare the effect of these indel sizes on indel calling software performance by simulating reads with constant indel sizes ranging from 1 to 10 nt with 1 additional simulation containing variable sizes of 1–10 nt. All the simulated runs were designed with a read length of 72 nt and constant high coverage of 100 nt. We found higher indel length to be correlated with a mild but significant decrease in PPV (0.997–0.982; 1–0.987; $P < 0.005$) in GATK and Dindel and a more prominent decrease in unfiltered VarScan (0.72–0.4; $P < 0.005$). Though some of these changes in PPV are mild, they become more significant when covering large genomic sequences that include many indels. Indel size did not affect sensitivity in any of the software, but filtered VarScan, which in addition to demonstrating high sensitivity across indel size second only to the unfiltered version, also demonstrated a mild sensitivity decrease with indel size increase (0.95–0.919; $R^2 = 0.82$; $P < 0.001$) (Figure 3). The other software did not display any difference in performance due to changes in indel sizes, though as mentioned above, higher indel sizes could affect insertion calling efficiency in lower read lengths.

Effect of read depth (coverage)

Previous studies demonstrated positive correlation between variant calling sensitivity and increased read depth [11]. Since higher read depth requires the researcher to either reduce the selected target region in which indels are called or perform a higher number of sequencing cycles to produce longer reads to cover the target region, demonstrating the minimal effective read depth for indel calling can potentially extend our target area or save resources. We set out to test the coverage effect on indel calling software performance. As expected, we found coverage to be significantly positively correlated with sensitivity ($P < 0.001$). We observed that the software most affected by coverage was VarScan, with a 93% increase in sensitivity (0.516–0.994) in the unfiltered version when raising the average coverage from 10× to 30× and a 73% increase (0.498–0.86) in the filtered version when coverage is raised from 30× to 50×. GATK also had a significant increase in sensitivity when coverage was

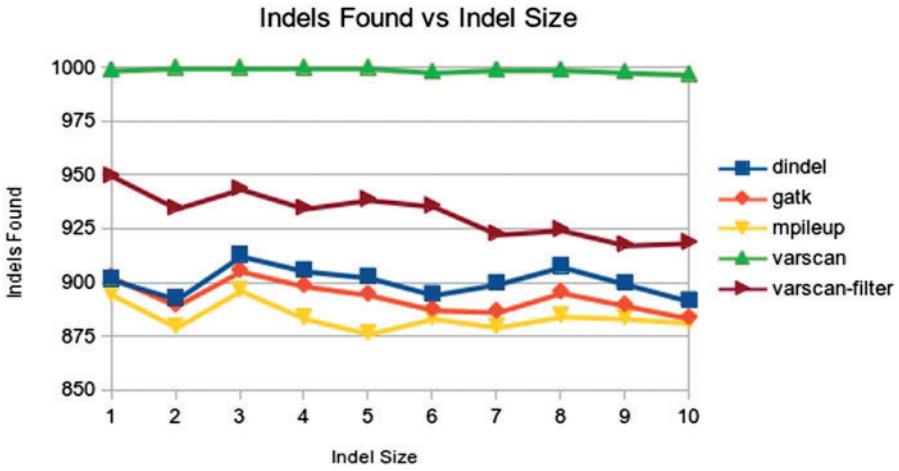

Figure 3: Indels found versus indel size shows the similar sensitivity of GATK, mpileup and Dindel and the higher sensitivity of VarScan and filtered VarScan across all indel sizes. Only filtered VarScan demonstrated a significant decrease in sensitivity as indel size increases.

increased from 10× to 30×, discovering 15% more indels (0.759–0.875). Since GATK presents a high PPV (>0.998), it is advised to lower the minimal quality threshold when detecting indels with coverage lower than 30 in order to increase sensitivity. Dindel presented high sensitivity across coverage range, detecting 84% of the indels in the lowest tested coverage (5×). It was interesting to find that the majority of indels undetected by GATK, SAMtools mpileup and Dindel in low coverage (∼10%) remained undetected even when coverage was 10 times higher. The majority of these elusive indels were inside a nonunique region in the reference (a sequence that is present in more than one location in the reference genome). We also noticed that Dindel, SAMtools mpileup and unfiltered VarScan reach a plateau of true indel calls when coverage rises over 30, whereas GATK and filtered VarScan show a steady mild increase in true indel calls as average coverage increases. All the software besides SAMtools mpileup demonstrated a decrease in PPV as the coverage increased ($P < 0.005$) with unfiltered VarScan reaching a PPV rate of ∼40% when average coverage was 150 (Figure 4).

Real data analysis

In order to further estimate the performance of each indel detection tool, we utilized data produced by Hillier *et al.* [26], which include whole-genome sequencing of *Caenorhabditis elegans* and a data set of 202 validated single-base indels across the genome. The genome was sequenced using Solexa sequence analyzer, producing >37 million reads with an average length of 31 nt that resulted in 9x depth of coverage across the genome. As the read length, coverage and validated indel lengths resemble our simulated data, it serves as an appropriate set for testing and validation our simulations. We aligned the sequenced data against the *C. elegans* WS170 reference sequence using BWA and analyzed the results using each of the tested software. Consistent with our low coverage simulations, Dindel demonstrated the highest sensitivity, detecting 175 (87%) of the indels, whereas mpileup and GATK detected 153 and 145 of the indels, respectively (due to minimal quality thresholds that reduce their sensitivity for indels with low coverage). VarScan detected 103 indels supporting our previous observation that VarScan requires higher coverage in order to reach its optimal sensitivity rate (Table 1).

DISCUSSION

Accurate indel detection is imperative for variant profiling. Deep sequencing is producing vast amounts of sequence data that require precise and comprehensive interpretation to infer the presence of SNPs, CNVs and other structural variants. Correct interpretation of the data depends on a combination of exact sequence mapping and valid reliable variance inference software. This work tested the latter paying particular attention to the different features and the performance variability between publicly available software for indel calling.

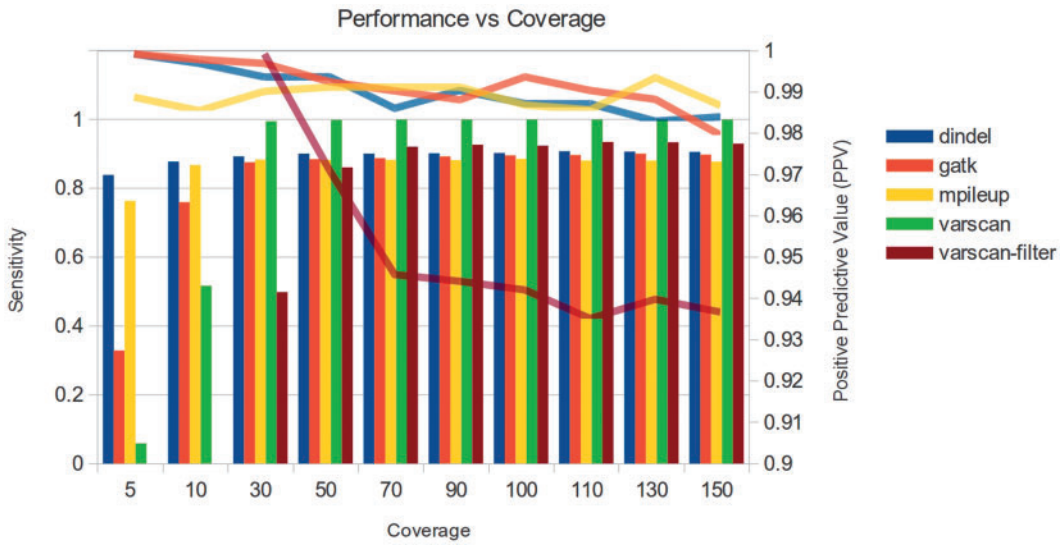

Figure 4: Performance versus coverage, both detected indels (bars) and PPV (lines) against coverage. Our set parameters for the filtered VarScan do not permit indel calling with coverage <20×, so it did not detect any indels in coverage ≤10×. The figure depicts the combined increase in detected indels and decrease in PPV as coverage increases. Unfiltered VarScan's PPV is not presented in this figure since it is much lower (0.77–0.41) than the rest of the software.

Table 1: Indel calling performance for each of the tested tools implemented on sequencing data

Tool name	Indels found	Indels missed	Insertions found	Insertions missed	Deletions found	Deletions missed
Dindel	175	27	97	16	78	11
GATK	145	57	84	29	61	28
mpileup	153	49	90	23	63	26
VarScan	103	99	62	51	41	48

Source: Ref. [26]; also see text.
These results support our simulation-based observations in which Dindel presents the highest sensitivity in low coverage experiments, whereas GATK and VarScan require additional parameter modification in order to reach their optimal sensitivity values.

In order to produce an evaluation model for future indel calling tools, we examined both technical and biological parameters that affect detection capabilities across software. We demonstrated a significant correlation between these parameters and a variety of performance indicators such as sensitivity and precision.

Variable effects

We presented a model for indel calling performance-associated variables testing, describing each variable's predicted effect on various elements of the detection process. Increased coverage and read length were shown to be significantly correlated with increased sensitivity. We conclude that when one aims to increase coverage, increasing the sequenced reads' length rather than amount of reads should be favored in order to increase sensitivity while maintaining a high PPV. Our data also demonstrate that raising the coverage by increasing the number of sequenced reads beyond 30× only mildly improves sensitivity and even when extremely high coverage was implemented, there were still indels that could not be accounted for due to their location within genomic repetitive regions. Only when other measures such as increased read length or software combination were implemented we could detect some of these indels. We also showed that working with short read lengths (36 nt), insertions longer than >3 nt, are significantly more likely to remain undetected and proper consideration should be taken when searching for the presence of such insertions.

Software effects

Our comparative study demonstrated the importance of software-specific parameter settings. In order to exemplify this, VarScan was run naively with its default parameters and then again with a more strict indel inclusion parameters setting. This resulted in an extensive decrease in the number of false positive calls across all tests and only a mild decrease in sensitivity. We should state that VarScan outputs several different variations of a detected indel; this accounts for a large portion of its low PPV demonstrated in our tests and the decrease in PPV as indel length increases. When comparing the different software's performance (Table 2), VarScan's performance was highly affected by coverage, presenting a significant rise in sensitivity for coverage higher than 30× and the highest sensitivity for coverage > 70×. When dealing with low coverage (<30×), Dindel and SAMtools mpileup presented the highest sensitivity. Higher coverage resulted in similar performance for GATK and Dindel. Dindel's performance can be due to its inherent testing of each aligner-detected indel, which also results in longer processing times, making it the most time consuming of the tested tools. We found that the majority of undetected indels were shared across software, concluding that the benefit from summing two indel calling methods (accepting indels called in any of the two) results in only a mild increase in sensitivity, at the cost of a similar mild decrease in PPV (Figure 5). However, including indels supported by at least two software did not change the sensitivity, although the PPV was significantly higher. This is highly important when dealing with high coverage data, in which our data demonstrated a mild decrease in PPV for each of the tools (average PPV excluding unfiltered VarScan = 0.972). In this high coverage data, including only the indels supported by at least two software, resulted in a PPV of 0.991.

We emphasize two important factors for improving performance: tool selection and parameter setting. The latter was demonstrated using the indel calling tool VarScan and was shown to affect performance to a greater extent. Since VarScan calls

Table 2: Advantages and limitations for each of the tested indel detection tools

Tool name	Advantages	Limitations
GATK	Highly supported with good overall performance	Low sensitivity at very low coverage (<10×; can be improved by less stringent parameters)
Dindel	Best performance at low coverage	Only suitable for Illumina data analysis and has long running time
SAMtools mpileup	High PPV and simple use	Lowest sensitivity at high coverage (>50×)
VarScan	High sensitivity at intermediate/high coverage (>30×) and simple use	Low PPV at default parameter settings and low sensitivity at low coverage (<30×)

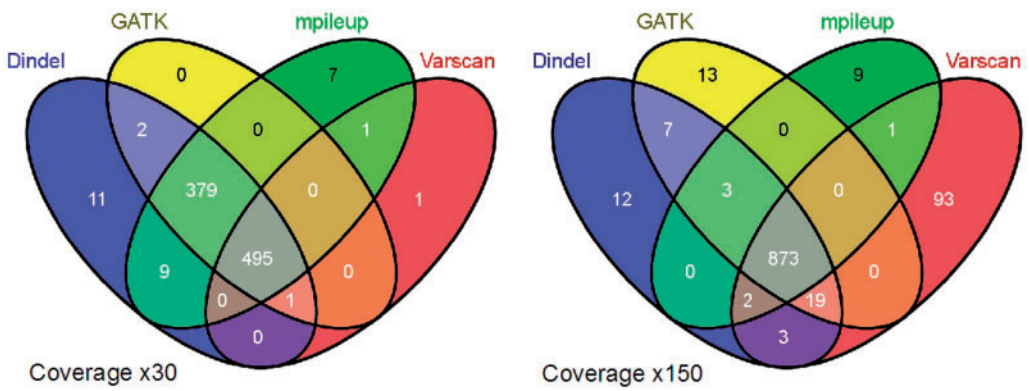

Figure 5: Venn diagram depicting the number of indels found for each software with 30× and 150× coverage and read length 72. Inclusion of indels called in any of the software results in a decrease in PPV with only a mild sensitivity improvement. Inclusion of indels supported by at least two software results in a sensitivity improvement for some of the software and a significant increase in PPV, crucial in high coverage data.

indels even when supported only by low mapping quality reads, it demonstrated not only the highest sensitivity rate across tests but also the lowest PPV. We proved that the performance variability could be greatly accounted for by a more rigorous indel calling settings. We strongly recommend researchers to be aware of this performance variability and consider the appropriate parameters, allowing higher sensitivity where additional indel confirmation tests are available. Applying a strict setting is important in particular when relying solely on one specific detection tool. GATK, the variant detection tool from the Broad Institute, is constantly evolving, with a variety of calling parameters for performance optimization. Since testing each parameter setting effect was beyond the scope of this analysis, we tested GATK's performance with its default suggested parameters, demonstrating high PPV across tests, with only mild performance variability, that could possibly be alleviated utilizing a different, more specific setting. Such is the case with our low coverage simulations in which the decrease in GATK's sensitivity can be attributed to high quality thresholds for inclusion. GATK's online support (http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit; http://getsatisfaction.com/gsa) is very helpful for such parameter setting considerations. We urge researchers to consult these resources for call optimization. Dindel, developed by the Wellcome Trust Sanger Institute, was also run using its default parameters, maintaining a consistent high sensitivity and high PPV, even in demanding settings, making it a suitable tool for low coverage experiments and tool combination possibilities.

CONCLUSION

Our analysis demonstrated the performance of several currently available indel calling software. We show that insertions longer than 3 nt will be challenging to identify when working with short read lengths ($\leq$36 nt); using at least two indel calling methods resulted in only a mild increase in sensitivity; however, accepting indels detected by at least two indel calling methods significantly increases the PPV without a major effect on sensitivity; for increasing PPV and reducing the false positive calls, coverage should be increased by extending the read length rather than the amount of reads. Finally, appropriate management of these features will result in improved accuracy and comprehensive and reliable indel calling.

Key Points

- In order to infer indel presence, deep sequencing data have to undergo comprehensive computational analysis.
- Selecting which indel calling software to use can often skew the results and may affect downstream analysis.
- We evaluated the performance of several indel calling software for short indel detection.
- We pinpoint key features that assist successful experimental design and appropriate tool selection.
- Our study serves as a basis for future evaluation of additional indel calling methods.

Acknowledgements

We thank D. Golan for assistance in the statistical analysis. We gratefully thank L.W. Hiller and A.R. Quinlan for sharing validated indel data.

FUNDING

Chief Scientist Office, Ministry of Health, Israel (Grant No. 3-4876); the Israel Cancer Association and the Wolfson family Charitable Fund, in part; I-CORE Program of the Planning and Budgeting Committee and the Israel Science Foundation (Grant No. 41/11); Edmond J. Safra Bioinformatics program at Tel Aviv University (to O.I.). This work was performed in partial fulfillment of the requirements for a PhD degree of O.I. and an MD thesis of J.N., Sackler Faculty of Medicine, Tel Aviv University.

References

1. Mullaney JM, Mills RE, Pittard WS, *et al*. Small insertions and deletions (INDELs) in human genomes. *Hum Mol Genet* 2010;**19**:R131–6.
2. Budde SM, van den Heuvel LP, Janssen AJ, *et al*. Combined enzymatic complex I and III deficiency associated with mutations in the nuclear encoded NDUFS4 gene. *Biochem Biophys Res Commun* 2000;**275**:63–8.
3. Dayi SU, Tartan Z, Terzi S, *et al*. Influence of angiotensin converting enzyme insertion/deletion polymorphism on long-term total graft occlusion after coronary artery bypass surgery. *Heart Surg Forum* 2005;**8**:E373–7.
4. Lee S-A, Mun H-S, Kim H, *et al*. Naturally occurring hepatitis B virus X deletions and insertions among Korean chronic patients. *J Med Virol* 2011;**83**:65–70.
5. Väli U, Brandström M, Johansson M, *et al*. Insertion-deletion polymorphisms (indels) as genetic markers in natural populations. *BMC Genet* 2008;**9**:8.
6. Bischoff SR, Tsai S, Hardison NE, *et al*. Identification of SNPs and INDELS in swine transcribed sequences using

short oligonucleotide microarrays. *BMC Genomics* 2008;**9**: 252.

7. Lin MH, Tseng CH, Tseng CC, *et al*. Real-time PCR for rapid genotyping of angiotensin-converting enzyme insertion/deletion polymorphism. *Clin Biochem* 2001;**34**:661–6.
8. Ahmed RPH, Ivaskevicius V, Kannan M, *et al*. Identification of 32 novel mutations in the factor VIII gene in Indian patients with hemophilia A. *Haematologica* 2005;**90**:283–4.
9. Shendure J, Ji H. Next-generation DNA sequencing. *Nat Biotechnol* 2008;**26**:1135–45.
10. Ding L, Ellis MJ, Li S, *et al*. Genome remodelling in a basal-like breast cancer metastasis and xenograft. *Nature* 2010;**464**:999–1005.
11. Krawitz P, Rödelsperger C, Jäger M, *et al*. Microindel detection in short-read sequence data. *Bioinformatics* 2010;**26**: 722–9.
12. Lunter G. Probabilistic whole-genome alignments reveal high indel rates in the human and mouse genomes. *Bioinformatics* 2007;**23**:i289–96.
13. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 2009;**25**: 1754–60.
14. Novoalign software (Novocraft Technologies). http://www.novocraft.com. (09 March 2012, date last accessed).
15. Tuzun E, Sharp AJ, Bailey JA, *et al*. Fine-scale structural variation of the human genome. *Nat Genet* 2005;**37**: 727–32.
16. Koboldt DC, Chen K, Wylie T, *et al*. VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* 2009;**25**:2283–5.
17. Albers CA, Lunter G, Macarthur DG, McVean G, Ouwehand WH, Durbin R. Dindel: accurate indel calls from short-read data. *Genome Res* 2011;**21**(6):961–73.
18. Li H, Handsaker B, Wysoker A, *et al*. The sequence alignment/map format and SAMtools. *Bioinformatics* 2009;**25**: 2078–9.
19. McKenna A, Hanna M, Banks E, *et al*. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* 2010; **20**:1297–303.
20. Qi J, Zhao F, Buboltz A, *et al*. inGAP: an integrated next-generation genome analysis pipeline. *Bioinformatics* 2010;**26**:127–9.
21. Jorde LB, Wooding SP. Genetic variation, classification and ''race''. *Nat Genet* 2004;**36**:S28–33.
22. Mills RE, Luttig CT, Larkins CE, *et al*. An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Res* 2006;**16**:1182–90.
23. Clark TG, Andrew T, Cooper GM, *et al*. Functional constraint and small insertions and deletions in the ENCODE regions of the human genome. *Genome Biol* 2007;**8**:R180.
24. Voelkerding KV, Dames SA, Durtschi JD. Next-generation sequencing: from basic research to diagnostics. *Clin Chem* 2009;**55**:641–58.
25. Zhang Z, Gerstein M. Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res* 2003;**31**:5338–48.
26. Hillier LW, Marth GT, Quinlan AR, *et al*. Whole-genome sequencing and variant discovery in *C. elegans*. *Nat Methods* 2008;**5**:183–8.