OXFORD

Genome analysis

MetaSV: an accurate and integrative structural-variant caller for next generation sequencing

Marghoob Mohiyuddin[1,†], John C. Mu[1,†], Jian Li[1], Narges Bani Asadi[1], Mark B. Gerstein[2], Alexej Abyzov[3], Wing H. Wong[4,5] and Hugo Y.K. Lam[1,*]

[1]Bina Technologies, Roche Sequencing, Redwood City, CA 94065, USA, [2]Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06520, USA, [3]Department of Health Sciences Research, Center for Individualized Medicine, Mayo Clinic, Rochester, MN 55905, USA, [4]Department of Statistics, Stanford University, Stanford, CA 94035, USA and [5]Department of Health Research and Policy, Stanford University, Stanford, CA 94035, USA

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Inanc Birol

Abstract

Summary: Structural variations (SVs) are large genomic rearrangements that vary significantly in size, making them challenging to detect with the relatively short reads from next-generation sequencing (NGS). Different SV detection methods have been developed; however, each is limited to specific kinds of SVs with varying accuracy and resolution. Previous works have attempted to combine different methods, but they still suffer from poor accuracy particularly for insertions. We propose MetaSV, an integrated SV caller which leverages multiple orthogonal SV signals for high accuracy and resolution. MetaSV proceeds by merging SVs from multiple tools for all types of SVs. It also analyzes soft-clipped reads from alignment to detect insertions accurately since existing tools underestimate insertion SVs. Local assembly in combination with dynamic programming is used to improve breakpoint resolution. Paired-end and coverage information is used to predict SV genotypes. Using simulation and experimental data, we demonstrate the effectiveness of MetaSV across various SV types and sizes.

Availability and implementation: Code in Python is at http://bioinform.github.io/metasv/.

Contact: rd@bina.com

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

1 Introduction

SVs have been implicated in contributing to genomic diversity as well as genomic disorders (Stankiewicz and Lupski, 2010). Therefore, a significant amount of work has been done on detecting SVs. Generally, a tool for detecting SVs uses one or more of the following signals from read alignments: split-read [reads with split alignments, e.g. Pindel (Ye *et al.*, 2009)], read-pair [abnormal

paired-end alignments, e.g. BreakDancer (Chen *et al.*, 2009)], depth-of-coverage [abnormal coverages, e.g. CNVnator (Abyzov *et al.*, 2011)], junction-mapping [alignments to known SV breakpoints, e.g. BreakSeq2 (Abyzov *et al.*, 2015; Lam *et al.*, 2010)] or assembly around potential breakpoints [e.g. MindTheGap (Rizk *et al.*, 2014)]. However, there is no signal that comprehensively detects all types of SVs since each has a niche of SV types and sizes where it

works well. This necessitates the development of tools which integrate multiple methods to improve SV detection.

Prior work (Lam *et al.*, 2012; Mills *et al.*, 2011) has shown that variant calls made by multiple tools and methods generally are more accurate. For this reason, tools have been developed to employ multiple methods, e.g. DELLY (Rausch *et al.*, 2012), LUMPY (Layer *et al.*, 2014) and those that merge the results from multiple tools, such as SVMerge (Wong *et al.*, 2010). However, LUMPY and DELLY are unable to detect insertions and SVMerge ignores the SV resolution of individual tools when merging calls. Our work, therefore, attempts to address the limitations of existing SV merging tools for detecting SVs of different types and sizes with high accuracy and resolution.

2 Methods

MetaSV uses multiple SV-detection methods and tools to find a high-confidence and precise SV callset. The novelty of MetaSV lies in the combination of the following key ideas: calls reported by multiple orthogonal methods are generally better quality and that local assembly with dynamic programming can be used to refine the SV breakpoints.

2.1 Multi-method SV detection
MetaSV proceeds in the following steps (Fig. 1):

- **Intra-tool merging:** Potential duplicate calls generated by the same tool are merged here. Note that two calls are considered duplicate if they have significant overlap.
- **Inter-tool merging:** Calls which are generated by multiple tools are merged together. While determining the breakpoints for calls common to multiple tools, priority is given to methods known to be precise, e.g. split-read over read-pair. Note that this method-aware merging is crucial to ensuring that the breakpoints of the SVs reported are as precise as possible.
- **Local assembly:** Local assembly is performed on the SV regions reported by the tools to gather additional evidence as well as determine the SV sequences. The SPAdes (Bankevich *et al.*, 2012) assembler is used due to its unique ability to use paired-end information for assembly.
- **Breakpoint resolution:** The assembled SV sequences are aligned against the reference to detect or refine the breakpoints using dynamic programming (Abyzov and Gerstein, 2011).
- **Genotyping:** Read coverage around the SV breakpoints are used to determine the zygosity of the SVs. The final output is then generated as a VCF file with the genotypes for each SV.
- **Annotation:** MetaSV standardizes the inputs as well as the outputs in VCF. Each SV is annotated to indicate the corresponding calls made by the individual tools and to classify its confidence level. SVs which are detected by multiple tools, are considered high-confidence.

2.2 Insertion detection enhancement
The overall sensitivity of MetaSV by simply merging calls from multiple tools is upper bounded by the sensitivity of the union of all SVs detected by the individual tools. Therefore, for long insertions, which are underestimated by existing tools due to ascertainment bias, we augmented MetaSV with a soft-clip based method to boost insertion detection sensitivity. Soft-clips in read alignments are used to generate a set of candidate insertion intervals. These intervals are processed during the local assembly step to generate the final set of
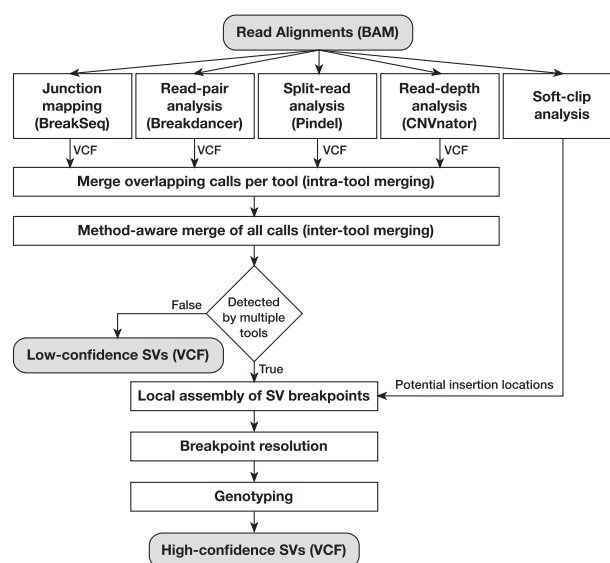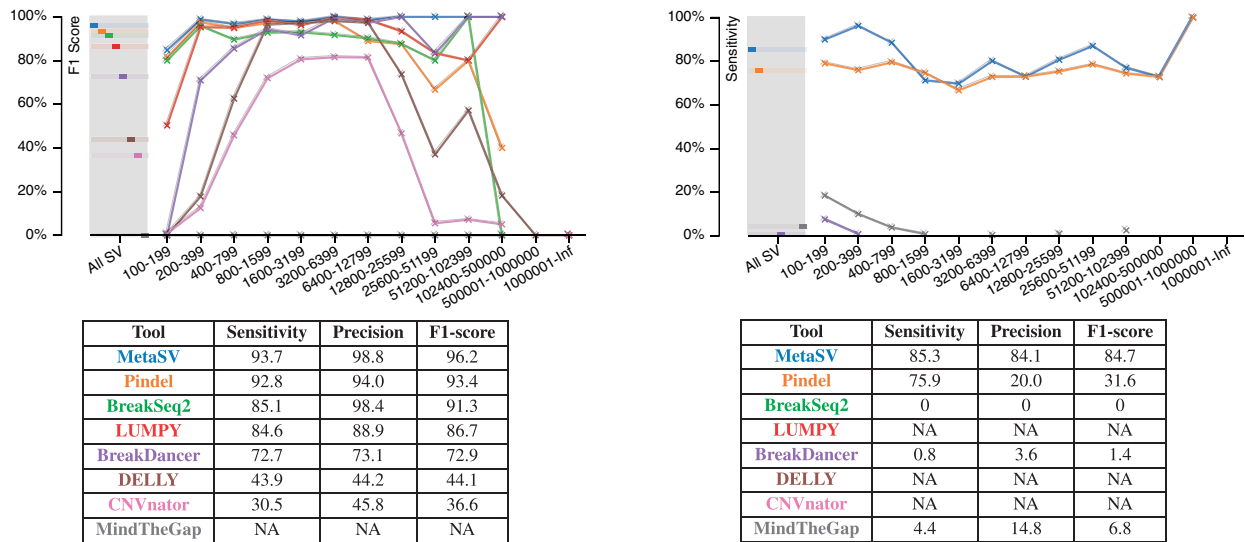

Fig. 1. High-level view of the MetaSV methodology

insertion locations. Even though assembly would not be able to determine insertion lengths for long insertions due to short read length, their locations can still be predicted accurately, which provides valuable information for interpretation. The Supplementary Text describes our method in more detail.

3 Results

We demonstrate the effectiveness of MetaSV using the VarSim simulation and validation framework (Mu *et al.*, 2014). Simulated 2×100 bp NGS reads were generated at $50\times$ coverages for the NA12878 genome using published variant sets. The reads were aligned using BWA-MEM. For comparing reported SVs against the ground truth, we use a reciprocal overlap of 90% and a wiggle of 100 bp which captures accuracy at a high breakpoint resolution. The SV size cutoff was set to 100 bp since smaller variants are a target of indel callers such as GATK's HaplotypeCaller. Our results show that each method has varying performance in different SV size ranges. By integrating multiple methods, MetaSV achieved a steady performance across all sizes (Fig. 2). We report accuracy as F1-score, which is the harmonic mean of sensitivity and precision. For this dataset, MetaSV achieved an F1-score of 96.2% (sensitivity and precision were 93.7 and 98.8%, respectively) for deletions, indicating high accuracy and resolution. For insertions, it achieved an F1-score of 84.7% (sensitivity and precision were 85.3 and 84.1%, respectively) comparing to less than 65% for all the individual tools analyzed. Insertion length was omitted from the accuracy analysis since long insertions cannot be assembled completely with NGS reads. Nevertheless, the significantly enhanced detection of insertion events can definitely improve interpretation largely as they may cause impactful disruption in the genome. Finally, genotyping accuracies of 95.2 and 95.5% were achieved for deletions and insertions, respectively.

Figure 2 shows detailed accuracy comparisons for both deletion and insertion detection across different SV sizes and tools. For deletions (Fig. 2a), MetaSV performance was the highest across all SV sizes. In most cases, it improves upon the best performing individual tool for a given size. For insertions (Fig. 2b), the improvement due to MetaSV was more significant since all tools (with the exception

Tool	Sensitivity	Precision	F1-score
MetaSV	93.7	98.8	96.2
Pindel	92.8	94.0	93.4
BreakSeq2	85.1	98.4	91.3
LUMPY	84.6	88.9	86.7
BreakDancer	72.7	73.1	72.9
DELLY	43.9	44.2	44.1
CNVnator	30.5	45.8	36.6
MindTheGap	NA	NA	NA

(a) Deletion accuracy summary for different tools for 50× coverage.

Tool	Sensitivity	Precision	F1-score
MetaSV	85.3	84.1	84.7
Pindel	75.9	20.0	31.6
BreakSeq2	0	0	0
LUMPY	NA	NA	NA
BreakDancer	0.8	3.6	1.4
DELLY	NA	NA	NA
CNVnator	NA	NA	NA
MindTheGap	4.4	14.8	6.8

(b) Insertion accuracy summary for different tools for 50× coverage.

Fig. 2. Accuracy comparisons for deletions and insertions. Accuracy metrics are shown on a per size bin basis in the plots. The tables below the plots show the aggregate accuracy scores. If a tool does not support detecting the SV type, an NA is indicated in the table. Each tool name is color coded to match the color code in the plots. DELLY's suboptimal deletion performance was due to its lower breakpoint resolution. For insertions, although Pindel's sensitivity was close to MetaSV, it had a significantly lower precision and overall accuracy

of Pindel) almost detect no insertions due to inherent limitations of the methods used. Therefore, almost all improvement in accuracy is due to our insertion detection enhancement—our soft-clip based approach is very sensitive while the assembly step for insertion detection yields high precision in contrast to Pindel which had low precision. As with deletions, small insertions are more difficult to detect, in general. Detailed accuracy comparisons for other SV types are discussed in the Supplementary Text.

In order to study the impact on accuracy and runtime of varying coverages, we generated additional simulation datasets with 2×100 bp reads at 10× and 30× coverages. We also generated 2×250 bp reads at 50× coverage to investigate the impact of increasing read length for the same coverage. Although accuracy dropped for lower coverages, MetaSV was still the most stable and most accurate, with deletion F1-scores of 89.1 and 95.8% for 10× and 30× coverages, respectively. For 250 bp read length, F1-scores of 96.8 and 80.9% were achieved for deletions and insertions, respectively—insertion accuracy dropped slightly over 100 bp reads due to reduced read count for the same coverage. Furthermore, it took around 25 h to run all the four aforementioned tools for MetaSV as well as MetaSV on a single node with dual-hexcore X5675 Intel Xeon processors for 50× coverage. Because MetaSV is parallelized in all its steps, its speed should scale linearly with the number of available cores.

In addition to simulation, we used the publicly available Illumina Platinum Genomes sequencing data for NA12878 as a real testbed. Due to the lack of high-confidence comprehensive SV calls, particularly for insertions, false positive rates cannot be accurately determined using real data. Therefore, only sensitivity for deletions was reported here. For deletions, a sensitivity of 90.2% was achieved against the Complete Genomics high-confidence callset for NA12878 (Drmanac *et al.*, 2009) generated using their analysis pipeline version 2.0, which is similar to our simulation results. Complete Genomics was used since it is an orthogonal sequencing platform, providing a less biased validation.

4 Conclusions

MetaSV significantly improves the accuracy of SV-calling by integrating orthogonal methods and tools. In addition, it is augmented with soft-clip based insertion detection for significantly higher accuracy compared with the state of the art.

We consider MetaSV as a proof of concept of the effectiveness of using an ensemble approach for calling SVs. The approach is not limited to only using the four aforementioned tools—it can be easily adapted to use additional or even a different set of tools.

References

Abyzov,A. and Gerstein,M. (2011) AGE: defining breakpoints of genomic structural variants at single-nucleotide resolution, through optimal alignments with gap excision. *Bioinformatics*, **27**, 595–603.

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.

Abyzov,A. *et al.* (2015) Analysis of deletion breakpoints from 1,092 humans reveals details of mutation mechanisms. *Nat. Commun.*, **6**, 7256, doi: 10.1038/ncomms8256.

Bankevich,A. *et al.* (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.*, **19**, 455–477.

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

Drmanac,R. *et al.* (2009) Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science*, **327**, 78–81.

Lam,H.Y. *et al.* (2010) Nucleotide-resolution analysis of structural variants using BreakSeq and a breakpoint library. *Nat. Biotechnol.*, **28**, 47–55.

Lam,H.Y. *et al.* (2012) Detecting and annotating genetic variations using the HugeSeq pipeline. *Nat. Biotechnol.*, **30**, 226–229.

Layer,R.M. *et al.* (2014) LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.*, **15**, R84.

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

Mu,J.C. *et al.* (2014) VarSim: a high-fidelity simulation and validation framework for high-throughput genome sequencing with cancer applications. *Bioinformatics*, **31**, 1469–1471.

Rausch,T. *et al.* (2012) DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, **28**, i333–i339.

Rizk,G. *et al.* (2014) MindTheGap: integrated detection and assembly of short and long insertions. *Bioinformatics*, **30**, 3451–3457.

Stankiewicz,P. and Lupski,J.R. (2010) Structural variation in the human genome and its role in disease. *Annu. Rev. Med.*, **61**, 437–455.

Wong,K. *et al.* (2010) Enhanced structural variant and breakpoint detection using SVMerge by integration of multiple detection methods and local assembly. *Genome Biol.*, **11**, R128.

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.