SOFTWARE TOOL ARTICLE

Tools for annotation and comparison of structural variation [version 1; referees: 1 approved, 2 approved with reservations]

Fritz J. Sedlazeck [1], Andi Dhroso[2], Dale L. Bodian[3], Justin Paschall[4], Farrah Hermes[5], Justin M. Zook[6]

[1]Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX, USA
[2]Worcester Polytechnic Institute, Worcester, MA, USA
[3]Inova Translational Medicine Institute, Inova Health System, Falls Church, VA, USA
[4]University of California, Berkeley, Berkeley, CA, USA
[5]Virginia Commonwealth University, Richmond, VA, USA
[6]Genome-scale Measurements Group, National Institute of Standards and Technology, Gaithersburg, MD, USA

v1 **First published:** 03 Oct 2017, **6**:1795 (doi: 10.12688/f1000research.12516.1)
Latest published: 03 Oct 2017, **6**:1795 (doi: 10.12688/f1000research.12516.1)

Abstract

The impact of structural variants (SVs) on a variety of organisms and diseases like cancer has become increasingly evident. Methods for SV detection when studying genomic differences across cells, individuals or populations are being actively developed. Currently, just a few methods are available to compare different SVs callsets, and no specialized methods are available to annotate SVs that account for the unique characteristics of these variant types. Here, we introduce SURVIVOR_ant, a tool that compares types and breakpoints for candidate SVs from different callsets and enables fast comparison of SVs to genomic features such as genes and repetitive regions, as well as to previously established SV datasets such as from the 1000 Genomes Project. As proof of concept we compared 16 SV callsets generated by different SV calling methods on a single genome, the Genome in a Bottle sample HG002 (Ashkenazi son), and annotated the SVs with gene annotations, 1000 Genomes Project SV calls, and four different types of repetitive regions. Computation time to annotate 134,528 SVs with 33,954 of annotations was 22 seconds on a laptop.

Open Peer Review

Referee Status: ? ✓ ?

	Invited Referees 1	2	3
version 1 published 03 Oct 2017	? report	✓ report	? report

1 **Mitchell A. Bekritsky**, Illumina Cambridge Ltd., UK

2 **Andrew Carroll**, DNAnexus, USA

3 **Aaron R. Quinlan**, University of Utah, USA

Discuss this article

Comments (0)

Corresponding author: Fritz J. Sedlazeck (fritz.sedlazeck@bcm.edu)

Author roles: Sedlazeck FJ: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Dhroso A**: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Bodian DL**: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Paschall J**: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Hermes F**: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Zook JM**: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

How to cite this article: Sedlazeck FJ, Dhroso A, Bodian DL *et al.* **Tools for annotation and comparison of structural variation [version 1; referees: 1 approved, 2 approved with reservations]** *F1000Research* 2017, **6**:1795 (doi: 10.12688/f1000research.12516.1)

Grant information: This work was supported by the National Science Foundation awards (DBI-1350041) and National Institutes of Health awards (R01-HG006677 and UM1-HG008898), and by the Inova Health System.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

First published: 03 Oct 2017, **6**:1795 (doi: 10.12688/f1000research.12516.1)

Introduction

The advent of high throughput sequencing (HTS) facilitates the investigation of genomic differences among and within organisms, populations, and even diseases such as cancer. While the identification of single nucleotide polymorphisms (SNPs) is currently well established, structural variant (SV) calling remains challenging and little is known about the sensitivity (correctly inferring SVs) and false discovery rate (FDR) (falsely inferring SVs) of structural variation detection (Guan & Sung, 2016). Recent SV discovery methods, such as LUMPY (Layer *et al.*, 2014) and PBHoney (English *et al.*, 2014), focus on one callset per technology, but SV detection and call evaluation would benefit from comparison of the data from multiple technologies. However, many challenges exist in comparing and merging SV calls due to uncertainty in breakpoints, sequencing errors, and multiple possible representations of SVs in repetitive regions (Wittler *et al.*, 2015). In addition, Sudmant *et al.* (Sudmant *et al.*, 2015) as well as Jeffares *et al.* (Jeffares *et al.*, 2017) mention that the methods often lack sensitivity and suffer from an inestimable FDR. Jeffares *et al.* coped with this problem by merging SV calls generated by multiple callers to reduce the FDR, but this approach also slightly reduced sensitivity (Jeffares *et al.*, 2017).

To enable comparison and evaluation of SV callsets generated by different algorithms, we developed methods to compare and annotate SV calls, represented in variant call format (VCF), with other SVs as well as other genomic features. Genomic features can include gene annotations, mappability tracks, and any feature that can be represented as a region in BED or GFF format. SNPs can also be used for annotation by representing them as regions of 1bp.

As a proof of concept, we apply these novel methods to the Genome in a Bottle (GiaB) data generated on the Ashkenazi son (NIST Reference Material 8391, aka HG002 and NA24385) to explore SV type and breakpoint concordance of SV calling algorithms. GiaB provides SV calls generated using five different technologies (including Illumina short read sequencing, Complete Genomics nanoball sequencing, Pacific Biosciences long read sequencing, 10X Genomics linked reads, and BioNano optical mapping) and 16 different SV calling algorithms on the same genome. We used SURVIVOR (Jeffares *et al.*, 2017) to merge SV calls and our novel method (SURVIVOR_ant) to annotate and predict more precise breakpoints. All data sets (including merged SV and annotated SV) and methods used in this manuscript are available at https://github.com/NCBI-Hackathons/svcompare.

Methods

Implementation

SURVIVOR_ant. To enable annotation and comparison of the SV callsets, we implemented a new extension of SURVIVOR that aims to assign genomic features, including previously known/established SVs, to merged SV callsets produced by SURVIVOR. SURVIVOR_ant takes any VCF file (list of SVs) as an input (-i) as well as annotation sets specified as a list of BED files (--bed), GFF files (--gff) and additional VCF files (--vcf). Each of the three file types are optional and the user can specify multiple files for each type, separated by commas. SURVIVOR_ant reads in the original VCF file to be annotated and constructs a self-balancing interval tree originally taken from SURVIVOR (Jeffares *et al.*, 2017). Next, it reads in any annotations in VCF files (e.g. from the 1000 Genomes Project) and compares these to the original VCF entries in the interval tree. The comparison is based on the individual breakpoints, given a maximum distance parameter (by default 1kb). Subsequently, SURVIVOR_ant runs through the BED files and GFF files and parses the provided intervals and identifiers. In the case of a GFF file, SURVIVOR parses the first name in the 9th column: gene=. For BED files, SURVIVOR_ant uses the fourth column as the name for each entry (or the file name, if the BED file does not include a four columns).

Each entry of a BED or GFF file is assigned to deletions and duplications in the SURVIVOR_ant VCF if they overlap the SV +/- a user-defined distance/wobble parameter (by default 1kb). For translocations, insertions and inversions, SURVIVOR_ant only takes the breakpoints into account and assigns genomic features within a user-defined distance/wobble parameter (by default 1kb). Figure 1 shows the schematic based on three genes. The distance/wobble parameter is necessary to account for differences in accuracy of the technology, mapping, or the SV calling algorithm. Often breakpoints are positioned in repeated regions which makes it hard to place the breakpoints accurately.

After all files are read in and compared to the original VCF file, SURVIVOR_ant prints the original VCF file and extends the INFO field with information on how many VCF files have supportive information ("overlapped_VCF=") as well as how many genomic features within the VCF files could be assigned per original SV ("total_Annotations="). If SURVIVOR_ant found overlapping genomic features the names associated to these are printed out in a comma separated list ("overlapped_Annotations="). SURVIVOR_ant is maintained at https://github.com/NCBI-Hackathons/svcompare.

Summary statistics scripts. These statistics were generated with code available at https://github.com/NCBI-Hackathons/svcompare, which analyzes the data as structured by data_structures.pl. The R script stats_plots_v2.R can be used to generate a variety of figures like those in this paper.

SV analyses. Several statistics were computed by event, by call set, and by variant type:

1. The number of callers supporting an event is the count of the number of callers for which SURVIVOR identified a variant call at the same location for an event, subject to the 1 kb wobble parameter and independent of the variant type.

2. For the total number of variant calls per callset by variant type, the SVs of each type were counted separately. For events in which a single caller made more than one call (sub-calls), each sub-call is counted separately (e.g., if a callset has two different deletions that SURVIVOR merged into a single VCF row, it is counted as two deletions).

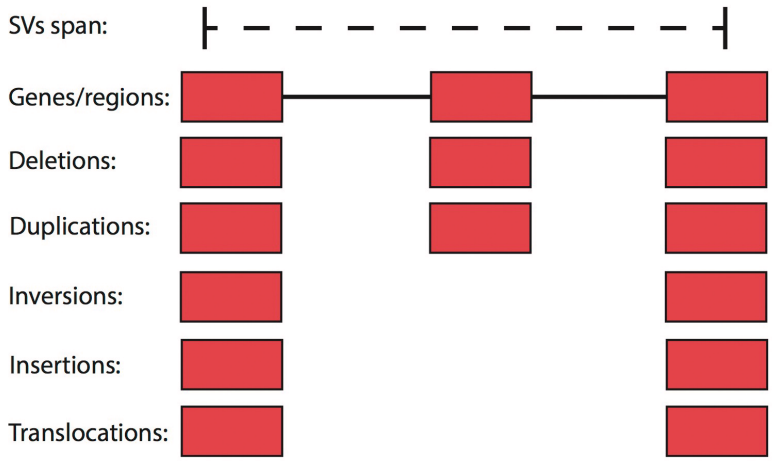

Figure 1. SV type specific overlap schema of SURVIVOR_ant to identify which genomic annotations overlaps with which type of SV. By default SURVIVOR_ant takes 1kbp surrounding the start and stop coordinates into account. Furthermore, for deletions and duplications we take the overlapping regions into account.

3. For analysis of the SURVIVOR_ant annotations, the number of events with at least one annotation of the selected type were counted.
4. Breakpoints were compared for events supported by at least four callers, and for which all calls were of the same type. The type was examined for all calls supporting that event, including multiple calls from any one caller. First, the median start and end positions for were calculated for all calls for that event. Second, the distance of each call's start from the median start, and each call's end from the median end were computed. For calls with multiple sub-calls from a single caller, the minimum start position and maximum end position were used as the start and end, respectively.

Operation

SURVIVOR_ant is based on C++ and does not require any preinstalled packages or libraries. The analysis scripts are using BioPerl (http://bioperl.org/).

Genomic data. We used 16 candidate-SV callsets from the GiaB Ashkenazi son data set available at https://github.com/genome-in-a-bottle. We reformatted the files if they did not correspond to the VCF 4.1 standard then merged the SVs in the 16 files into one multi-sample VCF using SURVIVOR (Jeffares *et al.*, 2017) with 1 kb as the distance parameter and without requiring type specificity. Next we downloaded three BED files defining repetitive regions from the GA4GH Benchmarking Team at https://github.com/ga4gh/benchmarking-tools/tree/master/resources/stratification-bed-files. Furthermore, we downloaded the gene annotations for GRCh37 from ensembl. Population genomic data was downloaded for the 1000 Genomes Project from dbVar (estd219) (Sudmant *et al.*, 2015) and filtered to produce a unique set of variant sites. SURVIVOR_ant (Version 0.0.1) was used to annotate the merged SVs with all the annotation data sets. The merged SVs are referred to as "events."

Results

We merged the output of 16 different callers containing variants >19 bp (Table 1) that were run on the Ashkenazi son data (GiaB) using SURVIVOR (Jeffares *et al.*, 2017). The resulting VCF file contained 134,528 SV events and was annotated by our novel method SURVIVOR_ant. We annotated the SVs with genes from hg19 (GFF), the 1000 Genomes project (dbVar) population-based structural variant calls, and repetitive regions from the GA4GH Benchmarking Team (3 bed files). SURVIVOR_ant compared the five files to the merged SV calls for the Ashkenazi son data within 22 seconds. It identified 4,506 overlapping SVs between the 1000 Genomes Project and our data set (Table 2). Furthermore, SURVIVOR_ant identified genomic features in the 3 BED files and the GFF gene list overlapping 66,166 SVs out of the total merged 134,528 VCF entries.

The SURVIVOR_ant output is also useful for comparing the output of callers. Each caller assigns an SV type (e.g., insertion, deletion, translocation, etc.) and breakpoints for each SV call. Overall we identified 125,909 (93.6%) SVs that were supported by fewer than four callsets. Figure 2 depicts the widely varying number of candidate SV calls of different types across callsets, which contributes to the large fraction of calls that are supported by fewer than four callsets. Of 134,528 calls from the Ashkenazi son data from the callers in Table 1, 11,474 (8.5%) had more than one SV type discovered by different callers in the same region, and 6,280 (4.7%) had more than one SV type discovered by the same caller in the same region. It is possible either that these different types are due to errors in the calls or that there is a true complex SV consisting of multiple nearby SV types. In addition, duplications of a large region in tandem could be described as an insertion

Table 1. Structural variant callsets for the Ashkenazi son.

Sequencing Technology	Structural Variant Caller	Call Set Name	Reference
Illumina	Mobile Element Insertion finder - no current name	HG002.TE_insertions.recover_filt_mod	(Hénaff *et al.*, 2015)
Illumina	CommonLaw	HG002.commonlaw.deletions.bilkentuniv.082815	(Zhao *et al.*, 2013)
Illumina	FermiKit	HG002.fermikit.sv	(Li, 2015)
Illumina	FreeBayes	HG002_ALLCHROM_hs37d5_novoalign_Ilmn150bp300X_FB_delgt19	(Garrison & Marth, 2012)
Illumina	GATK Haplotype Caller	HG002_ALLCHROM_hs37d5_novoalign_Ilmn150bp300X_GATKHC_delgt19	(McKenna *et al.*, 2010)
Illumina	CNVnator	HG002_CNVnator_deletions.hs37d5.sort	(Abyzov *et al.*, 2011)
Illumina	MetaSV	MetaSV_151207_variants	(Mohiyuddin *et al.*, 2015)
PacBio	Assemblytics	hg002.Assemblytics_structural_variants	(Nattestad & Schatz, 2016)
PacBio	MultibreakSV	hg002_attempt1.1_MultibreakSV_mod	(Ritz *et al.*, 2014)
PacBio	Parliament - forced Illumina assembly	parliament.assembly.H002	(English *et al.*, 2015)
PacBio	Parliament - forced PacBio call	parliament.pacbio.H002	(English *et al.*, 2015)
PacBio	smrt-sv	smrt-sv.dip_indel	(Chaisson *et al.*, 2015)
PacBio	Assemblytics	trio2.Assemblytics_structural_variants	(Nattestad & Schatz, 2016)
PacBio	PBHoney	PBHoney_15.8.24_HG002.tails_20	(English *et al.*, 2014)
Complete Genomics	Complete Genomics	vcfBeta-GS000037263-ASM_delgt19	(Carnevali *et al.*, 2012)
Bionano		son_hap_refsplit20160129_1kb	(Mak *et al.*, 2016)

Table 2. Summary over the overlapping annotation for the SVs data set.

Annotation type	# of overlapping SVs
Ensembl genes	22,184
Repeats	7,264
1000 genomes SVs	4,506

by some callers and as a duplication by other callers. These results illustrate the disagreement of multiple callers over the same data set, as well as the complexity of integrating calls from different methods.

For characterization of the consistency of breakpoint prediction of the different callers, we analyzed the 5,386 SV events with support from at least four callsets, and for which all calls are of the same type, so that a useful median start and end position could be calculated. Figure 3 depicts example histograms of distance to the median start position for two callsets, one from long reads and one from short reads. In general, more of the short read caller's start positions are closer to the median breakpoint, but this could be due to a variety of factors, including lower error rates in short reads, easier less repetitive sites detected by short reads, filtering rules, etc. Note that since the number of callers per technology varies and calls supported by more callsets are likely to be easier to detect, this likely introduces a bias in the variants assessed. We calculated these statistics as an example of using our methods, not as a generalizable estimate of breakpoint accuracy.

Conclusions and future work

In this paper, we introduced SURVIVOR_ant, an annotation and comparison tool especially designed for comparing SVs and genomic features (e.g. genes). SURVIVOR_ant is novel in that it enables a type-specific comparison to multiple genomic annotations and other features of interest. The resulting VCF file can be loaded in existing methods such as IGV or bedtools for further manual inspections. SURVIVOR_ant and all resources used here are available at https://github.com/NCBI-Hackathons/svcompare. This tool is an important first step to enable the comparison of SVs to each other, to known SVs, and to genomic features. Here, we defined genomic features as being information about the properties of the underlying genome sequence (e.g., repetitive regions), as well as annotations such as genes or even chromatin assays. Furthermore, we have made available scripts to

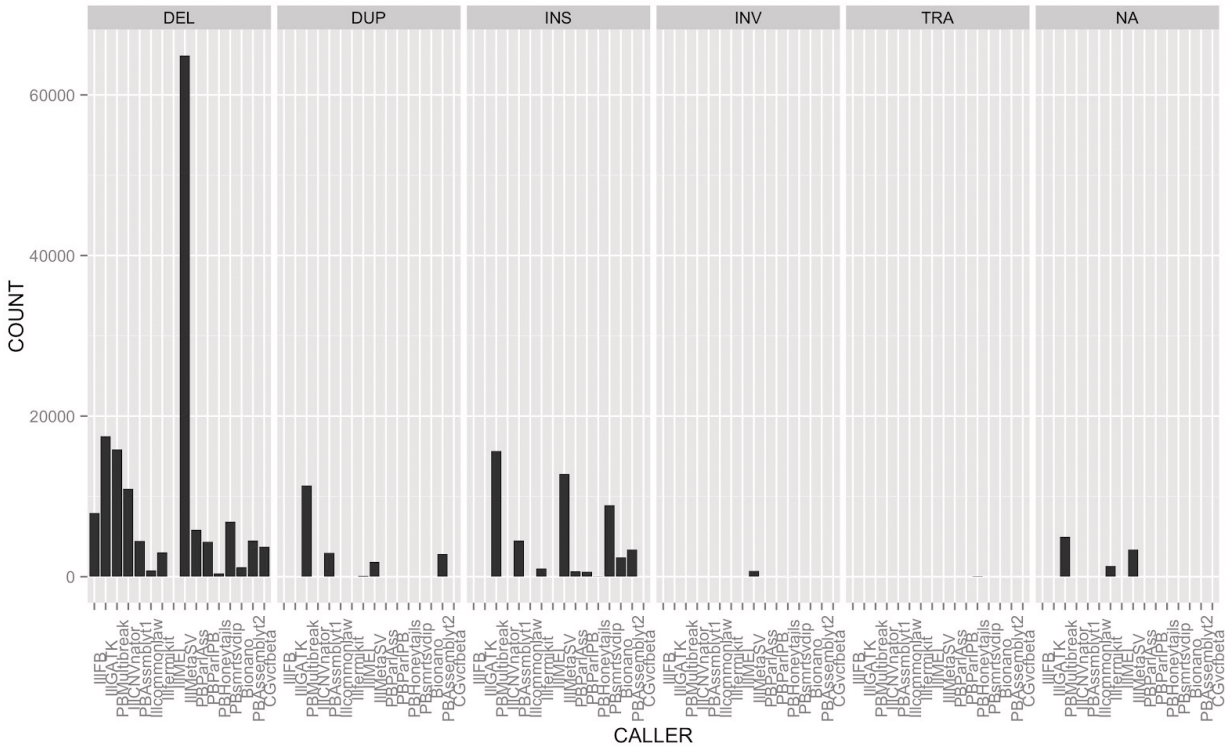

Figure 2. Number of calls per callset for each type of SV, including filtered calls.

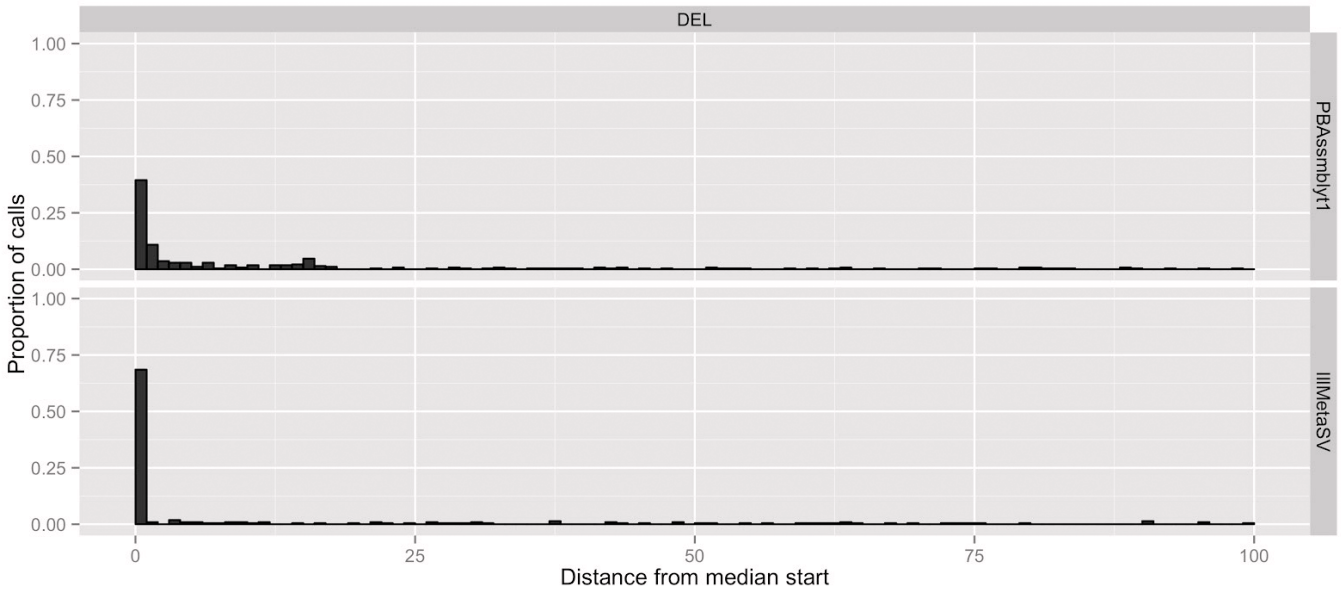

Figure 3. Histogram of distance from the median start position for deletion calls 400 to 999 bp in size for a PacBio-based Assemblytics callset and for an Illumina-based MetaSV callset. Only sites with calls from at least 4 different callsets were included in order to calculate a useful median value at each site.

calculate a variety of statistics that characterize the similarity and differences between many callsets from a single genome, including the number of callsets supporting similar calls in a region and concordance between their breakpoints.

Future work will include estimating the underlying breakpoints for each SV, potentially based on machine learning methods that utilize information gained from the GiaB consortium on the accuracy of different technologies for different SVs types and sizes. In addition, future work will involve comparing predicted SVs in repetitive regions, since these can often be represented in multiple ways in multiple locations in the genome.

In summary, we present a method (SURVIVOR_ant) for fast annotation of SVs and represents a first step in understanding type and breakpoint concordance for any type of SV, as well as the potential impact of SVs on genes.

Data and software availability

All the datasets used in this study are available at https://github.com/NCBI-Hackathons/svcompare.gi. Additional raw data can be obtained at https://github.com/genome-in-a-bottle.

Archived source code of the software used as at the time of publication is available at: http://doi.org/10.5281/zenodo.898078 (dbodian *et al.*, 2017)

Competing interests

No competing interests were disclosed.

Grant information

This work was supported by the National Science Foundation awards (DBI-1350041) and National Institutes of Health awards (R01-HG006677 and UM1-HG008898), and by the Inova Health System.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements

We thank Timothy Hefferon for advice on dbVar datasets. Furthermore, we thank Ben Busby and the NCBI Hackathon Team of August 2016 for helpful discussions. We thank Lisa Federer, NIH Library, for editing assistance. We thank members of the Genome in a Bottle Consortium for generating the data and SV calls used in this study. Certain commercial equipment, instruments, or materials are identified in this paper only to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

Abyzov A, Urban AE, Snyder M, *et al.*: **CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing.** *Genome Res.* 2011; **21**(6): 974–984.
PubMed Abstract | Publisher Full Text | Free Full Text

Carnevali P, Baccash J, Halpern AL, *et al.*: **Computational techniques for human genome resequencing using mated gapped reads.** *J Comput Biol.* 2012; **19**(3): 279–292.
PubMed Abstract | Publisher Full Text

Chaisson MJ, Huddleston J, Dennis MY, *et al.*: **Resolving the complexity of the human genome using single-molecule sequencing.** *Nature.* 2015; **517**(7536): 608–611.
PubMed Abstract | Publisher Full Text | Free Full Text

dbodian, adhroso, Sedlazeck F, *et al.*: **NCBI1.5-Hackathons/svcompare: Initial release.** *Zenodo.* 2017.
Data Source

English AC, Salerno WJ, Hampton OA, *et al.*: **Assessing structural variation in a personal genome-towards a human reference diploid genome.** *BMC Genomics.* 2015; **16**(1): 286.
PubMed Abstract | Publisher Full Text | Free Full Text

English AC, Salerno WJ, Reid JG: **PBHoney: identifying genomic variants via long-read discordance and interrupted mapping.** *BMC Bioinformatics.* 2014; **15**: 180.
PubMed Abstract | Publisher Full Text | Free Full Text

Garrison E, Marth G: **Haplotype-based variant detection from short-read sequencing**. 2012.
Reference Source

Guan P, Sung WK: **Structural variation detection using next-generation sequencing data: A comparative technical review.** *Methods.* 2016; **102**: 36–49.
PubMed Abstract | Publisher Full Text

Hénaff E, Zapata L, Casacuberta JM, *et al.*: **Jitterbug: somatic and germline transposon insertion detection at single-nucleotide resolution.** *BMC Genomics.* 2015; **16**: 768.
PubMed Abstract | Publisher Full Text | Free Full Text

Jeffares DC, Jolly C, Hoti M, *et al.*: **Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast.** *Nat Commun.* 2017; **8**: 14061.
PubMed Abstract | Publisher Full Text | Free Full Text

Layer RM, Chiang C, Quinlan AR, *et al.*: **LUMPY: a probabilistic framework for structural variant discovery.** *Genome Biol.* 2014; **15**(6): R84.
PubMed Abstract | Publisher Full Text | Free Full Text

Li H: **FermiKit: assembly-based variant calling for Illumina resequencing data.** *Bioinformatics.* 2015; **31**(22): 3694–3696.
PubMed Abstract | Publisher Full Text | Free Full Text

Mak AC, Lai YY, Lam ET, *et al.*: **Genome-Wide Structural Variation Detection by Genome Mapping on Nanochannel Arrays.** *Genetics.* 2016; **202**(1): 351–362.
PubMed Abstract | Publisher Full Text | Free Full Text

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**(9): 1297–1303.
PubMed Abstract | Publisher Full Text | Free Full Text

Mohiyuddin M, Mu JC, Li J, *et al.*: **MetaSV: an accurate and integrative structural-variant caller for next generation sequencing.** *Bioinformatics.* 2015; **31**(16): 2741–2744.
PubMed Abstract | Publisher Full Text | Free Full Text

Nattestad M, Schatz MC: **Assemblytics: a web analytics tool for the detection of variants from an assembly.** *Bioinformatics.* 2016; **32**(19): 3021–3023.
PubMed Abstract | Publisher Full Text

Ritz A, Bashir A, Sindi S, *et al.*: **Characterization of structural variants with single molecule and hybrid sequencing approaches.** *Bioinformatics.* 2014; **30**(24): 3458–3466.
PubMed Abstract | Publisher Full Text | Free Full Text

Sudmant PH, Rausch T, Gardner EJ, *et al.*: **An integrated map of structural variation in 2,504 human genomes.** *Nature.* 2015; **526**(7571): 75–81.
PubMed Abstract | Publisher Full Text | Free Full Text

Wittler R, Marschall T, Schönhuth A, *et al.*: **Repeat- and error-aware comparison of deletions.** *Bioinformatics.* 2015; **31**(18): 2947–2954.
PubMed Abstract | Publisher Full Text

Zhao M, Wang Q, Wang Q, *et al.*: **Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives.** *BMC Bioinformatics.* 2013; **14 Suppl 11**: S1.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Referee Status: ? ✔ ?

Version 1

Referee Report 01 November 2017

doi:10.5256/f1000research.13552.r26610

? **Aaron R. Quinlan**
Departments of Human Genetics and Biomedical Informatics, USTAR (Utah Science Technology and Research) Center for Genetic Discovery, University of Utah, Salt Lake City, UT, USA

Detecting variation in genome structure is notoriously difficult with existing sequencing technologies and algorithms. Consequently, researchers typically utilize multiple SV detection algorithms, and in some cases, multiple sequencing technologies to maximize accuracy.

Sedlazeck and colleagues introduce SURVIVOR_ant as a new tool in the SURVIVOR package for annotating a SV callset in VCF format with SV predictions from other tools as well as genome features in BED or GFF format. While the motivation, performance and implementation are sound, I think the authors need to compare their software to both bedtools intersect and especially vcfanno given that vcfanno already provides all or nearly all of the functionality described here. In particular, vcfanno was designed to properly annotate SV events, as it takes the confidence interval about an SV's predicted breakpoints into account. I suggest that the authors extend the manuscript to provide a paragraph comparing the functionality and speed of SURVIVOR_ant to that of vcfanno and bedtools intersect, as this will help readers to better understand its strengths and weaknesses with respect to existing solutions.

Secondly, I think the authors should state up front exactly what types of events they consider to be SVs, as some call sets listed in Table one restricted to SNPs and short INDELs, which are typically driven by different mutational mechanisms and typically not referred to as SVs.

References

1. Pedersen BS, Layer RM, Quinlan AR: Vcfanno: fast, flexible annotation of genetic variants.*Genome Biol*. 2016; **17** (1): 118 PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Referee Report 16 October 2017

doi:10.5256/f1000research.13552.r26611

Andrew Carroll
DNAnexus, Mountain View, CA, USA

This work deals with the methods to compare and overlap structural variant events and presents an analysis of these methods against one of the best studied single genome for structural variant calling - the Genome in a Bottle HG002 - and the extensive population-level structural variant callset from the 1000 Genomes Project. Although in SNP and Indels the problem of determining whether two representations of a call are the same event, this problem is significantly more difficult more structural variant events. In the current state of the field, this is an open and challenging problem.

The presented analyses are valid. Any review of this work can point out an almost infinite number of parameters for the analysis that can be argued over, such as:

What are reasonable size ranges to merge events?
What are proper criteria for filtering events?
Is it appropriate to include calls from so many methods in merging - should we identify some as unreliable?
Should certain callers be weighted more highly than other based on reliability?
How do we consider orthogonal support from different methods - are 3 Illumina methods based on different signals (coverage, insert size) as good as 2 Pacbio?
Are the differences in how we discover events types (INS/TRA/DEL) significant enough that we need new comparison methods for some?
What are we missing, but we're not aware we're missing. Do the current methods have any major blindspots?

Etc... This list could be nearly infinite, mostly because so many questions in this field are unsettled.

The most important contribution of this work is as a foundation that frames these discussion points. The ability to point to data around a strategy for overlapping and analysis is an important step in progressing beyond discussion of all the items that can be considered.

The results may be valuable to these tool developers to identify structural variant types, sizes, and genomic contexts which their methods perform poorly on. In addition, the annotation ability may be useful in the annotation and interpretation of structural variant events (one could imagine making BED files for genome regions identified in ExAC or gnomAD as important for and scanning samples or structural variant events that look unusual or damaging in an individual or which impact genes associated with phenotypes of interest).

The number of possible future directions for this work is unusually large (the questions listed above are a good start).

There are a number of decisions about the approach that I disagree with and would do differently, but to determine which of those opinions/approaches are valid would require work to the scope of being one (or many) additional publications.

I look forward to community efforts which continue to refine the methods to compare and annotate structural variant calls building on the concepts outlined here.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: Employee at DNAnexus

Referee Expertise: Structural variant calling, short-read analysis, long-read analysis, benchmarking methods for NGS tools

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Referee Report 09 October 2017

doi:10.5256/f1000research.13552.r26608

Mitchell A. Bekritsky
Illumina Cambridge Ltd., Little Chesterford, UK

In this paper, Sedlazeck et al. describe a tool, SURVIVOR_ant, that uses interval trees to intersect multiple SV sets and annotations to produce a single VCF that can be used to compare SV callsets to each other, to known SV datasets, or to other reference annotations. The use of a 'wobble' parameter described in the manuscript enables them to overlap features that are nearby, but may not directly overlap. This is particularly important because SV representations can be pretty variable, although I wouldn't mind seeing a little more direct evidence from the authors motivating the need for this parameter.

The authors demonstrate an analysis possible with SURVIVOR_ant on 16 HG002 SV callsets, along with the 1000 Genomes Project Phase 3 SV calls, and several reference annotation datasets. While they demonstrate that their analysis produces interesting results, some of their observations might benefit from a little more analysis to round them out. In addition, one of the more interesting points that is apparent in one of the figures is not mentioned in the text, which might be worth bringing up. While I realize this manuscript is not intended to be a thorough analysis of the HG002 SV callsets, it is a very good opportunity for the authors to give their readers some interesting insights into these datasets, and could also help point SURVIVOR_ant users in the direction of some useful questions they can ask once they've run this tool.

Although no direct analog to SURVIVOR_ant currently exists, some of its functionality overlaps with bedtools intersect. It seems as though their analysis pipeline might be similar to extending the input SV callset by the wobble parameter on either side, translating it into a BED file, then intersecting it with the other datasets. I think SURVIVOR_ant is distinct enough in its focus on SV in VCFs without further intermediate steps that it's a meaningfully distinct tool.

Software
The code for SURVIVOR_ant is written in C++. Some of it could be organized a little better (for instance, the SVS_Node class was in Parser.h, which I found a little confusing). When I tried to compile the code on my Mac, it did not compile because I didn't have a compiler configured with OpenMPI. While that's not a dealbreaker, I'm not sure why there's an OpenMPI requirement in a program with no indication of any parallelization. Perhaps it's from an external library?

Another concern that I have is the lack of any testing for the interval tree structure implemented by the authors, especially because it can be a somewhat complex data structure to implement. From the header, it looks like this was written a while ago and might have been taken from another package, where perhaps there was more testing done. If that's the case, it might still be good to include the tests here so that any new bugs might be uncovered by future users of this tool. If not, perhaps some inspiration can be taken from bedtools's tests?

Missing wobble parameter implementation?
I was able to clone and compile the code on a different machine and run it without any arguments, so the code seems sound. However, I did not attempt to rerun the authors' analysis since the wobble parameter did not appear to be implemented. With this option missing, I did not believe the software provided in the repo linked to the manuscript would allow me to replicate the analysis described by the authors. Not finding any 'wobble' parameter implementation in the C++ code or in the compiled executable, which seems to be a feature that's emphasized in the manuscript, is my primary concern about this manuscript. When I downloaded the source code as at the time of publication, it appeared that the SURVIVOR_ant directory was actually empty. Perhaps the authors neglected to update the code in the repo they refer to

in the text to a later version?

Wobble intervals

The motivation for the wobble parameter could be backed up by a little more data (although I don't doubt its need). Perhaps one of these two plots might be helpful:

1. The length of an event on one axis and the distance to the nearest overlapping event on the other (0 for when the events overlap) might make it clear that lots of events throughout the SV size spectrum that are nearby do not overlap, which would provide more motivation for this parameter.
2. Similarly, for SV calls that are grouped into a single region by SURVIVOR_ant using the wobble parameter, a plot of the reciprocal overlaps for the variants would show that calls that are grouped in a region typically have no overlap, and therefore the wobble parameter is needed.

Additionally, I wonder if a fixed wobble size (defined as 1 kb in the manuscript) is the right choice. My concerns are motivated by 2 factors:

1. For small events, let's say, on the order of 100 bp, extending the interval by 1kb on either side might be a bit liberal, assuming that the size of the SV isn't grossly underestimated and that the breakpoints are in unique regions. In this case, something that's 1kb away from that event might not be a good overlap candidate.
2. For very large events, on the order of 10s to 100s of kb or more, extending the SV on either side by 1kb might conversely be too conservative. Particularly if breakpoints are uncertain, one could imagine the ends of the event moving by 5kb+.

Additional SV analyses

The analyses proposed by the authors are very interesting, but might perhaps be trivially extended to provide additional useful information. One option that might be especially interesting to see is an extension of their 4th analysis proposal (breakpoint variation) to genotype variation. Namely, if 4+ callers have made calls of the same type, it would be good to know if they all have consistent genotypes (e.g. single or double null at a site where a deletion has been called).

One thing that was unclear to me from the current list of SV analyses is whether there's any per-locus report of the type diversity. It is obviously discussed later in the results section, so I clearly misunderstood one of the analyses described. My money's on 2 or 3. Perhaps the descriptions of these analyses could be made clearer?

SVs overlapping with HG002

The authors report that 4,506 1000 Genomes SVs overlap with their merged SV set for HG002 out of a total of 134,528 total events. Some context would be helpful for these 2 numbers. In the case of SNPs, Figure 3 from the 1000 Genomes project[1] suggests that if the 1000 Genomes project had comprehensively identified all SVs in their study cohort, HG002 should have a significantly higher overlap with previously characterized common SVs. Instead, the data here suggests that just 3% of the SVs identified in HG002 were previously described in the Phase 3 SV release from the 1000 Genomes Project. This is a bit lower than I would expect. There are, of course, possible explanations for this lower overlap that the authors could explore:

- Variants of any type in HG002 are not generally found in the 1000 Genomes Project
- Either due to the methods used or the sequencing data available, the 1000 Genomes project SV dataset is far from complete, and the authors are describing novel SVs.
- It remains possible that many of the novel SVs described in the merged HG002 SV set are false positives (not SURVIVOR_ant's problem, that's a callset problem)

While the point of this paper isn't to thoroughly analyze extant HG002 callsets, this number is a significant enough departure from expectation that it might be worth addressing.

Breakpoint prediction consistency
In the authors' explanation of why short read caller's start positions are closer to the median breakpoint, they posit that the sites detected by short read callers may be easier and less repetitive. Given that the authors have overlapped their calls with the GA4GH repetitive region BED files, they are in a position to substantiate that claim if it were true, or invalidate if it were false. A breakdown of repetitive overlap status for shared calls, calls unique to short read technologies, and calls unique to long read technologies should be done to dig into this claim a little bit more if the authors are going to make it.

Additionally, the authors that the number of callers per technology varies, but by my count, there are 7 Illumina callers and 7 PacBio callers. At least within the subset of calls detected by either of these 2 technologies, the number of callers is equal, so perhaps it is not a bias is introduced simply due to the number of callers for these 2 technologies.

Figures

Figure 2 is a little difficult to read -- perhaps log-scale the y-axis and color / sort the bars by platform / technology?

This plot also makes it apparent that many of the calls in this analysis come from a single dataset and a single variant type -- deletions from MetaSV on Illumina data. This is at least worth noting in the manuscript, and it may be worth providing numbers for the analyses done with and without this dataset since it is such a significant outlier. Again, perhaps not the point of the paper, but maybe something the authors could elaborate on a bit.

Misc
A few small suggested edits as well:

- Page 3, Column 2, Paragraph 1 (end of paragraph): if the BED file does not include a four columns -> if the BED file does not include four columns
- Page 4, Column 1, Paragraph 3: The analysis scripts are using BioPerl -> The analysis scripts use BioPerl
- Page 5, Column 2, Paragraph 1: including lower error rates in short reads, easier less repetitive sites detected by short reads, filtering rules, etc. -> including lower error rates in short reads; easier, less repetitive sites detected by short reads; filtering rules, etc. OR including lower error rates in short reads, easier and less repetitive sites detected by short reads, filtering rules, etc.

References
1. 1000 Genomes Project Consortium, Abecasis GR, Altshuler D, Auton A, Brooks LD, Durbin RM, Gibbs RA, Hurles ME, McVean GA: A map of human genome variation from population-scale sequencing. *Nature*. 2010; **467** (7319): 1061-73 PubMed Abstract | Publisher Full Text
2. Quinlan AR, Hall IM: BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*. 2010; **26** (6): 841-2 PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: Employee at Illumina.

Referee Expertise: Genomics, copy number variants, short read sequencing analysis

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.