bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

# Detection of complex structural variation from paired-end sequencing data

Joseph G. Arthur[*] Xi Chen[*] Bo Zhou[†] Alexander E. Urban[†]
Wing Hung Wong[*‡]

**Abstract**

Detecting structural variants (SVs) from sequencing data is key to genome analysis, but methods using standard whole-genome sequencing (WGS) data are typically incapable of resolving complex SVs with multiple co-located breakpoints. We introduce the ARC-SV method, which uses a probabilistic model to detect arbitrary local rearrangements from WGS data. Our method performs well on simple SVs while surpassing state-of-the-art methods in complex SV detection.

Since observations of microscopically visible aneuploidies and gene duplications, the study of large-scale genomic alterations has been a key component of genome analysis [1]. Structural variants, typically defined as mutations affecting at least 50 bp of sequence, are more rare than single-nucleotide variants (SNVs), but SVs are known to account for a much larger portion of sequence difference between human individuals [2, 3]. Functionally, SVs are also more likely than SNVs to impact gene expression, with larger effect sizes on average [4, 5]. The rapid improvements to sequencing technologies have prompted development of numerous SV detection methodologies [6], some of which have been applied in population-scale sequencing projects [4, 7].

The vast majority of sequencing-based SV callers detect deletions, duplications, inversions, and/or translocations [6]. If only these SV types are present, any SV can be identified by a single breakpoint, or pair of bases adjacent in the sample but not the reference. Detection of these breakpoints is theoretically straightforward given high coverage data and accurate alignments. Misaligned reads make SV detection difficult in practice, and investigators typically apply multiple methods together with heuristic filters in order to achieve high accuracy [4, 7–10]. Structural variants with complexity beyond the scope of most detection algorithms have also been observed in a variety of phenotypic contexts [4, 11–13]. Our definition of a complex SV (cxSV) is any rearrangement not reducible to non-overlapping deletions, tandem duplications, novel insertions, and inversions; we focus here on the localized cxSVs commonly observed in the germline.

[*]Department of Statistics, Stanford University
[†]Department of Psychiatry and Behavioral Sciences and Department of Genetics, Stanford University School of Medicine
[‡]Department of Biomedical Data Science, Stanford University

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Identification of cxSV structures requires both detection and phasing of breakpoints. However, methods targeting simple SVs generally equate breakpoint detection with SV detection. Thus cxSVs are either inadequately represented by a single SV or incorrectly detected as multiple overlapping SVs. While cxSVs have been studied more carefully by inspection of paired-end mapping patterns [13] and the use of long sequencing reads or mate pairs [4, 12, 14, 15], methods and software for standard WGS data [16–18] are scarce and still fairly untested. Thus, we have developed the Automated Detection of Complex SVs (ARC-SV) method, which can detect a broad class of localized rearrangements from paired-end WGS data.

There are numerous possible cxSV structures, each producing a signature of read depth, mapped read orientations, insert sizes, and split alignments. The observed signature will depend on both sequencing library characteristics (especially insert size) as well as the SV. Several previous methods [12, 17, 18] classify cxSV structures by direct comparison to these mapping signatures. To achieve more general detection of cxSV structures, ARC-SV proposes candidate diploid rearrangements and scores each one using a probabilistic model that incorporates all available reads, as illustrated in Figure 1 and described in our Methods. Notably, ARC-SV simultaneously detects and genotypes SVs using all relevant reads, instead of relying only on discordant reads. Previous work most similar to ours includes SVelter [16], which detects cxSVs using another "propose and score" approach, and SWAN [19], which uses a detailed probabilistic model for insertion and deletion detection.

In assessing cxSV detection methods, a major difficulty is that comprehensive gold sets of cxSVs are not available; it is also not *a priori* clear how to define similarity between general rearrangements. To comprehensively evaluate SV calling accuracy, we aligned each variant's rearranged sequence against a published assembly or long read sequencing. Validation of an SV requires that the optimal alignment spans the entire rearrangement, aligns each rearranged segment with high sequence identity, and aligns across simple deletion breakpoints. Additionally, we require that the same validation criterion is not met by alignment to the original reference (Methods). This validation procedure is similar to a previous method [16], though we additionally incorporate the SV structure into our scoring function.

We applied ARC-SV and 5 other SV callers to WGS data derived from two healthy human samples — Venter and NA12878 — chosen so we could validate predictions against a diploid assembly [3] and PacBio data [20], respectively (Methods). In what follows, we have excluded variants with 85% overlap to tandem repeat regions, removed near-duplicate SV calls within each call set, and removed NA12878 variants with very low PacBio read coverage (Methods).

We first compare validation rates between our alignment-based method and the usual 50% reciprocal overlap criterion, using sets of known NA12878 and Venter deletions [8, 9]. In terms of the relative sensitivity and specificity of different callers, our alignment-based validation gives nearly identical results to those based on the gold sets (Supp. Figures 6 and 7). Across all size ranges below 10 kb, we observed high concordance between the sets of true positives from each validation method; concordance among false positives was somewhat lower but may reflect incompleteness in the gold sets (Supp. Figure 8). Thus, for simple SVs, our SV validation approach is highly consistent with the standard approach based on gold sets.

Next, we use our approach to evaluate the performance of SV callers across all SV types.

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

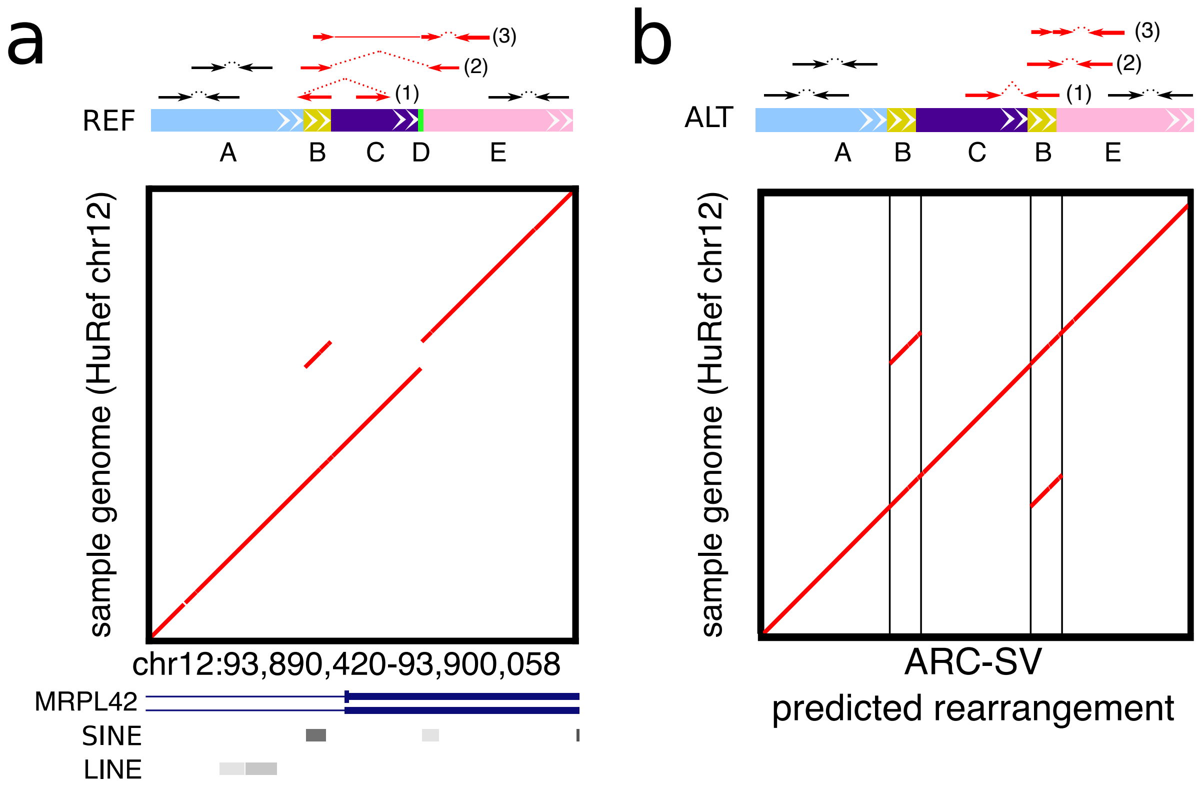

a
(3)
(2)
(1)
REF
A
B
C
D
E
sample genome (HuRef chr12)
chr12:93,890,420-93,900,058
MRPL42
SINE
LINE
b
(3)
(2)
(1)
ALT
A
B
C
B
E
sample genome (HuRef chr12)
ARC-SV
predicted rearrangement

Figure 1: A: Sequence comparison between the reference genome and a corresponding region in the sample (ground truth sequence from HuRef assembly). Discordant alignments are consistent with both (1) a tandem duplication of $BC$; and (2, 3) a deletion of $C$ or $CD$. B: Comparison using ARC-SV's predicted rearrangement instead of the reference. The reads producing discordant alignments are more likely given this alternative structure.

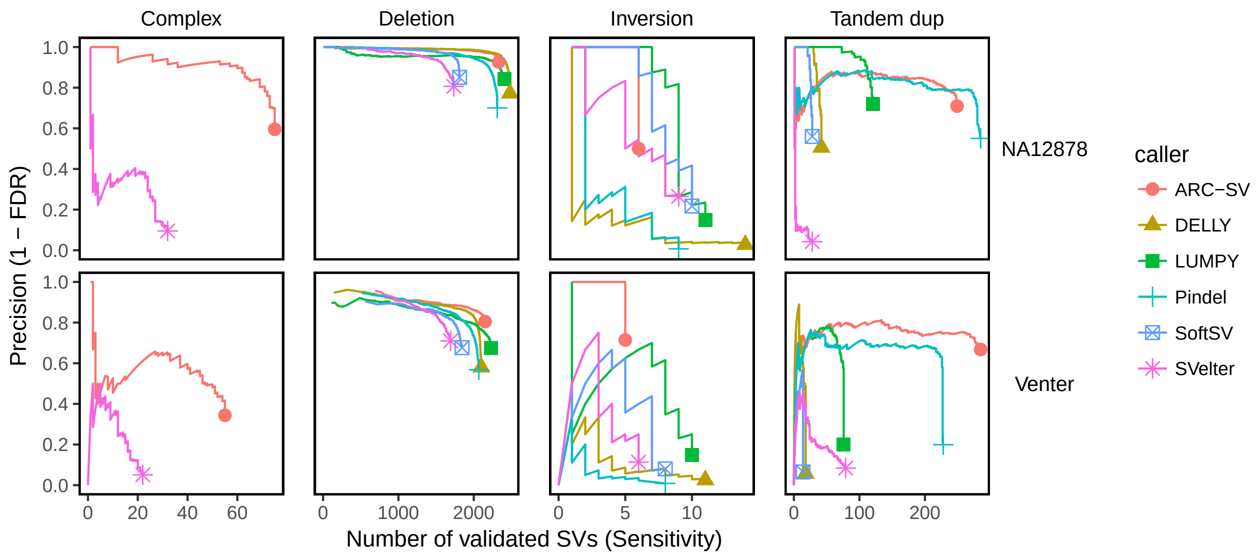
We compared the accuracy of the initial call sets (Supp. Table 1) as well as calls after filtering for quality (Figure 2A; stratified by SV size in Supp. Figures 9, 10, 11). Each call set was filtered by a random forest classifier trained on validation results from the opposite sample, using the relevant features from each algorithm's output (Supp. Table 5; Methods).

Before random forest filtering, ARC-SV achieves the highest precision in both samples and across all variant types except Tandem Duplications in NA12878, where LUMPY is more accurate (empirical FDR of 28% vs 29% in ARC-SV) but much less sensitive. The sensitivity-precision curves suggest that, after filtering, ARC-SV and DELLY are most competitive for deletions, while ARC-SV and Pindel are best for Tandem Duplications. There are too few validated simple inversions for a precise comparison, but ARC-SV's improved precision may result from its ability to correctly resolve complex inversions. However, ARC-SV's accuracy comes at a cost of significantly reduced sensitivity for some variant classes. Most notably, LUMPY correctly detects 25% more deletions $> 1$ kb across both samples (1227 vs 984), though with a large drop in precision compared to ARC-SV (63% vs 91%).

Complex SVs make up 2.5% (130/5229) of all validated ARC-SV predictions: there are 55 validated complex events in Venter, 75 in NA12878, and 16 calls in NA12878 excluded for low PacBio coverage. At 129/130 (99%) of these complex sites, one or more simple SVs

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

**a**



**b**

| Complex SV type | ARC-SV | SVelter |
|---|---|---|
| Interspersed DUP [+ DEL] | 35 (49%) | 15 (12%) |
| Inverted DUP [+ DEL] | 36 (59%) | 23 (19%) |
| INV + DEL | 32 (71%) | 13 (23%) |
| Other | 27 (25%) | 3 (1%) |

Figure 2: (A) Precision and sensitivity results based on raw SV calls (points) and cross-sample random forest-based filtering (curves). The results without filtering (points) are the same as those given in Supp. Table 1. Less reliable calls are removed as we increase the stringency of the filter. (B) Types of cxSVs discovered across NA12878 and Venter (combined results, validation rates in parentheses).

was called by other methods on the same sample (with at least 50% overlap to the complex call). Many of the overlapping simple SVs were validated by our alignment method, but the results from ARC-SV suggest that additional complexity was missed.

Most validated complex events fall into three categories: interspersed (non-tandem) duplications, inverted duplications (mostly non-tandem), and inversions flanked on one or both sides by deletions (Figure 2B, Supp. Fig. 13, Supp. Tables 3 and 4). A number of other complex SV structures were also observed (Supp. Fig. 14), illustrating one advantage of a highly general cxSV caller.

Complex SVs containing duplicated segments make up a significant portion of all validated ARC-SV duplications (249 tandem vs 55 complex duplications in NA12878; 285 vs 38 in Venter). Out of 71 interspersed duplications correctly called by ARC-SV across our two samples (without any merging or deduplication of events), 51% involve an inversion of the duplicated segment, and 70% have an accompanying deletion at the insertion site. The median sizes of these variants were 187 bp duplicated, an insertion site 1155 bp away, and, among cases with deletions, 13.5 bp of deleted sequence (Supp. Figure 15).

Consistent with recent surveys of complex variation [4, 12], we find few validated simple

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

inversions compared to complex events with inverted segments (ARC-SV: 6 simple vs 49 complex in NA12878; 5 vs 36 in Venter). The most common structures seen among complex inversions are inverted duplications and inversions with deletions at one or both flanks (Figure 2B, Supp. Table 3).

We used targeted sequencing to experimentally interrogate several Venter SV calls. The validation rate was 80% (4/5) for simple SVs and 80% (4/5) for complex SVs; genotype calls were correct for 7 of the 8 validated SVs (Methods; Supp. Table 6). We also investigated SVs called from whole genome sequencing of HepG2 cells, whose abnormal karyotype does not match ARC-SV's modeling assumption of a diploid genome (Supp. Tables 7, 8). Sequencing across predicted breakpoints yielded validation rates of 93% (40/43) for deletions and 80% (16/20) for tandem duplications; testing 5 non-tandem duplications confirmed both breakpoints for 3 variants and a single breakpoint of another variant.

ARC-SV and SVelter each required approximately 110 CPU hours per sample, compared to LUMPY's 9 hours and DELLY's 18 hours. Future versions of the ARC-SV software will likely incorporate computational speedups, but for now, our detection of cxSV structures comes at an increased computational cost.

Limitations of our current methodology include a lack of novel insertion detection or non-reference sequence determination in general. Also, unambiguously resolving some SVs (e.g., large homozygous tandem duplications) requires phasing information not available in a single paired-end sample. It is well-known that paired-end data lack full sensitivity for SV detection [10], but ARC-SV's probabilistic model is well-suited for extension to similar data types, e.g., mate pairs or linked reads. Moreover, the existing wealth of data from paired-end sequencing experiments provides ample opportunity for the use of methods like ARC-SV.

## Methods

### Breakpoint detection

**Soft-clipped and split alignments.** Soft-clipped and split alignments produced by BWA–MEM are both used for breakpoint detection. Split alignments are assigned a type (deletion, duplication, left/right side of inversion) and a breakpoint interval based on the orientation and mapped positions of the primary and secondary alignment. Soft-clipped reads without split alignments are also used. We require the soft-clipped portion is at least 5 bases long and has median base quality 15 or higher. For increased sensitivity on soft-clipped reads, we require a MAPQ score of 10 rather than the threshold of 20 used elsewhere in ARC-SV.

Soft-clipped reads within 5 bp and with the same clip orientations are merged; the breakpoint position whose supporting reads have the highest total MAPQ is used as a candidate breakpoint. Also, microhomology can lead to two soft-clip clusters at a single insertion or inversion breakpoint. We thus merge each pair of soft-clip clusters with overlapping aligned bases and opposite clip orientations whenever the two breakpoints are within 25 bp; the region between the breakpoints is then used as a breakpoint interval.

**Discordant read pair extraction.** Discordant read pairs (DRPs) are read pairs whose abnormal alignment suggests the presence of structural variation. Pairs with abnormal in-

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

sert sizes suggest deletions and insertions, and pairs with abnormal mapping orientation suggest tandem duplications and inversions. We treat DRPs from the left and right sides of inversions separately.

Deletion and insertion read pairs are defined according to insert size cutoffs. Conventionally, discordant insert sizes are those outside of $(\mu - n\sigma, \mu + n\sigma)$, where $\mu$ and $\sigma$ are the mean and standard deviation of the estimated insert size distribution $f$, and the multiplier $n$ is commonly set to 3 [21, 22]. To allow for a variety of different insert size distributions, we instead use cutoffs based on a likelihood ratio. Because discordant reads are extracted during the first pass of the data, the insert size distribution is estimated beforehand using a random subsample of reads (1 million by default). To avoid including read pairs that clearly span large deletions, we truncate the insert size distribution at 3 times the median insert size. We define deletion pairs as those with insert sizes exceeding $z^*$, where $z^*$ is the smallest value such that, for all $z > z^*$,

$$\frac{\max_{S \geq 0} f(z - S)}{f(z)} > C.$$

The ratio above involves the probability of observing an insert size $z$ under the reference genome versus a genome with a homozygous deletion of maximal likelihood. We set $C = \exp(3^2/2)$ so that, if $f$ is a normal distribution, the cutoff $z^*$ is the commonly-used value $\mu + 3\sigma$. Insertion read pairs are defined by insert sizes smaller than $z_*$, where $z_*$ is defined analogously to $z^*$ but with maximization over $\{S \leq 0\}$.

**Discordant read pair clustering.** For discordant pair $i$, we denote the mapped insert size as $z_i$ and the $3'$-most mapped read positions as $a_i$ (leftmost read in the reference) and $b_i$ (rightmost). After extracting DRPs, we construct a graph with nodes corresponding to DRPs. Two DRPs $i$ and $j$ are connected if they are of the same type, if both $|a_i - a_j|$ and $|b_i - b_j|$ are less than $\mu + 3\sigma$, and if $\max\{a_i, a_j\} < \min\{b_i, b_j\}$. The third condition requires that some region is spanned by both DRPs. (For insertions, we require $\max\{a_i, a_j\} - h < \min\{b_i, b_j\}$ to allow for up to $h = 20$ base pairs of microhomology at the insertion site.) The parameters $\mu$ and $\sigma$ are estimated from the insert size distribution using the median and 1.35 times the interquartile range, respectively. After constructing the graph of compatible read pairs, we cluster the read pairs by taking the largest clique(s) in each connected component (ignoring singleton components).

**Discordant read pair FDR.** We select "significant" clusters using a procedure that adapts to the coverage of the data and only requires a target false discovery rate (FDR). For a given DRP type with $M$ total clusters, we assign each cluster $j$ a heuristic score $s_j$ indicating the strength of evidence. For inversions and tandem duplications, $s_j$ is the cluster size. For insertions and deletions, $s_j$ is a log-likelihood ratio statistic. For deletion clusters, we first estimate the deletion size $\widehat{D}_j$:

$$D_j = \min\left\{\frac{1}{|\mathcal{C}_j|}\sum_{i \in \mathcal{C}_j} z_i - \mu, \min_{i \in \mathcal{C}_j} b_i - \max_{i \in \mathcal{C}_j} a_i\right\},$$

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

where $\mathcal{C}_j$ is the set of read pairs within the cluster. The cluster score is

$$s_j = \prod_{i \in \mathcal{C}_j} \frac{\phi(z_i; \mu + D_j, \sigma)}{\phi(z_i; \mu, \sigma)},$$

where $\phi(\cdot; \mu, \sigma)$ is the density function for the normal distribution. The case for insertions is completely analogous.

Under a null hypothesis of no structural variation, we would still expect some false DRP clusters to form when spurious discordant alignments occur close together. To simplify, we assume these errors happen independently. To simulate the formation of false clusters, we move each DRP to a new reference position uniformly at random, without changing the insert size or mapping orientations. Repeating the clustering process on these DRPs yields $M_0$ clusters. The resulting distribution of cluster scores is used as an estimate $\widehat{F}_0$ of the null score distribution. We also use $\widehat{\pi}_0 = \min\{1, M_0/M\}$ to estimate the proportion of false clusters in the original sample. If FDR control at level $\alpha$ is desired, we use a cluster score cutoff

$$s^* = \min_j \left\{ s_j : \frac{M\widehat{\pi}_0(1 - \widehat{F}_0(s_j))}{\#\{k : s_k \geq s_j\}} \leq \alpha \right\}.$$

Then all clusters $i$ of the appropriate type with scores $s_i \geq s^*$ are kept as significant.

**Breakpoint intervals.** Discordant read clusters alone do not give precise breakpoint locations, so we provide an estimated breakpoint location along with a confidence interval. A natural estimate for the breakpoint position uses the locations of the $3'$-most aligned bases among all reads in the cluster. For example, a deletion cluster with $3'$ positions $\{a_i\}$ and $\{b_i\}$ yields breakpoint estimates $\max_i a_i$ and $\min_i b_i$. The true distribution of $3'$ positions around each breakpoint will depend on the insert size distribution. For simplicity, we approximate using a uniform distribution on an unknown but fixed genomic interval $[\underline{\theta}, \overline{\theta})$ and use a 95% confidence interval for the breakpoint position (see Supplementary Note for a derivation).

**Breakpoint merging.** After the above steps, we have breakpoint locations derived from soft-clipped reads, split reads, and discordant read pairs. We merge all overlapping breakpoint intervals and exact breakpoints (represented as intervals of length 1); we keep merged intervals with a supporting DRP cluster or at least 2 supporting split/clipped reads. Given merged breakpoint intervals $\{I_j = [a_j, b_j)\}_{j=1}^{n_{\text{bp}}}$, along with $I_0 = [0, 1)$ and $I_{n_{\text{bp}}+1} = [L, L+1)$, where $L$ is the length of the chromosome, we define blocks $B_j = [b_{j-1} - 1, a_j)$ for $j = 1, \ldots, n_{\text{bp}} + 1$. (Note that if there is breakpoint uncertainty, i.e., if $b_j - a_j > 1$ for some $j$, then there will be gaps between some adjacent blocks.) These blocks are the genomic segments that will be rearranged during SV detection.

**Insertion breakpoint detection.** In order to mitigate some false positives, ARC-SV internally detects novel insertions (though the current version does not output them). Before the main inference procedure, at breakpoints derived from an insertion-type read cluster or from both orientations of soft-clipped reads, we compute the likelihood (model described below) of heterozygous and homozygous insertions over a range of lengths. If the likelihood surpasses that of no insertion then we add an insertion block $S_i$ of the inferred length.

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

## Adjacency graph

**Graph creation.** The adjacency graph is an undirected graph with two nodes for each block, $B_j^{\text{in}}$ and $B_j^{\text{out}}$, corresponding to the start and end relative to the reference coordinates; insertion blocks $S_i$ are treated likewise. "Block edges" are added between each block's pair of nodes. Other edges in the graph represent adjacencies between genomic segments and are called "adjacency edges." We do not include any adjacency edges across gaps in the reference.

Adjacency edges $B_j^{\text{out}} - B_{j+1}^{\text{in}}$ implied by the reference sequence are added automatically, as are edges connecting candidate insertions $S_i$ to their flanking blocks. Other adjacency edges must be supported by the data. For example, a read pair aligned to the forward strand of $B_i$ and to the reverse strand of $B_k$ is considered to support the edge $B_j^{\text{out}} - B_k^{\text{in}}$ if the implied insert size under that adjacency falls within the expected range (middle 95% of the insert distribution). An exception is made for adjacencies $B_i^{\text{out}} - B_i^{\text{in}}$, i.e., those supporting a duplicated block. In this case paired-end support requires that the implied insert size under no tandem duplication falls outside the expected range. Split reads whose splits occur at candidate breakpoints automatically add support to the implied novel adjacencies. The final adjacency graph is formed by removing edges with only 1 supporting read, as well as edges spanning more than 2 Mb. This distance is longer than, for example, the largest deletion in the Venter gold set [23].

**Graph partitioning.** Paths through the adjacency graph will correspond to candidate SVs. To limit computational costs, we consider a set of subgraphs formed as follows. A reference block $B_j$ as "spanned" by the adjacency graph if there is some edge connecting $B_i$ to $B_k$, where $i < j < k$. Define a "back edge" from $B_j^{\text{out}}$ as any edge connecting to a block $B_i$ with $i \leq j$; similarly, a back edge from $B_j^{\text{in}}$ connects $B_j^{\text{in}}$ to some block $B_k$ with $k \geq j$. We call $j$ a cut point in case $B_j$ is not spanned and there are no back edges connected to $B_j^{\text{in}}$ or $B_j^{\text{out}}$. Additionally, the first and last reference blocks, as well as blocks adjacent to assembly gaps, are considered cut points. Thus, we have cut points $1 = j_1 < j_2 < \cdots < j_P = J$ within the graph. For each 2 consecutive cut points $j_m$ and $j_{m+1}$, we will consider the subgraph formed by blocks $B_{j_m}, \ldots, B_{j_{m+1}}$ as well as any insertion blocks contained within that genomic region. Subgraphs consisting of only a pair of reference blocks are discarded. Finally, each subgraph is extended on either side to ensure that some minimal amount of sequence flanks any structural variants, and subgraphs that overlap as a result are merged.

**Subgraph traversal.** Given two nodes $s$ and $e$, a valid traversal through the graph is a path from $s$ to $e$ that alternates between block edges and adjacency edges, beginning and ending with block edges. Any valid traversal is equivalent to some sequence of oriented blocks, i.e., a genomic rearrangement. For tractable computation, we limit the number of times (by default, 2) each node may be exited using any back edges.

## Probabilistic model

Suppose a subgraph/region $R$ consists of reference blocks $B_i, B_{i+1}, \ldots, B_j$. We call the set of valid traversals $\Theta$; each $\theta \in \Theta$ represents a possible rearrangement of $R$. The set of genomic

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

coordinates within $R$ under rearrangement $\theta$ is called $\mathcal{R}_\theta$. We aim to maximize the likelihood of the observed paired-end alignments, $P(\text{data}; \theta_1, \theta_2)$, as a function of the sample's diploid genotype $(\theta_1, \theta_2) \in \Theta^2$.

**Fragment and alignment model.** Fragments of DNA are assumed to be sampled from $\mathcal{R}_\theta$ with frequencies proportional to the insert size density. If $x, y \in \mathcal{R}_\theta$ and $x < y$, fragment $[x, y)$ is sampled with probability

$$P(x, y; \theta) = \frac{f(y - x)}{\sum\limits_{u,v \in \mathcal{R}_\theta, u<v} f(u - v)}.$$

After the ends of fragment $[x, y)$ are sequenced, we observe a paired-end alignment to the reference genome, $\underline{a} = (a_1, a_2)$. Specifically, the alignment $a_j$ is represented as $a_j = (s, o, m)$. The variable $s$ is the sequence of oriented reference blocks corresponding to the alignment. For example, a read from the forward strand of $B_1$ has $s = (B_1)$, whereas a read from the reverse strand has $s = (B_1')$; a split alignment supporting the deletion of $B_2$ may have $s = (B_1, B_3)$. We use $o$ to denote the read's start position, or offset, within the first block of $s$. By assuming $s$ and $o$ are directly observed, we implicitly assume that alignments are perfect and that split reads are always mapped correctly (Supp. Fig. 3). We do not explicitly model soft-clipping: clipped bases on the $3'$ end are ignored, and reads with clipped $5'$ ends have their offsets $o$ adjusted so the fragment length is calculated correctly.

**Hanging read model.** The variable $m$ indicates the mapping outcome of the sampled read. Ordinary paired-end alignments $(a_1, a_2)$ mapped back to the region $R$ have $m_1 = m_2 = 1$. A "hanging pair" in which only read 1 is mapped is indicated by $m_1 = 1, m_2 \neq 1$. In this case we set $m_2 = 0$ if read 2 is unmapped and $m_2 = -1$ if read 2 is mapped outside of $R$. Pairs in which neither read is mapped to $R$ (i.e., $m_1 \neq 1$ and $m_2 \neq 1$) are assumed unobserved. Thus, we perform inference conditional on the complementary event, $m_1 = 1 \vee m_2 = 1$.

In the model used by ARC-SV, the frequency of hanging and unmapped reads depends on the unobserved variable $\underline{I} = (I_1, I_2)$, where $I_j$ is 1 if read $j$ is derived from a novel insertion and 0 if it derives from reference sequence. For the case where both reads derive from reference sequence, we estimate the distribution $P(m_1, m_2 | I_1 = I_2 = 0)$ using the observed frequencies of mapped and hanging pairs across all reads. This is a conservative overestimate since we include insertion-derived reads. To handle reads within insertions, we assume the probability of successfully mapping a reference-derived read does not depend on the location of its mate:

$$P(m_1 = 1 | I_1 = 0, I_2 = 1) = P(m_1 = 1 | I_1 = 0, I_2 = 0).$$

Conditional on the reference-derived read mapping successfully ($m_1 = 1$ above), we assume the insertion-derived read is either unmapped or mapped distantly ($m_2 = 0, -1$) with equal probability. If the one reference-derived read is unmapped, or if both reads derive from inserted sequence, then we assume neither $m_1, m_2$ is equal to 1, and the pair is not observed.

**Likelihood evaluation.** Given a paired end alignment $\underline{a} = (a_1, a_2)$ and a candidate haplotype $\theta$, we consider the set of fragments from $\mathcal{R}_\theta$ that have block sequences and offsets

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

identical to those of $(a_1, a_2)$. Call this set of "compatible" fragments $C_\theta(\underline{a})$. Each member of $C_\theta(\underline{a})$ is an interval $[x, y) \subseteq \mathcal{R}_\theta$ and a variable $\xi \in \{1, 2\}$, which indicates which one of $a_1$ or $a_2$ corresponds to the "$x$ end" of the fragment. For example, suppose read 1 matches the forward strand of $B_1$, and read 2 matches the reverse strand of $B_3$. Then there is 1 compatible fragment under $\theta = B_1B_3$, none under $\theta = B_1B_2$, and 2 under $\theta = B_1B_2B_3B_3$. If only one read is mapped, then $C_\theta(\underline{a})$ consists of all fragments matching the mapped end. For more illustration, see Supp. Fig. 3.

We assume that $\underline{a}$ is observed conditional on the mapped end(s) falling within the region $R$. Summing over the compatible fragments and incorporating the hanging read model, we have the likelihood

$$P_\theta(\underline{a}|\text{mapped ends in } \mathcal{R}_\theta) = \frac{\displaystyle\sum_{([x,y),\xi)\in C_\theta(\underline{a})} f(y-x)P\left(\underline{m}|\underline{I} = \eta_\theta([x,y),\xi)\right)}{\displaystyle\sum_{x,y\in\mathcal{R}_\theta : x<y}\ \sum_{\xi\in\{1,2\}} f(y-x)P\left(m_1 = 1 \vee m_2 = 1|\underline{I} = \eta_\theta([x,y),\xi)\right)}$$
$$\equiv \frac{p_\theta(\underline{a})}{G_\theta},$$

where $\eta_\theta$ indicates whether the reads derive from novel insertion sequence.

If the rearrangement $\theta$ contains no novel insertions, then the normalizing constant $G_\theta$ merely requires computing a double sum $\sum_{x=0}^{L}\sum_{y=x+1}^{L} f(y-x)$, where $L$ is the length of the haplotype. Adjusting for $N$ novel insertions requires an additional computation of $O(N^2)$ similar double sums. Each such term is computed in $O(1)$ time using pre-computed values of $F(t) = \sum_{x=0}^{t} f(x)$ and $\overline{F}(t) = \sum_{x=0}^{t} F(x)$.

The likelihood of the data, assuming fragments are sampled independently, is then

$$P_\theta(\text{data}) = \frac{1}{G_\theta^n}\prod_{i=1}^{n} p_\theta(\underline{a}_i).$$

Note that $p_\theta(\underline{a}_i) = 0$ if $\underline{a}_i$ has no compatible fragments under $\theta$. This may occur even if $\theta$ is correct, for example if there is a spurious split alignment or an error in mapping orientation. Thus, we use the following "robust" likelihood,

$$\tilde{P}_\theta(\text{data}) = \frac{1}{G_\theta^n}\prod_{i=1}^{n}(\pi + (1-\pi)p_\theta(\underline{a}_i)).$$

This modified likelihood gives a small probability to all possible mappings, eliminating the effect of extreme outliers. Similar techniques have been used in classical statistical problems — see e.g., the "redescending M-estimators" discussed in [24]. For this work, we used a fixed value of $\pi = 10^{-6}$, but future versions of ARC-SV will adjust this parameter based on the scale of the insert size distribution.

**Diploid rearrangements.** Since we are working with diploid genomes, ARC-SV attempts to maximize the likelihood over diploid rearrangements $(\theta_1, \theta_2) \in \Theta_R^2$:

$$\tilde{P}_{\theta_1,\theta_2}(\text{data}) = \frac{1}{(G_{\theta_1} + G_{\theta_2})^n}\prod_{i=1}^{n}(2\pi + (1-\pi)(p_{\theta_1}(\underline{a}_i) + p_{\theta_2}(\underline{a}_i)).$$

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Specifically, we use the following procedure:

1. Enumerate rearrangements $\theta \in \Theta_R$ based on traversals through the current subgraph $R$. If the size of $\Theta_R$ is greater than 1000, skip this subgraph.

2. Rank each $\theta$ by its homozygous likelihood, $\tilde{P}_{\theta,\theta}(\text{data})$, and let $\overline{\Theta}_R \subseteq \Theta_R$ contain the best 50 rearrangements.

3. Maximize the diploid likelihood $\tilde{P}_{\theta_1,\theta_2}$ over $(\theta_1, \theta_2) \in \overline{\Theta}_R^2$ to produce the final SV call.

**Implementation details.** Recall that there may be gaps between some blocks due to breakpoint uncertainty. It is possible (though rare in practice) that an alignment falls entirely into one of these gaps; these reads are ignored. Also, when rearranging blocks, we take the median of any breakpoint gap to be the best guess for the breakpoint location. For example, a deletion of $B$ from $ABC$ will leave behind half of any gaps between $A$ and $B$, $B$ and $C$.

ARC-SV only uses mapped reads with a MAPQ score of at least 20, except when determining candidate breakpoints from soft-clipped reads. The unmapped or “distantly mapped” ends of hanging reads are not subject to a MAPQ constraint.

Split reads with breakpoint positions that do not agree with our merged, filtered breakpoints are ignored. (Breakpoint filtering requires two supporting reads, so isolated split reads do not contribute to the final breakpoints.)

## Computational validation of SVs

**Motivation.** In the validation of many previous methods, a simple SV call is often labeled a correct detection if it has high overlap to a known SV [9, 16, 21, 25]. However, complex SVs do not generally correspond to unique sequences of operations on intervals (e.g., deletions and inversions; see Supp. Fig. 4). Thus, it is not obvious how to extend the usual SV overlap criterion to complex variants. We instead validate SV predictions by direct sequence comparison to a ground truth, either a high-quality assembly or long reads. The SV is considered validated if the predicted sequence matches the truth; additionally, we check that the predicted sequence is significantly different than the original reference when subjected to the same comparison.

**Altered reference and ground truth sequences.** We construct a predicted sequence for each SV call by rearranging the reference genome accordingly and including 1000 bp of flanking sequence on each side. This altered sequence is annotated with the locations and sizes of any deletions as well as the boundaries of the rearranged reference blocks. We say block $B_k$ is deleted if $B_k$ is absent in the rearrangement and the subsequence $B_{k-1}B_{k+1}$ is present in either orientation.

If an assembly is used as ground truth, SV predictions from each chromosome are aligned to the assembled versions, as well as all unplaced contigs. When validating against a set of long reads, each SV prediction is aligned to the set of long reads mapping discordantly (i.e., with an indel $\geq$ 50 bp, a split alignment, or at least 100 soft-clipped bases) to the appropriate chromosome.

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

**Alignment and heuristics.** For SV validation, the following options are used with BWA–MEM [26]:

```
bwa mem -a -D 0 -w 1000 -x intractg {ground_truth} {altered_seq}
```

These options are chosen to obtain a large, sensitive set of alignments that include large indels.

Sometimes BWA–MEM will produce multiple alignments to a single ground truth sequence. We find the single best alignment chain according to the same scoring function used by BWA–MEM. Formally, we have alignment segments with coordinates $[s_i^q, e_i^q)$ in the query (i.e., predicted SV) sequence and coordinates $[s_i^r, e_i^r)$ in the reference (ground truth). Segment $j$ can follow $i$ in the alignment chain if their orientations are the same and if $j$ appears after $i$ in both query and reference: $e_i^q \leq s_j^q$ and $e_i^r \leq s_j^r$ (for positive strand alignments). In order to properly chain together large segments with small amounts of overlap — e.g., a pair of segments with query coordinates $[0, 100)$ and $[95, 200)$ — we split each segment at all points of overlap (in reference or query coordinates) with other segments.

Finding the best alignment chain from $N$ segments costs $O(N^2)$ time with the standard dynamic programming solution (though speedups are possible [27]), so we incorporate several heuristics. We may ignore alignment gaps longer than the length $L$ of the entire query sequence since, with our choice of penalties, such an alignment chain would be inferior to any single aligned segment. We also ignore segments smaller than 0.5% of the smallest non-flanking reference block's length.

It is possible that the predicted SV is correct, but (say) a long read that matches the prediction is too short to align across the entire flanking sequence. The best overall alignment may be to a read covering both flanking sequences in the query but not matching the predicted SV. Thus, when an alignment extends at last 200 bp out into the query's flanking sequence but cannot continue because the reference sequence ends, we adjust the alignment score upward as if the alignment continued to the end of the query. This adjustment is only to select the best alignment chain and does not otherwise affect the validation score (described below) for that SV.

Finally, if the ground truth sequence is an oriented assembly contig, we ignore alignments to negative strand. We will fail to validate any variants contained within large inversions, but in general using negative strand alignments tends to validate many incorrect inversions, as in Supp. Fig. 5.

**Block-wise SV validation score.** Using the best alignment, we compute a score for each simple deletion (i.e., a block not present but whose adjacent segments remain in standard orientation) and non-flanking block in the predicted SV. Validation requires that the minimum blockwise score is at least 0.5 and that at least 100 bp of flanking sequence is aligned on both sides of the SV region; validation of the complex SV $ABCBE$, for example, requires that the inner 3 blocks are aligned well, and that the alignment spans 100 bp into the flanking blocks. We additionally require the validation fails when comparing the SV to the original reference.

Our score for query blocks is

$$\frac{\#\text{aligned bases} - \#\text{assigned gapped bases}}{\text{block size}}.$$

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

The score for a deletion is 0 if not spanned at least 50 bp on each side by the alignment; otherwise the score is

$$\frac{\text{deletion size} - \#\text{assigned gapped bases}}{\text{deletion size}}.$$

Alignment gaps may span multiple blocks or deletions, but are assigned conservatively to the smallest such spanned region. If segment $j$ follows directly after $i$ in the optimal alignment chain, we say there is a reference gap if $r_{ij} = s_r^j - e_r^i > 0$, and a query gap if $s_q^j - e_q^i > 0$. The reference gap's location in the query sequence is the interval $Q_{ij} = [e_q^i, s_q^j)$. We assign the gap error $r_{ij}$ to the smallest non-flanking query block or predicted deletion within 1 bp of $Q_{ij}$. For this purpose, a length $D$ deletion is represented by a window of total width $2\max\{50, D\}$. Query gaps are assigned in the same way, but only to deletions. An exception is made for query gaps within flanking sequence but within $\max\{50, \text{block size}\}$ of the outermost non-flanking block, in which case the gap may be assigned to that block.

**Analysis of validation data.** Our validation analysis excludes certain SV calls. As stated, variants with significant overlap to simple and tandem repeat regions were considered separately (Supp. Table 2); note these validation results are likely less reliable. We remove all variants affecting fewer than 50 bp, having predicted sequences exceeding 2 Mb, or overlapping gaps in the reference genome. We also excluded NA12878 calls for which the relevant region of the reference is covered by fewer than 5 long reads, in order to avoid bias against large inversions and duplications. To avoid overcounting both true and false positives, we remove duplicates (same SV type and 90% reciprocal overlap) within each call set, keeping the calls with higher validation scores; only simple SV calls are filtered in this way, primarily from Pindel, but also from LUMPY, DELLY, and SoftSV.

Finally, we did not consider breakend (BND) type events as reported by lumpy and delly, as they represent single breakpoint and not complex SV structures. Neither do we consider replacement (RPL) events from Pindel, as Pindel does not distinguish between inserted sequences that are *de novo* and those that derive from reference sequence.

**Compound SV calls.** Some calls from ARC-SV and SVelter consist of multiple non-overlapping simple SVs; these were termed compound events. Both the number of calls and the validation rate were slightly lower than for complex SVs (Supp. Fig. 12).

**Random forest filtering.** Random forest filtering was performed on all SV calls. We fit a separate random forest for each SV type, caller, and sample. The model predicts SV validation (a binary outcome) given features taken from the SV caller's output (see Supp. Table 5). We used the `randomforest` package in R (`https://cran.r-project.org/package=randomForest`) with 5000 trees per forest. Filtering was performed on each call set using the classifier trained on the opposite (independent) sample.

## Experimental validation of SVs

**Venter SVs.** We experimentally tested 2 deletions, 3 tandem duplications, and 5 complex SV calls, as well as two control regions containing no ARC-SV calls (Supp. Table 6). First,

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

we applied long-range PCR with multiplexed sequencing on an Oxford Nanopore MinION device (flowcell version R7). Passing 2D reads were aligned to the reference with `bwa mem -w 1000 -x ont2d`, alignments with MAPQ $\geq$ 20 were divided into two groups based on the presence or absence of any large errors: split alignments, indels $\geq$ 50 bp, or soft-clipping $\geq$ 200 bp. We inspected a sample of 50 dotplots within each group to estimate the total number of reads supporting the reference, the variant call, some other structure, or else ambiguous. Sanger sequencing was also applied in cases where the long-range PCR failed to amplify the alternative allele. For detailed experimental procedures, see Supplementary Note 3.

**HepG2 SVs.** For the HepG2 SV calls we conducted Sanger sequencing across the breakpoints of 43 deletions, 20 tandem duplications, and 5 non-tandem duplications (Supp. Tables 7, 8). We required breakpoints to be verified both by PCR and Sanger sequencing (with 70% identity to the expected sequence).

## Reproducibility

**Read alignment.** Reads used for SV calling were aligned to the GRCh37 reference genome using `bwa mem` [26] version 0.7.12-r1044 with default parameters, and PCR duplicates were marked using Picard tools (http://broadinstitute.github.io/picard).

**Other SV software.** The following SV calling tools were used (with default parameters unless noted):

- DELLY [21]; version 0.7.3, using the default list of excluded regions
- LUMPY [22]; version 0.2.13, using the `lumpyexpress` command
- Pindel [28]; version 0.2.5b8, using the options `-x 4 -M 3 -B 0`
- SoftSV [29] version 1.4.2; we require both paired end and split read support
- SVelter [16]; Git commit `deb24b5` (July 8 2016); we remove calls with score $\leq$ 0

**Data availability.** The Venter/HuRef sample was sequenced to approximately 40× coverage by 2 × 100 bp paired-end reads (SRA accession SRX1016818) [9]. The NA12878 paired-end sequences were obtained from the Illumina Platinum Genomes Project (ENA accession ERR194147) and amount to 50× coverage by 2 × 100 bp reads [30].

For validation we have used the HuRef diploid assembly of Levy et al.:

https://www.ncbi.nlm.nih.gov/assembly/GCA_000252825.1/

and error-corrected PacBio reads derived from NA12878 by Mt. Sinai School of Medicine from the Genome in a Bottle Consortium [20]:

ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NA12878_PacBio_MtSinai/

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

We also used two sets of high-quality deletions from previous literature: the Venter gold set from [9] and the NA12878 deletion call set from [8]. The NA12878 set was created with preliminary 1000 Genomes Project results, so we augment the calls with NA12878 deletions from Phase 3 of the Project [4] (excluding deletions overlapping ≥50% reciprocally with the original call set).

**Software availability.** The ARC-SV software is available at https://github.com/jgarthur/arcsv.

## Acknowledgements

We thank Pegah T. Ashfar, Timothy Daley, and Angel Rubio for helpful suggestions and Alberto Lovell at the Stanford PAN Facility for Sanger sequencing.

## Funding sources

J.G.A. received support from NIH training grant T32-GM096982 and NSF Graduate Fellowship DGE-114747. A.E.U. was supported by NIH grant P50-HG007735 and the Stanford Medicine Faculty Innovation Program, and B.Z. was additionally supported by NIH training grant T32-HL110952-04. W.H.W. was supported by NIH grants P50-HG007735 and R01-HG007834.

## References

[1] Escaramís, G., Docampo, E., and Rabionet, R. A decade of structural variants: Description, history and methods to detect structural variation. *Briefings in Functional Genomics*, 14(5):305–314, 2015. ISSN 20412657. doi:10.1093/bfgp/elv014.

[2] Pang, A. W., et al. Towards a comprehensive structural variation map of an individual human genome. *Genome biology*, 11(5):R52, 2010. ISSN 1465-6914. doi:10.1186/gb-2010-11-5-r52.

[3] Levy, S., et al. The diploid genome sequence of an individual human. *PLoS Biology*, 5(10):2113–2144, 2007. ISSN 15449173. doi:10.1371/journal.pbio.0050254.

[4] Sudmant, P. H., et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, 2015. ISSN 0028-0836. doi:10.1038/nature15394.

[5] Chiang, C., et al. The impact of structural variation on human gene expression. *Nature Genetics*, 2017.

[6] Guan, P. and Sung, W.-K. Structural variation detection using next-generation sequencing data: A comparative technical review. *Methods*, 102:–, 2016. ISSN 1046-2023. doi:http://dx.doi.org/10.1016/j.ymeth.2016.01.020.

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[7] Hehir-Kwa, J. Y., et al. A high-quality human reference panel reveals the complexity and distribution of genomic structural variants. *Nature Communications*, 7:12989, 2016. ISSN 2041-1723. doi:10.1038/ncomms12989.

[8] Parikh, H., et al. Svclassify: a Method To Establish Benchmark Structural Variant Calls. *BMC genomics*, 17(1):64, 2016. ISSN 1471-2164. doi:10.1186/s12864-016-2366-2.

[9] Mu, J. C., et al. Leveraging long read sequencing from a single individual to provide a comprehensive resource for benchmarking variant calling methods. *Scientific reports*, 5(August):14493, 2015. ISSN 2045-2322. doi:10.1038/srep14493.

[10] Chaisson, M. J., et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *bioRxiv*, page 193144, 2017.

[11] Quinlan, A. R. and Hall, I. M. Characterizing complex structural variation in germline and_nihms339318. *Trends in Genetics*, pages 1–19, 2012.

[12] Collins, R. L., et al. Defining the diverse spectrum of inversions, complex structural variation, and chromothripsis in the morbid human genome. *Genome biology*, pages 1–21, 2017. ISSN 1474-760X. doi:10.1186/s13059-017-1158-6.

[13] Yalcin, B., et al. The fine-scale architecture of structural variants in 17 mouse genomes. *Genome Biol*, 13(3):R18, 2012. ISSN 1465-6914. doi:10.1186/gb-2012-13-3-r18.

[14] Frith, M. C. and Khan, S. A survey of localized sequence rearrangements in human dna. *Nucleic acids research*, 2017.

[15] Stancu, M. C., et al. Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *bioRxiv*, page 129379, 2017.

[16] Zhao, X., et al. Resolving complex structural genomic rearrangements using a randomized approach. *Genome Biology*, 17(1):126, 2016. ISSN 1474-760X. doi:10.1186/s13059-016-0993-1.

[17] Iakovishina, D., et al. SV-Bay: Structural variant detection in cancer genomes using a Bayesian approach with correction for GC-content and read mappability. *Bioinformatics*, 32(7):984–992, 2016. ISSN 14602059. doi:10.1093/bioinformatics/btv751.

[18] Yang, L., et al. Diverse mechanisms of somatic structural variations in human cancer genomes. *Cell*, 153(4):919–929, 2013. ISSN 00928674. doi:10.1016/j.cell.2013.04.010.

[19] Xia, L. C., et al. A genome-wide approach for detecting novel insertion-deletion variants of mid-range size. *Nucleic acids research*, 44(15):e126–e126, 2016.

[20] Zook, J. M., et al. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nature Biotechnology*, 32(3):246–251, 2014. ISSN 1087-0156. doi:10.1038/nbt.2835.

bioRxiv preprint doi: https://doi.org/10.1101/200170; this version posted February 1, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[21] Rausch, T., et al. DELLY: Structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, 28(18):333–339, 2012. ISSN 13674803. doi:10.1093/bioinformatics/bts378.

[22] Layer, R. M., Chiang, C., Quinlan, A. R., and Hall, I. M. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biology*, 15(6):R84, 2014. ISSN 1465-6914. doi:10.1186/gb-2014-15-6-r84.

[23] Mu, J. C., et al. Leveraging long read sequencing from a single individual to provide a comprehensive resource for benchmarking variant calling methods. *Scientific reports*, 5(August):14493, 2015. ISSN 2045-2322. doi:10.1038/srep14493.

[24] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011.

[25] Hormozdiari, F., Alkan, C., Eichler, E. E., and Sahinalp, S. C. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Research*, 19(7):1270–1278, 2009. ISSN 10889051. doi:10.1101/gr.088633.108.

[26] Li, H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[27] Zhang, Z., Raghavachari, B., Hardison, R. C., and Miller, W. Chaining multiple-alignment blocks. *Journal of Computational Biology*, 1(3):217–226, 1994.

[28] Ye, K., et al. Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25(21):2865–2871, 2009. ISSN 13674803. doi:10.1093/bioinformatics/btp394.

[29] Bartenhagen, C. and Dugas, M. Robust and exact structural variation detection with paired-end and soft-clipped alignments: SoftSV compared with eight algorithms. *Briefings in Bioinformatics*, 17(1):51–62, 2016. ISSN 14774054. doi:10.1093/bib/bbv028.

[30] Eberle, M. A., et al. A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Research*, 27(1):157–164, 2017.