A probabilistic framework for sensitive structural variant discovery

Ryan M. Layer[1], Greg G. Faust[1], Ira M. Hall[*2], and Aaron R, Quinlan[†2,3]

[1]Department of Computer Science, University of Virginia, Charlottesville, VA
[2]Department of Biochemistry and Molecular Genetics, University of Virginia, Charlottesville, VA
[3]Department of Public Health Sciences and Center for Public Health Genomics, University of Virginia, Charlottesville, VA

October 5, 2012

1 Abstract

Comprehensive discovery of structural variation (SV) in human genomes from DNA sequencing requires the integration of multiple alignment "signals" including read-pair, split-read and read-depth. However, owing to inherent technical challenges, most existing SV discovery approaches utilize only one signal and consequently suffer from low sensitivity, especially at low sequence coverage and for smaller SVs. We present a novel and extremely flexible probabilistic SV discovery framework that is capable of integrating any number of SV detection signals including those generated from read alignments or prior evidence. We demonstrate improved sensitivity over extant methods by combining both paired-end and split-read alignments and emphasize the utility of our framework for comprehensive studies of structural variation in often heterogeneous tumor genomes. We further discuss the utility of this generic approach for probabilistic interpretation of diverse genomic data.

2 Introduction

Differences in chromosome structure are a prominent source of natural genetic variation in the human population. While such structural variants, including both balanced (e.g., inversions and reciprocal translocations) and unbalanced (e.g., deletions and duplications) genomic rearrangements, are less common than other forms of genetic variation such as single-nucleotide polymorphisms (SNPs), they are typically much larger, and as such, have the potential for substantial functional consequence. Spontaneous structural variants underlie many genomic disorders and polymorphic SVs have been shown to be associated with common diseases [1]. Moreover, extensive genomic rearrangement is a hallmark of tumor genomes, and complex rearrangements have been negatively correlated with a tumor's response to chemotherapy [2].

Our detailed understanding of the landscape of SV in the human genome and the mechanisms that form them has, in large part, been driven by advances in DNA sequencing technologies. However, the discovery and genotyping of SV is fundamentally more complicated than SNPs; SVs vary

[*]Corresponding author
[†]Corresponding author

considerably in size, architecture and genomic context, and SV discovery algorithms must incorporate several distinct alignment "signals" (or a lack thereof) for accurate discovery. Consequently, the accuracy of existing SV discovery tools remains substantially lower than for smaller genetic variants. More importantly, most existing SV discovery tools utilize only a single SV detection signal, typically discordant paired-end alignment or read-depth. Discovery sensitivity is often limited as a result. The impact of this reduced sensitivity is particularly acute in studies of heterogenous cancer samples where where genomic rearrangements may be present in a subset of cells and thus difficult to detect at standard levels of sequence coverage. Therefore, even as the throughput and cost of DNA sequencing continues to improve, the need for sensitive methods will remain in order to detect evermore rare genomic events in cancer and in other studies of somatic genome variability.

3 Results

In this manuscript, we describe a general probabilistic framework for integrating diverse SV alignment signals from multiple sources. Our framework is motivated by the relatively poor sensitivity of SV discovery algorithms that utilize only one type of alignment signal. By providing a flexible framework that can exploit *all* alignment signals, we seek to improve both discovery sensitivity and specificity and to increase the resolution of predicted SV breakpoint regions.

3.1 Overview of the probabilistic framework

Our probabilistic framework is based upon a general representation of a SV breakpoint that accommodates multiple structural rearrangement signals (for a comprehensive review, see [3]). The use of a generalized framework allows all SV alignment signals to be integrated into a single discovery process (Methods). In turn, SV discovery is more sensitive, especially at lower coverage, since many additional SVs may be recovered that would otherwise be missed were only a single alignment signal considered. Similarly, greater specificity can be realized by asserting that, assuming reasonable sequence coverage, evidence for true SV should be apparent through multiple alignment signals.

We define a breakpoint as a pair of bases that are adjacent in a sample genome but not in a reference genome. To account for the varying level noise inherent to different types of alignment evidence (e.g., "split" vs. "discordant" alignments), we represent a breakpoint with pair of probability distributions spanning the predicted breakpoint regions (Figure 1, Methods). Each position in the two intervals is assigned a probability that represents the relative likelihood that the given position represents one end of the breakpoint.

Our framework provides distinct modules that map signals from each alignment evidence type to our common probability interval pair. For example, paired-end sequence alignments are projected to a pair of intervals upstream or downstream (depending on orientation) of the mapped ends (Figure 1). The size of the intervals and the likelihood at each position is based on the size distribution of the sample's DNA fragment library. The distinct advantage of this approach is that *any* type of evidence can be considered, as long there exists a direct mapping from the alignment signal to breakpoint likelihoods. Here we provide three modules for converting SV alignment signals to breakpoint likelihood intervals: paired-end, split-read, and generic. We emphasize that our framework is extensible to possible new alignment signals from forthcoming DNA sequencing technologies [**?**]. The paired-end module maps the output of a paired-end sequence alignment algorithm (e.g., BWA [4]), the split-read module maps the output of a split-read sequence alignment algorithm (e.g., YAHA[5], BWA-SW[6]), and the generic module allows users to include the result SV signal types that do not have specific modules implemented (e.g., *a priori* knowledge such as known sites of SV, and/or output from copy-number variation discovery tools).

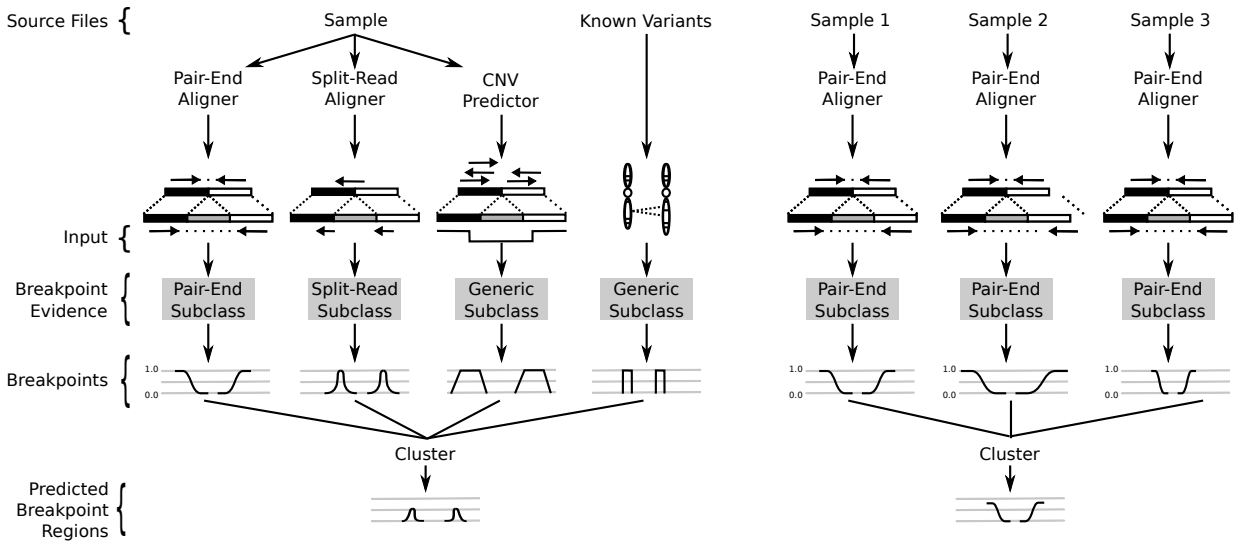

Figure 1: The LUMPY probabilistic SV discovery framework. Two example workflows are presented using extensible SV discovery framework. One workflow (left) uses three different signals (paired-end, split-read, and read-depth) from one sample, as well as prior knowledge regarding known variant sites. The second workflow (right) integrates a single signal type (in this case, paired-end) from three different samples to improve discovery among sensitivity among all three samples.

Once all of the evidence from the different classes is mapped to breakpoint intervals, all breakpoints with overlapping intervals are clustered and the probability intervals are integrated to refine the evidence for rearrangement and the predicted breakpoint interval (see Methods for details). Any clustered breakpoint region that contains sufficient evidence (based on user-defined arguments) is returned as predicted SV. Similar to the breakpoint probability, the clustered probabilities give the relative likelihood of a breakpoint. The resolution of the predicted breakpoint regions is improved by trimming the positions with probabilities in the lower (e.g., the lowest 5 percent) percentile of the distribution.

We have implemented this framework into an open source C++ software package (LUMPY) that is capable of detecting SV from multiple alignment signals in BAM alignment file from one or more samples.

3.2 Comparison of discovery performance on simulated datasets

In order to assess the performance of our framework, we compared LUMPY's discovery accuracy using paired-end (PE) alignments, split-read (SR) alignments, and both signals to three widely used SV discovery packages: Hydra [7], GASVPro [8] and DELLY [9]. We created a simulated experimental genome by simulating 1000 deletions, duplications, insertions, and inversions (4000 events total) throughout chromosome 10 of the hg19 genome using SVsim (Faust et al, *in preparation*). For each SV event type, half of the variants were less than 1kb and the other half were greater than 1kb (see Methods for details). We used the WGSIM (Heng Li, unpublished) paired-end read simulator to s̈equenceẗhe simulated genome to 2, 5, and 20 fold haploid coverage (Methods).

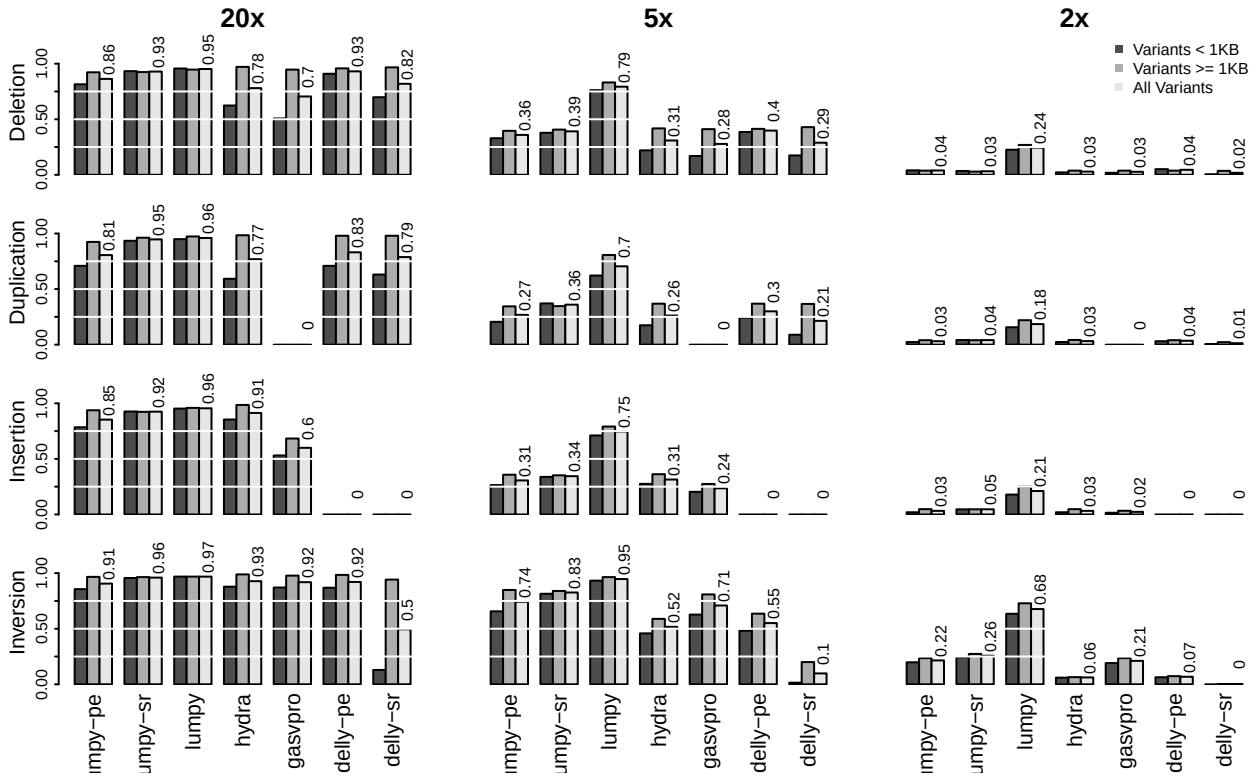

Figure 2: SV discovery sensitivity for LUMPY, HYDRA, GASVPRO, and DELLY for different SV types across multiple genome coverage levels. *lumpy-pe* reflects LUMPY sensitivity using paired-end alignments only; *lumpy-sr* reflects LUMPY sensitivity using split-read alignments only; *lumpy* describes sensitivity using both paired-end and split-read alignments; *delly-pe* reflects DELLY sensitivity using paired-end alignments only; *delly-sr* reflects DELLY sensitivity using both paired-end alignments and split-read refinement of paired-end SV predictions.

3.3 Discovery sensitivity

The predicted SV breakpoints from each discovery approach were compared to the simulated breakpoints in order to measure each approach's sensitivity (Figure 2) and false discovery rate (FDR; Table 1). Not surprisingly, for each approach, breakpoint discovery sensitivity increases with greater genome coverage. Moreover, LUMPY's sensitivity is improved when both paired-end and split-read alignments are integrated into the probabilistic framework, as compared to discovery with either signal alone. In addition, LUMPY is consistently more sensitive than other approaches at lower coverage for all SV types. For example, LUMPY detects 24.5% and 79.3% of all deletions at 2 and 5 fold genome coverage, whereas HYDRA, the next most sensitive approach, detects 2.9% and 30.9%, respectively.

At lower coverage (i.e., 2 and 5X), LUMPY is consistently more sensitive than all other approaches across all SV types. At most, LUMPY was 8.4 times more sensitive than the second most sensitive approach at low coverage (LUMPY 24.5% vs. HYDRA 2.9% for deletions at 2X coverage). At worst, it was 1.3 times more sensitive for inversions at 5X coverage (LUMPY 94.7% vs. GASVPRO 70.9%). At higher (20X) coverage, LUMPY's sensitivity advantage persists; it ranges from 95.2% to 96.9% across all SV types, whereas HYDRA and GASVPRO range from 76.9% to 92.8% and 59.9% to 91.9%, respectively (excluding duplications for which GASVPRO is incapable

of making predictions).

Unlike the other tools compared, LUMPY has nearly equal sensitivity for both smaller (i.e. <1kb) and larger (>1kb) events. Whereas at 20X coverage, LUMPY detects 95.6% and 94.6% of deletions less and greater than 1kb, respectively, GASVPRO and HYDRA each have much lower sensitivity for small variants than for large (62.3% vs 97.1% for HYDRA and 50.7% and 94.6% for GASVPRO). This increased sensitivity is especially important given that smaller SVs are much more common than larger events [10].

3.4 False discovery rate

Improved SV discovery sensitivity is crucial for comprehensive characterizations of the full spectrum of genetic variation in human genomes, yet high sensitivity at the cost of an inflated false discovery rate (FDR) is undesirable given the time and cost associated with pursuing the putative biological impact of spurious variation.

We compared the FDR for each SV discovery tool using the same simulated SVs as described above (Table 1).The false discovery rate for all tools ranged from 0.0% to 24.2%. Overall, DELLY-SR had the lowest FDR across all SV types and genome coverage levels, yet the conservative calling comes at the cost of lower sensitivity compared to the other tools. While LUMPY's FDR was slightly higher than GASVPRO for deletions, its FDR was consistently low (0.0% - 5.0%) across all SV types and coverage levels. In contrast, GASVPRO had much higher FDRs for insertions and inversions and its FDR increased at lower coverage levels. HYDRA had consistent FDRs across SV types and coverage levels (0.0% to 8.9%), yet these rates were always higher than the analogous LUMPY FDRs. These results indicate that LUMPY's probabilistic framework afford substantial improvements in discovery sensitivity while maintaining low false discovery rates.

Table 1: False discovery rates for each SV discovery approach.

Coverage	20x	5x	2x	20x	5x	2x	20x	5x	2x	20x	5x	2x
Variety	Deletions			Duplications			Insertions			Inversions		
lumpy-pe	0.014	0.008	0	0.004	0	0	0.042	0.016	0	0.004	0	0
lumpy-sr	0.006	0	0	0.005	0.003	0	0.013	0.006	0	0.004	0.001	0
lumoy	0.018	0.005	0	0.007	0.001	0	0.05	0.009	0	0.009	0.002	0
hydra	0.038	0.006	0	0.03	0.011	0	0.089	0.034	0.03	0.054	0.006	0
gasvpro	0	0	0	N/A	N/A	N/A	0.242	0.065	0.041	0.005	0.082	0.079
delly-pe	0.002	0	0	0	0	0.028	N/A	N/A	N/A	0	0	0
delly-sr	0	0	0	0	0	0	N/A	N/A	N/A	0.004	0	0

3.5 Increased breakpoint resolution

By integrating both paired-end and split-read SV signals, the resolution of our predicted breakpoint intervals is increased relative to the resolution yielded by examining either signal on its own (Figure 3). The biggest increase in resolution comes from split-read alignments, as, in principle, split-reads map SV breakpoints to a single base pair. In practice, however, there is often local alignment ambiguity at the breakpoint which reduces breakpoint resolution to 2-5bp [7]. Therefore, in order to capture all split-reads supporting a given breakpoint, we examine a "window" surrounding a putative breakpoint (e.g. +/- 20bp). This allows greater *discovery* sensitivity (i.e., more alignments are included) at the cost of slightly reduced breakpoint *resolution*. Consequently, at 2X coverage, our combined deletion breakpoint resolution is 42bp. However, at 20X coverage,

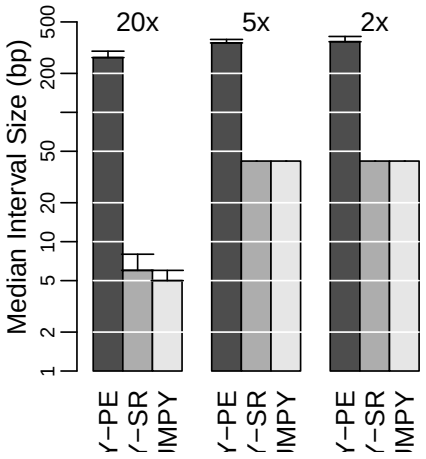

Figure 3: Breakpoint resolution measured as the median interval size in bp for true positive deletion calls at 20x, 5x, and 2x coverage.

our resolution increases to 5bp. Importantly, the breakpoint resolution for all other SV types were nearly identical to resolution for deletions (data not shown).

3.6 Benchmarks using 1000 Genomes datasets

Speed with 1, 2, 3, ..., 10 samples. How many events per sample? Need to use both PE and SR. To assess the utility of our framework for SV discovery in typical human genome sequencing datasets, we combined whole-genome alignment signals from between 1 and 10 samples from the 1000 Genomes Project [11]. The runtimes of our software ranged from X to Y minutes when exploring alignments from 1 and 10 samples, respectively, and the maximal memory usage was Z, indicating that our framework provides improved SV discovery in reasonable time-frames using resources that are available in typical research computing environments. When combining alignments from all 10 samples, the median number of predicted SV breakpoints per sample was X, which is well in the range of SV breakpoints for a typical human genome [10].

4 Discussion

We have developed a general probabilistic framework for accurate SV discovery, and we have demonstrated that our framework is more sensitive than existing discovery tools across all SV types and coverage levels. Importantly, the increased sensitivity does not come at the cost of excessive spurious SV predictions. While our FDR was slightly higher for deletions than other tools, we note that the integration of third-party copy-number predictions into our calling framework (using the generic subclasses, Figure 1) would bring our deletion FDR to nearly zero.

A notable difference between the LUMPY framework and other tools that can incorporate multiple signals such as GASVPro (paired-end and read-depth) and DELLY (paired-end and split-read), is the point at which the second (or third) signal is considered. LUMPY pools all signals before making breakpoint predictions, while both GASVPro and DELLY consider signals in order; the paired-end signal is used to call breakpoints and the second signal refines those calls. An implication of ordered signal consideration is that the extra signals cannot increase the number of true positive predictions. In a case where evidence for a particular breakpoint exists in multiple

signals, but not above some threshold in any one signal, ordered consideration can not combine those signals to make a true positive prediction. In contrast, when signals are pooled before predictions are made the amount of evidence for that breakpoint can meet the threshold. This is especially important for cases of low sequence coverage. Our simulation results showed that true positives for deletions at 2x coverage decreased by 60 percent when signals were ordered (from 45 in DELLY PE to 18 in DELLY SR), and increased over six fold when both signals were pooled (from 32 in LUMPY-PE and 39 in LUMPY-SR to 245 in LUMPY-PESR).

In this paragraph we want to 1. summarize why this is an advance for SV detection, with regards to sensitivity, high resolution breakpoint prediction, using all possible signals from current and future sequence data, and incorporating prior probability basied on external sources of evidence ; 2. point out how useful the lumpy framework is for integrating different types of genomic data (ChIP-seq, etc?). If we don't include point2, it needs to be removed from the abstract.

Given the increased sensitivity, efficiency, and ease of use of our framework, we emphasize its utility for comprehensive SV discovery in human genomes. Increased sensitivity is especially important for cancer studies, as tumors are often comprised of highly heterogeneous genomes. As we demonstrate clearly superior sensitivity at low coverage (which is a proxy for low frequency rearrangements), we argue that our framework well-suited to detailed studies of tumor heterogeneity and somatic genome variability.

5 Methods

We propose a breakpoint prediction framework that can accommodate multiple classes of evidence from multiple sources in the same analysis. To accomplish this, we define a high-level breakpoint type that represents the consensus breakpoint location from different pieces of evidence. Our framework makes use of an abstract breakpoint evidence type to define a set of functions that serve as an interface between specific evidence subtypes (e.g., paired-end sequence alignments and split-read mappings) and the breakpoint type. Any class of evidence for which these functions can be defined may be included in our framework. To demonstrate the applicability of this abstraction, we defined three breakpoint evidence subtypes: paired-end sequencing, split-read mapping, and a general breakpoint interface.

Since our framework combines evidence from multiple classes, it extends naturally to include evidence from multiple sources. The sources that can be considered in a single analysis may be any combination of evidence from different samples, different evidence subclasses from the same samples, or data sets from known genomic features. We refer to a given data set as a breakpoint evidence instance, and assume that each instance contains only one evidence subtype and is from a single sample. To help organize the results of analysis with multiple samples or multiple instances for a single sample, each instance is assigned an id that can be shared across instances.

5.1 Breakpoint

A breakpoint is a pair of genomic sequences that are adjacent in a sample genome but not in a reference genome. Breakpoints can be detected, and their locations predicted by various evidence classes (e.g., paired-end sequence alignments and split-read mappings). To support the inclusion of different evidence classes into a single analysis, we define a high-level breakpoint type as a collection of the evidence that corroborates the location and variety of a particular breakpoint. Since many evidence classes provide a range of possible breakpoint locations, we represent the breakpoint's

location with a pair of breakpoint intervals where each interval has a a start position, an end position, and a probability array that represents the likelihood that a given position in the interval is one end of the breakpoint. More formally, a breakpoint is a tuple $b = \langle E, l, r, v \rangle$ where

- E is the set of evidence that corroborates the location and variety of a particular breakpoint
- l and r are left and right breakpoint intervals each with values $\langle s, e, p \rangle$ where
  - s and e are the start and end genomic coordinates
  - p is a probability array where $|p| = e - s$ and $p[i]$ is the relative probability that position $s + i$ is one end of the breakpoint
- v is the breakpoint variety (e.g., DELETION, DUPLICATION, etc.)

If there exits two breakpoints b and c in the set of all breakpoints B where b and c intersect ($b.r$ intersects $c.r$, $b.l$ intersects $c.r$, and $b.v = c.v$), then b and c are *merged* into interval m, b and c are removed from B, and m is placed into B. The merged breakpoint m is defined as $\langle E = b.E + c.E, l_n, r_n, v = b.v = c.v \rangle$, where $l_n.s = \max(b.l.s, c.l.s)$, $l_n.e = \min(b.l.e, c.l.e)$, similar for r_n. Once all evidence has been considered, the breakpoints in B are enumerated. Since each genomic interval has a probability array associated with it, the intervals may be trimmed to include only the positions that meet are in the top percentile (e.g., top 99.9 percent of values).

5.2 Breakpoint Evidence

To combine the distinct SV alignment signals like paired-end and split-read alignments to the general breakpoint type defined above, we define an abstract breakpoint evidence type. This abstract type defines an interface that allows for the inclusion of any data that can provide the following functions:

- IS_BP determines if a particular instance of the data contains evidence of a break point
- GET_V determines the breakpoint variety (e.g., deletion, duplication, inversion, etc.)
- GET_BPI maps the data to a pair of breakpoint intervals

To demonstrate the applicability of this abstraction, we defined three breakpoint evidence instances: paired-end sequencing alignments, split-read mapping, and a general breakpoint interface. Paired-end sequencing and split read mapping are among the most frequently used data types for breakpoint detection, and the general interface provides a mechanism to include any other breakpoint information such as known breakpoints or output from other analysis pipelines. As technologies evolve and our understanding of structural variations improves, other instances can be easily added.

5.2.1 Paired-End Sequencing Alignments

Paired-end sequencing involves fragmenting genomic DNA into roughly uniformly sized segments, and sequencing both ends of each segment to produce the sequence pair $\langle x, y \rangle$. The ends of the pair are aligned to a reference genome $R(x) =< o, s, e >$, where $o = +|-$ indicates the alignment orientation, and s and e delineate the start and end positions of the matching sequence in the reference genome. To simply the explanation, we let the genome be one contiguous interval of concatenated chromosomes so that all sequences can be referred to by offset only. Translocations

can still be identified in this model since the positions on different chromosomes will be far apart. We also assume that both x and y align uniquely to the reference and that $R(x).s < R(x).e < R(y).s < R(y).e$. While it is often not possible find the exact position of a sequence in the sample genome, it is useful to refer to $S(x) =< o, s, e >$ as the alignment of x with respect to the originating sample's genome.

Assuming the reads were made on an Illumina platform, pairs are expected to align to the reference genome with a $R(x).o = +, R(y).o = -$ orientation, and at distance $R(y).e - R(x).s$ roughly equivalent to the fragmentation length from the sample preparation step. Any pair that aligns with an unexpected configuration can be evidence of a breakpoint. These unexpected configurations include matching orientation $R(x).o = R(y).o$, alignments with switched orientation $R(x).o = -, R(y).o = +$, and an apparent fragment length $(R(y).e - R(x).s)$ that is either shorter or longer than expected. We estimated the expected fragment length to be the sample mean $\bar{l}$ fragment length, and the fragment length standard deviation to be the sample standard deviation $\bar{s}$ from the set of properly mapped pairs (as defined by the SAM spec) in the sample data set. Considering the variability in the sequencing process, we extend the expected fragment length to include sizes $\bar{l} \pm v_l\bar{s}$, where v_l is a tuning parameter that reflects spread in the data.

The breakpoint variety for $\langle x, y\rangle$ can be inferred from the orientation that x and y align to in the reference. If the orientations match, then the breakpoint was caused by an inversion event, and if the $R(x).o = -$ and $R(y).o = +$ then there was a duplication event. When $R(x).o = +$ and $R(y).o = -$, the breakpoint variety is ambiguous between an insertion and a deletion. This ambiguity is also true for other types of evidence types (e.g., split-read mappings). While it may be possible to determine which event caused the breakpoint in a post-processing step, breakpoint correlation is a complex process and is beyond the scope of this framework. Since we cannot distinguish between the two varieties, any pair with a $+/-$ orientation configuration is marked as a deletion.

To map $\langle x, y\rangle$ to breakpoint intervals l and r, the ranges of possible breakpoint locations must be determined and probabilities assigned to each position in those ranges. By convention, x maps to l and y to r, and for the sake of brevity we will focus on x and l since the same process applies to y and r. Assuming that a single breakpoint exists between x and y, then the sign of x determines if l will be upstream or downstream of x. If the $R(x).s = +$, then the breakpoint interval begins after $R(x).e$ (downstream), otherwise the interval ends before $R(x).s$ (upstream).

The length of each breakpoint interval is proportional to the expected fragment length $\bar{l}$ and standard deviation $\bar{s}$. Since we assume that only one breakpoint exists is between x and y, and that it is unlikely that the distance between the ends of a pair in the sample genome $(S(y).e - S(x).s)$ is greater than $\bar{l}$, then it is also unlikely that one end of the breakpoint is at a position greater than $R(x).s + \bar{l}$, assuming that $R(x).o = +$. If $R(x).o = -$, then it is unlikely that a breakpoint is at a position less than $R(x).e - \bar{l}$. To account for variability in the fragmentation process, we extend the breakpoint to $R(x).e + (\bar{l} + v_f\bar{s})$ when $R(x).o = +$, and $R(x).s - (\bar{l} + v_f\bar{s})$ when $R(x).o = -$, where v_f is a tuning parameter that, like v_l, reflects the spread in the data.

The probability that a particular position i in the breakpoint interval l is part of the actual breakpoint can be estimated by the probability that x and y span that position in the sample. For x and y to span i, the fragment that produced $\langle x, y\rangle$ must be longer than then distance from the start of x to i, otherwise y would occur before i and x and$y$ would not span i (contradiction). The resulting probability is $P(S(y).e - S(x).s > i - R(x).s)$ if $R(x).o = +$, and $P(S(y).e - S(x).s > R(x).e - i)$ if $R(x).o = -$. While we cannot directly measure the sample fragment length $(S(y).e - S(x).s)$, we can estimate its distribution by constructing a frequency-based cumulative distribution D of fragment lengths from the same sample that was used to find $\bar{l}$ and $\bar{s}$, where $D(j)$ gives the proportion of the sample with fragment length greater than j(Algorithm 1 and

Algorithm 2).

5.2.2 Split-Read Alignments

A split-read alignment is a single DNA fragment X that does not uniquely align to the reference genome, but contains a contiguous ordered set of substrings $(x_1, x_2, \ldots, x_n)$ where $X = x_1 x_2 \ldots x_n$, each substring aligns uniquely to the reference $R(x_i) = \langle o, s, e \rangle$, and adjacent substrings align to non-adjacent location in the reference genome $R(x_i).e \neq R(x_{i+1}).s + 1$ for $1 \leq i \leq n - 1$. A single split-read alignment maps to a set of adjacent split-read sequence pairs $(\langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle, \ldots, \langle x_{n-1}, x_n \rangle)$, and each pair $\langle x_i, x_{i+1} \rangle$ is considered individually.

By definition, a split-read mapping is evidence of a breakpoint and therefore the function IS_BP trivially returns TRUE.

Both orientation and mapping location must be considered to infer the breakpoint variety for $\langle x_i, x_{i+1} \rangle$. When the orientations match $R(x_i).o = R(x_{i+1}).o$, the event was either a deletion or a duplication. Assuming the $R(x_i).o = R(x_{i+1}).o = +$, $R(x_i).s < R(x_{i+1}).s$ indicates a gap caused by a deletion and $R(x_i).s > R(x_{i+1}).s$ indicated a repeated sequenced caused by a duplication. These observations are flipped when orientations $R(x_i).o = R(x_{i+1}).o = -$. Similar to paired-end alignments, we do not mark breakpoints as insertions since we cannot distinguish between deletions and insertions. When the orientations do not match $R(x_i).o \neq R(x_{i+1}).o$, the event was an inversion and the mapping locations do not need to be considered.

The possibility of errors in the sequencing and alignment processes create some ambiguity in the exact location of the breakpoint associated with a split-read sequence pair. To account for this, each pair $\langle x_i, x_{i+1} \rangle$ maps to two breakpoint intervals l and r centered at the split. The probability vectors $l.p$ and $r.p$ are highest at the midpoint and exponentially decreasing toward their edges. The size of this interval is a configurable parameter v_s and is based on the quality of the sample under consideration and the specificity of the alignment algorithm used to map the sequences to the reference.

Depending the breakpoint variety, the intervals l and r are centered on either the start or the end of $R(x_i)$ and $R(x_{i+1})$. When the breakpoint is a deletion l is centered at $R(x_i).e$ and r at $R(x_{i+1}).s$, and when the breakpoint is a duplication l is centered at $R(x_i).s$ and r at $R(x_{i+1}).e$. If the breakpoint is an inversion, l and r are both centered either at the start positions or end positions of $R(x_i)$ and $R(x_{i+1})$, respectively. Assuming that $R(x_i).s < R(x_{i+1}).s$, if $R(x_i).o = +$ then l and r are centered at $R(x_i).e$ and $R(x_{i+1}).e$, otherwise they are centered at $R(x_i).s$ and $R(x_{i+1}).s$. If $R(x_i).s > R(x_{i+1}).s$, then the conditions are swapped (Algorithm 3).

5.2.3 Generic Evidence

The generic evidence subclass provides a mechanism to directly encode breakpoint intervals using the BEDPE format [12]. BEDPE is an extension of the popular BED format that provides a means to specify a pair of genomic coordinates, in this case the pair is a breakpoint. This subclass extends our framework to include SV signal types that do not have a specific subclass implemented yet. For example, a copy number variation prediction algorithm may report segments of the genome that are either duplicated or deleted. This signal can be included in the analysis by expanding the edges of the predicted intervals to create a breakpoints, and encoding that breakpoints in BEDPE format. Each BEDPE entry is assumed to be real breakpoint(IS_BP), the variety is encoded in the auxiarly fields in BEDPE (GET_V), and the intervals are directly encoded in BEDPE (GET_BPI).

5.2.4 Simulation

Simulated data was used to compare the sensitivity and false discovery rate of LUMPY to other SV detection algorithms that rely on either a single signal (Hydra and DELLY PE) or multiple signals (GASVPro and DELLY SR). The seed sequence for all simulations was chromosome 10 from the human reference genome (hg19). For each SV variety considered (deletions, duplications, insertions, and inversions), we used SVsim [XX] to simulate a new version of the seed that contained 1000 randomly placed, non-overlapping variants ranging between 100 bp and 10000 bp. Next, wgsim [XX] was used to sample pair-end reads with a 150bp read length, 500 bp mean outer distance with a 50 bp standard deviation, and default error rate settings. Each simulated genome was sampled to 20x, 5x, and 2x coverage. Paired-end reads were mapped to the seed sequence with BWA using default parameters. From the BWA output, all split-reads and unmapped reads were realigned with the split-read aligner YAHA using a word length of 11 and a minimum match of 15. The TWA output was used as input to LUMPU PE (paired-end), Hydra, DELLY (both versions) and GASVPro, the YAHA output was used as input to LUMPY SR (split-read), and both BWA and YAHA output were used as input to LUMPY PESR (paired-end and split-read). In all algorithms, the minimum evidence threshold was four. For LUMPY, the turning parameters v_l and v_f were set to four, and alignments with mapping qualities equal to zero were not considered.

The reads predicted by each algorithm were compared to the events produced by SVsim. A true positive was a predicted breakpoint that intersected both ends of a simulated breakpoint, all other predictions were considered to be false positives, and all other missed simulated events were false negatives. Since the output of DELLY is a single interval, we took the 100 bp regions flanking the ends of the predicted interval as the predicted breakpoint. A similar conversion was performed for Hydra.

References

[1] S. A. McCarroll and D. M. Altschuler, "Copy-number variation and association studies of human disease.," *Nature Genetics*, vol. 39, pp. S37–S42, 2007.

[2] T. Rausch *et al.*, "Genome sequencing of pediatric medulloblastoma links catastrophic dna rearrangements with tp53 mutations.," *Cell*, vol. 418, no. 1, pp. 59–71, 2012.

[3] C. Alkan *et al.*, "Genome structural variation discovery and genotyping.," *Nature Reviews Genetics*, vol. 5, pp. 363–376, 2011.

[4] H. Li and R. D. Durbin, "Fast and accurate short read alignment with burrowswheeler transform.," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[5] G. G. Faust and I. M. Hall, "YAHA: fast and flexible long-read alignment with optimal breakpoint detection," *Bioinformatics*, vol. 28, no. 19, pp. 2417–2424, 2012.

[6] H. Li and R. D. Durbin, "Fast and accurate long read alignment with burrows-wheeler transform," *Bioinformatics*, vol. 26, no. 5, pp. 589–595, 2010.

[7] A. R. Quinlan *et al.*, "Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome.," *Genome Research*, vol. 20, pp. 623–635, 2010.

[8] S. S. Sindi *et al.*, "An integrative probabilistic model for identification of structural variation in sequencing data.," *Genome Biology*, vol. 13, p. R22, 2012.

[9] T. Rausch *et al.*, "Delly: structural variant discovery by integrated paired-end and split-read analysis.," *Cell*, vol. 28, pp. 333–339, 2012.

[10] R. E. Mills *et al.*, "Mapping copy number variation by population-scale genome sequencing.," *Nature*, vol. 470, pp. 59–65, 2011.

[11] The 1000 Genomes Project Consortium, "A map of human genome variation from population-scale sequencing," *Nature*, vol. 467, no. 7319, pp. 1061–1073, 2010.

[12] A. R. Quinlan and I. M. Hall, "BEDTools: a flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, no. 6, pp. 841–842, 2011.

A Algorithms

A.1 Paired-End Sequencing Alignments

Algorithm 1: Breakpoint evidence function that maps one end of a sequence pair to one end of a breakpoint interval.

Input: Reference genome R, One end of a sequence pair z, expected fragment length $\bar{l}$ and standard deviation $\bar{s}$, tuning parameter v_f, fragment length cumulative distribution D
Output: One end of a breakpoint interval t

Function GET_ONE_BPI
begin
  if $R(x).o = +$ **then**
    $t.s \leftarrow R(z).e$
    $t.e \leftarrow R(z).e + \bar{l} + v_f * \bar{s}$
    for $i = 1 \rightarrow (t.e - t.s)$ **do**
      $t.p[i] \leftarrow D(j)$
    end
  else
    $t.e \leftarrow R(z).s$
    $t.s \leftarrow R(z).s - (\bar{l} + v_f * \bar{s})$
    for $i = 1 \rightarrow (l.e - l.s)$ **do**
      $t.p[(t.e - t.s) - i] \leftarrow D(j)$
    end
  return t
end

Algorithm 2: Breakpoint evidence function that maps a sequence pair alignment to a breakpoint interval.

Input: Reference genome R, Sequence pair $\langle x, y \rangle$, expected fragment length $\bar{l}$ and standard deviation $\bar{s}$, tuning parameter v_f, fragment length cumulative distribution D
Output: Breakpoint intervals l and r

Function GET_BPI
begin
  $l \leftarrow$ GET_ONE_BPI$(R, x, \bar{l}, \bar{s}, v_f, D)$
  $r \leftarrow$ GET_ONE_BPI$(R, y, \bar{l}, \bar{s}, v_f, D)$
  return l, r
end

A.1.1 Split-Read Alignments

Algorithm 3: Breakpoint evidence function that maps a sequence pair alignment to a breakpoint interval.

Input: Reference genome R, Split-read pair $\langle x_i, x_{i+1} \rangle$, tuning parameter v_s, breakpoint variety v
Output: Breakpoint intervals l and r

Function GET_BPI
begin
  $l_c \leftarrow NULL\ r_c \leftarrow NULL$
  if $v =$ INVERSION **then**
    if $R(x_i).s < R(x_{i+1}).s$ **then**
      if $R(x_i).o = +$ **then** $l_c \leftarrow R(x_i).e, r_c \leftarrow R(x_{i+1}).e$
      else $l_c \leftarrow R(x_i).s, r_c \leftarrow R(x_{i+1}).s$
    else
      if $R(x_i).o = +$ **then** $l_c \leftarrow R(x_i).s, r_c \leftarrow R(x_{i+1}).s$
      else $l_c \leftarrow R(x_i).e, r_c \leftarrow R(x_{i+1}).e$
    end
  else if $v =$ DELETION **then** $l_c \leftarrow R(x_i).e, r_c \leftarrow R(x_{i+1}).s$
  else if $v =$ DUPLICATION **then** $l_c \leftarrow R(x_i).s, r_c \leftarrow R(x_{i+1}).e$
  $l.s \leftarrow l_c - v_s, l.e \leftarrow l_c + v_s$
  $r.s \leftarrow r_c - v_s, r.e \leftarrow r_c + v_s$
  $\lambda = \log(1e-10)/-v_s$
  for $i = 1 \rightarrow v_s$ **do**
    $l.p[i] \leftarrow r.p[i] \leftarrow \exp^{-\lambda(v_s - i)}$
  end
  for $i = v_s \rightarrow 2 * v_s$ **do**
    $l.p[i] \leftarrow r.p[i] \leftarrow \exp^{-\lambda(i - v_s)}$
  end
  return l, r
end