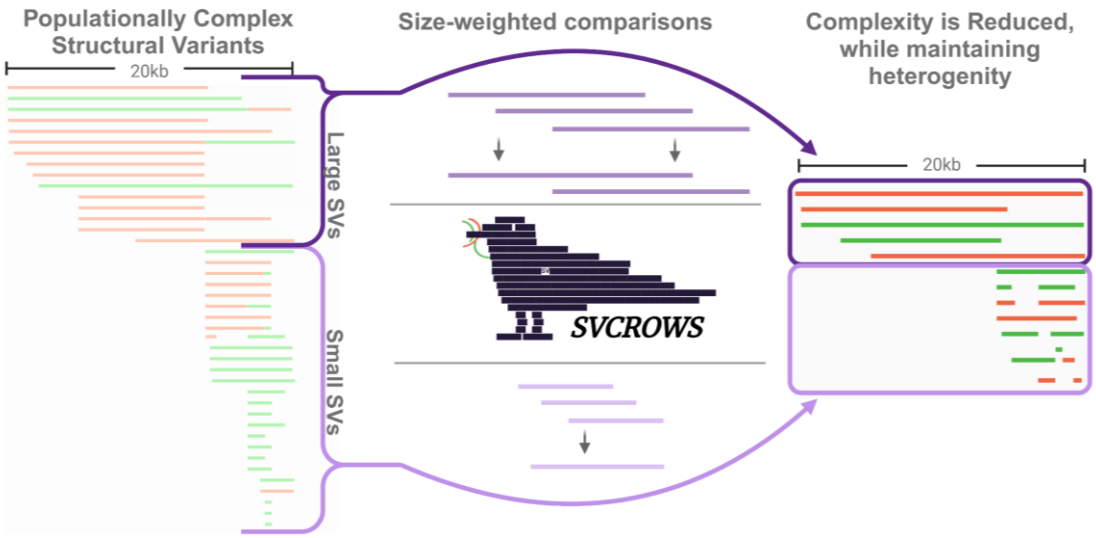
# SVCROWS: A User-Defined Tool for Interpreting Significant Structural Variants in Heterogeneous Datasets

**Authors:** Noah Brown*^, Charles Danis*, Vazira Ahmedjanova*, Jennifer L. Guler*^
**Affiliations:** *Department of Biology, University of Virginia. Charlottesville, VA 22903
^Co-corresponding authors




**Graphical Abstract**

## Abstract

**Summary:** Structural variants (SVs) are abundant across all life, and have major impacts on the genome and transcriptome. However, it is difficult to appreciate the individual significance of SVs when they are heterogeneously distributed across a genomic neighborhood. Further, low-input sequencing technologies or sequencing of many individuals across a population introduce variance that complicates SV counting and association studies. Tools exist to simplify SV datasets, but these SV mergers begin to fail on large or highly variable datasets. To address this issue, we introduce a new SV merger called SVCROWS (Structural Variation Consensus with Reciprocal Overlap and Weighted Sizes). This option-rich R package merges and summarizes SV regions using a size-weighted reciprocal overlap framework, effectively accounting for skewed impacts of variable-length SVs. User input directs stringency of comparisons across a range of sizes, enabling different levels of resolution in complex genome regions that harbor both small and large SVs. When compared to other SV merging programs, SVCROWS accurately merges SVs while maintaining less frequent genotypes of the unmerged SV calls. SVCROWS proves to be especially useful with large and highly variable single-cell datasets for enabling SV discovery. Overall, the novel size-weighted comparisons of SVCROWS presents a framework for improved interpretation of SV calls, and its ease of use allows it to be applied to virtually any upstream analyses.

## Introduction

Genomic structural variations (SVs) are large (>50 bp) duplications, deletions, inversions, insertions, and/or translocations. SVs are both evolutionarily important for organisms and causative factors in many human disease[1–5]. Their abundance in all life makes them prime targets of study. Recent advances in sequencing technologies have improved detection of SVs, as well as the sequence accuracy and breakpoint mapping of those calls[6]. Validation of these programs and protocols has become more accessible using benchmarking datasets like the *Genome in a Bottle* consortium[7], and the *1000 Genomes Project* consortium[8]. However, newly developed tools and benchmarking programs largely focus on single-sample SV detection (i.e. from bulk sequencing), abstaining from more complex comparisons across individuals in a population (i.e. single-cell (SC) genomics or metagenomics).

As studies focusing on the association between rare genotypes and phenotype become more common, researchers must evaluate larger datasets with increased heterogeneity. Therefore, SV merging from multi-sample data is a key step prior to SV interpretation including checking for an expected genotype (i.e. validation) or detecting new genotypes (i.e. discovery). The outcome of the merging is an influential step in many analysis pipelines. In the case of both validation and discovery, merged SVs have greater statistical strength to make associations with a phenotype[6,9–12]. Determining how to merge SVs is challenging because small shifts in their position can have major impacts on SV interpretation. Accurate SV merging must maintain a balance between removing redundancy across a dataset by merging similar SVs and preserving heterogeneous and rare SVs.

How best to address merging challenges is still debated, but strides have been made by several SV merging algorithms to reconcile this issue[10,13–19]. However, these existing approaches have limitations. Firstly, many rely on high-depth coverage and have only been validated in that context. When sequencing depth is limited (e.g. for SC genomics), the reference genome is low quality, or SVs sit in highly repetitive regions of the genome, these programs have reduced effectiveness[20]. Large population studies also remain challenging despite high sequencing quality, because the increased SV heterogeneity leads to overmerging[14,18,20]. Another common limitation is incompatibility of mergers with different data types because many mergers are incorporated into larger analysis pipelines to increase effectiveness. This places constraints on the input type or upstream SV caller, invalidating certain study designs. Further, while SV mergers generally have options to control SV matching parameters, their direct impact is not immediately transparent to the user without performing several iterations. Overall, overmerging of variable regions across many samples leads to a loss of information, or generates misleading information. Therefore, flexible SV merging programs that effectively resolve variance are critical to defining accurate genotype-phenotype relationships in the current big data era.

Here, we introduce a new SV-merger, 'SVCROWS' (Structural Variation Consensus with Reciprocal Overlap and Weighted Sizes) that fulfills these criteria. SVCROWS merges SVs into representative SV regions (SVRs), in a way that simplifies irrelevant variation while preserving rare alleles. Additionally, size-weighted reciprocal overlap (RO) comparisons allow the user to control stringency based on the study design. Because SVCROWS incorporates multiple lines of evidence to construct SVRs, this application is particularly useful for resolving multi-sample datasets and parsing high-variance regions that require a more granular evaluation. SVCROWS is an R-package that uses an original input format that is compatible with virtually all SV calling platforms. Overall, the size-weighted comparisons introduced in SVCROWS offers a new and relevant framework for SV merging from any input data format.

## Materials and Methods

### A framework of size-weighted comparisons to capture significant SVs

SVCROWS uses the principle of RO thresholds to determine whether to place SVs in the same SVR (i.e. matching/merging SVs). While the RO-based approach is used in existing applications[19,21,22], these rely on a single, fixed RO threshold value. This static method overlooks the disparity in SV lengths and their potential biological impacts, creating a false equivalency between small and large SVs. For instance, if we measure biological impact by the number of genes within an SV (**Fig. 1A-B**), a larger SV is more likely to have a greater effect (e.g. encompasses more genes and regulatory regions) on the phenotype than a smaller SV (e.g. encompasses few genes or even a promotor region). Specifically, as depicted in **Figure 1A-B**, there are two obvious inconsistencies in treating different sized SVs with the same stringency: 1) SV2 matches both large and small reference SVs, despite causing a 3X increase in potential impact in the larger SV (i.e., involving 3 genes); 2) When either SV size lacks just one gene compared to the reference, the small SV matches at a 55% RO (**Fig. 1A,** SV2) while the larger SV matches at 95% (**Fig. 1B,** SV1). Using this reasoning, we posit that when a size-static matching parameter like RO is used, it is more likely to reflect genotypes disproportionate to their impact. This is basis for the size-weighted caparisons used by SVCROWS.

### Selecting parameters to dictate stringency using size-weighted comparisons

To address high variance in SV length and position, the user directs SVCROWS to increase or decrease the requirements of the RO threshold and other factors involved in the decision to merge two SVs (i.e. stringency) according to the SV size (**Fig. 1C**). Dynamic RO thresholding by SVCROWS involves 6 input parameters to define the stringency based on analysis needs (**Fig. 1C,** colored boxes). First, SVCROWS creates size categories to group SV-comparison stringencies by SV size. By default, these parameters are determined by the first and third quartiles of SV length in the given dataset. These can be further refined based on the minimum and maximum SV sizes the user considers relevant in their organism or study interest. For example, in an analysis of CNVs the 'small' category could be the average exon size, while the 'large' category could reflect the average size of CNVs reported in that genome. (**Fig. 1C,** blue inputs 1&2). Secondly, users dictate bounds that allow SVCROWS to match breakpoints between input SVs (**Fig. 1C,** green inputs 3&4, **Fig. 1A**). Finally, users define RO thresholds at the boundaries of the size categories (**Fig. 1C,** yellow inputs 5&6). Collectively, these inputs set the SVCROWS algorithm to respond to SV size (**Fig. 1C** red line). More details on parameters and functions are provided in the ***Supplemental Methods***.

### SVR consensus generation

SVCROWS merges SVs and organizes data into a table of distinct SVRs according to the input parameters. The algorithm processes SVRs in descending order of length (**Fig. 1D, top**). Therefore, the largest SV in the region dictates boundaries, and the breakpoints reflect those provided by the upstream SV-caller. When used as recommended, the stringency is higher for larger SVs than smaller SVs (**Fig 1D**); though stringency can be assigned so that the opposite is true as well. SVCROWS quantifies matching SVs on the number of RO and SV breakpoint (BP) matches at 5' and 3' ends of an SV (**Fig. 1D,** bottom).

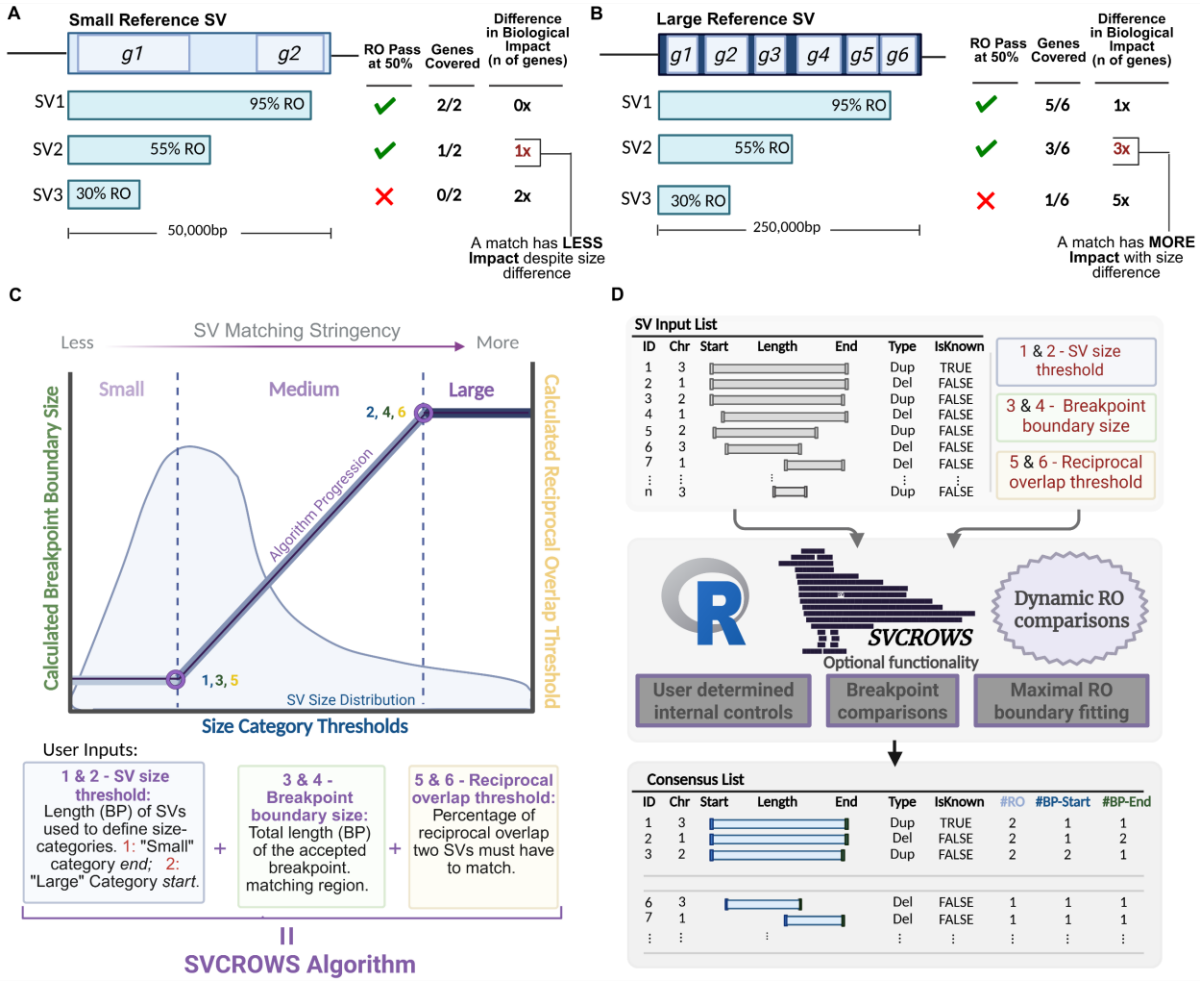



**Figure 1. The SVCROWS algorithm compares SVs with regard to size, and potential impact. A)** A small (50,000bp) reference SV (light blue) encompassing 2 genes and, **B)** A large (250,000bp) reference SV (dark blue) encompassing 6 genes; being compared to 3 potential SVs (teal) at a static 50% Reciprocal Overlap threshold. Using each gene as a unit of biological impact; smaller SVs are more tolerant to changes in size when reflecting similarity to the original SV. **C**) Six inputs defined by the user weight the reciprocal overlap determination by size. x-axis: "Small" and "Large" SV category thresholds (variables 1&2). Left y-axis: The width (in base pairs) of BP-Matching region (see **Fig. S1,** variables 3&4 respectively). Right y-axis: The required level of RO-threshold (as a percentage) for SV matching (variables 3&4 respectively). These inputs generate the SVCROWS Calculation Trajectory (purple) which defines the dynamic RO stringencies. **D)** Logical workflow of SVCROWS merging function. A list of SVRs is produced, from a single input list. The number of matches is tabulated (bottom). 'Dup' = duplication, 'Del' = Deletion, '#RO' = Number of matching SVs, 'BP-Start/End' = matching breakpoints on the 3'/5' ends of the SVs. Options to augment the matching style are also available.

**Optional inputs and functions**

In addition to size-weighted comparisons, SVCROWS incorporates several features to construct and quantify SVRs. While some features mirror existing functionality, SVCROWS integrates these functions into a single platform and provides other novel functionality (denoted with * below). Seven major optional features are outlined below, with detailed descriptions available in the ***Supplemental Methods***:

i. *Automatic input parameter calculation: SVCROWS calculates six input parameters based on the first and third quartiles of input SV sizes. This allows SVCROWS to dynamically adjust to the specific characteristics of the dataset.

ii. *Breakpoints as a secondary SV matching factor: Users can opt to incorporate breakpoint matching to aid SV merging (**Fig. S1A**). When enabled, SVCROWS reduces the required RO threshold to its minimum value (user input 5) for SVs with matching breakpoints. This feature is useful for identifying matches in repetitive genomic regions where one breakpoint is consistent, but the other is misaligned.

iii. *Use of 'Known' regions in input: A user can define input 'known' SVs, which reduces all requirements for matching in that region to only a single base pair (**Fig. S1B**). For example, by defining the region of *AMY1* in the human genome as a 'known' CNV[23], a user can remove all variation in this genomic neighborhood.

iv. *Convenient annotation of all input SVs by sample: SVCROWS uses a niche input format that allows the program to individually track the outcome of all input SVs. This is especially convenient for workflows that require a per-sample analysis.

v. Expansion of SVRs to maximum size: The user can employ this option to set SVR size based on the maximum and minimum matching breakpoint of all SVRs in a region (**Fig. S1C**). This functionality may be useful when downstream applications benefit from larger SVRs (e.g., gene ontology studies).

vi. Querying against a list of features: The user can alternatively use the "Hunt" function, which takes a secondary input, a 'feature list', for comparison with an input list of SVs (or SVRs). For example, this functionality allows the user to compare a set of genes to SVRs determined from "Scavenge mode" using the size-weighted framework.

vii. Enumeration of other variables: As SVs match, the user can provide numeric information that 1) takes the number or reads originally used to call the SV and adds them together and 2) takes the quality score of each SV and averages them together.

**HG002 dataset processing and analysis**

The HG002 SV Truth Set (HG002_SVs_Tier1_v0.6.tsv) is part of the *Genome in a Bottle* (GIAB) consortium, a high confidence, multi-validation genome assembly[24–28]. This dataset consists of 9641 individual SVs (>50bp) confirmed by at least 3 independent SV calling programs, including both long- and short-read data. For this analysis, we used only Tier1 (GIAB certified highest confidence) SVs, including representative SVs that allow assessment of merger efficacy in regions with multiple SV calls across the same region. The HG002 SV Truth Set includes both deletions and insertions, but because not every merger can handle insertions, we excluded them from this analysis.

To test SV mergers, we used HG002 genome SV calls from 5 different programs as input: LUMPY[29], pbsv[30], Sniffles[31], MrCaNaVaR[32], and MetaSV[33]. We combined all calls into a single 'aggregate list' (termed the SV aggregate list), many of which are not part of the HG0002 SV Truth Set (~90%), either because they are false positive calls, or they did not reach the threshold to be called in the Truth Set (i.e. called by 3 programs). SV calls not represented in the HG002 SV Truth Set are included in the SV aggregate list in order to assess accuracy in the context of typical variance experienced across callers. Further, because the largest

SV in the HG002 SV Truth Set was ~1Mb, we excluded SVs >2Mb. This represents the manual filtering step, which is part of most SV calling pipelines.

We calculated Jaccard indices of the SV mergers using the Bedtools-jaccard method on the SV aggregate list, which essentially takes into account both the coverage and depth of represented regions to calculate differences in merging (see ***Supplemental Methods***).

To calculate merger accuracy, the we first further filtered the SV aggregate list to include only those that had overlapping bases with the HG002 SV Truth Set (termed 'on-truths', [VCFTools ∩ Truth_Set] ≥ 1bp) After merging the 'on-truths', we then overlapped the SVRs against the HG002 SV Truth Set using bedtools at 3 stringencies of static RO (50%, 80%, and 95%). We counted true positive/negative merges (uniquely matched against the HG002 SV Truth Set), false negatives (non-unique matches to HG002 SV Truth Set), and false positives (SVRs that had no matches to the HG002 SV Truth Set) and used these numbers to calculate the F1 score (see **Fig. 1A** and ***Supplemental Methods***).

**Single Cell dataset acquisition and SV calling**

We used Pyega3 download the single-cell ovarian cancer dataset, EGAD00001009455, with permission of the European Genome Agency (EGA)[34]. The authors note their compliance with best practices detailed by the EGA. The dataset consists of 326 single cell (SC) whole genome sequences derived from a solid ovarian cancer tumor from a single patient (termed SC dataset). Notably the patient also had triple negative breast cancer. The full Galaxy-based pipeline can be found at https://usegalaxy.eu/u/sillycrow/w/lumpy. Briefly, we used downloaded BAM files to generate files containing only split and discordant reads as determined by previous sequence alignments. Then, all three BAM files are used as input into the LUMPY SV caller. LUMPY was chosen for this analysis due to its high precision breakpoints and utility in single-cell datasets[29,35,36]. We used default parameters except for the expected read length (150 base pairs) and the minimum read mapping quality for inclusion (≥5). For this analysis, we excluded BAM files with less than 1000 elements in either the split-read or discordant-read files due to issues with LUMPY performance in low-read support samples[37,38]. Additionally, we limited the SC analysis to only duplications and deletions with a length of less than 1Mb called by LUMPY for simplicity. Prior analysis of this dataset detected high frequency SVs with variable breakpoints, a common feature in cancers[39]. Overall, 4771 total SVs were detected across the dataset (**Tables S1 & S2**).

**SV merging for HG002 SV Truth Set and SC dataset**

Our study compares SVCROWS to several SV mergers with a variety of algorithm types: Jasmine[17], Survivor[40], SVimmer[15], Truvari[18], and CNVRuler[19]. We chose these five SV mergers because they are compatible with most upstream SV callers. Because each program uses a different framework for SV comparison, we instructed each program to function with parameters as close to the parameters of SVCROWS as possible; where the 'default' parameter assignment was used and the breakpoint matching and expansion of RO region options enabled (see **Fig. S1**). Default SVCROWS parameters use the first and third quartiles of SV sizes to demark size boundaries, 10% of respective lengths to set the size of breakpoint boundaries, and 40% and 70% as the small and large RO thresholds.

The precession of each program ranges from a single basepair to no more than 500bp. Because all mergers (excluding SVCROWS) include breakpoint distance to some capacity in merging decisions, we ran all mergers with a lenient 1000bp buffer region around either end of the SV; this is larger than previously used in order to encourage SV comparison[15]. See ***Supplemental Methods*** for the precise command-line run for each program.

We manually compiled the SV input lists for SVCROWS and CNVRuler[19] and organized them into a TSV files used for each run matching their individual formats (**Tables S1 & S2**). Jasmine[17], Survivor[40], and SVimmer[15] accept the raw VCF files from the SV caller. For the HG002 SV Truth Set analysis, SVs were first consolidated for Truvari[18] as recommended in their user wiki BCFTools[41], which essentially concatenates all SVs together without merging. We noted this concatenation step with BCFTools led to artifacts in complex regions in our dataset (see **Fig. S2B**), and opted to use a different BCFTools condition that did merge purely redundant SVs (i.e. those with precisely matching breakpoints) before further merging. This largely fixed the artifacts (See ***Supplementary Methods*** for more details). To normalize for any inconsistencies between output from each caller, all alternative alleles were manually converted to either "<DUP>" or "<DEL>", and we replaced the sample field with a ".".

For the SC dataset specifically, we tuned SVCROWS to interpret differences in the human genome by using the average size of a human exon (~1.3kb[42]) and the average size of a gene-plus-intergenic region (~38kb[43–45]) as small and large SV sizes, respectively (see **Table S3** for details of specific run parameters and output).

## Results

### SVCROWS preserves accurate and sensitive SVR calls compared to other SV mergers

Besides SVCROWS' size-weighted algorithm, we assessed performance from 4 major types of SV mergers including: distance-based algorithms (SVimmer[15] and Survivor[40]), a minimum spanning forest-based algorithm (Jasmine[17]), a multifactored algorithm (Truvari[18]), a static RO algorithm (CNVRuler[19]), as well as a pure redundancy removal algorithm (VCFTools[46]) denoted by color below. In order to directly compare SV merger performance, we employed the HG002 SV benchmarking dataset[28] form the (GIAB) consortium as an SVR 'Truth Set'. The test SV dataset consisted of simple, high-quality deletions compiled from 5 independent SV callers (see **Materials and Methods**). After SV merging using the 6 total tools, we compared the resulting SVR lists to each other, and then to the HG002 SV Truth Set (**Fig. 2A**).

We first used Jaccard indices to compare SV merger performance on the SV aggregate list. Higher Jaccard values reflect more similarity between lists generated by two different tools. Because each SV merger had the same input data, even small differences in the Jaccard index represent regional under- or overmerging. While, SVCROWS bared higher than average similarity to the other mergers (SVCROWS = 0.91, study average = 0.86), the wide range (0.62 – 0.99) of indices we observe denote marked differences in the performance of the SV mergers (**Fig. 2B, Table S4**). For example, Truvari was more dissimilar across all comparisons (index average = 0.70), and CNVRuler was relatively more similar to other mergers (index average = 0.81), but still less than the study average (0.86). Because CNVRuler and Truvari both reported far fewer total SVRs than the other mergers (**Fig. 2C**), it is possible that both programs overmerge SVRs. However, the Jaccard index between Truvari and CNVRuler is the lowest of the analysis (0.62) indicating algorithm-specific differences independently increased the rate of merging by each algorithm. Meanwhile, the similar SVimmer and Survivor algorithms exhibited a high Jaccard index (0.997). Overall, the Jaccard analysis emphasized the importance merging algorithm choice; despite the relative simplicity of the dataset, the internal architecture of the SV merging algorithm affects the final SVR list.

Each SV merger produced between ~70,000-80,000 SVRs from the SV aggregate list (**Fig. 2C**). As expected, due to their similar algorithms, Survivor and SVimmer yielded a similar number of SVRs. Meanwhile, Jasmine and SVCROWS yielded the most SVRs. When we compared the size distribution of resulting SVRs, all SV mergers produced a nearly identical SV size range except for Jasmine (**Fig. 2D**). Upon further investigation, Jasmine introduced SVR size alterations in the form of both truncations and extensions that

are not found in the un-merged SV aggregate list (**Fig. S2A**). Despite its dependence on size, the SVCROWS algorithm reflected a size distribution similar to other SV mergers, indicating a lack of size-related bias.

We also compared the merged SVRs to the HG002 SV Truth Set to assess merging accuracy using F1 scores. Jasmine and SVCROWS had the highest F1 scores of any SV mergers (**Fig. 2E**). In all 3 overlap stringencies, the number of true positive SVRs were ~30% higher for SVCROWS and Jasmine compared to other programs while false positives were 20-40% lower on average. SVCROWS and Jasmine had similar false positive rates (only 3% higher for SVCROWS, on average across all stringencies). However, Jasmine consistently had false negative merges where SVCROWS had none (**Table S5**). A lack of false negatives reflected SVCROWS' tendency to avoid undermerging (**Fig. 2A**) and therefore circumvents loss of statistical power in downstream analysis. Truvari's F1 score was the lowest in this analysis, which mirrors its low Jaccard indices (**Fig. S2B**). When we directly visualized Truvari's SVRs, we observed occasional merging artifacts that increase the rate of false negatives (**Fig. S2B**, examples 1&2). Further, in complex SVRs, Truvari fails to preserve genotypes from the SV aggregate list (**Fig. S2B**, example 3). Combined, these inconsistencies led to both under- and overmerging by the Truvari algorithm. Overall, using a high-quality dataset, SVCROWS strikes a balance between accuracy and sensitivity to maintain the potential for rare genotype discovery with no indication of SV size-related biases.

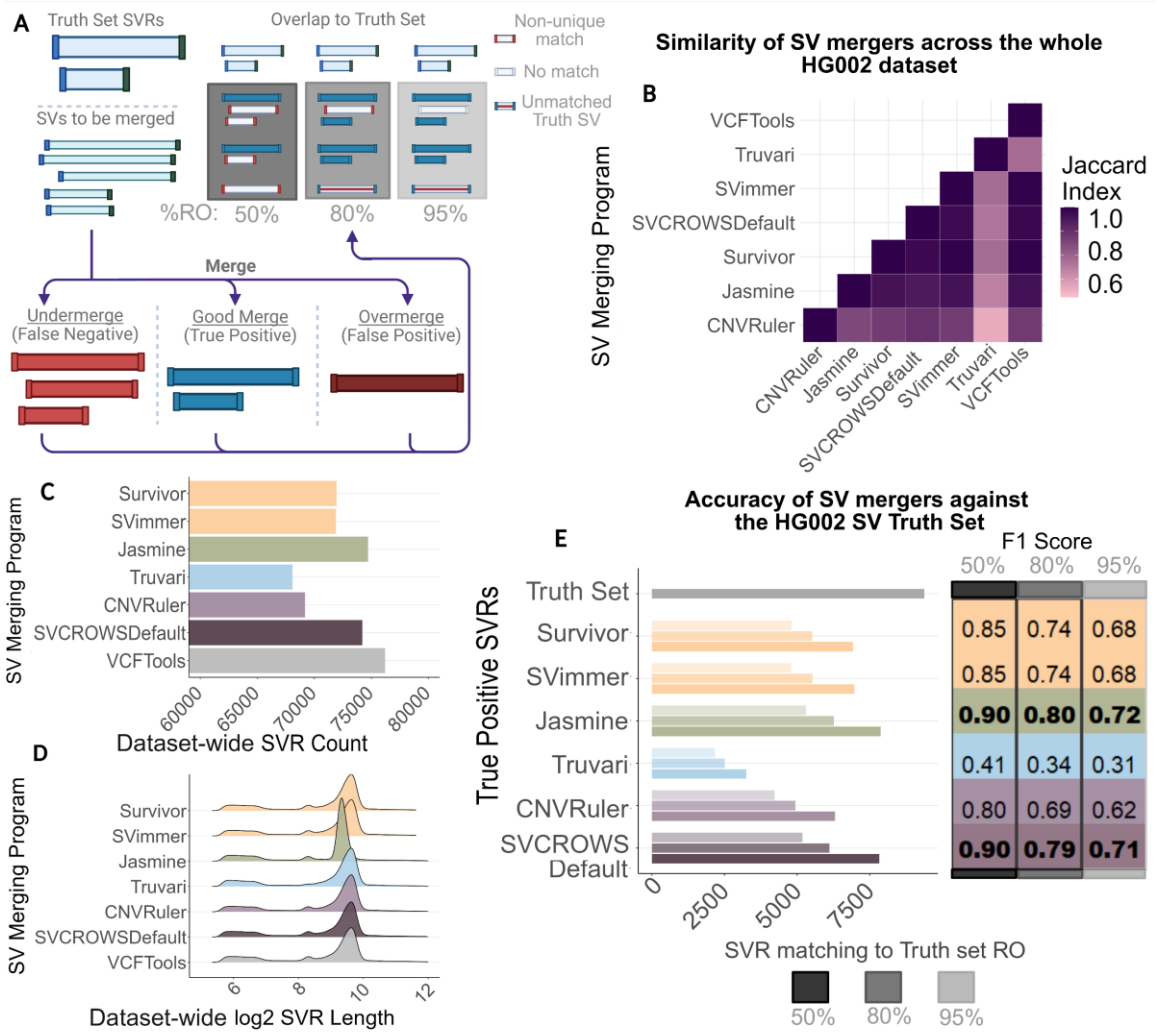



**Figure 2. Comparison of SVRs from 6 different SV mergers to the HG002 SV Truthset**. An SV aggregate list from the GIAB consortium's HG002 reference genome. The aggregate list was merged using 6 different SV merging algorithm types (yellow; distance based (Survivor, SVimmer), green; minimum spanning forest based (Jasmine), blue; multifactored (Truvari), light purpl; static RO (CNVRuler), durk purple; size weighted RO (SVCROWS), gray pure redundancy (VCFTools)) and then assessed for a variety of metrics. **A)** Schematic for assessing quality of SV mergers. SVs from 5 SV callers were combined, merged and compared to each other, and then assessed for incidences of false positive and false negative merging. The resulting SVRs were then overlapped to a subset of the HG002 SV Truth Set (on-truths) at 3 stringencies, which determined the rate of true positive calling. **B)** Comparison of SV merger output using Jaccard index of region overlap (**Table S4**). **C)** Comparison of total SVR number after merging. **D**) Distribution of SVR sizes after merging. **E**) Accuracy of merger validation against the HG002 SV Truth Set. True positive SVRs uniquely overlapped to a single Truth Set SV. False negative SVRs exhibit non-unique matching or do not match to any Truth Set SV. False positives are those that had no overlaps (**Table S5**).

## SVCROWS allows control of SV merging in heterogeneous datasets

While SVCROWS performs well in the context of accurately defining SVRs compared to a small, high-quality truth set (**Fig. 2**), we also wanted to assess its capabilities on a larger and more heterogeneous dataset (the SC dataset). Using real-world data, we sought to evaluate how specific factors drive differences in the final SVR output.

We first assessed the three SVCROWS user input parameters using this dataset: size categories (blue lines, **Figs. 3A-C**), breakpoint (BP) overlapping region sizes (green lines, **Figs. 3A-C**), and RO threshold (yellow lines, **Figs. 3A-C**). As expected, RO had the strongest effect on the number of final SVRs (**Fig. 3A**). The median size of the resulting SVRs was largely not affected by parameter stringency (**Fig. 3B**). However, low stringency in breakpoint matching lead to an increase in size, which suggests that this parameter is useful for preferentially merging relatively smaller SVs.

Further, we assessed two optional features that directly impact SVR construction: breakpoint matching (BP, *optional feature ii*), and expanding the RO boundary (ExRO, *optional feature v*). We noticed that higher stringency parameters were more affected by these SVCROWS features (**Fig. 3C**). While not as impactful as input stringency, BP matching helps to offset false negative SV merging decisions at high stringencies or when used in conjunction with ExRO.

Next, we assessed the SC dataset using three stringencies with consistent secondary options (BP and ExRO enabled). We then adjusted for RO threshold stringency (see ***Supplemental Methods***). As expected, only the final number of SVRs was impacted by stringency, while the distribution of sizes and SV type remained relatively unchanged (**Fig. 3D-F**).

To better understand the direct impacts of SVCROWS' stringency on SVR discovery, we visualized and quantified a complex centromeric region of the genome on chromosome 6 using the SC dataset (**Fig. 3G-K**). There were 143 raw calls originally across this region, with only 43% harboring breakpoints that exactly match with another SV (i.e. pure redundancy). However, only using pure redundancy removal when using an algorithm like VCFTools likely contributes to undermerging, making it difficult to draw significance from most downstream analyses (**Fig. 3G**). The three stringencies of SVCROWS provide different resolutions of the complex loss-SV pattern within the region (**Fig. 3H, I**). Importantly, SVCROWS maintained the major genotypes (the large gain-SV and boxed loss-SV), preserving the same breakpoints called in the input dataset (**Fig. 3J**). The algorithm also conserved unique SVs (only appearing in a single sample across the dataset) indicating avoidance of overmerging (**Fig. 3K**). Ultimately, with the appropriate parameter set, SVCROWS reduced the complexity in this region while maintaining some level of its original heterogeneity.

SVCROWS uses size-weighted RO, which may assist in interpreting regions prone to variation like cis-regulatory elements (CREs) or super cis-regulatory elements (SCREs). They can rapidly alter expression patterns in ways that drive developmental changes, phenotypic diversity, and adaptive evolution[47]. In order to assess SVCROWS performance in these regions, we examined a SCRE on chr16 in the SC dataset (as called by the ENCODE database[48,49], **Fig. S5**). Upon initial inspection, it was evident that several distinct SVs were present in regions known for both upregulating and downregulating 4 genes immediately downstream[49]. All 4 of these genes are implicated in ovarian and breast cancer (both of which this patient had); some genes are more highly transcribed in cancers, while others are less transcribed in cancers (**Fig. S5**)[50–53]. By preserving the variation across this region, this region showcases how SVCROWS can be tuned to limit merging across regions that contribute to important phenotypes.

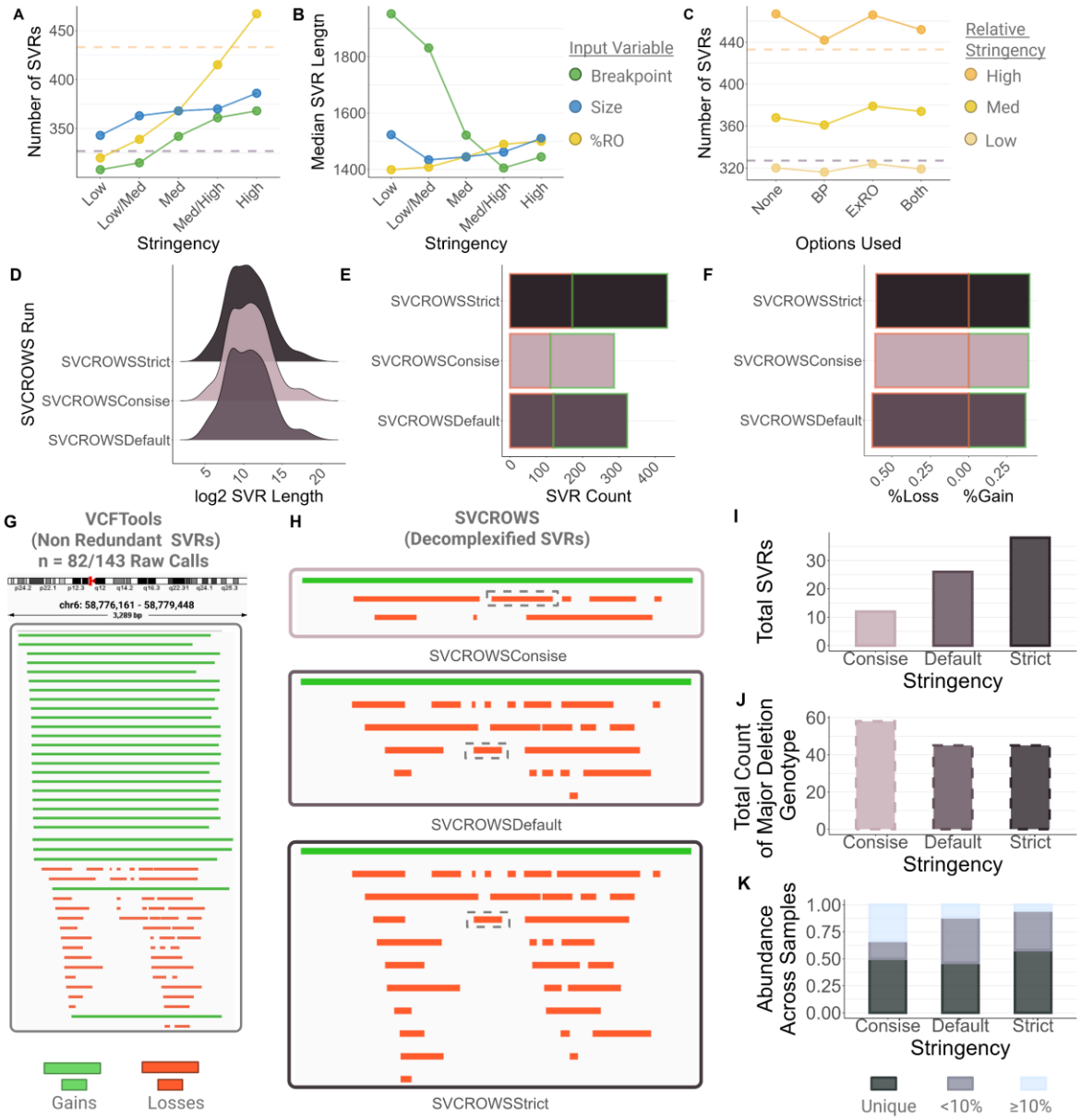



**Figure 3. Options to control stringency and SVR construction in SVCROWS allows for optimized merging.** A complex, single cell (SC) dataset was used to evaluate SVR generation by SVCROWS. **A-C)** Comparison of SVCROWS output to 2 common SV mergers: Svimmer – yellow dashed line, and CNVRuler – purple dashed line. Run inputs are provided in **Table S3. A)** Effect of input factors on SVR totals over different SVCROWS stringencies (see *Supplemental Methods* and Table S3 for more details). **B)** Effect of input factors on median resulting SVR length. **C)** Effect of breakpoint factor (BP) and expanding RO regions (ExRO) on SVR totals under 3 levels of RO threshold stringency. **D-F)** SVCROWS applied to the dataset over optimized stringencies (strict, concise, or default, seeee *Supplemental Methods* for more details): **D)** Distribution of SVR sizes in SC dataset in log2 to better visualize differences, **E)** Total SVRs in SC Dataset. **F)** Distribution of losses and gains in the SC dataset, where losses and gains are outlined by red and green boxes respectively. **G, H)** Integrated Genome Viewer visualization of resulting SVRs in a representative 3.2kb centromeric region of chromosome 6 from: **G)** VCFTools, which removes purely redundant SVRs; **H)** SVCROWS over three different stringencies. Dashed boxes indicate the most abundant loss genotype detected in the dataset. **I-K)** Quantification of panel H: **I)** total number of unique SVRs in the chr6 region, **J)** total number of SVs matched to the SVR representing the abundant deletion present in ~14% of the population, **K)** abundance of each of resulting SVRs across SC dataset samples.

**SVCROWS shows advantages for SV discovery compared to other SV mergers using a heterogeneous dataset**

Using the same SV merging tools used to evaluate the HG002 SV Truth Set, we sought to appreciate differences in SV merger algorithm behavior on the multi-sample SC dataset (**Fig. 4A-C**). This dataset has not been directly quantified for SVs, and represents an opportunity to compare SV merger capabilities for SV discovery. Overall, basic characteristics of generated SVRs were more algorithm-dependent for the SC dataset than the HG002 SV Truth Set. Specifically, we detected a wider range of resulting SVR sizes (**Fig. 4A** compared to **Fig. 2D**) and SVR numbers (**Fig. 4B** compared to **Fig. 2C**). Additionally, algorithms that exhibited high levels of merging tended to report more losses across the higher complexity SC dataset (**Fig. 4B and 4C**). These differences reflect that as the number of samples and heterogeneity rises in a study, differences in SV merging strategy become more pronounced.

Visualizing the differences between the SV mergers in a genomic region with a high rate of unique SVs revealed information about each algorithm's biases regarding SV discovery (**Fig. 4D and 4E**, **Fig. S3**). Firstly, Jasmine undermerged both small and large SVs across this region (**Fig. 4D**, green). In contrast to its good performance on a high quality dataset (**Fig. 1**), Jasmine failed to reduce complexity of this region compared to other programs, perhaps due to the large range of SV sizes. On the other hand, distance-based algorithms are well-balanced with regard to undermerging (**Fig. 4D**, yellow), but do have clear indications of over-merging of smaller SVs (e.g., SVimmer has no small gain alleles towards the 3' end). Truvari likely undermerged smaller SVs, leaving overlapping SVs with as little difference as 90bp (9.5% of the total size) unmerged. Further, by comparing SVCROWS to CNVRuler, we noticed that even the slight change in algorithm type can lead to large changes in the outcome of large regions (**Fig. S4**).

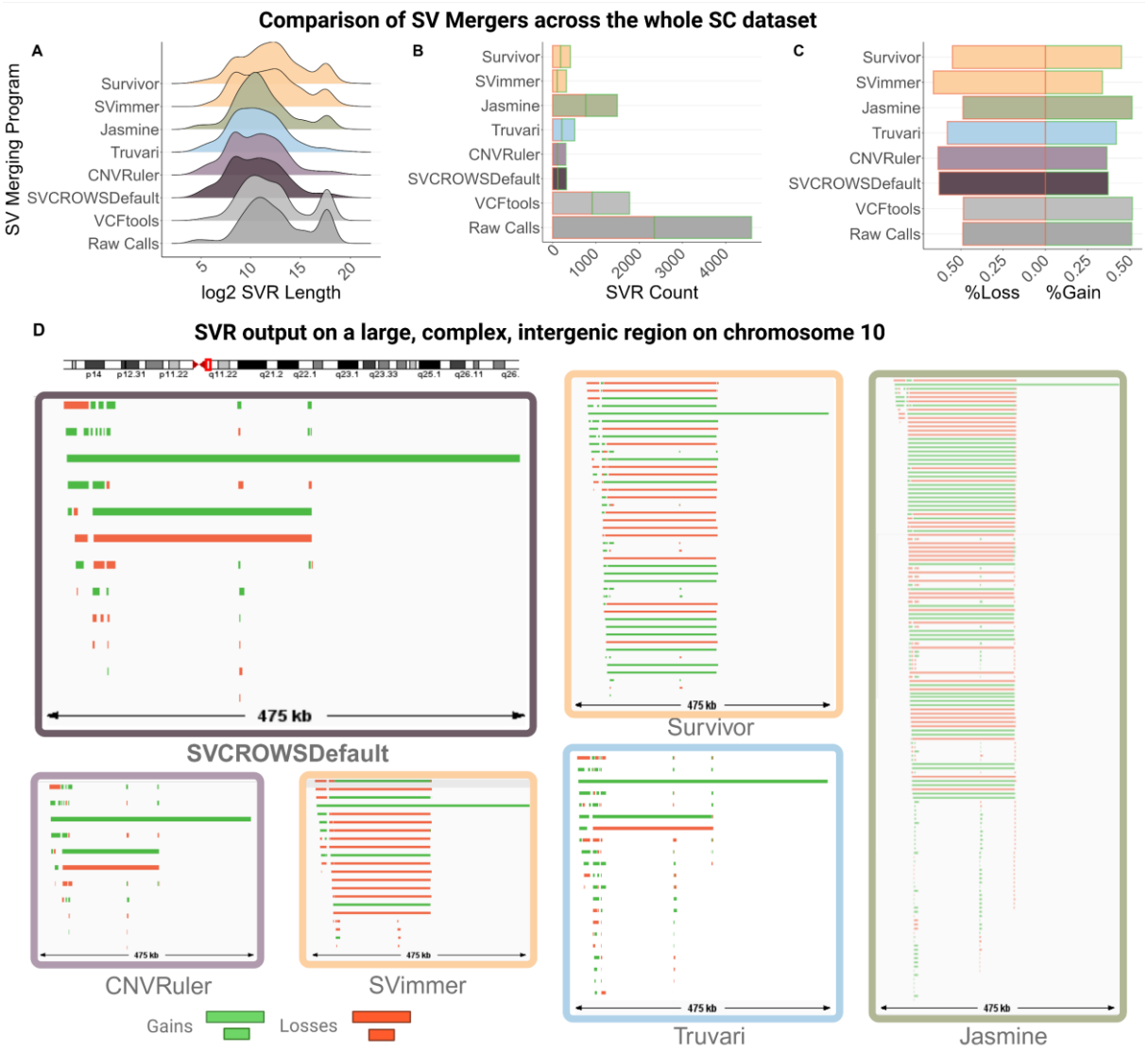



**Figure 4. Comparison of SVCROWs to 5 SV merging programs using a heterogeneous SC dataset.** SV merging performance of different algorithms using a single cell (SC) dataset. **A)** Size distribution of resulting SVRs. **B)** Final SVR count of each tool, compared to the unmerged raw lumpy calls. **C)** Distribution of gains and losses in final SVRs. **D)** Integrated Genome Viewer visualization of resulting SVRs in the chr10 region for each program.

## Discussion

From an analysis perspective, SV breakpoints can vary across samples due to inherent inconsistency in large data sets[54] or by using different SV-calling applications on the same data set[55]. This makes it challenging to identify whether multiple SVs within a genomic neighborhood are distinct. Resolving SV location relies on accurate sequencing, precise mapping, and suitable coverage across a genomic region of interest. Such standards are hard to achieve, especially across repetitive regions like telomeric, centromeric, or rDNA arrays[56,57] or when using low-input methods like single-cell sequencing[58,59]. Therefore, researchers need flexible tools that provide a robust framework for SV comparison.

While some SV mergers are a part of more comprehensive pipelines, their merging function cannot be used independent of the rest of the pipeline, and/or the merging function has only been validated with defined upstream components (**Fig. 4E,F**) [15–17]. SVCROWS, on the other hand, is a standalone merger that is accessible and applicable for both SV validation and discovery. To our knowledge, it is the only R-based SV merger, which requires little prior computational knowledge for use. Further, the text-based input format of SVCROWS is flexible enough to incorporate all data types and formats, including VCF files. Importantly, the algorithm offers a variety of options that incorporates standard functions found in other programs. Finally, its construction around RO as the main means of determining SV matching (i.e. no assignment of 'scores' or 'weights') simplifies how the program will function under different scenarios.

By comparing SVCROWS with several modern and widely used SV merging algorithms, we showed that SVCROWS is as accurate as existing tools for validation (**Fig. 2**) while also being capable of maintaining diversity during SV discovery in a multi-sample dataset (**Fig. 4**). Additionally, SVCROWS can be uniquely tailored at different levels of stringency (**Fig. 3**). Ultimately, it is the user's choice as to whether it is more important to remove or maintain variation in the final SVR data set. Because of this choice and its potential impact, no merger can be said to be objectively 'better' than another. Instead, we emphasize the importance of options and balanced outcomes, which are core principles of SVCROWS, when choosing an SV merger.

Overall, we demonstrated here that there are disparities between SV merging algorithms. The choice of which to use is up to the researcher, but we present three considerations based on direct comparisons of SV merging performance on both simple and complex datasets (**Fig. 5**). The first consideration is that the output of the program must reduce the complexity of the region (**Fig. 5**, 'Output'), especially overcoming size disparities. In this analysis, all mergers were able to reduce SV redundancy; however, most failed to strike a balance in under- or overmerging especially when a range of SV sizes were found in the same genomic neighborhood (e.g. Jasmine, Truvari, and Survivor, **Fig. 2E** and **4D**). We showed that SVCROWS achieves this balance by simplifying SVs across the region while maintaining the diversity of the original dataset (**Figs. 3H** and **3K**). A second consideration is that the architecture of the algorithm needs to be compatible with the study design (**Fig. 5**, 'Technical'). The user should recognize that not all mergers work on all platforms, nor are they compatible with all upstream steps. As mentioned above, SVCROWS can accommodate most upstream steps and input formats. A third consideration is that the algorithm type determines how the input SVs contribute to the resulting SVR (**Fig. 5**, 'Algorithm Type'). For example, Jasmine includes RO as an optional matching parameter, but prioritizes size and start location for merging decisions. Conversely, SVCROWS prioritizes RO and better reflects the relationship between SV size and potential functionality of those SV.

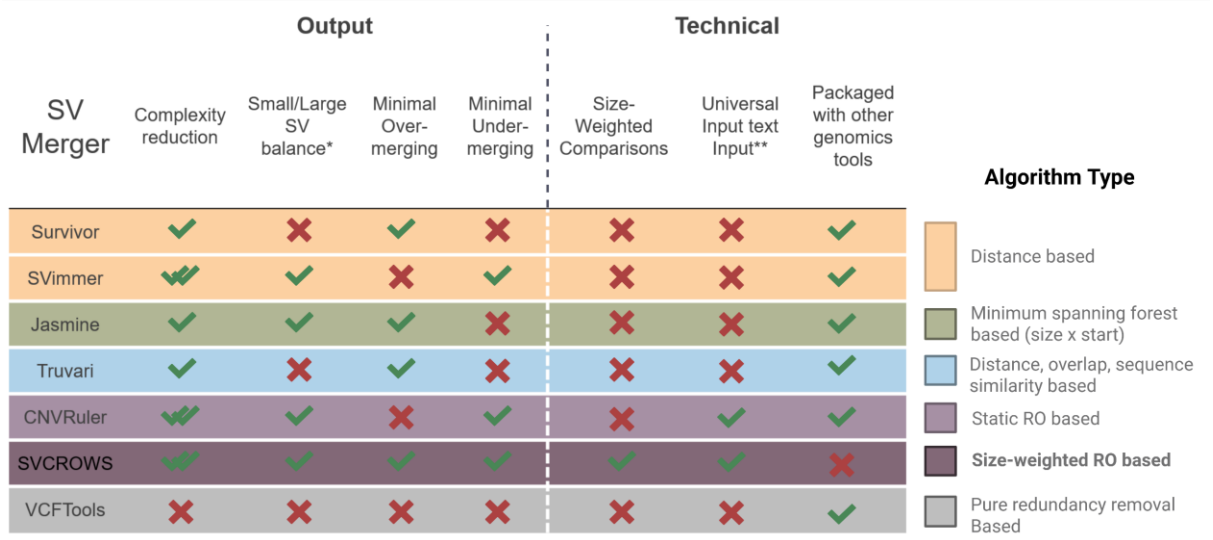



**Figure 5. Comparison of tools based on proposed considerations to make when choosing an SV merging algorithm.** SV merging evaluation of different features of each program (left); Key for algorithm each type of program uses (right).

We recognize that there are some limitations to SVCROWS implementation. While the SVCROWS' input format is flexible, manual conversion from some formats may be necessary and add extra steps to a pipeline. While running SVCROWS in R is convenient, this may lengthen the computational time compared to other programs, especially when running on Windows platforms. Further, the SVCROWS algorithm does not include sequence information during merging decisions Other tools like Truvari or PanPop[16] can evaluate sequence identity at single-basepair resolution in certain SVs (e.g. >50bp insertions and deletions), which may be more beneficial. Finally, manual inspection of SVRs showed a few instances of poor merging decisions from SVCROWS outputs; however, this was rare and also observed across all tools, especially in complex regions (**Figs. S2** and **S4**). This observation emphasizes the continued challenges of SVR construction prior to analysis of important datasets.

We propose that the size-weighted framework of SV comparisons can be applied in all SV-handling algorithms, which may better reflect changes in genotype that produces a tangible outcome in phenotype. While SV merging tools have been used to great effect in the past, it is worth continuing to implement improvements to the field. This is especially true given the growing number of large, multi-sample datasets that are harder to computationally refine accurately. Conversely, having tools that are capable of discovering uncommon variants are critical to investigating diseases caused by rare mutations. The balance of SV validation and discovery SVCROWS can be tailored to provide reflects these patterns in the field.

**Conflicts of Interest**
The authors report no conflicts of interest.

**Funding**
We acknowledge funding from NIH (**R01AI150856, to JLG**) and NSF-NRT award (**2021791**, to NJB).

**Data Availability**
SVCROWS can be downloaded as a Package or R-Markdown file at:
https://github.com/A-Crow-Nowhere/SVCROWS.git

Links to the GIAB project and EGA data files can be found in the *Supplementary Methods.*

**Contact**: jlg5fw@virginia.edu; njb8sg@virginia.edu

**Author Contributions Statement**

Noah Brown: Conceptualization, Data curation, Investigation, Formal analysis, Software, Validation, Visualization, Writing-original draft, writing-review and editing

Charles Danis: Data curation, Formal analysis, Software

Vazira Ahmedjanova: Software

Jennifer Guler: Conceptualization, Funding acquisition, Methodology, Project admiration, Resources, Writing-original draft, writing-review and editing

## Bibliography

1. Flint, J. *et al.* High frequencies of alpha-thalassaemia are the result of natural selection by malaria. *Nature* **321**, 744–750 (1986).
2. Gonzalez, E. *et al.* The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. *Science* **307**, 1434–1440 (2005).

3. Lukácová, M., Barák, I. & Kazár, J. Role of structural variations of polysaccharide antigens in the pathogenicity of Gram-negative bacteria. *Clin. Microbiol. Infect. Off. Publ. Eur. Soc. Clin. Microbiol. Infect. Dis.* **14**, 200–206 (2008).
4. Marshall, C. R. *et al.* Structural Variation of Chromosomes in Autism Spectrum Disorder. *Am. J. Hum. Genet.* **82**, 477 (2008).
5. Songsomboon, K. *et al.* Genomic patterns of structural variation among diverse genotypes of Sorghum bicolor and a potential role for deletions in local adaptation. *G3 GenesGenomesGenetics* **11**, jkab154 (2021).
6. Carvalho, C. M. B. & Lupski, J. R. Mechanisms underlying structural variant formation in genomic disorders. *Nat. Rev. Genet.* **17**, 224–238 (2016).
7. Dwarshuis, N. *et al.* The GIAB genomic stratifications resource for human reference genomes. *Nat. Commun.* **15**, 9029 (2024).
8. Auton, A. *et al.* A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).
9. Tan, K.-T., Slevin, M. K., Meyerson, M. & Li, H. Identifying and correcting repeat-calling errors in nanopore sequencing of telomeres. *Genome Biol.* **23**, 180 (2022).
10. Sedlazeck, F. J. *et al.* Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods* **15**, 461–468 (2018).
11. Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32**, 246–251 (2014).
12. Alonge, M. *et al.* Major Impacts of Widespread Structural Variation on Gene Expression and Crop Improvement in Tomato. *Cell* **182**, 145-161.e23 (2020).

13. Heller, D. & Vingron, M. SVIM-asm: structural variant detection from haploid and diploid genome assemblies. *Bioinformatics* **36**, 5519–5521 (2021).

14. Zheng, Z. *et al.* A sequence-aware merger of genomic structural variations at population scale. *Nat. Commun.* **15**, 960 (2024).

15. Eggertsson, H. P. *et al.* GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. *Nat. Commun.* **10**, 5402 (2019).

16. Zheng, Z. *et al.* A sequence-aware merger of genomic structural variations at population scale. *Nat. Commun.* **15**, 960 (2024).

17. Kirsche, M. *et al.* Jasmine and Iris: Population-scale structural variant comparison and analysis. *Nat. Methods* **20**, 408–417 (2023).

18. English, A. C., Menon, V. K., Gibbs, R. A., Metcalf, G. A. & Sedlazeck, F. J. Truvari: refined structural variant comparison preserves allelic diversity. *Genome Biol.* **23**, 271 (2022).

19. CNVRuler: a copy number variation-based case–control association analysis tool | Bioinformatics | Oxford Academic. https://academic.oup.com/bioinformatics/article/28/13/1790/234696.

20. Olson, N. D. *et al.* Variant calling and benchmarking in an era of complete human genome sequences. *Nat. Rev. Genet.* **24**, 464–483 (2023).

21. Handsaker, R. E. *et al.* Large multiallelic copy number variations in humans. *Nat. Genet.* **47**, 296–303 (2015).

22. Handsaker, R. E., Korn, J. M., Nemesh, J. & McCarroll, S. A. Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nat. Genet.* **43**, 269–276 (2011).

23. Venkatapoorna, C. M. K. *et al.* Association of Salivary Amylase (AMY1) Gene Copy Number with Obesity in Alabama Elementary School Children. *Nutrients* **11**, 1379 (2019).

24. Rhie, A. *et al.* The complete sequence of a human Y chromosome. *Nature* **621**, 344–354 (2023).

25. Rautiainen, M. *et al.* Telomere-to-telomere assembly of diploid chromosomes with Verkko. *Nat. Biotechnol.* **41**, 1474–1482 (2023).

26. Wang, T. *et al.* The Human Pangenome Project: a global resource to map genomic diversity. *Nature* **604**, 437–446 (2022).

27. Nurk, S. *et al.* The complete sequence of a human genome. *Science* **376**, 44–53 (2022).

28. Zook, J. M. *et al.* A robust benchmark for detection of germline large deletions and insertions. *Nat. Biotechnol.* **38**, 1347–1355 (2020).

29. Layer, R. M., Chiang, C., Quinlan, A. R. & Hall, I. M. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.* **15**, R84 (2014).

30. PacificBiosciences/pbsv. PacBio (2025).

31. Smolka, M. *et al.* Detection of mosaic and population-level structural variants with Sniffles2. *Nat. Biotechnol.* **42**, 1571–1580 (2024).

32. Alkan, C. *et al.* Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.* **41**, 1061–1067 (2009).

33. Mohiyuddin, M. *et al.* MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. *Bioinformatics* **31**, 2741–2744 (2015).

34. Freeberg, M. A. *et al.* The European Genome-phenome Archive in 2021. *Nucleic Acids Res.* **50**, D980–D987 (2021).

35. Duan, D.-M. *et al.* Comparisons of performances of structural variants detection algorithms in solitary or combination strategy. *PLOS ONE* **20**, e0314982 (2025).

36. Funnell, T. *et al.* Single-cell genomic variation induced by mutational processes in cancer. *Nature* **612**, 106–115 (2022).

37. Mahmoud, M. *et al.* Structural variant calling: the long and the short of it. *Genome Biol.* **20**, 246 (2019).

38. Duan, D.-M. *et al.* Comparisons of performances of structural variants detection algorithms in solitary or combination strategy. *PLOS ONE* **20**, e0314982 (2025).

39. T, F. *et al.* Single-cell genomic variation induced by mutational processes in cancer. - Abstract - Europe PMC.

40. Jeffares, D. C. *et al.* Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat. Commun.* **8**, 14061 (2017).

41. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *GigaScience* **10**, giab008 (2021).

42. Human molecular genetics - NLM Catalog - NCBI. https://www.ncbi.nlm.nih.gov/nlmcatalog/101523906.

43. Shabalina, S. A., Ogurtsov, A. Y., Kondrashov, V. A. & Kondrashov, A. S. Selective constraint in intergenic regions of human and mouse genomes. *Trends Genet.* **17**, 373–376 (2001).

44. Kołomański, M., Szyda, J., Frąszczak, M. & Mielczarek, M. DNA sequence features underlying large-scale duplications and deletions in human. *J. Appl. Genet.* **63**, 527–533 (2022).

45. Rajic, Z. A. *et al.* Size of the protein-coding genome and rate of molecular evolution. *J. Hum. Genet.* **50**, 217–229 (2005).

46. Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27**, 2156–2158 (2011).

47. Wittkopp, P. J. & Kalay, G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nat. Rev. Genet.* **13**, 59–69 (2012).

48. Huang, H. *et al.* Defining super-enhancer landscape in triple-negative breast cancer by multiomic profiling. *Nat. Commun.* **12**, 2242 (2021).

49. Leistico, J. R. *et al.* Epigenomic tensor predicts disease subtypes and reveals constrained tumor evolution. *Cell Rep.* **34**, 108927 (2021).

50. Kozole, S. L. & Beningo, K. A. Myosin Light Chains in the Progression of Cancer. *Cells* **13**, 2081 (2024).

51. Liu, Y. *et al.* Depletion of VPS35 attenuates metastasis of hepatocellular carcinoma by restraining the Wnt/PCP signaling pathway. *Genes Dis.* **8**, 232–240 (2020).

52. Qi, G. *et al.* SHCBP1 promotes cisplatin resistance of ovarian cancer through AKT/mTOR/Autophagy pathway. *Apoptosis Int. J. Program. Cell Death* **30**, 83–98 (2025).

53. Lin, Y. *et al.* ORC6, a novel prognostic biomarker, correlates with T regulatory cell infiltration in prostate adenocarcinoma: a pan-cancer analysis. *BMC Cancer* **23**, 285 (2023).

54. Conrad, D. F. & Hurles, M. E. The population genetics of structural variation. *Nat. Genet.* **39**, S30–S36 (2007).

55. Joe, S. *et al.* Comparison of structural variant callers for massive whole-genome sequence data. *BMC Genomics* **25**, 318 (2024).

56. Levy-Sakin, M. *et al.* Genome maps across 26 human populations reveal population-specific patterns of structural variation. *Nat. Commun.* **10**, 1025 (2019).

57. Rothschild, D. *et al.* Diversity of ribosomes at the level of rRNA variation associated with human health and disease. *bioRxiv* 2023.01.30.526360 (2024) doi:10.1101/2023.01.30.526360.

58. Mallory, X. F., Edrisi, M., Navin, N. & Nakhleh, L. Methods for copy number aberration detection from single-cell DNA-sequencing data. *Genome Biol.* **21**, 1–22 (2020).

59. Ning, L. *et al.* Current Challenges in the Bioinformatics of Single Cell Genomics. *Front. Oncol.* **4**, 7 (2014).