SurVIndel2: improving CNVs calling from next-generation sequencing using novel hidden information

Ramesh Rajaby[1,2] and Wing-Kin Sung[1,2,3,4,5,*]

[1]Hong Kong Genome Institute, Hong Kong Science Park, Shatin, Hong Kong, China
[2]A*STAR Genome Institute of Singapore, 60 Biopolis Street, Singapore 138672, Singapore
[3]Department of Chemical Pathology, The Chinese University of Hong Kong, Hong Kong, China
[4]Laboratory of Computational Genomics, Li Ka Shing Institute of Health Sciences, The Chinese University of Hong Kong, Hong Kong, China
[5]School of Computing, National University of Singapore, 13 Computing Drive, Singapore 117417, Singapore
[*]Corresponding author: Wing-Kin Sung, kwksung@cuhk.edu.hk

Abstract

Deletions and tandem duplications (commonly called CNVs) represent the majority of structural variations. They can be identified using short reads, but because they frequently occur in repetitive regions, existing methods fail to detect most of them. This is because CNVs in repetitive regions often do not produce the evidence needed by existing short reads-based callers (split reads, discordant pairs or read depth change). Here, we introduce a new CNV short reads-based caller named SurVIndel2. SurVindel2 builds on statistical techniques we previously developed, but also employs a novel type of evidence, hidden split reads, that can uncover many CNVs missed by existing algorithms. We use public benchmarks to show that SurVIndel2 outperforms other popular callers, both on human and non-human datasets. Then, we demonstrate the practical utility of the method by generating a catalogue of CNVs for the 1000g project that contains hundreds of thousands of CNVs missing from the most recent public catalogue. We also show that SurVIndel2 is able to complement small indels predicted by Google DeepVariant, and the two software used in tandem produce a remarkably complete catalogue of variants in an individual. Finally, we characterise how the limitations of current sequencing technologies contribute significantly to the missing CNVs.

1 Introduction

Structural Variations (SVs) are mutations in a genome that involve 50 nucleotides or more. Although they are less frequent than point mutations, SVs account for more heritable differences in populations due to their large scale and play a major role in many genetic diseases [31]. Deletions and tandem duplications are two important classes of SVs. Deletions occur when a contiguous sequence of bases is lost, while tandem duplications occur when a source sequence of DNA is duplicated in-place; they are called unbalanced SVs because they alter the copy number of the genome. A recent study of the SVs in 14,891 human genomes [6] identified deletions as the most frequent SV and tandem duplications as the third most common, after insertions. However, the SVs were detected using short reads, and existing SV callers miss most duplications, as demonstrated in the following sections. When detecting SVs using high-quality long reads, tandem duplications are as common as deletions (Section 2.3). Aside from being an important source of polymorphism, deletions and tandem duplications are implicated in many genetic diseases, such as Smith-Magenis syndrome, Potocki-Lupski syndrome and Williams-Beuren syndrome [31]. Deletions and tandem duplications are often described as *copy number variants* (CNV). In this manuscript, we will refer to them as such.

Nowadays, CNVs can be accurately detected using long read sequencing technologies, like PacBio HiFi reads and ONT nanopore reads. However, these technologies are still expensive and this limits their use in large-scale population studies. A less costly solution is to use short-reads sequencing technologies such as Illumina, which is often used for sequencing large populations [6, 1, 3]. Short read datasets are rapidly increasing in number, with studies on tens of thousands of genomes already published. As the Hong Kong Genome Project, Singapore SG100K, the European '1+ Million Genomes' Initiative, and the All of Us Research Project (among others) continue to generate large amounts of sequencing data, short-read whole genome sequencing will remain at the forefront of genomics for the foreseeable future.

However, detecting deletions and especially tandem duplications from short reads is very challenging. Despite a large number of callers published in the last decade, the sensitivity of even the best methods is still low. Manta, a widely used SV caller employed in many large-scale population studies [6, 3], as well as the best performing tool according to recent benchmarking efforts [22, 4], detects around 40% of the existing deletions and less than 10% of the existing tandem duplications when tested on a recent benchmark dataset (see Section 2.3).

In a previous study [18], we found that most of the CNVs missed by existing callers were located in tandem repetitive regions. This is because CNVs in tandem repetitive regions often do not generate split reads; this result was confirmed in this study (Section 2.1). Furthermore, shorter CNVs generate few or no discordant pairs, and read depth methods are known to only work for large CNVs [27]. Thus, naively counting the number of discordant pairs and split reads will miss many CNVs in repetitive regions. Finally, we showed that the problem can be mitigated by considering the insert sizes of all the read pairs containing the putative CNVs, and performing statistical tests to determine whether they deviate from expectation. Using this technique we developed SurVIndel [18], a CNV caller that improved the detection rate in repetitive regions over existing methods.

However, SurVIndel suffered from two problems. First, it generated many false positives. This is a serious issue when generating catalogues of CNVs for large populations. Second, because split reads are often unavailable in repetitive regions, many of the CNVs detected had imprecise boundaries. Subsequent steps in population studies, such as clustering of the CNVs across samples, regenotyping and downstream analysis, all rely and greatly benefit from precise breakpoints.

To address these issues, we have developed a new caller called SurVIndel2, which significantly reduces the number of false positives while improving the sensitivity of the original SurVIndel. Additionally, SurVIndel2 generates precise breakpoints for most of the called CNVs. This new caller utilizes split reads, discordant pairs, and a novel type of evidence called hidden split reads to detect candidate CNVs. Hidden split reads allow us to identify the existence and the precise breakpoints of CNVs in repetitive regions, which leads to a more accurate and comprehensive set of candidate CNVs. SurVIndel2 then applies the statistical approach of SurVIndel, along with depth-based filters, to generate a set of confident CNVs.

This paper is organised as follows. In Section 2.1, we introduce the notion of hidden split reads, and we explain why they can improve CNV detection. Section 2.2 gives an overview of the algorithm of SurVIndel2. Sections 2.3 and 2.4 use publicly available datasets, both human and non-human, to compare the performance of SurVIndel2 to the best methods in the literature, according to a recent benchmark [4]. Then, in Section 2.5, we use SurVIndel2 to generate a dataset of deletions and duplications in the 1000 Genomes Project. We show that our dataset is significantly more complete than that produced by a state of the art pipeline: it contains more than twice as many deletions and five times as many duplications, with high validation rates according to long reads. In Section 2.6 we illustrate that SurVIndel2 is also able to effectively call CNVs that are smaller than 50 bp, and it can be used to enhance callsets generated by state-of-the-art small variants callers such as DeepVariant [23]. Finally, in Section 2.7 we investigate why many CNVs are hard to detect, and we reach the surprising conclusion that it is not only due to the repetitive nature of the genome, but mostly due to the sequencing platforms employed being unable to correctly sequence many regions of the genome.

SurVIndel2 is fully open source and available at `https://github.com/kensung-lab/SurVIndel2`. It is easy to run and only requires a BAM or a CRAM file and a reference genome, and it outputs a fully standard VCF file.

2 Results

2.1 Hidden split reads

Let D be a deleted region in a sample and let L and R be its left and right-flanking regions, respectively. L and R will be adjacent in the sample genome, but not in the reference genome. The concatenation of L and R (we refer to it as LR) is called the junction sequence of the deletion. When the junction sequence is a novel sequence, i.e., it is not present in the reference genome, the reads sequenced from it will be "split", meaning that a portion of the read will align to L and the rest will align to R. Split reads are commonly used to detect deletions. Fig. 1a shows an example of

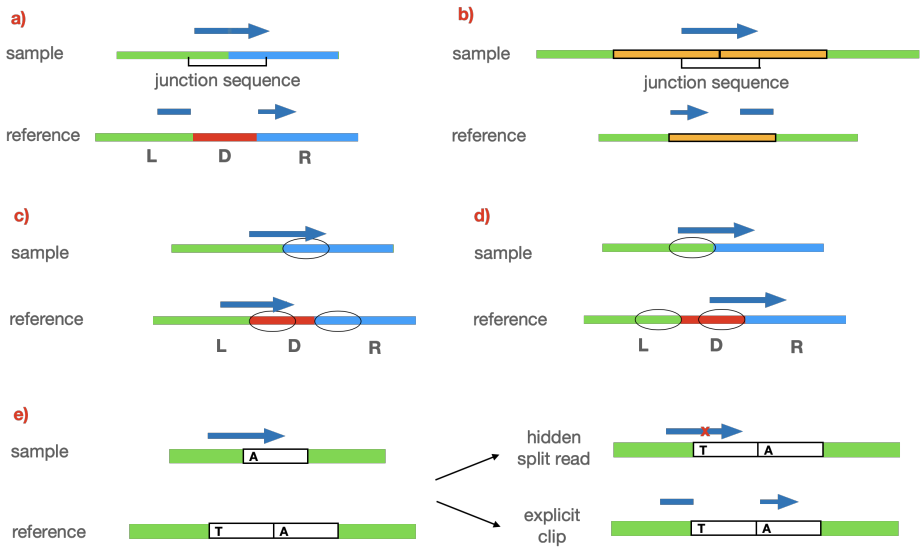

Figure 1: (a) The red segment (D) of the reference is deleted in the sample, causing the green (L) and the blue (R) regions to be adjacent. Reads sequenced partly from the L and partly from R may not fully align to the reference, causing the read to be clipped. (b) Similarly, tandem duplications may also create clipped reads. (c) When the prefixes (circled) of D and R are homologous, many reads sequenced across the breakpoint of the deletion will fully align to the reference, instead of being clipped. If the length of the prefixes is longer than the read length, no clipped reads will be produced. (d) Similarly, when the suffixes of D and L are homologous, little or no clipped reads will be produced. (e) Deletions of full copies in tandem repeats are a prime example of (c), which is why they often do not cause clipped reads. However, in practice, different copies of a tandem repeat are often not identical, and we may get hidden split reads. In this example, a sequence (white) is repeated twice in the reference, but only once in the sample. The two copies of the sequence are identical except for one position (first copy has a T, the second has an A). The single copy in the sample has an A, therefore we want to report that the first copy was deleted. Reads spanning both the green left-flanking region and the A character can be aligned to the reference in two ways: fully with one mismatch, or explicitly clipped. Aligners will normally choose the former. While this is usually a reasonable choice, in this case it will hide the existence of the deletion. We call such reads hidden split reads.

a deletion creating a split read. Similarly, tandem duplications may also produce split reads (Fig. 1b).

Some deletions will not produce split reads. One reason for this is that the junction sequence is not novel. For example, if a sufficiently long prefix of D is homologous to a prefix of R, the deletion will not create split reads (Fig. 1c). Similarly, when a suffix of D is homologous to a suffix of L, no split reads will support the deletion (Fig. 1d). Tandem repeats are common examples of this.

We have a tandem repeat when a sequence (*repeat unit*) appears consecutively multiple times. Deletions and duplications of entire repeat units often do not create split reads [18]: because L, R and D are identical, the junction sequence LR will be identical to LD and to DR, so any read sequenced from LR will fully align to either LD or DR (Supplementary Fig. S4a).

However, in practice, the copies in a tandem repeat may present some differences between each other. Suppose D and R are similar but not identical, and let r be a read sequenced from LR. The algorithm that aligns r to the reference genome has two choices:

1. split r between L and R, obtaining two (partial) perfect alignments; or
2. align r fully to either LD or DR, accepting mismatches or gaps in the alignment.

Which option the aligner chooses depends on many factors, including the mapping algorithm used, its parameters and how similar L, D and R are. The more different they are, the more likely option 1 will be chosen. When option 2 is chosen, we say that R is a *hidden split read* (see Fig. 1e for an example). Similarly, tandem duplications may also produce hidden split reads (Supplementary Fig. S4b).

Next, we provide a formal definition of hidden split read. Given a genomic fragment s, let $score(s)$ be the score of the alignment of s to the reference genome. Given a read $r[1..n]$, we say that r is a hidden split read if there exists $i \in [1..n-1]$ such that $score(r) < score(r[1..i]) + score(r[i+1..n])$; in other words, if r is allowed to split into two without penalty and the two parts can align independently, the score of the alignment will increase. $score(r[1..i]) + score(r[i+1..n]) - score(r)$ is the *HSR-score* of r. See Supplementary Fig. S5 for an example of HSR-score calculation.

In order to confirm that CNVs mostly delete or insert full repeat units and the importance of hidden split reads in detecting such CNVs, we use two publicly available human genomes: HG002 and HG00512. A detailed explanation of the data used and the experiments mentioned below is provided in the Supplementary Methods.

Around 64% of the deletions and 93% of the tandem duplications in HG002 (Fig. 2a) are located in tandem repetitive regions. Fig. 2b demonstrates that indeed most CNVs delete or insert an integer number of repeat units. For this reason, the number of CNVs in tandem repeats supported by split reads is low. Only 25.2% of the deletions and 29.0% of the duplications in tandem repeats were supported by split reads, compared to 90.2% and 83.3%, respectively, for deletions and duplications not in tandem repeats (Fig. 2c, Fig. 2d). However, hidden split reads can help us uncover a significant number of CNVs not supported by split reads. Using both split reads and hidden split reads, the number of deletions in tandem repeats that we are potentially able to discover is more than double (25.2% using split reads, 58.9% using both split reads and hidden

split reads, 2.3×). We obtain similar figures for duplications (29.0% using split reads, 63.7% using both split reads and hidden split reads, 2.2×). Supplementary Fig. S6 shows that the statistics are virtually identical for HG00512.

Although hidden split reads have the potential to identify a substantial portion of repetitive CNVs that are typically missed by traditional split-read methods, their use presents significant challenges. Fig. 2e shows the binned HSR-score distribution for hidden split reads that support CNVs in HG002. Most hidden split reads have low HSR-score. Fig. 2f shows that most CNVs detected by realigning hidden split reads are actually noise. Interestingly, reads with high HSR-score do not produce significantly less false positives, in relative terms. Distinguishing hidden split reads that support existing CNVs is not trivial, and novel and sophisticated techniques need to be developed. In the remainder of this paper, we propose an algorithm that uses hidden split reads to detect deletions and duplications in a sample genome, and significantly outperforms the current state of the art.

2.2 Overview of the method

SurVIndel2 requires (1) a reference genome (in FASTA format) and (2) a BAM or CRAM alignment file containing paired reads sequenced from a single sample aligned to the reference genome. It outputs a VCF file containing a list of predicted deletions and tandem duplications. Fig. 3 shows a high-level overview of the full algorithm. Detailed explanations and examples for individual steps are provided in Supplementary Methods.

As a first step, SurVIndel2 iterates through the alignment file and gathers the evidence that will be later used to determine the candidate CNVs: split reads, hidden split reads and discordant pairs. Split reads and hidden split reads were described in Section 2.1. SurVIndel2 considers potential hidden split reads to be all the reads that are not clipped and present at least three differences with the reference genome (defined as number of mismatches plus number of indels). Discordant read pairs are pairs of reads that map inconsistently to the library preparation parameters. A deletion may produce pairs where the two reads map very far apart (Fig. S15a). Let μ and σ be the mean and the standard deviation of the distribution of insert sizes for the library: we define as discordant any pair that has insert size greater than $\mu + 3\sigma$ (we will refer to this quantity as max_{IS} throughout the paper). On the other hand, a duplication may produce pairs that have the wrong relative orientation, i.e., the forward-facing read is mapped downstream of the reverse-complemented read (Fig. S15b). We define such pairs as discordant.

Given the split reads, hidden split reads and discordant pairs, candidate CNVs are generated by two different modules: the *split module* uses split reads and hidden split reads, while the *discordant module* uses discordant pairs. Then, we calculate a set of features for each putative CNV, such as the number of supporting split and hidden split reads, their mapping quality, the number of discordant and concordant pairs, alignment score of the junction sequence when realigned to the reference, depth of the CNV and flanking regions, and others (Fig 3h). Additionally, we use the distribution of the insert sizes of the read pairs around the CNV to calculate a confidence interval for its size, as well as a p-value estimating the probability that the CNV is a false positive. Such features were first

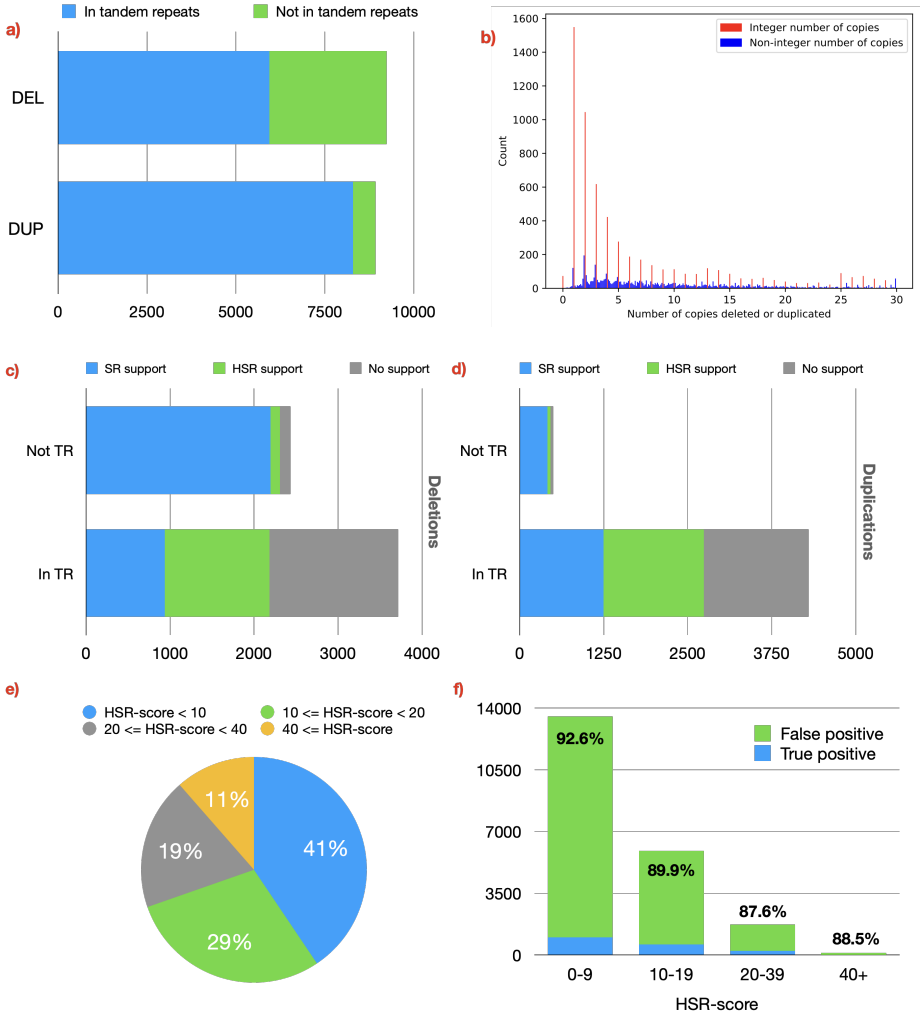

Figure 2: The impact of CNVs deleting or duplicating full repeat units and the importance of hidden split reads in detecting such CNVs, studied on HG002. (a) We consider a CNV to be contained in a tandem repeat if at least 90% of the CNV is covered by the repetitive region. Nearly two thirds of the deletions and more than 93% of the tandem duplications in this dataset are contained in a tandem repeat. (b) For every CNV that overlaps with a tandem repeat, we calculated the number of repeat units that are deleted or inserted, by dividing the length of the CNV by the length of the repeat unit. Bars representing an integer number of copies (i.e., the length of the CNV is a multiple of the length of the repeat unit) are coloured red. Most CNVs add or delete an integer number of repeat units. (c,d) Number of deletions (c) and tandem duplications (d) inside and outside of tandem repeats that are supported by split reads (SR), by hidden split reads (HSR), and by neither. Hidden split reads have the potential to recover a significant number of deletions and duplications missed by regular split reads. (e) HSR-scores of the hidden split reads supporting the existing CNVs in HG002. Most reads have low (<20) HSR-score. (f) We found all hidden split reads in 10,000 randomly sampled repetitive regions, and for each we tried to predict a CNV by finding the optimal split alignment. Most CNVs (>91%) detected this way were false positive. This shows that using hidden split reads to detect CNVs is challenging, and effective filters must be employed to distinguish real from false CNVs.

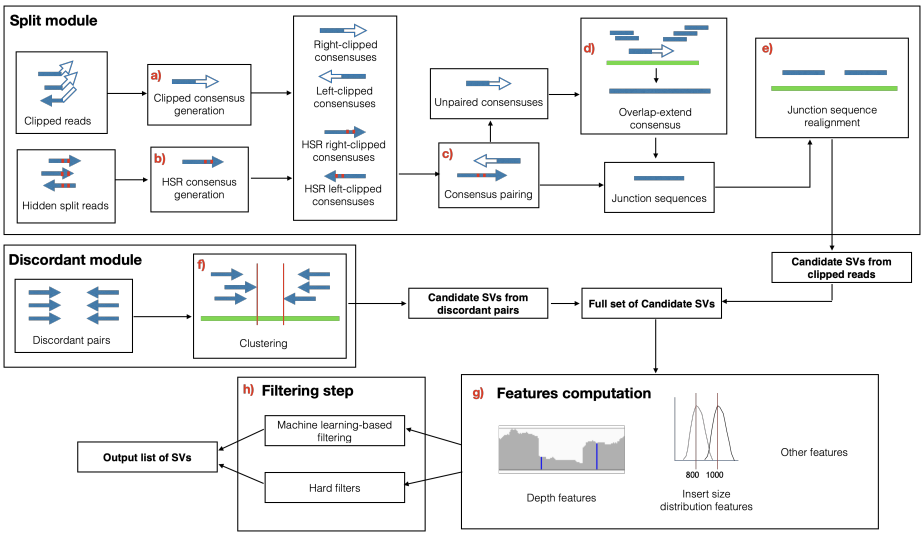

Figure 3: Overview of the SurVIndel2 method. It primarily identifies candidate CNVs by using split reads and hidden split reads. Candidate CNVs that do not generate clipped reads are identified using discordant pairs. Finally, the candidate CNVs are tested by a series of filters, mainly (i) statistically testing the distribution of the read pairs surrounding the CNV and (ii) testing changes in read depth in the CNV region as well as its flanking regions.

introduced by SurVIndel [18]. Finally, all of the features are used in the filtering step (Fig 3g), which aims at removing most false positives, while retaining the correct CNVs. In order to do so, it uses a pre-trained random forest. If a pre-trained forest is not available, a set of fixed hard filters are used. Below, we explain how the split and the discordant modules find candidate CNVs.

The split module first generates consensus sequences by clustering left-clipped reads and hidden split reads into left-clipped consensus sequences and right-clipped reads and hidden split reads into right-clipped sequences (Fig 3a, 3b). The consensus sequences are partitioned into left and right-clipped. Consensus sequences in each group potentially represents either the left or the right breakpoint of a CNV: furthermore, using the mates of the clustered reads, we can calculate a putative range for the location of the second breakpoint (Fig. S1): we call this range *opposite breakpoint range*. We pair the consensuses so that each pair contains a left-clipped and a right-clipped consensus, and the consensuses are within each other's opposite breakpoint range (Fig 3c). For each pair we generate a *junction sequence* by overlapping the left-clipped and right-clipped consensuses. The junction sequence is expected to be the DNA sequence at the precise junction where two different genomic regions have been joined together as a result of the variant (the left and right-flanking regions for deletions, the two copies of the duplicated region for tandem duplications, examples in Fig. 1a and Fig. 1b). For consensuses that are not paired, we attempt generate the junction sequence by extending them using an overlap-layout-consensus approach (Fig 3d). Finally we realign the junction sequence to the reference genome to obtain the precise breakpoints of the CNV (Fig 3e).

The discordant module is considerably simpler, and it clusters discordant pairs in order to detect CNVs (Fig 3f). One drawback is that, unlike split reads, discordant pairs often provide an approximate position for the breakpoints, rather than a precise one. When the same CNV is predicted by both the split module and the discordant module, the coordinates from the split module are retained, since they are deemed precise.

2.3 HGSVC2 benchmark

As mentioned in Section 2.1, the Human Genome Structural Variation Consortium (HGSVC) has published a catalogue (called HGSVC2) of insertions and deletions in 35 human genomes (34 from the 1000 genomes project, plus HG002), identified from long-read PacBio whole-genome sequencing and Strand-seq data [7]. Furthermore, the New York Genome Center (NYGC) has recently performed high-coverage, PCR-free sequencing for the 2,504 original samples in the 1000g genome project, along with 698 additional samples. We downloaded the CRAM files published by NYGC for the 34 samples in the HGSVC2 benchmark (30x coverage, expect for one sample which had 60x coverage). We also obtained a 50x short-read dataset for HG002 by downloading runs from SRR1766442 to SRR1766486, and mapping them to hg38 using BWA-MEM. We use these datasets to benchmark the performance of SurVIndel2, and three highly cited methods that performed well in a recent benchmarking analysis [4]: Delly [19], Lumpy [12] (which we ran using the Smoove package, as recommended by the authors) and Manta [5]. These SV callers were used in some of the major studies on SVs in human populations [6, 1, 3]. We also included the original SurVIndel [18] in the comparison. Note that the callers may report some of the duplications as insertions. For this reason, the sensitivity for each caller was evaluated using both the duplications and the insertions. Precision was only computed on the duplications.

The machine learning model of SurVIndel2 was trained using the 34 samples in HGSVC2 sampled by the NYGC. For this reason, when testing them, we use a leave-one-out strategy. More precisely, for each sample S in HGSVC2 we train a separate model by excluding S from the training set, and we use it to filter the raw calls by SurVIndel2 detected from S.

SurVIndel2 outperforms the other methods in both deletions and duplications, often by a large margin (Fig. 4a and Fig. 4b). For deletions, Manta is the only method that almost matches SurVindel2 precision (average precision is 0.84 for Manta, 0.87 for SurVIndel2), but at the expense of greatly reduced sensitivity (average sensitivity is 0.41 for Manta, 0.60 for SurVIndel2, 46% increase). SurVIndel is the second most sensitive method (0.5 average sensitivity and average precision 0.57), but it is considerably less precise than both Manta and SurVIndel2.

Tandem duplications are far more difficult to predict than deletions using short reads. SurVIndel2 is the most sensitive and precise method (average sensitivity 0.35 and average precision 0.8). The next best method in term of sensitivity is SurVIndel (average sensitivity 0.3 and average precision 0.3). The next best method in term of precision is Manta (average sensitivity 0.1 and average precision 0.53)

The datasets used for training the model are very homogeneous: they were all sequenced by one sequencing center (NYGC), with the same platform (Illumina NovaSeq 6000), with the same sequencing depth (30×, except for one which was 60×), which can lead to overfitting. However, the HG002 sample confirms that this is not the case for our model. HG002

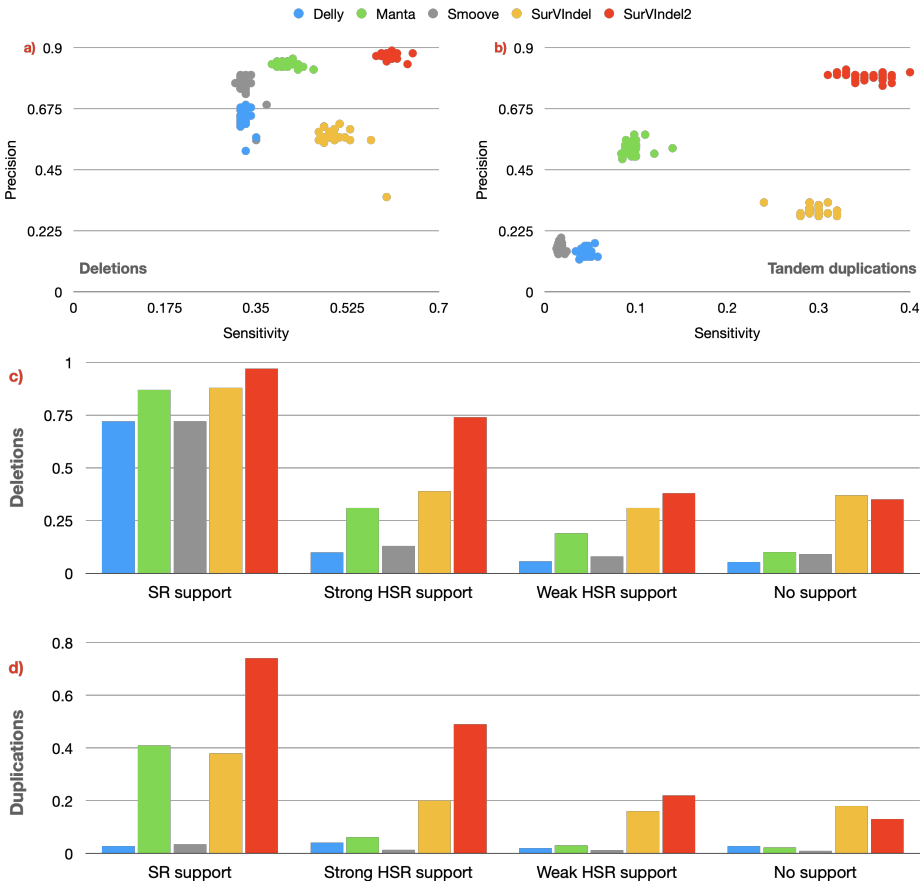

Figure 4: Sensitivity and precision for deletions (a) and tandem duplications (b) on the in the HGSVC2 catalogue. Each sample is a dot, its x coordinate is the sensitivity while its y coordinate is the precision. SurVIndel2 has higher sensitivity and precision than the other methods. (c) Sensitivity of the methods in predicting deletions in HG002, stratified by supporting evidence. Split reads (SR)-supported deletions are relatively easy for all methods, although SurVIndel2 has a clear advantage. However, SurVIndel2 is the only method that can detect most deletions with strong HSR support. (d) For tandem duplications, existing methods perform poorly even for SR-supported events. SurVIndel2 can detect nearly twice as many duplications as existing methods that are supported by split and hidden split reads.

was sequenced by a different entity (GIAB Consortium) using a different platform (Illumina HiSeq 2500) and its depth is different (50×). SurVIndel2 performs just as well on it: sensitivity and precision were 0.64 and 0.84 for deletions, 0.37 and 0.76 for duplications.

In Section 2.1 we have partitioned the CNVs in HG002 depending on whether they were supported by split reads, hidden split reads, or unsupported. In order to maintain a high precision, our algorithm only considers HSRs that have at least three differences with the reference genome (an indel or a mismatch relative to the reference is counted as one difference, see Supplementary Methods). Let us call such HSRs as *strong HSRs*. Therefore, we expect that SurVIndel2 will be able to detect HSR-supported CNVs that have good support from strong HSRs, but not the rest. To study this, we partition HSR-supported CNVs into two categories: CNVs with strong HSR support (supported by at least 5 strong HSRs) and with weak HSR support (supported by less than 5 strong HSRs). 53% and 67% of the HSR-supported deletions and duplications, respectively, have strong HSR support.

All tested methods tend to perform well in detecting deletions that are supported by split reads (Fig. 4c). SurVIndel2 detects nearly all true positive deletions in this category (0.97 sensitivity). As expected, SurVIndel2 has a large advantage over the other methods in finding deletions supported by strong HSRs (0.74 sensitivity for SurVIndel2, 0.39 for SurVIndel). Predictably, detection power drops for deletions with weak or no support, although SurVIndel2 and the original SurVIndel are still the most sensitive methods.

Surprisingly, SurVIndel2 has a large advantage not only for tandem duplications supported by strong HSRs, but also for those supported by split reads (Fig. 4d). Methods other than SurVIndel and SurVIndel2 are essentially unable to call duplications not supported by split reads, which explains their poor sensitivity.

We observed from Fig. 2c and Fig. 2d that CNVs outside of repetitive regions are more likely to be supported by split reads, while the CNVs within repetitive regions are mostly supported by hidden split reads or they are unsupported. The proportion of split reads support, hidden split reads support and no support in repetitive regions is similar between deletions and duplications; however, because a much larger proportion of duplications is within repetitive regions (Fig. 2a: 64% of deletions and 93% of duplications are in tandem duplication regions), it follows that a larger portion of duplications is not supported by split reads compared to deletions. This might explain, together with our observations from Section 2.7, why tandem duplications are so difficult to call compared to deletions.

2.4 Performance on different organisms

The filtering model we provide was trained on the CNVs called from the 34 human samples. We call this the *default model*. For this reason, we are interested in (i) how SurVIndel2 performs on different organisms using the default model and (ii) what are the implications on the sensitivity and precision of the method if a more targeted training set is used (i.e., the model is retrained using data from the target organism). To address these questions, we conducted an analysis by running the callers on seven Arabidopsis thaliana samples that were sequenced as part of the 1001 Genomes Project. These samples were mapped to the TAIR10 reference

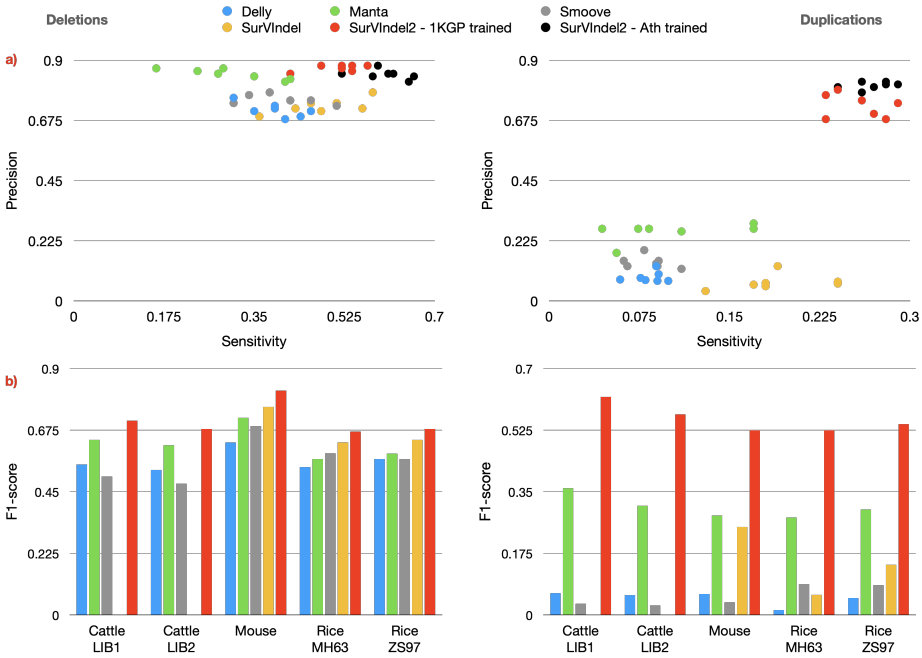

Figure 5: (a) Sensitivity and precision for deletions (left) and tandem duplications (right) in the Arabidopsis Thaliana benchmark. Unsurprisingly, the A. Thaliana-trained SurVIndel2 performs the best; however, the human-trained SurVIndel2 still performs better than the existing methods. (b) F1-score for deletions (left) and duplications (right) for five non-human libraries: two *bos taurus* (Cattle LIB1 and Cattle LIB2), one *mus musculus* (Mouse) and two *oryza sativa* (Rice Minghui 63 and Rice Zhenshan 97). Benchmark catalogues were obtained by running Sniffles2 on PacBio HiFi reads. SurVIndel2 performs better for every organism. The advantage is particularly evident for tandem duplications. SurVIndel failed to run for the two Cattle libraries.

genome. We generated a benchmark dataset by using Sniffles2 [24] on PacBio HiFi reads for the same samples. For SurVIndel2, we used two classification models: one trained on the 34 human samples with long reads in the 1000 Genome Project (Section 2.3), and one trained on the seven A. Thaliana using a leave-one-out cross-validation (i.e., when classifying sample S, sample S is excluded from the training set).

SurVIndel2 outperforms all of the existing methods (Fig. 5a). Two key points that must be noted are a) when trained on A. Thaliana data, SurVIndel2 shows noticeably improved performance in calling A. Thaliana CNVs compared to when trained on human data and b) SurVIndel2 trained on human data still outperforms existing methods. This means that if a reliable catalogue of CNVs (i.e., obtained from long reads) is available for the target organism, it is possible to increase the performance of SurVIndel2; however, if such catalogue is not available, the default model works well.

We also selected samples of other species (bos taurus [8], mus musculus [16] and oryza sativa [30, 25]) for which both Illumina paired reads and PacBio HiFi reads were available. All of the selected organisms are

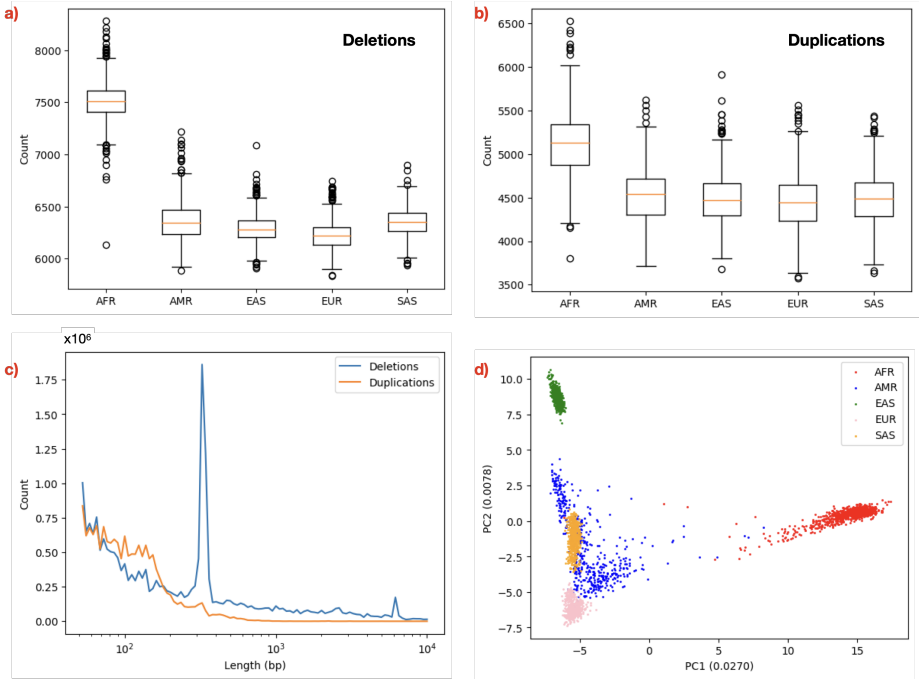

Figure 6: (a,b) Count of deletions and duplications in different superpopulations. Africans have more deletions and duplications compared to non-Africans: the median count was 7,509 deletions and 5,129 duplications for Africans vs 6,294 deletions and 4,478 duplications for non-Africans. (c) The size distribution for deletions and duplications detected in 1000g, adjusted by their frequency in the population. For deletions, the expected peaks associated with ALU (∼310bp) and LINE (∼6000bp) deletions are visible. (d) PCA is able to clearly separate the superpopulations.

important either as food sources or as model organisms. Once again, SurVIndel2 has the best performance for every sample, both for deletions and duplications (Fig. 5b). We expect that the performance would be even better if the model was trained for the specific organism, as in the case of A. Thaliana.

2.5 A catalogue of deletions and tandem duplications for the 3,202 genomes in the 1000 genomes project

We used SurVIndel2 to generate a catalogue of deletions and tandem duplications for 3,202 samples from the 1000g genomes project sequenced by the NYGC. SurVIndel2 was run on each individual sample, and the calls were left-aligned and clustered across samples (Supplementary Methods). The final catalogue contained 247,822 deletions and 232,227 tandem duplications. 205,258 deletions (82.8%) and 189,413 duplications (81.6%) were rare (detected in 1% of the population or less), and 90,137 deletions (36.4%) and 74,369 duplications (32.0%) were singletons (detected in only one individual). Africans had more deletions (Fig. 6a) and duplications

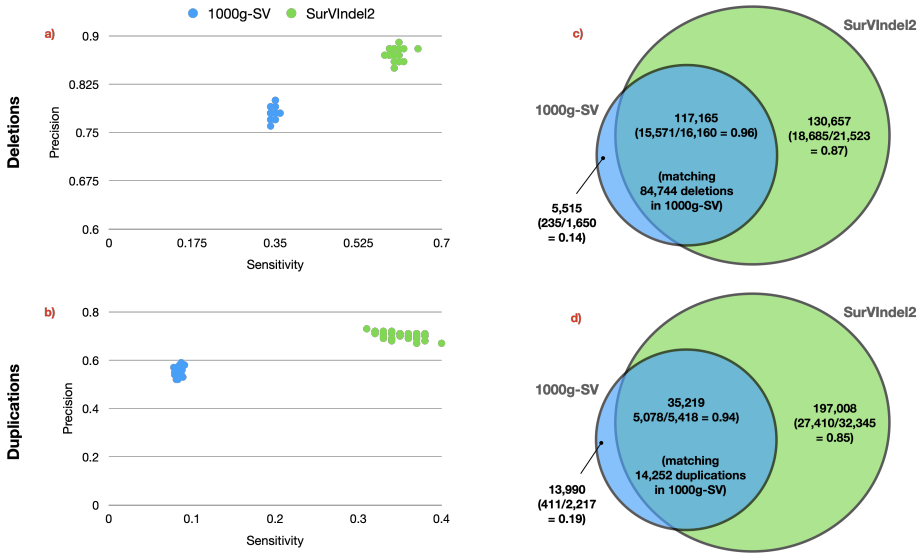

Figure 7: Comparison between the 1000g-SV and SurVIndel2 datasets. (a,b) Sensitivity and the precision for the 34 samples in HGSVC2 for deletions (a) and tandem duplications (b). The dataset produced by SurVIndel2 is both more sensitive and more precise than the 1000g-SV dataset. (c) Overlap between deletions in 1000g-SV and SurVIndel2 datasets. Between parentheses, the validation rates of calls in samples with long reads. Not only SurVIndel2 covers nearly all of the deletions in 1000g-SV, but it predicts 130,657 deletions that are not present in the latter. The validation rate for SurVIndel2-unique events is far higher than the validation rate of 1000g-SV unique events. (d) Similarly, SurVIndel2 predicts a large portion of the tandem duplications in 1000g-SV, and the ones that are not predicted have very low validation rate. SurVIndel2 also predicts 197,008 duplications that are not present in 1000g-SV, with high validation rate.

(Fig. 6b) compared to non-Africans: the median count was 7,509 deletions and 5,129 duplications for Africans vs 6,294 deletions and 4,478 duplications for non-Africans. The number of events was similar in the remaining superpopulations. Fig. 6c shows the size distribution for the deletions and duplications predicted. The peaks associated with ALU (∼310 bp) and LINE (∼6 kbp) deletions are clearly visible. PCA is able to separate the superpopulations (Fig. 6d). This is also true when performing the PCA using deletions and tandem duplications separately (Fig. S16). Furthermore, PCA was able to segregate the superpopulations into subpopulation (Fig. S17).

The NYGC also generated a catalogue of SVs (including deletions and tandem duplications) [3] by integrating multiple callers, including Manta. We refer to this catalogue as 1000g-SV. Fig. 7 shows a comparison between 1000g-SV and our catalogue. Similarly to Section 2.3, we measured sensitivity and precision for 1000g-SV and for SurVIndel2 on the 34 samples in HGSVC2. SurVIndel2 had consistently higher sensitivity and precision. For deletions (Fig. 7a), SurVIndel2 was on average nearly twice as sensitive (average sensitivity was 0.345 for 1000g-SV and 0.60

for SurVIndel2), while being more precise (average precision was 0.78 for 1000g-SV and 0.87 for SurVIndel2). For tandem duplications (Fig. 7b), SurVIndel2 was more than 4× more sensitive (average sensitivity was 0.084 for 1000g-SV and 0.35 for SurVIndel2) and significantly more precise (0.55 for 1000g-SV, 0.80 for SurVIndel2). When comparing the full catalogues (Fig. 7c, 7d), SurVIndel2 captures the vast majority of the deletions in 1000g-SV, and deletions in 1000g-SV but not predicted by SurVIndel2 have low validation rate according to HGSVC2 (0.14). Furthermore, SurVIndel2 predicts 130,657 deletions not present in 1000g-SV, with high validation rate (0.87). The difference in detected tandem duplications is even more striking. Duplications predicted by 1000g-SV but not by SurVIndel2 have again very low validation rate (0.19). The two datasets share 35,219 duplications, and SurVIndel2 predicts an additional 197,008 duplications, with high validation rate (0.85). Overall, it appears that SurVIndel2 is able to predict a more accurate and significantly more complete catalogue of both deletions and tandem duplications.

2.6 SurVIndel2 enhances small indels callers

Although SurVIndel2 was developed and tuned to detect SVs (≥50 bp), its approach is theoretically able to recover some of the smaller indels, as long as they produce a sufficient number of split or hidden split reads. Therefore, we investigated whether SurVIndel2 is able to complement and complete the calls provided by DeepVariant, since it has been found the best-performing tool for small indels [26, 11]. We focused on indels of size between 30 and 50 bp, since smaller indels are unlikely to generate enough evidence for SurVIndel2 to use.

In order to measure the sensitivity of the methods, we use the benchmark catalogue for small indels provided by HGSVC2. However, we noticed that this benchmark underestimates the precision of the methods. Upon manual inspection using PacBio HiFi reads and IGV [21], most of the calls by DeepVariant or SurVIndel2 that were missing from the benchmark were clearly supported by the long reads. For this reason, in order to more accurately measure precision, we created a more complete benchmark for both HG00512 and HG002 by concatenating three callsets: i) HGSVC2; ii) Sniffles2 calls on PacBio HiFi reads and iii) DeepVariant calls on PacBio HiFi reads.

Fig. 8a and Supplementary Fig. S18 illustrate the improved sensitivity achieved by using both DeepVariant and SurVIndel2 for the detection of deletions. On average, the sensitivity for DeepVariant alone was 0.6, whereas this increased to 0.71 when SurVIndel2 was added. This increase was not at the expense of precision, which stayed the same (0.97).

The improvement in sensitivity was even more pronounced for small insertions (Fig. 8b): by adding SurVIndel2 the average sensitivity increased from 0.38 to 0.52. There was an apparent decrease in precision, from 0.99 with DeepVariant alone to 0.93 when combined with SurVIndel2 for HG00512, and from 0.97 to 0.92 for HG002. However, upon manual inspection of PacBio HiFi reads using IGV, we noticed that a significant portion of the insertions marked as false positives had clear support from the alignment of the reads. In particular, we randomly selected 30 examples of insertions marked as false positives in HG00512, and for 15 of them (50%) the HiFi long reads supported the presence of an insertion (IGV views available for all 30 in Data Availability). Similarly for HG002, 17 out of 30 (57%) of the apparent false positive had clear support from

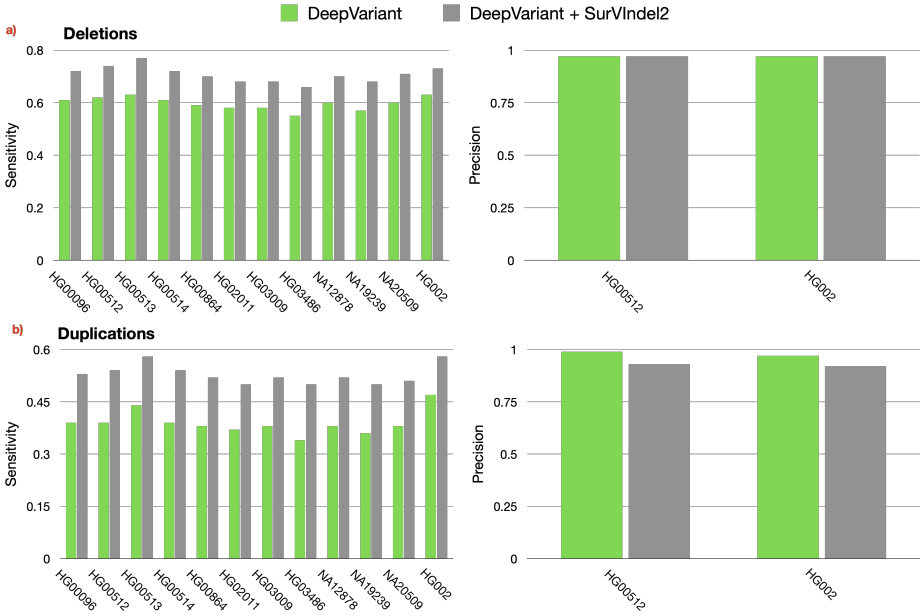

Figure 8: SurVIndel2 recovers small indels (30 to 50 bp) missed by DeepVariant. a) By using both DeepVariant and SurVIndel2, we are able to recover more true positive small deletions compared to using DeepVariant alone (average sensitivity was 0.6 for DeepVariant, 0.71 for DeepVariant plus SurVIndel2). The precision was not changed (0.97 for both). b) The increase in sensitivity was even greater for small insertions. The average sensitivity for DeepVariant was 0.38, which increased to 0.52 when introducing SurVIndel2. Precision apparently decreased from 0.99 to 0.93; however, when manually examining 30 random calls marked as true positives, 15 of them showed clear support from PacBio HiFi reads. Therefore, the projected precision of DeepVariant plus SurVIndel2 is 0.975. Supplementary Fig. S18 and S19 show the sensitivity for all 35 samples in HGSVC2.

the HiFi long reads (IGV views available for all 30 in Data Availability). Consequently, the projected precision for the DeepVariant and SurVIndel2 combination is recalibrated to be approximately 0.975 for HG00512 and 0.966 for HG002, which suggests that the decrease in precision was far more modest than it initially appeared.

Overall, our findings indicate that the integration of SurVIndel2 with DeepVariant can provide a remarkably sensitive and precise catalogue of variants for an individual, from very small to very large, far more complete than what was possible without SurVIndel2.

2.7 Reasons for unsupported CNVs

Many existing CNVs have no support. In particular, in Fig. 2c and Fig. 2d we showed that 41% of deletions and 36% of duplications in tandem repetitive regions in HG002 have no sufficient support from split and hidden split reads. We will refer to them as *unsupported CNV*s. This section aims to find the reasons behind the lack of supporting reads for unsupported

CNVs.

As mentioned in Section 2.1, we expect that the major cause of unsupported CNV is that the sequences flanking the junction sequence created by the CNV (deletion or duplication) is not a novel sequence. Every read sequenced from such junction sequence will perfectly align to the reference genome, hence it will not support the existence of a CNV. This scenario is illustrated in Fig. 1c and Fig. 1d.

To formalise this notion, we present a probability measure for each CNV, which we call the *expected support (ES) score*. The ES score for a CNV represents the likelihood that a read which contains a breakpoint of the CNV will provide evidence for the presence of the CNV when it is aligned to the reference sequence (i.e., it will be a split or a hidden split read). When ES is 0, we do not expect any read to support the CNV, while a ES of 1 means that we expect every read containing a breakpoint to support the CNV. The calculation of the ES value is described in Supplementary Methods. We expect that most unsupported CNVs will be due to a very low ES score.

A large portion of the unsupported deletions in HG002 does indeed have low ES score (Fig. 9a, left), while supported deletions tend to have much higher ES values (Fig. 9a, right). However, more than 40% of the unsupported deletions have high ES (≥ 0.5), which suggests that a low ES is not the only reason for lack of support. We found that many deletions are unsupported because the reads sequenced from the breakpoint region are very low in quality. Supplementary Fig. S7 and S8 show examples of such deletions.

To demonstrate how common this issue is, we built high-quality alternative alleles for a subset of CNVs in HG002 (Supplementary Methods). Then, we used BWA MEM to map the paired reads of HG002 (Illumina HiSeq 2500) to the alternative alleles, and, for each CNV, we counted the reads correctly sequenced (i.e., without mismatches or indel) that contain a breakpoint. Given that the dataset has a coverage of about 50x, we expect the number of reads covering the breakpoint of the deletion to be around 25 for a heterozygous deletion and 50 for a homozygous one. Fig. 9b shows that unsupported deletions with high ES tend to have fewer reads covering the breakpoint than expected: 91% of them have < 10 reads, and 64% have < 5. Conversely, supported deletions tend to have more reads containing the breakpoints (average 18.4 for supported high ES deletions, 3.7 for unsupported high ES deletions). Therefore, two ingredients determine the level of support for a deletion: the ES score, and the number of reads that contain the breakpoint.

Surprisingly, unlike unsupported deletions, we observed that most unsupported tandem duplications (77%) have high ES score (Fig. 10a). Only 16% have low ES score. In fact, excluding the very low ES values (0-0.1), the distribution of ES score is similar between unsupported (left) and supported (right) duplications. The issue of poorly sequenced breakpoint regions is even more prevalent in duplications than it is in deletions: 96% have < 10 reads, and 85% have < 5. As for deletions, supported duplications tend to have a much higher number of reads containing a breakpoint (average 16.6 for supported high ES duplications, 1.6 for unsupported high ES duplications).

Repeating these experiments on the HG00512 sample sequenced by the NYGC using a different platform (Illumina NovaSeq 6000) yields the same conclusions (Supplementary Fig. S9 and S10). These experiments demonstrate that there is a serious issue in sequencing a large number

of the regions containing CNVs, and variant calling is unlikely to further improve significantly unless such issues are mitigated.

To reinforce our hypothesis, we create a synthetic dataset for HG002, which we refer to as HG002-SIM, that closely resembles actual sequencing data but does not have the issue of poorly sequenced regions. We achieve this by using the ART read simulator [10] to simulate an Illumina HiSeq 2500 50x dataset from a high-quality HG002 T2T assembly [29, 20]. The results match out expectations: nearly all unsupported deletions and duplications have low ES (Supplementary Fig. S11, S12), and their number is greatly reduced. For this reason the sensitivity of SurVIndel2 is noticeably higher on the synthetic dataset, especially for duplications (Supplementary Fig. S13 and S14).

In conclusion, our findings reveal that lack of evidence for the existing CNVs is only partially due to the regions containing the CNVs being very repetitive: another major reason is that the sequencing of many such regions is extremely noisy. This seems to be especially true for regions prone to tandem duplications. We have shown through a realistic synthetic dataset that without this issue, the number of CNVs detectable using short reads would be much higher.

3 Discussion

Existing CNV callers rely on read depth, split reads and discordant pairs when detecting CNVs from short reads. In this paper, we described how CNVs in repetitive regions often do not generate such evidence. For this reason, we introduced a new type of evidence, called hidden split reads. By using it, we are potentially able to discover more than double the number of deletions and duplications compared to only using split reads. From this observation, we developed SurVIndel2, a CNV caller that uses split reads and discordant pairs as well as hidden split reads to produce a comprehensive set of candidate CNVs. SurVIndel2 borrows the idea of statistical testing from its predecessor, SurVIndel, and uses it in a machine-learning filtering algorithm to produce a comprehensive and accurate catalogue of CNVs.

We used publicly available benchmarks to show that SurVIndel2 outperforms the state of the art in CNV calling. On the HGSVC2 benchmark, a comprehensive catalogue of the CNVs in 35 human samples, SurVIndel2 was the most sensitive and precise caller for both deletions and duplications, often by a large margin. The improvements were particularly large for duplications, since all of the existing methods had either very low sensitivity or very low precision.

Our machine learning filtering was trained on human data, so we were interested in two key questions: how does it performs on non-human data, and is it possible to improve the performance by training for the target organism. We investigated this using a variety of different organisms, and we learned that (i) the human model works well for every tested organism and (ii) the performance increase even further when the model is trained for the specific organism.

To demonstrate how SurVIndel2 is useful in practice, we used it to generate a catalogue of deletions and tandem duplications for the 1000 Genome Project, and compared it to a recently published catalogue generated using state-of-the-art pipelines. Our catalogue revealed hundreds of thousands of events that were not present in the existing one, and the

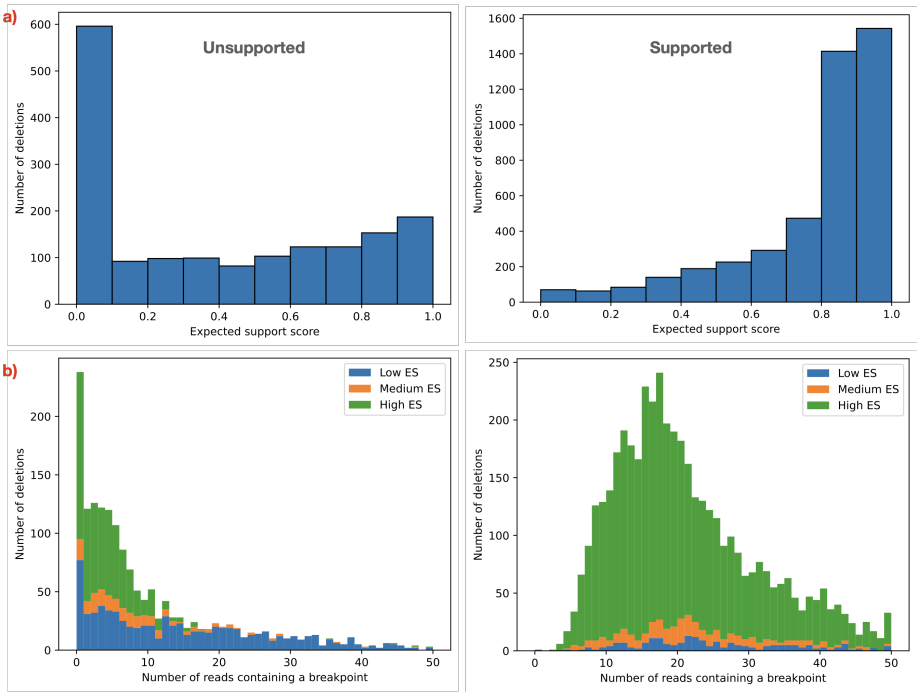

Figure 9: Factors that contribute to absence of support for deletions in HG002. a) As expected, a large portion of unsupported deletions have low expected support (ES) score (left). ES score is the probability that a read containing the breakpoint of the deletion will support its existence, i.e., it will be either a split or a hidden split read. On the other hand, most deletions with support have very high ES score (right). However, a low ES score is not the only factor that contributes to lack of support: more than 40% of the unsupported deletions have ES score ≥ 0.5. b) Another factor is a lack of correctly sequenced reads containing the breakpoint of the deletions. For each deletion with a high quality alternative allele (Supplementary Methods), we mapped the short reads of HG002 to the alternative allele and counted the reads correctly sequenced (i.e., without mismatches or indels) and containing the deletion breakpoint. We further stratified the deletions into low ES (< 0.3), medium ES (≥ 0.3 and < 0.5) and high ES (≥ 0.5). High ES unsupported deletions (left) tend to have very few or no reads correctly sequenced that contain the breakpoint, while deletions with a higher number of reads containing the breakpoint have low ES. Supported deletions (right), on the other hand, have both ingredients necessary for support: reads containing the breakpoint, and high ES.

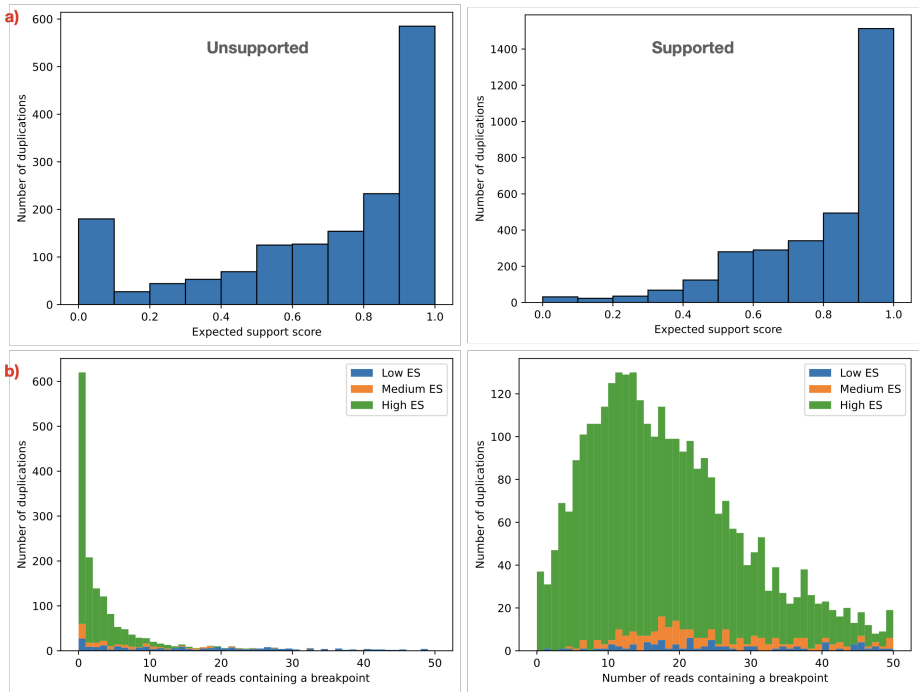

Figure 10: Factors that contribute to absence of support for duplications in HG002. a) Surprisingly, only 16% of the unsupported duplications (left) have low ES (< 0.3). Most unsupported duplications (77%) have high ES (≥ 0.5). Supported duplications also have high ES, as expected (right). b) The effect of poorly sequenced regions is even more dramatic on tandem duplications than it is on deletions. Despite the high ES, most unsupported sequences have very few or no reads containing a breakpoint.

long read-based validation rates for those events were high. We expect that SurVIndel2 will be beneficial to the community, and will help uncover a large number of CNVs that would be missed by using existing tools, while maintaining the number of false positives to a minimum.

We also examined the ability of SurVIndel2 to complement small variant callers, in particular DeepVariant. Our findings indicate that incorporating SurVIndel2 with DeepVariant significantly improved the sensitivity for both deletions and small insertions detection. Notably, this increase in sensitivity did not compromise precision. Using DeepVariant with SurVIndel2 provides a view of the full spectrum of variants in an individual, from small to large, that is more precise and complete than what was previously possible.

Finally, we investigated why a large portion of CNVs lacked support from both split and hidden split reads. We found that the reads in the breakpoint regions of many CNVs were extremely noisy, and for many CNVs not a single correctly sequenced reads contained a breakpoint. This was especially serious for tandem duplications, and we have also shown by simulation that if this issue was not present, the number of true positive duplications predicted would greatly increase. Although it is known that some genomic regions are difficult to sequence, this was the first study, to our knowledge, that systematically investigated the magnitude of the problem and its impact on CNV calling.

References

[1] Haley J Abel, David E Larson, Allison A Regier, Colby Chiang, Indraniel Das, Krishna L Kanchi, Ryan M Layer, Benjamin M Neale, William J Salerno, Catherine Reeves, Steven Buyske, NHGRI Centers for Common Disease Genomics, Tara C Matise, Donna M Muzny, Michael C Zody, Eric S Lander, Susan K Dutcher, Nathan O Stitziel, and Ira M Hall. Mapping and characterization of structural variation in 17,795 human genomes. *Nature*, 583(7814):83–89, Jul 2020.

[2] G Benson. Tandem repeats finder: a program to analyze dna sequences. *Nucleic Acids Res*, 27(2):573–80, Jan 1999.

[3] Marta Byrska-Bishop, Uday S Evani, Xuefang Zhao, Anna O Basile, Haley J Abel, Allison A Regier, André Corvelo, Wayne E Clarke, Rajeeva Musunuri, Kshithija Nagulapalli, Susan Fairley, Alexi Runnels, Lara Winterkorn, Ernesto Lowy, Human Genome Structural Variation Consortium, Paul Flicek, Soren Germer, Harrison Brand, Ira M Hall, Michael E Talkowski, Giuseppe Narzisi, and Michael C Zody. High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440.e19, Sep 2022.

[4] Daniel L Cameron, Leon Di Stefano, and Anthony T Papenfuss. Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software. *Nat Commun*, 10(1):3240, Jul 2019.

[5] Xiaoyu Chen, Ole Schulz-Trieglaff, Richard Shaw, Bret Barnes, Felix Schlesinger, Morten Källberg, Anthony J Cox, Semyon Kruglyak, and Christopher T Saunders. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, 32(8):1220–2, 04 2016.

[6] Ryan L Collins, Harrison Brand, Konrad J Karczewski, Xuefang Zhao, Jessica Alföldi, Laurent C Francioli, Amit V Khera, Chelsea Lowther, Laura D Gauthier, Harold Wang, Nicholas A Watts, Matthew Solomonson, Anne O'Donnell-Luria, Alexander Baumann, Ruchi Munshi, Mark Walker, Christopher W Whelan, Yongqing Huang, Ted Brookings, Ted Sharpe, Matthew R Stone, Elise Valkanas, Jack Fu, Grace Tiao, Kristen M Laricchia, Valentin Ruano-Rubio, Christine Stevens, Namrata Gupta, Caroline Cusick, Lauren Margolin, Genome Aggregation Database Production Team, Genome Aggregation Database Consortium, Kent D Taylor, Henry J Lin, Stephen S Rich, Wendy S Post, Yii-Der Ida Chen, Jerome I Rotter, Chad Nusbaum, Anthony Philippakis, Eric Lander, Stacey Gabriel, Benjamin M Neale, Sekar Kathiresan, Mark J Daly, Eric Banks, Daniel G MacArthur, and Michael E Talkowski. A structural variation reference for medical and population genetics. *Nature*, 581(7809):444–451, May 2020.

[7] Peter Ebert, Peter A Audano, Qihui Zhu, Bernardo Rodriguez-Martin, David Porubsky, Marc Jan Bonder, Arvis Sulovari, Jana Ebler, Weichen Zhou, Rebecca Serra Mari, Feyza Yilmaz, Xuefang Zhao, PingHsun Hsieh, Joyce Lee, Sushant Kumar, Jiadong Lin, Tobias Rausch, Yu Chen, Jingwen Ren, Martin Santamarina, Wolfram Höps, Hufsah Ashraf, Nelson T Chuang, Xiaofei Yang, Katherine M Munson, Alexandra P Lewis, Susan Fairley, Luke J Tallon, Wayne E Clarke, Anna O Basile, Marta Byrska-Bishop, André Corvelo, Uday S Evani, Tsung-Yu Lu, Mark J P Chaisson, Junjie Chen, Chong Li, Harrison Brand, Aaron M Wenger, Maryam Ghareghani, William T Harvey, Benjamin Raeder, Patrick Hasenfeld, Allison A Regier, Haley J Abel, Ira M Hall, Paul Flicek, Oliver Stegle, Mark B Gerstein, Jose M C Tubio, Zepeng Mu, Yang I Li, Xinghua Shi, Alex R Hastie, Kai Ye, Zechen Chong, Ashley D Sanders, Michael C Zody, Michael E Talkowski, Ryan E Mills, Scott E Devine, Charles Lee, Jan O Korbel, Tobias Marschall, and Evan E Eichler. Haplotype-resolved diverse human genomes and integrated analysis of structural variation. *Science*, 372(6537), 04 2021.

[8] Camille Eché, Carole Iampietro, Clément Birbes, Andreea Dréau, Claire Kuchly, Arnaud Di Franco, Christophe Klopp, Thomas Faraut, Sarah Djebali, Adrien Castinel, Matthias Zytnicki, Erwan Denis, Mekki Boussaha, Cécile Grohs, Didier Boichard, Christine Gaspin, Denis Milan, and Cécile Donnadieu. A bos taurus sequencing methods benchmark for assembly, haplotyping, and variant calling. *Sci Data*, 10(1):369, Jun 2023.

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231. AAAI Press, 1996.

[10] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–4, Feb 2012.

[11] Manojkumar Kumaran, Umadevi Subramanian, and Bharanidharan Devarajan. Performance assessment of variant calling pipelines using

human whole exome sequencing and simulated data. *BMC Bioinformatics*, 20(1):342, Jun 2019.

[12] Ryan M Layer, Colby Chiang, Aaron R Quinlan, and Ira M Hall. Lumpy: a probabilistic framework for structural variant discovery. *Genome Biol*, 15(6):R84, Jun 2014.

[13] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem, 2013.

[14] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, Sep 2018.

[15] Dong-Xu Liu, Ramesh Rajaby, Lu-Lu Wei, Lei Zhang, Zhi-Quan Yang, Qing-Yong Yang, and Wing-Kin Sung. Calling large indels in 1047 arabidopsis with indelensembler. *Nucleic Acids Res*, 49(19):10879–10894, Nov 2021.

[16] Juana G Manuel, Hillary B Heins, Sandra Crocker, Julie A Neidich, Lisa Sadzewicz, Luke Tallon, and Tychele N Turner. High coverage highly accurate long-read sequencing of a mouse neuronal cell line using the pacbio revio sequencer. *bioRxiv*, Jun 2023.

[17] Ramesh Rajaby, Dong-Xu Liu, Chun Hang Au, Yuen-Ting Cheung, Amy Yuet Ting Lau, Qing-Yong Yang, and Wing-Kin Sung. Insurveyor: improving insertion calling from short read sequencing data. *Nat Commun*, 14(1):3243, Jun 2023.

[18] Ramesh Rajaby and Wing-Kin Sung. Survindel: improving cnv calling from high-throughput sequencing data through statistical testing. *Bioinformatics*, Apr 2019.

[19] Tobias Rausch, Thomas Zichner, Andreas Schlattl, Adrian M Stütz, Vladimir Benes, and Jan O Korbel. Delly: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, 28(18):i333–i339, Sep 2012.

[20] Mikko Rautiainen, Sergey Nurk, Brian P Walenz, Glennis A Logsdon, David Porubsky, Arang Rhie, Evan E Eichler, Adam M Phillippy, and Sergey Koren. Telomere-to-telomere assembly of diploid chromosomes with verkko. *Nat Biotechnol*, Feb 2023.

[21] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. Integrative genomics viewer. *Nat Biotechnol*, 29(1):24–6, Jan 2011.

[22] Varuni Sarwal, Sebastian Niehus, Ram Ayyala, Minyoung Kim, Aditya Sarkar, Sei Chang, Angela Lu, Neha Rajkumar, Nicholas Darfci-Maher, Russell Littman, Karishma Chhugani, Arda Soylev, Zoia Comarova, Emily Wesel, Jacqueline Castellanos, Rahul Chikka, Margaret G Distler, Eleazar Eskin, Jonathan Flint, and Serghei Mangul. A comprehensive benchmarking of wgs-based deletion structural variant callers. *Brief Bioinform*, 23(4), Jul 2022.

[23] Kishwar Shafin, Trevor Pesout, Pi-Chuan Chang, Maria Nattestad, Alexey Kolesnikov, Sidharth Goel, Gunjan Baid, Mikhail Kolmogorov, Jordan M Eizenga, Karen H Miga, Paolo Carnevali, Miten Jain, Andrew Carroll, and Benedict Paten. Haplotype-aware variant calling with pepper-margin-deepvariant enables high accuracy in nanopore long-reads. *Nat Methods*, 18(11):1322–1332, Nov 2021.

[24] Moritz Smolka, Luis F. Paulin, Christopher M. Grochowski, Medhat Mahmoud, Sairam Behera, Mira Gandhi, Karl Hong, Davut Pehlivan,

Sonja W. Scholz, Claudia M.B. Carvalho, Christos Proukakis, and Fritz J Sedlazeck. Comprehensive structural variant detection: From mosaic to population-level. *bioRxiv*, 2022.

[25] Jia-Ming Song, Wen-Zhao Xie, Shuo Wang, Yi-Xiong Guo, Dal-Hoe Koo, Dave Kudrna, Chenbo Gong, Yicheng Huang, Jia-Wu Feng, Wenhui Zhang, Yong Zhou, Andrea Zuccolo, Evan Long, Seunghee Lee, Jayson Talag, Run Zhou, Xi-Tong Zhu, Daojun Yuan, Joshua Udall, Weibo Xie, Rod A Wing, Qifa Zhang, Jesse Poland, Jianwei Zhang, and Ling-Ling Chen. Two gap-free reference genomes and a global view of the centromere architecture in rice. *Mol Plant*, 14(10):1757–1767, Oct 2021.

[26] Anna Supernat, Oskar Valdimar Vidarsson, Vidar M Steen, and Tomasz Stokowy. Comparison of three variant callers for human whole genome sequencing. *Sci Rep*, 8(1):17851, Dec 2018.

[27] Brett Trost, Susan Walker, Zhuozhi Wang, Bhooma Thiruvahindrapuram, Jeffrey R MacDonald, Wilson W L Sung, Sergio L Pereira, Joe Whitney, Ada J S Chan, Giovanna Pellecchia, Miriam S Reuter, Si Lok, Ryan K C Yuen, Christian R Marshall, Daniele Merico, and Stephen W Scherer. A comprehensive workflow for read depth-based identification of copy-number variation from whole-genome sequence data. *Am J Hum Genet*, 102(1):142–155, Jan 2018.

[28] Robert Vaser, Ivan Sović, Niranjan Nagarajan, and Mile Šikić. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res*, 27(5):737–746, May 2017.

[29] Ting Wang, Lucinda Antonacci-Fulton, Kerstin Howe, Heather A Lawson, Julian K Lucas, Adam M Phillippy, Alice B Popejoy, Mobin Asri, Caryn Carson, Mark J P Chaisson, Xian Chang, Robert Cook-Deegan, Adam L Felsenfeld, Robert S Fulton, Erik P Garrison, Nanibaa' A Garrison, Tina A Graves-Lindsay, Hanlee Ji, Eimear E Kenny, Barbara A Koenig, Daofeng Li, Tobias Marschall, Joshua F McMichael, Adam M Novak, Deepak Purushotham, Valerie A Schneider, Baergen I Schultz, Michael W Smith, Heidi J Sofia, Tsachy Weissman, Paul Flicek, Heng Li, Karen H Miga, Benedict Paten, Erich D Jarvis, Ira M Hall, Evan E Eichler, David Haussler, and Human Pangenome Reference Consortium. The human pangenome project: a global resource to map genomic diversity. *Nature*, 604(7906):437–446, Apr 2022.

[30] Xin Wei, Jie Qiu, Kaicheng Yong, Jiongjiong Fan, Qi Zhang, Hua Hua, Jie Liu, Qin Wang, Kenneth M Olsen, Bin Han, and Xuehui Huang. A quantitative genomics map of rice provides genetic insights and guides breeding. *Nat Genet*, 53(2):243–253, Feb 2021.

[31] Joachim Weischenfeldt, Orsolya Symmons, François Spitz, and Jan O Korbel. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nat Rev Genet*, 14(2):125–38, Feb 2013.