Genotyping structural variation in variation graphs with the vg toolkit

This manuscript (permalink) was automatically generated from jmonlong/manu-vgsv@b1fc19e on April 8, 2019.

Authors

Glenn Hickey[1,☯], David Heller[1,☯], Jean Monlong[1,☯], Adam Novak[1], Benedict Paten[1,†]

[☯] — These authors contributed equally to this work

[†] — To whom correspondence should be addressed: bpaten@ucsc.edu

1. UC Santa Cruz Genomics Institute, University of California, Santa Cruz, California, USA

Abstract

Variation graphs have been proposed to represent human genomes and improve on different aspects of genomics analysis, such as read mapping and variant calling. Structural variants (SVs) have been associated with diseases but they remain under-studied due to their complexity and certain technological challenges. Still, thousands of SVs have been characterized and better catalogs are now being generated thanks to new technologies. There is now an opportunity to integrate previously neglected SVs into the unified framework of variation graphs. We first show that the vg toolkit is capable of genotyping insertions, deletions and inversions, even in the presence of small errors in the variant definition. We then benchmarked vg across three high-quality sequence-resolved SV catalogs generated by recent studies. vg was compared to state-of-the-art SV genotypers using simulated and real Illumina short reads. On real data, vg produced the best genotype predictions systematically in all datasets. Finally, we found that better graphs could be constructed directly from de novo assembly alignment. We experimented with assemblies from 12 yeast strains and showed that SV genotyping was improved compared to graphs built from intermediate SV catalogs in the VCF format. Our results demonstrate the power of variation graphs for SV genotyping. Beyond single nucleotide variants and short insertions/deletions, the vg toolkit now includes SV in its unified variant calling framework and provides a natural solution to integrate high-quality SV catalogs and assemblies.

Introduction

Structural variation (SV) represents genomic mutation involving 50 bp or more and can take several forms, such as for example deletions, insertions, inversions, or translocations. SVs have long been associated with developmental disorders, cancer and other complex diseases and phenotypes[1]. However, SVs have been under-studied for technological reasons and due to their complexity as compared to other types of genomic variation. Although whole-genome sequencing (WGS) made it possible to assess virtually any type of structural variation, many challenges remain. SV-supporting reads are generally difficult to map to reference genomes, in part because most SVs are larger than the sequencing reads. Repeated sequences in the genome often confuse read mapping algorithms, which can produce mappings that seem to support an SV. In practice, large-scale projects had to combine several methods to achieve better accuracy. This methodology has been used to compile catalogs with tens of thousands of SVs in humans[2,3]. Overall, SV detection from short-read sequencing remains laborious and of lower accuracy than small variant detection. This explains why these variants and their impact have been under-studied as compared to single-nucleotide variants (SNVs) and small insertions/deletions (indels).

Over the last few years, exciting developments in sequencing technologies and library preparation have made it possible to produce long reads or retrieve long-range information over kilobases of sequence. This is particularly useful for SV detection and de novo assembly. Several recent studies using long-read or linked-read sequencing have produced large catalogs of structural variation, the majority of which was novel and sequence-resolved[4,5,6,7,8]. These technologies are also enabling the production of high-quality de novo genome assemblies[4,9], and large blocks of haplotype-resolved sequences[10]. Such technical advances promise to expand the amount of known genomic variation in humans in the near future. However, their cost prohibits their use in large-scale studies that require hundreds or thousands of samples, such as disease association studies.

At the same time, the reference genome is evolving from a linear reference to a graph-based reference that incorporates known genomic variation[11,12,13]. By including variants in the graph, both read mapping and variant calling become variant-aware and benefit in term of accuracy and sensitivity[12]. In addition, different variant types are called simultaneously by a unified framework. vg was the first openly available tool that scaled to multi-gigabase genomes and provides read mapping, variant calling and haplotype modeling[12]. In vg, graphs can be built from both variant catalogs in the VCF format or assembly alignment. Other genome graph implementations have also been used specifically to genotype variants. Using a sliding-window approach, GraphTyper realigns reads to a graph build from known SNVs and short indels[14]. BayesTyper build graphs of both short variants and SVs, and genotypes variants based on the khmer distribution of sequencing reads[15]. Here again, the graph-based approaches showed clear advantages over standard methods that use the linear reference.

Other SV genotyping approaches typically compare read mapping to the reference genome and to a sequence modified with the SV. For example SMRT-SV was designed to genotype SVs identified on PacBio reads[5]. SVTyper uses paired-end mapping and split-read mapping information and can genotype deletions, duplications, inversions and translocations[16]. Delly provides a genotyping feature in addition to its discovery mode and can genotype all types of SVs although the VCF needs special formatting for some[17]. SMRT-SV2 is a machine learning tool that was trained to genotype SVs from the alignment of read to the reference genome augmented with SV-containing sequences as alternate contigs[8].

We show that the unified variant calling framework implemented in vg is capable of genotyping deletions, insertions and inversions. We compare vg with state-of-the-art SV genotypers: SVTyper[16], Delly[17], BayesTyper[15] and SMRT-SV2[8]. On simulation, vg is robust to small errors in the breakpoint location and outperforms most other methods on shallow sequencing experiments. Starting from SVs discovered in recent long-read sequencing studies[18,19,20,8], we evaluated the genotyping accuracy when using simulated or real Illumina reads. Across all three datasets that we tested, vg is the best performing SV genotyper on real short-read data for all SV types and sizes. Going further, we show that building graphs from the alignment of de novo assemblies leads to better genotyping performance.

Results

Structural variation in vg

In addition to SNV and short indels, vg can handle large deletions, insertions and inversions (Figure 1a). As a proof-of-concept we simulated genomes and different types of SVs with a size distribution matching real SVs[18]. We compared vg against SVTyper, Delly and BayesTyper across different level of sequencing depth. Some errors were also added at the breakpoints to investigate their effect on genotyping (see Methods). The results are shown in Figure 1b. When using the correct breakpoints, vg tied with Delly as the best genotyper for deletions, and with BayesTyper as the best genotyper for insertions. For inversions, vg was the second best genotyper after BayesTyper. The differences between the methods were the most visible at lower sequencing depth. In the presence of 1-10 bp errors in the breakpoint location, the performance of Delly and BayesTyper dropped significantly. The dramatic drop for BayesTyper can be explained by its khmer-based approach that requires exact SV definition. In contrast, vg was only slightly affected by the presence of errors in the input VCF (Figure 1b). For vg, the F1 scores for all SV types decreased no more than of 0.07 point. Overall, these results show that vg is capable of genotyping SVs and is robust to errors in the input VCF.

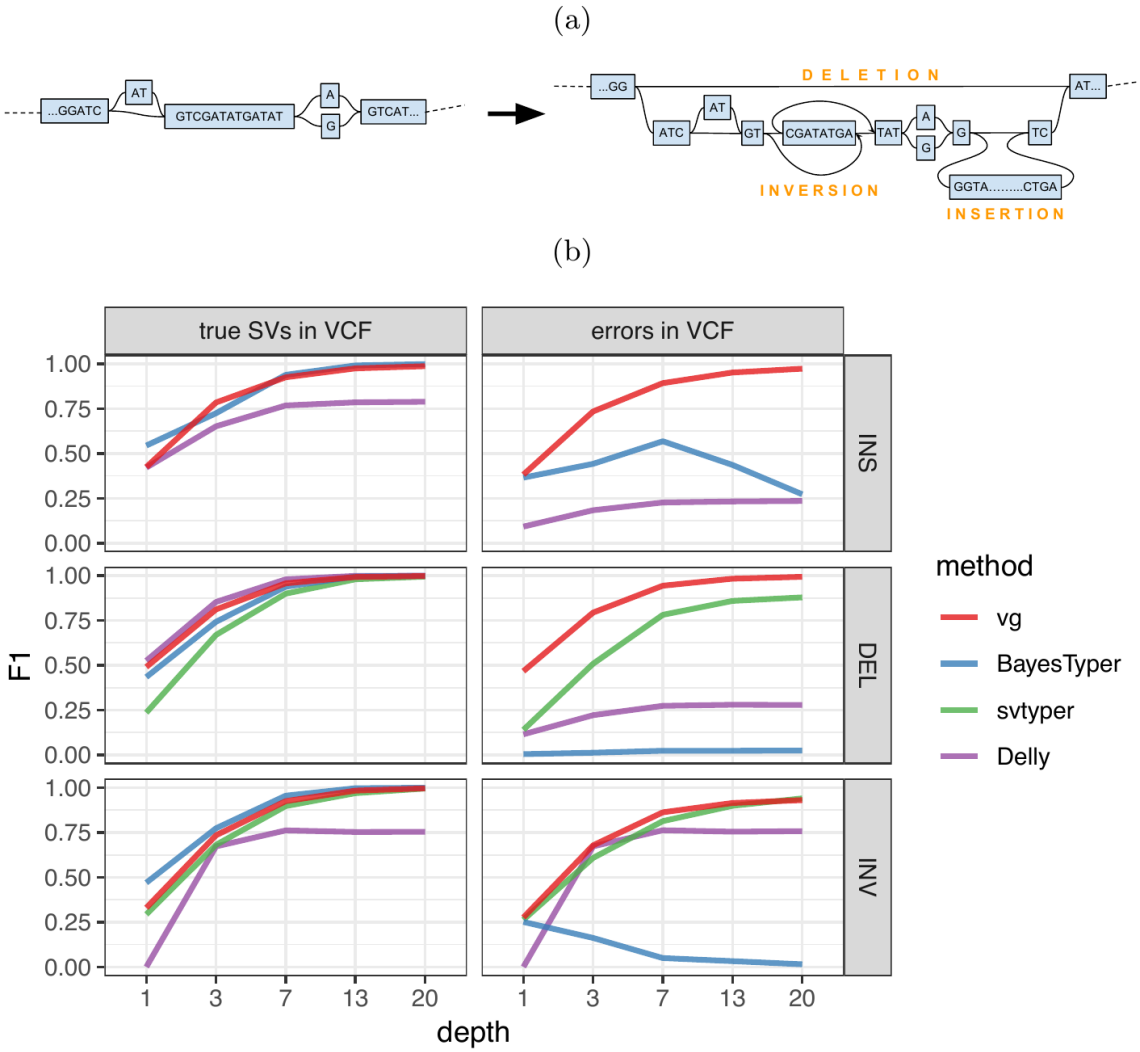

Figure 1: **Structural variation in vg.** a) Adding large deletions and insertions in a variation graph. b) Simulation experiment. For each experiment (method, depth and input VCF with/without errors), the maximum F1 was picked when using different quality thresholds, and is reported on the y-axis.

HGSVC dataset

The Human Genome Structural Variation Consoritum (HGSVC) generated a high-quality SV catalog of three samples, obtained using a consensus from different sequencing, phasing and variant caling technologies[18]. The three samples come from different human populations: a han Chinese individual (HG00514), a Puerto-Rican individual (HG00733), and a Yoruban Nigerian individual (NA19240). These SVs were used to construct a graph with vg and as input for the other genotypers. SVs were genotyped from short reads and compared with the original catalog (see Methods).

First, by simulating reads for HG00514, we compared the different methods in the ideal situation where the SV catalog is correct and matches exactly the SVs supported by the reads. While vg outperformed Delly and SVTyper, BayesTyper showed the best F1 score and precision-recall trade-

off (Figures 2 and S1, Table S1). When restricting the comparisons to regions not identified as tandem repeats or segmental duplications, the genotyping predictions were significantly better for all methods, with vg almost as good as BayesTyper on deletions (F1 of 0.944 vs 0.955). We observed similar results when evaluating the absence/presence of a SV instead of the exact genotype (Figures 2 and S2). Overall, both graph-based methods, vg and BayesTyper, outperformed the two other methods tested.

We then repeated the analysis using real Illumina reads from HG00514, to benchmark the methods on a more realistic experiment. Here vg clearly outperformed other approach, most likely because of its graph-based strategy and robustness to errors in the SV catalog (Figures 2 and S3). In non-repeat regions and across the whole genome, the F1 scores and precision-recall curves were higher for vg compared to other methods. For example, for deletions in non-repeat regions, the F1 score for vg was 0.801 while the second best method, Delly, had a F1 score of 0.692. We observed similar results when evaluating the absence/presence of a SV instead of the exact genotype (Figures 2 and S4).

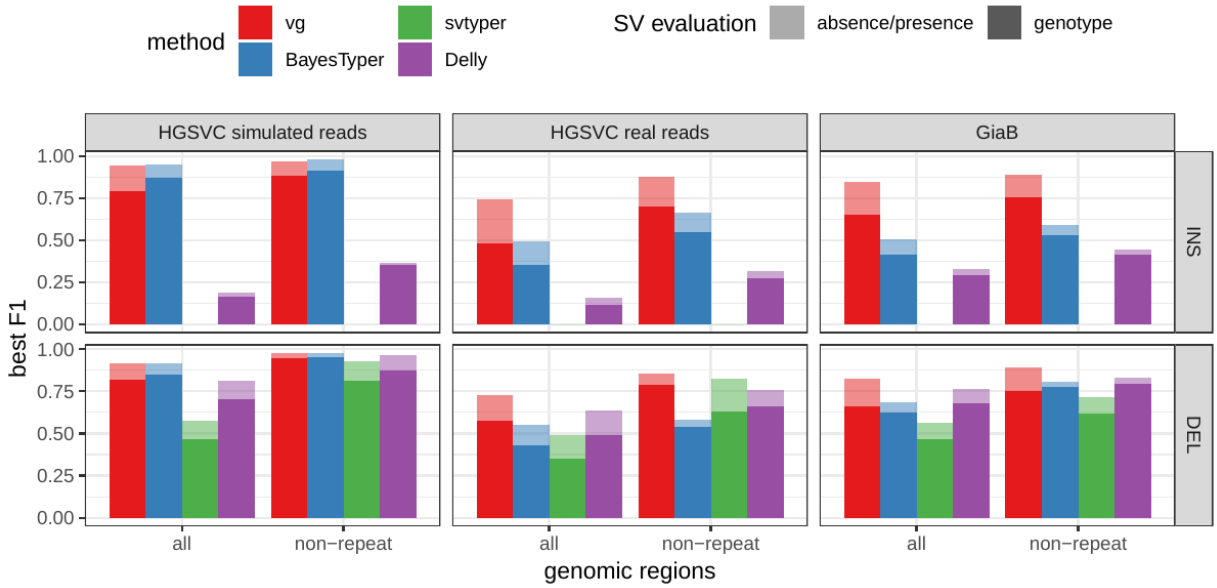

Figure 2: **Structural variants from the HGSVC and Genome in a Bottle datasets**. HGSVC: Simulated and real reads were used to genotype SVs and compared with the high-quality calls from Chaisson et al.[18]. Reads were simulated from the HG00514 individual. Using real reads, the three HG00514, HG00733, and NA19240 individuals were tested. GiaB: Real reads from the HG002 individual were used to genotype SVs and compared with the high-quality calls from the Genome in a Bottle consortium.[19,20]. Maximum F1 score for each method (color), across the whole genome or focusing on non-repeat regions (x-axis). The calling and genotyping evaluation are shown with different shades.

Other long-read datasets

The Genome in a Bottle (GiaB) consortium is currently producing a high-quality SV catalog for a Ashkenazim individual (HG002)[19,20]. Dozens of SV callers and datasets from short, long and linked reads were used to produce this set of SVs. vg performed similarly on this dataset than in the HGSVC dataset, with a F1 score of 0.75 for both insertions and deletions in non-repeat regions (Figures 2, S5 and S6, and Table S2). As before, other methods produced lower F1 scores in most cases, although Delly and BayesTyper predicted better genotype for deletions in non-repeat regions.

A recent study by Audano et al. generated a SV catalog using long-read sequencing across 15 individuals [8]. These variants were then genotyped from short reads across 440 individuals using SMRT-SV2, a machine-learning genotyper implemented for this study. We first called SVs from the pseudo-diploid genome and reads used to train SMRT-SV2 and constructed by merging datasets from two haploid cell lines[8]. The absence/presence predictions from vg were systematically better than SMRT-SV2 for both SV types but SMRT-SV2 produced better genotypes for deletions (see Figures 3, S7 and S10, and Table S3). Using publicly available Illumina reads, we then genotyped SVs in 3 of the 15 individuals that were used for discovery in Audano et al.[8]. Compared to SMRT-SV2, vg had a better precision-recall curve and a higher F1 for both insertions and deletions (Figures 3 and S11, and Table S4). SMR-SV2 produce *no-calls* when the read coverage is too low. We note that SMRT-SV2's recall increased when filtering these calls out. Interestingly, vg performed well even in regions where SMRT-SV2 produced *no-calls* (Figure S11 and Table S5). Finally, Audano et al. had identified 217 sequence-resolved inversions. vg correctly predicted the presence of around 14% of the inversions present in the three samples (Table S4). Inversions are often complex, harboring additional variation that makes their characterization and genotyping challenging.

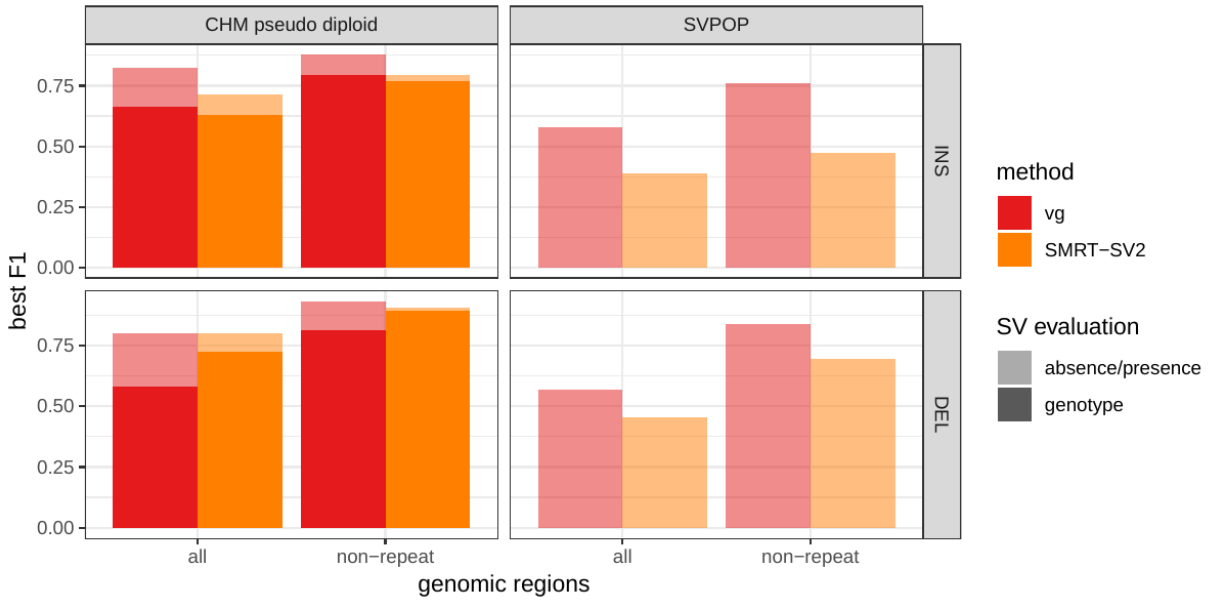

Figure 3: **Structural variants from Audano et al.[8]**. The pseudo-diploid genome built from CHM cell lines was used originally used to train SMRT-SV2 in Audano et al.[8]. The SVPOP panel shows the combined results for the HG5014, HG00733 and NA19240 individuals, 3 of the 15 individuals used to generate the high-quality SV catalog in Audano et al.[8]. Maximum F1 score for each method (color), across the whole genome or focusing on non-repeat regions (x-axis). The calling and genotyping evaluation are shown with different shades.

Breakpoint fine-tuning

Maybe better in discussion.

In addition to genotyping, vg can be used in “call” mode and use an augmentation step to modify the graph based on the read alignment. On the simulated SVs from Figure 1b, this approach was able to correct the many errors in the input VCF. The breakpoints were accurately fine-tuned for 93.8% of the insertions (Figure S12a and Table S5). For deletions, 78.1% of the variants were corrected when only one breakpoint had an error. In situations where both breakpoints of the deletions were incorrect, only 18.6% were correctly fine-tuned, and only when the amount of error was small (Figure S12b). The breakpoints of less than 20% of the inversions could be corrected. Across all SV types, the size of the variant didn’t affect the ability to fine-tune the breakpoints throught graph augmentation (Figure S12c).

Genotyping SV using vg and de novo assemblies

We investigated whether genome graphs derived from de-novo assembly alignments yield advantages for SV genotyping. To this end, we analyzed public sequencing datasets for 12 yeast strains from two related clades (S. cerevisiae and S. paradoxus) [21]. By generating genome graphs from only five of the strains we could measure how well variation from a small subset of strains represents the variation present in the full set of 12 strains. We generated and compared

two different types of genome graphs. The first graph type (in the following called *construct graph*) was created from a linear reference genome of the S.c. S288C strain and a set of SVs relative to this reference strain in VCF format. We compiled the SV set using the output of three methods for SV detection from genome assemblies: Assemblytics [22], AsmVar [23] and paftools [24]. All three methods were run to detect SVs between the reference strain S.c. S288C and each of the other strains. Merging the results from the three methods and four of the eleven strains provided us with a representative set of SVs occuring in the two yeast clades that we we could use to construct the *construct graph*. The second graph (in the following called *cactus graph*) was derived from a multiple genome alignment of the five strains using our Cactus tool [25]. While the *construct graph* is still mainly linear and highly dependent on the reference genome, the cactus graph is completely unbiased in that regard.

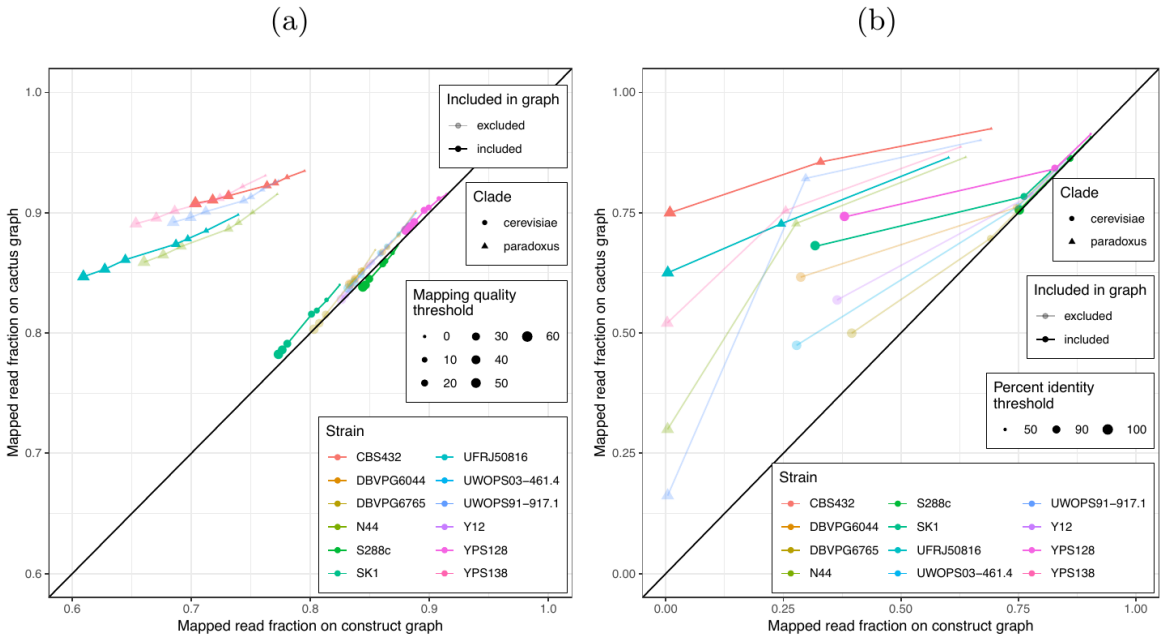

Figure 4: **Mapping comparison.** Short reads from all 12 yeast strains were aligned to both graphs. The fraction of reads mapped to the cactus graph (y-axis) and the construct graph (x-axis) are compared. a) Stratified by mapping quality threshold. b) Stratified by percent identity threshold. Colors and shapes represent the 12 strains and two clades, respectively. Transparency indicates whether the strain was included or excluded in the graphs.

In a first step, we tested our hypothesis that the *cactus graph* has higher mappability due to its better representation of sequence diversity among the yeast strains. Figure 4a shows the fraction of Illumina reads from the 12 strains that was mapped with a mapping quality above a certain threshold to the *cactus graph* and to the *construct graph*. Generally, more reads were mapped to the *cactus graph* than to the *construct graph* regardless of the chosen mapping quality threshold. Only for the reference strain S.c. S288C, the *construct graph* exhibited slightly better mappability. This suggests that not the higher sequence content in the *cactus graph* alone (15.4 Mb compared to 12.4 Mb in the *construct graph*) drives the improvement in mappability. Instead, our

measurements suggest that genetic distance to the reference strain increases the advantage of the *cactus graph* over the *construct graph*. Consequently. the benefit of the *cactus graph* is largest for strains in the S. paradoxus clade and smaller for reads from strains in the S. cerevisiae clade.

When we explored the mapping identity of the short reads on the graphs, we observed a similar trend (see Figure 4b). For strains in the S. paradoxus clade, the *cactus graph* enabled substantially more mappings with high percent identity than the *construct graph*. With strains in the S. cerevisiae clade, the difference was smaller, at least for a percent identity threshold up to 90%. When comparing read fractions with perfect identity (i.e. percent identity threshold = 100%), the *cactus graph* clearly outperforms the *construct graph* on 11 out of 12 samples (the exeption again being the reference strain S288C).

Interestingly, our measurements did not show a subtantial difference between strains included in the graph and excluded strains. The results suggest that two strains from each clade as well as the reference strain are sufficient to capture most of the genetic variation among all the strains. Only the number of alignments with perfect identity is substantially lower for the strains that were not included in the creation of the graphs (see Figure 4b). For a direct comparison, see Figure S13 which shows results of the same experiment on graphs generated from all 12 strains.

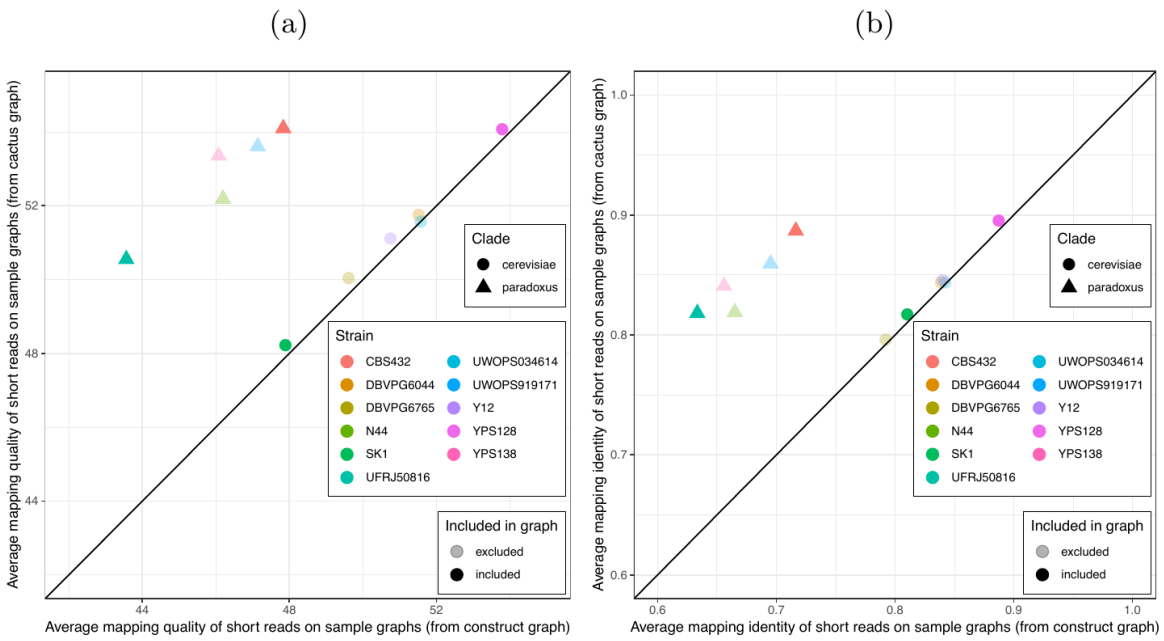

Figure 5: **SV genotyping comparison.** Short reads from all 11 non-reference yeast strains were used to genotype SVs contained in both graphs. Subsequently, sample graphs were generated from the resulting SV callsets. The short reads were again aligned to the sample graphs and the quality of the alignments was used to ascertain genotyping performance. a) Average mapping quality of short reads aligned to the sample graphs derived from *cactus graph* (y-axis) and *construct graph* (x-axis). b) Average mapping identity of short reads aligned to the sample graphs derived from *cactus graph* (y-axis) and *construct graph* (x-axis). Colors and shapes represent the 11 non-reference strains and two clades, respectively. Transparency indicates whether the strain was included or excluded in the graphs.

Next, we compared the SV genotype performance of both graphs. We mapped short reads from the 11 non-reference strains to both graphs and called variants for each strain using the vg toolkit's variant calling module. To evalute the callsets from both graphs, we generated a sample graph for each callset using the reference genome and the callset. Each sample graph is a graph representation of the respective callset. If a given callset is correct, we would expect that reads from the same sample can be mapped confidently and with high identity to the corresponding sample graph. Therefore, we compared the average mapping quality and percent identity of the short reads on both types of sample graphs (see Figures 5a and b). Similar to the results of our mapping analysis above, the *cactus graph* clearly outperformed the *construct graph* for strains in the S. paradoxus clade and performed slightly better for strains in the S. cerevisiae clade. Again, our measurements did not show a large difference between strains included in the graph and those that were excluded. For a direct comparison, see Figure S14 which shows results of the same experiment on graphs generated from all 12 strains.

Methods

Toil-vg

Toil-vg is a set of Python scripts for simplifying vg tasks such as graph construction, read mapping and SV genotyping. It uses the Toil workflow engine [26] to seamlessly run pipelines locally, on clusters or on the cloud. All variation graph analysis in this report was done using toil-vg, with the exact commands available here [todo: get everything in one place with reasonable docs]. The principal toil-vg commands used are described below.

Toil-vg construct

Toil-vg construct automates graph construction and indexing following the best practices put forth by the vg community. Graph construction is parallelized across different sequences from the reference fasta, and different whole-genome indexes are created side by side when possible. Phasing information from the input VCF can be used when available to preserve haplotypes in the GCSA2 pruning step, as well as to extract haploid sequences to simulate from.

Toil-vg map

Toil-vg map splits the input reads into batches, maps each batch in parallel, then merges the result.

Toil-vg call

A simple though very general variant caller has been implemented as vg call. Here it is used to genotype structural variants already present in the graph, but the same algorithm can also be used for smaller variants such as SNPS, as well as making de-novo calls. The algorithm is as follows:

1. The average read support for each node and edge, adjusted for mapping and base quality, is computed. The graph can optionally be augmented to include new variation from the reads.
2. The graph is then decomposed into snarls. Briefly, a snarl is a subgraph defined by two end nodes, where cutting the graph at these nodes disconnects the snarl from the rest of the graph. (todo: work on way to define this without getting into bidrected graphs, or is that a lost cause?). Snarls can be nested inside other snarls, and this nesting hierarchy forms a forest (todo: I don't think chains get used anywhere in vg call so we can ignore here). The snarl decomposition is a fundamental structure for identifying variants in a graph and are formally defined, along with an algorithm to identify them, in [27].
3. Root-level snarls from the decomposition are considered independently and in parallel. Only snarls whose two ends lie on a reference (ie chromosome) path are considered as the VCF format used for ouptut requires reference positions. The following steps are performed on each root snarl.

4. A set of paths between the snarls end nodes are computed. (todo, consult with Adam about writing up the RepresentativeTraversalFinder)
5. The paths are ranked according to their average support.
6. A genotype is determined using the relative support of the best traversals, as well as the background read depth.
7. The VCF variants are derived from the paths. (todo: expand and clarify these last steps. could leave them fairly brief here and go into detail in the supplement).

Due to the high memory requirements of the current implementation of vg call, toil-vg call splits the input graph into 2.5Mb overlapping chunks along the reference path. Each chunk is called independently in parallel and the results are concatenated into the output VCF.

Toil-vg sveval

The variants are first normalized with `bcftools norm` to ensure consistent representation between called variants and baseline variants. We then implemented an overlap-based strategy to compare SVs and compute evaluation metrics (sveval R package: https://github.com/jmonlong/sveval).

For deletions and inversions, the affected region in the reference genome is overlapped and matched between the two sets od SVs. First, we select pairs of SVs with at least 10% reciprocal overlap. Then for each variant we compute the proportion of its region that is covered by an overlapping variant in the other set. If this coverage proportion is higher than 80%, the variant is considered *covered*. True positives are covered variants from the call set or the truth set. False positives are variants from the call set that are not covered. False negative are variants from the truth set that are not covered.

For insertions, we select pairs of insertions that are located no farther than XX bp from each other. We then align the inserted sequences using a Smith-Waterman alignment. Then for each insertion we compute the proportion of its inserted sequence that aligns a matched variant in the other set. As for deletions/inversions, this coverage proportion is used to annotate variants as true positives, false positives and false negatives.

sveval accepts VCF files with symbolic or explicit representation of the SVs. If the explicit representation is used, multi-allelic variants are split and their sequences right-trimmed. When inversions are considered, the reverse-complement of the ALT sequence of variants larger than 10 bp is aligned with the REF sequence and classified as an inversion if more than 80% of the sequence aligns.

We assess either the *calling* performance (absence/presence of a SV) or the *genotyping* performance. For the *calling* evaluation, both heterozygous and homozygous alternate SVs are compared using the approach described above. To compute genotype-level metrics, the heterozygous and homozygous SVs are compared separately. Before splitting the variants by

genotype, consecutive heterozygous variants are first stitched together if located at less that 20 bp from each other, and then merged into homozygous variants if their reciprocal overlap is at least 80%.

Other SV genotypers

BayesTyper

If not specified otherwise BayesTyper was run as follow. Raw reads were mapped to the reference genome using `bwa mem`. GATK and Platypus were run on the mapped reads to call SNVs and short indels (<50bp). The VCFs with these variants and the SVs to genotype were combined and used as input for BayesTyper. The BayesTyper genotyping pipeline started by counting the kmers in the raw reads using `kmc -k55 -ci1 -m8`. These kmers are then passed through a Bloom filter using `bayesTyperTools makeBloom`. Finally variants are clustered and genotyped using `bayestyper cluster` and `bayestyper genotype --min-genotype-posterior 0`

Delly

The `delly call` command was run on the reads mapped by `bwa mem`, the reference genome fasta file and the VCF containing the SVs to genotype in their explicit representation.

SVTyper

The VCF containing deletions was converted to symbolic representation and passed to `svtyper` as long as the reads mapped by `bwa mem`. The output VCF was converted back to explicit representation, to facilitate variant normalization later, using `bayesTyperTools convertAllele`.

SMRT-SV2

SMRT-SV2 was run on VCFs generated for SMRT-SV2 using the command: `XXX` The output VCF was converted back to explicit representation, to facilitate variant normalization later??

Simulation experiment

We simulated a synthetic genome with 1000 insertions, deletions and inversions. Each variant was separated from the next by a buffer region of 500 bp. The size of deletions and insertions followed the distribution of real SVs from the HGSVC catalog. We used the same size distribution as deletions for inversions. Then a VCF file was produced for three simulated samples with random genotypes (homozygous reference, heterozygous, homozygous alternate).

We created another VCF file containing errors in the SV definition. One or both breakpoints of deletions were shifted between 1 and 10 bp. The location and sequence of insertions were also modified, either shifted or deleted at the flanks, again up to 10 bp.

Paired-end reads were simulated using `vg sim` on graph that contained the true SVs. Different read depth were tested: 1x, 3x, 7x, 10x, 13x, 20x. We used real Illumina reads from NA12878 provided by the Genome in a Bottle consortium to model base qualities and sequencing errors.

The different methods were tested using either the true VCF or the VCF that contained errors. For vg, a graph was constructed from the VCF file, indexed, then used to map simulated reads and call variants using toil-vg (see toil-vg). A beta version 1.5 of BayesTyper was run directly on the simulared reads and using only SVs in the input VCF. In order to run the other methods, reads were mapped to the linear reference sequence using `bwa mem` and sorted using `samtools`. For Delly, insertions and deletions were first genotyped together using these mapped reads and the `delly call` command. Inversions were genotyped separately using a VCF that was formatted according to Delly's preference. SVTyper was run on the mapped reads and a VCF that was converted to symbolic variant representation. All commands used for this analysis are available at ANALYSIS_REPO.

The genotypes called in each experiment (method/VCF with or without errors/sequencing depth) were compared to the true SV genotypes to compute the precision, recall and F1 score (see toil-vg sveval).

Breakpoint fine-tuning using graph augmentation

vg can call variants after augmenting the graph with the read alignments to discover new variants (see Toil-vg call). We tested if this approach could fine-tune the breakpoint location of SVs in the graph. We started with the graph that contained approximate SVs (1-10 bp errors in breakpoint location) and 20x simulated reads from the simulation experiment (see Simulation experiment). The variants called after graph augmentation were compared with the true SVs and considered fine-tuned if the breakpoints matched exactly.

HGSVC Analysis

Phased VCFs were obtained for the three HGSVC samples from the authors of [18] and combined with bcftools merge. A variation graph was created and indexed using the combined VCF and the HS38D1 reference with alt loci excluded. The phasing information was used to construct a GWBT index, from which the two haploid sequences from HG00514 were extracted. Illumina read pairs with 30x coverage were simulated from these sequences using vg, with an error model learned from real reads from the same sample. Still, these reads reflect the idealized situation where the breakpoints of the SVs being genotyped are exactly known a priori. The reads were mapped to the graph and the mappings used to genotype the SVs in the graph, which were finally compared back to the HG00514 genotypes from the HGSVC VCF. The process was repeated with the same reads on the linear reference, using bwa-mem for mapping and DELLY, SVTYPER and Bayestyper for SV genotyping.

Illumina HiSeq 2500 paired end reads were downloaded from the EBI's ENA FTP site for the three samples, using Run Accessions ERR903030, ERR895347 and ERR894724 for HG00514, HG00733 and NA19240, respectively. The graph and linear mapping and genotyping pipelines were run exactly as for the simulation, and the comparison results were aggregated across the three samples.

GIAB Analysis

Version 0.6 of the GIAB SV VCF for the Ashkenazim son (HG002) was obtained from the NCBI FTP site. Illumina reads downsampled to 50x coverage obtained as described in [12], were used to run the vg and linear SV genotyping pipelines described above though with GRCh37 instead of 38. Since this dataset contains only one sample, variants without a determined genotype (14649 out of 74012) were considered "false positives" as a proxy measure for precision.

Todo: I think some more thought may need to go into this comparison. I'd actually assumed before writing this up that the whole trio was in the VCF but this is not the case – it's just for the one sample. The curreny results will count as a false positive a call that was not assigned a genotype by GIAB. This is probably a pretty good estimate, but I think it's too hand wavy to publish. In the worst case, we can switch to just looking at recall on the whole set (which has the advantage of being exactly what that other paper did for deletions).

SMRT-SV2 Comparison (CHMPD and SVPOP)

The SMRT-SV2 genotyper can only be used to genotype VCFs that were created by SMRT-SV2, and therefore could not be run on our simulated, HGSVC or GIAB data. The authors shared their training and evaluation set, a pseudodiploid sample constructed from combining the haploid CHM1 and CHM13 samples, along with a negative control (NA19240). The high quality of the CHM assemblies makes this set an attractive alternative to using simulated reads. We used this two-sample pseudodiploid VCF along with the 30X read set to construct, map and genotype with vg, and also ran SMRT-SV2 genotyper with the "30x-4" model and min-call-depth 8 cutoff, and compared the two back to the original VCF.

In an effort to extend this comparison to a more realistic setting, we reran the three HGSVC samples against the SMRT-SV2 discovery VCF (which contains them in addition to 12 other samples) published in [8] using vg and SMRT-SV2 Genotyper. The discovery VCF does not contain genotypes so we did not distinguish between heterozygous and homozygous genotypes, looking at only the presence or absence of an alt allele in each variant.

SMRT-SV2 produce some explicit *no-calls* prediction when the read coverage is too low to produce accurate genotypes. These no-calls are considered homozygous reference in the main evaluation. We also explored the performance of vg and SMRT-SV2 in different sets of regions:

1. Non-repeat regions, i.e. excluding segmental duplications and tandem repeats.
2. Repeat regions defined as segmental duplications and tandem repeats.
3. Regions where SMRT-SV2 could call variants.
4. Regions where SMRT-SV2 produced no-calls.

Discussion

Potential topics for the discussion.

Performance across datasets

Although vg was overall the best genotyper in our benchmarks, other methods were superior in some datasets and some situations. Some of these differences might be explained by the quality of the input SV catalog. The GiaB catalog is more curated and, specifically for deletions in non-repeat regions, Delly and BayesTyper were better at predicting genotypes compared to vg. This might be because the breakpoint resolution for this type of SV in these regions is better in this dataset compared to the HGSVC dataset which was derived mostly from long-read sequencing. Similarly, SMRT-SV2 performs better for deletions in the pseudo-diploid genome constructed from two high quality genome assembly of CHM cell lines.

Providing a resource to be used by large-scale sequencing project

As a result of this study we provide a variation graph containing XX millions of SNVs and indels from the 1000 Genomes Project as well as XX thousands of SVs derived from long-read sequencing. This variation graph could serve as a richer reference for large scale projects that use short-read sequencing. For instance, more and more large-scale projects are sequencing the genomes of thousands or hundreds of thousands of individuals, e.g the Pancancer Analysis of Whole Genomes, the Genomics England initiative, and the TOPMed consortium(REFS). These large WGS studies will provide a deeper look into the mechanism of common diseases and, in some cases, will be used directly in a clinical setting. Clinicians and researchers are eager to use these growing WGS resources to interrogate the importance of SVs in disease at a scale never achieved before, either to get a more complete picture of the genetic factors of a disease or to produce a more comprehensive clinical report. As sequencing reaches the clinic, whole-genome sequencing will become routine for many patients. Clinicians will rely on variant calling and interpretation for diagnosis and treatment. For variant interpretation in particular, a comprehensive and unified characterization of the genomic variation will be extremely valuable.

Easier to use

Some methods require additional information or special VCF formatting [28]. SVTyper was designed to use VCFs created by Lumpy. The genotyping module from Delly was implemented for variants found by its discovery module. SMRT-SV requires a VCF with information about XXX. Nebula, a new khmer-based genotyper, requires reads from a sample containing the SV during khmer selection[29]. In contrast, vg can take as input either explicit or symbolic VCFs, as well as assembly alignment.

Assemblies are the future

Our results suggest that constructing a graph from de novo assembly alignment is more representative of the sequencing reads and leads to better SV genotyping. De novo assemblies for human are becoming more and more common, for example from optimized mate-pair libraries[30] or long-read sequencing[30]. For an optimal representation of the genomic variation, we expect the future graphs to include information from the alignment of numerous de novo assemblies. Aligning assembled contigs to existing variation graphs, like to ones created from SVs catalogs, is still experimental but could generate a genome graph augmented with both existing variant databases and new high-quality assemblies.

Future improvements in vg

The vg toolkit is in active development. Read mapping is an area of constant improvement, both in term of computational efficiency and accuracy. For example, haplotype information can be modeled in variation graph and, in the future, assist read mapping and variant calling. These upcoming developments will directly benefit SV genotyping with vg.

Limitations

Copy number variants (CNVs) are currently represented as deletions or insertions. For this reason duplications are represented as additional sequence rather than encoded as a loop in the graph. While this is sufficient to represent single copy changes, such as deletions or single tandem duplications, CNVs with multi-copies states are not addressed by the current implementation. The genotyping algorithm would need to be extended to model copy number in order to assess these variants.

Near-breakpoint resolution is necessary. Simulations have shown that SV genotyping with vg is robust to errors up to 10 bp in breakpoint location.

The genotyping evaluation of invertions is limited by the lack of existing gold-standards. We showed that vg is capable of genotyping simple inversions from simulation or the few discovered in the SV catalog from Audano et al.[8]. However most inversions are complex and involve small

insertions/deletions around their breakpoints(REF). While these complex variants are difficult to represent in the VCF format, they would naturally be represented through the alignment of de novo assemblies. For example, in our experiment with yeast assemblies, we identified XX variants that can be considered complex inversion as they contain at least XX inverted bases.

Supplementary Material

Table S1: Genotyping evaluation on the HGSVC dataset. Precision, recall and F1 score for the call set with the best F1 score. The numbers in parenthesis corresponds to the results in non-repeat regions.

Experiment	Method	Type	Precision	Recall	F1
Simulated reads	vg	INS	0.795 (0.885)	0.796 (0.883)	0.795 (0.884)
		DEL	0.869 (0.971)	0.771 (0.92)	0.817 (0.945)
	BayesTyper	INS	0.91 (0.935)	0.835 (0.9)	0.871 (0.917)
		DEL	0.898 (0.981)	0.806 (0.929)	0.849 (0.954)
	svtyper	DEL	0.809 (0.876)	0.328 (0.754)	0.467 (0.81)
	Delly	INS	0.767 (0.866)	0.093 (0.225)	0.166 (0.358)
		DEL	0.696 (0.903)	0.707 (0.846)	0.701 (0.874)
Real reads	vg	INS	0.431 (0.683)	0.541 (0.726)	0.48 (0.704)
		DEL	0.65 (0.886)	0.519 (0.708)	0.577 (0.787)
	BayesTyper	INS	0.601 (0.747)	0.254 (0.433)	0.357 (0.549)
		DEL	0.627 (0.91)	0.325 (0.381)	0.428 (0.537)
	svtyper	DEL	0.661 (0.733)	0.236 (0.551)	0.348 (0.629)
	Delly	INS	0.516 (0.621)	0.068 (0.176)	0.12 (0.275)

Experiment	Method	Type	Precision	Recall	F1
		DEL	0.55 (0.838)	0.445 (0.547)	0.492 (0.662)

Table S2: Genotyping evaluation on the Genome in a Bottle dataset. Precision, recall and F1 score for the call set with the best F1 score. The numbers in parenthesis corresponds to the results in non-repeat regions.

Method	Type	Precision	Recall	F1
vg	INS	0.658 (0.774)	0.646 (0.735)	0.652 (0.754)
	DEL	0.68 (0.768)	0.643 (0.735)	0.661 (0.751)
BayesTyper	INS	0.776 (0.879)	0.286 (0.379)	0.418 (0.53)
	DEL	0.808 (0.886)	0.512 (0.696)	0.627 (0.779)
svtyper	DEL	0.742 (0.818)	0.342 (0.496)	0.468 (0.618)
Delly	INS	0.822 (0.894)	0.177 (0.268)	0.291 (0.412)
	DEL	0.722 (0.822)	0.645 (0.768)	0.681 (0.794)

Table S3: Genotyping evaluation on the pseudo-diploid genome built from CHM cell lines in Audano et al.[8].

Method	Region	Type	Precision	Recall	F1
vg	all	INS	0.665	0.661	0.663
		DEL	0.688	0.500	0.579
	non-repeat	INS	0.806	0.784	0.795
		DEL	0.869	0.762	0.812
SMRT-SV2	all	INS	0.757	0.536	0.628
		DEL	0.848	0.630	0.723
	non-repeat	INS	0.880	0.680	0.767
		DEL	0.971	0.824	0.891

Table S4: Calling evaluation on the SVPOP dataset. Combined results for the HG5014, HG00733 and NA19240 individuals, 3 of the 15 individuals used to generate the high-quality SV catalog in Audano et al.[8].

Method	Region	Type	TP	FP	FN	Precision	Recall	F1
vg	all	INS	25838	22042	15772	0.540	0.621	0.577
		DEL	14545	6824	15425	0.681	0.485	0.567
		INV	27	26	173	0.509	0.135	0.213
	non-repeat	INS	8051	3258	1817	0.712	0.816	0.760
		DEL	3769	623	818	0.858	0.822	0.840
		INV	19	12	75	0.613	0.202	0.304
SMRT-SV2	all	INS	16270	26031	25340	0.385	0.391	0.388
		DEL	11793	10106	18177	0.539	0.393	0.455
	non-repeat	INS	4483	4659	5385	0.490	0.454	0.472
		DEL	2928	930	1659	0.759	0.638	0.693

Table S5: Calling evaluation on the SVPOP dataset in different sets of regions for the HG5014 individual.

Method	Region	Type	TP	FP	FN	Precision	Recall	F1
vg	all	INS	8618	7237	5416	0.546	0.614	0.578
		DEL	4762	2048	5145	0.696	0.481	0.569
		INV	11	8	54	0.579	0.169	0.262
	repeat	INS	6176	6923	4678	0.475	0.569	0.518
		DEL	2428	1701	4542	0.584	0.348	0.436
		INV	1	1	6	0.500	0.143	0.222
	non-repeat	INS	2677	987	514	0.731	0.839	0.781
		DEL	1180	176	321	0.869	0.786	0.825
		INV	7	4	20	0.636	0.259	0.368
	called in SMRT-SV	INS	3410	3789	2108	0.478	0.618	0.539
		DEL	2544	1092	1518	0.699	0.626	0.661
		INV	8	8	52	0.500	0.133	0.210
	not called in SMRT-SV	INS	4838	542	3678	0.899	0.568	0.696
		DEL	2034	26	3723	0.987	0.353	0.520
SMRT-SV2	all	INS	5245	8563	8789	0.394	0.374	0.384
		DEL	3741	3382	6166	0.533	0.378	0.442
	repeat	INS	3848	7125	7006	0.368	0.354	0.361
		DEL	1990	2832	4980	0.426	0.286	0.342

Method	Region	Type	TP	FP	FN	Precision	Recall	F1
	non-repeat	INS	1396	1468	1795	0.493	0.438	0.464
		DEL	901	308	600	0.745	0.600	0.665
	called in SMRT-SV	INS	4343	5595	1175	0.445	0.787	0.569
		DEL	3227	2451	835	0.573	0.794	0.666
	not called in SMRT-SV	INS	116	109	8400	0.551	0.014	0.026
		DEL	206	16	5551	0.911	0.036	0.069

Table S5: Breakpoint fine-tuning using augmentation through “vg call”. For deletions and inversions, either one or both breakpoints were shifted to introduce errors in the input VCF. For insertions, the insertion location and sequence contained errors.

SV type	Error type	Breakpoint	Variant	Proportion	Mean size (bp)	Mean error (bp)
DEL	one end	incorrect	220	0.219	422.655	6.095
		fine-tuned	784	0.781	670.518	5.430
	both ends	incorrect	811	0.814	826.070	6.275
		fine-tuned	185	0.186	586.676	2.232
INS	location/ seq	incorrect	123	0.062	428.724	6.667
		fine-tuned	1877	0.938	440.043	6.439
INV	one end	incorrect	868	0.835	762.673	5.161
		fine-tuned	172	0.165	130.244	5.884
	both ends	incorrect	950	0.992	556.274	5.624
		fine-tuned	8	0.008	200.000	1.375

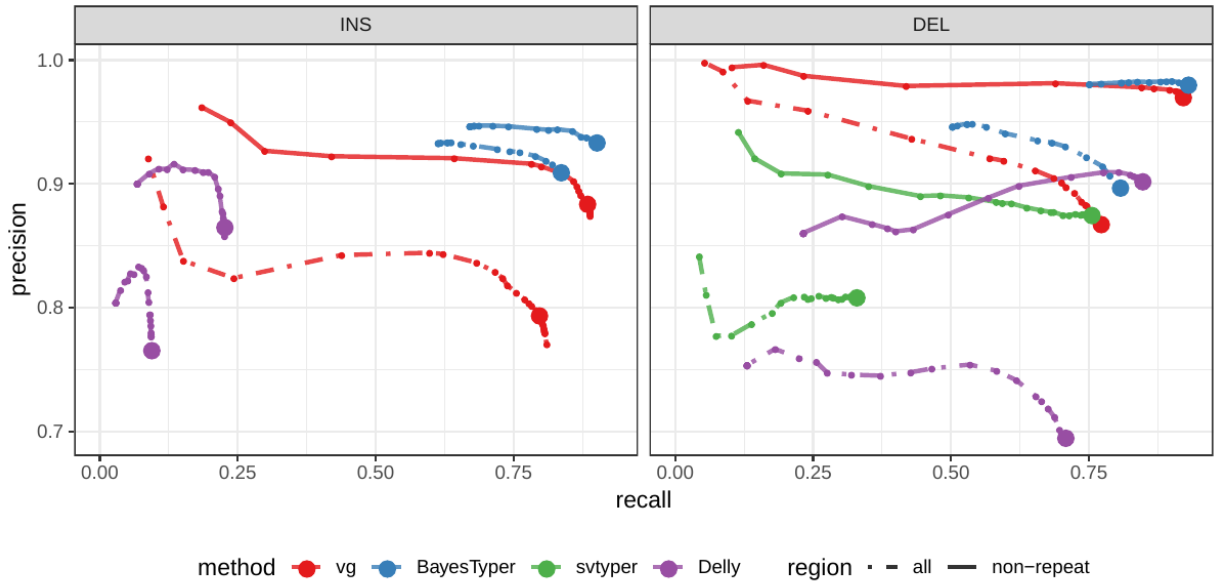

Figure S1: **Structural variants from the HGSVC dataset**. Genotyping evaluation for simulated reads.

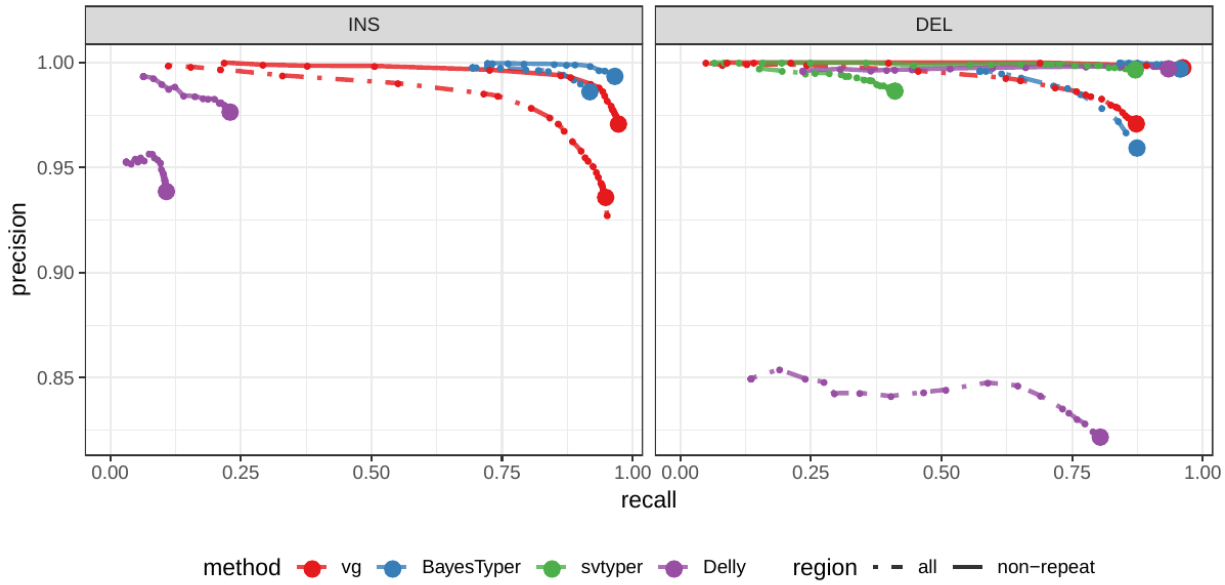

Figure S2: **Structural variants from the HGSVC dataset**. Calling evaluation for simulated reads.

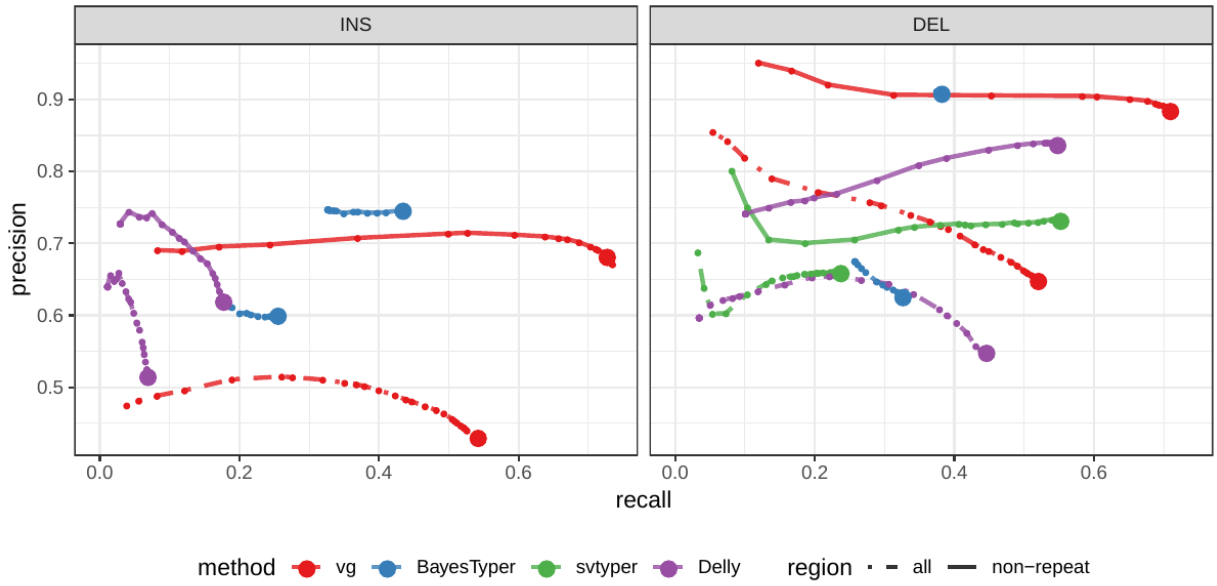

Figure S3: **Structural variants from the HGSVC dataset**. Genotyping evaluation for real reads.

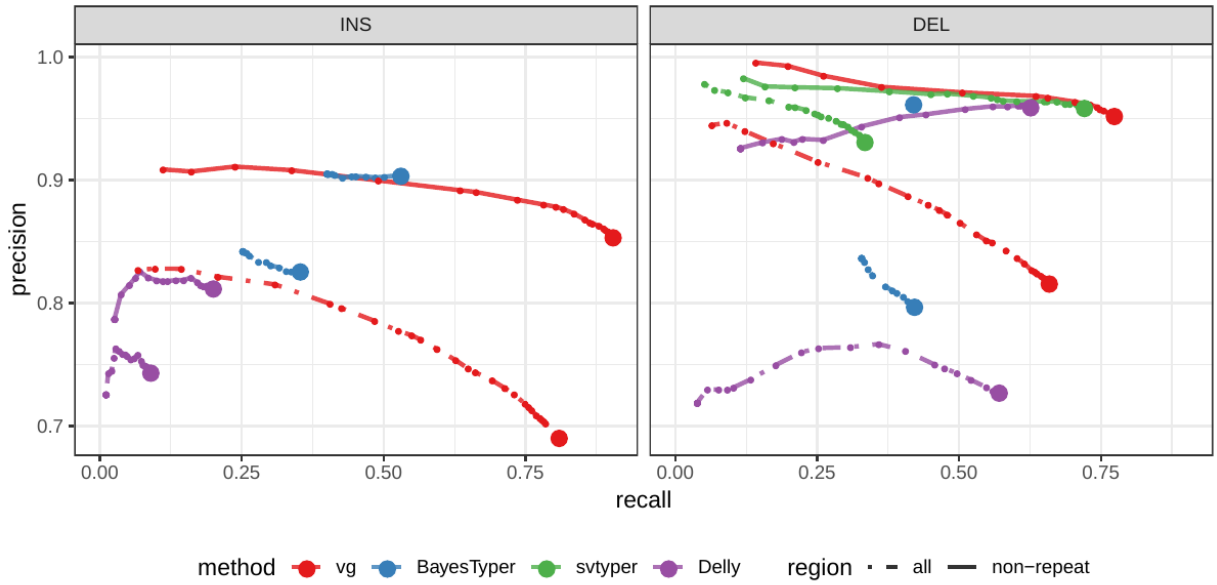

Figure S4: **Structural variants from the HGSVC dataset**. Calling evaluation for real reads.

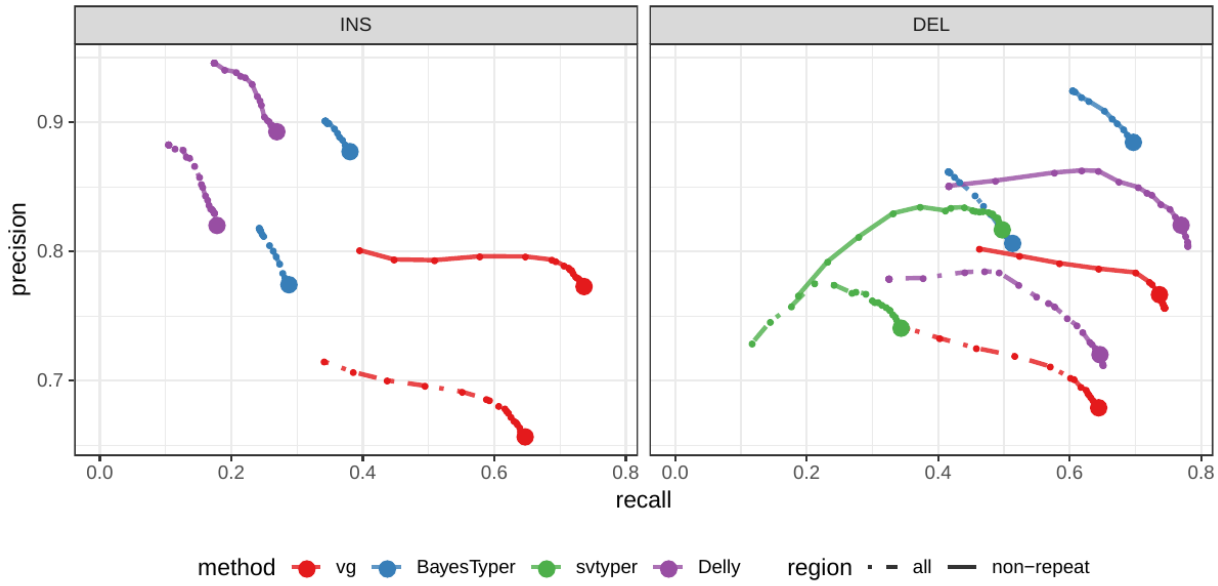

Figure S5: **Structural variants from the Genome in a Bottle dataset**. Genotyping evaluation.

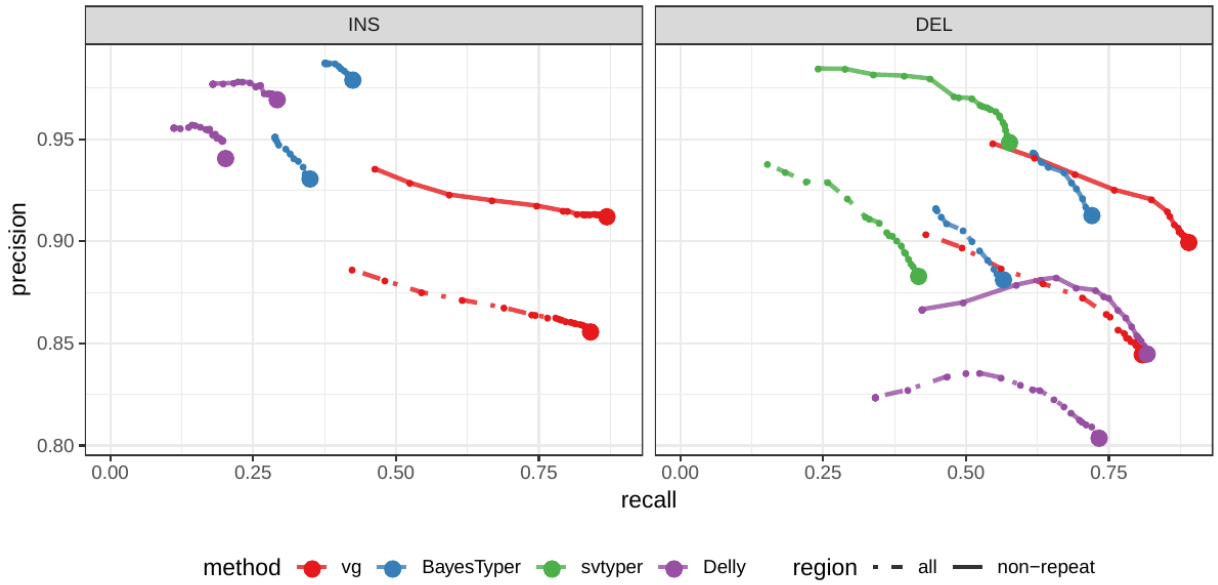

Figure S6: **Structural variants from the Genome in a Bottle dataset**. Genotyping evaluation.

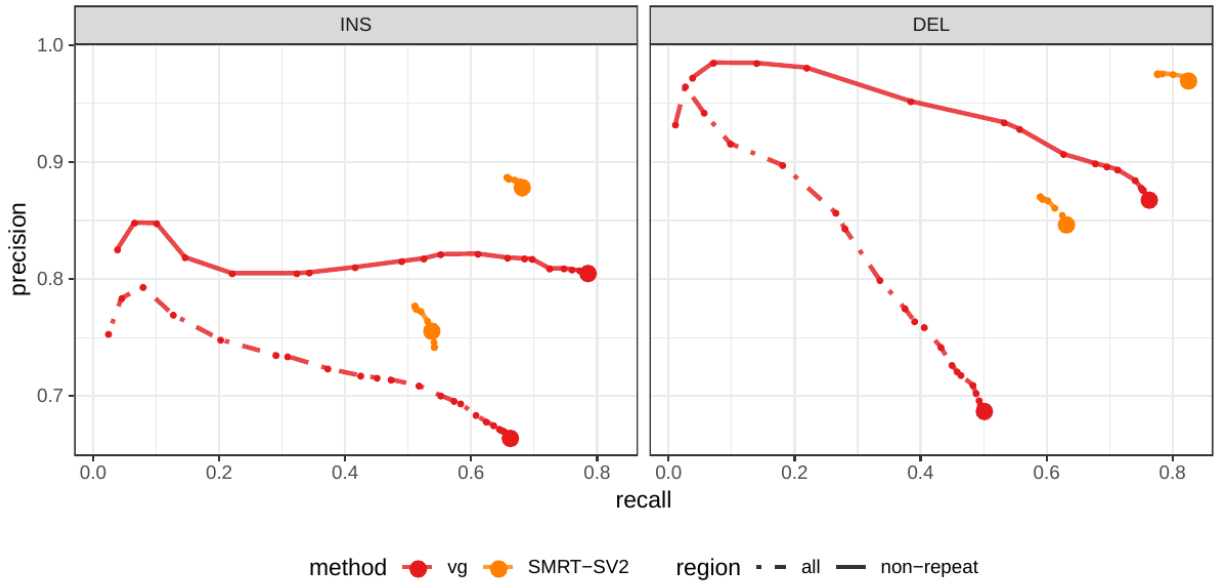

Figure S7: **Structural variants from the CHM pseudo-diploid dataset**. Genotyping evaluation.

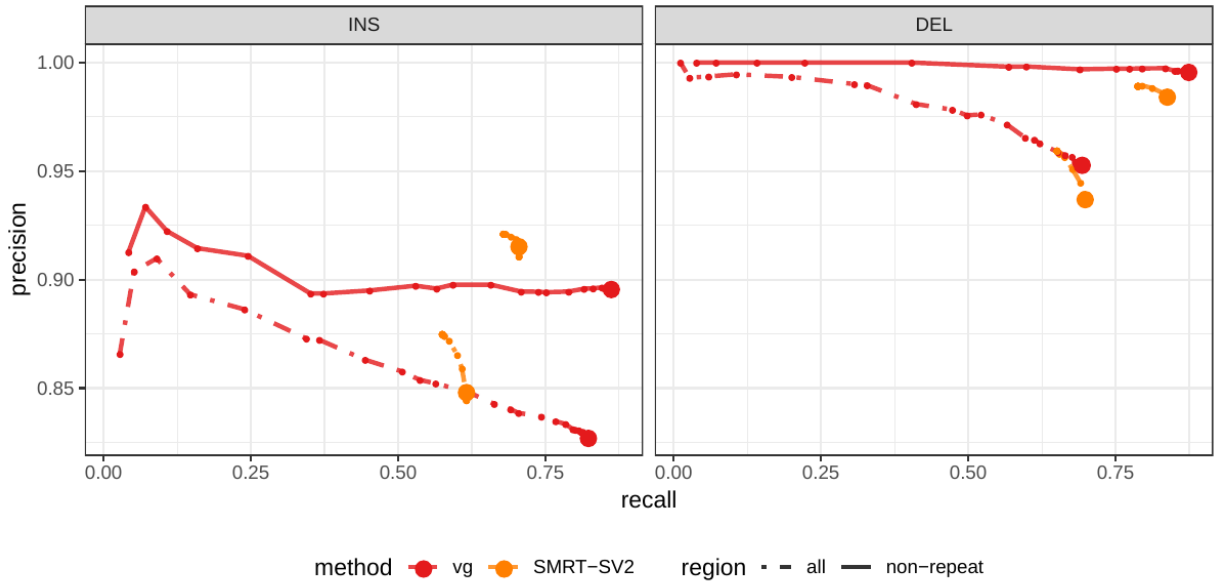

Figure S10: **Structural variants from the CHM pseudo-diploid dataset**. Calling evaluation.

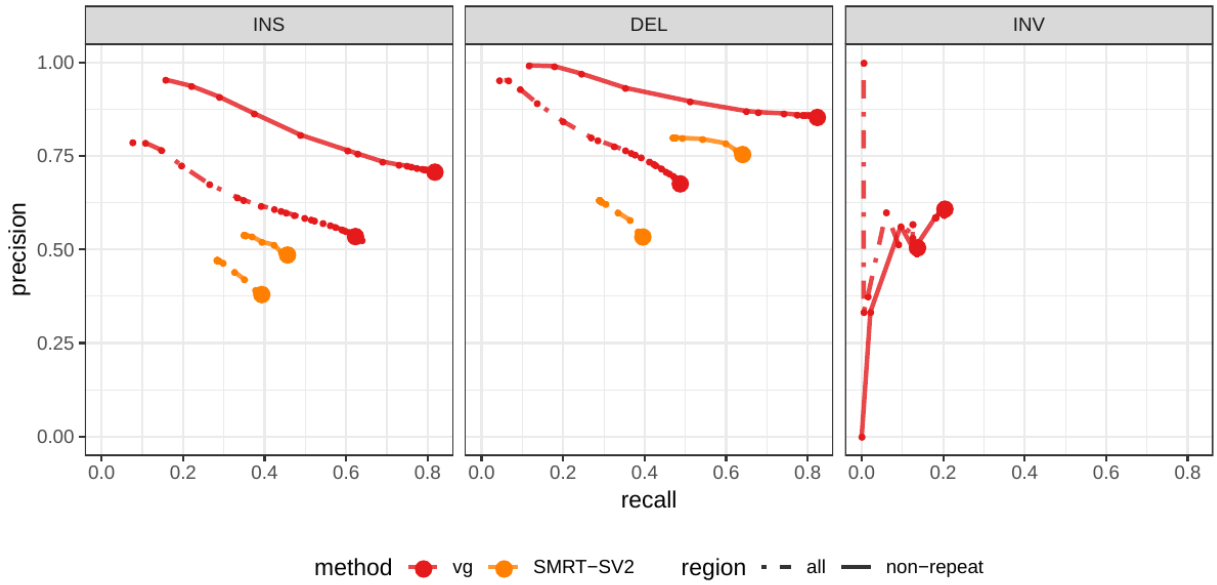

Figure S11: **Structural variants from the SVPOP dataset**. Calling evaluation.

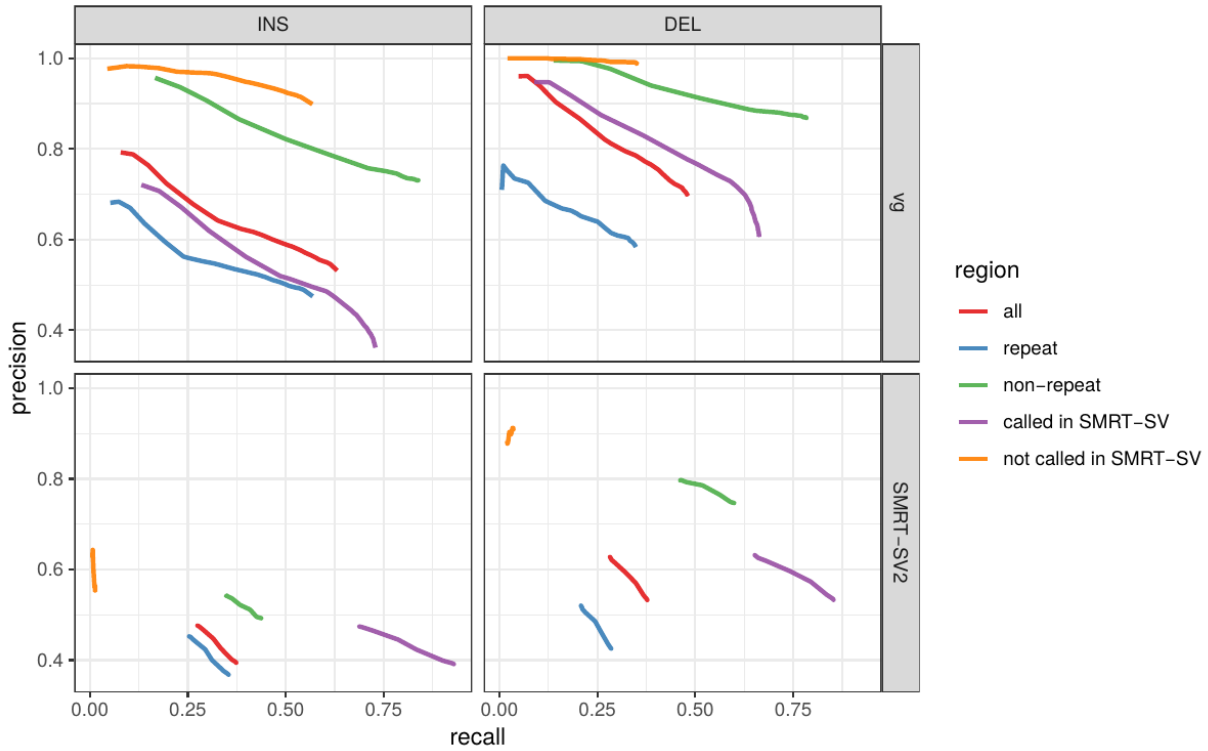

Figure S11: **Evaluation across different sets of regions in HG00514 (SVPOP dataset)**. Calling evaluation.

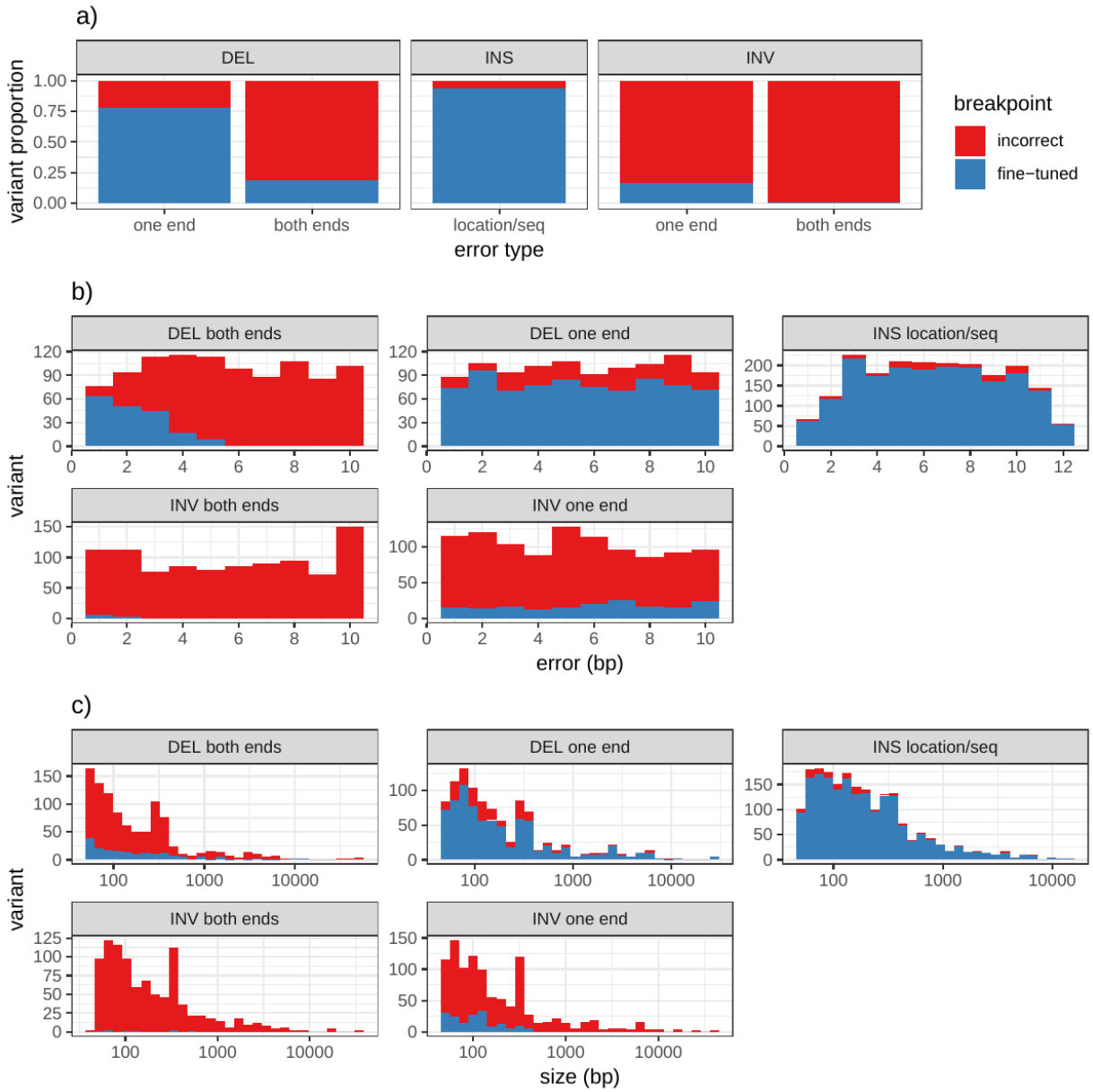

Figure S12: **Breakpoint fine-tuning using augmentation through "vg call".**. For deletions and inversions, either one or both breakpoints were shifted to introduce errors in the input VCF. For insertions, the insertion location and sequence contained errors. a) Proportion of variant for which breakpoints could be fine-tuned. b) Distribution of the amount of errors that could be corrected or not. c) Distribution of the size of the variants whose breakpoints could be fine-tuned or not.

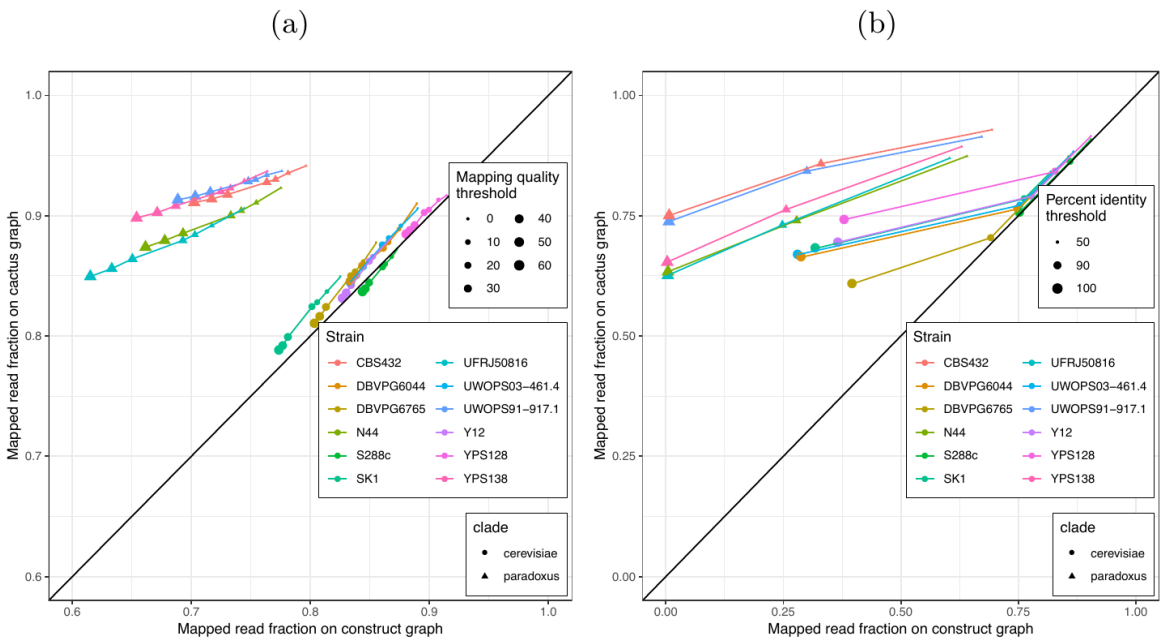

Figure S13: **Mapping comparison on graphs of all 12 strains.** Short reads from all 12 yeast strains were aligned to both graphs. The fraction of reads mapped to the cactus graph (y-axis) and the construct graph (x-axis) are compared. a) Stratified by mapping quality threshold. b) Stratified by percent identity threshold. Colors and shapes represent the 12 strains and two clades, respectively.

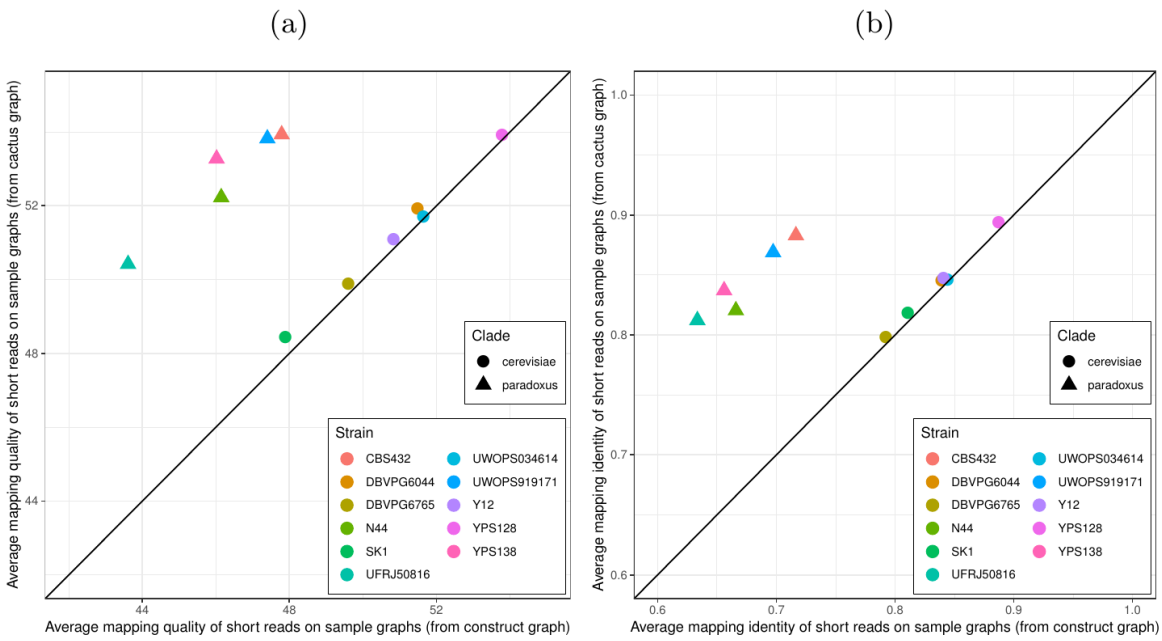

Figure S14: **SV genotyping comparison on graphs of all 12 strains.** Short reads from all 11 non-reference yeast strains were used to genotype SVs contained in both graphs. Subsequently, sample graphs were generated from the resulting SV callsets. The short reads were again aligned to the sample graphs and the quality of the alignments was used to ascertain genotyping performance. a) Average mapping quality of short reads aligned to the sample graphs derived from *cactus graph* (y-axis) and *construct graph* (x-axis). b) Average mapping identity of short reads aligned to the sample graphs derived from *cactus graph* (y-axis) and *construct graph* (x-axis). Colors and shapes represent the 11 non-reference strains and two clades, respectively.

References

1. **Phenotypic impact of genomic structural variation: insights from and for human disease**
Joachim Weischenfeldt, Orsolya Symmons, François Spitz, Jan O. Korbel
Nature Reviews Genetics (2013-02) https://doi.org/f4nhxh
DOI: 10.1038/nrg3373 · PMID: 23329113

2. **An integrated map of structural variation in 2,504 human genomes**
Peter H. SudmantTobias Rausch, Eugene J. Gardner, Robert E. Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, … Jan O. Korbel
Nature (2015-10) https://doi.org/73c
DOI: 10.1038/nature15394 · PMID: 26432246 · PMCID: PMC4617611

3. **Whole-genome sequence variation, population structure and demographic history of the Dutch population**
Laurent C FrancioliAndroniki Menelaou, Sara L Pulit, Freerk van Dijk, Pier Francesco Palamara, Clara C Elbers, Pieter BT Neerincx, Kai Ye, Victor Guryev, … Cisca Wijmenga
Nature Genetics (2014-06-29) https://doi.org/f6bxm8
DOI: 10.1038/ng.3021 · PMID: 24974849

4. **Resolving the complexity of the human genome using single-molecule sequencing**
Mark J. P. Chaisson, John Huddleston, Megan Y. Dennis, Peter H. Sudmant, Maika Malig, Fereydoun Hormozdiari, Francesca Antonacci, Urvashi Surti, Richard Sandstrom, Matthew Boitano, … Evan E. Eichler
Nature (2014-11-10) https://doi.org/w69
DOI: 10.1038/nature13907 · PMID: 25383537 · PMCID: PMC4317254

5. **Discovery and genotyping of structural variation from long-read haploid genome sequence data**
John Huddleston, Mark J.P. Chaisson, Karyn Meltz Steinberg, Wes Warren, Kendra Hoekzema, David Gordon, Tina A. Graves-Lindsay, Katherine M. Munson, Zev N. Kronenberg, Laura Vives, … Evan E. Eichler
Genome Research (2016-11-28) https://doi.org/f9x79h
DOI: 10.1101/gr.214007.116 · PMID: 27895111 · PMCID: PMC5411763

6. **Mapping and phasing of structural variation in patient genomes using nanopore sequencing**
Mircea Cretu Stancu, Markus J. van Roosmalen, Ivo Renkens, Marleen M. Nieboer, Sjors Middelkamp, Joep de Ligt, Giulia Pregno, Daniela Giachino, Giorgia Mandrile, Jose Espejo Valle-Inclan, … Wigard P. Kloosterman

Nature Communications (2017-11-06) https://doi.org/gftpt9
DOI: 10.1038/s41467-017-01343-4 · PMID: 29109544 · PMCID: PMC5673902

7. **Genome-wide reconstruction of complex structural variants using read clouds**
Noah Spies, Ziming Weng, Alex Bishara, Jennifer McDaniel, David Catoe, Justin M Zook, Marc Salit, Robert B West, Serafim Batzoglou, Arend Sidow
Nature Methods (2017-07-17) https://doi.org/gbnhwk
DOI: 10.1038/nmeth.4366 · PMID: 28714986 · PMCID: PMC5578891

8. **Characterizing the Major Structural Variant Alleles of the Human Genome**
Peter A. Audano, Arvis Sulovari, Tina A. Graves-Lindsay, Stuart Cantsilieris, Melanie Sorensen, AnneMarie E. Welch, Max L. Dougherty, Bradley J. Nelson, Ankeeta Shah, Susan K. Dutcher, … Evan E. Eichler
Cell (2019-01) https://doi.org/gfthvz
DOI: 10.1016/j.cell.2018.12.019 · PMID: 30661756

9. **Nanopore sequencing and assembly of a human genome with ultra-long reads**
Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, … Matthew Loose
Nature Biotechnology (2018-01-29) https://doi.org/gczffw
DOI: 10.1038/nbt.4060 · PMID: 29431738 · PMCID: PMC5889714

10. **Phased diploid genome assembly with single-molecule real-time sequencing**
Chen-Shan Chin, Paul Peluso, Fritz J Sedlazeck, Maria Nattestad, Gregory T Concepcion, Alicia Clum, Christopher Dunn, Ronan O'Malley, Rosa Figueroa-Balderas, Abraham Morales-Cruz, … Michael C Schatz
Nature Methods (2016-10-17) https://doi.org/f9fv4w
DOI: 10.1038/nmeth.4035 · PMID: 27749838 · PMCID: PMC5503144

11. **Genome graphs and the evolution of genome inference**
Benedict Paten, Adam M. Novak, Jordan M. Eizenga, Erik Garrison
Genome Research (2017-03-30) https://doi.org/f95nhd
DOI: 10.1101/gr.214155.116 · PMID: 28360232 · PMCID: PMC5411762

12. **Variation graph toolkit improves read mapping by representing genetic variation in the reference**
Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F Lin, … Richard Durbin
Nature Biotechnology (2018-08-20) https://doi.org/gd2zqs
DOI: 10.1038/nbt.4227 · PMID: 30125266 · PMCID: PMC6126949

13. **Fast and accurate genomic analyses using genome graphs**
Goran Rakocevic, Vladimir Semenyuk, Wan-Ping Lee, James Spencer, John Browning, Ivan J.

Johnson, Vladan Arsenijevic, Jelena Nadj, Kaushik Ghose, Maria C. Suciu, … Deniz Kural
Nature Genetics (2019-01-14) https://doi.org/gftd46
DOI: 10.1038/s41588-018-0316-4 · PMID: 30643257

14. **Graphtyper enables population-scale genotyping using pangenome graphs**
Hannes P Eggertsson, Hakon Jonsson, Snaedis Kristmundsdottir, Eirikur Hjartarson, Birte Kehr, Gisli Masson, Florian Zink, Kristjan E Hjorleifsson, Aslaug Jonasdottir, Adalbjorg Jonasdottir, … Bjarni V Halldorsson
Nature Genetics (2017-09-25) https://doi.org/gbx7v6
DOI: 10.1038/ng.3964 · PMID: 28945251

15. **Accurate genotyping across variant classes and lengths using variant graphs**
Jonas Andreas SibbesenLasse Maretty, Anders Krogh
Nature Genetics (2018-06-18) https://doi.org/gdndnz
DOI: 10.1038/s41588-018-0145-5 · PMID: 29915429

16. **SpeedSeq: ultra-fast personal genome analysis and interpretation**
Colby Chiang, Ryan M Layer, Gregory G Faust, Michael R Lindberg, David B Rose, Erik P Garrison, Gabor T Marth, Aaron R Quinlan, Ira M Hall
Nature Methods (2015-08-10) https://doi.org/gcpgfh
DOI: 10.1038/nmeth.3505 · PMID: 26258291 · PMCID: PMC4589466

17. **DELLY: structural variant discovery by integrated paired-end and split-read analysis**
T. Rausch, T. Zichner, A. Schlattl, A. M. Stutz, V. Benes, J. O. Korbel
Bioinformatics (2012-09-07) https://doi.org/f38r2c
DOI: 10.1093/bioinformatics/bts378 · PMID: 22962449 · PMCID: PMC3436805

18. **Multi-platform discovery of haplotype-resolved structural variation in human genomes**
Mark J.P. Chaisson, Ashley D. Sanders, Xuefang Zhao, Ankit Malhotra, David Porubsky, Tobias Rausch, Eugene J. Gardner, Oscar Rodriguez, Li Guo, Ryan L. Collins, … Charles Lee
Cold Spring Harbor Laboratory (2017-09-23) https://doi.org/gftxhc
DOI: 10.1101/193144

19. **Extensive sequencing of seven human genomes to characterize benchmark reference materials**
Justin M. Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E. Mason, Noah Alexander, … Marc Salit
Scientific Data (2016-06-07) https://doi.org/f84nqc
DOI: 10.1038/sdata.2016.25 · PMID: 27271295 · PMCID: PMC4896128

20. **Reproducible integration of multiple sequencing datasets to form high-confidence SNP, indel, and reference calls for five human genome reference materials**
Justin Zook, Jennifer McDaniel, Hemang Parikh, Haynes Heaton, Sean A Irvine, Len Trigg,

Rebecca Truty, Cory Y McLean, Francisco M De La Vega, Chunlin Xiao, …
Cold Spring Harbor Laboratory (2018-03-13) https://doi.org/gfwsmj
DOI: 10.1101/281006

21. **Contrasting evolutionary genome dynamics between domesticated and wild yeasts**
Jia-Xing Yue, Jing Li, Louise Aigrain, Johan Hallin, Karl Persson, Karen Oliver, Anders Bergström, Paul Coupland, Jonas Warringer, Marco Cosentino Lagomarsino, … Gianni Liti
Nature Genetics (2017-04-17) https://doi.org/f93kpp
DOI: 10.1038/ng.3847 · PMID: 28416820 · PMCID: PMC5446901

22. **Assemblytics: a web analytics tool for the detection of variants from an assembly**
Maria Nattestad, Michael C. Schatz
Bioinformatics (2016-06-17) https://doi.org/f9c485
DOI: 10.1093/bioinformatics/btw369 · PMID: 27318204 · PMCID: PMC6191160

23. **Discovery, genotyping and characterization of structural variation and novel sequence at single nucleotide resolution from de novo genome assemblies on a population scale**
Siyang LiuShujia Huang, Junhua Rao, Weijian Ye, Anders Krogh, Jun Wang
GigaScience (2015-12) https://doi.org/f75r4n
DOI: 10.1186/s13742-015-0103-4 · PMID: 26705468 · PMCID: PMC4690232

24. **Minimap2: pairwise alignment for nucleotide sequences**
Heng Li
Bioinformatics (2018-05-10) https://doi.org/gdhbqt
DOI: 10.1093/bioinformatics/bty191 · PMID: 29750242 · PMCID: PMC6137996

25. **Cactus: Algorithms for genome multiple sequence alignment**
B. Paten, D. Earl, N. Nguyen, M. Diekhans, D. Zerbino, D. Haussler
Genome Research (2011-06-10) https://doi.org/bk4697
DOI: 10.1101/gr.123356.111 · PMID: 21665927 · PMCID: PMC3166836

26. **Toil enables reproducible, open source, big biomedical data analyses**
John Vivian, Arjun Arkal Rao, Frank Austin Nothaft, Christopher Ketchum, Joel Armstrong, Adam Novak, Jacob Pfeil, Jake Narkizian, Alden D Deran, Audrey Musselman-Brown, … Benedict Paten
Nature Biotechnology (2017-04) https://doi.org/gfxbhs
DOI: 10.1038/nbt.3772 · PMID: 28398314 · PMCID: PMC5546205

27. **Superbubbles, Ultrabubbles, and Cacti**
Benedict Paten, Jordan M. Eizenga, Yohei M. Rosen, Adam M. Novak, Erik Garrison, Glenn Hickey
Journal of Computational Biology (2018-07) https://doi.org/gdw582
DOI: 10.1089/cmb.2017.0251 · PMID: 29461862 · PMCID: PMC6067107

28. **Evaluation of computational genotyping of Structural Variations for clinical diagnoses.**
Varuna Chander, Richard A Gibbs, Fritz J Sedlazeck
Cold Spring Harbor Laboratory (2019-02-22) https://doi.org/gfwf66
DOI: 10.1101/558247

29. **Nebula: Ultra-efficient mapping-free structural variant genotyper**
Parsoa Khorsand, Fereydoun Hormozdiari
Cold Spring Harbor Laboratory (2019-03-04) https://doi.org/gfwf67
DOI: 10.1101/566620

30. **Sequencing and de novo assembly of 150 genomes from Denmark as a population reference**
Lasse Maretty, Jacob Malte Jensen, Bent Petersen, Jonas Andreas Sibbesen, Siyang Liu, Palle Villesen, Laurits Skov, Kirstine Belling, Christian Theil Have, Jose M. G. Izarzugaza, … Mikkel Heide Schierup
Nature (2017-07-26) https://doi.org/gbpnnx
DOI: 10.1038/nature23264 · PMID: 28746312