

UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Genotyping structural variants in pangenome graphs using the vg toolkit

Permalink

<https://escholarship.org/uc/item/1s75d98k>

Journal

Genome Biology, 21(1)

ISSN

1474-760X

Authors

Hickey, Glenn

Heller, David

Monlong, Jean

et al.

Publication Date

2020-12-01

DOI

10.1186/s13059-020-1941-7

Peer reviewed

METHOD

Open Access

# Genotyping structural variants in pangenome graphs using the vg toolkit



Glenn Hickey<sup>1†</sup>, David Heller<sup>1,2†</sup>, Jean Monlong<sup>1†</sup>, Jonas A. Sibbesen<sup>1</sup>, Jouni Sirén<sup>1</sup>, Jordan Eizenga<sup>1</sup>, Eric T. Dawson<sup>3,4</sup>, Erik Garrison<sup>1</sup>, Adam M. Novak<sup>1</sup> and Benedict Paten<sup>1\*</sup>

## Abstract

Structural variants (SVs) remain challenging to represent and study relative to point mutations despite their demonstrated importance. We show that variation graphs, as implemented in the vg toolkit, provide an effective means for leveraging SV catalogs for short-read SV genotyping experiments. We benchmark vg against state-of-the-art SV genotypers using three sequence-resolved SV catalogs generated by recent long-read sequencing studies. In addition, we use assemblies from 12 yeast strains to show that graphs constructed directly from aligned de novo assemblies improve genotyping compared to graphs built from intermediate SV catalogs in the VCF format.

## Introduction

A structural variant (SV) is a genomic mutation involving 50 or more base pairs. SVs can take several forms such as deletions, insertions, inversions, translocations, or other complex events. Due to their greater size, SVs often have a larger impact on phenotype than smaller events such as single nucleotide variants (SNVs) and small insertions and deletions (indels) [1]. Indeed, SVs have long been associated with developmental disorders, cancer, and other complex diseases and phenotypes [2].

Despite their importance, SVs remain much more poorly studied than their smaller mutational counterparts. This discrepancy stems from technological limitations. Short-read sequencing has provided the basis of most modern genome sequencing studies due to its high base-level accuracy and relatively low cost, but is poorly suited for discovering SVs. The central obstacle is in mapping short reads to the human reference genome. It is generally difficult or impossible to unambiguously map a short read if the sample whose genome is being analyzed differs substantially from the reference at the read's location. The large size of SVs virtually guarantees that short reads derived from them will not map to the linear reference genome. For example, if a read

corresponds to sequence in the middle of a large reference-relative insertion, then there is no location in the reference that corresponds to a correct mapping. The best result a read mapper could hope to produce would be to leave it unmapped. Moreover, SVs often lie in repeat-rich regions, which further frustrate read mapping algorithms.

Short reads can be more effectively used to genotype known SVs. This is important, as even though efforts to catalog SVs with other technologies have been highly successful, their cost currently prohibits their use in large-scale studies that require hundreds or thousands of samples such as disease association studies. Traditional SV genotypers start from reads that were mapped to a reference genome, extracting aberrant mapping that might support the presence of the SV of interest. Current methods such as SVTyper [3] and the genotyping module of Delly [4] (henceforth referred to as Delly Genotyper) typically focus on split reads and paired reads mapped too close or too far from each other. These discordant reads are tallied and remapped to the reference sequence modified with the SV of interest in order to genotype deletions, insertions, duplications, inversions, and translocations. SMRT-SV v2 Genotyper uses a different approach: the reference genome is augmented with SV-containing sequences as alternate contigs and the resulting mappings are evaluated with a machine learning model trained for this purpose [5].

\* Correspondence: [bpaten@ucsc.edu](mailto:bpaten@ucsc.edu)

<sup>†</sup>Glenn Hickey, David Heller and Jean Monlong contributed equally to this work.

<sup>1</sup>UC Santa Cruz Genomics Institute, University of California, Santa Cruz, California, USA

Full list of author information is available at the end of the article



The catalog of known SVs in human is quickly expanding. Several large-scale projects have used short-read sequencing and extensive discovery pipelines on large cohorts, compiling catalogs with tens of thousands of SVs in humans [6, 7], using split read and discordant pair-based methods like Delly [4] to find SVs using short-read sequencing. More 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 [5, 8–11]. These technologies are also enabling the production of high-quality de novo genome assemblies [8, 12], and large blocks of haplotype-resolved sequences [13]. Such technical advances promise to expand the amount of known genomic variation in humans in the near future, and further power SV genotyping studies. Representing known structural variation in the wake of increasingly larger datasets poses a considerable challenge, however. VCF, the standard format for representing small variants, is unwieldy when used for SVs due its unsuitability for expressing nested or complex variants. Another strategy consists in incorporating SVs into a linear pangenome reference via alt contigs, but it also has serious drawbacks. Alt contigs tend to increase mapping ambiguity. In addition, it is unclear how to scale this approach as SV catalogs grow.

Pangenomic graph reference representations offer an attractive approach for storing genetic variation of all types [14]. These graphical data structures can seamlessly represent both SVs and point mutations using the same semantics. Moreover, including known variants in the reference makes read mapping, variant calling, and genotyping variant-aware. This leads to benefits in terms of accuracy and sensitivity [15–17]. The coherency of this model allows different variant types to be called and scored simultaneously in a unified framework.

vg is the first openly available variation graph tool to scale to multi-gigabase genomes. It provides read mapping, variant calling, and visualization tools [15]. In addition, vg can build graphs both from variant catalogs in the VCF format and from assembly alignments.

Other tools have used genome graphs or pangenomes to genotype variants. GraphTyper realigns mapped reads to a graph built from known SNVs and short indels using a sliding-window approach [18]. BayesTyper first builds a set of graphs from known variants including SVs, then genotypes variants by comparing the distribution of k-mers in the sequencing reads with the k-mers of haplotype candidate paths in the graph [19]. Paragraph builds a graph for each breakpoint of known variants [20], then, for each breakpoint, it pulls out all nearby reads from the linear alignment and realigns them to the graph. Genotypes are computed using the read coverage from the pair of breakpoint graphs

corresponding to each SV. These graph-based approaches showed clear advantages over standard methods that use only the linear reference.

In this work, we present a SV genotyping framework based on the variation graph model and implemented in the vg toolkit. We show that this method is capable of genotyping known deletions, insertions, and inversions and that its performance is not inhibited by small errors in the specification of SV allele breakpoints. We evaluated the genotyping accuracy of our approach using simulated and real Illumina reads and a pangenome built from SVs discovered in recent long-read sequencing studies [5, 21–23]. We also compared vg's performance with state-of-the-art SV genotypers: SVTyper [3], Delly Genotyper [4], BayesTyper [19], Paragraph [20], and SMRT-SV v2 Genotyper [5]. Across the datasets we tested, which range in size from 26 k to 97 k SVs, vg is the best performing SV genotyper on real short-read data for all SV types in the majority of cases. Finally, we demonstrate that a pangenome graph built from the alignment of de novo assemblies of diverse *Saccharomyces cerevisiae* strains improves SV genotyping performance.

## Results

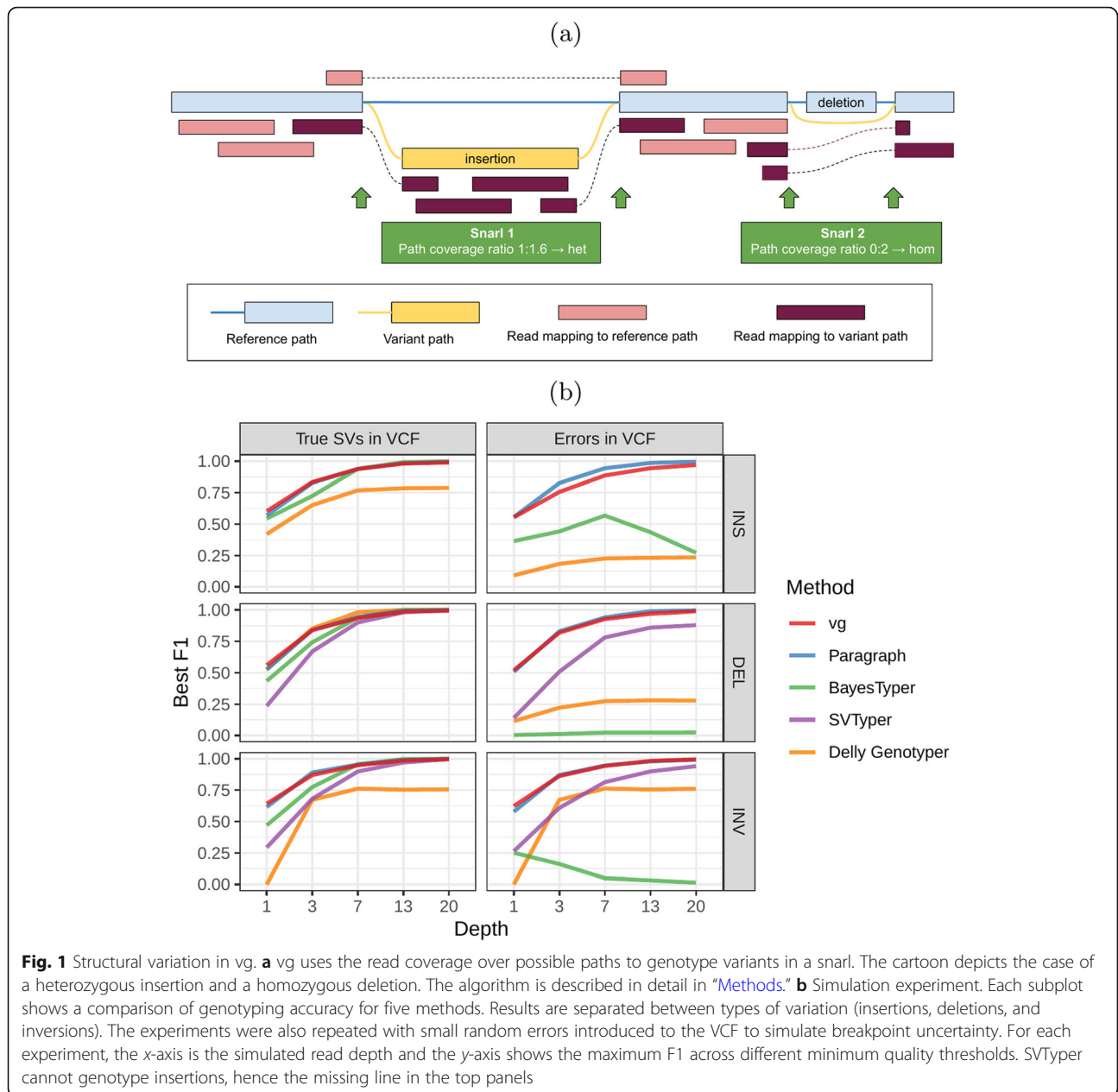
### Structural variation in vg

We used vg to implement a straightforward SV genotyping pipeline. Reads are mapped to the graph and used to compute the read support for each node and edge (see Additional file 1 for a description of the graph formalism). Sites of variation within the graph are then identified using the snarl decomposition as described in [24]. These sites correspond to intervals along the reference paths (e.g., contigs or chromosomes) which are embedded in the graph. They also contain nodes and edges deviating from the reference path, which represent variation at the site. For each site, the two most supported paths spanning its interval (haplotypes) are determined, and their relative supports used to produce a genotype at that site (Fig. 1a). The pipeline is described in detail in “Methods.” We rigorously evaluated the accuracy of our method on a variety of datasets and present these results in the remainder of this section.

### Simulated dataset

As a proof of concept, we simulated genomes and different types of SVs with a size distribution matching real SVs [22]. We compared vg against Paragraph, SVTyper, Delly Genotyper, and BayesTyper across different levels of sequencing depth. We also added some errors (1–10 bp) to the location of the breakpoints to investigate their effect on genotyping accuracy (see “Methods”). The results are shown in Fig. 1b.

When using the correct breakpoints, most methods performed similarly, with differences only becoming



**Fig. 1** Structural variation in vg. **a** vg uses the read coverage over possible paths to genotype variants in a snarl. The cartoon depicts the case of a heterozygous insertion and a homozygous deletion. The algorithm is described in detail in “Methods.” **b** Simulation experiment. Each subplot shows a comparison of genotyping accuracy for five methods. Results are separated between types of variation (insertions, deletions, and inversions). The experiments were also repeated with small random errors introduced to the VCF to simulate breakpoint uncertainty. For each experiment, the x-axis is the simulated read depth and the y-axis shows the maximum F1 across different minimum quality thresholds. SVTyper cannot genotype insertions, hence the missing line in the top panels

visible at very low sequencing depths. Only vg and Paragraph maintained their performance in the presence of 1–10-bp errors in the breakpoint locations. The dramatic drop for BayesTyper can be explained by its k-mer-based approach that requires precise breakpoints. Overall, these results show that vg is capable of genotyping SVs and is robust to breakpoint inaccuracies in the input VCF.

**HGSVC dataset**

A total of 72,485 structural variants from The Human Genome Structural Variation Consortium (HGSVC) were used to benchmark the genotyping performance of vg

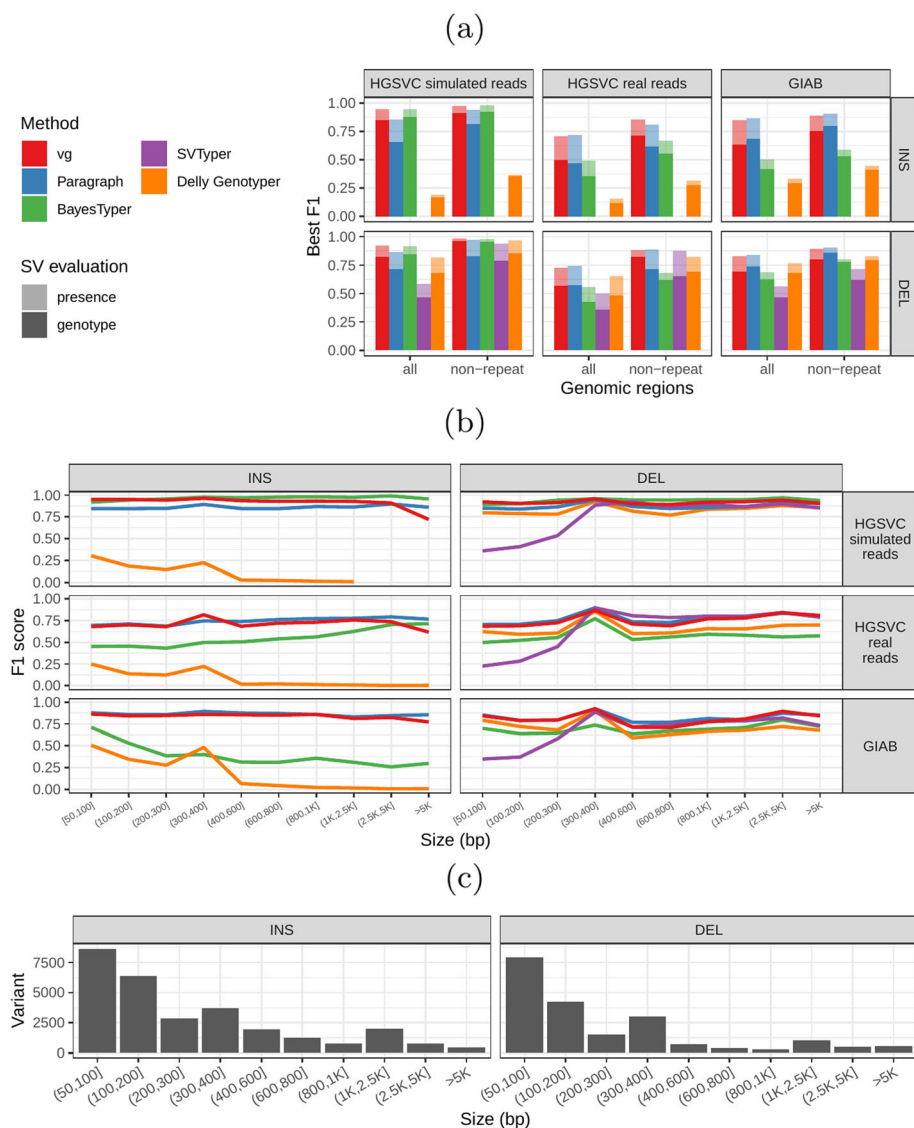
against the four other SV genotyping methods. This high-quality SV catalog was generated from three samples using a consensus from different sequencing, phasing, and variant calling technologies [22]. The three individual samples represent different human populations: Han Chinese (HG00514), Puerto-Rican (HG00733), and Yoruban Nigerian (NA19240). We used these SVs to construct a graph with vg and as input for the other genotypers. Using short sequencing reads, the SVs were genotyped and compared with the genotypes in the original catalog (see “Methods”).

First, we compared the methods using simulated reads for HG00514. This represents the ideal situation where

the SV catalog exactly matches the SVs supported by the reads. BayesTyper and vg showed the best F1 score and precision-recall trade-offs (Fig. 2a and Additional file 1: Figure S1, Additional file 1: Table S1), outperforming the other methods by a clear margin. When restricting the comparisons to regions not identified as tandem repeats or segmental duplications, the genotyping predictions were significantly better for all methods. We observed similar results when evaluating the presence of

an SV call instead of the exact genotype (Fig. 2a and Additional file 1: Figure S2).

We then repeated the analysis using real Illumina reads from the three HGSC samples to benchmark the methods on a more realistic experiment. Here, vg clearly outperformed other approaches (Fig. 2a and Additional file 1: Figure S3). In non-repeat regions and insertions across the whole genome, the F1 scores and precision-recall AUC were higher for vg compared to



**Fig. 2** Structural variants from the HGSC and Genome in a Bottle datasets. HGSC: Simulated and real reads were used to genotype SVs and compared with the high-quality calls from Chaisson et al. [22]. 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 [21, 23, 25]. **a** Maximum F1 score for each method (color), across the whole genome or focusing on non-repeat regions (x-axis). We evaluated the ability to predict the presence of an SV (transparent bars) and the exact genotype (solid bars). Results are separated across panels by variant type: insertions and deletions. SVTyper cannot genotype insertions, hence the missing bars in the top panels. **b** Maximum F1 score for different size classes when evaluating on the presence of SVs across the whole genome. **c** Size distribution of SVs in the HGSC and GIAB catalogs

other methods. For example, for deletions in non-repeat regions, the F1 score for *vg* was 0.824 while the second best method, Paragraph, had a F1 score of 0.717. We observed similar results when evaluating the presence of an SV call instead of the exact genotype (Fig. 2a and Additional file 1: Figure S4).

In general, the genotyped variants were matched 1-to-1 with variants in the truth set but some methods showed some signs of “over-genotyping” that is not reflected in the precision/recall/F1 scores. Methods like Paragraph, Delly Genotyper, or SVTyper tended to genotype on average more than one variant per truth-set variant (Additional file 1: Figure S5). Like other SV catalogs, the HGSCV catalog is not fully sequence-resolved and contains a number of near-duplicates with slightly different breakpoint definition. When genotyping a sample, multiple versions of a variant are genotyped multiple times by methods that analyze each variant independently. In contrast, *vg* follows a unified path-centric approach that only selects the best genotype in a region (see “Methods”).

We further evaluate the performance for different SV sizes and repeat content. In addition, *vg*'s performance was stable across the spectrum of SV sizes (Fig. 2b, c). By annotating the repeat content of the deleted/inserted sequence, we further evaluated *vg*'s performance across repeat classes. As expected, simple repeat variation was more challenging to genotype than transposable element polymorphisms (Additional file 1: Figure S6). Figure 3 shows an example of an exonic deletion that was correctly genotyped by *vg* but not by BayesTyper, SVTyper, or Delly Genotyper.

### Other long-read datasets

#### *Genome in a bottle consortium*

The Genome in a Bottle (GiAB) consortium is currently producing a high-quality SV catalog for an Ashkenazim individual (HG002) [21, 23, 25]. Dozens of SV callers operating on datasets from short, long, and linked reads were used to produce this set of SVs. We evaluated the SV genotyping methods on this sample as well using the GIAB VCF, which also contains parental calls (HG003 and HG004), all totaling 30,224 SVs. Relative to the HGSCV dataset, *vg* performed similarly but Paragraph saw a large boost in accuracy and was the most accurate method across all metrics (Fig. 2, Additional file 1: Figure S7 and S8, and Additional file 1: Table S2). As before, the remaining methods produced lower F1 scores.

#### *SMRT-SV v2 catalog and training data* [5]

A recent study by Audano et al. generated a catalog of 97,368 SVs (referred as SVPOP below) using long-read sequencing across 15 individuals [5]. These variants were then genotyped from short reads across 440 individuals

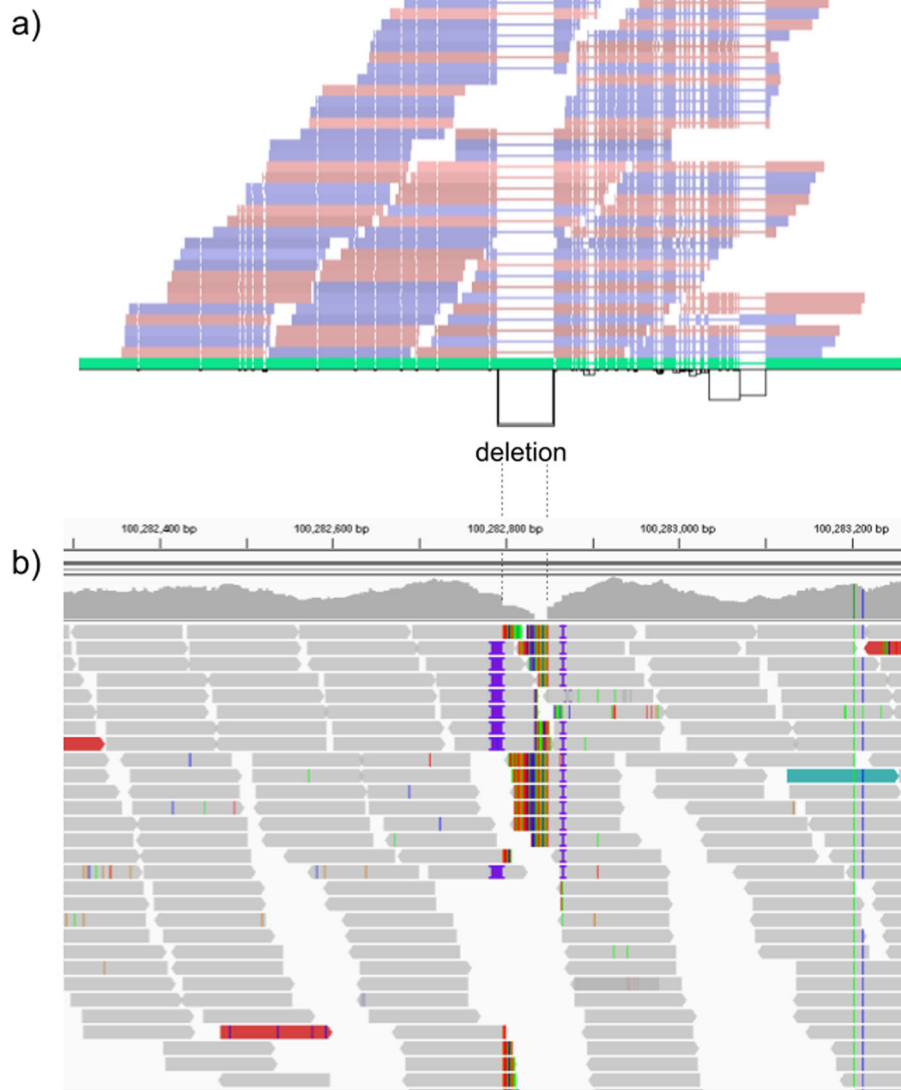
using the SMRT-SV v2 Genotyper, a machine learning-based tool implemented for that study. The SMRT-SV v2 Genotyper was trained on a pseudodiploid genome constructed from high-quality assemblies of two haploid cell lines (CHM1 and CHM13) and a single negative control (NA19240). We first used *vg* to genotype the SVs in this two-sample training dataset using 30× coverage reads and compared the results with the SMRT-SV v2 Genotyper. *vg* was systematically better at predicting the presence of an SV for both SV types, but SMRT-SV v2 Genotyper produced slightly better genotypes for deletions in the whole genome (see Fig. 4, Additional file 1: Figure S9 and S10, and Additional file 1: Table S3). To compare *vg* and SMRT-SV v2 Genotyper on a larger dataset, we then genotyped SVs from the entire SVPOP catalog with both methods, using the read data from the three HGSCV samples described above. Given that the SVPOP catalog contains these three samples, we once again evaluated accuracy by using the long-read calls as a baseline. Paragraph was included as an additional point of comparison.

Compared to SMRT-SV v2 Genotyper, *vg* had a better precision-recall curve and a higher F1 for both insertions and deletions (SVPOP in Fig. 4 and Additional file 1: Figure S11, and Additional file 1: Table S4). Paragraph's performance was virtually identical to *vg*'s. Of note, SMRT-SV v2 Genotyper produces *no-calls* in regions where the read coverage is too low, and we observed that its recall increased when filtering these regions out the input set. Interestingly, *vg* performed well even in regions where SMRT-SV v2 Genotyper produced *no-calls* (Additional file 1: Figure S12 and Additional file 1: Table S5). Audano et al. discovered 217 sequence-resolved inversions using long reads, which we attempted to genotype. *vg* correctly predicted the presence of around 14% of the inversions present in the three samples (Additional file 1: Table S4). Inversions are often complex, harboring additional variation that makes their characterization and genotyping challenging.

#### **Graphs from alignment of de novo assemblies**

We can construct variation graphs directly from whole-genome alignments (WGA) of multiple de novo assemblies [15]. This bypasses the need for generating an explicit variant catalog relative to a linear reference, which could be a source of error due to the reference bias inherent in read mapping and variant calling. Genome alignments from graph-based software such as Cactus [27] can contain complex structural variation that is extremely difficult to represent, let alone call, outside of a graph, but which is nevertheless representative of the actual genomic variation between the aligned assemblies. We sought to establish if graphs built in this fashion provide advantages for SV genotyping.

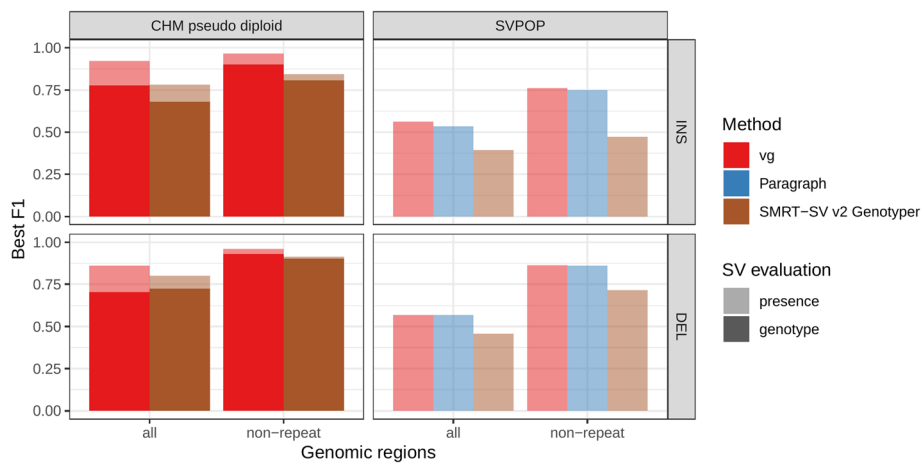




**Fig. 3** Exonic deletion in the HGSCV dataset correctly genotyped by *vg*. **a** Visualization of the HGSCV graph as augmented by reads aligned by *vg* at a locus harboring a 51-bp homozygous deletion in the UTR region of the *LONRF2* gene. At the bottom, a horizontal black line represents the topologically sorted nodes of the graph. Black rectangles represent edges found in the graph. Above this rendering of the topology, the reference path from GRCh38 is shown (in green). Red and blue bars represent reads mapped to the graph. Thin lines in the reference path and read mappings highlight relative gaps (either insertions or deletions) against the full graph. The *vg* read mappings show consistent coverage even over the deletion. **b** Reads mapped to the linear genome reference GRCh38 using *bwa mem* [26] in the same region. Reads contain soft-clipped sequences and short insertions near the deletion breakpoints. Part of the deleted region is also covered by several reads, potentially confusing traditional SV genotypers

To do so, we analyzed public sequencing datasets for 12 yeast strains from two related clades (*S. cerevisiae* and *S. paradoxus*) [28]. We distinguished two different strain sets, in order to assess how the completeness of the graph affects the results. For the *all-strain set*, all 12 strains were used, with *S.c. S288C* as the reference strain. For the *five-strain set*, *S.c. S288C* was used as the reference strain, and we selected two other strains from each of the two clades (see “Methods”). We compared genotyping results from a WGA-derived graph (*cactus graph*)

with results from a VCF-derived graph (*VCF graph*). The *VCF graph* was created from the linear reference genome of the *S.c. S288C* strain and a set of SVs relative to this reference strain in VCF format identified from the other assemblies in the respective strain set by three methods: Assemblytics [29], AsmVar [30], and *paftools* [31]. The *cactus graph* was derived from a multiple genome alignment of the strains in the respective strain set using *Cactus* [27]. The *VCF graph* is mostly linear and highly dependent on the reference genome. In contrast,



**Fig. 4** Structural variants from SMRT-SV v2 [5]. The pseudodiploid genome built from two CHM cell lines and one negative control sample was originally used to train SMRT-SV v2 Genotyper in Audano et al. [5]. It contains 16,180 SVs. The SVPOP panel shows the combined results for the HG00514, HG00733, and NA19240 individuals, three of the 15 individuals used to generate the high-quality SV catalog in Audano et al. [5]. Here, we report the maximum F1 score (y-axis) for each method (color), across the whole genome or focusing on non-repeat regions (x-axis). We evaluated the ability to predict the presence of an SV (transparent bars) and the exact genotype (solid bars). Genotype information is not available in the SVPOP catalog hence genotyping performance could not be evaluated

the *cactus graph* is structurally complex and relatively free of reference bias.

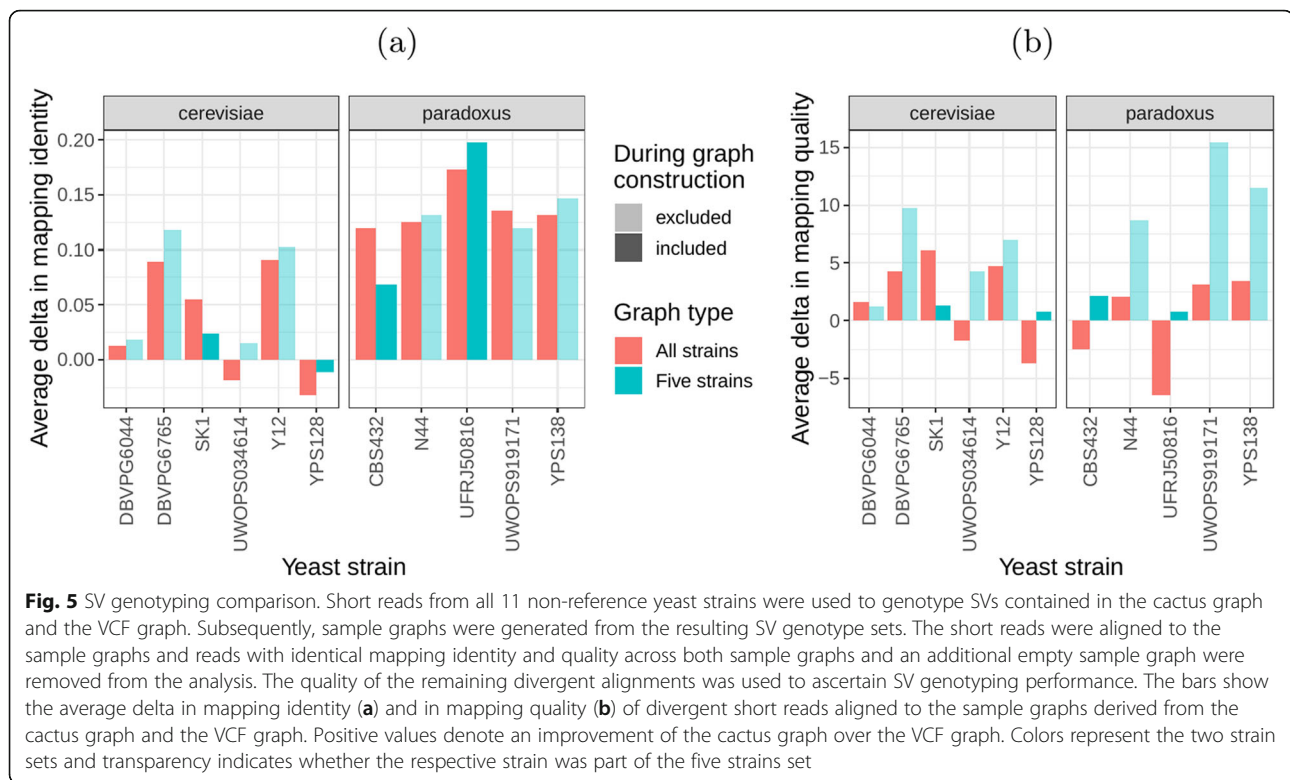
First, we tested our hypothesis that the *cactus graph* has higher mappability due to its better representation of sequence diversity among the yeast strains (see Additional file 1). Generally, more reads mapped to the *cactus graph* with high identity (Additional file 1: Figure S13a and S14a) and high mapping quality (Additional file 1: Figure S13b and S14b) than to the *VCF graph*. On average, 88%, 79%, and 68% of reads mapped to the *all strain cactus graph* with an identity of at least 50%, 90%, and 100%, respectively, compared to only 77%, 57%, and 23% of reads on the *all strain VCF graph*. Similarly, 88% of reads mapped to the *all strain cactus graph* with a mapping quality of at least 30 compared to only 80% of reads on the *all strain VCF graph*.

Next, we compared the SV genotyping performance of both graph types. We mapped short reads from the 11 non-reference strains to both graphs and genotyped variants for each strain using the *vg* toolkit's variant calling module (see "Methods"). There is no gold standard available for these samples to compare against which renders an evaluation using recall, precision, and F1 score impossible. Therefore, we used an indirect measure of SV genotyping accuracy. We evaluated each SV genotype set based on the alignment of reads to a *sample graph* constructed from the genotype set (see "Methods"). Conceptually, the sample graph represents the sample's diploid genome by starting out from the reference genome and augmenting it with the genotype results. If a given genotype set is correct, we expect that

reads from the same sample will be mapped with high identity and confidence to the corresponding sample graph. To specifically quantify mappability in SV regions, we excluded reads that produced identical mapping quality and identity on both sample graphs and an empty sample graph containing the linear reference only (see "Methods" and Additional file 1: Figure S15 for results from all reads). Then, we analyzed the average delta in mapping identity and mapping quality of the remaining short reads between both sample graphs (Fig. 5a, b).

For most of the strains, we observed an improvement in mapping identity of the short reads on the *cactus sample graph* compared to the *VCF sample graph*. The mean improvement in mapping identity across the strains (for reads differing in mapping identity) was 8.0% and 8.5% for the *all strains set* graphs and the *five strains set* graphs, respectively. Generally, the improvement in mapping identity was larger for strains in the *S. paradoxus* clade (mean of 13.7% and 13.3% for the two strain sets, respectively) than for strains in the *S. cerevisiae* clade (mean of 3.3% and 4.4%). While the higher mapping identity indicated that the *cactus graph* represents the reads better (Fig. 5a), the higher mapping quality confirmed that this did not come at the cost of added ambiguity or a more complex graph (Fig. 5b). For most strains, we observed an improvement in mapping quality of the short reads on the *cactus sample graph* compared to the *VCF sample graph* (mean improvement across the strains of 1.0 and 5.7 for the two strain sets, respectively).





## Discussion

Overall, graph-based methods were more accurate than traditional SV genotypers in our benchmarks, with *vg* performing best across most datasets. These results show that SV genotyping benefits from variant-aware read mapping and graph-based genotyping, a finding consistent with previous studies [15–19]. Paragraph, another graph-based genotyper which was released as we were submitting this work, was very competitive with *vg* and showed the best overall accuracy on the GIAB dataset. In addition to being featured prominently in Paragraph’s development and evaluation, the GIAB dataset we used was a different coverage (50×) than the other 30× datasets we used. Our simulation results show that Paragraph is slightly more robust than *vg* with respect to differences in coverage and perhaps this is a factor in the difference in performance. In the future, we would like to better model the expected read depth in the *vg* genotyper as it currently does not exploit this information. In contrast, *vg* is much more accurate than Paragraph on the HGSVC dataset and we speculate that this is due to the higher number of overlapping variants. Using the *snarl* decomposition, *vg* can genotype arbitrary combinations of SVs simultaneously, whereas Paragraph operates one at a time.

We took advantage of newly released datasets for our evaluation, which feature up to 3.7 times more variants than the more widely used GIAB benchmark. More and

more large-scale projects are using low-cost short-read technologies to sequence the genomes of thousands to hundreds of thousands of individuals (e.g., the PanCancer Analysis of Whole Genomes [32], the Genomics England initiative [33], and the TOPMed consortium [34]). We believe pangenome graph-based approaches will improve both how efficiently SVs can be represented, and how accurately they can be genotyped with this type of data.

A particular advantage of our method is that it does not require exact breakpoint resolution in the variant library. Our simulations showed that *vg*’s SV genotyping algorithm is robust to errors of as much as 10 bp in breakpoint location. However, there is an upper limit to this flexibility, and we find that *vg* cannot accurately genotype variants with much higher uncertainty in the breakpoint location (like those discovered through read coverage analysis). *vg* is also capable of fine-tuning SV breakpoints by augmenting the graph with differences observed in read alignments. Simulations showed that this approach can usually correct small errors in SV breakpoints (Additional file 1: Figure S16 and Additional file 1: Table S6).

*vg* uses a unified framework to call and score different variant types simultaneously. In this work, we only considered graphs containing certain types of SVs, but the same methods can be extended to a broader range of graphs. For example, we are interested in evaluating how genotyping SVs together with SNPs and small indels

using a combined graph affects the accuracy of studying either alone. The same methods used for genotyping known variants in this work can also be extended to call novel variants by first augmenting the graph with edits from the mapped reads. This approach, which was used only in the breakpoint fine-tuning portion of this work, could be further used to study small variants around and nested within SVs. Novel SVs could be called by augmenting the graph with long-read mappings. *vg* is entirely open source, and its ongoing development is supported by a growing community of researchers and users with common interest in scalable, unbiased pangenomic analyses and representation. We expect this collaboration to continue to foster increases in the speed, accuracy, and applicability of methods based on pangenome graphs in the years ahead.

Our results suggest that constructing a graph from de novo assembly alignment instead of a VCF leads to better SV genotyping. High-quality de novo assemblies for human are becoming more and more common due to improvements in technologies like optimized mate-pair libraries [35] and long-read sequencing [12]. We expect future graphs to be built from the alignment of numerous de novo assemblies, and we are presently working on scaling our assembly-based pipeline to human-sized genome assemblies. Another challenge is creating genome graphs that integrate assemblies with variant-based data resources. One possible approach is to progressively align assembled contigs into variation graphs constructed from variant libraries, but methods for doing so are still experimental.

## Conclusion

In this study, the *vg* toolkit was compared to existing SV genotypers across several high-quality SV catalogs. We showed that its method of mapping reads to a variation graph leads to better SV genotyping compared to other state-of-the-art methods. This work introduces a flexible strategy to integrate the growing number of SVs being discovered with higher resolution technologies into a unified framework for genome inference. Our work on whole-genome alignment graphs shows the benefit of directly utilizing de novo assemblies rather than variant catalogs to integrate SVs in genome graphs. We expect this latter approach to increase in significance as the reduction in long-read sequencing costs drives the creation of numerous new de novo assemblies. We envision a future in which the lines between variant calling, genotyping, alignment, and assembly are blurred by rapid changes in sequencing technology. Fully graph-based approaches, like the one we present here, will be of great utility in this new phase of genome inference.

## Methods

### SV genotyping algorithm

The input to the SV genotyping algorithm is an indexed variation graph in *xg* format along with a (single-sample) read alignment in *GAM* format. If the graph was constructed from a VCF, as was the case for the human-genome graphs discussed in this paper, this VCF can also be input to the caller. The first step is to compute a compressed coverage index from the alignment using this command, `vg pack <graph.xg> <alignment.gam> -Q 5 -o graph.pack`. This index stores the number of reads with mapping quality at least 5 mapped to each edge and each base of each node on the graph. Computing the coverage can be done in a single scan through the reads and, in practice, tends to be an order of magnitude faster than sorting the reads.

Variation graphs, as represented in *vg*, are bidirected. In a bidirected graph, every node can be thought of having two distinct *sides*. See, for example, the left and right sides of each rectangle in Fig. 1a. If  $x$  is the side of a given node  $A$ , then we use the notation  $x'$  to denote the other side of  $A$ . A snarl is defined by a pair of sides,  $x$  and  $y$ , that satisfy the following criteria:

1. Removing all edges incident to  $x'$  and  $y'$  disconnects the graph, creating a connected component  $X$  that contains  $x$  and  $y$ .
2. There is no side  $z$  in  $X$  such that  $\{x, z\}$  satisfies the above criteria. Likewise for  $y$ .

Snarls can be computed in linear time using a cactus graph decomposition [24]. They can be computed once for a given graph using *vg* snarls, or on the fly with *vg* call.

Once the snarls have been identified, the SV genotyping algorithm proceeds as follows. For every snarl in the graph for which both end nodes lie on a reference path (such as a chromosome) and that it is not contained in another snarl, the following steps are performed.

1. All VCF variants,  $v_1, v_2, \dots, v_k$  that are contained within the snarl are looked up using information embedded during graph construction. Let  $|v_i|$  be the number of alleles in the  $i$ th VCF variant. Then there are  $|v_1| \times |v_2| \times \dots \times |v_k|$  possible haplotypes through the snarl. If this number is too high ( $> 500,000$ ), then alleles with average support of less than 1 are filtered out.
2. For each possible haplotype, a corresponding bidirected path through the snarl (from  $x$  to  $y$ ) is computed.
3. For each haplotype path, its average support (over bases and edges) is computed using the compressed coverage index, and the two most

supported paths are selected (ties are broken arbitrarily).

4. If the most supported path exceeds the minimum support threshold (default 1), and has more than  $B$  (default 6) times the support of the next most supported path, the site is called homozygous for the allele associated with the most supported path.
5. Else if the second most supported path exceeds the minimum support threshold (default 1), then the site is deemed heterozygous with an allele from each of the top two paths.
6. Given the genotype computed above, it is trivial to map back from the chosen paths to the VCF alleles in order to produce the final output.

The command to do the above is `vg call <graph.xg> -k <graph.pack> -v variants.vcf.gz`. If the graph was not constructed from a VCF, then a similar algorithm is used except the traversals are computed heuristically searching through the graph. This is enabled by not using the `-v` option in the above command.

#### toil-vg

toil-vg is a set of Python scripts for simplifying vg tasks such as graph construction, read mapping, and SV genotyping. Much of the analysis in this report was done using toil-vg, with the exact commands available at [github.com/vgteam/sv-genotyping-paper](https://github.com/vgteam/sv-genotyping-paper). toil-vg uses the Toil workflow engine [36] to seamlessly run pipelines locally, on clusters or on the cloud. Graph indexing and mapping in particular are computationally expensive (though work is underway to address this) and well-suited to distribution on the cloud. 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. The graph is automatically annotated with paths corresponding to the different alleles in the input VCF. The indexes created are the following:

- `xg` index: This is a compressed version of the graph that allows fast node, edge and path lookups
- `gcsa2` index: This is a substring index used only for read mapping
- `gbwt` index: This is an index of all the haplotypes in the VCF as implied by phasing information. When available, it is used to help ensure that haplotype information is preserved when constructing the `gcsa2` index

- `snarls` index: The snarls represent sites of variation in the graph and are used for genotyping and variant calling.

#### toil-vg map

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

#### toil-vg call

toil-vg call splits the input graph by chromosome and calls each one individually. vg call has been recently updated so that this subdivision is largely unnecessary: the entire graph can be easily called at once. Still, toil-vg can be used to farm this task out to a single cloud node if desired.

#### toil-vg sveval

toil-vg sveval evaluates the SV calls relative to a truth set. Matching SV calls is non-trivial because two SV call-sets often differ slightly around the breakpoints. Even for a genotyping experiment, the same input SVs can have equivalent but different representations. Furthermore, SV catalogs often contain very similar SVs that could be potentially duplicates of the same true variant. To make sure that SVs are matched properly when comparing genotyped SVs and the truth set, we use an approach that overlaps variants and aligns allelic sequences if necessary. It was implemented in the sveval R package (<https://github.com/jmonlong/sveval>). Additional file 1: Figure S17 shows an overview of the SV evaluation approach which is described below. Of note, the variants are first normalized with bcftools norm (1.9) to ensure consistent representation between called variants and baseline variants [37].

For deletions and inversions, we begin by computing the overlaps between the SVs in the call set and the truth set. For each variant, we then compute the proportion of its region that is covered by a variant in the other set, considering only variants overlapping with at least 10% reciprocal overlap. If this coverage proportion is higher than 50%, we consider the variant *covered*. True positives (TPs) are covered variants from the call set (when computing the precision) or the truth set (when computing the recall). Variants from the call set are considered false positives (FPs) if they are not covered by the truth set. Conversely, variants from the truth set are considered false negatives (FNs) if they are not covered by the call set.

For insertions, we select pairs of insertions that are located no farther than 20 bp from each other. We then align the inserted sequences using a Smith-Waterman alignment. For each insertion, we compute the proportion of its inserted sequence that aligns a matched variant in the other set. If this proportion is at least 50%, the

insertions are considered covered. Covering relationships are used to define TPs, FPs, and FNs the same way as for deletions and inversions.

The results shown in this study used a minimum of 50% coverage to match variants but we also replicated the results using 90% minimum coverage and observed similar results (see Additional file 1: Figure S18).

The coverage statistics are computed using any variant larger than 1 bp but a minimum size is required for a variant to be counted as TP, FP, or FN. In this work, we used the default minimum SV size of 50 bp.

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 using the explicit representation and when the REF and ALT sequences are longer than 10 bp, the reverse-complement of the ALT sequence is aligned to the REF sequence to identify potential inversions. If more than 80% of the sequence aligns, it is classified as an inversion.

We assess both the ability to predict the presence of an SV and the full genotype. For the *presence* evaluation, both heterozygous and homozygous alternate SVs are compared jointly using the approach described above. To compute genotype-level metrics, the heterozygous and homozygous SVs are compared separately. Before splitting the variants by genotype, pairs of heterozygous variants with reciprocal overlap of at least 80% are merged into a homozygous ALT variant. To handle fragmented variants, consecutive heterozygous variants located at less than 20 bp from each other are first merged into larger heterozygous variants.

Precision-recall curves are produced by successively filtering out variants of low quality. By default, the *QUAL* field in the VCF file is used as the quality information. If *QUAL* is missing (or contains only 0s), the genotype quality in the *GQ* field is used.

The evaluation is performed using all variants or using only variants within high-confidence regions. In most analysis, the high-confidence regions are constructed by excluding segmental duplications and tandem repeats (using the respective tracks from the UCSC Genome Browser). For the GIAB analysis, we used the Tier 1 high-confidence regions provided by the GIAB consortium in version 0.6.

The inserted/deleted sequence was also annotated using RepeatMasker [38]. SVs were separated by repeat family if the annotated repeat element covered more than 80% of the sequence. We recomputed precision and recall in the most frequent repeat families.

The average number of genotyped variants per variant in the truth set (Additional file 1: Figure S5) was computed by dividing the number of TPs from the call set by the number of TPs from the truth set, i.e.,

the ratio of matched variants between the two variant sets.

### Other SV genotypers

#### *BayesTyper (v1.5 beta 6288d6)*

Where not specified otherwise, BayesTyper was run as follows. Raw reads were mapped to the reference genome using bwa mem [26] (0.7.17). GATK haplotypcaller [39] (3.8) and Platypus [40] (0.8.1.1) with assembly enabled were run on the mapped reads to call SNVs and short indels (< 50 bp) needed by BayesTyper for correct genotyping. The VCFs with these variants were then normalized using bcftools norm (1.9) and combined with the SVs across samples using bayesTyperTools to produce the input candidate set. k-mers in the raw reads were counted using kmc [41] (3.1.1) with a k-mer size of 55. A Bloom filter was constructed from these k-mers using bayesTyperTools makeBloom. Finally, variants were clustered and genotyped using bayestyper cluster and bayestyper genotype, respectively, with default parameters except --min-genotype-posterior 0. Non-PASS variants and non-SVs (GATK and Platypus origin) were filtered prior to evaluation using bcftools filter and filter-AlleleCallsetOrigin, respectively.

#### *Delly (v0.7.9)*

The delly call command was run on the reads mapped by bwa mem [26], the reference genome FASTA file, and the VCF containing the SVs to genotype (converted to their explicit representations).

#### *SVTyper (v0.7.0)*

The VCF containing deletions was converted to symbolic representation and passed to svtyper with the reads mapped by bwa mem [26]. The output VCF was converted back to explicit representation using bayesTyperTools convertAllele to facilitate variant normalization before evaluation.

#### *Paragraph (v2.3)*

Paragraph was run using default parameters using the multigmpy.py script, taking the input VCF and reads mapped by bwa mem [26] as inputs. We used the genotype estimates in the genotypes.vcf.gz output file. In order for Paragraph to run, we added padding sequence to problematic variants in the input VCFs of the GIAB and SVPOP catalogs.

#### *SMRT-SV v2 Genotyper (v2.0.0 Feb 21 2019 commit adb13f2)*

SMRT-SV v2 Genotyper was run with the “30x-4” model and min-call-depth 8 cutoff. It was run only on VCFs created by SMRT-SV, for which the required contig BAMs were available. The Illumina BAMs used where



the same as the other methods described above. The output VCF was converted back to explicit representation to facilitate variant normalization later.

#### **Running time and memory usage**

Running times and memory usage for the different tools are shown in Additional file 1: Table S7. The *Elapsed (wall clock) time* and the *Maximum resident set size* were extracted from the output of `/usr/bin/time -v`. We show the profiling results when genotyping the HGSVC SV catalog in the HG00514 sample.

#### **Simulation experiment**

We simulated a synthetic genome with 1000 insertions, deletions, and inversions. We separated each variant from the next by a buffer of at least 500 bp. The sizes of deletions and insertions followed the distribution of SV sizes from the HGSVC catalog. We used the same size distribution as deletions for inversions. A VCF file was produced for three simulated samples with genotypes chosen uniformly between homozygous reference, heterozygous, and homozygous alternate.

We created another VCF file containing errors in the SV breakpoint locations. We shifted one or both breakpoints of deletions and inversions by distances between 1 and 10 bp. The locations and sequences of insertions were also modified, either shifting the variants or shortening them at the flanks, again by up to 10 bp.

Paired-end reads were simulated using `vg sim` on the graph that contained the true SVs. Different read depths were tested: 1×, 3×, 7×, 10×, 13×, 20×. The base qualities and sequencing errors were trained to resemble real Illumina reads from NA12878 provided by the Genome in a Bottle Consortium.

The genotypes called in each experiment (genotyping 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 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 20× simulated reads from the simulation experiment (see “Simulation experiment”). The variants called after graph augmentation were compared with the true SVs. We considered fine-tuning correct if the breakpoints matched exactly.

#### **HGSVC analysis**

We first obtained phased VCFs for the three Human Genome Structural Variation Consortium (HGSVC)

samples from Chaisson et al. [22] and combined them 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 GBWT index [42], from which the two haploid sequences from HG00514 were extracted as a graph. Illumina read pairs with 30× coverage were simulated from these sequences using `vg sim`, with an error model learned from real reads from the same sample. These simulated reads reflect an 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. Finally, the SV calls were compared back to the HG00514 genotypes from the HGSVC VCF. We repeated the process with the same reads on the linear reference, using `bwa mem` [26] for mapping and `Delly Genotyper`, `SVTyper`, `Paragraph`, and `BayesTyper` for SV genotyping.

We downloaded Illumina HiSeq 2500 paired-end reads from the EBI’s ENA FTP site for the three samples, using Run Accessions ERR903030, ERR895347, and ERR894724 for HG00514, HG00733, and NA19240, respectively. We ran the graph and linear mapping and genotyping pipelines exactly as for the simulation, and aggregated the comparison results across the three samples. We used `BayesTyper` to jointly genotype the three samples.

#### **GIAB analysis**

We obtained version 0.5 of the Genome in a Bottle (GIAB) SV VCF for the Ashkenazim son (HG002) and his parents from the NCBI FTP site. We obtained Illumina reads as described in Garrison et al. [15] and downsampled them to 50× coverage. We used these reads as input for `vg call` and the other SV genotyping pipelines described above (though with GRCh37 instead of GRCh38). For `BayesTyper`, we created the input variant set by combining the GIAB SVs with SNV and indels from the same study. Variants with reference allele or without a determined genotype for HG002 in the GIAB call set (10,569 out of 30,224) were considered “false positives” as a proxy measure for precision. These variants correspond to putative technical artifacts and parental calls not present in HG002. For the evaluation in high-confidence regions, we used the Tier 1 high-confidence regions provided by the GIAB consortium in version 0.6.

#### **SMRT-SV v2 comparison (CHMPD and SVPOP)**

The SMRT-SV v2 Genotyper can only be used to genotype sequence-resolved SVs present on contigs with known SV breakpoints, such as those created by SMRT-SV v2, and therefore could not be run on the simulated,



HGSVC, or GIAB call sets. The authors shared their training and evaluation set: a pseudodiploid sample constructed from combining the haploid CHM1 and CHM13 samples (CHMPD), and 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 30× read set to construct, map, and genotype with *vg*, and also ran SMRT-SV v2 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 from the training data to a more realistic setting, we reran the three HGSVC samples against the SMRT-SV v2 discovery VCF (SVPOP, which contains 12 additional samples in addition to the three from HGSVC) published by Audano et al. [5] using *vg* and SMRT-SV v2 Genotyper. The discovery VCF does not contain genotypes. In consequence, we were unable to distinguish between heterozygous and homozygous genotypes and instead considered only the presence or absence of a non-reference allele for each variant.

SMRT-SV v2 Genotyper produces explicit *no-call* predictions when the read coverage is too low to produce accurate genotypes. These no-calls are considered homozygous reference in the main accuracy evaluation. We also explored the performance of *vg* and SMRT-SV v2 Genotyper in different sets of regions (Additional file 1: Figure S12 and Additional file 1: Table S5):

1. Non-repeat regions, i.e., excluding segmental duplications and tandem repeats (using the respective tracks from the UCSC Genome Browser).

2. Repeat regions defined as segmental duplications and tandem repeats.
3. Regions where SMRT-SV v2 Genotyper could call variants.
4. Regions where SMRT-SV v2 Genotyper produced no-calls.

#### Yeast graph analysis

For the analysis of graphs from de novo assemblies, we utilized publicly available PacBio-derived assemblies and Illumina short-read sequencing datasets for 12 yeast strains from two related clades (Additional file 1: Table 1) [28]. We constructed graphs from two different strain sets: For the *five strains set*, we selected five strains for graph construction (*S.c. SK1*, *S.c. YPS128*, *S.p. CBS432*, *S.p. UFRJ50816* and *S.c. S288C*). We randomly selected two strains from different subclades of each clade as well as the reference strain *S.c. S288C*. For the *all-strain set* in contrast, we utilized all 12 strains for graph construction. We constructed two different types of genome graphs from the PacBio-derived assemblies of the five or 12 (depending on the strains set) selected strains. In this section, we describe the steps for the construction of both graphs and the genotyping of variants. More details and the precise commands used in our analyses can be found at [github.com/vgteam/sv-genotyping-paper](https://github.com/vgteam/sv-genotyping-paper).

#### Construction of the VCF graph

We constructed the first graph (called the *VCF graph* throughout the paper) by adding variants onto a linear reference. This method requires one assembly to serve as a reference genome. The other assemblies must be converted to variant calls relative to this reference. The PacBio assembly of the *S.c. S288C* strain was chosen as

**Table 1** 12 yeast strains from two related clades were used in our analysis. Five strains were selected to be included in the five strains set and all strains were included in the all strains set. Graphs were constructed from strains in the respective strain set while all 11 non-reference strains were used for genotyping

Strain	Clade	Included in <i>five strains set</i>	Included in <i>all strains set</i>
S288C	<i>S. cerevisiae</i>	✓	✓
SK1	<i>S. cerevisiae</i>	✓	✓
YPS128	<i>S. cerevisiae</i>	✓	✓
UWOPS034614	<i>S. cerevisiae</i>		✓
Y12	<i>S. cerevisiae</i>		✓
DBVPG6765	<i>S. cerevisiae</i>		✓
DBVPG6044	<i>S. cerevisiae</i>		✓
CBS432	<i>S. paradoxus</i>	✓	✓
UFRJ50816	<i>S. paradoxus</i>	✓	✓
N44	<i>S. paradoxus</i>		✓
UWOPS919171	<i>S. paradoxus</i>		✓
YPS138	<i>S. paradoxus</i>		✓

the reference genome because this strain was used for the *S. cerevisiae* genome reference assembly. To obtain variants for the other assemblies, we combined three methods for SV detection from genome assemblies: Assemblytics [29] (commit df5361f), AsmVar (commit 5abd91a) [30], and pafTools (version 2.14-r883) [31]. We constructed a union set of SVs detected by the three methods (using bedtools [43]) and combined variants with a reciprocal overlap of at least 50% to avoid duplication in the union set. We merged these union sets of variants for each of the other (non-reference) strains in the strain set, and we then applied another deduplication step to combine variants with a reciprocal overlap of at least 90%. We then used vg construct to build the *VCF graph* with the total set of variants and the linear reference genome.

### Construction of the *cactus graph*

The second graph (called the *cactus graph* throughout the paper) was constructed from a whole-genome alignment between the assemblies. First, the repeat-masked PacBio-assemblies of the strains in the strain set were aligned with our Cactus tool [27]. Cactus requires a phylogenetic tree of the strains which was estimated using Mash (version 2.1) [44] and PHYLIP (version 3.695) [45]. Subsequently, we converted the HAL format output file to a variation graph with hal2vg (<https://github.com/ComparativeGenomicsToolkit/hal2vg>).

### Genotyping of SVs

Prior to genotyping, we mapped the Illumina short reads of all 12 yeast strains to both graphs using vg map. We measured the fractions of reads mapped with specific properties using vg view and the JSON processor jq. Then, we applied toil-vg call (commit be8b6da) to genotype variants, obtaining a separate genotype set for each of the 11 non-reference strains on both graphs and for each of the two strain sets (in total  $11 \times 2 \times 2 = 44$  genotype sets). From the genotype sets, we removed variants smaller than 50 bp and variants with missing or homozygous reference genotypes. To evaluate the filtered genotype sets, we generated a sample graph (i.e., a graph representation of the genotype set) for each genotype set using vg construct and vg mod on the reference assembly *S.c. S288C* and the genotype set. Subsequently, we mapped short reads from the respective strains to each sample graph using vg map. We mapped the short reads also to an empty sample graph that was generated using vg construct as a graph representation of the linear reference genome. In an effort to restrict our analysis to SV regions, we removed reads that mapped equally well (i.e., with identical mapping quality and percent identity)

to all three graphs (the two sample graphs and the empty sample graph) from the analysis. These filtered out reads most likely stem from portions of the strains' genomes that are identical to the reference strain *S.c. S288C*. We analyzed the remaining alignments of reads from SV regions with vg view and jq.

### Supplementary information

**Supplementary information** accompanies this paper at <https://doi.org/10.1186/s13059-020-1941-7>.

**Additional file 1:** Table S1-S6, Figure S1-S18, Supplementary information.

**Additional file 2:** Review history.

### Acknowledgements

We thank Peter Audano for sharing the CHMPD dataset and for his assistance with SMRT-SV v2.

### Review history

The review history is available as Additional file 2.

### Peer review information

Andrew Cosgrove was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

### Funding

Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under Award Number U54HG007990 and U01HL137183. This publication was supported by a Subagreement from European Molecular Biology Laboratory with funds provided by Agreement No. 2U41HG007234 from National Institute of Health, NHGRI. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of National Institute of Health, NHGRI or European Molecular Biology Laboratory. The research was made possible by the generous financial support of the W.M. Keck Foundation (DT06172015).

JAS was further supported by the Carlsberg Foundation. DH was supported by the International Max Planck Research School for Computational Biology and Scientific Computing doctoral program. JE was supported by the Jack Baskin and Peggy Downes-Baskin Fellowship. AMN was supported by the National Institutes of Health (5U41HG007234), the W.M. Keck Foundation (DT06172015), and the Simons Foundation (SFLIFE# 35190).

### Availability of data and materials

The commands used to run the analyses presented in this study are available at [github.com/vgteam/sv-genotyping-paper](https://github.com/vgteam/sv-genotyping-paper) under a MIT license. The version of the code used in this manuscript has been deposited in the following Zenodo repository: <https://doi.org/10.5281/ZENODO.3517248> [46]. The scripts to generate the manuscript, including figures and tables, are available at [github.com/jmonlong/manu-vgsv](https://github.com/jmonlong/manu-vgsv).

The simulated reference sequence and reads were deposited at <https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=vgsv2019/simulation>.

The sequencing data for HG00514, HG00733, and NA19240 was downloaded from SRA: ERR903030 [47], ERR895347 [48], ERR894724 [49].

The haplotype-resolved SVs from the Human Genome Structural Variation Consortium [22] were deposited on the GitHub repository [46]: <https://github.com/vgteam/sv-genotyping-paper/tree/master/human/hgsvc>. The simulated reads for HG00514 using the HGSCV genotypes were deposited at <https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=vgsv2019/sim-HG00514-30x.fq.gz>.

The Genome in a bottle catalog [23] was downloaded from [ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST\\_UnionSVs\\_12122017/svanalyzer\\_union\\_171212\\_v0.5.0\\_annotated.vcf.gz](ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_UnionSVs_12122017/svanalyzer_union_171212_v0.5.0_annotated.vcf.gz). Sequencing data for HG002 came from [ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002\\_NA24385\\_son/NIST\\_HiSeq\\_HG002\\_](ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_)

Homogeneity-10953946/NHGRI\_Illumina300X\_AJtrio\_novoalign\_bams/HG002.hs37d5.300x.bam.

The SV catalog from Audano et al. [5] was prepared from [http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data\\_collections/hgsv\\_sv\\_discovery/working/20181025\\_EEE\\_SV-Pop\\_1/VariantCalls\\_EEE\\_SV-Pop\\_1/EEE\\_SV-Pop\\_1.ALL.sites.20181204.vcf.gz](http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/hgsv_sv_discovery/working/20181025_EEE_SV-Pop_1/VariantCalls_EEE_SV-Pop_1/EEE_SV-Pop_1.ALL.sites.20181204.vcf.gz).

The genotype predictions for each method benchmarked are available at <https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=vgsv2019/vcfs/>

The yeast assemblies were downloaded from ([https://yjx1217.github.io/Yeast\\_PacBio\\_2016/data/](https://yjx1217.github.io/Yeast_PacBio_2016/data/)) The sequencing reads were downloaded from SRA: SRR4074413 [50], SRR4074412 [51], SRR4074411 [52], SRR4074394 [53], SRR4074385 [54], SRR4074384 [55], SRR4074383 [56], SRR4074358 [57], SRR4074258 [58], SRR4074257 [59], SRR4074256 [60], SRR4074255 [61].

#### Authors' contributions

EG, AN, GH, JS, JE, and ED implemented the read mapping and variant calling in the vg toolkit. GH, DH, JM, JAS, and EG performed analysis on the different datasets. GH, DH, JM, and BP designed the study. GH, DH, and JM drafted the manuscript. All authors read, reviewed, and approved the final manuscript.

#### Authors' information

These authors contributed equally: Glenn Hickey, David Heller, Jean Monlong.

Twitter handles: Jean Monlong @jmonlong; Jouni Sirén @jtsiren; Eric Dawson @erictdawson; Erik Garrison @erikgarrison; Adam Novak @amnovak; Benedict Paten @benedictpaten.

#### Ethics approval and consent to participate

Not applicable.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>UC Santa Cruz Genomics Institute, University of California, Santa Cruz, California, USA. <sup>2</sup>Max Planck Institute for Molecular Genetics, Berlin, Germany. <sup>3</sup>Department of Genetics, University of Cambridge, Cambridge, UK. <sup>4</sup>Division of Cancer Epidemiology and Genetics, National Cancer Institute, Rockville, MD, USA.

Received: 31 May 2019 Accepted: 20 January 2020

Published online: 12 February 2020

#### References

- Chiang C, Scott AJ, Davis JR, Tsang EK, Li X, Kim Y, Hadzic T, Damani FN, Ganel L, GTEx Consortium, Montgomery SB, Battle A, Conrad DF, Hall IM. The impact of structural variation on human gene expression. *Nat Genet.* 2017;49(5):692–9. <https://doi.org/10.1038/ng.3834>.
- Weischenfeldt J, Symmons O, Spitz F, Korbel JO. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nat Rev Genet.* 2013;14(2):125–38. <https://doi.org/10.1038/nrg3373>.
- Chiang C, Layer RM, Faust GG, Lindberg MR, Rose DB, Garrison EP, Marth GT, Quinlan AR, Hall IM. SpeedSeq: ultra-fast personal genome analysis and interpretation. *Nat Methods.* 2015;12(10):966–8. <https://doi.org/10.1038/nmeth.3505>.
- Rausch T, Zichner T, Schlattl A, Stutz AM, Benes V, Korbel JO. DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics.* 2012 Sep 7;28(18):3333–9. <https://doi.org/10.1093/bioinformatics/bts378>.
- Audano PA, Sulovari A, Graves-Lindsay TA, Cantsilieris S, Sorensen M, Welch AE, Dougherty ML, Nelson BJ, Shah A, Dutcher SK, Warren WC, Magrini V, McGrath SD, Li Yi, Wilson RK, Eichler EE. Characterizing the major structural variant alleles of the human genome. *Cell.* 2019;176(3):663–675.e19. <https://doi.org/10.1016/j.cell.2018.12.019>.
- Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, Zhang Y, Ye K, Jun G, Hsi-Yang Fritz M, Konkil MK, Malhotra A, Stutz AM, Shi X, Paolo Casale F, Chen J, Hormozdiari F, Dayama G, Chen K, Malig M, MJP C, Walter K, Meiers S, Kashin S, Garrison E, Auton A, HYK L, Jasmine Mu X, Alkan C, Antaki D, Bae T, Cerveira E, Chines P, Chong Z, Clarke L, Dal E, Ding L, Emery S, Fan X, Gujral M, Kahveci F, Kidd JM, Kong Y, Lameijer E-W, McCarthy S, Flicek P, Gibbs RA, Marth G, Mason CE, Menelaou A, Muzny DM, Nelson BJ, Noor A, Parrish NF, Pendleton M, Quitadamo A, Raeder B, Schadt EE, Romanovitch M, Schlattl A, Sebra R, Shabalina AA, Untergrasser A, Walker JA, Wang M, Yu F, Zhang C, Zhang J, Zheng-Bradley X, Zhou W, Zichner T, Sebati J, Batzer MA, SA MC, The 1000 Genomes Project Consortium, Mills RE, Gerstein MB, Bashir A, Stegle O, Devine SE, Lee C, Eichler EE, Korbel JO. An integrated map of structural variation in 2,504 human genomes. *Nature.* 2015;526(7571):75–81. <https://doi.org/10.1038/nature15394>.
- Francioli LC, Menelaou A, Pulit SL, van Dijk F, Palamara PF, Elbers CC, Neerincx PBT, Ye K, Guryev V, Kloosterman WP, Deelen P, Abdellaoui A, van Leeuwen EM, van Oven M, Vermaat M, Li M, Laros JFJ, Karsen LC, Kanterakis A, Amin N, Hottenga JJ, Lameijer E-W, Kattenberg M, Dijkstra M, Bylles J, van Setten J, van Schaik BDC, Bot J, Nijman IJ, Renkens I, Marschall T, Schönhuth A, Hehir-Kwa JY, Handsaker RE, Polak P, Sohail M, Vuzman D, Hormozdiari F, van Enckevort D, Mei H, Koval V, Moed MH, van der Velde KJ, Rivadeneira F, Estrada K, Medina-Gomez C, Isaacs A, McCarroll SA, Beekman M, de Craen AJM, Suchiman HED, Hofman A, Oostra B, Uitterlinden AG, Willemsen G, Study LC, Platteel M, Veldink JH, van den Berg LH, Pitts SJ, Potluri S, Sundar P, Cox DR, Sunyaev SR, den Dunnen JT, Stoneking M, de Knijff P, Kayser M, Li Q, Li Y, Du Y, Chen R, Cao H, Li N, Cao S, Wang J, Bovenberg JA, Pe'er I, Slagboom PE, van Duijn CM, Boomsma DI, van Ommen G-JB, de Bakker PIW, Swertz MA, Wijmenga C, The Genome of the Netherlands Consortium. Whole-genome sequence variation, population structure and demographic history of the Dutch population. *Nat Genet.* 2014;46(8):818–25. <https://doi.org/10.1038/ng.3021>.
- Chaisson MJP, Huddleston J, Dennis MY, Sudmant PH, Malig M, Hormozdiari F, Antonacci F, Surti U, Sandstrom R, Boitano M, Landolin JM, Stamatoyannopoulos JA, Hunkapiller MW, Korlach J, Eichler EE. Resolving the complexity of the human genome using single-molecule sequencing. *Nature.* 2014;517(7536):608–11. <https://doi.org/10.1038/nature13907>.
- Huddleston J, Chaisson MJ, Steinberg KM, Warren W, Hoekzema K, Gordon D, Graves-Lindsay TA, Munson KM, Kronenberg ZN, Vives L, Peluso P, Boitano M, Chin C-S, Korlach J, Wilson RK, Eichler EE. Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Res.* 2016;27(5):677–85. <https://doi.org/10.1101/gr.214007.116>.
- Cretu Stancu M, van Roosmalen MJ, Renkens I, Nieboer MM, Middelkamp S, de Ligt J, Pregno G, Giachino D, Mandrile G, Espejo Valle-Inclan J, Korzelius J, de Bruijn E, Cuppen E, Talkowski ME, Marschall T, de Ridder J, Kloosterman WP. Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *Nat Commun.* 2017;8(1) <https://doi.org/10.1038/s41467-017-01343-4>.
- Spies N, Weng Z, Bishara A, McDaniel J, Catoe D, Zook JM, Salit M, West RB, Batzoglou S, Sidow A. Genome-wide reconstruction of complex structural variants using read clouds. *Nat Meth.* 2017;14(9):915–20. <https://doi.org/10.1038/nmeth.4366>.
- Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Tyson JR, Beggs AD, Dilthey AT, Fiddes IT, Malla S, Marriott H, Nieto T, O'Grady J, Olsen HE, Pedersen BS, Rhie A, Richardson H, Quinlan AR, Snutch TP, Tee L, Paten B, Phillippy AM, Simpson JT, Loman NJ, Loose M. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat Biotechnol.* 2018; 36(4):338–45. <https://doi.org/10.1038/nbt.4060>.
- Chin C-S, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, Dunn C, O'Malley R, Figueroa-Balderas R, Morales-Cruz A, Cramer GR, Delledonne M, Luo C, Ecker JR, Cantu D, Rank DR, Schatz MC. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat Methods.* 2016;13(12):1050–4. <https://doi.org/10.1038/nmeth.4035>.
- Paten B, Novak AM, Eizenga JM, Garrison E. Genome graphs and the evolution of genome inference. *Genome Res.* 2017;27(5):665–76. <https://doi.org/10.1101/gr.214155.116>.
- Garrison E, Sirén J, Novak AM, Hickey G, Eizenga JM, Dawson ET, Jones W, Garg S, Markello C, Lin MF, Paten B, Durbin R. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat Biotechnol.* 2018 Aug 20;36(9):875–9. <https://doi.org/10.1038/nbt.4227>.
- Novak AM, Hickey G, Garrison E, Blum S, Connelly A, Dilthey A, Eizenga J, Elmohamed MAS, Guthrie S, Kahles A, Keenan S, Kelleher J, Kural D, Li H, Lin MF, Miga K, Ouyang N, Rakocevic G, Smuga-Otto M, Zaranek AW, Durbin R, McVean G, Haussler D, Paten B. Genome Graphs. *bioRxiv.* 2017;18:101378. <https://doi.org/10.1101/101378>.
- Rakocevic G, Semenyuk V, Lee W-P, Spencer J, Browning J, Johnson IJ, Arsenijevic V, Nadj J, Ghose K, Suci MC, Ji S-G, Demir G, Li L, Toptaş BÇ,

- Dolgoborodov A, Pollex B, Spulber I, Glotova I, Kómár P, Stachyra AL, Li Y, Popovic M, Källberg M, Jain A, Kural D. Fast and accurate genomic analyses using genome graphs. *Nat Genet.* 2019; <https://doi.org/10.1038/s41588-018-0316-4>.
18. Eggertsson HP, Jonsson H, Kristmundsdóttir S, Hjartarson E, Kehr B, Masson G, Zink F, Hjorleifsson KE, Jonasdóttir A, Jonasdóttir A, Jonsdóttir I, Gudbjartsson DF, Melsted P, Stefansson K, Halldorsson BV. Graph typer enables population-scale genotyping using pangenome graphs. *Nat Genet.* 2017;49(11):1654–60. <https://doi.org/10.1038/ng.3964>.
  19. Sibbesen JA, Maretty L. The Danish Pan-Genome Consortium, Krogh A. Accurate genotyping across variant classes and lengths using variant graphs. *Nat Genet.* 2018;50(7):1054–9. <https://doi.org/10.1038/s41588-018-0145-5>.
  20. Chen S, Krusche P, Dolzhenko E, Sherman RM, Petrovski R, Schlesinger F, Kirsche M, Bentley DR, Schatz MC, Sedlazeck FJ, Eberle MA. Paragraph: A graph-based structural variant typer for short-read sequence data. *bioRxiv.* 2019;24635011. <https://doi.org/10.1101/635011>.
  21. Zook JM, McDaniel J, Olson ND, Wagner J, Parikh H, Heaton H, Irvine SA, Trigg L, Truty R, McLean CY, De La Vega FM, Xiao C, Sherry S, Salit M. An open resource for accurately benchmarking small variant and reference calls. *Nat Biotechnol.* 2019;37(5):561–6. <https://doi.org/10.1038/s41587-019-0074-6>.
  22. Chaisson MJ, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, Gardner EJ, Rodriguez OL, Guo L, Collins RL, Fan X, Wen J, Handsaker RE, Fairley S, Kronenberg ZN, Kong X, Hormozdiari F, Lee D, Wenger AM, Hastie AR, Antaki D, Anantharaman T, Audano PA, Brand H, Cantsilieris S, Cao H, Cerveira E, Chen C, Chen X, Chin C-S, Chong Z, Chuang NT, Lambert CC, Church DM, Clarke L, Farrell A, Flores J, Galeev T, Gorkin DU, Gujral M, Guryev V, Heaton WH, Korlach J, Kumar S, Kwon JY, Lam ET, Lee JE, Lee J, Lee W-P, Lee SP, Li S, Marks P, Viaud-Martinez K, Meiers S, Munson KM, Navarro FCP, Nelson BJ, Nodzak C, Noor A, Kyriazopoulou-Panagiotopoulou S, Pang AWC, Qiu Y, Rosanio G, Ryan M, Stütz A, Spierings DCJ, Ward A, Welch AE, Xiao M, Xu W, Zhang C, Zhu Q, Zheng-Bradley X, Lowy E, Yakneen S, McCarroll S, Jun G, Ding L, Koh CL, Ren B, Flicek P, Chen K, Gerstein MB, Kwok P-Y, Lansdorp PM, Marth GT, Sebat J, Shi X, Bashir A, Ye K, Devine SE, Talkowski ME, Mills RE, Marschall T, Korbel JO, Eichler EE, Lee C. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun.* Apr. 2019;16, 10(1) <https://doi.org/10.1038/s41467-018-08148-z>.
  23. Zook JM, Hansen NF, Olson ND, Chapman LM, Mullikin JC, Xiao C, Sherry S, Koren S, Phillippy AM, Boutros PC, SME S, Huang V, Rouette A, Alexander N, Mason CE, Hajirasouliha I, Ricketts C, Lee J, Tearle R, Fiddes IT, Barrio AM, Wala J, Carroll A, Ghaffari N, Rodriguez OL, Bashir A, Jackman S, Farrell JJ, Wenger AM, Alkan C, Soylev A, Schatz MC, Garg S, Church G, Marschall T, Chen K, Fan X, English AC, Rosenfeld JA, Zhou W, Mills RE, Sage JM, Davis JR, Kaiser MD, Oliver JS, Catalano AP, Chaisson MJ, Spies N, Sedlazeck FJ, Salit M, the Genome in a Bottle Consortium. A robust benchmark for germline structural variant detection. *bioRxiv.* 2019; <https://doi.org/10.1101/664623>.
  24. Paten B, Eizenga JM, Rosen YM, Novak AM, Garrison E, Hickey G. Superbubbles, Ultrabubbles, and cacti. *J Comput Biol.* 2018;25(7):649–63. <https://doi.org/10.1089/cmb.2017.0251>.
  25. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Weng Z, Liu Y, Mason CE, Alexander N, Henaff E, McIntyre AB, Chandramohan D, Chen F, Jaeger E, Moshrefi A, Pham K, Stedman W, Liang T, Saghbini M, Dzakula Z, Hastie A, Cao H, Deikus G, Schadt E, Sebra R, Bashir A, Truty RM, Chang CC, Gulbahce N, Zhao K, Ghosh S, Hyland F, Fu Y, Chaisson M, Xiao C, Trow J, Sherry ST, Zaranek AW, Ball M, Bobe J, Estep P, Church GM, Marks P, Kyriazopoulou-Panagiotopoulou S, Zheng GX, Schnall-Levin M, Ordóñez HS, Mudivarti PA, Giorda K, Sheng Y, Rypdal KB, Salit M. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci Data.* 2016;3:160025. <https://doi.org/10.1038/sdata.2016.25>.
  26. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv.* arXiv. 2013; Report No.: 1303.3997v2. <http://arxiv.org/abs/1303.3997v2>.
  27. Paten B, Earl D, Nguyen N, Diekhans M, Zerbino D, Haussler D. Cactus: algorithms for genome multiple sequence alignment. *Genome Res.* 2011; 21(9):1512–28. <https://doi.org/10.1101/gr.123356.111>.
  28. Yue J-X, Li J, Aigrain L, Hallin J, Persson K, Oliver K, Bergström A, Coupland P, Warringer J, Lagomarsino MC, Fischer G, Durbin R, Liti G. Contrasting evolutionary genome dynamics between domesticated and wild yeasts. *Nat Genet.* 2017;49(6):913–24. <https://doi.org/10.1038/ng.3847>.
  29. Nattestad M, Schatz MC. Assemblytics: a web analytics tool for the detection of variants from an assembly. *Bioinformatics.* 2016;32(19):3021–3. <https://doi.org/10.1093/bioinformatics/btw369>.
  30. Liu S, Huang S, Rao J, Ye W. The Genome Denmark Consortium, Krogh A, Wang J. Discovery, genotyping and characterization of structural variation and novel sequence at single nucleotide resolution from de novo genome assemblies on a population scale. *GigaSci.* 2015;4(1) <https://doi.org/10.1186/s13742-015-0103-4>.
  31. Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics.* 2018;34(18):3094–100. <https://doi.org/10.1093/bioinformatics/bty191>.
  32. The Pancancer Analysis of Whole Genomes (PCAWG). <https://dcc.icgc.org/pcawg/>.
  33. Genomics England 100,000 Genomes Project. <https://www.genomicsengland.co.uk/>.
  34. Whole Genome Sequencing in the NHLBI Trans-Omics for Precision Medicine (TOPMed). <https://www.nhlbiwgs.org/>.
  35. Maretty L, Jensen JM, Petersen B, Sibbesen JA, Liu S, Villesen P, Skov L, Belling K, Theil Have C, JMG I, Grosjean M, Bork-Jensen J, Grove J, Als TD, Huang S, Chang Y, Xu R, Ye W, Rao J, Guo X, Sun J, Cao H, Ye C, van Beusekom J, Espeseth T, Flindt E, Friberg RM, Halager AE, Le Hellard S, Hultman CM, Lescai F, Li S, Lund O, Løngren P, Mailund T, Matey-Hernandez ML, Mors O, CNS P, Sicheritz-Pontén T, Sullivan P, Syed A, Westergaard D, Yadav R, Li N, Xu X, Hansen T, Krogh A, Bolund L, TIA S, Pedersen O, Gupta R, Rasmussen S, Besenbacher S, Børjlum AD, Wang J, Eiberg H, Kristiansen K, Brunak S, Schierup MH. Sequencing and de novo assembly of 150 genomes from Denmark as a population reference. *Nature.* 2017;548(7665): 87–91. <https://doi.org/10.1038/nature23264>.
  36. Vivian J, Rao AA, Nothaft FA, Ketchum C, Armstrong J, Novak A, Pfeil J, Narkizian J, Deran AD, Musselman-Brown A, Schmidt H, Amstutz P, Craft B, Goldman M, Rosenbloom K, Cline M, O'Connor B, Hanna M, Birger C, Kent WJ, Patterson DA, Joseph AD, Zhu J, Zaranek S, Getz G, Haussler D, Paten B. Toil enables reproducible, open source, big biomedical data analyses. *Nat Biotechnol.* 2017 Apr;35(4):314–6. <https://doi.org/10.1038/nbt.3772>.
  37. Bcftools 1.9.
  38. Smit A, Hubley R, Green P. RepeatMasker Open-4.0.
  39. DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, Rivas MA, Hanna M, McKenna A, Fennell TJ, Kernysky AM, Sivachenko AY, Cibulskis K, Gabriel SB, Altshuler D, Daly MJ. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet.* 2011;43(5):491–8. <https://doi.org/10.1038/ng.806>.
  40. Rimmer APhan H, Mathieson I, Iqbal Z, Twigg SRF, WGS500 Consortium, AOM W, McVean G, Lunter G. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat Genet.* 2014;46(8):912–8. <https://doi.org/10.1038/ng.3036>.
  41. Kokot M, Długosz M, Deorowicz S. KMC 3: counting and manipulating k-mer statistics. *Bioinformatics.* 2017;33(17):2759–61. <https://doi.org/10.1093/bioinformatics/btx304>.
  42. Sirén J, Garrison E, Novak AM, Paten B, Durbin R. Haplotype-aware graph indexes. *Bioinformatics.* 2019; <https://doi.org/10.1093/bioinformatics/btz575>.
  43. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics.* 2010;26(6):841–2. <https://doi.org/10.1093/bioinformatics/btq033>.
  44. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.* 2016;20, 17(1) <https://doi.org/10.1186/s13059-016-0997-x>.
  45. Felsenstein J. PHYLIP - phylogeny inference package (version 3.2). *Cladistics.* 1989;5:164–6.
  46. Hickey G, Monlong J, Heller D, Sibbesen JA, Sirén J, Eizenga J, Dawson ET, Garrison E, Novak AM, Paten B. *vgteam/sv-genotyping-paper* Github. 2019; <https://doi.org/10.5281/ZENODO.3517248>.
  47. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=ERR903030>
  48. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=ERR895347>
  49. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=ERR894724>
  50. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074413>
  51. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074412>

52. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074411>
53. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074394>
54. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074385>
55. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074384>
56. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074383>
57. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074358>
58. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074258>
59. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074257>
60. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074256>
61. NCBI Sequence Read Archive, <https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR4074255>

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

