vcfdist: Accurately benchmarking phased small variant calls in human genomes

Tim Dunn[1*] and Satish Narayanasamy[1]

[1]Computer Science and Engineering, University of Michigan, 2260 Hayward Street, Ann Arbor, 48109, Michigan, USA.

*Corresponding author. E-mail: timdunn@umich.edu;
Contributing authors: nsatish@umich.edu;

Abstract

Accurately benchmarking small variant calling accuracy is critical for the continued improvement of human whole genome sequencing. In this work, we show that current variant calling evaluations are biased towards certain variant representations and may misrepresent the relative performance of different variant calling pipelines. We propose solutions, first exploring the affine gap parameter design space for complex variant representation and suggesting a standard. Next, we present our tool "vcfdist" and demonstrate the importance of enforcing local phasing for evaluation accuracy. We then introduce the notion of partial credit for mostly-correct calls and present an algorithm for clustering dependent variants. Lastly, we motivate using alignment distance metrics to supplement precision-recall curves for understanding variant calling performance. We evaluate the performance of 64 phased "Truth Challenge V2" submissions and show that vcfdist improves measured (SNP, INDEL) performance consistency across variant representations from $\boldsymbol{R^2} = \mathbf{(0.14542, 0.97243)}$ for vcfeval to $\mathbf{(0.99999, 0.99996)}$ for vcfdist.

The first draft reference of the human genome was assembled with great difficulty in 2000 at an estimated cost of $300 million (1). Following this massive effort, it became standard practice for whole genome sequencing (WGS) analyses to align sequencing reads to this reference to determine an individual's genome (2). Because there is only around 0.1% difference in the genomic sequence of two individuals when excluding structural variants (SVs) (3; 4),

the final sequence is stored as a set of variations from the reference genome and reported in variant call format (VCF). These small germline variants (under 50 base pairs) are then classified as either single-nucleotide polymorphisms (SNPs) or insertions/deletions (INDELs). Any variants which cannot be represented as a single SNP or INDEL are called "complex".

The past decade has been a time of rapid advancement in DNA sequencing chemistries, machine learning, and bioinformatics algorithms (5; 6). In such an environment, the ability to accurately compare the performance of variant calling pipelines is crucial. Firstly, accurate comparisons are necessary for identifying the current best-performing variant calling pipeline; this choice will impact real clinical decisions (7). Secondly, researchers must be able to correctly identify promising avenues of further research. Thirdly, the curation of high-quality databases of germline mutations is only made possible with accurate comparison techniques. These databases are later used for linking genotypes with clinically-relevant phenotypes (4; 8).

The current standard benchmarking tool for variant calling is vcfeval, introduced by Real Time Genomics (RTG) in 2015 (9) and later backed by the Global Alliance for Genomics and Health (GA4GH) Benchmarking Team (10). It is now widely accepted as the standard tool for benchmarking small variant calls (10; 11; 12; 13). vcfeval uses a clever pruned search algorithm to determine the largest possible matching subsets of query and truth variants. Regardless of the query and truth variant representations, vcfeval is able to determine equivalency as long as there is a complete exact match. A noteworthy extension is VarMatch, which uses a similar algorithm but implements several different optimization criteria and presents an algorithm to partition the inputs into smaller subproblems without a loss of accuracy (14).

Comparing a set of variant calls (a "query" callset) to a known "truth" callset is a deceptively difficult problem, arising from the fact that there are many possible representations for the same set of variants. Prior works have invested significant effort in defining one accepted representation for a given variant. This process of variant normalization requires that variants are decomposed, trimmed, and left-shifted (14; 15); an example can be seen in Supplementary Figure S4. Although normalization is sufficient to ensure there is one possible representation for a single variant, there is not a unique representation for multiple adjacent ("complex") variants. This makes evaluating variant calling accuracy difficult (13).

Despite its impressive advancement over previous state of the art algorithms for variant comparison, vcfeval suffers from several key limitations which we attempt to address in this paper. Each limitation increases the extent to which vcfeval's results depend upon the initial representation of variants in the query and truth VCFs. Firstly, vcfeval has no notion of partial credit and will not report any true positives if no subsets of query variants match exactly with truth variants. Figure 1 demonstrates a simplified example of where different

variant representations of the same sequence lead to significantly different measured accuracies. Refer to Supplementary Figure S1 for a realistic example. Secondly, vcfeval outputs precision-recall curves for SNP and INDEL variants separately. Although precision and recall are concise metrics for understanding approximate variant calling accuracy, they are dependent on both the reference FASTA and query VCF representation. For complex variants, especially in low-complexity (repetitive) regions, the same query sequence can be represented in numerous equally valid ways with a differing total count of SNPs and INDELs (see Figures 1a and 2b). Thirdly, vcfeval was designed to handle unphased query variants, and allows arbitrary local phasing of adjacent heterozygous variants. Allowing different phasings leads to different sets of possible query sequences depending on the original variant representation. While this would be relatively easy to change within vcfeval, enforcing phase is not currently a supported option (9).

In combination, these characteristics bias evaluations towards a particular variant representation that uses many SNPs rather than a few INDELs to represent genomic variation. This bias becomes most evident in high-quality variant callsets where most remaining variant calling errors occur in low-complexity or high-variance regions. As read sequencing technologies have recently improved in terms of both average read length and quality, we are now able to call variants in more difficult regions than ever before (16). INDELs are orders of magnitude more likely to occur in low-complexity repetitive regions (17), and so it is important that variant calling evaluation is not biased against newer tools which allow copy number variation and other common INDELs. Lastly, vcfeval suffers from a few practical constraints such as high memory usage and the inability to evaluate large clusters with many nearby variants due to its exponential complexity (14; 9). Because these benchmarking issues occur during VCF evaluation, they are broadly applicable to all sequencing technologies.

In this work we present vcfdist, an alignment-based small variant calling evaluator that standardizes query and truth VCF variants to a consistent representation, requires local phasing of both input VCFs, and gives partial credit to variant calls which are mostly (but not exactly) correct. We show that the SNP and INDEL precision-recall curves reported by vcfdist are stable across variant representations. Furthermore, we introduce alignment distance based metrics for evaluation which are entirely independent of variant representation, and only measure the distance between the final diploid truth and query sequences. We then introduce a novel variant clustering algorithm which reduces downstream computation while also discovering long range variant dependencies missed by previous approaches.

We evaluate all 64 submissions from the precisionFDA's 2020 variant calling "Truth Challenge V2" using both vcfeval and vcfdist on two high-quality ground truth datasets: the NIST v4.2.1 WGS benchmark, and the "Challenging Medically Relevant Genes" (CMRG) dataset. We find that vcfdist improves

(a)

Ref. `ACCCTTTTTTG` **Query** `ACCTTTG` **Truth** `ACCCTTTG`

Query VCF Representation 1

POS	REF	ALT
3	CCTTT	C

Query VCF Representation 2

POS	REF	ALT
1	AC	A
4	CTTT	C

Truth VCF

POS	REF	ALT
4	CTTT	C

(b)

vcfeval Summary Statistics

	TP	FP	FN	PP	Prec.	Recall	F1	F1 Q-score
Query Repr. 1	0	1	1	0	0.00	0.00	0.00	0.00
Query Repr. 2	1	1	0	0	0.50	1.00	0.67	4.77

(c)

vcfdist Summary Statistics

	TP	FP	FN	PP	Prec.	Recall	F1	F1 Q-score
Query Repr. 1	0	0	0	1	0.67	0.67	0.67	4.77
Query Repr. 2	1	1	0	0	0.50	1.00	0.67	4.77

(d)

vcfdist Distance Summary

	ED	DE	DE Q-score	ED Q-score	ALN Q-Score
Reference	3	1			
Query Repr. 1	1	1	4.77	0.00	3.01
Query Repr. 2	1	1	4.77	0.00	3.01

Fig. 1 A simple example of vcfeval and vcfdist evaluations, demonstrating vcfeval's dependence on variant representation and the usefulness of partial credit. **(a)** Reference, query, and truth sequences, as well as the query and truth VCFs. **(b)** vcfeval and **(c)** vcfdist count of true positive, false positive, false negative, and partial positive variants, as well as the calculated precision, recall, and F1 scores. Note that although both query VCF variant calls result in the exact same query sequence, the summary statistics differ. Partial credit alleviates this problem for vcfdist. **(d)** Distance-based summary statistics reported by vcfdist: edit distance, distinct edits, and alignment distance, which are independent of variant representation. An explanation of these summary statistics can be found in the *Online Methods* section.

measured SNP and INDEL performance consistency across variant representations from vcfeval's $R^2 = 0.14542$ and 0.97243 to $R^2 = 0.99999$ and 0.99996, respectively. We note that vcfeval's SNP R^2 value is only so poor (0.14542) because one of our selected representations (design point A in Figure 2) never uses SNPs to represent variants (see Figure 6a). Excluding design point A, vcfeval's SNP R^2 value is 0.96888; however, we believe this design point is important to include in our analyses for reasons discussed later.

Results

The affine gap design space for selecting variant representations. As demonstrated in Figure 1, the main issue with a difference-based format such as VCF is that often there are multiple reasonable sets of variant calls that can be used to represent the same final sequence relative to a reference

FASTA. Since DNA sequencing only measures the final sequence, there is no way of knowing which set of variants physically occurred. We can only select a representation which contains the "most likely" set of variants, based on the relative likelihoods of various mutations occurring.

This problem of variant representation can be viewed as a pairwise global alignment problem, and the path of the alignment with the minimum penalty score can be used to derive an edit path, representing the most likely set of variants. Allowed operations include matching, substituting, inserting and deleting bases with corresponding penalties m, x, $g(n)$, and $g(n)$. Under the gap-affine model originally proposed in (18), $g(n) = o + ne$, where n is the length of the gap, o is a gap-opening penalty, and e is a gap-extension penalty. The relative value of substitution, insertion, and deletion penalties is critical, as it determines which variant representation is selected in the VCF.

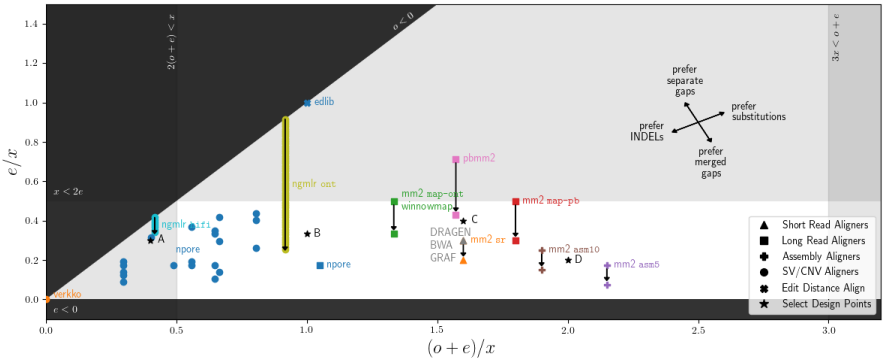
Recent efforts such as (19) and (20) have demonstrated how to transform gap cost models into other equivalent representations. In Figure 2a we normalize penalties such that $m = 0$ and then explore the design space for a general-purpose aligner with an affine-gap cost model. Black shaded areas represent the invalid areas where $o < 0$ or $e < 0$; opening and extending gaps should be penalized, not preferred over matching bases. Gray shaded areas represent parameter selections that poorly reflect the approximate likelihood of certain events in the general case. For example, when $2(o + e) < x$, a substitution will instead always be represented as a single-base insertion and a single-base deletion. Note that although these grayed parameter ranges are not ideal for a general-purpose aligner, configurations such as $2(o+e) < x$ may make sense in repetitive regions of the genome where copy number variants are likely.

Figure 2a plots the default parameter configurations for the most commonly used aligners in the 2020 pFDA variant calling challenge. This plot also includes the default parameters for a wider range of tools, such as those used for structural variant (SV) detection (verkko, NGMLR) (21; 22), copy number variant (CNV) detection (nPoRe) (17), edit distance calculation (edlib) (23), and assembly alignment (minimap2's `asm5` and `asm10` configurations) (24). The original and normalized affine-gap parameters for each tool configuration are included in Supplementary Figure S6.

The various configurations for minimap2 are depicted with two data points and a transition, since minimap2 uses double affine gap alignment (24). Under double affine gap alignment, there are two separate gap-opening (o_1, o_2) and gap-extension (e_1, e_2) penalties, and the minimum of $g_1(n) = o_1 + ne_1$ and $g_2(n) = o_2 + ne_2$ is preferred. Typically, $o_2 > o_1$ and $e_2 < e_1$. As a result, the penalties will be equal at $n = \frac{o_2 - o_1}{e_1 - e_2}$; prior to this point, $g_1 < g_2$ and afterwards $g_2 < g_1$. Thus g_1 determines the penalty for a short gap and g_2 determines the penalty for a long gap.

Any tools using homopolymer (HP) compression or simple tandem repeat (STR) compression such as verkko (21) are plotted at $(0, 0)$ in Figure 2a,

(a)



(b)

Reference `AGGCGACA` Query `ATACCGAGCTTA`

Point A	Point B	Point C	Point D
$m, x, o, e = 0, 10, 1, 3$	$m, x, o, e = 0, 3, 2, 1$	$m, x, o, e = 0, 5, 6, 2$	$m, x, o, e = 0, 5, 9, 1$

Alignment

```
AGG---CGA-C--A      A-GGCGA-C--A      A-GGCGA---CA      AGGCGAC-------A
A--TACCGAGCTTA      ATACCGAGCTTA      ATACCGAGCTTA      A---TACCGAGCTTA
```

VCF

POS	REF	ALT	POS	REF	ALT	POS	REF	ALT	POS	REF	ALT
1	AGG	A	1	A	AT	1	A	AT	1	AGGC	A
3	G	GTAC	2	G	A	2	G	A	5	G	T
6	A	AG	3	G	C	3	G	C	7	C	CCGAGCTT
7	C	CTT	6	A	AG	6	A	AGCT			
			7	C	CTT	7	C	T			

Fig. 2 **(a)** The design space for an affine-gap aligner with match, mismatch, gap opening, and gap extension penalties m, x, o, and e. This plot includes the aligners used in pFDA's Truth Challenge V2, as well as assembly, edit distance, and CNV/SV aligners for comparison. All parameters have been normalized so that $m = 0$ (see Supplementary Figure S6 for details). **(b)** Aligning the same query and reference sequence using different gap parameters (design points A, B, C, and D) results in different sets of reported variants.

since compressed gaps are not penalized in repetitive regions during alignment. Repetitive sequence compression is common practice for long-read de novo assemblers (21; 25; 26), since incorrect estimation of homopolymer run length is the dominant error mode of nanopore sequencing (27). nPoRe penalizes copy number variation in these repetitive regions only slightly, depending on the copy number and repeat unit length (17). This results in many different points on the left side of Figure 2a. NGMLR, an aligner for structural variant detection, also does not heavily penalize INDELs (22). In contrast to minimap2, NGMLR uses a smoother convex gap cost model, decreasing the penalty for each additional gap extension as the length of the gap grows. As

a result, NGMLR's gap model is represented in Figure 2a as a smooth gap extension penalty decrease.

In Figure 2a, design point A was selected as the approximate centroid of the CNV/SV aligners. Likewise, point C was selected as the approximate centroid of common short and long read alignment parameter configurations, and point D was selected for assembly aligners. Lastly, point B was selected because this design point will simultaneously minimize the edit distance (ED) and number of distinct edits (DE) in the variant representation, weighting DE twice as heavily as ED. It is also the approximate midpoint of design points A and C. All selected points were chosen such that their substitution and gap penalties can be represented by small positive integer constants (see Figure 2b).

A standard complex variant representation: design point C. As Figure 2b shows, for complex variants there are many reasonable representations which depend upon the selected affine gap alignment parameters. By varying alignment penalties m, x, o, and e, Figure 2b shows that aligning the same query and reference sequence will result in a different VCF representation at each of the four selected design points A, B, C, and D. If we define a standard set of affine gap parameters, however, then there is usually just one possible representation for complex variants such as this.

We propose using design point C, where $(m, x, o, e) = (0, 5, 6, 2)$, as the standard set of affine gap parameters for reporting variants. When normalized to $m = 0$ and $x = 1$, $C = (0, 1, 1.2, 0.4)$. Design point C was selected because it is the approximate centroid of widely-accepted parameters used to align reads of three different popular sequencing technologies (Illumina short reads, PacBio HiFi reads, and Oxford Nanopore long reads). For Illumina short reads, DRAGEN, BWA, GRAF, and minimap2 `sr` use the same normalized alignment parameters $(0, 1, 1.3, 0.3)$ (Supplementary Figure S6). For ONT long reads, minimap2 `map-ont` and winnowmap use $(0, 1, 0.833, 0.5)$. For PacBio HiFi reads, minimap2 `map-pb` uses $(0, 1, 1.3, 0.5)$ and pbmm2 uses $(0, 1, 0.857, 0.714)$. Of the 53 submission methodologies documented from the pFDA Truth Challenge V2, 52 aligned reads using one of these above-mentioned gap parameter configurations, all of which are relatively similar. The other aligner used was NovoAlign, whose relative alignment penalties depend upon base quality (28).

INDEL/SNP precision and recall depend on variant representation. Variant calling accuracy is currently evaluated by measuring separately the SNP and INDEL precision-recall curves, and then reporting the precision and recall where the F1 score is maximized (9; 10; 11; 12). This metric is useful because it gives an intuitive overview of what percentage of variants were called correctly and incorrectly for SNPs and INDELs separately. Figures 3a and 3b, however, show that precision-recall curves can change significantly depending on variant representation, even when the evaluated query and truth sequences are the exact same. This is partially because the total number of SNP and

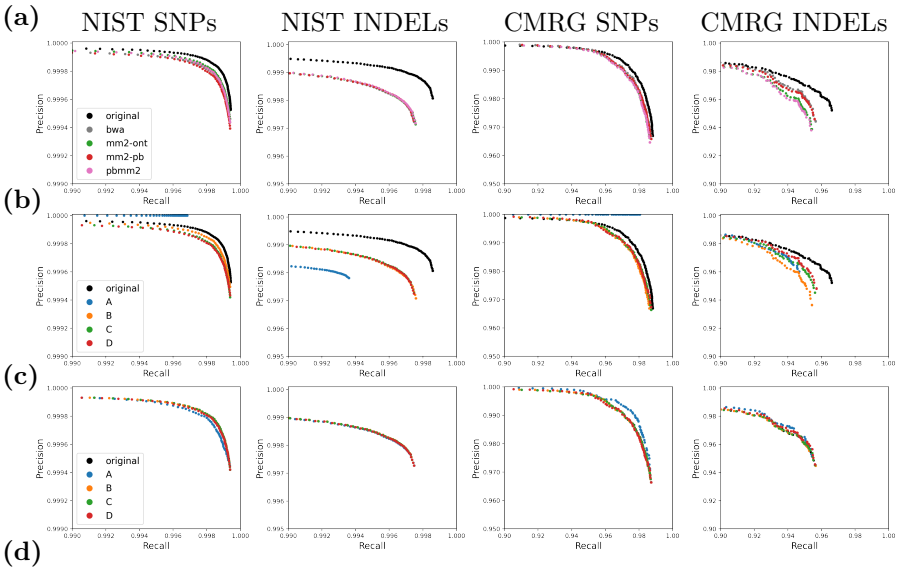
INDEL variants reported in a VCF will vary depending on the selected variant representation.

This issue is highlighted in Figure 3b by design point A, which represents all SNPs as a 1-base insertion and 1-base deletion (because $2(o + e) < x$; see Figure 2a). Since VCFs represented using point A report no SNPs, any false positives will be categorized as INDEL FPs. True positive variants will be SNP TPs when the truth VCF contains a SNP and INDEL TPs when the truth VCF contains an INDEL. This results in query VCFs at point A having perfect SNP precision, but a lower INDEL precision than other representations. Similar problems exist for other variant representations, although to a lesser extent. The stabilized precision-recall curves following vcfdist's variant standardization can be seen in Figures 3c and 4d.

INDELs are common in repetitive regions: design point A. Demonstrating that standardizing query VCFs using design point A changes the resulting evaluation may initially appear to be a contrived argument, since point A is not similar to any aligners used in the pFDA Truth Challenge V2. Figure 3a shows little difference in the performance of VCFs represented using common aligner parameters such as BWA, minimap2 `map-ont`, minimap2 `map-pb`, and pbmm2 (24; 29; 30). This is because all of these general-purpose read aligners were designed for aligning reads in non-repetitive regions, to which most bioinformatics analyses have been restricted until recent years (31; 32). As such, their parameters are clustered in a small region of the affine gap penalty design space in Figure 2a. We selected design point A precisely because its low gap penalties model how alignment occurs in repetitive regions by dedicated copy number variant (CNV) aligners or any aligners performing homopolymer (HP) or simple tandem repeat (STR) compression.

As the field shifts towards true whole genome evaluation, we expect variant callsets to frequently include merged results from general-purpose aligners/callers in non-repetitive regions and CNV/SV aligners/callers in more repetitive regions. In low-complexity regions, CNVs and other small gaps are orders of magnitude more likely than elsewhere (17), necessitating the inclusion of variants whose representation falls of the left-hand side of Figure 2a near design point A. As can be seen from Figure 2a, several existing SV and CNV callers already occupy this space. When these repetitive INDEL callsets are merged with callsets from non-repetitive regions, we must ensure that our evaluation of variant calling accuracy remains unbiased.

Correct local phasing of dependent variants is critical. Incorrect local phasing of heterozygous variants leads to an entirely different sequence of bases for both haplotypes, which may impact clinical decisions (33). Despite the relative ease of local phasing given new long read sequencing technologies, vcfeval does not enforce that evaluated VCFs contain phasing information. In fact, vcfeval discards this information when available, considering any possible local phasing to be correct (9). To illustrate why this is a problem, the original query



(d)

Original Query VCF

CHROM	POS	REF	ALT	GT
chr1	19672401	TTCC	T	0|1
chr1	19672410	C	A	0|1
chr1	19672411	T	G	0|1
chr1	19672412	C	A	0|1
chr1	19672413	A	AGAG,G	1|2

Query VCF, standardized at C

CHROM	POS	REF	ALT	GT
chr1	19672401	TTCCTCC	T	0|1
chr1	19672413	A	AGAG	1|1

Truth VCF

CHROM	POS	REF	ALT	GT
chr1	19672401	TTCC	T	1|1
chr1	19672413	A	AGAG	1|1

Original Query VCF Summary

4 SNP TP, 2 INDEL TP

Query VCF at C Summary

1 INDEL TP, 1 INDEL FP, 1 INDEL FN

Fig. 3 vcfeval precision-recall plots for Truth Challenge V2 submission `K4GT3` on the NIST and CMRG datasets. **(a)** Evaluating the original query VCF and after changing the query variant representation using the alignment parameters of common aligners (see Figure 2). **(b)** Evaluating the original query VCF and after changing the query variant representation to design points A, B, C, and D (see Figure 2). **(c)** Standardizing the five representations from (b) using vcfdist prior to evaluating with vcfeval dramatically improves consistency. **(d)** A real example demonstrating why the original `K4GT3` query VCF appears to significantly outperform other representations in (a) and (b). Because vcfeval discards query phasing information and allows any possible local phasing, the original fractured variant representation is considered entirely correct whereas the more succinct standardized representation at C is not.

VCF representation shown in Figure 3d unnecessarily fragments a single heterozygous variant into five heterozygous variants. For these 5 variants, there are now $2^5 = 32$ possible local phasings, each resulting in different sequences. Perhaps unsurprisingly, one of these 32 phasings (and not the phasing initially reported) results in a match with the ground truth, and all variants are considered true positives. This serves to heavily bias evaluations towards fragmented query variant representations.

In contrast, vcfdist requires locally phased truth and query input VCFs, and requires local phasing to be correct in order to consider variants true positives.

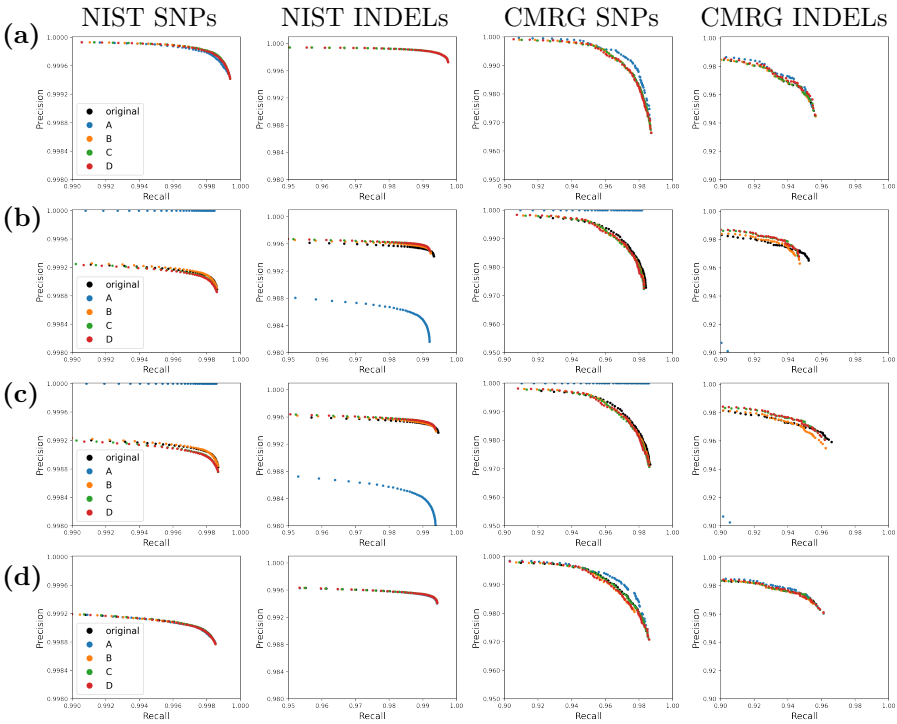


Fig. 4 **(a)** Standardized vcfeval and **(b,c,d)** vcfdist precision-recall plots for Truth Challenge V2 submission `K4GT3` on the NIST and CMRG datasets. For all plots, the original query VCF is evaluated before and after changing its variant representation to design points A, B, C, and D (see Figure 2) **(a)** This plot is identical to Figure 3c, but using axes consistent with the remainder of this figure. **(b)** Evaluation with vcfdist, turning the options for standardization and partial credit off; this can be directly compared to vcfeval's performance in Figure 3b. Note that the original representation no longer outperforms other representations, since local phasing is enforced. **(c)** Evaluation with vcfdist, allowing partial credit but not standardization, resulting in minor CMRG recall improvements. **(d)** Evaluation with vcfdist, allowing partial credit and standardizing variant representation. This results in the most consistent results, and is the recommended usage.

This results in a lower precision-recall curve in Figure 4d than Figure 4a, which we believe more accurately represents the true performance. It also makes evaluation of the original fragmented representation (black) much more consistent with the other representations (Figure 4b vs Figure 3a). Importantly, vcfdist does not require correct global phasing of variants, and allows for arbitrary switch errors to occur between clusters of dependent variants.

Clustering variants by genomic distance is insufficient. Several previous works such as vcfeval determine which variants are dependent on one another using a simple distance-based heuristic, assuming that any variants within 50 bases of one another are dependent representations of a single complex variant (9). Although this approach works fairly well in practice, it will result in numerous false positives that will unnecessarily increase cluster size and

slow down later computation. Since we do not allow phase switches within a cluster, eliminating false positive dependencies is also important to allow phase switches to occur between independent clusters. We developed a clustering algorithm (see *Online Methods*) to determine variant dependency that reduces false positive dependencies while eliminating all false negative dependencies. Figure 5 shows an overview of our tool, vcfdist.

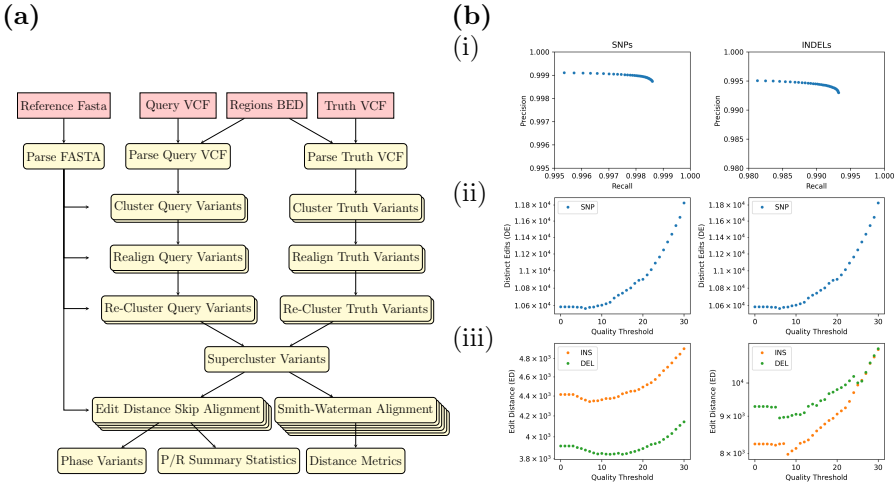


Fig. 5 **(a)** Overview of vcfdist; more details regarding each step can be found in the *Online Methods* section. **(b)** Visual summary of vcfdist results for submission `K4GT3`; a textual summary is given in Supplementary Figure S2. **(i)** Precision-Recall curve **(ii)** SNP and **(iii)** INDEL edit distance (ED) and distinct edit (DE) error curves, sweeping possible quality thresholds.

False negative variant dependencies commonly occur when dependent variants are separated by long yet repetitive regions. Supplementary Figure S3 shows three examples where missed dependencies affect not just query VCF evaluation, but the chosen representation of a complex variant within the NIST v4.2.1 ground truth VCF. For these particular cases, the truth VCF contains an unreasonably verbose variant representation that is not explainable by simple differences in affine gap alignment parameters.

On the NIST v4.2.1 benchmarking dataset of $5,393,709$ variants, the 50 base gap clustering heuristic results in $4,725,342$ clusters ($668,367$ dependencies), with an average of 1.14 variants and 4.33 bases per cluster. Our Smith-Waterman based clustering algorithm results in $5,304,220$ clusters ($89,489$ dependencies), with an average of 1.02 variants and 1.17 bases per cluster. This 86.6% reduction in variant dependencies shrinks the average cluster size and makes downstream $O(b_i^2)$ alignment algorithms more efficient, where b_i is the number of bases in cluster i. In fact, there was a 21.99× reduction in

worst-case alignment matrix cells computed ($\sum_{i=0}^{c} b_i^2$, where c is the number of clusters). Our method also discovered 743 missed dependencies between variants greater than 50 bases apart. Handling such cases correctly during evaluation is becoming increasingly important as fewer total errors are made by variant callers.

A standard variant representation stabilizes precision-recall curves. The precision-recall curve stabilization gained by using a standard variant representation (point C) is clearly demonstrated in Figures 3 and 4. In fact, standardizing variant representation is sufficient for prior work vcfeval to obtain consistent results across variant representations (Figure 3c). Researchers could use vcfdist as a pre-processing step for variant standardization, and then perform evaluations as usual through vcfeval. It is important to note, however, that consistency does not imply accuracy. The precision-recall curve for vcfdist in Figure 4d is not the same curve as for vcfeval in Figure 4a. The other improvements presented in this paper in regards to clustering, phasing, and partial credit have a large impact on the final results.

As can be seen in Figures 6a and 6b, the final F1 scores reported on both the NIST and CMRG datasets are much more consistent for vcfdist than vcfeval as variant representation changes. For each graph, the coefficient of determination (R^2) is greater and the AMRC is lower for vcfdist in comparison to vcfeval. We define AMRC ("Average Maximum Rank Change") in the *Online Methods* section. It is an average measure of how much a particular submission's rank (in performance relative to the other 63 submissions) would change when using the best versus the worst possible variant representation for that particular submission. Ideally, values of $R^2 = 1$ and AMRC $= 0$ would indicate that our evaluation is entirely independent of variant representation.

Partial credit more accurately reflects variant calling performance. As demonstrated in Figure 1, variant calls which are nearly but not exactly correct are penalized under current evaluation methods. Figure 4c shows a slight improvement in variant recall over Figure 4b for the CMRG dataset after vcfdist assigns partial credit to mostly-correct variants.

Distance-based performance metrics provide additional insight. As mentioned earlier, precision-recall curves are dependent upon query and truth VCF variant representations, as well as the reference FASTA. In contrast, distance-based performance metrics such as the edit distance, number of distinct edits, and alignment score are determined solely by the query and truth sequences and are unchanged by different variant representations or references. Figure 6c demonstrates this. The very minor remaining differences in distance-based metrics are caused by false-positive variant dependencies during clustering. Since slightly different clusters are created, depending on the original variant representations, enforcing consistent local phasing within a cluster may lead to different results.

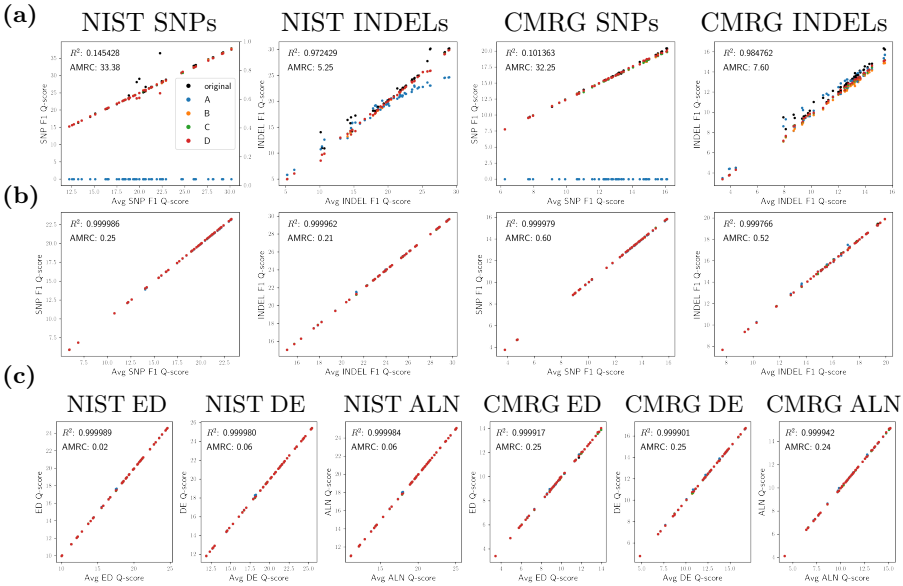


Fig. 6 **(a)** vcfeval and **(b)** vcfdist F1 Q-score plots for all Truth Challenge V2 submissions on both the NIST and CMRG datasets. **(c)** The same information, using new sequence distance-based summary metrics (see *Online Methods* for a full explanation of each). On each graph, average Q-score is plotted against the Q-score for the original representation and at points A, B, C, and D (see Figure 2). Clearly, vcfdist results in (b) and (c) are much more stable in regards to variant representation than vcfeval in (a).

Although a similar level of evaluation consistency can be gained by using a standard variant representation with vcfdist (Figure 6b), distance-based summary metrics are useful to gain a better understanding of overall performance. Supplementary Figure S5b shows that of the three multi-technology pFDA submissions which tied for the winning performance on the NIST dataset, Sentieon's solution performed best on SNPs, but Roche's solution performed best on INDELs. Using precision-recall curves for SNPs and INDELs separately, it is hard to get a clear picture of which submission has the best overall performance. This is because SNPs are much more common than INDELs, and the two variant types do not always have similar accuracy characteristics (performance on INDELs is usually worse). Supplementary Figure S5c clearly demonstrates that Roche's solution performs better overall on the NIST dataset, as it has lower edit distance, distinct edits, and alignment distance. Interestingly, Sentieon's solution achieves better performance on the CMRG dataset.

Runtime and efficiency. We evaluated the runtime of submission `K4GT3` on the CMRG and NIST datasets for each of the five representations in Figure 4 (original, A, B, C, D). On the relatively small CMRG dataset, the average vcfeval single-threaded runtime was 153.6s and memory usage was 5.26GB. In comparison, vcfdist's average single-threaded runtime was 441.2s and maximum memory usage was 4.99GB. For smaller datasets, loading the reference

FASTA accounts for the majority of memory usage (≈ 3GB). For larger datasets such as NIST, vcfdist's memory usage (5.86GB) scales much better than vcfeval's (24.91GB). Although vcfdist has a slower single-threaded runtime (2648.2s vs 1489.6s) on this dataset, more concurrent threads can run due to vcfdist's 4.25× lower memory usage.

Discussion

The first major step towards affordable whole genome sequencing was the development of second-generation sequencing methods, which enabled massively parallel sequencing of short reads up to 500 bases (34). With such short reads, resolving phasing and calling variants in repetitive genomic regions was incredibly difficult. It was in this context that vcfeval was developed. With the introduction of third-generation sequencing methods in recent years, however, the scope of genomic regions under active analysis has expanded. The 2019 NIST v4.2.1 benchmarking dataset used in this paper includes 17% more SNVs, 176% more INDELs and 12% larger benchmarking regions than the 2016 NIST v3.3.2 dataset (35). Additionally, now that datasets often have average read lengths of 10-100Kb, some recent genome assemblies contain phase block N50s of more than 20Mb (36).

As the state of the field and technology shifts, so too must the standards for evaluation. The accuracy of third-generation sequencing has improved tremendously in the past few years (5; 6), and most small variant calling errors are now limited to low-complexity regions and complex variants. This work aims to ensure that evaluation is consistent in low-complexity regions and for complex variants where many representations are possible. We do so by proposing a standard variant representation, enforcing consistent local phasing, improving variant clustering, attributing partial credit, and motivating the use of alignment distance metrics for evaluation.

As genome sequencing becomes cheaper and more accurate, and as reads lengthen, a gradual shift is beginning from reference-based variant calling to de-novo diploid assembly (21; 36). Even if the most common format for storing a human genome shifts from a haploid reference FASTA and an accompanying VCF to a diploid FASTA, there will always be the need to compare human genomes to one another. Whether this comparison is stored using a VCF or a graph format, the same challenge of multiple possible alignments remains. As a result, we expect this work to remain relevant for the foreseeable future.

Online Methods

Datasets. All datasets used in this manuscript have been made publicly available in prior work, and were for human genome HG002 using reference GRCh38. This includes precisionFDA Truth Challenge V2 variant callsets (13), in addition to the NIST v4.2.1 (16) and CMRG v1.00 (31) ground truth callsets from the Genome in a Bottle Consortium

(GIAB). For the NIST v4.2.1 dataset, we used the phased truth VCF `HG002_GRCh38_1_22_v4.2.1_benchmark_hifiasm_v11_phasetransfer.vcf.gz` instead of the unphased `HG002_GRCh38_1_22_v4.2.1_benchmark.vcf.gz` VCF. For more information, please refer to the *Data Availability* section at the end of this manuscript.

Preprocessing. Running our pipeline required two modifications to the phased NIST ground truth VCF; the CMRG ground truth VCF was used unchanged. Firstly, there were erroneously two additional blank FORMAT field values in the Major Histocompatibility Complex (MHC) region on chr6 that were removed. Secondly, WhatsHap only accepts integer phase set values, whereas the NIST truth VCF contained phase sets named with strings. The phase set (`PS`) tags in the NIST ground truth VCF were removed using `bcftools annotate`. This does not affect our downstream evaluation because vcfdist only assumes variants are phased locally, and not globally.

Several of the pFDA Truth Challenge V2 query VCFs required modification as well. Submissions `OGOOR` and `BARQS` reported some reference or ungenotyped variants as haploid, which were filtered. Chromosomes were renamed (i.e. `1` to `chr1`) for the same two query VCFs. Submissions `OGOOR`, `BARQS`, and `HB8P3` reported variants on decoy contigs, which were filtered. Additionally, the VCF header for `HB8P3` did not store the lengths of each decoy contig. Submission `Y8QR8` did not report any contig lengths, which were added manually. Submissions `WGQ43` and `YGOTK` did not contain variant quality scores. Submissions `WX8VK` and `CZA1Y` contained an erroneous variant where `ALT` was the same as `REF` at position `chr6:28,719,765`. This was removed.

Phasing input variant callsets. vcfdist requires input query and truth VCFs to be locally phased. However, most VCFs submitted to precisionFDA's Truth Challenge V2 were unphased. In order to phase these VCFs prior to evaluation, we downloaded all the original HG002 sequencing read FASTAs provided to participants of the competition. This dataset contained 35× genome coverage of Illumina short paired-end reads, 35× coverage of PacBio HiFi reads, and 50× coverage of Oxford Nanopore reads, sequenced on a PromethION and basecalled with Guppy v3.6.0. All reads were aligned to GRCh38 with minimap2 using the default parameters corresponding to each sequencing technology (`sr`, `map-pb`, and `map-ont`). Supplementary and secondary alignments were filtered. The aligned reads were then phased using `whatshap haplotag`. Lastly, all variants in the 64 Truth Challenge V2 submission callsets were phased using `whatshap phase` and the phased reads from all three technologies.

Clustering dependent variants. When multiple variants occur at the same locus, there are often multiple ways that differences between the draft reference FASTA and the resulting sequence could be represented in a VCF. This is still true after performing standard variant normalization techniques. For example, see Figure 2b. In order to fairly evaluate the differences between two VCFs, we must evaluate an entire set of nearby variants (or "cluster") in such regions

simultaneously. vcfeval uses the heuristic that any variants located more than 50 bases away from one another on the reference are likely independent (9). While this is usually true, it is not always the case. Furthermore, considering any two variants less than 50 bases apart to be dependent creates many false positive dependencies. VarMatch proposes a clustering algorithm that can be used to identify "separator regions" that are long and non-repetitive sections of the reference genome that contain no variants and can be used to divide variants into independent clusters (14).

Using a similar approach, we note that two clusters of variants can be considered independent of one another for a given Smith-Waterman configuration if all optimal alignment paths return to the main diagonal between the two clusters. This "main diagonal" is the reference region between the two clusters which contains no variants. In order to determine whether two adjacent clusters are dependent, we begin extending from the start of the first cluster and note the rightmost reference position to which we can extend while avoiding the main diagonal and having an alignment penalty of lesser or equal value to that of the original representation of variants within the first cluster.

We mirror this process for the second cluster, extending from its end to the leftmost reference position possible while avoiding the main diagonal and limiting the alignment penalty. If the latter position is less than or equal to the first, then these two paths are able to meet off the main diagonal, and these two clusters are dependent. We use an iterative algorithm, initializing each cluster to a single variant. If two adjacent clusters are dependent, we merge them into a a single larger cluster. We repeat this process until all clusters have not grown from the previous iteration or the maximum iteration count is reached. An overview of this clustering algorithm can be seen in Supplementary Figure S7.

A standard variant representation. As shown in Figure 2a, point C was selected to be the Smith-Waterman design point for a standard variant representation, since it is the approximate centroid of widely-accepted parameters used to align reads of three different popular sequencing technologies (Illumina short reads, PacBio HiFi reads, and Oxford Nanopore long reads). The integer affine-gap parameters corresponding to point C are $(m, x, o, e) = (0, 5, 6, 2)$. Each haplotype was standardized independently, for both the truth and query VCFs. After clustering variants using the previously-described algorithm, within each cluster all variants are applied to the draft reference FASTA and the resulting sequence is realigned using the standard affine-gap parameters for point C. A new set of variant calls is then defined from the resulting alignment string, now in the standard representation.

Superclustering variants. After clustering and realigning the truth and query variants separately, we need some means of associating query variants with their corresponding truth variants and vice-versa. In order to do so, we group together clusters on both the truth and query haplotypes that reach within 50 base pairs of one another. An example is shown in Supplementary

Figure S8. This resulting "supercluster" may be composed of several variant clusters on any of the four haplotypes (two each for the variant and query). All subsequent analyses take place within a supercluster, which are considered fully independent of one another.

Precision and recall calculations. Since vcfdist assumes variants have been locally phased, we do not consider the possibility of genotype or switch errors within a supercluster. However, because vcfdist makes no assumptions regarding global phasing of the truth and query VCFs, we consider the relative phasing of each supercluster to be unknown and allow phase switches in between superclusters. In order to phase superclusters, we first perform each of the four possible alignments of truth and query haplotypes: Truth1 to Query1, Truth1 to Query2, Truth2 to Query1, and Truth2 to Query2.

During this alignment, we represent each query haplotype as a graph (merged with the reference sequence) to allow skipping query variants without penalty, since there may be false positive variant calls. Note that although this algorithm is similar to ordinary sequence-to-graph alignment, it is not equivalent since reference-to-query transitions (and vice versa) are not allowed within truth variants. An example of this alignment is shown in Supplementary Figure S10. The minimum edit distance is calculated for each of the four alignments, and the phasing which minimizes total edit distance is selected: (X) Truth1 to Query1 and Truth2 to Query2, or (Y) Truth1 to Query2 and Truth2 to Query1. This represents the best possible phasing for the supercluster.

Once a phasing has been selected, summary statistics (e.g. counting true positive (TP) and false positive (FP) variants) for each supercluster can be calculated using the resulting alignments. Unlike prior work, vcfdist attributes partial credit to variant calls which are not entirely correct yet still reduce the overall edit distance. We call these "partial positives" (PP). These partial positives can be used to more accurately represent the true performance of a variant caller, since it is common to report the nearly-correct length of INDEL variants (such as in Figure 1). The first step is identifying "sync points", where the optimal alignment path passes through a reference base on the main diagonal (i.e. a base that is not within either a truth or query variant). Such sync points are marked in red in Supplementary Figure S10d.

Between each sync point, query and truth variants are considered to be true positives if the new alignment's edit distance is zero, false positives (or false negatives for the truth VCF) where the edit distance is unchanged, and partial positives where the edit distance is reduced. For the purposes of precision and recall calculation, partial positives are converted to a fractional positive plus a fractional negative using the old and new edit distances. For example, the partial positive query variant call in Figure 1 reduces the edit distance from 3 to 1 and is counted as $\frac{2}{3}\text{TP} + \frac{1}{3}\text{FP}$. These statistics are calculated separately for each haplotype, and then summed. As a result, we count a false positive homozygous variant (1|1) as two false positives.

Distance calculations. When measuring the distance between two sequences, we find it useful to define two metrics: "edit distance" (ED) and "distinct edits" (DE). For example, an insertion of length 5 has an edit distance of 5, but is 1 distinct edit. We would like to minimize both DE and ED simultaneously, placing greater importance on distinct edits than edit distance, since erroneously lengthening a gap is less detrimental than introducing new edits. If we aim to globally minimize 2DE+ED, this minimization problem is equivalent to Smith-Waterman global alignment with the affine-gap parameters $(m, x, o, e) = (0, 3, 2, 1)$. This is because opening a gap (o) increases DE, extending a gap (e) increases ED, and a substitution (x) increases both ED and DE. This Smith-Waterman design point is B in Figure 2a.

New summary metrics.
For each complex variant or INDEL, we allow partial positive variants to receive partial credit by comparing the reference sequence edit distance ($\mathrm{ED_{ref}}$) and the query sequence edit distance ($\mathrm{ED_{query}}$). This query sequence is derived by applying query VCF variants to the reference sequence.
Partial positives:

$$\mathrm{PP_{query}} = \left(1 - \frac{\mathrm{ED_{query}}}{\mathrm{ED_{ref}}}\right)\mathrm{TP_{query}} + \frac{\mathrm{ED_{query}}}{\mathrm{ED_{ref}}}\mathrm{FP_{query}}$$

$$\mathrm{PP_{truth}} = \left(1 - \frac{\mathrm{ED_{query}}}{\mathrm{ED_{ref}}}\right)\mathrm{TP_{truth}} + \frac{\mathrm{ED_{query}}}{\mathrm{ED_{ref}}}\mathrm{FN_{truth}}$$

In order to increase stability of precision calculations, vcfdist uses the number of true positives from the truth VCF rather than the query VCF.
GA4GH precision definition: *vcfdist precision definition:*

$$\mathrm{Precision} = \frac{\mathrm{TP_{query}}}{\mathrm{TP_{query}} + \mathrm{FP_{query}}} \qquad \mathrm{Precision} = \frac{\mathrm{TP_{truth}}}{\mathrm{TP_{truth}} + \mathrm{FP_{query}}}$$

In order to evaluate approximate genome variant calling quality, we define the F1 Q-score metric as follows:

$$\mathrm{F1_{Qscore}} = -10\log_{10}(1 - \mathrm{F1_{score}})$$

We make similar definitions for ED Q-score, DE Q-score, and ALN Q-score. These metrics are Q-score estimates of overall variant calling quality, based on the remaining edit distance, distinct edits, and alignment score of the query to truth sequences, relative to the difference between the reference and truth sequences:

$$\mathrm{ED}_{\mathrm{Qscore}} = -10\log_{10}\left(\frac{\mathrm{ED}_{\mathrm{query}}}{\mathrm{ED}_{\mathrm{ref}}}\right) \qquad \mathrm{DE}_{\mathrm{Qscore}} = -10\log_{10}\left(\frac{\mathrm{DE}_{\mathrm{query}}}{\mathrm{DE}_{\mathrm{ref}}}\right)$$

$$\mathrm{ALN}_{\mathrm{Qscore}} = -10\log_{10}\left(\frac{\text{Alignment Score}_{\mathrm{query}}}{\text{Alignment Score}_{\mathrm{ref}}}\right)$$

Lastly, we define the "Average Maximum Rank Change" (AMRC) to evaluate the stability of a performance evaluation on a dataset in regards to variant representation. In this work, we evaluate the Q-score performance of pFDA Truth Challenge V2 submissions $S = \{S^{(1)}, ..., S^{(i)}, ..., S^{(n)}\}$. For each submission $S^{(i)}$, we use both the original (O) variant representations and when normalized to points A, B, C, and D.

$$S^{(i)} = \{S_O^{(i)}, S_A^{(i)}, S_B^{(i)}, S_C^{(i)}, S_D^{(i)}\}$$

First, we note the Q-score performance metrics for each submission $S^{(i)}$:

$$Q^{(i)} = \mathrm{Q}_{\mathrm{score}}(S^{(i)}) =$$
$$\{\mathrm{Q}_{\mathrm{score}}(S_O^{(i)}), \mathrm{Q}_{\mathrm{score}}(S_A^{(i)}), \mathrm{Q}_{\mathrm{score}}(S_B^{(i)}), \mathrm{Q}_{\mathrm{score}}(S_C^{(i)}), \mathrm{Q}_{\mathrm{score}}(S_D^{(i)})\}$$

Next, we create a sorted sequence (in increasing order) of the median Q-score for each submission, our best guess at each submission's "true" performance.

$$M = \mathrm{sort}(\{m^{(i)} | m^{(i)} = \mathrm{med}(Q^{(i)})\})$$

Then we find the Q-scores for the worst- and best-performing representations of each submission.

$$W = \{w^{(i)} | w^{(i)} = \min(Q^{(i)})\} \qquad B = \{b^{(i)} | b^{(i)} = \max(Q^{(i)})\}$$

Lastly, we find the average change in rank for submission $S^{(i)}$ when ranking submissions in order of Q-score if we use the best versus the worst possible representation. Note that we must omit the median of the current submission, $\mathrm{med}(Q^{(i)})$, when calculating the rank. We define $\mathrm{pos}(x, Y)$ as the first position that value x could be inserted into sorted sequence Y such that it remains sorted.

$$\mathrm{AMRC} = \sum_{i=1}^{n} \left(\mathrm{pos}(B^{(i)}, M - \mathrm{med}(Q^{(i)})) - \mathrm{pos}(W^{(i)}, M - \mathrm{med}(Q^{(i)}))\right) / n$$

Global phasing calculation. To phase superclusters, we use a simple dynamic programming algorithm which minimizes the total number of switch

errors and supercluster phasing errors. A detailed example is shown in Supplementary Figure S9. Firstly, superclusters are first categorized as type X or Y depending on the best mapping of query to truth haplotypes (X: Truth1 to Query1 and Truth2 to Query2, or Y: Truth1 to Query2 and Truth2 to Query1). It is also possible for both mappings result in the same minimum edit distance, in which case the supercluster is not categorized. During the forwards pass, switch errors and supercluster phasing errors are minimized, as the global phase state switches back and forth between X and Y. During the backwards pass, phase blocks are recovered and the total number of switch errors and supercluster phasing errors is reported.

Output. In addition to a concise summary output (shown in Supplementary Figure S2), vcfdist outputs multiple verbose TSV files to aid in further analysis of variant calling performance. vcfdist also outputs both the original and new standard representations of the query and truth VCFs. Lastly, several analysis scripts are provided which can be used to conveniently display the output of vcfdist.

Code Availability

All code for vcfdist and the benchmarking pipelines developed for this manuscript are available in a public Github repository: https://github.com/TimD1/vcfdist. Publicly available software used to phase the input callsets includes minimap2 v2.24-r1122, samtools v1.16.1, and WhatsHap v1.7. To evaluate the phased callsets, vcfdist v1.0.0, hap.py v0.3.15, and rtg-tools vcfeval v3.12.1 were used.

Data Availability

All datasets used in this manuscript were previously released by other researchers and are publicly available online. The unphased precisionFDA Truth Challenge V2 VCF submissions and corresponding read FASTAs used to phase the VCFs are hosted on the NIST public data repository at https://data.nist.gov/od/id/mds2-2336 (13). The phased ground truth VCF and BED files for the HG002 whole-genome and challenging medically-relevant genes datasets are available in the `NISTv4.2.1` and `CMRG_v1.00` directories of the Genome In A Bottle Consortium's FTP release folder, respectively: https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG002_NA24385_son (16; 31). The GRCh38 reference FASTA is likewise available at https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/references/. More details on the exact files and versions used in this manuscript are available online at https://github.com/TimD1/vcfdist/tree/master/data.

References

[1] Kris Wetterstrand. The cost of sequencing a human genome., 2021. https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost.

[2] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

[3] NHGRI. Genetics vs. genomics fact sheet., 2018. https://www.genome.gov/about-genomics/fact-sheets/Genetics-vs-Genomics.

[4] Stephen T Sherry, M-H Ward, M Kholodov, J Baker, Lon Phan, Elizabeth M Smigielski, and Karl Sirotkin. dbsnp: the ncbi database of genetic variation. *Nucleic acids research*, 29(1):308–311, 2001.

[5] Ryan R Wick, Louise M Judd, and Kathryn E Holt. Performance of neural network basecalling tools for oxford nanopore sequencing. *Genome biology*, 20(1):129, 2019.

[6] Medhat Mahmoud, Yongqing Huang, Kiran Garimella, Peter A Audano, Wan Wan, Nripesh Prasad, Robert E Handsaker, Salina Hall, Arianna Pionzio, Michael C Schatz, et al. Utility of long-read sequencing for all of us. *bioRxiv*, pages 2023–01, 2023.

[7] Daniel C Koboldt. Best practices for variant calling in clinical sequencing. *Genome Medicine*, 12(1):1–13, 2020.

[8] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. Review the cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary Oncology/Współczesna Onkologia*, 2015(1):68–77, 2015.

[9] John G Cleary, Ross Braithwaite, Kurt Gaastra, Brian S Hilbush, Stuart Inglis, Sean A Irvine, Alan Jackson, Richard Littin, Mehul Rathod, David Ware, et al. Comparing variant call files for performance benchmarking of next-generation sequencing variant calling pipelines. *BioRxiv*, page 023754, 2015.

[10] Peter Krusche, Len Trigg, Paul C Boutros, Christopher E Mason, M Francisco, Benjamin L Moore, Mar Gonzalez-Porta, Michael A Eberle, Zivana Tezak, Samir Lababidi, et al. Best practices for benchmarking germline small-variant calls in human genomes. *Nature biotechnology*, 37(5): 555–560, 2019.

[11] Junhao Su, Zhenxian Zheng, Syed Shakeel Ahmed, Tak-Wah Lam, and Ruibang Luo. Clair3-trio: high-performance nanopore long-read variant calling in family trios with trio-to-trio deep neural networks. *Briefings in Bioinformatics*, 23(5):bbac301, 2022.

[12] Ryan Poplin, Pi-Chuan Chang, David Alexander, Scott Schwartz, Thomas Colthurst, Alexander Ku, Dan Newburger, Jojo Dijamco, Nam Nguyen, Pegah T Afshar, et al. A universal snp and small-indel variant caller using deep neural networks. *Nature biotechnology*, 36(10):983–987, 2018.

[13] Nathan D Olson, Justin Wagner, Jennifer McDaniel, Sarah H Stephens, Samuel T Westreich, Anish G Prasanna, Elaine Johanson, Emily Boja, Ezekiel J Maier, Omar Serang, et al. Precisionfda truth challenge v2: Calling variants from short and long reads in difficult-to-map regions. *Cell Genomics*, 2(5):100129, 2022.

[14] Chen Sun and Paul Medvedev. Varmatch: robust matching of small variant datasets using flexible scoring schemes. *Bioinformatics*, 33(9): 1301–1308, 2017.

[15] Adrian Tan, Gonçalo R Abecasis, and Hyun Min Kang. Unified representation of genetic variants. *Bioinformatics*, 31(13):2202–2204, 2015.

[16] Justin Wagner, Nathan D Olson, Lindsay Harris, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Byunggil Yoo, Neil Miller, et al. Benchmarking challenging small variants with linked and long reads. *Cell Genomics*, 2(5):100128, 2022.

[17] Tim Dunn, David Blaauw, Reetuparna Das, and Satish Narayanasamy. npore: n-polymer realigner for improved pileup variant calling. *bioRxiv*, 2022.

[18] Osamu Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162(3):705–708, 1982. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(82)90398-9. URL https://www.sciencedirect.com/science/article/pii/0022283682903989.

[19] Jordan M Eizenga and Benedict Paten. Improving the time and space complexity of the wfa algorithm and generalizing its scoring. *bioRxiv*, 2022.

[20] Mykola Akulov and Ragnar Groot Koerkamp. Transforming match bonus into cost, 2022. URL https://curiouscoding.nl/posts/alignment-scores-transform/.

[21] Mikko Rautiainen, Sergey Nurk, Brian P Walenz, Glennis A Logsdon, David Porubsky, Arang Rhie, Evan E Eichler, Adam M Phillippy,

and Sergey Koren. Verkko: telomere-to-telomere assembly of diploid chromosomes. *bioRxiv*, 2022.

[22] Fritz J Sedlazeck, Philipp Rescheneder, Moritz Smolka, Han Fang, Maria Nattestad, Arndt Von Haeseler, and Michael C Schatz. Accurate detection of complex structural variations using single-molecule sequencing. *Nature methods*, 15(6):461–468, 2018.

[23] Martin Šošić and Mile Šikić. Edlib: a c/c++ library for fast, exact sequence alignment using edit distance. *Bioinformatics*, 33(9):1394–1395, 2017.

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

[25] Kishwar Shafin, Trevor Pesout, Ryan Lorig-Roach, Marina Haukness, Hugh E Olsen, Colleen Bosworth, Joel Armstrong, Kristof Tigyi, Nicholas Maurer, Sergey Koren, et al. Nanopore sequencing and the shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nature biotechnology*, 38(9):1044–1053, 2020.

[26] Sergey Nurk, Brian P Walenz, Arang Rhie, Mitchell R Vollger, Glennis A Logsdon, Robert Grothe, Karen H Miga, Evan E Eichler, Adam M Phillippy, and Sergey Koren. Hicanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome research*, 30(9):1291–1305, 2020.

[27] Miten Jain, Ian T Fiddes, Karen H Miga, Hugh E Olsen, Benedict Paten, and Mark Akeson. Improved data analysis for the minion nanopore sequencer. *Nature methods*, 12(4):351–356, 2015.

[28] Novocraft Technologies. Novoalign, 2022. URL http://www.novocraft.com/products/novoalign/. Powerful tool designed for mapping of short reads onto a reference genome from Illumina, Ion Torrent, and 454 NGS platforms.

[29] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[30] Armin Toepfer. pbmm2, 2022. URL https://github.com/PacificBiosciences/pbmm2. A minimap2 SMRT wrapper for PacBio data.

[31] Justin Wagner, Nathan D Olson, Lindsay Harris, Jennifer McDaniel, Haoyu Cheng, Arkarachai Fungtammasan, Yih-Chii Hwang, Richa Gupta, Aaron M Wenger, William J Rowell, et al. Curated variation benchmarks for challenging medically relevant autosomal genes. *Nature biotechnology*, 40(5):672–680, 2022.

[32] Justin Wagner, Nathan D Olson, Lindsay Harris, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Aaron M Wenger, William J Rowell, et al. Benchmarking challenging small variants with linked and long reads. *BioRxiv*, 2020.

[33] Simone Maestri, Maria Giovanna Maturo, Emanuela Cosentino, Luca Marcolungo, Barbara Iadarola, Elisabetta Fortunati, Marzia Rossato, and Massimo Delledonne. A long-read sequencing approach for direct haplotype phasing in clinical settings. *International Journal of Molecular Sciences*, 21(23):9177, 2020.

[34] Barton E Slatko, Andrew F Gardner, and Frederick M Ausubel. Overview of next-generation sequencing technologies. *Current protocols in molecular biology*, 122(1):e59, 2018.

[35] Justin M Zook, Jennifer McDaniel, Nathan D Olson, Justin Wagner, Hemang Parikh, Haynes Heaton, Sean A Irvine, Len Trigg, Rebecca Truty, Cory Y McLean, et al. An open resource for accurately benchmarking small variant and reference calls. *Nature biotechnology*, 37(5):561–566, 2019.

[36] Erich D Jarvis, Giulio Formenti, Arang Rhie, Andrea Guarracino, Chentao Yang, Jonathan Wood, Alan Tracey, Francoise Thibaud-Nissen, Mitchell R Vollger, David Porubsky, et al. Semi-automated assembly of high-quality diploid human reference genomes. *Nature*, pages 1–13, 2022.

Supplementary Information

(a)

Reference `TTCCTTTCTTTCTTCCTTTCTTTCTTTCTT`

Query `TTCCTTTCTTCTTTCTT` **Truth** `TTCCTTTCCTTCTT`

Query VCF

POS	REF	ALT
8	CTTTCTTCCTTTCT	C

Alt. Query VCF

POS	REF	ALT
3	CCTTTCTTTCTTCCTTT	C
24	C	CTTC

Truth VCF

POS	REF	ALT
3	CCTTTCTTTCTTCCTTT	C
25	T	C

(b)

vcfeval INDEL Summary Statistics

	TP	FP	FN	PP	Prec.	Recall	F1	F1 Q-score
Query	0	1	1	0	0.00	0.00	0.00	0.00
Alt. Query	1	1	0	0	0.50	1.00	0.67	4.77

(c)

vcfdist INDEL Summary Statistics

	TP	FP	FN	PP	Prec.	Recall	F1	F1 Q-score
Query	0	0	0	1	0.82	0.82	0.82	7.53
Alt. Query	1	1	0	0	0.50	1.00	0.67	4.77

(d)

vcfdist Distance Summary

	ED	DE	DE Q-score	ED Q-score	ALN Q-Score
Reference	17	2			
Query	3	2	7.27	0.00	4.56
Alt. Query	3	2	7.27	0.00	4.56

Fig. S1 Example vcfeval and vcfdist evaluation of Truth Challenge V2 winning submission `K4GT3` on GRCh38 region `chr5:140,941,200-140,941,230`. This region is within the protocadherin alpha gene cluster, genes PCDHA1-PCDHA13 in the CMRG dataset. **(a)** Reference, query, and truth sequences, as well as the query and truth VCFs. An alternate query VCF representation (resulting in the same query sequence) is shown as well. **(b)** vcfeval and **(c)** vcfdist count of true positive, false positive, false negative, and partial positive INDEL variants, as well as the calculated precision, recall, and F1 scores. Note that although both the original and alternate query VCF variant calls result in the exact same query sequence, the summary statistics differ. **(d)** Distance-based summary statistics reported by vcfdist (edit distance and distinct edits), which are independent of variant representation. An explanation of these summary statistics can be found in the *Online Methods* section.

(a)

TYPE	THRESH	TRUTH TP	QUERY TP	TRUTH FN	QUERY FP	PREC	RECALL	F1 SCORE	F1 QSCORE
SNP	Q >= 0	4636790	4640475	5502	32780	0.993	0.999	0.996	23.78
SNP	Q >= 0	4636790	4640475	5502	32780	0.993	0.999	0.996	23.78
INDEL	Q >= 0	744931	742121	3600	5990	0.991	0.994	0.993	21.27
INDEL	Q >= 7	744844	742055	3687	5776	0.992	0.994	0.993	21.31

(b)

TYPE	THRESH	EDIT DIST	DISTINCT EDITS	ED QSCORE	DE QSCORE	ALN QSCORE
ALL	Q >= 0	62704	48117	20.36	20.50	20.44
ALL	Q >= 6	62612	48126	20.37	20.50	20.45
ALL	Q >= 61	6819150	5394057	0.00	0.00	0.00
SNP	Q >= 0	37545	37545	20.92	20.92	
SNP	Q >= 0	37545	37545	20.92	20.92	
SNP	Q >= 61	4643522	4643522	0.00	0.00	
INDEL	Q >= 0	25159	10572	19.37	18.51	
INDEL	Q >= 8	24857	10488	19.42	18.54	
INDEL	Q >= 61	2175628	750535	0.00	0.00	

Fig. S2 Example vcfdist output, evaluating pFDA Truth Challenge V2 winner `K4GT3`. **(a)** Precision, recall, and F-score summary statistics **(b)** Distinct edits and edit distance summary statistics. For each type, results are shown for when no variants are filtered (`Q >= 0`), at the optimal Q-score cutoff, and in (b) when all variants are filtered (`Q >= 61`).

(a)

chr1:102,750,470-102,750,531

Reference	TCTTTATATATATATATATATATATATATATATATATATATATATATATATATATATATACACAG
Truth VCF	TCTTTATATATATATATATATATATATATATATATATATATATATATATATATATATATATACACAG
vcfdist	TCTTTATATATATATATATATATATATATATATATATATATATATATATATATATATATATACAG
Manual	TCTTTATATATATATATATATATATATATATATATATATATATATATATATATATATATACACAG

(b)

chr1:103,816,966-103,816,986

Reference	ATATATGTGTGTGTATATAT
Truth VCF	ATATATGTGTGTGTGTATAT
vcfdist	ATATATGTGTGTGTATATAT

(c)

chr1:94,689,680-94,689,765

Reference	GTCTAAAAAAAAAAAAAAAGAAAGAAAGAAAGAGAAAGAAAGAAAGAAAGAAAG AAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAAATGG
Truth VCF	GTCTAAAAAAAAAAAAAAAGAAAGAAAGAAAGAGAAAGAAAGAAAGAAAGAAA GAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAATGG
vcfdist	GTCTAAAAAAAAAAAAAAAGAAAGAAAGAAAGAGAAAGAAAGAAAGAAAGAAAG AAAGAAAGAAAGAAAGAAAGAAAGAAAGAAAAATGG

Fig. S3 Examples of variant clusters where the truth VCF does not contain the most succinct (or most likely) set of variants. Each example includes the reference sequence, and then shows various possible representations of same set of variants. Deletions are shown in red, insertions in green, and substitutions in yellow. **(a)** This variation can be represented by a single `C` to `T` SNP, as vcfdist finds. However, it is also possible to represent this as a complex copy number variation: an `AT` insertion and `AC` deletion. **(b)** The truth VCF representation contains an unnecessary substitution that can be avoided by reporting a different deletion of the same length. **(c)** The truth VCF reports two insertions and two deletions where two shorter deletions are sufficient to represent the new sequence.

Reference `AAGGAAATC` **Query** `ATCGAAAATC`

Alignment

(a)	**(b)**	**(c)**	**(d)**
`AAGGAAA-TC`	`AAGGAAA-TC`	`AAGGAAA-TC`	`AAGG-AAATC`
`·  ····  ··`	`·  ····  ··`	`·  ····  ··`	`·  ·  ·····`
`ATCGAAAATC`	`ATCGAAAATC`	`ATCGAAAATC`	`ATCGAAAATC`

VCF

POS	REF	ALT	POS	REF	ALT	POS	REF	ALT	POS	REF	ALT
2	AG	TC	2	A	T	2	A	T	2	A	T
6	AATC	AAATC	3	G	C	3	G	C	3	G	C
			6	AATC	AAATC	7	A	AA	4	G	GA
Original			Decomposed			Trimmed			Left shifted		

Fig. S4 Variant normalization of a multi-nucleotide polymorphism (MNP) and single base insertion. In **(a)**, the original MNP and insertion is reported. In **(b)**, the MNP is decomposed into two single nucleotide polymorphisms (SNPs). In **(c)**, the unchanged reference bases reported in the VCF on either side of the insertion are trimmed. In **(d)**, the insertion is left-shifted as far as possible.

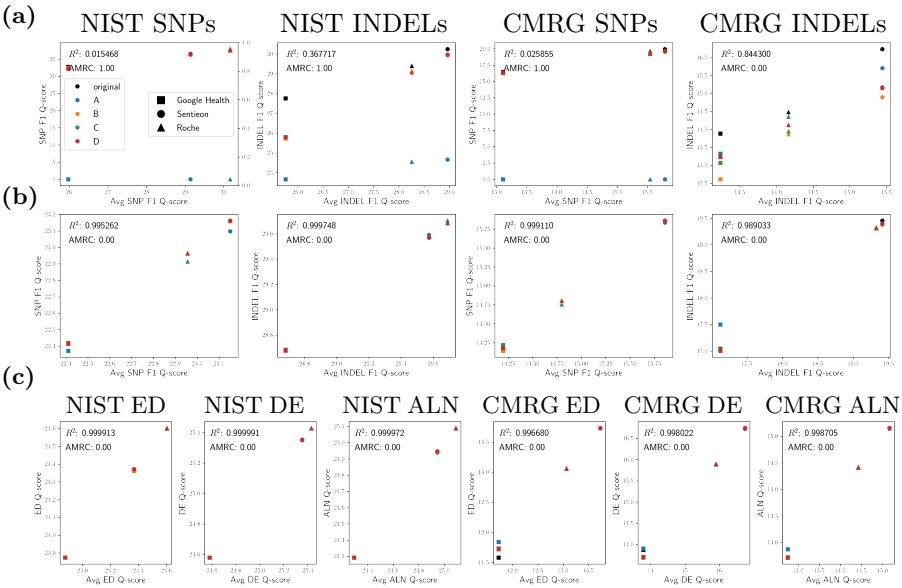


Fig. S5 **(a)** vcfeval and **(b)** vcfdist F1 Q-score plots for the three overall winning Truth Challenge V2 submissions using multiple sequencing technologies on both the NIST and CMRG datasets. **(c)** The same information, using new sequence distance-based summary metrics (see *Online Methods* for a full explanation of each). On each graph, average Q-score is plotted against the Q-score for the original representation and at points A, B, C, and D (see Figure 2).

(a)

Tools	Configs	m	x	o_1	e_1	o_2	e_2
BWA, GRAF, DRAGEN	default	-1	4	6	1		
minimap2	`sr`	-2	8	12	2	32	1
winnowmap, minimap2	default, `map-ont`	-2	4	4	2	24	1
minimap2	`map-pb`	-1	4	6	2	26	1
pbmm2	default, `CCS`	-2	5	5	4	56	2
minimap2	`asm5`	-1	19	39	3	81	1
minimap2	`asm10`	-1	9	16	2	41	1
ngmlr	`pacbio`	2	-5	-5	-5		-1
ngmlr	`ont`	3	-3	-1	-1		-0.5

(b)

Tool	Config	m	x	o_1	e_1	o_2	e_2
BWA, GRAF, DRAGEN	default	0	1	1.3	0.3		
minimap2	`sr`	0	1	1.3	0.3	2.5	0.2
winnowmap, minimap2	default, `map-ont`	0	1	0.833	0.5	4.167	0.333
minimap2	`map-pb`	0	1	1.3	0.5	5.3	0.3
pbmm2	default, `CCS`	0	1	0.857	0.714	8.14	0.429
minimap2	`asm5`	0	1	1.975	0.175	4.075	0.075
minimap2	`asm10`	0	1	1.65	0.25	4.15	0.15
ngmlr	`pacbio`	0	1	0	0.917		0.25
ngmlr	`ont`	0	1	0	0.417		0.333

Fig. S6 **(a)** Original and **(b)** normalized affine-gap parameters for tools used in the pFDA Truth Challenge V2. Normalized parameters are plotted in Figure 2.

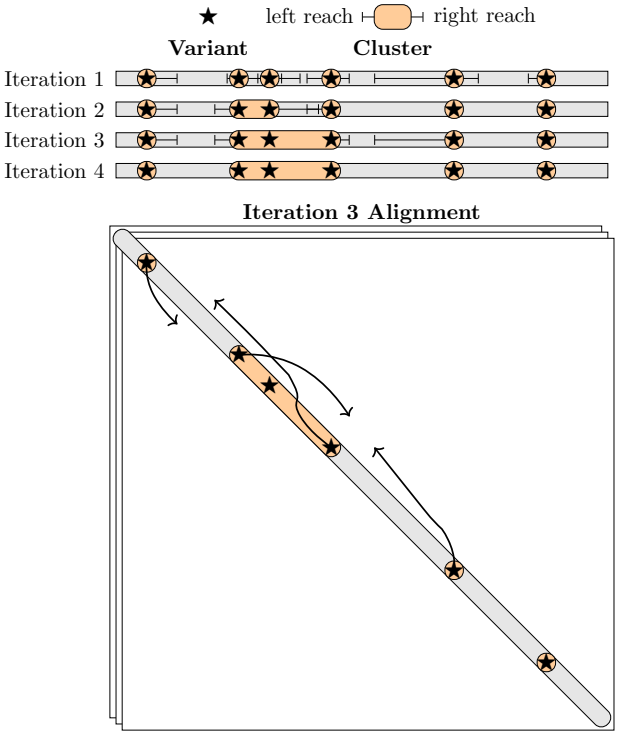


Fig. S7 An overview of the clustering algorithm. Clusters are initialized to each variant and then iteratively grown by extending leftwards (from each active cluster's end) and rightwards (from each active cluster's start) under Smith-Waterman alignment (disallowing reference diagonal matches, the gray shaded region) until a score limit is reached. Clusters are considered "active" if adjacent to a cluster that was merged in a previous iteration. Two clusters are considered dependent if the leftmost extension of the right cluster overlaps with the rightmost extension of the left cluster. Dependent clusters are merged after each iteration until no more clusters are active.

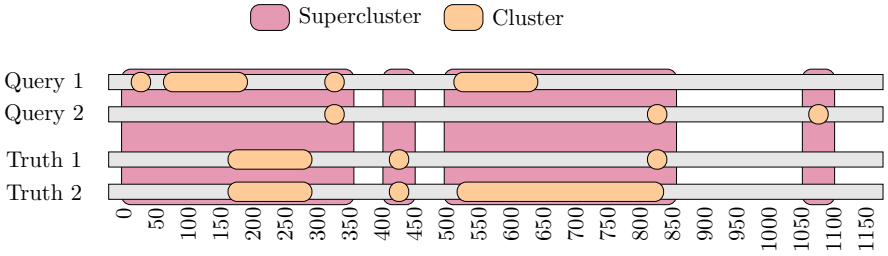


Fig. S8 An example demonstrating how variant clusters are grouped into superclusters, which span all haplotypes. Clusters within 50 bases of another cluster (on any haplotype) are merged into a single supercluster.

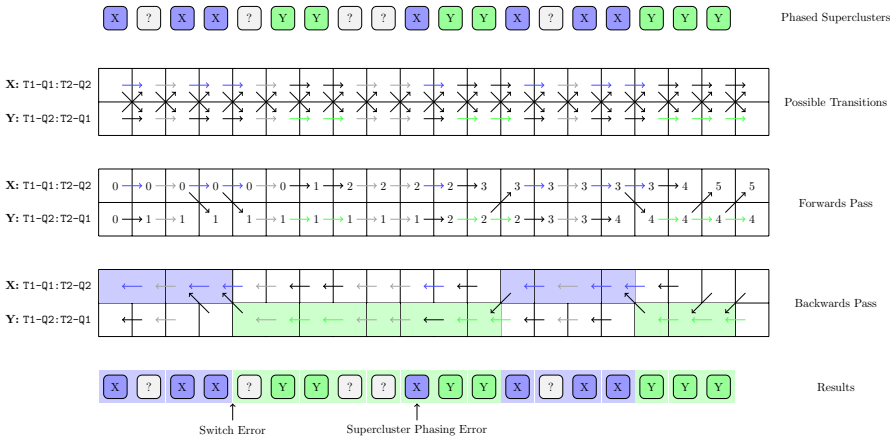


Fig. S9 An overview of the phasing algorithm, which steps through the two possible relative phasings of truth and query haplotypes (X: Truth1 to Query1 and Truth2 to Query2, or Y: Truth1 to Query2 and Truth2 to Query1) for each supercluster. It minimizes the total number of switch errors and supercluster phasing errors using a dynamic programming algorithm. Black arrows show penalized transitions. Note that either phase state X or Y is allowed in an uncategorized supercluster, and that switch errors are always penalized. Phase blocks are highlighted for phase states X (blue) and Y (green).

(a) **Reference** `ATGCTCC`

Query VCF

POS	REF	ALT	GT
2	T	C	1|0
4	C	CG	1|1
6	C	A	1|0

Truth VCF

POS	REF	ALT	GT
2	T	C	1|0
4	C	CGG	1|1
7	C	G	1|1

(b) **Query Graphs** **Truth Sequences**

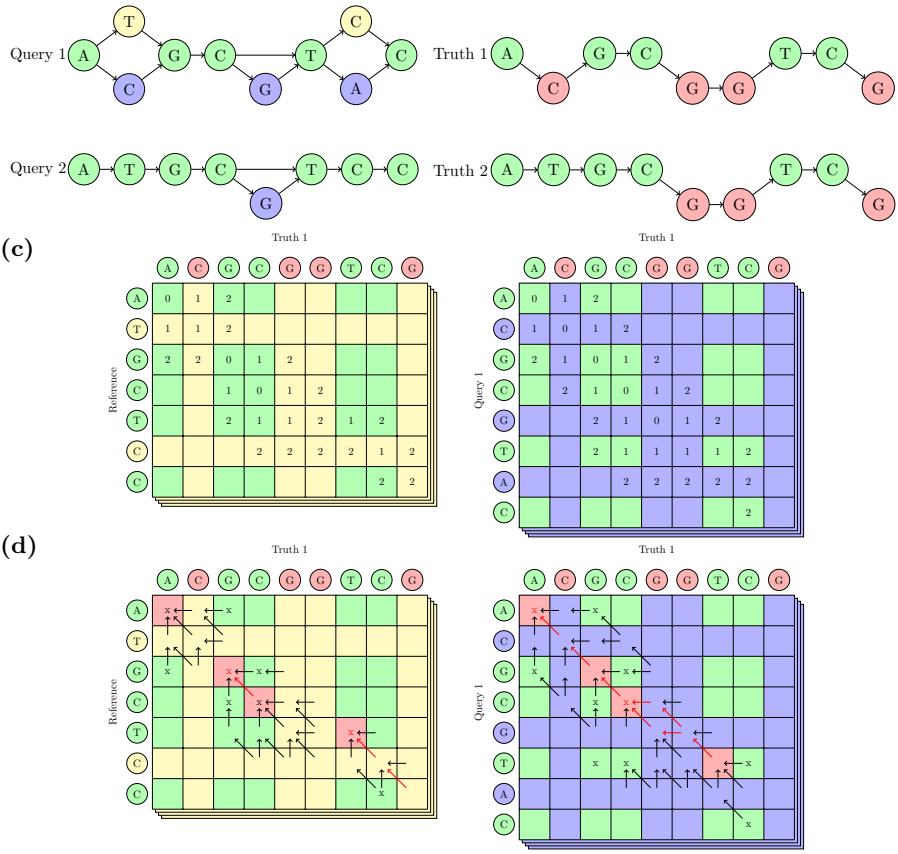


Fig. S10 Overview of minimum edit distance algorithm, allowing false positive query variants. **(a)** Inputs: reference sequence, phased query VCF, and phased truth VCF **(b)** For truth haplotypes, sequences are generated by applying variants (red) to the reference (green). For query haplotypes, variants (blue) are applied similarly but the reference sequence (green/yellow) is also retained, resulting in a graph. **(c)** For each of the four possible alignments of truth and query haplotypes, the forward-pass Dijkstra minimum edit distance algorithm performs alignment using two matrices. Transitions between the two matrices are allowed when both indices are corresponding reference bases (shown in green). **(d)** During the backwards pass, another Dijkstra search is required to maximize the number of false positive variants, among several possible minimum edit distance paths (red arrows). Corresponding reference bases with a single outgoing path edge are labelled "sync points" (red), and are used for calculating summary statistics.