UVC: universality-based calling of small variants using pseudo-neural networks

Xiaofei Zhao[1*], Allison Hu[1], Sizhen Wang[1], and Xiaoyue Wang[2*]

[1]Genetron Health (Beijing) Co. Ltd, Beijing 102208, China
[2]State Key Laboratory of Medical Molecular Biology, Center for Bioinformatics, Institute of Basic Medical Sciences, Chinese Academy of Medical Sciences, Peking Union Medical College, Beijing 100005, China

March 10, 2020

Abstract

We describe UVC (`https://github.com/genetronhealth/uvc`), an open-source method for calling small somatic variants. UVC is aware of both unique molecular identifiers (UMIs) and the tumor-matched normal sample. UVC utilizes the following power-law universality that we discovered: allele fraction is inversely proportional to the cubic root of variant-calling error rate. Moreover, UVC utilizes pseudo-neural network (PNN). PNN is similar to deep neural network but does not require any training data. UVC outperformed Mageri and smCounter2, the state-of-the-art UMI-aware variant callers, on the tumor-only datasets used for publishing these two variant callers. Also, UVC outperformed Mutect2 and Strelka2, the state-of-the-art variant callers for tumor-normal pairs, on the Genome-in-a-Bottle somatic truth sets. UVC outperformed Mutect2 and Strelka2 on 21 *in silico* mixtures simulating 21 combinations of tumor purity and normal purity. Performance is measured by using sensitivity-specificity trade off for all called variants. The improved variant calls generated by UVC from previously published UMI-based sequencing data are able to provide additional biological insight about DNA damage repair. The versatility and robustness of UVC makes it a useful tool for variant calling in clinical settings.

1 Introduction

Accurate detection of somatic mutations is crucial to the diagnosis, prognosis, and treatment monitoring of cancer. Various next-generation sequencing (NGS)-based methods have been routinely used to discover variants at different allele fractions in different types of tumor samples. However, the accuracy of the software that calls these variants is compromised by various biases and errors in NGS experiments. Moreover, some enhanced library preparation techniques for NGS, such as molecular barcoding with unique molecular identifiers (UMIs) and duplex sequencing with duplex UMIs, were applied in clinical settings. Currently, most variant callers are not aware of these enhanced techniques, and the knowledge gained from the literature of duplex sequencing has not been used to improve any variant-calling model.

Here we described and evaluated UVC, a Universal and Versatile variant Caller, which utilizes universality and pseudo-neural network (PNN). Universality is the observation that properties of a large class of systems are independent of the dynamic detail of the system. Here, universality refers to the following observation that we discovered: if the coverage depth of a variant is sufficiently high, then allele fraction is inversely proportional to the cubic root of variant-calling error probability regardless of variant type and error type. PNN resembles a deep neural network in which the weight of each

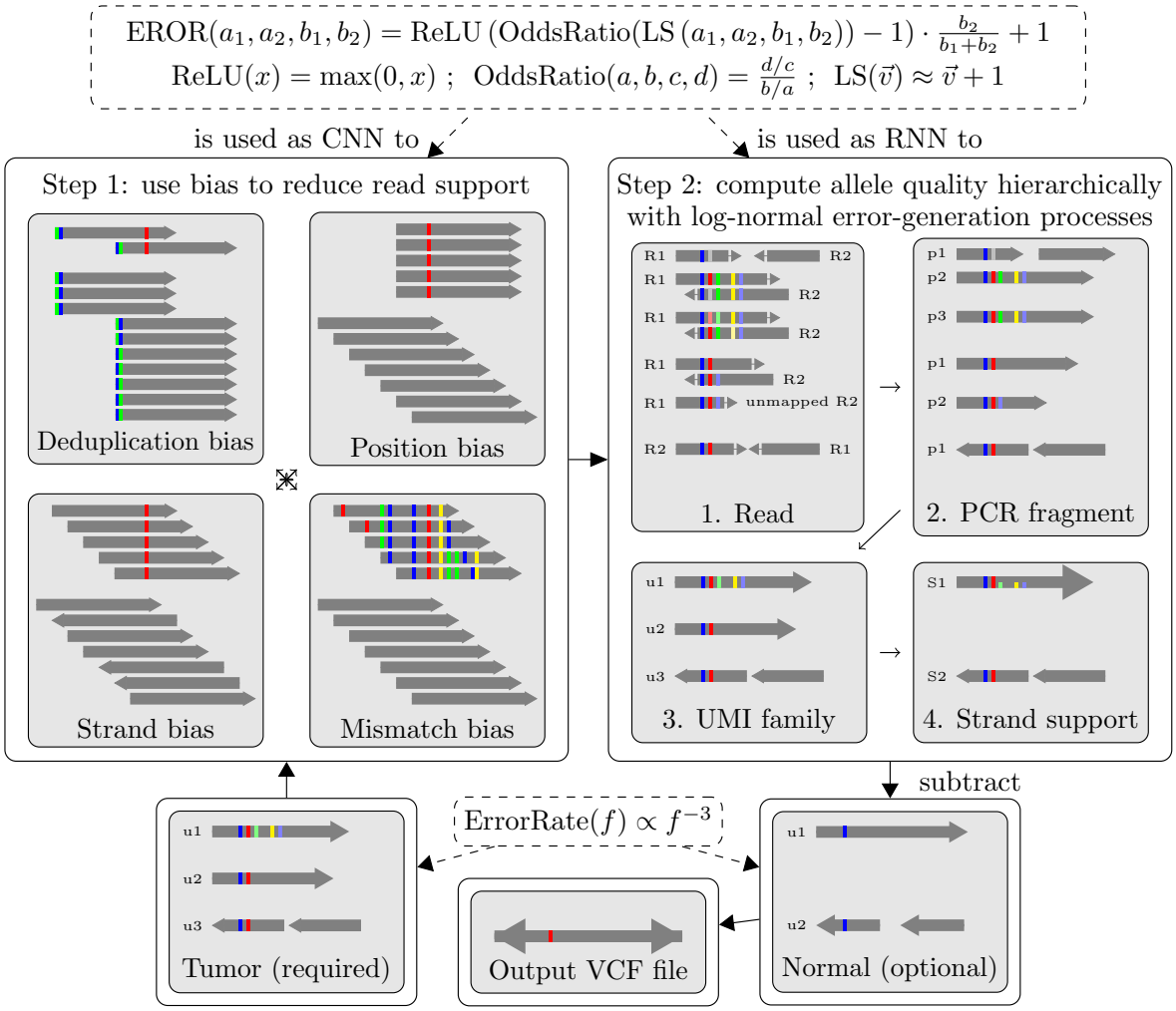


Fig. 1: UVC workflow with the following acronyms: evidence-reduction odds ratio (EROR), rectified linear unit (ReLU), Laplace smoothing (LS), convolutional/recursive neural network (CNN/RNN), unique molecular identifier (UMI), and variant-call format (VCF). Allele fraction is denoted by f.

connection between two neurons is predefined to be a mathematical constant such as one. UVC is able to call somatic single-nucleotide variants (SNVs) and insertion-deletions (InDels) without training data or any prior knowledge. UVC is designed to be aware of both UMI (including duplex UMI) and the tumor-matched normal control (Supplementary Fig. 1). Nevertheless, if UMI and/or the tumor-matched normal control is not provided, UVC can still accurately call variants (Supplementary Fig. 1). In addition, we showed that the UVC-based re-analysis of a published dataset, which was already thoroughly analyzed, can provide additional biological insight. Finally, UVC runs fast, making its usage possible in clinical settings.

2 Results

2.1 Key ideas for improving variant calling

Most variant callers (e.g. VarScan2[20] and MuTect[6]) first detect variants and then check for the existence of any bias in the detected variants. However, a typical manual review of variant candidates based on the read alignments visualized in integrated genomic browser (IGV) first filters out reads characterized by any strong bias and then checks if the remaining reads provide sufficiently high variant signal[2]. Therefore, a manual review is often required to increase the accuracy of variant calling. We reasoned that by following the order of operations recommended by the standard operating procedure of manual review would improve variant calling. Our UVC workflow mimics the process of manual review with IGV. First, UVC discards the reads that support a variant with strong bias. Then, UVC uses the remaining reads to determine the quality of the variant (Fig. 1). Our method resembles a deep neural network in which the weight for each connection between two neurons is some predefined mathematical constant. Consequently, our approach is referred to as PNN.

To adjust the strength of variant support, we used a universal bias function to calculate an evidence-reduction odds ratio (EROR) for each allele. EROR is normalized by read depth, robust to outliers, and able to unify the measures of all biases (e.g., strand bias, position bias, and mismatch bias) in NGS into one single measure. EROR can divide and thus reduce the original number of variant-supporting reads. UVC computes one EROR for each bias and then uses the maximum EROR to reduce variant support. Similarly, a convolutional neural network (CNN) computes one convolution for each local region and then uses max-pooling to reduce feature space[22].

Many variant callers assumed that the signals on different reads are statistically independent, which is not true because one single source of error (e.g., higher rate of oxidative damage to DNA due to technical variability) can affect multiple reads. To model the positive correlation between the errors on different reads, we used lognormal distributions. By using such lognormal model, UVC transforms hierarchically structured base qualities into a variant quality. Similarly, a recursive neural network (RNN) transforms a structured input into an output prediction[16].

In addition, we discovered that deviation from expected allele fraction can be modeled with a power law with the exponent of 3 (Supplementary Fig. 2). This power law corresponds to the process of linear preferential attachment described by Barabási and Albert[1] and is consistent with the artifacts in germline variant calling described by Li[27]. With this power law, the allele fraction of each variant candidate is translated into an upper bound for the probability that the candidate is false positive. In UVC, this power law is universally applied for calling germline SNP, germline InDel, somatic SNV, and somatic InDel.

The variant quality generated by UVC can reliably rank variants by confidence, where variant quality refers to QUAL in the VCF file format specification[8]. Thus, we can simply apply different cutoffs on variant quality to filter variants for different applications, respectively. Furthermore, the variant statistics produced by UVC can be used as input features for both filtering variants (e.g., hard filter in VarScan2[20] and MuTect[6]) and recalling variants (e.g., ensemble mode in NeuSomatic[33]).

2.2 Performance comparison on tumor-only sequencing datasets with UMI

To evaluate the performance of UVC on UMI data, we used the dataset of the physical mixtures (mixed at 1:9 ratio) of the reference standard HD734 and the blood of healthy donors. This dataset was originally published by Mageri[40]. Mageri first trained its model on the dataset of the eight normal samples from the healthy donors, and this dataset is exactly the ground truth for all the true negative calls. Without any training, UVC outperformed Mageri by achieving higher sensitivity (Table 1).

Another UMI-based dataset used for our evaluation is the physical mixture (mixed at 1:99 ratio) of the reference materials NA12878 and NA24385 which are both well-characterized for germline polymorphisms[45]. Xu et al.[45] have shown that smCounter2 outperformed all other applicable methods on this dataset. Therefore, we compared UVC with only smCounter2. The UMI-specific variant caller

Variant caller	Number of false positive calls										
	0	1	2	3	4	5	6	7	8	9	10
UVC	6	5	4	4	3	3	3	3	3	3	2
Mageri	9	9	8	8	8	8	8	7	N/A		

Table 1: Number of false negative calls versus number of false positive calls by varying the threshold of variant quality QUAL. We evaluated UVC and Mageri on the dataset originated from eight HD734 Reference Standard samples and eight normal blood samples from healthy donors. This dataset was described, used, and published by Mageri and is publicly available under the accession SRP064450[40]. A total of 56 true positive variants with variant allele fractions down to 0.1% are expected. A true/false negative is scored if any HD734 variant is called for any of the eight HD734/normal-blood samples. Non-HD734 variants are ignored.

smCounter2 already trained its machine-learning model on multiple independent datasets generated by the same assay on the same sample[45]. Thus, smCounter2 already knows *a priori* the sample-specific, technology-specific, and experiment-specific error profiles. Meanwhile, UVC was not aware of any such error profiles before running with its default parameters. Figure 2 shows that UVC outperformed smCounter2 for calling SNVs and InDels, even though smCounter2 was enhanced with UMI-specific repetitive region filters[45]. Notably, without the hard filter by EROR, UVC would generate approximately 10 times more false positive SNV calls, even though the hard filter by EROR rejects approximately 10% true positive SNV calls (Fig. 2). Hence, the use of EROR for hard filtering is essential for the superior performance of UVC. Interestingly, the EROR threshold of 1.8 is universally applied for both SNV and InDel and for all types of biases including deduplication bias, position bias, strand bias, and mismatch bias, even though these biases seem to be unrelated to each other and traditionally require completely different statistical tests. Therefore, it seems that that EROR requires no further normalization and is therefore self-normalizing.

2.3 Performance comparison on WGS datasets of tumor-normal pairs

To evaluate somatic call accuracy on the whole-genome sequencing (WGS) data of tumor-normal pairs, we used the data derived from the physical mixture (mixed at 3:7 ratio) of NA12878 and NA24385. The mixture is used as the tumor sample, and pure NA24385 is used as the matched normal sample.

Mutect2 has no built-in option to run efficiently with multiple threads, so we manually parallelized the run of Mutect2 over different chromosomes. By default, Strelka2 uses empirical variant score (EVS) to rank variants by confidence. EVS is computed by a random-forest model, and the model was trained on the Platinum Genome dataset. However, the Platinum Genome dataset already included the ground truth variants for the reference materials NA12878 and NA24385. The strong correlation between EVS and the ground truth implies that EVS might not generalize to samples that are not related to the ground truth. For example, the MSI regions with certain characteristics can all be of reference allele in the ground truth but of non-reference allele in some other clinical samples not related to the ground truth. Therefore, we also evaluated the performance of Strelka2 without using EVS.

Figure 3 shows that UVC always generates more true positive calls regardless of how many false positive calls are tolerated for SNVs compared with Mutect2 and Strelka2. Furthermore, compared with Mutect2, UVC always generates more true positive calls regardless of how many false positive calls are tolerated for InDels (Fig. 3). Compared with EVS-enabled Strelka2, UVC achieved better sensitivity and approximately the same specificity for InDels. In terms of running time, UVC is reasonably fast (Supplementary Table 1).

In silico mixtures may be prone to mixing artifact but can model a variety of sequencing scenarios.

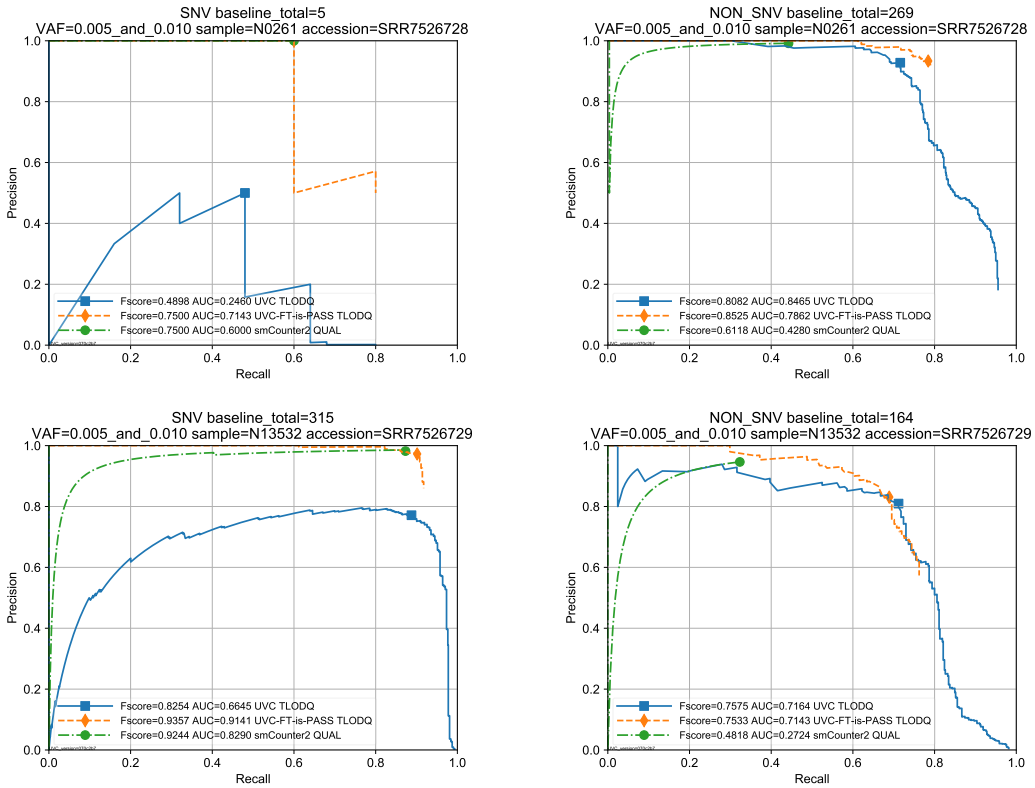


Fig. 2: Evaluation of UVC and smCounter2 on the samples N13532 and N0261 which are both 1%/99% mixtures of the reference materials NA12878/NA24385. This dataset was described, used, and published by smCounter2 and is publicly available under the accession SRP153933 [45]. Homozygous and heterozygous germline polymorphisms present in NA12878 but not in NA24385 are considered to be tumor variants with fractions of 1% and 0.5%, respectively. The keyword "FT-is-PASS" in the legend indicates that every variant call with its EROR > 1.8 was filtered out. The precision-recall (PR) that maximizes F-score is marked. The maximized F-score and AUC (area under the curve) are shown in legends.

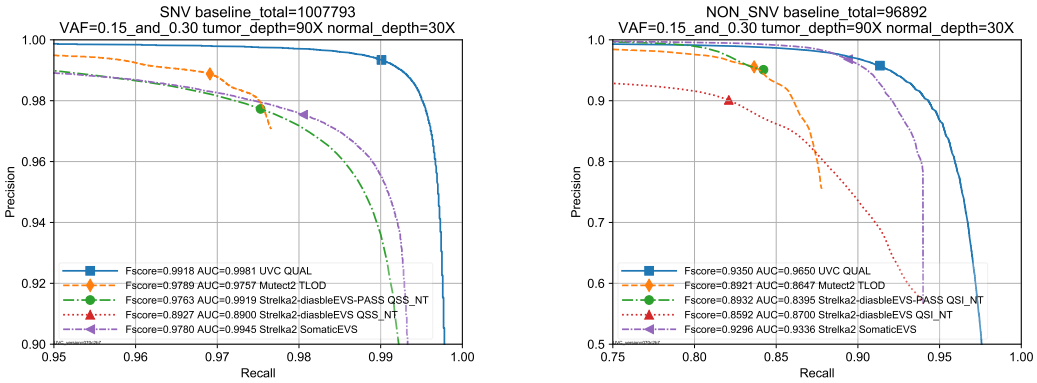


Fig. 3: Performance of UVC, Mutect2, and Strelka2 on the GIAB somatic truth sets. A 30%/70% physical (not *in silico*) mixture of NA12878/NA24385 is used as the tumor sample, pure NA24385 is used as the matched normal sample, and germline polymorphisms present in only NA12878 are considered to be somatic variants[48]. By default, Strelka2 uses empirical variant score (EVS) as variant quality, but the model for EVS was also trained on NA12878 and NA24385[35]. Hence, performance with EVS disabled is also of interest. The keyword "PASS" in the legend indicates that only variants with the FILTER string PASS in the VCF file were kept for evaluation. The maximized F-score and AUC (area under the curve) are shown in legends.

Thus, we used *in silico* mixtures to estimate the performance of UVC, Mutect2, and Strelka2 at various tumor and normal purities. UVC outperformed Mutect2 and Strelka2 on the *in silico* mixed NA24385 and NA12878 especially if the tumor purity and/or the normal purity is low (Table 2 and Supplementary Fig. 3 and 4).

2.4 Discovery of new somatic variants generated by ultraviolet radiation through bioinformatics analysis by UVC

UVC is the only variant caller that is aware of the UMIs in both tumor and normal samples, so we applied it to a ultraviolet (UV)-treated subclonal mutation dataset[12]. In this dataset, SiMSen-seq[14], an UMI-based ultra-sensitive amplicon sequencing technique for liquid biopsy, was used to detect subclonal mutations at selected promoter regions in cells with deficiency in DNA repair. Using the four samples not treated with UV as the "control" and the corresponding four samples treated with UV as "tumor", UVC identified the previously reported subclonal mutations at the *RPL13A* -116 bp `TTCCG` promoter hotspot site with an allele fraction between 0.05% and 0.5% in all four samples (Supplementary Table 2) on default settings. These mutations were all `C>T` transversions with a flanking `TTCCG` sequence, consistent with previously identified mutational signatures caused by UV radiation in cancer[14;9]. On the other hand, if the untreated samples were used as "tumor" and the treated samples were used as "normal", no variant was called by UVC, suggesting a low false positive rate of the variants called on this dataset. Together these results demonstrated that UVC could achieve precise variant calling using a unified model on data generated with different sequencing methods.

Interestingly, in the regions that the original paper did not include, UVC discovered a `C>T` variant before the sequence `TTCCG` 8 bp upstream of the transcription start site of the *DPH3* gene (position 3:16306504 on the GRCh37 genomic coordinates). In the raw data, the mutation is supported by a lot of UMI families in each UV-treated sample but is not supported by any UMI families in each control sample (Supplementary Fig. 5). The newly discovered variant further supports the conclusion made by Elliott et al.[12]: intrinsic properties of UV radiation, instead of locally inhibited DNA repair, is the core mechanism resulting in the high prevalence of `C>T` base substitution before the sequence `TTCCG`.

	Purity		F-Score SNV			AUC-PR SNV			F-Score InDel			AUC-PR InDel		
	Tumor	Normal	UVC	Mutect2	Strelka2	UVC	Mutect2	Strelka2	UVC	Mutect2	Strelka2	UVC	Mutect2	Strelka2
1	0.0196	1.0000	**0.4756**	0.2829	0.3241	**0.3728**	0.1301	0.1913	**0.2987**	0.2113	0.1283	**0.2108**	0.0826	0.0647
2	0.0566	1.0000	**0.8748**	0.8310	0.8522	**0.8899**	0.7201	0.7584	**0.7334**	0.7109	0.6387	**0.7579**	0.5957	0.4989
3	0.1228	1.0000	**0.9893**	0.9770	0.9861	**0.9960**	0.9804	0.9859	**0.9306**	0.9257	0.9097	**0.9590**	0.9115	0.8796
4	0.2063	1.0000	**0.9970**	0.9914	0.9960	**0.9989**	0.9952	0.9959	**0.9765**	0.9647	0.9587	**0.9840**	0.9482	0.9479
5	0.2958	1.0000	**0.9983**	0.9954	0.9976	**0.9994**	0.9967	0.9977	**0.9807**	0.9651	0.9673	**0.9836**	0.9390	0.9534
6	0.3827	1.0000	**0.9987**	0.9969	0.9980	**0.9993**	0.9965	0.9983	**0.9833**	0.9691	0.9728	**0.9829**	0.9437	0.9576
7	0.4624	1.0000	**0.9986**	0.9976	0.9981	**0.9992**	0.9970	0.9988	**0.9800**	0.9675	0.9723	**0.9792**	0.9379	0.9531
8	0.0183	0.9986	**0.4377**	0.2540	0.2851	**0.3296**	0.1147	0.1637	**0.2845**	0.1919	0.1087	**0.1961**	0.0740	0.0525
9	0.0529	0.9958	**0.8290**	0.7907	0.8109	**0.8404**	0.6556	0.6990	**0.6908**	0.6691	0.5683	**0.7047**	0.5280	0.4163
10	0.1152	0.9903	**0.9772**	0.9727	0.9720	**0.9899**	0.9730	0.9650	0.9105	**0.9111**	0.8481	**0.9393**	0.8936	0.7751
11	0.1947	0.9821	**0.9946**	0.9895	0.9745	**0.9987**	0.9950	0.9671	**0.9686**	0.9593	0.8858	**0.9785**	0.9447	0.8308
12	0.2809	0.9714	**0.9951**	0.9938	0.9635	**0.9992**	0.9943	0.9450	**0.9748**	0.9650	0.8746	**0.9820**	0.9431	0.8021
13	0.3657	0.9584	**0.9951**	0.9909	0.9710	**0.9988**	0.9846	0.9547	**0.9787**	0.9677	0.9125	**0.9798**	0.9434	0.8648
14	0.4444	0.9432	**0.9953**	0.9735	0.9859	**0.9991**	0.9501	0.9885	**0.9773**	0.9476	0.9477	**0.9827**	0.8993	0.9286
15	0.0169	0.9972	**0.3794**	0.2100	0.2339	**0.2671**	0.0855	0.1284	**0.2461**	0.1607	0.0784	**0.1569**	0.0574	0.0331
16	0.0491	0.9917	**0.7811**	0.7473	0.7597	**0.7858**	0.5969	0.6313	**0.6187**	0.6120	0.4732	**0.6134**	0.4610	0.3183
17	0.1075	0.9808	**0.9601**	0.9562	0.9138	**0.9806**	0.9579	0.8791	0.8664	**0.8751**	0.7081	**0.8982**	0.8599	0.5849
18	0.1827	0.9649	**0.9921**	0.9871	0.8784	**0.9983**	0.9894	0.7968	**0.9557**	0.9537	0.7067	**0.9673**	0.9349	0.5808
19	0.2654	0.9445	**0.9921**	0.9690	0.8255	**0.9978**	0.9534	0.7305	**0.9583**	0.9335	0.6644	**0.9693**	0.8945	0.5215
20	0.3478	0.9201	**0.9931**	0.9043	0.8718	**0.9980**	0.8287	0.7879	**0.9684**	0.8980	0.7607	**0.9750**	0.8212	0.6494
21	0.4252	0.8925	**0.9908**	0.7883	0.9502	**0.9974**	0.6515	0.9200	**0.9647**	0.7850	0.8600	**0.9705**	0.6441	0.8281

0.2 0.4 0.6 0.8 1.0

Table 2: Performance of UVC compared with Mutect2 and Strelka2 on different scenarios created with the *in silico* mixtures of NA24385 and NA12878. Reads in the alignment files of NA24385 and NA12878 were assumed to be originated from the tumor and normal tissues, respectively. AUC-PR is the area under the curve of precision-recall. More detailed results are provided in Supplementary Fig. 3 and 4.

3 Discussion

Here we demonstrated that UVC, a universality-based variant caller with PNN, is able to call SNVs and InDels with high accuracy. UVC is both UMI-aware and able to compare the signal in the tumor sample against the signal in the matched normal sample. UVC outperformed other state-of-the-art variant callers on the gold-standard datasets in terms of sensitivity-specificity trade off. In addition, the improved variant calling of UVC leads to more comprehensive biological insight about DNA damage repair from a well-analyzed existing dataset. The gain in performance is mainly due to the power-law universality (Eq. (2)) which translates any allele fraction into a false positive rate corresponding to such allele fraction. In addition, the bias equation (Eq. (20)) is also universally applied to a lot of seemingly unrelated tasks such as the quantification of all types of biases.

We hypothesize that the techniques used in UVC can also be used for calling structural variants (SVs) with high accuracy. The reason is that, fundamentally, SV is the same as SNV and InDel: they are all characterized by a reference allele and a non-reference allele, and the reference allele is usually supported by many times more reads than the non-reference allele. Then, for calling SVs, the only additional remaining work is to estimate the probability that a chimeric read is artifactual. In the future, we might implement the feature of calling SVs in UVC.

We hypothesize that the generation of NGS errors shares some fundamental principles with some other seemingly unrelated phenomena associated with the linear preferential attachment proposed by Barabási and Albert [1]. For example, the number of times that a scientific paper has been cited is also distributed according to Eq. (2), a power law with the exponent of 3 [1]. Therefore, the generation of NGS errors and these other phenomena belong to the same universality class, and phenomena in the same universality class can be explained by the same mechanism [3].

We hypothesize that pseudo-neural network (PNN), which is used in the design of UVC, has the potential to provide some fundamental insight into the general design of neural network. Currently, neural network is considered to be a black-box approach for solving problems especially because neural network requires training to optimize the weight of each connection between two neurons. Nevertheless, in PNN, such weight can be some meaningfully predefined mathematical constant. Thus, PNN does not require any training. Therefore, PNN is a white-box approach instead of a black-box approach. Hence, neural networks with certain structures can potentially be less black-box and easier to train.

4 Methods

UVC is designed to be aware of both unique molecular identifiers (UMIs) and the tumor-matched normal (Supplementary Fig. 1). However, neither the presence of UMIs nor the availability of the tumor-matched normal is required for UVC to analyze sequencing data (Supplementary Fig. 1). Thus, UVC uses as much information from the data as possible but does not require any specific information to be present in the data.

First, UVC optionally estimates tumor-in-normal contamination with a simple strategy. Detail of such estimation is presented in Section 4.1.

Unlike other variant callers, UVC assumes that the depth of coverage can be possibly infinite. The assumption of infinite depth reveals a power-law relationship with the exponent of 3 between allele fraction and probability of false-positive variant call for both germline and somatic variants for a wide range of allele fractions. Such power-law is described in more detail in Section 4.2.

Then, the power-law model is integrated with the classical binomial model for calling germline polymorphisms to compute genotypes (GTs) and genotype likelihoods (GLs). Detail about such integration is presented in Section 4.3.

After obtaining sufficiently accurate germline calls, the power-law model is combined with a linear model of variant-evidence reduction and a log-normal model of variant-evidence accumulation for calling somatic variants. Details about variant-evidence reduction and accumulation are presented in Section 4.4 and Section 4.5, respectively. In brief, UVC computes biases first to reduce the evidence for calling variants and then calls variants with the biases-reduced evidence. The reduction of evidence is achieved by a novel use of odds ratio. Odds ratio is a broadly applied mathematical concept especially because of its presence in logistic regression[41]. Here, we heavily used odds ratio. Equation (22) describes how to calculate odds ratio. However, using plain odds ratio without any normalization usually results in an over-estimation of the effect. Therefore, we came up with Eq. (20), which normalizes odds ratio with a combination of Laplace smoothing and neural-network rectifier. Importantly, the rectifier makes Eq. (20) a special neural network. Unlike other neural networks, this neural network does not require any training. Here, we let pseudo-neural network (PNN) denote any such special neural network that does not require any training. It turns out that Eq. (20) and PNN can be applied to a lot of problems in calling somatic variants such as calculation of all biases, computation of variant quality, tumor-normal comparison, etc., even though these problems seem to be unrelated to each other. Here, we will describe the different components of our method and the way that Eq. (20) is applied to each component if applicable.

Next, the comparison between the tumor and its matched normal is presented in Section 4.6. This comparison results in a somatic variant quality used for ranking variants. If the matched normal is not available, then a special empty alignment file with zero depth of coverage at each genetic locus is used as the matched normal.

Finally, differences between calling SNVs and calling InDels are described in Section 4.7.

4.1 Optional estimation of tumor-in-normal contamination

The tumor sample may contaminate its matched normal sample especially if the tumor is liquid and/or at a late stage[35]. Thus, contamination rate can be practically estimated from the origin and stage of the tumor. If such information is not available, then UVC is still able to computationally estimate it. Calling variants requires estimating contamination rate, and yet estimating contamination rate requires calling variants. This problem of circular dependency can be resolved by a bootstrapping algorithm such as expectation maximization. However, to improve runtime and robustness, UVC uses a built-in mini variant caller, referred to as miniVC, to only call SNVs with some simple hard filters. First, for both the tumor and its matched normal, miniVC keeps each raw non-deduplicated read support for SNV if and only if the following conditions are all met.

1. Mapping quality of the read is at least 50.

2. Base quality of the supporting base is at least 7 more than the average base quality of the read.

Then, using the raw non-deduplicated read support for SNV that were kept, each SNV is called by miniVC if and only if the following conditions are all met. Every SNV satisfying the following conditions are kept.

1. Tumor and normal raw total depths are both at least 30X.
2. Tumor allele depth and fraction are at least 5X and 5%, respectively.
3. Tumor allele depth is at least 3 times the normal allele depth.
4. Tumor allele fraction is at least 2.5 times the normal allele fraction.

Finally, given the summed depths of all called variant alleles for both the tumor and its matched normal, the summed depth for the tumor divided by the summed depth for the normal is considered to be the rate of contamination from the tumor to its matched normal.

Although miniVC may be characterized by high false negative rate, it is sufficient for estimating contamination and simple to understand. Hence, we used miniVC to estimate tumor-in-normal contamination if it is required to do so.

4.2 Power-law model of the relationship between allele fraction and false positive probability at infinite depth of coverage

It has been well known that variant candidates with low allele fractions are likely to be false positive. However, the relationship between allele fraction and false positive rate has never been explicitly modeled using a wide range of allele fractions. Mathematical models that fit to a wide range of values exhibit the properties of scale-invariance which is a key characteristic of power-law. Hence, we used a power-law model of false positive rate using allele fraction.

In our power-law model, we assumed that allele depth is infinitely large. Although our assumption is invalid, it allows us to deal with a much simplified version of the complex problem of variant calling. Then, under this assumption, we hypothesized that the observed and expected allele fractions f and g are related to each other as follows.

$$\mathrm{PLE}(f,g) = \left(\max\left(\frac{f}{g}, \frac{g}{f}\right)\right)^{-3} \tag{1}$$

which is equivalent to the following formula in Phred-scale.

$$\mathrm{PLQ}(f,g) = 3 \times 10 \times \log_{10} \min\left(\frac{f}{g}, \frac{g}{f}\right) \tag{2}$$

where $\mathrm{PLE}(f,g)$ and $\mathrm{PLQ}(f,g)$ denote the raw and Phred-scaled likelihoods that the observed allele with a fraction of f is a true biological variant which is expected to have a fraction of g. In other words, if the observed allele fraction f is doubled or halved and deviates further from its expected allele fraction g, then the likelihood that the observed allele is false positive increases by approximately eight. Equation (2) is a power-law distribution with the exponent of 3. The exponent of 3 is at the bifurcation point above/below which the variance of the power-law distribution is finite/infinite, which conforms to our intuition that the allele fractions of false positive variants have extremely large but finite variance. In addition, a power-law distribution with the exponent of 3 can be theoretically generated by the well-known process of linear preferential attachment[1]. Furthermore, a duplex consensus sequence (DCS) which requires two single-strand consensus sequences (SSCSs) has an error rate that is approximately $\frac{1}{10}$ the error rate of an SSCS[36], and $\frac{1}{10}$ is close to the $\frac{1}{2^3}$ predicted by the power law with the exponent of 3. Most importantly, Supplementary Fig. 2 shows that, indeed, the relationship between false positive rate and allele fraction empirically follows a power law with the exponent of 3. As we

will show, Eq. (2) can be applied to many problems in NGS involving allele fractions: calling germline variants in the tumor, calling germline variants in the matched normal, calling somatic variants in the tumor, comparing signal in the tumor with the signal in its matched normal, etc., all using the power law with the exponent of 3.

The genotype (GT) field, which is specified in the VCF file format specification, can be either homozygous reference denoted as homref, heterozygous (including tri-allelic site) denoted as hetero, or homozygous non-reference denoted as homalt at a genetic locus for a diploid living organism such as human[8]. Each GT has its own expected allele fraction g. In addition, the error-generation process, such as sequencing error and cross-sample contamination, is also assumed to have its own expected allele fraction ϵ. Then, given infinite coverage depth, Eq. (2) can be applied as follows to compute the genotype likelihood (GL) specified in the VCF file format specification.

$$\mathrm{GL}_{\mathrm{PowerLaw}}(f, \mathrm{homref}) = 10 \times 3 \times \min\left(0, \log_{10}\left(\frac{\epsilon \times (1 - f')}{f'}\right)\right) \tag{3}$$

$$\mathrm{GL}_{\mathrm{PowerLaw}}(f, \mathrm{hetero}) = 10 \times 3 \times \min\left(0, \log_{10}\left(\min\left(\frac{f'}{1 - f'}, \frac{1 - f'}{f'}\right)\right)\right) \tag{4}$$

$$\mathrm{GL}_{\mathrm{PowerLaw}}(f, \mathrm{homalt}) = 10 \times 3 \times \min\left(0, \log_{10}\left(\frac{\epsilon \times f'}{1 - f'}\right)\right) \tag{5}$$

where f is the observed allele fraction, f' is the normalized allele fraction, and ϵ denotes the allele fraction generated by error processes such as sample contamination. By default, $\epsilon = 0.02$ which is slightly higher than the minimum contamination rate of 0.015 that ContEst recommended based on the evaluation on the TCGA Ovarian dataset[5]. To normalize allele fractions, $f' = f$ for SNVs and $f' = f/(f + (1 - f)/\,\mathrm{AlignBias}(L, S))$ for InDels, where AlignBias is defined in Section 4.7.

Equation (5) indicates that the maximum value for $\mathrm{GL}(f, \mathrm{hetero})$ is zero and that the minimum value for $\max(\mathrm{GL}(f, \mathrm{homref}), \mathrm{GL}(f, \mathrm{homalt}))$ is approximately -51 at $f = 0.5$. Thus, the maximum genotype quality of a heterozygous call is 51, which is within the range of false positive rates between 1/100kb and 1/200kb for the filtered heterozygous calls estimated by Li[27]. Also, 51 is close to the maximum genotype quality of 50 capped by Freebayes[15]. Moreover, Eq. (5) provides a good fit to the distribution of allele imbalance studied by Muyas et al.[32]. In Section 4.3, we will relax the assumption of infinite coverage depth for computing GTs and GLs.

Under the assumption of infinite coverage depth, the germline calling errors were estimated. Thus, we can now rule out inherited germline polymorphisms as a source of error. Thus, in addition of assuming infinite coverage depth, we assumed that germline polymorphisms inherited from biological parents are all called with 100% accuracy. Then, under these two assumptions, Eq. (5) implies that the allele fraction f is heuristically related to its odds of being false positive $\mathrm{PLE}(f)$ as follows.

$$\mathrm{PLE}(f) = 10^{-9} \times f^{-3} \tag{6}$$

which is equivalent to the following formula in Phred-scale.

$$\mathrm{PLQ}(f) = 90 + 3 \times 10 \times \log_{10}(f) \tag{7}$$

where $\mathrm{PLE}(f)$ and $\mathrm{PLQ}(f)$ are the raw and Phred-scale odds that a variant at a base position is false positive. Interestingly, Eq. (7) strongly conforms to our intuition as demonstrated by the following facts.

1. $\mathrm{PLE}(1) \approx 10^{-9}$. The constant 10^{-9} is approximately the rate of germline mosaicism[46], which is approximately the probability that a *de novo* germline mutation, which cannot be distinguished from a somatic mutation, occurs *in vivo* at a base position.

2. For Illumina sequencing data, $\mathrm{PLE}(10^{-3}) \approx 1$ according to the common practice of discarding bases with basecall qualities of less than 30.

3. For any type of sequencing data, $\mathrm{PLE}(0) = \infty$. If a variant candidate is not supported by any read at all at extremely high depth of coverage, then this variant candidate is almost surely a true negative.

Most importantly, Supplementary Fig. 2 shows that Eq. (7) is empirically valid. In Section 4.5, we will show how to relax the assumption of infinite coverage depth.

Unique molecular identifiers (UMIs) are able to detect and correct the NGS errors that occur after the attachment of UMIs into the DNA fragments of interest. Schmitt et al.[36] found that a typical UMI-aware method yielded a mutation frequency of 1.5×10^{-4} per base pair whereas a standard sequencing approach with quality filtering for a Phred score of 30 resulted in a mutation frequency of 2.7×10^{-3} for measuring the mutation rate in mitochondrial DNA. Hence, the application of UMI reduces NGS error by approximately ten-fold. Thus, for UMI data, the power-law distribution in Eq. (7) is rescaled, so each UMI allele fraction corresponds to ten times the non-UMI allele fraction in terms of false positive rate. More specifically,

$$\mathrm{PLE}_{\mathrm{UMI}}(f) = 10^{-9} \times \min(1, 10 \times f)^{-3} \tag{8}$$

which is equivalent to the following formula in Phred-scale.

$$\mathrm{PLQ}_{\mathrm{UMI}}(f) = 90 + 3 \times 10 \times \log_{10}(\min(10 \times f, 1)) \tag{9}$$

In Section 4.5, we will show how to relax the assumption of infinite coverage depth for UMI data.

A matched normal control is able to provide evidence for the presence and/or absence of systematic error. Again, we can apply Eq. (2) to compare tumor and normal at a genetic locus.

$$\mathrm{PLE}(t, n) = \left(\min\left(t/n, 10^{2.5/3}\right)\right)^{-3} \tag{10}$$

which is equivalent to the following formula in Phred-scale.

$$\mathrm{PLQ}(t, n) = 3 \times 10 \times \log_{10}\left(\min\left(t/n, 10^{2.5/3}\right)\right) \tag{11}$$

where t denotes the tumor variant allele fraction (VAF) and n denotes the normal VAF. The number $10^{2.5/3}$ implies that $\frac{t}{n}$ is effectively capped at around 7. In Section 4.6, we will show how to relax the assumption of infinite coverage depth.

4.3 Germline variant call at finite depth of coverage

To call somatic variants, we have to first know if the variant is of germline origin. Therefore, sufficiently accurate germline calling is required. At each genetic locus, classical germline calling with a binomial model can be done as follows with variant allele fraction f and total depth d.

$$\mathrm{GL}_{\mathrm{Binomial}}(f, \mathrm{homref}) = 10/\log(10) \times d \times \mathrm{KL}_{\mathrm{Bernoulli}}(\min(g(\epsilon), f), f) \tag{12}$$

$$\mathrm{GL}_{\mathrm{Binomial}}(f, \mathrm{hetero}) = 10/\log(10) \times d \times \mathrm{KL}_{\mathrm{Bernoulli}}(g(1/2), f) \tag{13}$$

$$\mathrm{GL}_{\mathrm{Binomial}}(f, \mathrm{homalt}) = 10/\log(10) \times d \times \mathrm{KL}_{\mathrm{Bernoulli}}(\max(1 - g(\epsilon), f), f) \tag{14}$$

where $\mathrm{KL}_{\mathrm{Bernoulli}}$ denotes the Kullback–Leibler divergence from an expected Bernoulli distribution to another observed Bernoulli distribution with two rate parameters. More specifically, $\mathrm{KL}_{\mathrm{Bernoulli}}(x, y) = y\log(y/x) + (1 - y)\log((1 - y)/(1 - x))$. Here, we also let $\epsilon = 0.02$ by default to account for any contamination from unknown sources. To normalize allele fractions, $g(x) = x$ for SNVs and $g(x) = x/(x + (1 - x) \times \mathrm{AlignBias}(L, S))$ for InDels, where AlignBias is defined in Section 4.7.

To compute genotype likelihood using both power-law and binomial models, we can combine Eq. (14) and Eq. (5) as follows.

$$\mathrm{GL}(f, \mathrm{homref}) = \max(\mathrm{GL}_{\mathrm{PowerLaw}}(f, \mathrm{homref}), \mathrm{GL}_{\mathrm{Binomial}}(f, \mathrm{homref})) \tag{15}$$

$$\mathrm{GL}(f, \mathrm{hetero}) = \max(\mathrm{GL}_{\mathrm{PowerLaw}}(f, \mathrm{hetero}), \mathrm{GL}_{\mathrm{Binomial}}(f, \mathrm{hetero})) \tag{16}$$

$$\mathrm{GL}(f, \mathrm{homalt}) = \max(\mathrm{GL}_{\mathrm{PowerLaw}}(f, \mathrm{homalt}), \mathrm{GL}_{\mathrm{Binomial}}(f, \mathrm{homalt})) \tag{17}$$

For each alternate (ALT) allele, we can compute its genotype (GT) and genotype likelihoods (GLs) in both the tumor and its matched normal.

First, we assumed that there is no contamination from the tumor into its matched normal. Then, the genotype of the matched normal is either homozygous reference (REF), homozygous ALT, or heterozygous. Therefore, the Phred-scaled likelihood of homozygous reference is $Q^{\text{normal}}_{\text{homref}} = \text{GL}(f, \text{homref}) - \max(\text{GL}(f, \text{homalt}), \text{GL}(f, \text{hetero}))$ for the matched normal. The genotype of the tumor is not well defined because the tumor sample consists of a heterogeneous mixture of tumor and normal cells. Nonetheless, we can still estimate the probability of germline polymorphism at a locus if only the tumor sample is provided. As mentioned by Muyas et al. [32], the probability of a germline polymorphism decreases rapidly as the observed allele fraction deviates from the theoretical probability of observing this germline polymorphism. The Phred-scaled likelihood of homozygous reference is estimated to be $Q^{\text{tumor}}_{\text{homref}} = -3 - \max(\text{GL}(f, \text{homalt}), \text{GL}(f, \text{hetero}))$ for the tumor, where the number -3 normalizes the median of $Q^{\text{tumor}}_{\text{homref}}$ to zero if the true underlying genotype of the locus is either homozygous ALT or heterozygous. Finally, the NLOD is computed to be

$$\max(Q^{\text{tumor}}_{\text{homref}}, Q^{\text{normal}}_{\text{homref}}) \tag{18}$$

The NLOD that we used in UVC has the same interpretation as the NLOD used by Cibulskis et al. [6] in Mutect even though our NLOD is applicable to tumor-only sample.

If there is some contamination from the tumor into its matched normal, then a binomial model of contamination is introduced. More specifically,

$$\text{GL}'_{\text{Binomial}}(f, \text{homref}, \eta, t, n) = \max(\text{GL}_{\text{Binomial}}(f, \text{homref}), (t+n) \times \text{KL}_{\text{Bernoulli}}(\eta, \max(\eta, n/(t+n)))) \tag{19}$$

where η is the contamination rate, t is the variant allele depth in the tumor, and n is the variant allele depth in the normal. In short, $\text{GL}'_{\text{Binomial}}$ is the contamination-aware equivalent of $\text{GL}_{\text{Binomial}}$.

4.4 Somatic variant call at finite depth of coverage - reduction of variant evidence by bias

Bias is common in NGS. Bias may have different interpretations. A typical analysis pipeline run with default parameters filters out variants characterized by any type of strong bias such as strand bias. These filters are referred to as hard filters because each filter lets a variant candidate either pass or fail. Hard filters are easy to implement and interpret but cannot clearly separate true positive variants and false positive variants. Therefore, a different type of filters based on variant quality score recalibration (VQSR) is sometimes used. Given enough training data, VQSR assigns a weight to each type of bias and then considers a combination of all types of biases to reduce the variant quality score (VQS). As a result, the reduced VQS may not be able to pass the VQS threshold, so the reduced VQS implicitly filters out any variant candidate that is overall characterized by strong bias. The VQSR-based filters are referred to as soft filters because a combination of all filters is considered as a whole. However, VQSR is hard to interpret and requires a large amount of training data because VQSR is inherently based on machine-learning.

We designed a novel type of filters called evidence-reduction filters. Evidence-reduction filters are characterized by both of the strengths, but none of the weaknesses, of soft filters and hard filters. Basically, evidence-reduction filter uses bias to reduce the read support associated with the bias, which in turn reduces the evidence for the variant. For each type of bias, evidence-reduction filter produces a bias-induced evidence-reduction odds ratio (EROR) between one and infinity. EROR combines odds ratio, Laplace smoothing, and neural-network rectifier [17] into one single equation. EROR is defined as follows for four positive real numbers as input.

$$\text{EROR}(a_1, a_2, b_1, b_2) = \text{ReLU}\left(\text{OddsRatio}(\text{LS}(a_1, a_2, b_1, b_2)) - 1\right) \times \frac{b_2}{b_1 + b_2} + 1 \tag{20}$$

where

$$\mathrm{ReLU}(x) = \max(0, x) \tag{21}$$

$$\mathrm{OddsRatio}(a_1, a_2, b_1, b_2) = \frac{a_1 \times b_2}{a_2 \times b_1} = \frac{a_1/a_2}{b_1/b_2} = \frac{b_2/b_1}{a_2/a_1} \tag{22}$$

and where the letters a and b denote two properties (such as all alleles and ALT allele, or REF allele and ALT allele), and the subscripted 1 and 2 denote two outcomes (such as forward/+ strand and reverse/- strand). The function LS is defined in Eq. (25) and can be regarded as the identify function if a_1, a_2, b_1, and b_2 are all sufficiently large. The function ReLU is the rectifier, the most commonly used activation function in neural networks[17]. The rectifier zeros out negative input values and does not modify in any way positive input values. Equation (20) has the following important properties if LS is assumed to be the identity function.

1. a_1 is exactly directly proportional to the value of EROR.
2. a_2 is approximately inversely proportional to the value of EROR if EROR is large.
3. In the case $b_1 \ll b_2$, b_1 is approximately inversely proportional to the value of EROR if EROR is large. In the case $b_1 \gg b_2$, the square of b_1 is approximately inversely proportional to value of EROR if EROR is large.
4. b_2 is approximately directly proportional to the value of EROR if EROR is large.
5. If OddsRatio is one or less, then EROR is also one. Hence, the signal b_2 is not reduced if no bias is observed in the signal.
6. If $b_2 \ll b_1$, then EROR equals to one. Hence, the signal b_2 is not reduced if the signal is already too weak.
7. $\lim_{b_2 \to \infty} \frac{\mathrm{EROR}(a_1, a_2, b_1, b_2)}{\mathrm{OddsRatio}(a_1, a_2, b_1, b_2)} = 1$, which means that EROR is approximately equivalent to odds ratio as the signal strength approaches infinity. In addition, $\frac{b_2}{\mathrm{OddsRatio}(a_1, a_2, b_1, b_2)} = \frac{a_1 \times b_1}{a_2}$. Therefore, even if b_2 is an outlier, which means that b_2 can be arbitrary large, the level of evidence normalized (i.e., divided) by EROR still cannot be higher than $(a_1 \times b_1)/a_2$. Therefore, EROR is robust.

Consequently, EROR determines by how much multiplicative factor the number of reads with property b_2 that support the ALT allele should be reduced. More precisely, for EROR, a value of 1 means no reduction, a value of 2 means reduction by half, a value of 3 means reduction by 2/3, etc. As a result, EROR is interpreted as the ratio of the amount of non-bias-adjusted to the amount of bias adjusted evidence. Therefore, EROR can directly reduce the number of reads supporting the ALT allele. EROR has a simple interpretation, requires no training data, is easy to implement, and runs fast. Furthermore, the EROR of each type of bias can be used in not only evidence-reduction filter but also in hard filter and soft filter. The reason is that we can reject variant candidates with high EROR s and use EROR s as the input features to VQSR. Therefore, EROR is backward compatible with the current filtering strategies with soft and/or hard filters.

If $b_1 = 0$ or $a_2 = 0$, then division by zero occurs when using Eq. (22) for computing odds ratio. Add-one smoothing solves this problem of division by zero by adding one to each category, including categories with nonzero counts. The added one is referred to as pseudocount. Laplace smoothing is a generalized version of add-one smoothing. Laplace smoothing adds a positive nonzero number p, which is not necessarily an integer, to each category. The additive Laplace smoothing of two

properties and two outcomes, which correspond to a total of four categories, is defined as follows with the pseudocount p as an additional parameter.

$$\mathrm{LSadd}(a_1, a_2, b_1, b_2, p) = (a_1 + p, a_2 + p, b_1 + p, b_2 + p) \tag{23}$$

Given the known incidence rate of each combination of property and outcome, Laplace smoothing can be further generalized such that a category-specific number is added to each category. This generalized Laplace smoothing uses possibly different nonnegative real numbers as pseudocounts to the different categories. In our case, we would like to assign more weight to the two categories of the property with higher counts because such assignment preserves the property-specific incidence rates. Thus, to incorporate the prior belief about the frequency of occurrence of each property, we came up with the following multiplicative normalization formula based on generalized Laplace smoothing.

$$\mathrm{LSmul}(a_1, a_2, b_1, b_2, p) = (a_1 + \alpha, a_2 + \alpha, b_1 + \beta, b_2 + \beta) \tag{24}$$

where $\alpha = \mathrm{ReLU}\left(p \times \left(\frac{a_1 + a_2}{b_1 + b_2} - 1\right)\right)$ and $\beta = \mathrm{ReLU}\left(p \times \left(\frac{b_1 + b_2}{a_1 + a_2} - 1\right)\right)$. The composition of Eq. (23) and Eq. (24) results in the following normalization formula for handling division by zero.

$$\mathrm{LS}(a_1, a_2, b_1, b_2, p) = \mathrm{LSmul}(\mathrm{LSadd}(a_1, a_2, b_1, b_2, p), p) \tag{25}$$

For each specific type of bias, $p = \frac{1}{2}$ unless stated otherwise. The default value of $\frac{1}{2}$ for p implies that the rate parameter of a binomial distribution is *a priori* distributed according to the Beta($\alpha = \frac{1}{2}, \beta = \frac{1}{2}$) distribution which is itself the Jeffrey's prior of this rate parameter. Jeffrey's prior is statistically uninformative, but we usually do have some prior knowledge about each bias from the scientific literature. For example, cytosine deamination, the major cause of baseline noise in NGS, is unlikely to occur at high allele fraction and can be detected by read-orientation strand bias[4;10]. Thus, p can be set to a value that deviates from $\frac{1}{2}$ as a function of the observed allele fraction, which corresponds to a Bayesian update of the Jeffrey's prior after incorporating some knowledge about the bias of interest.

In fact, every application of the ReLU function can be regarded as an attempt to activate an artificial neuron. Thus, Eq. (20) is composed of three non-degenerated neurons: two neurons are respectively used for computing α and β, and a third neuron is used for subsequently computing EROR. The terms ± 1 and $\pm p$ are equivalent to degenerate neurons that always produce 1 and p, respectively. The term $\frac{b_2}{b_1+b_2} + 1$ applies linear weighting to the output of the third neuron. Hence, the function OddsRatio, which uses division, is the only component that is outside the scope of a typical neural network. A typical neural network does not use division presumably because division is not numerically stable. However, our use of Laplace smoothing ensures that division and EROR are stable. Hence, we introduced OddsRatio in the computation of EROR. We will show that this OddsRatio-supplemented neural network, which corresponds to Eq. (20), can be used for many different tasks in the same manner, which is similar to the way that convolutional neural networks (CNNs) repeatedly uses the same network for extracting features in an image[43]. Because of its lack of requirement for any training data, this OddsRatio-supplemented neural network based on EROR is a type of PNN.

Usually, $a_1 + a_2 \geq b_1 + b_2$ because the depth of the variant allele is always at most the summed depth of all alleles and almost always at most the depth of the reference allele. In this case,

$$\mathrm{LS}\left(a_1, a_2, b_1, b_2, p\right) = \left(a_1 + \frac{a_1 + a_2}{b_1 + b_2} \times p, a_2 + \frac{a_1 + a_2}{b_1 + b_2} \times p, b_1 + p, b_2 + p\right) \tag{26}$$

which corresponds to the additive smoothing generalized to the case of known incidence rates.

Different types of NGS biases may not be independent of each other. For example, position bias and read-orientation strand bias may occur together. Thus, we conservatively assumed that different types of biases may be completely correlated with each other. Thus, we used the maximum value of all biases as the final combined bias so that evidence from read support is not over-reduced. This maximization is exactly equivalent to the max-pooling scheme typically used in a deep convolutional neural network[43]. In the next few subsections, we will show how EROR is used for each type of bias.

4.4.1 Deduplication bias

Normally, all original molecules should be amplified with approximately the same efficiency regardless of the alleles that the original molecules support. Therefore, if the molecules supporting one allele B are under-amplified with respect to the molecules supporting another allele A, then the molecules supporting B are likely observed due to some artifacts. The bias associated with the under-amplification of an allele is referred to as deduplication bias. More precisely, deduplication bias is calculated with Eq. (20). For deduplication bias, we substituted the variables in Eq. (20) as follows: a_1 is the total read depth that could be subtracted during the process of removing duplicates, a_2 is the total read depth after performing this subtraction, b_1 is the allele-specific read depth that could be subtracted during the process of removing duplicates, and b_2 is the allele-specific read depth after performing this subtraction.

Deduplication bias is especially effective for removing false positives in the data generated by sequencing a lot of over-amplified PCR products. For example, the false positive variants in the data from SRP064450[40] are overall characterized by a high level of deduplication bias.

4.4.2 Position bias with multiple position-odds ratios

Position bias occurs if there is strong correlation between the identity of an allele and the position of the allele in the reads supporting the allele. The canonical method to detect position bias is based on the Mann-Whitney rank sum test. Unless explicitly stated otherwise, rank sum test denotes the Mann-Whitney rank sum test. Rank sum test checks whether there is significant difference in the values of two populations of normally distributed data. In other words, rank sum test determines if the values in one population are stochastically greater than the values in the other population. Rank sum test produces a P value. Therefore, rank sum test is sample-size dependent. Moreover, rank sum test cannot check if two distributions have different shapes. For example, rank sum test is not able to determine that the two distributions $\text{Normal}(\mu = 100, \sigma^2 = 1)$ and $\text{Normal}(\mu = 100, \sigma^2 = 100)$ are different.

Instead of using a formal statistical test by assuming a parametric distribution, UVC mimics how humans manually check for position bias by visualizing with a tool such as the integrated genomic viewer (IGV). In fact, humans check if the alignments of reads supporting one allele end more prematurely than the alignments of reads supporting another allele. Therefore, we pre-specified some cutoffs for the distance, or equivalently the number of bases, between the position of an allele in a read and the leftmost/rightmost aligned position of the read, where the distance is one for any allele at the end of a read. In UVC, the cutoffs are hard-coded to be $\frac{1\times2}{2}, \frac{2\times3}{2}, \frac{3\times4}{2}, \dots, \frac{11\times12}{2}$. Then, for each pre-specified cutoff, UVC calculates its EROR. More precisely, for a group of reads covering different alleles at a given position, the position bias for each cutoff is calculated with Eq. (20). For position bias, we substituted the variables in Eq. (20) as follows: a_1 is the total number of reads longer than the cutoff and supporting any allele, a_2 is the total number of reads shorter than or equal to the cutoff and supporting any allele, b_1 is the number of reads longer than the cutoff and supporting the variant allele under test, and b_2 is the number of reads shorter than or equal to the cutoff and supporting the variant allele under test.

However, the mutation itself can result in position bias. In fact, long InDels are especially known to cause a lot of position bias. Therefore, we used running minimum, or equivalently sliding window minimum, on the ERORs associated with the pre-specified cutoffs. This minimization improves recall because position bias has to affect multiple consecutive cutoffs to pass the filtering by sliding window. Then, the final EROR for position bias is defined as the maximum of the running minima of cutoff ERORs. This maximization ensures good precision because position bias can occur at any cutoff value. The default size of the sliding window is 3.

Interestingly, similar to a CNN, the minimization of ERORs corresponds to a layer of min-pooling and the maximum of running minima corresponds to a subsequent layer of max pooling[43], where min pooling is equivalent to max pooling through the following identity: $\min(x_1, x_2, \dots) =$

$-\max(-x_1, -x_2, \dots)$.

4.4.3 Mismatch bias

Mismatch bias occurs if the reads supporting the variant allele are characterized by a much higher number of mismatches than the reads supporting the non-variant alleles. Thus, mismatches that occur in the reads supporting the non-variant/variant allele reduce/increase mismatch bias.

More precisely, for a group of reads covering different alleles at a given position, the mismatch bias for each cutoff is calculated with Eq. (20). For mismatch bias, we substituted the variables in Eq. (20) as follows: a_1 is the total number of reads with less than x number of mismatches and supporting any allele, a_2 is the total number of reads with at least x number of mismatches and supporting any allele, b_1 is the number of reads with less than x number of mismatches and supporting the variant allele, and b_2 is the number of reads with at least x number of mismatches and supporting the variant allele. Here, x varies from 1 to 11, a running minimum of d mismatch biases is computed for each of the 7 windows, and then the maximum of these 7 running minima is the final mismatch bias. Variants with lower allele fractions are more likely to be false positive. Thus, for a given allele fraction f, the value for d is heuristically computed as follow.

$$d = \min(\max(4, 4 + \lceil \log_5(40 \times f) \rceil), 6) \tag{27}$$

The number d means that a read supporting the variant allele must have at least d more mismatches than another read supporting the non-variant allele to be considered as biased. Interestingly, both position bias and mismatch bias are characterized by the max-pooling scheme of CNN in the same manner[43].

Mismatch bias is similar to the High Discrepancy Region (HDR) or Multiple Mismatches (MM) tags presented by Barnell et al.[2]. However, mismatch bias does not reject true somatic variants in hypervariable regions such as human leukocyte antigen (HLA) regions because mismatches in reads supporting the reference allele counter-balance the mismatches in reads supporting the variant allele.

4.4.4 Strand bias

First and foremost, there are two types of strand bias: sequencing strand bias and read-orientation strand bias. Sequencing strand is the strand of the genome to which the sequenced read end, which can be either the R1 end or the R2 end, is mapped to. Read-orientation strand is the strand of the genome to which the PCR-fragment is mapped to. Consequently, the R1 end in forward direction and the R2 end in reverse direction both have the same read-orientation, and vice versa. Unless explicitly stated as sequencing strand bias, strand bias denotes read-orientation strand bias.

Strand bias occurs if two strands support the same allele with different levels of evidence after normalization by read depth. The canonical method to detect strand bias is based on Fisher's exact test. Fisher's exact test checks whether the following null hypothesis is true: strands and variant allele fractions (VAFs) are independent. Fisher's exact test will produce small P values if the two VAFs respectively observed on the two strands significantly differ. Fisher's exact test considers read depth in such a way that higher read depth is more likely to result in rejection of the null hypothesis.

Although statistically sound, Fisher's exact test does not consider effect size. Therefore, similar to many statistical tests that are based on P values, Fisher's exact test will be able to detect infinitesimal correlation between strand and VAF given sufficiently high read depth. This behavior is not intended because a low level of strand bias is expected to intrinsically exist in NGS data. Therefore, GATK used the symmetrical strand-odds-ratio (SOR) for quantifying strand bias. SOR is defined as follows.

$$\mathrm{SOR}(a_1, a_2, b_1, b_2) = \left(\frac{a_2 \times b_1}{a_1 \times b_2} + \frac{a_1 \times b_2}{a_2 \times b_1} \right) \times \frac{\min(a_1, a_2)}{\max(a_1, a_2)} \times \frac{\max(b_1, b_2)}{\min(b_1, b_2)} \tag{28}$$

where a_1 is the depth of REF on the forward strand, a_2 is the depth of REF on the reverse strand, b_1 is the depth of ALT on the forward strand, and b_2 is the depth of ALT on the reverse strand. SOR

performs better than Fisher's exact test for high or uneven read depth especially for positions near the end of an exon. However, SOR still cannot differentiate between false positives and false negatives that a simple manual check can differentiate. For example, let $\alpha = \mathrm{SOR}(1000, 1000, 1, 5)$ and let $\beta = \mathrm{SOR}(1000, 5, 1000, 1)$. Then, $\alpha = \beta = 26$ due to the symmetry in Eq. (28), but the four depths used to calculate α provide much less evidence to support the variant candidate than the four depths used to calculate β. For detecting somatic variants, a manual check would label the α-associated variant candidate as false positive and the β-associated variant candidate as true positive. Even for the purpose of detecting germline polymorphisms, the β-associated variant candidate would still be labeled as true positive for heterozygosity. In this case, the hard filtering by SOR would generate either one false positive or one false negative. Soft filtering by SOR has the potential, but is not guaranteed, to reduce the weight given to SOR to alleviate the problem associated with this anomaly.

We explicitly broke the symmetry in Equation (28) to come up with Eq. (20). Equation (20) is simpler but has none of the anomalies found in Eq. (28). To test if a strand of interest is biased, we substituted the variables in Eq. (20) as follows: a_1 is the summed depth of all alleles on the other strand, a_2 is the summed depth of all alleles on the strand of interest, b_1 is the depth of the ALT allele on the other strand, and b_2 is the depth of the ALT allele on the strand of interest. For strand bias, $p = \frac{1}{2} + \min(2, 100 \times f)$, which incorporates the prior belief that a variant with higher allele fraction f is less likely to be strand-biased[4;10].

An edge case exists for strand bias: if the strand with low VAF is characterized by some other bias, but the strand with high VAF is not characterized by any other bias, then the evidence on the strand with high VAF should not be reduced, which means that the strand with high VAF is not biased. The reason is that a second bias found within a first bias reduces the strength of the first bias by counteraction. The strength of counteraction is assumed to be proportional to: the average statistic (which can be the inverse of the sequencing-error probability averaged on the Phred-scale, and/or the number of base positions to read end) of the variant allele on the bias-suspected strand with high VAF, divided by the average statistic of all alleles on the other strand with low VAF. Moreover, the constant of proportionality is exactly one, the maximum strength of counteraction is considered to be the final strength of counteraction, and the counteraction-normalized strand bias is simply the product of the raw strand bias and the strength of counteraction. For example, if a bias-suspected strand has a VAF of 10% with an average variant-allele base quality of 35, and the other strand has a VAF of 1% with an average all-allele base quality of 15, then the strength of counteraction is $10^{35-15} = 100$, implying that the EROR for strand bias is reduced by 100. In this case, the normalization by the counteraction induced by base quality bias nullifies an otherwise strong strand bias.

4.5 Somatic variant call at finite depth of coverage - accumulation of variant evidence by hierarchical quality adjustment

As previously mentioned, Eq. (20) can calculate any bias (duplication bias, position bias, strand bias, etc.). Surprisingly, as we will show, Eq. (20) can be applied to even more problems. In fact, Eq. (20) is able to calculate not only bias but also quality. The reason is that the existence of true variant signal manifests itself as a bias with respect to background errors. Hence, we can compute any type of quality, such as variant quality, from its corresponding signal-to-noise odds ratio with the formula for the evidence-reduction odds ratio (EROR) presented in Eq. (20).

The fraction of NGS errors at a given position is usually modeled with a beta-binomial distribution[39]. Unfortunately, as shown in Figure 2c of Xu et al.[45], the distribution of errors may be more heavy-tailed than the beta distribution. Hence, a more heavy-tailed distribution provides a better model for the distribution of NGS errors. In fact, the number of products produced by a typical series of physical or chemical reactions tend to follow a log-normal distribution[28]. As a matter of fact, all errors in NGS are produced by physical or chemical reactions. Thus, the number of independent events that generated a significant quantity of NGS errors is probably logarithmic with respect to the signal-to-noise ratio. Equivalently, the logarithm of the raw sample size of the NGS errors is probably

proportional to the effective sample size of the same errors if the raw sample size is sufficiently large. Nonetheless, when the quantity of NGS errors is small, the number of errors should be exactly equal to the number of independent events that generated these errors. The following formula for computing quality considers both small amount of errors and large amount of errors.

$$\mathrm{CQual}(a_1, a_2, b_1, b_2, r) = \mathrm{dlog}\left(\mathrm{EROR}(a'_1, a'_2, b_1, b_2), r\right) \times 10 \times \log_{10}\left(\mathrm{OddsRatio}(a'_1, a'_2, b_1, b_2)\right) \quad (29)$$

where $a'_1 = a_1 \times b_1 \div a_1 = b_1$, $a'_2 = a_2 \times b_1 \div a_1$, the function EROR is defined in Eq. (20), the function OddsRatio is defined in Eq. (22), and where the function dlog is defined as follows.

$$\mathrm{dlog}(n, r) = \log_r\left(n \times (r-1) + 1\right). \quad (30)$$

In the above equation, a_1 is the number of expected reads supporting any allele, a_2 is the number of expected reads supporting the variant allele, b_1 is the number of observed reads supporting any allele, and b_2 is the number of observed reads supporting the variant allele. In UVC, a_1 and a_2 are both multiplied by the same constant $b_1 \div a_1$ to generate a'_1 and b'_1 such that $a'_1 = b_1$. This multiplication removes the effect of the number of expected reads on the computation of quality. Unless explicitly stated otherwise, the observed reads are assumed to be deduplicated or collapsed by consensus so that PCR products presumably derived from the same template molecule were merged into one read. The function $\mathrm{dlog}(n, r)$ answers the following question: how many terms in a geometric series with ratio r and first term equaling to one will produce a sum of n. In brief, this geometric series captures the phenomena that every error increases the likelihood of the subsequent errors, and the effect of such increase becomes more apparent as a larger quality of errors is observed.

Intuitively, the variable r should be larger for errors that occur more downstream in an assay, and vice versa, because more downstream errors are more dependent of each other. Here, we would assign different values of r to different scenarios and justify our assignment. The sources of errors that were present after sample preparation (such as oxidative damage related to the sample storage condition) but before the first cycle of PCR amplification affect the entire sample and is sample-specific. The sample-specific error is subject to sample-to-sample variation. Therefore, different observations of the same sample-specific error, such as the same nucleotide substitution in different deduplicated read families, are not independent of each other. Here, we let $r = 1.5$ and $r = 1.25$ for comparing different deduplicated read families in the same sample for amplicon-based and capture-based (including whole-genome sequencing (WGS)) assays, respectively. Amplicon-based assays generate read fragments that start and end at the same positions, whereas capture-based assays generated read fragments that have different start and/or end positions, so fragments generated by amplicon sequencing are less independent of each other and therefore are characterized by a higher r. Li used the constant 0.85 in his Gompertz model for the correlation between different reads, and his 0.85 is approximately equivalent to our 1.25 when the coverage is low[25]. The sources of errors that were present after the first cycle of PCR amplification but before the sequencing stage affect a subset of reads in the corresponding read family of PCR duplicates and are family-specific. One example of such family-specific errors is base incorporation error. The family-specific error is subject to family-to-family variations. The family-specific error strongly depends on the PCR cycle at which the error is generated, as each PCR cycle approximately doubles the number of erroneous raw reads. Hence, we let $r = 2$ for comparing different raw non-deduplicated reads in the same read family of PCR duplicates. The sources of errors at the basecall stage can affect both R1 and R2 ends of a paired read. The errors on R1 and R2 are highly correlated with each other. Hence, we let $r = \infty$ for merging R1 and R2 base qualities into one base quality.

Tuning the parameter r with a machine-learning algorithm, such as grid search or bisection search, is possible at the cost of some additional runtime. However, we found that the aforementioned values assigned to r work well in practice for all sequencing data. Hence, we simply used the aforementioned values by default.

In Section 4.4, we showed that the function EROR resembles a neural network. Hence, the function CQual, which uses EROR, is also related to this neural network. In fact, the function CQual in Eq. (29)

is similar to a recursive neural network (RNN). The reason is that CQual is repeatedly applied for combining qualities at different hierarchies and each hierarchy is characterized by its own value for the parameter r. In other words, CQual is recursively applied to combine R1 and R2 read ends into a PCR fragment, to combine PCR fragments into a UMI family, and to combine UMI families into a single-strand allele support. Such recursion is similar to the recursion in RNN. Because of its lack of requirement for any training data, this lognormal-supplemented neural network based on CQual is a type of PNN.

4.5.1 Base quality adjustment by base alignment quality (BAQ)

First, UVC adjusts base quality by base alignment quality (BAQ). Given a base quality q and a number d of bases to the closest end (either left or right end) of the read, the BAQ-adjusted base quality is estimated to be $\min(q, 4 \times d)$. The term $4 \times d$ represents the Phred-scaled probability that the read sequence from the position of interest to the end of the read is mapped by random chance. Our BAQ formula is less accurate, but also much simpler and faster to compute, than the one presented by Li [24]. In addition, our BAQ formula works well in practice and is therefore used.

4.5.2 From paired-end reads to merged read

The Illumina and BGI sequencers sequence each fragment twice, one time from each end. Hence, the middle region of a fragment, or even the entire fragment, can be sequenced twice. The twice-sequenced region should be counted only once instead of twice. At the same time, the quality of each base in the twice-sequenced region needs to be adjusted. There are various schemes of adjusting such base qualities [29;30;47]. However, all these various schemes assumed that errors coming from the two read ends are independent of each other. Such assumption is quite strong as a lot of sources of errors, such as PCR errors, can affect both ends. Instead, we assumed that there is up to 100% correlation between the errors on both ends. If the two bases from both ends agree at a position, then the quality of the merged base is simply the maximum of the qualities of the two bases. If the two bases from both ends disagree at a position, then the most likely non-erroneous base is the one with higher base quality, and the difference of the two qualities of the two disagreeing bases is approximately the Phred-scale probability that the most likely non-erroneous base is erroneous. The above reasoning leads to the following formula for computing the combined fragment quality in the twice-sequenced region where the R1 and R2 ends overlap with each other.

$$\text{R1R2overlapQ}(q_1, q_2) = \begin{cases} \max(q_1, q_2) & \text{if } b_1 = b_2 \\ \max(q_1, q_2) - \min(q_1, q_2) & \text{if } b_1 \neq b_2 \end{cases} \tag{31}$$

where q_1 and q_2 are the qualities of the bases b_1 and b_2 from the paired-ends R1 and R2, respectively, such that b_1 and b_2 originated from the same fragment. If the data of interest is produced by single-end sequencing or the region of interest is only covered by either the R1 end or the R2 end of a fragment, then Eq. (31) is not applicable and is thus not used.

We can derive Eq. (31) from Eq. (29) by letting r approach ∞. In fact, $\lim_{r\to\infty} \text{dlog}(n, r) = 1$ so that $\lim_{r\to\infty} \text{CQual}(a_1, a_2, b_1, b_2, r) = 10 \times \log_{10}(\text{OddsRatio}(a_1, a_2, b_1, b_2))$. Let us keep only the base with the quality of $\max(q_1, q_2)$. In this case, if $b_1 = b_2$, then $\text{OddsRatio}(a_1, a_2, b_1, b_2) = \max(q_1, q_2)$ because the base with the quality of $\min(q_1, q_2)$ is discarded as it provides no additional information. Otherwise, $\text{OddsRatio}(a_1, a_2, b_1, b_2) = \max(q_1, q_2) - \min(q_1, q_2)$, where the subtraction corresponds to a Bayesian update after observing the disagreement between the two bases.

4.5.3 From merged reads to deduplicated reads: finding duplicates based on read densities

Typically, reads mapped to the same chromosome, insert start position, and insert end position are assumed to be duplicates of each other. Unfortunately, this assumption may not hold as Sena [37] observed

the following paradox: the PCR products presumably derived from the same molecule according to their UMIs may map to slightly different genomic locations. This paradox can be partially explained by PCR stutter[37]. In other words, if a large number of reads span the same genomic interval, then any other read spanning a slightly different genomic interval should also be considered as a duplicate of the large number of reads. Therefore, to better cluster reads into duplicate families, we developed a novel density-based deduplication algorithm.

First, at each genomic position, the number of reads that start and/or end at this position is computed. Then, each position, referred to as the centroid, attracts other nearly positions, referred to as the satellites. The attraction strength is directly proportional to the number x of reads that start and/or end at the centroid, is inversely proportional to the number y of reads that start and/or end at the satellite, and decays exponentially as a function of the number z of positions between the centroid and the satellite. In UVC, both constants of proportionality are set to 1, and the exponential decay constant is set to ln(5). More specifically, the attraction strength s is defined as follows.

$$s = \frac{x}{y} \times 5^{-z} \tag{32}$$

Positions that are strongly attracted to a second position (i.e., $s > 1$) are considered to be in the same duplicate family as the second position and are therefore clustered with the second position. And the second position is the centroid of the cluster if the second position is not attracted to any other position. In sum, the clustering procedure is similar to the one presented in Figure 2 of Edgar and Flyvbjerg[11], except that correction is performed for start and end positions instead of sequencing errors. This clustering procedure is supposed to especially improve the deduplication of PCR-amplified and UMI-labeled reads.

4.5.4 From merged reads to deduplicated reads: adaptive base-quality filter for duplicates

Filter by some type of quality is an important step in the analysis of NGS data. Low-quality bases are assumed to be generated by basecalling errors during sequencing and are therefore filtered out. Similarly, UMI families that are too small in size are assumed to be of low quality and are therefore also filtered out. Unfortunately, quality-based filter is commonly done with some rules of thumb. The reason is that no theoretical foundation exists for filtering by quality. In fact, different positions may have different optimal base quality thresholds for filtering. In this subsection, we will elaborate on the definition and computation of optimal base-quality and UMI-quality thresholds.

First, there is a one-to-one correspondence between quality and error probability because quality is inherently defined as error probability in PHRED scale. An error-probability threshold is defined in such a way that most false positives do not pass the threshold and most true positives do pass the threshold. Let us assume that different errors corresponding to different qualities are generated independently. Then, each quality that passes the threshold corresponds to a likelihood ratio of the probability that the observed signal is erroneous to the probability that the observed signal is correct. In the worst case, the quality barely passes the threshold by the smallest margin. Therefore, we can assume the following: the quality threshold corresponds to a probability that the observed signal is erroneous, referred to as error probability p which is close to zero. Then, $(1-p)/p \approx 1/p$ is the likelihood ratio of non-error probability to error probability. Nonetheless, the number n of observed signals (e.g., number of reads covering a given position) also affects the overall quality of the observed signals as a whole. Because of the aforementioned assumption of statistical independence, we can multiply the likelihood ratios respectively corresponding to multiple different observed signals together. The result of such multiplication is maximized at a certain quality threshold. This threshold, in theory, also maximizes the ratio of the likelihood that the observed signals are all generated non-erroneously to the likelihood that the observed signals are all generated erroneously. More formally, the process of generating n observed signals, such that each value has p probability to be generated erroneously, is modeled as a Bernoulli process of n variables with an error probability of p. This model

results in the following formula for determining optimal quality threshold.

$$\text{thres}(P) = \arg\max_{p \in P} \left(p^{-|\{p' \in P | p' \leq p\}|} \right) \tag{33}$$

where P is the set of all error probabilities of all observations and thres calculates the threshold to be used for filtering out the observations corresponding to P. Equation (33) provides a formal definition of a "good" basecall-quality scoring scheme proposed by Ewing and Green [13].

4.5.5 From merged reads to deduplicated reads: base quality score adjustment (BQSA) for duplicates

Base Quality Score Recalibration (BQSR) refers to the process of detecting and correcting systematic errors generated by the sequencer when the sequencer estimates the error probability of each base call. Typically, tools such as GATK exclude known sequence variations in the sequencing data. Then, any remaining variation with respect to the known reference sequence is treated as error. Afterwards, all these remaining errors are assumed to be base call errors, and the correlation between error rate and other observed values, such as the non-calibrated base quality, sequencing cycle, sequence context (identity of the bases near each remaining error, whether downstream or upstream of the error), etc. are considered to assign a potentially different quality score to each non-calibrated base.

In short, BQSR uses the empirically observed quality score to recalibrate the quality score estimated by the sequencer. However, BQSR only partially captures site-to-site variations through sequence context. Without any additional information, we cannot obtain more accurate base qualities. However, with unique molecular identifiers (UMIs), we can further improve the estimation of base qualities by combining different base qualities into a consensus base quality.

UMIs are barcodes attached to the original template molecules before PCR amplification of these template molecules occurs. UMIs were first introduced as a technique for detecting and filtering out PCR and sequencing errors in Safe-SeqS [19]. Basically, different short sequences of nucleotides, known as molecular barcodes or UMIs, were attached to the original DNA fragments before the PCR amplification of these fragments. Afterwards, any error arising during the downstream experimental procedure can be detected by redundant sequencing: the multiple PCR products derived from the same original molecule should have the same sequence if the UMI part of the sequence is excluded from sequence comparison. It is important to notice that any error introduced during the first PCR cycle is clonally amplified and therefore not detected by UMI.

If a UMI family is of sufficiently large size (i.e., has many PCR products), then a consensus base with high confidence is usually observed. Hence, in the same UMI family, any base that is different from its consensus base represents an error generated during or after PCR amplification. As a result, the empirical error rate at each position can be estimated as the number of erroneous bases divided by the total number of bases. If we include low-quality bases which are supposed to be outlier bases characterized by high error rates, then the position-wise empirical error rate, which is sensitive to outlier bases with high error rate, is also high. Therefore, we used Eq. (33) described in Section 4.5.4 to filter out low-quality bases because such bases introduce more noise than signal.

A base-quality threshold t used for filtering is also generated by the process of filtering. The bases that remain after filtering are used to compute the empirical error rate of each base at each position. Because each PCR cycle approximately doubles the number of PCR products, we let $r = 2$ and then use Eq. (29) to compute the effective number of raw reads in each UMI family. If the assay involves no UMI, then raw reads that are duplicates of each other are assumed to be in the same UMI family. Then, the effective number of reads is multiplied by t to generate the quality of the corresponding UMI. More specifically, the formula for computing UMI quality is as follows.

$$\text{UMIQ}(b_1, b_2, t) = \text{CQual}(b_1, b_1 \times 10^{-t/10}, b_1, b_2, r = 2) \tag{34}$$

where b_1 is the number of observed raw reads supporting any allele, b_2 is the number of observed raw reads supporting the majority allele, and t is the base quality threshold used for filtering. Raw

reads refers to non-deduplicated reads. As a side note, b_1 is also the number of expected raw reads supporting any allele, and $b_1 \times 10^{-t/10}$ is also the number of expected raw reads supporting the majority allele.

Equation (34) is similar to the notion of UMI efficiency proposed by Xu et al.[45]. Equation (34) is a non-linear formula requiring no additional dataset. However, UMI efficiency is estimated to be a weighted linear combination of features such that the weights must be learned from other training datasets. Unfortunately, the generation of training datasets requires additional NGS assays, and it is almost always impossible to obtain the ground truth variants for any training dataset.

PCR duplicates can be detected by mapping coordinates and/or UMI sequences. Reads that are detected as PCR duplicates of each other can be grouped into one family. Such family is referred to as a single-strand consensus sequence (SSCS). Then, the consensus base at each position should have a higher quality than any individual base quality. Naively using Eq. (34) to combine base qualities can result in unreasonably large combined qualities. Hence, the consensus base quality in a family of PCR duplicates is capped at a value around 50.

Different nucleotide-substitution and InDel errors are known to occur at different frequencies. For example, transitions occur more frequently than transversions[36]. Therefore, by default, SSCS base quality is maxed out at 44 for the C:G>T:A transition, 48 for the T:A>C:G transition, 52 for all transversions, and 60 for all InDels. The default values of these caps can be further optimized by a machine-learning algorithm. However, these default values work well in practice.

4.5.6 From deduplicated reads to single-strand variant quality (SSVQ)

Equation (34) computes the quality of each base in each deduplicated read. Then, on each strand, these qualities are combined to give a single-strand variant quality (SSVQ) as follows.

$$\mathrm{SSVQ}(b_1, B_2, e, T) = \max_{t \in T} \left(\mathrm{CQual}\left(b_1, \frac{b_1}{10^{t/10}}, \min\left(|\{b_2 \in B_2 | b_2 \geq t\}|, \frac{|B_2|}{e} \right), r \in \{1.25, 1.5\} \right) \right) \tag{35}$$

where e is the evidence-reduction odds ratio EROR computed by Eq. (20), b_1 is the number of observed deduplicated reads (i.e., read families) supporting any allele, B_2 is the set of the qualities of all observed deduplicated reads supporting the variant allele on one strand, and T is the set of base-quality thresholds. As a side note, b_1 is also the number of expected deduplicated reads supporting any allele, and $b_1 \times 10^{-t/10}$ is also the number of expected deduplicated reads supporting the variant allele, and each quality in B_2 was computed by Eq. (34).

4.5.7 From single-strand variant quality (SSVQ) to double-strand variant quality (DSVQ)

Li and Ruan[23] assumed that errors arising from the two strands of the original DNA fragment are completely independent. However, such assumption may not hold if basecall quality (BQ, also known as sequencing quality) is very high or a consensus base is obtained by grouping reads having the same UMI. Schmitt et al.[36]; Kennedy et al.[18] showed that the error rate of a duplex consensus sequence (DCS) is approximately 10^{-6} and that the error rate of a single-strand consensus sequence (SSCS) is approximately 10^{-5}. Thus, we let the error rate of DCS to be 10^{-6} which is equivalent to 60 in PHRED scale. The duplex error affects the two strands of the original DNA fragment. Therefore, for each strand of interest, we incorporated the evidence from the other strand to compute the strand-aware variant quality of the strand of interest as follows.

$$v_1' = v_1 + v_2 \times \min\left(1, \frac{\min(w_1 + w_2, w_0) - w_1}{w_1}\right) \tag{36}$$

where v_1' is the strand-aware variant quality of the strand of interest, v_1 and v_2 are the strand-unaware variant qualities of the strand of interest and of the other strand, w_1 and w_2 are the thresholds

of raw base qualities or UMI-consensus base qualities of the strand of interest and of the other strand computed by Eq. (33), and w_0 is the Phred-scaled error rate of DCS. The value $\max(v_1', v_2')$ is considered to be the strand-combined variant quality, where the formula for v_2' can be derived from Eq. (36) by symmetry. In Eq. (36), if $w_1 + w_2 \leq w_0$, then $\max(v_1', v_2') = v_1 + v_2$, which corresponds to the assumption mentioned by Li and Ruan [23]: the assumption that errors coming from the two strands of the original DNA are independent.

4.5.8 Adjustment of double-strand variant quality (DSVQ) by duplex families

Intuitively, duplex families provide additional information about the truthfulness of the variant, and this additional information can be used to adjust variant quality. First, let us make the following three presumptions.

1. The expected value of a DSVQ after adjustment should be identical to such value before adjustment. Otherwise, the meaning of DSVQ depends on whether duplex sequencing is used.

2. The error rate of single-strand consensus sequence (SSCS), which corresponds to SSVQ, is about 10 times higher than that of duplex consensus sequence (DCS), which corresponds to DSVQ [36;18].

3. The allele fraction of a typical false positive variant is much less than 10%, so the probability of two SSCSs erroneously supporting the same allele by chance is negligible.

These three presumptions lead to the following rule: if the two SSCSs that form the DCS both support the same allele, then DSVQ is increased by 10; otherwise, DSVQ is decreased by 1. If duplex sequencing is not used, then no adjustment by duplex families is made.

4.5.9 Adjustment of double-strand variant quality (DSVQ) by power law and other qualities

Equation (7) described the power-law relationship between allele fraction and false positive rate at infinite depth of coverage for the allele of interest. However, coverage depth is always finite. Thus, we simply capped DSVQ with Eq. (7).

A common step in the analysis of NGS data is to filter out variants supported by bases of low base-call quality (BQ). Unfortunately, every hard filter results in the loss of the ability to make sensitivity-specificity trade off because every hard filter generates a yes-or-no answer. Therefore, instead of using any hard filter, we imposed a maximum variant quality if the BQ of the bases supporting the variant is overall too low. More specifically, when the root-mean-square (RMS) BQ of the variant-supporting bases drops below a threshold (default value of 25 for amplicon-based assay and 20 for capture-based assay), the minimum of the original DSVQ and the RMS-BQ is considered to be the final DSVQ. In addition, the final DSVQ is capped at $\frac{1}{12} \times \text{RMS-BQ}^2$.

The maximum mapping quality of the reads aligned with BWA MEM is 60 [26]. We assumed that the maximum mapping error rate is one read per one million reads and each mapping error results in one false positive variant per 100 mapped bases. Then, the mapping quality (MQ) of 60 corresponds to a variant-call quality of 80. Thus, UVC capped the final DSVQ at $\frac{4}{3} \times \text{RMS-MQ}$ so that the MQ-induced DSVQ cap is 80 for each variant supported by reads with a RMS-MQ of 60.

Reads that are considered as duplicates of each other may have slightly different MQs. We did not find a theoretically sound way to combine the MQs of duplicated reads. However, for the purpose of normalizing variant quality with MQ, the use of deduplicated reads makes a significant difference only if very few template molecules with outlier MQs are over-amplified. In addition, we observed that the signal from a true variant is usually amplified more efficiently than the background noise in NGS (data not shown). Therefore, we only used the MQs of raw non-deduplicated reads.

4.6 Somatic variant call at finite depth of coverage - comparison between the tumor and its matched normal samples

First, let us define the following terms.

1. a_1 is the summed consensus family quality of all alleles in the normal sample,
2. a_2 is the summed consensus family quality of the variant allele in the normal sample,
3. b_1 is the summed consensus family quality of all alleles in the tumor sample, and
4. b_2 is the summed consensus family quality of the variant allele in the tumor sample.
5. $\mathrm{DSVQ}^{\mathrm{tumor}}$ and $\mathrm{DSVQ}^{\mathrm{normal}}$ denote the normalized double-strand variant qualities of the tumor and its matched normal computed in Section 4.5.9, respectively

Usually, at a given site, a false positive noise present in a tumor sample is also present in its matched normal sample. The false positive noise can be generated by mapping error, library preparation error, oxidative damage, etc. Unlike germline polymorphism, the range of expected variant allele fractions (VAFs) for somatic variants is not known in advance. The observed variants in the tumor and normal samples are considered to be signal and noise, respectively. The comparison between the tumor and its matched normal is done in two parts: the TNreward part rewards high variant evidence in the tumor relative to its matched normal, and the TNpenal part penalizes low variant evidence in the tumor with respect to its matched normal.

4.6.1 Reward for high signal intensity in the tumor relative to its matched normal

The formula for $\mathrm{TNreward}_{\mathrm{Binomial}}$ is as follows.

$$\mathrm{TNreward}_{\mathrm{Binomial}}(a_1, a_2, b_1, b_2) = \begin{cases} 10/\log(10) \times \left(a_1 \times \mathrm{KL}_{\mathrm{Bernoulli}}\left(\frac{b_2}{b_1}, \frac{a_2}{a_1}\right)\right) & \text{if } \frac{b_2}{b_1} > \frac{a_2}{a_1} \\ 0 \text{ otherwise} \end{cases} \tag{37}$$

At infinite depth of coverage, Eq. (11) alone is sufficient to compute TNreward. At finite depth of coverage, there are two qualities computed by Eq. (37) and Eq. (11). Then, we used the principle of maximum entropy to select one of these two qualities, so the quality which is closer to zero is selected.

The principle of maximum entropy implies that, when selecting two models with the same likelihood, the model which assigns more uniform probability to each category should be selected. Therefore, the model that assigns closer to 50% probability to each binary outcome should be selected if multiple such models of equal likelihood are given. In the logarithmic space of likelihood, it means that the model with log odds-ratios of outcomes closer to zero should be selected. More formally, for two likelihoods a and b respectively produced by two models, the final likelihood $\max H$ after model selection is defined as follows.

$$\max H(a, b) = \begin{cases} a & \text{if } \max(a, -a) < \max(b, -b) \\ b & \text{otherwise} \end{cases} \tag{38}$$

4.6.2 Penalty for low signal intensity in the tumor relative to its matched normal

The Phred-scaled likelihood TNpenal of systematic error is computed as follows.

$$\mathrm{TNpenal} = \mathrm{ReLU}\left(\mathrm{DSVQ}^{\mathrm{normal}} - \rho \times (\mathrm{ReLU}(\mathrm{OddsRatio}(a_1, a_2, b_1, b_2) - 1))^{-2}\right) \tag{39}$$

meaning that the random noise which remains after normalization by systematic error is fit to a Gaussian noise model. Higher ρ results in higher decrease in likelihood if the tumor allele fraction deviates from the normal allele fraction. By default, $\rho = 12.5$, and this default value is approximately the Phred-scaled equivalent of 3/67, the fraction of mutations with a MAF (%CV) value of more than 50% in Table 7 of MSK-IMPACT [31].

4.6.3 Combining evidence for the absence of germline event and the presence of somatic event

The final somatic variant quality is computed to be the minimum of (NLOD + 31) and TLOD. NLOD was computed by Eq. (18) in Section 4.3, the number 31 is the Phred-scaled probability that a base position is heterozygous in the genome of a human individual (two human genomes typically differ at 4 to 5 million loci out of 3 billion base pairs[7], and 55% to 65% of loci are heterozygous[34]), and TLOD is defined as follows.

$$\mathrm{TLOD} = \mathrm{DSVQ}^{\mathrm{tumor}} + \mathrm{TNreward} - \mathrm{TNpenal} \tag{40}$$

4.7 Extension of variant-calling methods from SNVs to InDels

So far, our method seems to be only applicable to single nucleotide variants (SNVs). It is not clear how InDels can be handled. In fact, InDels can be treated in the same way as SNVs.

First, each genomic position is split into two sub-positions. One sub-position has the statistics of each nucleotide class among {A,C,G,T}, and the other sub-position has the statistics of each class of InDel gap. InDel gaps are grouped into the following seven classes by the short tandem repeat (STR) pattern of the inserted or deleted sequence.

1. neither insertion nor deletion
2. insertion of one STR unit
3. insertion of two STR units
4. insertion of three or more STR units, or insertion of a non-STR sequence
5. deletion of one STR unit
6. deletion of two STR units
7. deletion of three or more STR units, or deletion of a non-STR sequence

The occurrence of each InDel observed in data is assigned to a class mentioned above. Then, each class of InDel is simply treated as a class of SNV.

It is possible that different inserted sequences fall into the same class. For example, if ATG and AGT are both inserted at the same genomic position that is not part of a STR, then the class "insertion of a non-STR sequence" includes these two sequences. In this case, the two inserted sequences are shown as two alleles in the FORMAT field of the result VCF file. However, for each class of InDel, only the most frequently observed InDel sequence is shown in the ALT field of the result VCF file.

A typical sequencing platform generates quality scores for mismatches but not for InDels. In addition, unlike mismatches, the number of possible InDels in a read is theoretically infinite because any sequence can be inserted at any position. UVC estimates the basecall-like quality of an InDel in a read with the following formula.

$$\mathrm{InDelQual}(x, y, a, b, c) = \min\left(x + 10, y + 10, \mathrm{STRQual}(a, b)\right) + 10 \times \log_{10}(c) \tag{41}$$

where

$$\mathrm{STRQual}(a, b) = 44 - 10 \times \log_{10}\left(1 + \frac{8 \times \mathrm{softplus}\left(a \times b - 8\right)}{\theta \times a^2}\right) \tag{42}$$

$$\mathrm{softplus}(v) = \log(1 + \exp(v)) \approx \mathrm{ReLU}(v) \tag{43}$$

and where x and y are respectively the two qualities of the two bases that are immediately before and after the InDel, a is the length of each repeating unit in the STR region, b is the number of repeating

units in the STR region, and c is the number of repeating units that are inserted or deleted. If the InDel is not in a STR region, then $a = b = 1$.

Equation (42) is formulated according to the profile of PCR errors estimated by Shinde et al.[38]. In Eq. (42), the number 8 is approximately the number of bases in the reactive site of a typical PCR polymerase such as *Taq*[38], In Eq. (42), the number 44 is approximately the InDel error rate of Illumina sequencing platforms[21].

The value of θ in Eq. (42) is also assigned according to the statistics provided by Shinde et al.[38]. If the InDel is the deletion of one STR unit, then $\theta = \frac{1}{4}$. Otherwise, $\theta = 1$.

In Eqs. (5) and (14), AlignBias is used for normalizing allele fractions before computing genotype likelihoods for InDels. Nonetheless, Eqs. (5) and (14) did not provide the formula for AlignBias. For an InDel of L bases with an STR context of length S (e.g., $S = 2 \times 3$ for the STR with reference sequence `CACACA`), AlignBias is defined as follows.

$$\text{AlignBias}(L, S) = \frac{150}{\max(30, 150 - 3 \times L + S + 6 \times 2 + 5 \times 2)} \tag{44}$$

where 6, 5, and 30 are the default gap-opening penalty, soft-clip penalty, and output alignment score for BWA MEM[26], and 150 is the number of cycles for a typical sequencing run. These parameters can be adjusted depending on the sequencing platform and alignment (including realignment) options that were used to generate the BAM file. Still, these parameters should work well in practice in most situations with their default values.

We found that using an InDel error rate that is not aware of any base quality (BQ) results in slightly worse sensitivity-specificity trade off (data not shown). Hence, we incorporated x and y in Eq. (41) so that Eq. (41) considers the BQs of the bases adjacent to the InDel in the computation of the basecall-like quality of the InDel.

The treatment of the twice-sequenced InDel class in the overlap between the R1 and R2 ends is similar to that of SNV in Eq. (31). However, for InDels, $\text{R1R2overlapQ}(q_1, q_2) = \max(q_1, q_2)$ regardless of whether $b_1 = b_2$. The reason is that the read end with the lower quality of InDel class is often mis-aligned, and mis-alignment effectively results in a quality of zero for its associated InDel class.

For InDels, true positive and false positive signals tend to co-occur at the same locus[2]. Thus, for a true positive InDel variant, a significant proportion of its read support can be generated by sequencing artifacts, so the matched normal may be characterized by some variant signal. Therefore, to improve sensitivity, TNpenal is set to zero in Eq. (40) for InDels.

4.8 Evaluation detail

Unless stated otherwise, we ran each variant caller with its default command-line parameters, and we used the vcfeval utility from the RTG tools to perform our evaluation (`https://github.com/RealTimeGenomics/rtg-tools`). The vcfeval utility considers representational difference in the VCF format: if the application of two or more variants on the same genome would result in the same remaining sequence, then these variants are considered to be equivalent even if these variants are represented in different forms. The vcfeval utility from the RTG tools is unable to handle any tri-allelic site properly. Therefore, we wrote a custom script to generate Table 1 instead of using vcfeval. If the ground-truth VCF for assessing variant calls is not provided, then we used the command `bcftools isec -c both` to create such ground-truth VCF, so SNPs that are at the same position are considered to be identical, and InDels that are at the same position are considered to be identical.

The reference genome hg19 was used for evaluating UVC with Mageri and smCounter2 because hg19 was previously used by Mageri and smCounter2 in their own studies. The reference genome GRCh37 was used for evaluating UVC with Mutect2 and Strelka2 because GRCh37 was also used by the Genome in a Bottle (GIAB) consortium.

In our evaluation, we tried to run the latest release of each software, with an allocation of 32 logical CPU cores, on a Red Hat 4.8.5-28 server with 128 GB of RAM and 28/56 physical/logical CPU cores

(Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz), except that Mutect2 which is part of GATK4 was run with 12 processes and 4 threads per process for a total of 48 logical CPU cores. First, BWA MEM version 0.7.17 was used for mapping reads in the raw FASTQ files to the reference genome. Then, we attempted to run the latest release of each variant caller with its default parameters. More specifically, the variant callers UVC version 0.0.28.070c2b7, Mageri version 2.0, smCounter2 version 2 (commit c475863b), Strelka2 version 2.9.10, and GATK version 4.1.3.0 were used for evaluating their performance. Due to its limited multithreading capability, Mutect2 was manually parallelized with GNU parallel[42] to run on each chromosome such that the chromosome-specific variant calls were then combined to generate the final variant calls. Finally, RTG tools version 3.10.1 was used for evaluating the sensitivity-specificity trade off of each variant caller. The sensitivity-specificity curve was plotted by varying the following VCF tags: QUAL, SomaticEVS, QSI_NT, QSS_NT, and TLOD. If QUAL is not available or not applicable, then we attempted to use SomaticEVS; if SomaticEVS is not available or not applicable, then we attempted to use QSI_NT, etc.

For evaluating UVC and Mageri, the HD734 variants of interest were retrieved from `https://github.com/mikessh/mageri-paper` commit d064c9b. For evaluating UVC and smCounter2, the targeted genomic regions were retrieved from `https://github.com/qiaseq/smcounter-v2-paper` commit 69c4e20. For evaluating UVC, Mutect2, and Strelka2, additional data described in the file at `ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/use_cases/mixtures/UMCUTRECHT_NA12878_NA24385_mixture_10052016/README-NA12878_NA24385_mixture.txt` were used. Evaluation procedures for UVC, smCounter2, MuTect2, and Strelka2 (excluding Mageri) all followed the best practice at `ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/NA12878_HG001/NISTv3.3.2/README_NISTv3.3.2.txt`.

For GATK, deduplication and base recalibration were performed, which is recommended by the best practice for using GATK. UVC and Strelka2 used the raw BAM files (which were directly generated by BWA MEM or NovoAlign) as their inputs.

For the *in silico* mixtures, we directly used the BAM files pre-aligned by NovoAlign version 3.02.07. More detail about the pre-aligned BAM files is at `https://ftp.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/README_NHGRI_Novoalign_bams` and `https://ftp.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NIST_NA12878_HG001_HiSeq_300x/NHGRI_Illumina300X_novoalign_bams/README`. More specifically, reads in the alignment files of NA24385 (`https://ftp.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/HG002.GRCh38.300x.bam`) and NA12878 (`https://ftp.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NIST_NA12878_HG001_HiSeq_300x/NHGRI_Illumina300X_novoalign_bams/HG001.hs37d5.300x.bam`) were assumed to be originated from the tumor and normal tissues, respectively. Then, a carefully designed *in silico* mixture of the tumor and normal reads was created for each autosome from 1 to 21 (tumor reads were subsampled from the original NA24385 BAM file with the command `samtools view -s 0.`x ($x \in \{0.01, 0.03, 0.07, 0.13, 0.21, 0.31, 0.43\}$) using samtools version 1.6; normal reads were directly taken from the original NA12878 BAM file; finally, for each read, if the X31 hashes modulo 100 of its read name is at least/less than y ($y \in \{0, 7, 14, 21\}$ for tumor reads and $y = 50$ for normal reads), then the read belongs to the virtual tumor/normal BAM file, respectively). Before running UVC, we ran miniVC on chromosomes 7/14/21 to estimate the tumor-in-normal contamination (denoted as η) for chromosomes 1-7/8-14/15-21, respectively. Then, $1.8 \times \eta$ rounded down with two decimal places of accuracy was used as input to UVC.

For the re-analysis of the SiMSen-Seq dataset SRP158874, the QUAL threshold was set to 53, and the parameter ρ in Eq. (39) was set to zero. We let $\rho = 0$ because the project SRP158874 used the same sample without any exposure to UV instead of a different sample (such as a tumor-matched normal) as the negative control, and control with the same sample is supposed to be subject to less variability than control with a different sample. Initially, UVC only discovered the *DPH3* -8 variant on the two samples from patients with Cockayne Syndrome, because the *DPH3* -8 variant is characterized by strong signal in these two samples. Then, we manually discovered that the *DPH3* -8 variant is

characterized by some weak signal in the two samples from patients with Xeroderma Pigmentosum.

4.9 Statistics

Hypothesis testing can be performed with the results shown in Table 2. More specifically, we have the following two hypotheses.

1. Null hypothesis: the best F-score/AUC-PR of SNVs/InDels is selected by random chance among UVC, Mutect2, and Strelka2 for each scenario.

2. Alternative hypothesis: the best F-score/AUC-PR of SNVs/InDels is more likely to be the one from UVC for each sceneario.

For the F-score of InDels, UVC outperformed Mutect2 and Strelka2 in 19 scenarios out of 21. For the other cases (i.e., F-score of SNVs, AUC-PR of SNVs, AUC-PR of InDels), UVC outperformed Mutect2 and Strelka2 in all 21 scenarios out of 21. Thus, the P values of the binomial test of our hypothesis are respectively 8.441×10^{-8} and 9.56×10^{-11} calculated by the commands `binom.test(19, 21, 1/3, alternative='greater')` and `binom.test(21, 21, 1/3, alternative='greater')` in R version 3.6.3.

5 Acknowledgements

We would like to thank Jie Yang for providing a lot of useful feedback about the usage of UVC in clinical settings, Shaobo Li for using UVC the first time, Jinhuan Liu for reviewing some IGV screenshots, Da Tan for some discussion about variant calling in general, Qianqian Song for some discussion about the characteristics of clinical data, Guangyu Li for some discussion about other projects related to variant calling, Xiangyu Zhou for running some other variant callers for some other projects, Linlin Wei for finding one variant that is called by UVC and is likely to be false positive in clinical settings, Lin Zhou for some general discussion about the best practice in NGS, and Zhaohui Chen for providing a lot of user experience about the post-processing of the VCF files generated by UVC.

6 Author Contributions

Xiaofei Zhao designed the UVC variant-calling model, developed the UVC algorithm, evaluated the performance of each algorithm, and wrote the manuscript. Allison Hu verified the scientific results in this work. Sizhen Wang supervised and supported part of the study. Xiaoyue Wang wrote the manuscript, reviewed the manuscript, and supervised the study.

7 Competing Interests

The algorithms presented in this manuscript are patent-pending. S.W. is one of the founders of Genetron Health. X.W. is a scientific advisor for Genetron Health. The remaining authors declare no competing interests.

8 Additional Information

Supplementary Information is available for this paper.

Correspondence and requests for materials should be addressed to Xiaofei Zhao and Xiaoyue Wang.

Reprints and permissions information is available at www.nature.com/reprints.

9 Data availability

The authors declare that the data supporting the findings of this study are available within the paper and its supplementary information files.

10 Code availability

The source code for UVC is publicly available at `https://github.com/genetronhealth/uvc`.

References

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

[2] Erica K Barnell, Peter Ronning, Katie M Campbell, Kilannin Krysiak, Benjamin J Ainscough, Lana M Sheta, Shahil P Pema, Alina D Schmidt, Megan Richters, Kelsy C Cotto, et al. Standard operating procedure for somatic variant refinement of sequencing data with paired tumor and normal samples. *Genetics in Medicine*, 21(4):972, 2019.

[3] Baruch Barzel and Albert-László Barabási. Universality in network dynamics. *Nature physics*, 9 (10):673, 2013.

[4] Guoli Chen, Stacy Mosier, Christopher D Gocke, Ming-Tseh Lin, and James R Eshleman. Cytosine deamination is a major cause of baseline noise in next-generation sequencing. *Molecular diagnosis & therapy*, 18(5):587–593, 2014.

[5] Kristian Cibulskis, Aaron McKenna, Tim Fennell, Eric Banks, Mark DePristo, and Gad Getz. Contest: estimating cross-contamination of human samples in next-generation sequencing data. *Bioinformatics*, 27(18):2601–2602, 2011.

[6] Kristian Cibulskis, Michael S Lawrence, Scott L Carter, Andrey Sivachenko, David Jaffe, Carrie Sougnez, Stacey Gabriel, Matthew Meyerson, Eric S Lander, and Gad Getz. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature biotechnology*, 31(3):213, 2013.

[7] 1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

[8] Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A Albers, Eric Banks, Mark A DePristo, Robert E Handsaker, Gerton Lunter, Gabor T Marth, Stephen T Sherry, et al. The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158, 2011.

[9] Evgeniya Denisova, Barbara Heidenreich, Eduardo Nagore, P Sivaramakrishna Rachakonda, Ismail Hosen, Ivana Akrap, Víctor Traves, Zaida García-Casado, José Antonio López-Guerrero, Celia Requena, et al. Frequent DPH3 promoter mutations in skin cancers. *Oncotarget*, 6(34): 35922, 2015.

[10] Hongdo Do and Alexander Dobrovic. Sequence artifacts in DNA from formalin-fixed tissues: causes and strategies for minimization. *Clinical chemistry*, 61(1):64–71, 2015.

[11] Robert C. Edgar and Henrik Flyvbjerg. Error filtering, pair assembly and error correction for next-generation sequencing reads. *Bioinformatics*, 31(21):3476–3482, 07 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv401. URL `https://doi.org/10.1093/bioinformatics/btv401`.

[12] Kerryn Elliott, Martin Boström, Stefan Filges, Markus Lindberg, Jimmy Van den Eynden, Anders Ståhlberg, Anders R Clausen, and Erik Larsson. Elevated pyrimidine dimer formation at distinct genomic bases underlies promoter mutation hotspots in UV-exposed cancers. *PLoS genetics*, 14 (12):e1007849, 2018.

[13] Brent Ewing and Phil Green. Base-calling of automated sequencer traces using phred. ii. error probabilities. *Genome research*, 8(3):186–194, 1998.

[14] Nils Johan Fredriksson, Kerryn Elliott, Stefan Filges, Jimmy Van den Eynden, Anders Ståhlberg, and Erik Larsson. Recurrent promoter mutations in melanoma are defined by an extended context-specific mutational signature. *PLoS genetics*, 13(5):e1006773, 2017.

[15] Erik Garrison and Gabor Marth. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, 2012.

[16] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 347–352. IEEE, 1996.

[17] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

[18] S. R. Kennedy, M. W. Schmitt, E. J. Fox, B. F. Kohrn, J. J. Salk, E. H. Ahn, M. J. Prindle, K. J. Kuong, J. C. Shen, and R. A. Risques. Detecting ultralow-frequency mutations by duplex sequencing. *Nature Protocols*, 9(12):2586, 2014.

[19] Isaac Kinde, Jian Wu, Nick Papadopoulos, Kenneth W. Kinzler, and Bert Vogelstein. Detection and quantification of rare mutations with massively parallel sequencing. *Proceedings of the National Academy of Sciences*, 108(23):9530–9535, 2011. ISSN 0027-8424. doi: 10.1073/pnas.1105422108. URL `https://www.pnas.org/content/108/23/9530`.

[20] Daniel C Koboldt, Qunyuan Zhang, David E Larson, Dong Shen, Michael D McLellan, Ling Lin, Christopher A Miller, Elaine R Mardis, Li Ding, and Richard K Wilson. Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research*, 22(3):568–576, 2012.

[21] David Laehnemann, Arndt Borkhardt, and Alice Carolyn McHardy. Denoising dna deep sequencing data—high-throughput sequencing errors and their correction. *Briefings in bioinformatics*, 17(1):154–179, 2015.

[22] Y le Cun. Generalization and network design strategies. Citeseer, 1989.

[23] H Li and R Ruan, Jdurbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research*, 18(11):1851–1858, 2008.

[24] Heng Li. Improving SNP discovery by base alignment quality. *Bioinformatics*, 27(8):1157–1158, 2011.

[25] Heng Li. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21):2987–2993, 2011.

[26] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*, 2013.

[27] Heng Li. Toward better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics*, 30(20):2843–2851, 2014.

[28] Eckhard Limpert, Werner A Stahel, and Markus Abbt. Log-normal distributions across the sciences: keys and clues: on the charms of statistics, and how mechanical models resembling gambling machines offer a link to a handy way to characterize log-normal distributions, which can provide deeper insight into variability and probability—normal or log-normal: that is the question. *BioScience*, 51(5):341–352, 2001.

[29] Tanja Magoč and Steven L Salzberg. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, 27(21):2957–2963, 2011.

[30] Andre P Masella, Andrea K Bartram, Jakub M Truszkowski, Daniel G Brown, and Josh D Neufeld. PANDAseq: paired-end assembler for illumina sequences. *BMC bioinformatics*, 13(1): 31, 2012.

[31] MSK-IMPACT. EVALUATION OF AUTOMATIC CLASS III DESIGNATION FOR MSK-IMPACT (Integrated Mutation Profiling of Actionable Cancer Targets). URL `https://www.accessdata.fda.gov/cdrh_docs/reviews/DEN170058.pdf`.

[32] Francesc Muyas, Mattia Bosio, Anna Puig, Hana Susak, Laura Domènech, Georgia Escaramis, Luis Zapata, German Demidov, Xavier Estivill, Raquel Rabionet, et al. Allele balance bias identifies systematic genotyping errors and false disease associations. *Human Mutation*, 40(1): 115–126, 2019.

[33] Sayed Mohammad Ebrahim Sahraeian, Ruolin Liu, Bayo Lau, Karl Podesta, Marghoob Mohiyuddin, and Hugo YK Lam. Deep convolutional neural networks for accurate somatic mutation detection. *Nature communications*, 10(1):1041, 2019.

[34] David C Samuels, Jing Wang, Fei Ye, Jing He, Rebecca T Levinson, Quanhu Sheng, Shilin Zhao, John A Capra, Yu Shyr, Wei Zheng, et al. Heterozygosity ratio, a robust global genomic measure of autozygosity and its association with height and disease risk. *Genetics*, 204(3):893–904, 2016.

[35] Kim Sangtae, Scheffler Konrad, Halpern Aaron L., Bekritsky Mitchell A., Noh Eunho, Morten Källberg, Chen Xiaoyu, Kim Yeonbin, Beyter Doruk, and Krusche Peter. Strelka2: fast and accurate calling of germline and somatic variants. *Nature Methods*, 2018.

[36] Michael W. Schmitt, Scott R. Kennedy, Jesse J. Salk, Edward J. Fox, Joseph B. Hiatt, and Lawrence A. Loeb. Detection of ultra-rare mutations by next-generation sequencing. *Proceedings of the National Academy of Sciences of the United States of America*, 109(36):14508–14513, 2012.

[37] Johnny Sena. Unique Molecular Identifiers reveal a novel sequencing artefact with implications for RNA-Seq based gene expression analysis. *Scientific Reports*, 8, 09 2018. doi: 10.1038/s41598-018-31064-7.

[38] Deepali Shinde, Yinglei Lai, Fengzhu Sun, and Norman Arnheim. Taq DNA polymerase slippage mutation rates measured by PCR and quasi-likelihood analysis:(CA/GT) n and (A/T) n microsatellites. *Nucleic acids research*, 31(3):974–980, 2003.

[39] Mikhail Shugay, Andrew R. Zaretsky, Dmitriy A. Shagin, Irina A. Shagina, Ivan A. Volchenkov, Andrew A. Shelenkov, Mikhail Y. Lebedin, Dmitriy V. Bagaev, Sergey Lukyanov, and Dmitriy M. Chudakov. Mageri: Computational pipeline for molecular-barcoded targeted resequencing. *PLOS Computational Biology*, 13(5):1–17, 05 2017. doi: 10.1371/journal.pcbi.1005480. URL `https://doi.org/10.1371/journal.pcbi.1005480`.

[40] Mikhail Shugay, Andrew R. Zaretsky, Dmitriy A. Shagin, Irina A. Shagina, Ivan A. Volchenkov, Andrew A. Shelenkov, Mikhail Y. Lebedin, Dmitriy V. Bagaev, Sergey Lukyanov, and Dmitriy M. Chudakov. MAGERI: Computational pipeline for molecular-barcoded targeted resequencing. *Plos Computational Biology*, 13(5), 2017.

[41] Magdalena Szumilas. Explaining odds ratios. *Journal of the Canadian academy of child and adolescent psychiatry*, 19(3):227, 2010.

[42] Ole Tange. *GNU Parallel 2018*. Ole Tange, March 2018. ISBN 9781387509881. doi: 10.5281/zenodo.1146014. URL `https://doi.org/10.5281/zenodo.1146014`.

[43] John J Weng, Narendra Ahuja, and Thomas S Huang. Learning recognition and segmentation of 3-D objects from 2-D images. In *1993 (4th) International Conference on Computer Vision*, pages 121–128. IEEE, 1993.

[44] Ethan P White, Brian J Enquist, and Jessica L Green. On estimating the exponent of power-law frequency distributions. *Ecology*, 89(4):905–912, 2008.

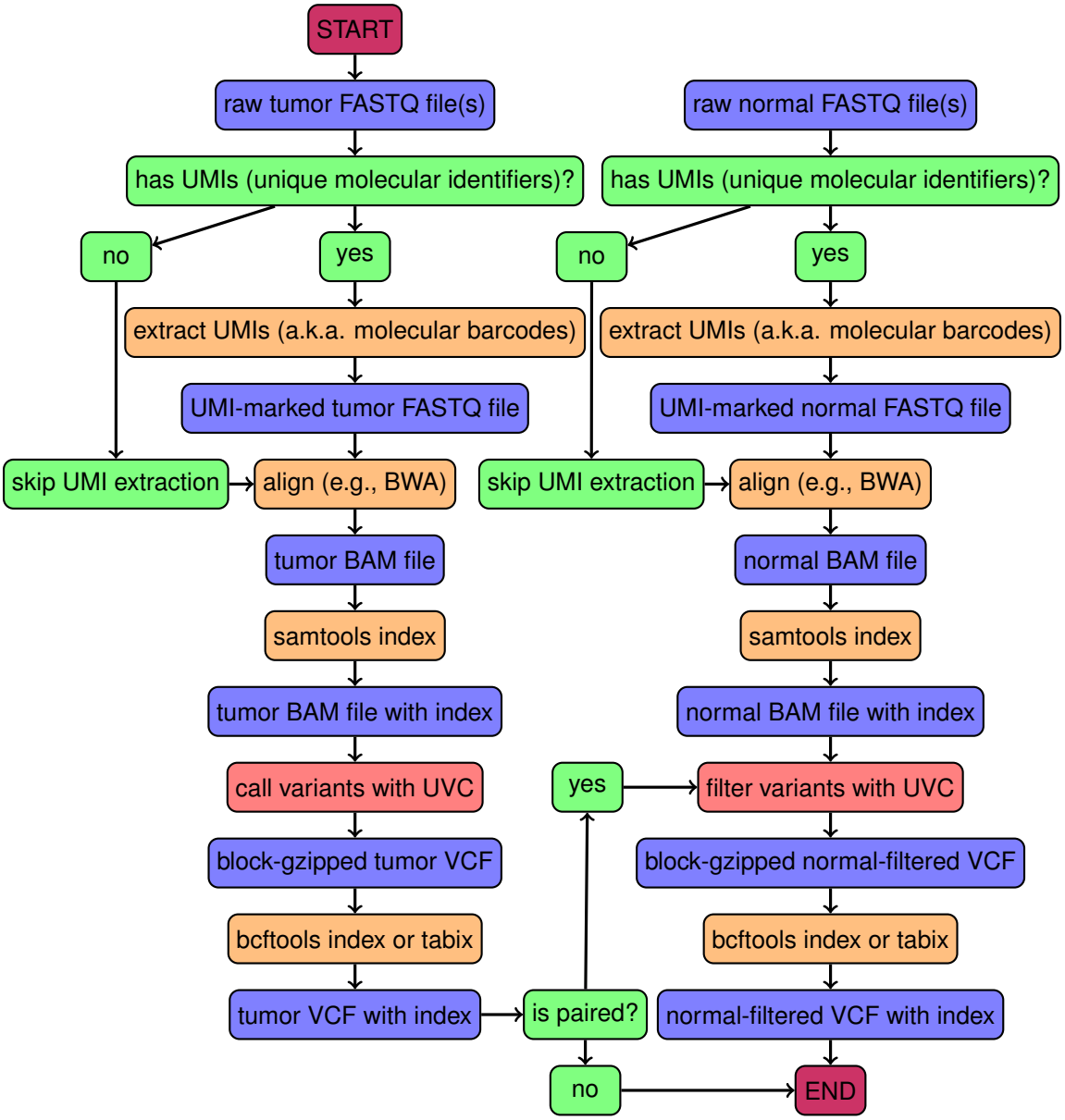
[45] Chang Xu, Xiujing Gu, Raghavendra Padmanabhan, Zhong Wu, Quan Peng, John DiCarlo, and Yexun Wang. smcounter2: an accurate low-frequency variant caller for targeted sequencing data with unique molecular identifiers. *Bioinformatics*, 35(8):1299–1309, 2019.

[46] Adam Yongxin Ye, Yanmei Dou, Xiaoxu Yang, Sheng Wang, August Yue Huang, and Liping Wei. A model for postzygotic mosaicisms quantifies the allele fraction drift, mutation rate, and contribution to de novo mutations. *Genome research*, 28(7):943–951, 2018.

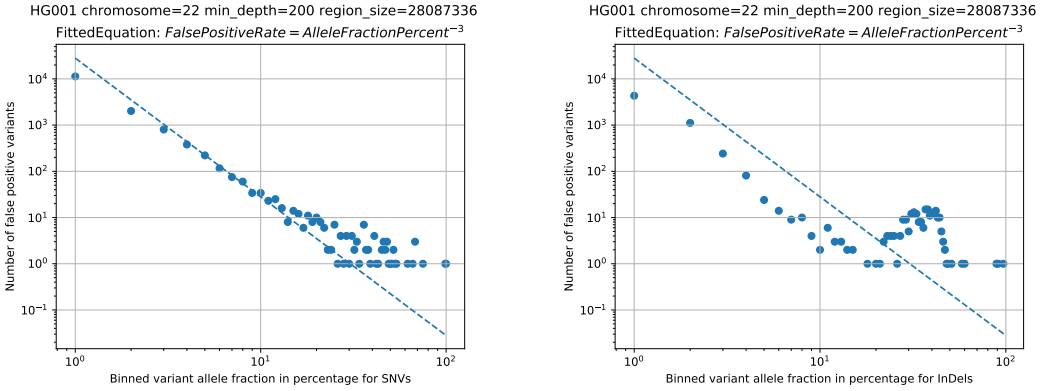
[47] Jiajie Zhang, Kassian Kobert, Tomáš Flouri, and Alexandros Stamatakis. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*, 30(5):614–620, 2013.

[48] Justin M Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E Mason, Noah Alexander, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data*, 3:160025, 2016.

10.1 Supplementary figures and tables



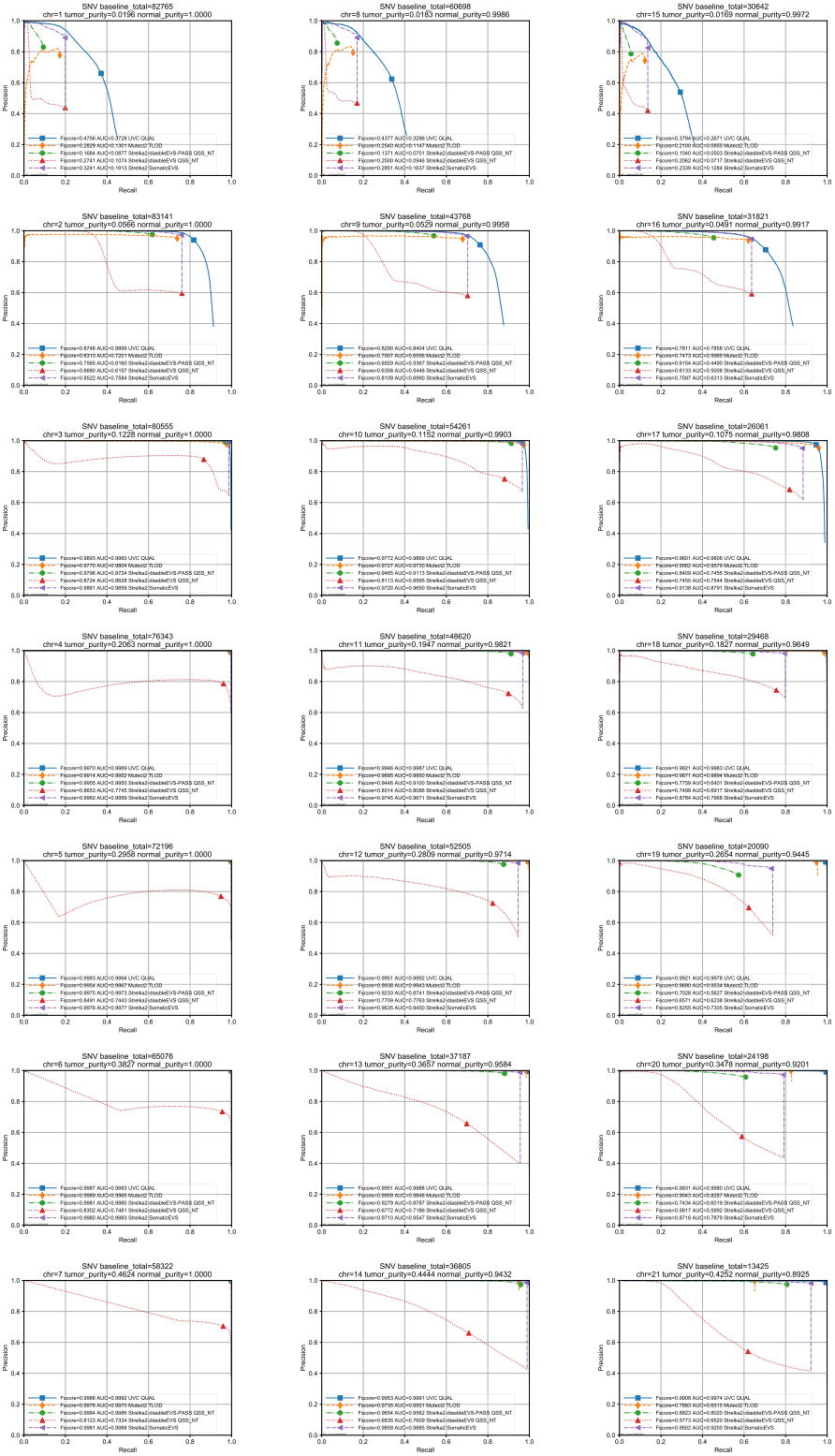
Supplementary Fig. 1: Bioinformatics pipeline for variant calling with UVC.



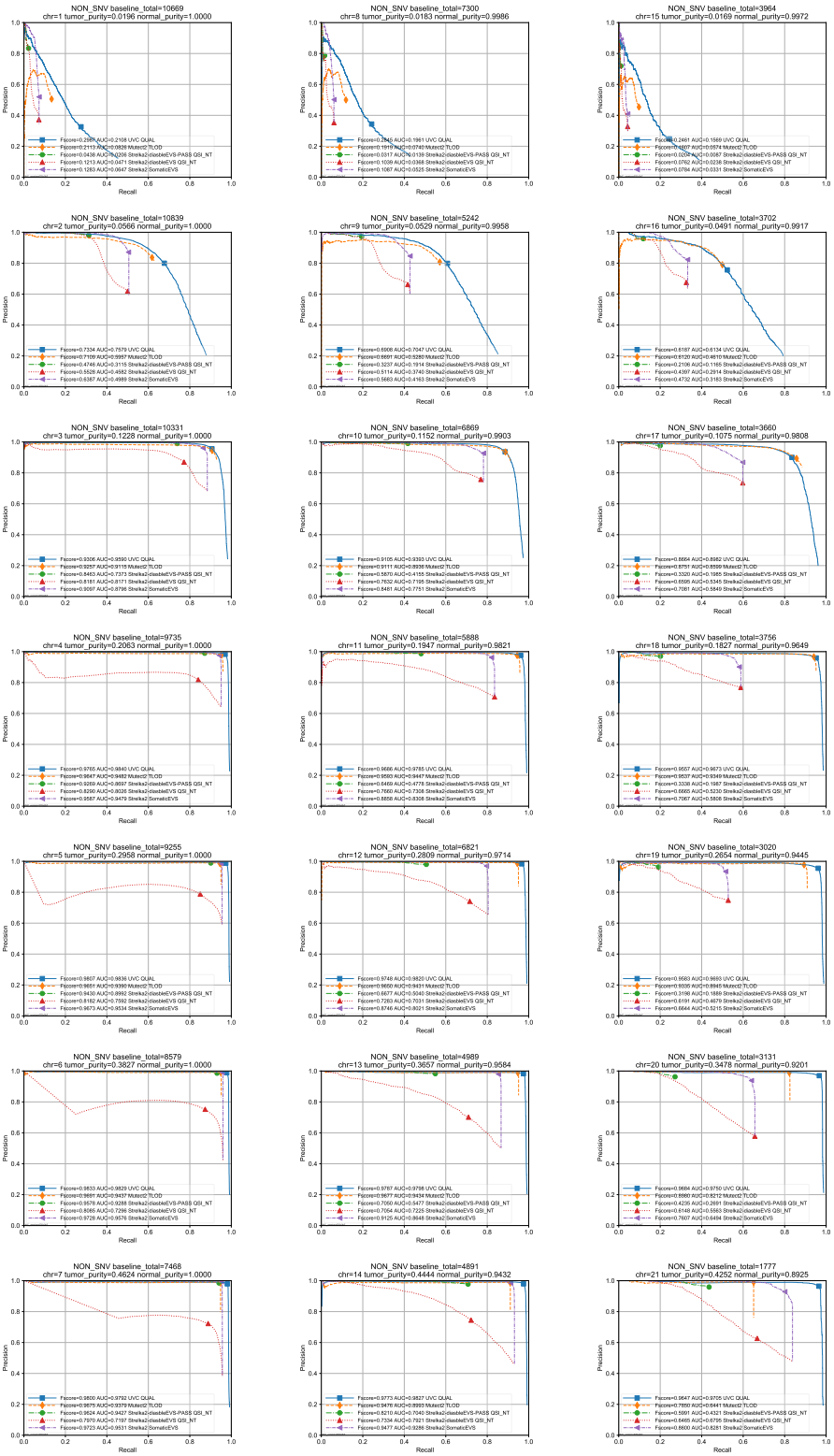
Supplementary Fig. 2: Likelihood that a variant is false positive as a function of its allele fraction after removing known germline variants for SNVs and InDels. This figure shows that the generation of NGS errors belongs to the universality class with the critical exponent of 3. For both SNVs and InDels, the observed false positive rate seems to be higher than expected at the right tail, but log-log plots are known to under-estimate the absolute value of the exponent at the right tail [44], so the power law is still supposed to hold at the right tail despite this visual artifact. InDels at the allele fraction of around 50% are characterized by unusually high false positive rates presumably because some of these false positive InDels can be real germline InDels that are not known yet (in the truth set, the number of germline InDels increased to 505169 in v3.3.1 from 358753 in v3.3 [48], so the current truth set v3.3.2 is still expected to exclude some real germline InDels).

Software	Implementation language	Runtime in second	
		wall-clock	CPU
BWA MEM	C	84 512	2 675 778
GATK4 MarkDuplicates	Java	47 187	263 039
GATK4 BaseRecalibrator	Java	78 404	85 764
GATK4 ApplyBQSR	Java	65 742	100 598
GATK4 Mutect2	Java	350 678	632 879
GATK4 Mutect2 (parallelized)	Java	36 280	752 861
Strelka2	C++	10 018	193 735
UVC	C++	36 922	820 301

Supplementary Table 1: The runtimes of different variant callers listed in Fig. 3, as well as the runtimes of the software packages that these variant callers depend on, on the Illumina-sequenced 30%/70% physical mixture of the reference materials NA12878/NA24385. For GATK4, we also ran MarkDuplicatesSpark, the multithreaded version of MarkDuplicates. However, MarkDuplicatesSpark with 32 threads is actually even slower than MarkDuplicates (data not shown). BaseCalibratorSpark and ApplyBQSRSpark, which are respectively the multithreaded versions of BaseCalibrator and ApplyBQSR, are still under beta-test and hence not assessed for their runtimes. Mutect2 has no option to run efficiently with multiple CPUs. Hence, for each chromosome, we ran Mutect2 to generate its chromosome-specific VCF output. At most 12 chromosomes were allowed to be processed by Mutect2 in parallel at the same time. Then, we combined these chromosome-specific VCF files, which were generated in parallel, into one single final VCF. UVC runs at least five times faster than Mageri, smCounter2, and Debarcer (data not shown), which are three variant callers specially designed for data with unique molecular identifiers (UMIs).



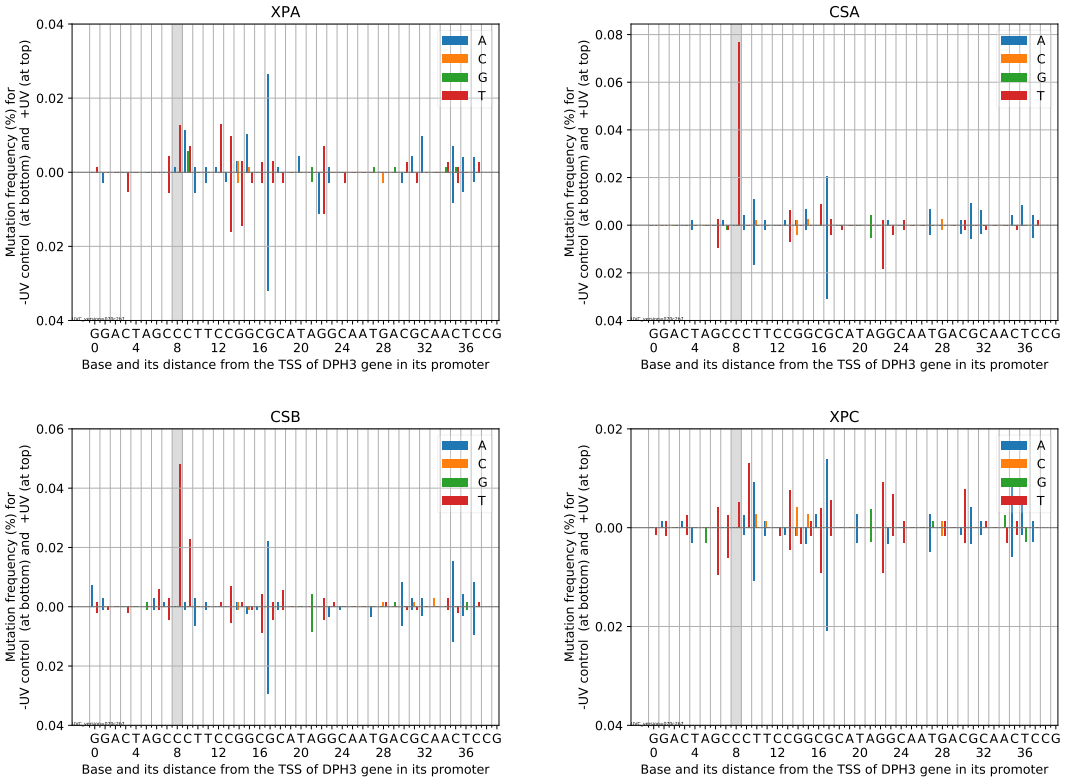
Supplementary Fig. 3: Detailed results of evaluating the SNV-calling performance on the *in silico* mixtures of NA24385/NA12878. The corresponding summary results are in Table 2.



Supplementary Fig. 4: Detailed results of evaluating the InDel-calling performance on the *in silico* mixtures of NA24385/NA12878. The corresponding summary results are in Table 2.

Sample	*RPLA* -116 bp `C>T`		*DPH3* -8 bp `C>T`	
	with UV	without UV	with UV	without UV
XPA	19 / 24134	0 / 12282	9 / 40246	0 / 19941
CSA	81 / 14584	0 / 19713	36 / 22245	0 / 27255
CSB	16 / 25370	0 / 23081	34 / 42455	0 / 52634
XPC	35 / 19723	0 / 18404	4 / 36724	0 / 32382

Supplementary Table 2: Variant allele fractions of the `C>T` substitution at the *RPLA* -116 bp and *DPH3* -8 bp `TTCCG` hotspot sites. This table complements Supplementary Fig. 5.



Supplementary Fig. 5: This figure shows that UV exposure of cultured cells induces mutations preferably at the *DPH1* -8 bp `TTCCG` hotspot site independently of nuclear-excision repair (NER), which further supports the conclusion made in Figure 2 of Fredriksson et al.[14]. In brief, four samples of human fibroblasts with NER deficiencies were subjected to a single UVB dose (20 J/m^2) during approximately 2 seconds. Then, the four samples were sequenced and the sequencing data are publicly available under the accession SRP158874. We used UVC to re-analyze the sequencing data and discovered that, in addition of rediscovering the *RPL13A* -116 bp `TTCCG` promoter hotspot mutation mentioned by Fredriksson et al.[14], UVC also discovered the new *DPH3* -8 bp `TTCCG` promoter hotspot mutation which is shown in this figure. TSS denotes transcription start site. In each UMI family, if the most frequently occurring base (referred to as the consensus base) occurs at least twice and at least four times as much as the other bases, then this consensus base is used for calculating the variant allele fractions in this figure and Supplementary Table 2, and vice versa.