








FVC as an adaptive and accurate method for filtering variants from popular NGS analysis pipelines

Yongyong Ren ^{1,2,6}, Yan Kong ^{1,2,6}, Xiaocheng Zhou ¹, Georgi Z. Genchev ^{3,4,5}, Chao Zhou ^{1,2}, Hongyu Zhao ^{4,7}✉ & Hui Lu ^{1,2,5,7}✉

The quality control of variants from whole-genome sequencing data is vital in clinical diagnosis and human genetics research. However, current filtering methods (Frequency, Hard-Filter, VQSR, GARFIELD, and VEF) were developed to be utilized on particular variant callers and have certain limitations. Especially, the number of eliminated true variants far exceeds the number of removed false variants using these methods. Here, we present an adaptive method for quality control on genetic variants from different analysis pipelines, and validate it on the variants generated from four popular variant callers (GATK HaplotypeCaller, Mutect2, Varscan2, and DeepVariant). FVC consistently exhibited the best performance. It removed far more false variants than the current state-of-the-art filtering methods and recalled ~51–99% true variants filtered out by the other methods. Once trained, FVC can be conveniently integrated into a user-specific variant calling pipeline.

¹State Key Laboratory of Microbial metabolism, Joint International Research Laboratory of Metabolic & Developmental Sciences, Department of Bioinformatics and Biostatistics, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, China. ²SJTU-Yale Joint Center for Biostatistics and Data Science, National Center for Translational Medicine, Shanghai Jiao Tong University, Shanghai, China. ³Research Affairs, Faculty of Medicine, Chulalongkorn University, Bangkok, Thailand. ⁴Department of Biostatistics, Yale University, New Haven, CT, USA. ⁵Center for Biomedical Informatics, Engineering Research Center for Big Data in Pediatric Precision Medicine, Shanghai Children's Hospital, Shanghai, China. ⁶These authors contributed equally: Yongyong Ren, Yan Kong. ⁷These authors jointly supervised this work: Hongyu Zhao, Hui Lu. ✉email: hongyu.zhao@yale.edu; huilu@sjtu.edu.cn

Whole-genome sequencing (WGS) has been widely used in diagnosing genetic disorders in the pediatrics^{1–4}, exploring causative relations with tumor progression^{5–7}, studying genetic variation underlying pharmaceutical response^{8–10}, performing genome-level comparative analysis^{11,12}, assessing gene expression^{13–15}, and providing clinical insights and instructions^{16–18}. One prominent application with clinical relevance is the utilization of WGS data and bioinformatics tools to identify single nucleotide variants (SNV) and insertion and deletion (INDEL) variants in a single individual genome. The procedure includes at least two main software elements: the variant caller and the variant filter. Variant callers, such as GATK HaplotypeCaller^{19,20}, Mutect2²¹, Varscan2^{22,23}, and DeepVariant²⁴, are utilized to identify the positions and the genotypes of the genomic variants. Variant filters are then applied to eliminate false variants made by the variant caller. This filtering step is necessary due to the fact that there may be tens of thousands of false variants present in the variant call sets.

Current state-of-the-art filtering methods include Frequency²⁵, Hard-Filter²⁰, VQSR²⁶, GARFIELD²⁷, VEF²⁸, ForestQC²⁹ and so on, which employ different strategies in addressing the filtering task. The Frequency model defines variant calls with the variant allelic frequency (VAF) less than 20% or the allelic depth (AD) less than 5 as false variants. The Hard-Filter model applies more user-selected filter conditions to determine the true and false variants³⁰. VQSR uses a Gaussian mixture machine learning algorithm to model the technical profile of variants with high quality and low quality and filter out probable false variants^{30,31}. GARFIELD uses a deep learning method to learn the different characteristics of true and false variants from a standard cross-validated data set (NA12878³²). VEF provides a supervised learning method to build a filtering model and predict the probability of the variants to be true. ForestQC filters variants by combining a traditional filtering method and a machine learning approach.

Although these methods address the problem rigorously and are of great utility, some aspects of the available filtering tools and their performance still merit further improvement. For example, the source code of the Frequency method must be modified to adapt to different variant callers. The Hard-Filter, VQSR, and GARFIELD are developed to quality control variant calls identified by GATK. The VEF constructs a filtering model by selecting a subset of features from the existing features in variant calling results. However, in some cases, such as variants identified by Varscan2 and DeepVariant, no feature could be selected from the variant calling results. Thus, these state-of-the-art methods are limited to particular variant callers.

Furthermore, the Frequency method removes all true variants with low variant allelic frequency (VAF < 20%) based on its definition criteria. Hard-Filter removes true variants even when they are very close to the threshold³³. VQSR is recommended to be used with at least 30 samples²⁹, which may not perform well in a single sample. GARFIELD is explicitly designed for whole-exome data²⁷. VEF only uses the integer or float format features when constructing the filtering model, but the features in character format are also informative. ForestQC cannot be utilized on single-sample sequencing data. Five of these filtering methods (Frequency, VQSR, Hard-Filter, GARFIELD, and VEF) are available for quality control of variants from single-sample sequencing data and showed high performance in F1-major and accuracy²⁸. However, these state-of-the-art methods were unsatisfactory when measured with the Matthews correlation coefficient (MCC) metric²⁷, which is a highly suggested measurement for imbalanced data³⁴, i.e., the WGS variant calls. Moreover, these five filtering methods had a poor performance in balancing minimizing the filtering of true variants and maximizing the removal of false variants. As a result, the number of eliminated

true variants far exceeds the number of removed false variants by using these methods. Therefore, an improved filtering method is required to provide accurate variant call sets derived from a single WGS sample and broaden the scope of the application.

Here, we present an adaptive filtering method FVC (filtering for variant calls) for quality control variant calls from different analysis pipelines. We validated FVC on the genetic variants identified by GATK HaplotypeCaller (abbreviated as GATK), Mutect2, Varscan2, and DeepVariant. Compared to the current state-of-the-art methods, FVC achieved the highest AUC and MCC scores in most cases when assessing with the leave-one-individual-out cross-validation method. We further tested FVC on an additional data set and performed the assessment using the leave-one-chromosome-out cross-validation method. FVC had a consistently superior performance, which has the potential to be used as a general method for quality control variant calls from different analysis pipelines.

Results

Construction of FVC. As illustrated in Fig. 1, FVC incorporates four modules: feature construction, data construction, supervised learning, and filtering module. Taking the VCF and BAM files as the input, FVC uses feature construction module to build three types of features related to sequence content, sequencing experiment, and bioinformatic analysis process. If there is no pre-trained model that can be utilized to the user-specific pipeline, an adaptive filtering model is constructed using the data construction module coupled with the supervised learning module. The variant calls are finally labeled as true or false using the filtering module of FVC, and the probability of the variants being true can be found in the INFO field of the output VCF file.

To assess the classification accuracy of the FVC incorporated with different features, different methods of constructing training data, and different machine learning methods, we considered sixteen evaluation metrics and performed the assessments on the gold-standard variant calls derived from WGS datasets (HG001, HG003, HG004, and HG006) at 30× coverage^{32,35}. Considering different variant callers may give different initial output, we performed the comparisons on the variant calls identified by GATK, Mutect2, Varscan2, and DeepVariant, separately. All the comparisons were implemented using the leave-one-individual-out cross-validation method. Specifically, sampling was performed four times on the four individuals (HG001, HG003, HG004, and HG006). Each time, a different individual was left out, the genetic variants from the withheld individual formed the test set, and the others formed the training set.

As it can be seen in the Supplementary Tables 1–3 and Supplementary Figs. 1–3, the FVC model containing the constructed features and trained on the imbalanced training data and embedding the XGBoost³⁶ as the supervised machine learning method demonstrated the best filtering performance and was incorporated in the final FVC modules (Supplementary Data 1).

Performance comparison of FVC and the current state-of-the-art methods. After constructing FVC, we performed a head-to-head comparison of FVC with five other state-of-the-art methods (VEF, Frequency, Hard-Filter, VQSR, and GARFIELD) in 4 modes. First, we compared filtering performance with a focus on all SNV or INDEL variants; second, we compared filtering performance with a focus on high-frequency (VAF ≥ 20%) or low-frequency (VAF < 20%) variants; third, we compared filtering performance with a focus on hard-to-detect or easy-to-detect variants; and fourth, we compared filtering performance with a focus on coding or non-coding variants. The comparisons were

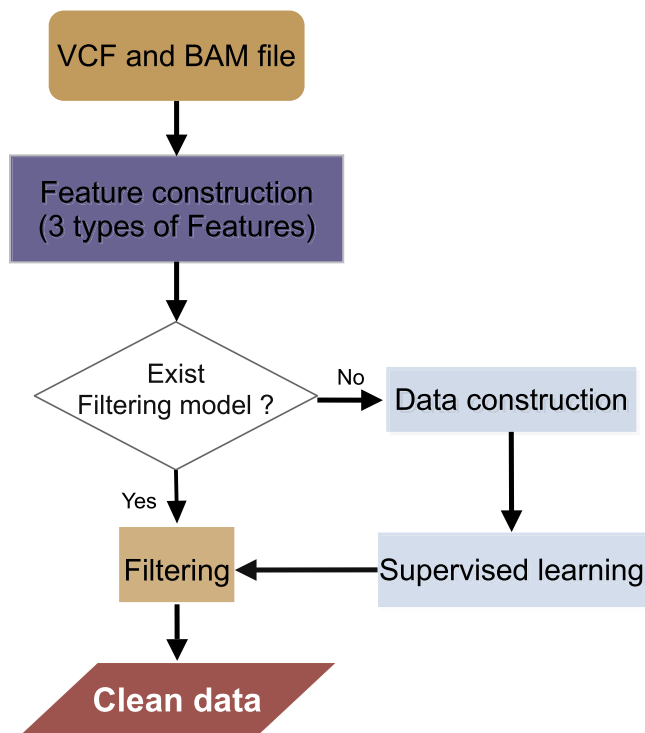


Fig. 1 Workflow of FVC. Taking the VCF and BAM files as input, FVC uses the feature construction module to build three types of features related to sequence content, sequencing experiment, and bioinformatic analysis process. If a pre-trained model is already built for the specific pipeline, the variants can be immediately classified as true or false using the filtering module. Otherwise, the pre-trained model can be built using the data construction and supervised learning modules of FVC.

performed on the variants derived from 4 individuals (HG001, HG003, HG004, HG006) using the leave-one-individual-out cross-validation method. To remove possible bias, we performed the comparisons using the leave-one-chromosome-out cross-validation method and validated the filtering performance on an additional dataset (HG007).

Figure 2 summarized the performance of different filtering methods when applied to all SNV or INDEL variants (30× sequencing coverage). As it can be seen, when applied to the SNV variants identified by GATK, FVC scored the highest average AUC of 0.998, while the rest of the methods scored lower as follows: 0.989 (VEF), 0.785 (Frequency), 0.870 (Hard-Filter), 0.926 (VQSR), and 0.981 (GARFIELD). Concerning the INDEL variants, FVC exhibited even more improvements with an average AUC of 0.984. The rest of the methods scored lower as follows: 0.819 (VEF), 0.853 (Frequency), 0.733 (Hard-Filter), 0.836 (VQSR), and 0.783 (GARFIELD). When running with the default cut-off value of 0.5, FVC was not the best method of eliminating the highest number of false INDEL variants, but it exhibited the best performance in eliminating the total number of false variants (Table 1). The filtering improvements achieved by FVC were also observed when applied to the variants identified by Mutect2, Varscan2, and DeepVariant (Supplementary Tables 4–6). It is worth noting that FVC removed far more false variants than the current state-of-the-art methods in most cases, and it recalled ~51–99% true variants filtered out by the others. Moreover, FVC decreased the ratio of the eliminated true variants versus the removed false variants (OFO) from 0.05–1661.28 to 0.02–0.57 (Supplementary Table 7).

Figure 3 summarized the performance of different filtering methods when applied to high-frequency (VAF ≥ 20%) or low-

frequency (VAF < 20%) variants (30× sequencing coverage). When running FVC with the default cut-off value 0.5 and running other filtering methods with their suggested criteria, FVC also exhibited significant improvements than the other five filtering methods in the two subgroup variants, reflected by the highest MCC score and the lowest OFO score ($p < 0.05$, one-sided paired T-test). All filtering methods except VEF demonstrated better performance when applied to the low-frequency variants than high-frequency ones.

Figure 4 summarized the performance of different filtering methods when applied to easy-to-detect or hard-to-detect variants (30× sequencing coverage). The easy-to-detect variants are defined as the variants that are consistently and correctly classified by the three unsupervised methods (Frequency, Hard-Filter, VQSR), the others are defined as hard-to-detect variants that are incorrectly classified by at least one of the three methods. As can be observed, FVC exhibited superior performance when applied to the hard-to-detect variants in all cases. With respect to the easy-to-detect variants identified by Mutect2 and Varscan2, FVC also achieved significant improvements, reflected by the highest MCC and the lowest OFO ($p < 0.05$, one-sided paired T-test). When assessing on the easy-to-detect variants identified by DeepVariant, VEF scored the highest MCC but FVC achieved the lowest OFO. Both FVC and VEF exhibited better performance than GARFIELD in all cases.

Figure 5 summarized the performance of different filtering methods when applied to coding or non-coding variants. We could find that FVC not only achieved the highest MCC score both in coding and non-coding variants but was also the only method that consistently obtained OFO < 1 in these two types of variants.

We then assessed the performance of different filtering methods on an additional WGS dataset (HG007) which was not used in the above experiments. In this experiment, the pre-trained FVC and VEF were built on the training variants derived from four individuals (HG001, HG003, HG004, and HG006). As shown in Supplementary Fig 4, FVC consistently achieved the highest AUC score on SNV and INDEL variants. The improvements can also be observed when assessing with the corresponding area under the precision-recall-gain curves (AUPRG)³⁷, the MCC score, the accuracy (ACC), the balanced accuracy (BACC), and the OFO (Supplementary Data 2).

The different filtering methods were also assessed using the leave-one-chromosome-out cross-validation method to remove possible bias. Specifically, we used the autosome variants derived from five human samples (HG001, HG003, HG004, HG006, and HG007). Sampling was implemented on the 22 chromosomes 22 times. Each time, a different chromosome was left out, the genetic variants from the left chromosome formed the test data, and the others formed the training data. There is no duplication between the training and testing data. Consistent with the leave-one-individual-out cross-validation measurement results, FVC achieved the highest AUC scores when applied to SNV and INDEL variants (Supplementary Fig. 5).

To further validate the classification performance of FVC, the above comparisons were also performed on the sequencing data at 50× coverage. In particular, the pre-trained models of FVC and VEF built on 30× coverage data were applied to the data at 50× coverage to test their generalization ability. Similar improvements of FVC were also observed in all cases (Supplementary Data 2–4).

Discussion

Herein, with the goal to adapt different variant callers and address the fact that a large number of variant calls are misclassified by current filtering methods, especially in the case of low-frequency

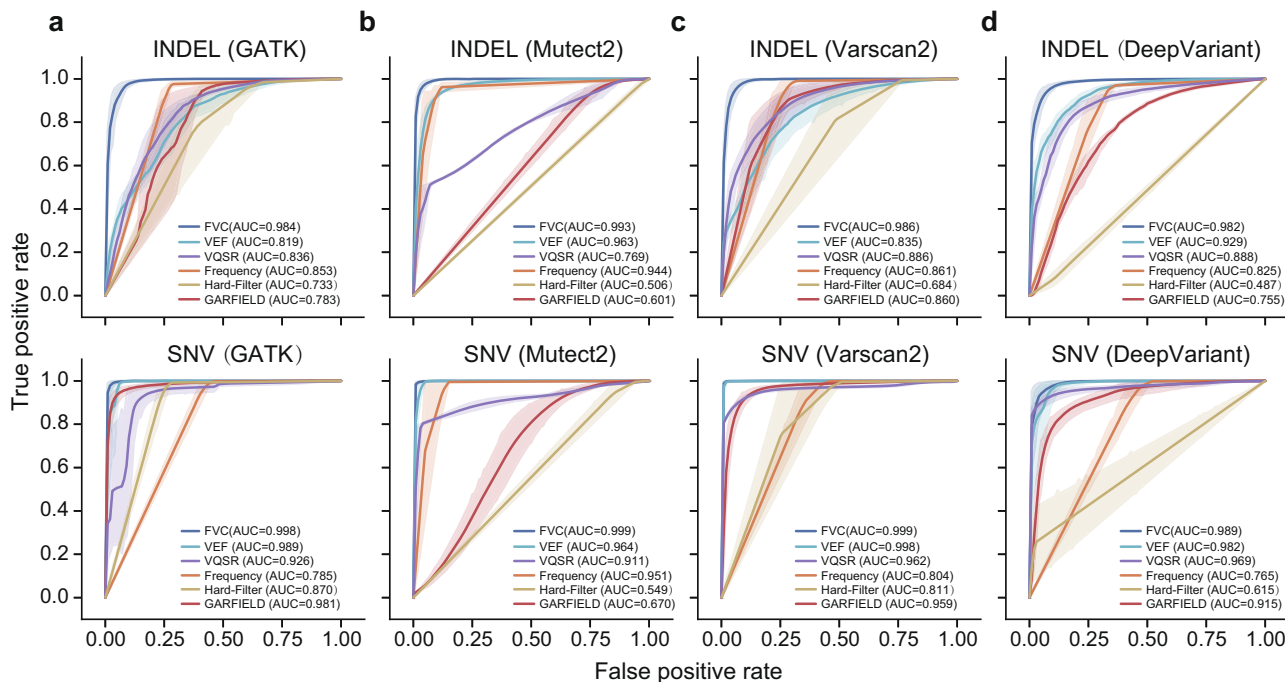


Fig. 2 The performance of different filtering methods when applied to SNV or INDEL variants. The SNV and INDEL variants used as testing data are derived from whole-genome sequencing datasets (HG001, HG003, HG004, and HG006) at 30× coverage. The performance of different methods is assessed on the SNV and INDEL variants identified by **a** GATK HaplotypeCaller; **b** Mutect2; **c** Varscan2; and **d** DeepVariant. The performance is assessed by using the leave-one-individual-out cross-validation method. The shaded area indicates the 95% confidence intervals ($n = 4$ biologically independent samples). FVC consistently achieves the highest AUC score when applied to both SNV and INDEL variants.

Table 1 The average performance of filtering methods when applied to the whole-genome sequencing datasets (HG001, HG003, HG004, and HG006) at 30× coverage and identified by GATK HaplotypeCaller.

Variant	Filter	AUC	AUPRG	MCC	OFO	F1-minor	BACC	NPV	TN	FN
SNV	FVC	0.998	0.92	0.89	0.08	0.89	0.93	0.93	5,869	425
	Frequency	0.785	0.33	0.50	1.54	0.49	0.79	0.45	3,902	5,046
	GARFIELD	0.981	0.35	0.28	10.49	0.19	0.86	0.11	5,119	42,888
	Hard-Filter	0.870	0.14	0.37	6.29	0.31	0.87	0.21	5,123	25,187
	VEF	0.989	0.87	0.84	0.17	0.84	0.91	0.86	5,627	894
INDEL	VQSR	0.926	0.05	0.18	24.96	0.09	0.86	0.05	5,121	112,859
	FVC	0.984	0.68	0.70	0.16	0.68	0.79	0.87	543	86
	Frequency	0.853	0.04	0.21	19.13	0.12	0.85	0.07	668	10,855
	GARFIELD	0.783	0.13	0.09	69.39	0.03	0.79	0.01	642	46,167
	Hard-Filter	0.733	0.02	0.14	22.98	0.08	0.73	0.04	477	10,636
	VEF	0.819	0.05	0.07	8.67	0.06	0.52	0.11	37	306
	VQSR	0.836	0.08	0.15	13.66	0.13	0.62	0.09	246	2,464

TN is the number of false variants that are filtered; FN is the number of true variants that are filtered. OFO equivalents to the ratio of the number of true variants that are eliminated (FN) versus the number of false variants that are removed (TN).

or hard-to-detect SNV and INDEL variants, we presented FVC—an adaptive method for quality control of variant calls, and demonstrated the improved performance both in minimizing the eliminating of true variants and maximizing the removal of false variants.

In most cases, the filtering method developed for a specific variant caller cannot be applied to other variant callers. It is partly because the development of filtering methods relies on the features generated by a specific variant caller. However, for the same variation, different variant callers will generate different numbers and types of features, making the filtering method unable to be used in different variant callers. To solve this problem, we developed a data construction module to build consistent features for the variants from different variant callers. Ideally, it can be

immediately integrated into any variant calling pipeline. However, though we have tested its ability on four widely used variant calling pipelines, limitations still exist when the specific pipeline doesn't provide the variants in VCF format.

Additionally, we found that the FVC models using the imbalanced training data outperformed those using balanced ones, which is different from the previous work³⁸. This aspect is partly due to the fact that the ratio of the two classes in our study is up to 2204:1, which is much higher than the imbalance ratio of 10:1 in the previous study, and the number of samples (millions) used in this study is far more than the number of samples (thousands) in the study mentioned above.

Concerning the adaptive ability, we have provided four pre-trained models for filtering variants detected by GATK, Mutect2,

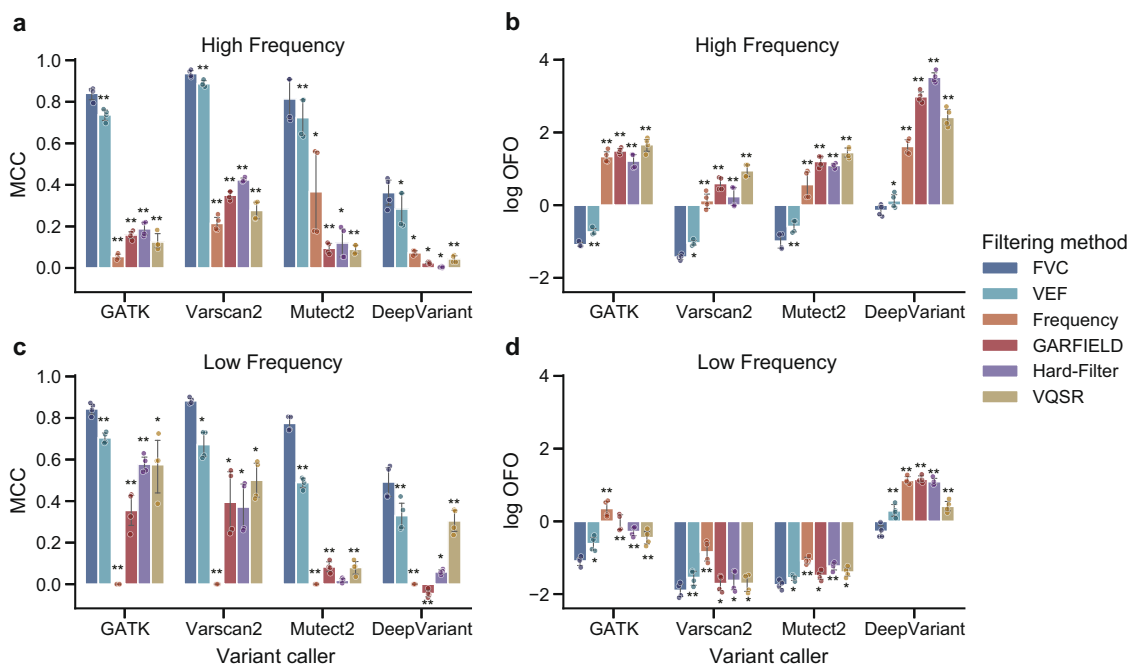


Fig. 3 The performance of different filtering methods when applied to high-frequency variants or low-frequency variants. The high-frequency variants and low-frequency variants used as testing data are derived from whole-genome sequencing datasets (HG001, HG003, HG004, and HG006) at 30× coverage and identified by GATK, Varscan2, Mutect2, and DeepVariant, separately. Variant calls with variant allelic frequency (VAF) of more than 20% are defined as high-frequency variants. The other variant calls are defined as low-frequency variant calls. The performance of the filtering methods is assessed on the **a** high frequency variants using MCC; **b** high frequency variants using OFO; **c** low frequency variants using MCC; and **d** low frequency variants using OFO. The circle indicates the metric score achieved by FVC when applied to each specified testing data. The error bar indicates the 95% confidence intervals ($n = 4$ biologically independent samples). Asterisk denotes the significance of the comparison using a one-sided paired T-test ($*p < 0.05$, $**p < 0.001$), where the null hypothesis is that the FVC performs no better than the compared method. FVC consistently shows $\log \text{OFO} < 0$ when applied to high-frequency and low-frequency variants.

Varscan2, and DeepVariant. However, the needs of users are not limited to the four variant callers. In case the user's goal is to filter variants identified by a different pipeline, such as Pindel³⁹ or Strelka2⁴⁰, an adaptive filtering model could be built by providing three or four gold standard samples' variants identified by the variant caller. It should be noted that the same model features, such as the median base quality (MBQ) or the variant allelic frequency (AF), showed different rankings of feature importance in different pre-trained models (Supplementary Fig 6). Hence, we suggest that the pre-trained model should be used on the variants derived from the same caller.

FVC was developed and assessed on the sequencing data consisting of 150bp and 250bp paired-end reads, and exhibited excellent performance on the variant calls derived from sequencing data at 30× coverage and 50× coverage. However, other usage scenarios of FVC, such as on data produced by utilizing different sequencing libraries (such as 2×300bp pair-end sequencing), different sequencing machines (such as Hiseq X), or different sequencing coverages (such as 10×), are feasible. In such cases, an adaptive pre-trained model could also be built by providing the gold standard samples' variants derived from a public database or private sequencing data. The main limitation of FVC is that it focuses on the application in a single sample, and the information from other samples is not incorporated into FVC. For this case, other methods such as VQSR and ForestQC may be preferred, but FVC also supports users splitting the VCF file with multiple samples into multiple VCF files with a single sample and then performing quality control independently.

Considering that the comparisons in this study are all measured on the germline variants, the analytical performance may differ in tumor samples as somatic variants are often at lower

than 20% variant allele frequency⁴¹. However, we find that FVC achieved similar performance when applied to the low-frequency variants and the high-frequency variants in some circumstances. For example, when applied to the variants detected by GATK HaplotypeCaller, FVC achieved similar MCC scores between high-frequency variants (0.841) and low-frequency variants (0.844). When applied to the low-frequency variants detected by Mutect2 and Varscan2, FVC exhibited slightly lower MCC scores within 0.06 on average. Thus, it may imply that FVC could perform well for somatic variant filtration when enough gold-standard somatic variant calls become available for building a pre-trained model.

Furthermore, FVC could also be helpful for post-filtering of RNA-sequencing mutation detection pipelines as they are similar with the DNA-sequencing mutation detection pipelines^{20,42}, such as performing read alignment by using the BWA or other alignment tools and detecting variations by using GATK, SNPiR⁴³ or other variant callers^{44,45}.

Users should comprehensively consider the evaluation results of multiple metrics when deciding which software to use. In this study, we performed the comparisons by using sixteen metrics. FVC does not perform well on all evaluation metrics (Supplementary Data 3 and 4). Actually, these filtering methods exhibited little difference in the F1-major score, sensitivity, and specificity. For example, we can find that all filtering methods scored F1-major score > 0.96, sensitivity > 0.93, and specificity > 0.998 when applied to GATK detected variants in HG001 sample, the recall was decreased and the precision was improved after filtering by any one of the filtering methods (Supplementary Data 3).

However, when assessing with the newly defined metric OFO, the filtering methods exhibited extremely different performance.

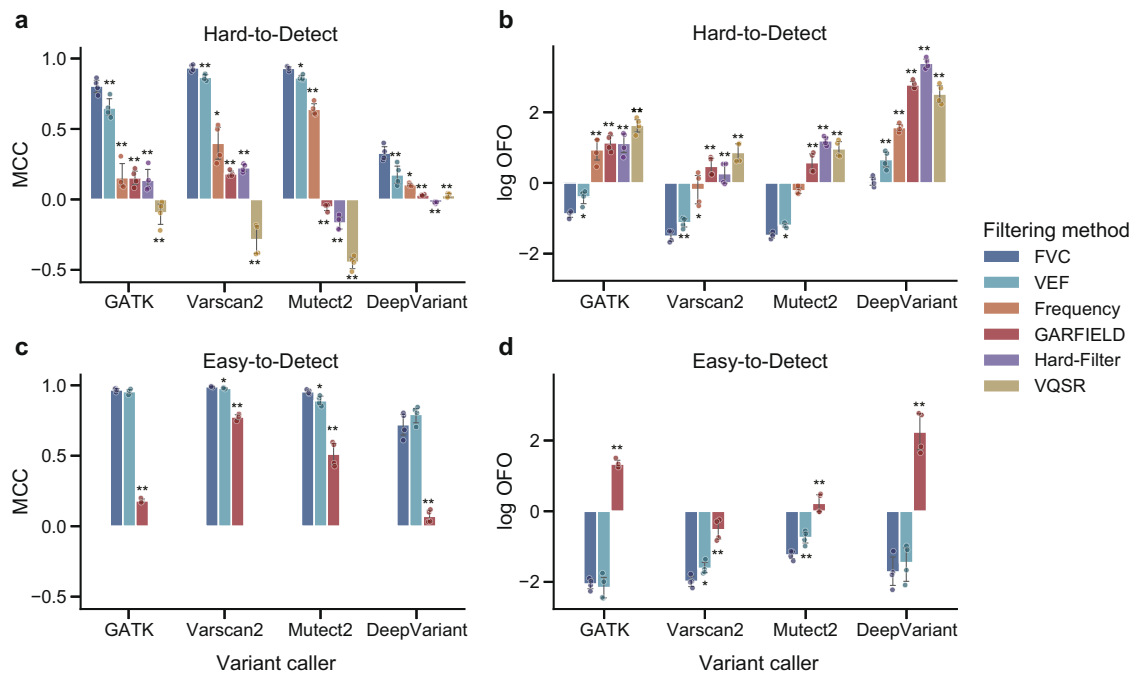


Fig. 4 The performance of different filtering methods when applied to hard-to-detect or easy-to-detect variants. The easy-to-detect and hard-to-detect variant calls used as testing data are derived from the whole-genome sequencing datasets (HG001, HG003, HG004, and HG006) at 30× coverage and identified by GATK, Varscan2, Mutect2, and DeepVariant, separately. The easy-to-detect variants are defined as the variants that are consistently and correctly classified by all the unsupervised filtering methods (Frequency, Hard-Filter, and VQSR). The other variants are defined as hard-to-detect variants. The performance of the filtering methods is assessed on the **a** hard-to-detect variants using MCC; **b** hard-to-detect variants using OFO; **c** easy-to-detect variants using MCC (Frequency, Hard-Filter, and VQSR consistently scored MCC = 1); and **d** easy-to-detect variants using OFO (Frequency, Hard-Filter, and VQSR consistently scored logOFO = $-\infty$). The circle indicates the metric score achieved by the filtering method when applied to each specified testing data. The error bar indicates the 95% confidence intervals ($n = 4$ biologically independent samples). Asterisk denotes the significance of the comparison using a one-sided paired T-test (* $p < 0.05$, ** $p < 0.001$), where the null hypothesis is that the FVC performs no better than the compared method.

For example, when applied to the GATK detected variants from WGS Human sample HG001 (Supplementary Table 8), the Hard-Filter eliminated 24.06 true INDELS and 9.83 true SNVs per removing one false variant correspondingly. FVC exhibited not extremely different with the Hard-Filter in recall, precision, and F1-major score. But it decreased the loss of true variants to 0.16 and 0.08 per filtering one false INDEL and one false SNV, respectively. Though the Hard-Filter achieved similar F1-major score with FVC, the presented FVC method demonstrated more suitable in the filtering of GATK detected variants. Therefore, in the choosing of which filter to use, users should make further decisions based on OFO and other comprehensive metrics, such as AUC and MCC, when the objective metrics achieved similar scores.

Taking it all together, FVC presented a superior performance in the accuracy, generalization ability, application scope, which could potentially be used to integrate variant calls detected by multiple variant callers.

Methods

Data preparation. Whole-genome sequencing data from five individuals were used in this study: one pilot genome HG001/NA12878 from the HapMap project⁴⁶; two Ashkenazim individuals—HG003/NA24149 and HG004/NA24143; and two Chinese individuals—HG006/NA24694 and HG007/NA24695. There is no consanguinity between these five samples and all these sequencing data were released by NIST's GIAB consortium^{32,35}. The downloaded whole-genome sequencing datasets were generated on Illumina HiSeq 2500 platform (Illumina Inc, San Diego, USA) with 2 × 148bp (HG001, HG006, HG007) or 2 × 250bp (HG003 and HG004) paired-end reads. The mean coverage of the sequencing data ranged from 50× to 300×. The source of the dataset is listed in the Supplementary Method.

Firstly, the downloaded sequencing data was realigned to human genome build GRCh37 with the same pipeline. The sequencing reads were firstly randomly downsampled to ~30× and ~50× coverage by using Samtools⁴⁷. Such levels are

commonly used in WGS studies^{48–50}, and few uncovered or uncalled bases above these depths⁵¹. Then, the reads in FASTQ format were realigned and converted to BAM format by performing sequencing realignment, marking duplicates, and local realignment using the BWA-MEM, Dedup, and Realigner software which were integrated into Sentieon's DNaseq⁵² pipeline.

Then, the genetic variants were derived by GATK HaplotypeCaller (version 4.0.11, with default parameters), Mutect2 (integrated in GATK version 4.1.9 with default parameters), Varscan2 (version 2.3.9 with default parameters, except where parameters—min-coverage 3,—p-value 0.10,—min-var-freq 0.01 were used), and DeepVariant (version 1.2, with default parameters). All variants were stored in variant call format (VCF) files.

Finally, true variants and false variants in the VCF files were labeled based on whether the variants were contained in the GIAB's gold-standard variants or not with the help of RTG-vcfEval⁵³ software (version 3.10 with parameters 'squash-ploidy' to ignore the zygosity differences). The resultant 'true' and 'false' label-containing genetic variants from the four individuals (HG001, HG003, HG004, and HG006) were then utilized downstream in the leave-one-individual-out cross-validation assessment. The remainder variants from the HG007 sample were used as independent testing data. The true and false variants distributions in different subgroups are listed in Supplementary Table 9 and Supplementary Data 5.

FVC feature construction module. A total of 20 features associated with each variant are selected or constructed by the FVC feature construction module. Firstly, FVC adds features for each genetic variant in the raw input VCF file by using GATK VariantAnnotator (version 4.1.9). Then, FVC selects and constructs features based on the variant position, variant types (SNV or INDEL), INFO column, and FORMAT column in the aforementioned processed VCF file. All the constructed features can be classified into three categories: sequence-related features ($n = 4$), experiment-related features ($n = 8$), and analysis-related features ($n = 8$). The definitions of each feature can be found in the Supplementary Method. The features in raw VCF and the features built by the FVC feature construction module are listed in Supplementary Table 10.

FVC data construction module. In the data construction module, the RTG vcfEval software and the imbalanced method are introduced to construct training and testing data. Firstly, all the variants derived from four individuals (HG001, HG003, HG004, and HG006) are labeled as true or false genetic variants by using RTG

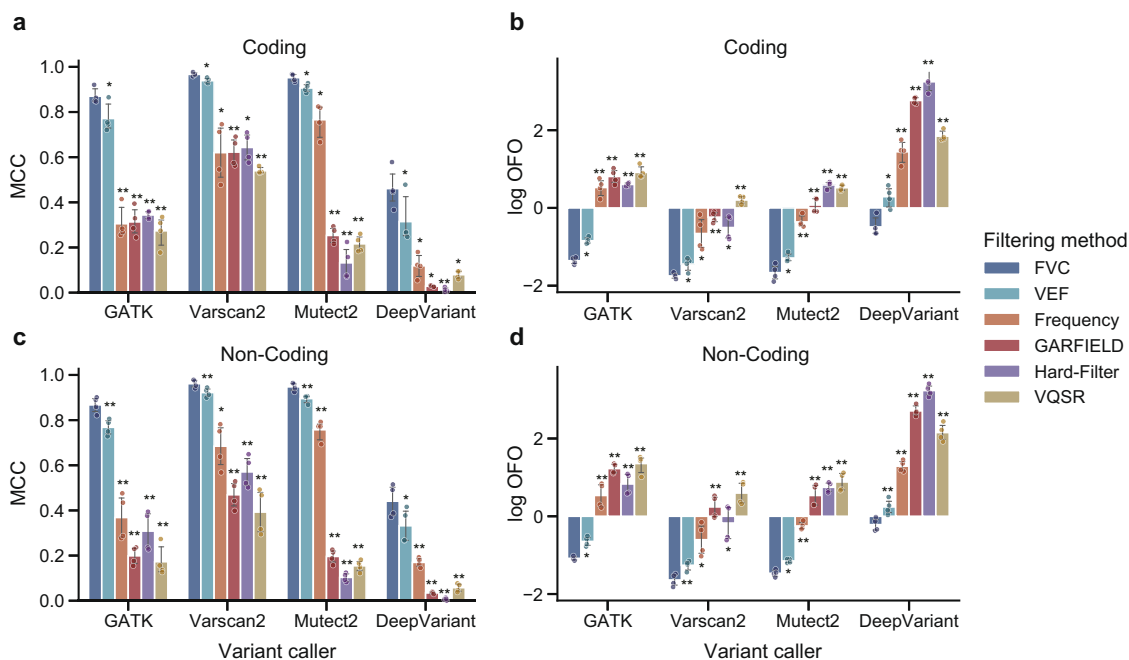


Fig. 5 The performance of different filtering methods when applied to coding or non-coding variants. The coding and non-coding variant calls used as testing data are derived from whole-genome sequencing data (HG001, HG003, HG004, and HG006) at 30× coverage and identified by GATK, Varscan2, Mutect2, and DeepVariant, separately. The different filtering methods are separately assessed on the **a** coding variants using MCC; **b** coding variants measured using OFO; **c** non-coding variants measured using MCC; and **d** non-coding variants using OFO. FVC achieves the highest MCC and the lowest log OFO when applied to both types of variants identified by GATK HaplotypeCaller, Varscan2, Mutect2, and DeepVariant. The circle indicates the metric score achieved by the filtering method when applied to each specified testing data. The error bar indicates the 95% confidence intervals ($n = 4$ biologically independent samples). Asterisk denotes the significance of the comparison using a one-sided paired T-test ($*p < 0.05$, $**p < 0.001$), where the null hypothesis is that the FVC performs no better than the compared method.

vcfeval software according to comparison results with the gold standard variants. Then, the variants from three of four individuals are combined as the training data without balancing the two categories (true and false variants) in the leave-one-individual-out cross-validation step, and the variants from the left individual are regarded as the testing data. The genetic variants with the same chromosome, genome position, reference allele, alternative allele, features, and categories (true variant or false variant) are regarded as duplicates and are removed from the training data if they are also included in the test data in case of data leakage.

FVC supervised learning and filtering module. FVC applies the XGBoost as the supervised learning algorithm and models the technical profile of true and false variants based on the features and labels built by the feature construction module and data construction module. Once trained, the FVC can be utilized to filter the variants identified from the same pipeline. In the filtering results, FVC provides a self-defined score in INFO field to indicate the probability of the variant being correct and specifies “Filtered” in the FILTER field in VCF if the score is less than 0.5. The default parameters settings of XGBoost and the other candidate machine learning methods are listed in Supplementary Table 11.

Performance analysis and dealing with data leakage. We compared the FVC with five other methods when applied to the WGS data. FVC used by default a threshold of 0.5, i.e., the genetic variant is regarded as a false variant if its probability of being correct is less than 0.5. The false variants defined by the other five methods were based on their suggested criteria (Supplementary Method).

The filtering methods were assessed in a head-to-head comparison using the leave-one-individual-out cross-validation method. For the assessment, we used the variant calls derived from 4 human samples (HG001, HG003, HG004, and HG006) and performed 4 subgroup analysis, including: 1) Assessing on SNV and INDEL variants, separately; 2) Assessing on high-frequency ($VAF \geq 20\%$) and low-frequency ($VAF < 20\%$) variants, separately; 3) Assessing on hard-to-detect and easy-to-detect variants separately; 4) Assessing on coding and non-coding variants, separately.

To deal with the potential data leakage and bias, the same variant calls were removed from the training data if they also appeared in the testing data. Moreover, we compared the filtering methods on an additional variant call-set derived from the human sample HG007 which is not genetically related to the other individuals. Subsequently, the performance of the different filtering methods was further assessed using the leave-one-chromosome-out cross-validation method to ensure that there is no duplicate between the training and testing data. As not all gold-

standard variant call-sets contain sexual chromosomes, the variants located on the autosomes were used in the cross-validation assessment. Specifically, sampling was implemented on the dataset 22 times. Each time a different chromosome was left out, the variant calls from the left chromosome formed the test dataset, and the others formed the training dataset.

Evaluation metrics. Sixteen evaluation metrics were utilized to assess the performance of these different filtering methods. Three of them were used to assess the comprehensive performance under different thresholds, including the area under the receiver characteristic operator curve (AUC), the area under the precision-recall-gain curve (AUPRG), and the area under the precision-recall curve (AUPRC). It is worth noting that the outputs of Frequency and Hard-Filter were dichotomous. Therefore, we calculate these metrics in the two cases using the curve with three points: $(0, y_1)$, (x_2, y_2) , $(1, y_3)$, where x and y are the values corresponding to the axes. For example, in calculating the AUC, y_1 is the value of TPR (i.e., 0) when $FPR = 0$, x_2 and y_2 are the values of FPR and TPR using the suggested thresholds, respectively, y_3 is the value of TPR (i.e., 1) when $FPR = 1$.

Four of these metrics were used to demonstrate the count of variants that were correctly or incorrectly retained or filtered by using the particular filtering method. True Positive (TP): the number of true variants that were retained; False Positive (FP): the number of false variants that were retained; False Negative (FN): The number of true variants that were eliminated; True Negative (TN): The number of false variants that were eliminated.

Seven metrics were used to demonstrate the comprehensive performance under the suggested threshold, including: F1 score for the majority class (F1-major); F1 score for the minority class (F1-minor); Matthews Correlation Coefficient (MCC); Accuracy (ACC); Balanced Accuracy (BACC); Precision; Sensitivity or named as True Positive Rate (TPR); Specificity or named as True Negative Rate (TNR).

The proportion of the false variants in the filtered variants can be assessed by Negative Predictive Value (NPV) in Eq. (1). However, one of the motivations in the filtering task is to minimize the number of true variants and maximize the number of false variants in the eliminated variant set (the predictive negative class). The NPV is necessary but cannot be used for this purpose. Therefore, we introduced the odds of false omission in the predicted negative class to intuitively describe the proportion of true and false variants in the eliminated variant set, which was abbreviated to OFO in this study (Eq. (2)). It is equivalent to the ratio of the number of eliminated true variants (FN) versus the number of eliminated false

variants (TN).

$$NPV = \frac{TN}{TN + FN} \quad (1)$$

$$OFO = \frac{FOR}{1 - FOR} = \frac{FOR}{NPV} = \frac{FN}{TN} \quad (2)$$

Here, the false omission rate (FOR) = $\frac{FN}{TN + FN}$. OFO ranges from 0 to $+\infty$. When OFO = 1, the number of eliminated false variants is equal to the number of eliminated true variants. When OFO = 0, it means that there is no true variant in the eliminated variants, which seems perfect. $+\infty$ occurs when $FN \gg TN$, i.e., the number of true variants far exceeds the number of false variants in the filtered variant sets, which indicates the worst performance. In some circumstances, the filtering performance may be overestimated with OFO = 0. For example, if the filtering method eliminates only one genetic variant, and it is the false variant, i.e., the $FN = 0$, and the $TN = 1$, the filtering method will be overestimated with OFO = 0. Therefore, similar to sensitivity and specificity, OFO cannot be used alone to measure the model's comprehensive performance.

Statistics and reproducibility. One-sided paired T-test was applied for pairwise group comparisons where the null hypothesis was that FVC performed no better than the compared filtering method. A $p < 0.05$ was considered statistically significant. * $p < 0.05$, ** $p < 0.001$. All replicate experiments were performed using the leave-one-out cross-validation method. The statistical analysis and plotting were completed using the scikit-learn library in Python (version 3.6).

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The raw data that support the findings of this study are publicly available in NIST's GIAB repository (https://github.com/genome-in-a-bottle/giab_data_indexes/tree/master)⁵⁴. The processed data that support the findings of this study are committed on the Dryad Digital Repository (<https://doi.org/10.5061/dryad.hdr7sqvkm>)⁵⁵. The source data underlying the graphs are provided within Supplementary Data files 1–5 (excel).

Code availability

FVC is implemented in Perl (version 5.0) and Python (version 3.6), and the source code is publicly available at <https://github.com/yyren/FVC> and Zenodo (<https://doi.org/10.5281/zenodo.6379296>)⁵⁶.

Received: 9 July 2021; Accepted: 22 April 2022;

Published online: 16 September 2022

References

- Wright, C. F., FitzPatrick, D. R. & Firth, H. V. Paediatric genomics: diagnosing rare disease in children. *Nat. Rev. Genet.* **19**, 253–268 (2018).
- Stranneheim, H. et al. Integration of whole genome sequencing into a healthcare setting: high diagnostic rates across multiple clinical entities in 3219 rare disease patients. *Genome Med.* **13**, 40 (2021).
- Wade, C. H., Tarini, B. A. & Wilfond, B. S. Growing up in the genomic era: implications of whole-genome sequencing for children, families, and pediatric practice. *Annu. Rev. Genomics Hum. Genet.* **14**, 535–555 (2013).
- Jiang, J. et al. Genomic analysis of a spinal muscular atrophy (SMA) discordant family identifies a novel mutation in TLL2, an activator of growth differentiation factor 8 (myostatin): a case report. *BMC Med. Genet.* **20**, 204 (2019).
- Newell, F. et al. Whole-genome landscape of mucosal melanoma reveals diverse drivers and therapeutic targets. *Nat. Commun.* **10**, 3163 (2019).
- Priestley, P. et al. Pan-cancer whole-genome analyses of metastatic solid tumours. *Nature* **575**, 210–216 (2019).
- Zhao, E. Y., Jones, M. & Jones, S. J. M. Whole-genome sequencing in cancer. *Cold Spring Harb. Perspect. Med.* **9**, a034579 (2019).
- Lorenzo-Salazar, J. M. & Flores, C. Assessing asthma medication responses in U.S. minority children by whole-genome sequencing. *Am. J. Respir. Crit. Care Med.* **197**, 1513–1514 (2018).
- Cordero, P. & Ashley, E. A. Whole-genome sequencing in personalized therapeutics. *Clin. Pharm. Ther.* **91**, 1001–1009 (2012).
- Mak, A. C. Y. et al. Whole-genome sequencing of pharmacogenetic drug response in racially diverse children with asthma. *Am. J. Respir. Crit. Care Med.* **197**, 1552–1564 (2018).
- Oti, M. & Sammeth, M. Comparative genomics in homo sapiens. *Methods Mol. Biol.* **1704**, 451–472 (2018).
- Allentoft, M. E. et al. Population genomics of Bronze Age Eurasia. *Nature* **522**, 167–172 (2015).
- Ochoa, D. et al. The functional landscape of the human phosphoproteome. *Nat. Biotechnol.* **38**, 365–373 (2020).
- Li, X. et al. The impact of rare variation on gene expression across tissues. *Nature* **550**, 239–243 (2017).
- Werling, D. M. et al. Whole-genome and RNA sequencing reveal variation and transcriptomic coordination in the developing human prefrontal cortex. *Cell Rep.* **31**, 107489 (2020).
- Jiang, J., Gu, J., Zhao, T. & Lu, H. VCF-Server: a web-based visualization tool for high-throughput variant data mining and management. *Mol. Genet. Genom. Med.* **7**, e00641 (2019).
- van Dessel, L. F. et al. The genomic landscape of metastatic castration-resistant prostate cancers reveals multiple distinct genotypes with potential clinical impact. *Nat. Commun.* **10**, 5251 (2019).
- Wise, A. L. et al. Genomic medicine for undiagnosed diseases. *Lancet* **394**, 533–540 (2019).
- McKenna, A. et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20**, 1297–1303 (2010).
- Van der Auwera, G. A. et al. From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. *Curr. Protoc. Bioinforma.* **43**, 1–33 (2013).
- Benjamin, D., et al. Calling somatic SNVs and indels with Mutect2. Preprint at <https://www.biorxiv.org/content/10.1101/861054v1> (2019).
- Koboldt, D. C. et al. VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* **25**, 2283–2285 (2009).
- Koboldt, D. C. et al. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.* **22**, 568–576 (2012).
- Poplin, R. et al. A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36**, 983–987 (2018).
- Huang, K. L. et al. Pathogenic germline variants in 10,389 adult cancers. *Cell* **173**, 355–370 (2018).
- Highnam, G. et al. An analytical framework for optimizing variant discovery from personal genomes. *Nat. Commun.* **6**, 6275 (2015).
- Ravasio, V., Ritelli, M., Legati, A. & Giacopuzzi, E. GARFIELD-NGS: genomic vARiants filtering by dEep learning moDEls in NGS. *Bioinformatics* **34**, 3038–3040 (2018).
- Zhang, C. & Ochoa, I. VEF: a variant filtering tool based on ensemble methods. *Bioinformatics* **36**, 2328–2336 (2020).
- Li, J. et al. ForestQC: Quality control on genetic variants from next-generation sequencing data using random forest. *PLoS Comput. Biol.* **15**, e1007556 (2019).
- DePristo, M. A. et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43**, 491–498 (2011).
- Carson, A. R. et al. Effective filtering strategies to improve data quality from population-based whole exome sequencing studies. *BMC Bioinforma.* **15**, 125 (2014).
- Zook, J. M. et al. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32**, 246–251 (2014).
- Adelson, R. P. et al. Empirical design of a variant quality control pipeline for whole genome sequencing data using replicate discordance. *Sci. Rep.* **9**, 16156 (2019).
- Chicco, D. & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **21**, 6 (2020).
- Zook, J. M. et al. An open resource for accurately benchmarking small variant and reference calls. *Nat. Biotechnol.* **37**, 561–566 (2019).
- Chen T., Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794 (ACM, 2016).
- Flach P. A., Kull M. Precision-recall-gain curves: PR analysis done right. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 15, 838–846 (NIPS, 2015).
- Wei, Q. & Dunbrack, R. L. Jr The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS One* **8**, e67863 (2013).
- Ye, K., Schulz, M. H., Long, Q., Apweiler, R. & Ning, Z. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* **25**, 2865–2871 (2009).
- Kim, S. et al. Strelka2: fast and accurate calling of germline and somatic variants. *Nat. Methods* **15**, 591–594 (2018).
- Jones, W. et al. A verified genomic reference sample for assessing performance of cancer panels detecting small variants of low allele frequency. *Genome Biol.* **22**, 111 (2021).
- Coudray, A., Battenhouse, A. M., Bucher, P. & Iyer, V. R. Detection and benchmarking of somatic mutations in cancer genomes using RNA-seq data. *PeerJ* **6**, e5362 (2018).

43. Piskol, R., Ramaswami, G. & Li, J. B. Reliable identification of genomic variants from RNA-seq data. *Am. J. Hum. Genet* **93**, 641–651 (2013).
44. Neums, L. et al. VaDiR: an integrated approach to Variant Detection in RNA. *Gigascience* **7**, 1–13 (2018).
45. Gu, M. et al. RNAmut: robust identification of somatic mutations in acute myeloid leukemia using RNA-sequencing. *Haematologica* **105**, e290–e293 (2020).
46. International HapMap C. et al. A second generation human haplotype map of over 3.1 million SNPs. *Nature* **449**, 851–861 (2007).
47. Li, H. et al. The sequence alignment/map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
48. Griffith, M. et al. Optimizing cancer genome sequencing and analysis. *Cell Syst.* **1**, 210–223 (2015).
49. Gudbjartsson, D. F. et al. Sequence variants from whole genome sequencing a large group of Icelanders. *Sci. Data* **2**, 150011 (2015).
50. Plassais, J. et al. Whole genome sequencing of canids reveals genomic regions under selection and variants influencing morphology. *Nat. Commun.* **10**, 1489 (2019).
51. Bentley, D. R. et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456**, 53–59 (2008).
52. Kendig, K. I. et al. Sentieon DNaseq variant calling workflow demonstrates strong computational performance and accuracy. *Front. Genet* **10**, 736 (2019).
53. Krusche, P. et al. Best practices for benchmarking germline small-variant calls in human genomes. *Nat. Biotechnol.* **37**, 555–560 (2019).
54. Zook J. M., et al. An open resource for accurately benchmarking small variant and reference calls. *Github* https://github.com/genome-in-a-bottle/giab_data_indexes/tree/master (2019).
55. Ren Y. Y., et al. FVC is an adaptive method for filtering variant calls from different analysis pipelines. *Dryad* <https://doi.org/10.5061/dryad.hdr7sqvkm> (2022).
56. Ren Y. Y., et al. FVC is an adaptive method for filtering variant calls from different analysis pipelines. *Zenodo* <https://doi.org/10.5281/zenodo.6379296> (2022).

Acknowledgements

This work was supported by the National Key R&D Program of China (2018YFC0910500); Clinical Research Plan of SHDC (SHDC2020CR6028, SHDC2020CR1047B); SJTU-Yale Collaborative Research Seed Fund; the Neil Shen's SJTU Medical Research Fund; and the Second Century Fund (C2F), Chulalongkorn University. We thank Ms. Fang Dai for helping to check the consistency of the data in the text, figures, and tables, and thank Drs Tao Wang, Yue Zhang, Ningshan Li, and Jianlei Gu from Shanghai Jiao Tong University for their helpful discussions and suggestions. We also thank the editors and reviewers for their valuable comments and suggestions.

Author contributions

Y.R. and H.L. conceived the concept of the study. Y.R. performed the sequencing data analysis, model design, statistical analysis, and drafted the manuscript. Y.K. participated in the study design, variant feature extraction, and drew the figures. X.Z. participated in the study design, compared different filtering models, and drew the result figures. G.G. contributed to the data interpretation and manuscript writing. C.Z. and H.Z. supported the statistical analysis. H.L. and H.Z. supervised all aspects of the study. All authors reviewed and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42003-022-03397-7>.

Correspondence and requests for materials should be addressed to Hongyu Zhao or Hui Lu.

Peer review information *Communications Biology* thanks Mark Cowley and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editors: Melanie Bahlo and Luke R. Grinham.

Reprints and permission information is available at <http://www.nature.com/reprints>

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022