Synth4bench: a framework for generating synthetic genomics data for the evaluation of tumor-only somatic variant calling algorithms

Styliani-Christina Fragkouli[1,2,*], **Nikos Pechlivanis**[2], **Anastasia Anastasiadou**[2], **Georgios Karakatsoulis**[2], **Aspasia Orfanou**[2], **Panagoula Kollia**[1], **Andreas Agathangelidis**[1], **Fotis Psomopoulos**[2,*]

[1]Department of Biology, National and Kapodistrian University of Athens, Athens, GR,
[2]Institute of Applied Biosciences, Centre for Research and Technology Hellas, Thessaloniki, GR

*Correspondence to: sfragkoul@certh.gr, fpsom@certh.gr

Abstract

Motivation

Somatic variant calling algorithms are widely used to detect genomic alterations associated with cancer. Evaluating their performance, even though being crucial, can be challenging due to the lack of high-quality ground truth datasets. To address this issue, we developed a synthetic data generation framework for benchmarking these algorithms, focusing on the TP53 gene, utilizing the NEATv3.3 simulator. We thoroughly evaluated the performance of Mutect2, Freebayes, VarDict, VarScan2 and LoFreq and compared their results with our synthetic ground truth, while observing their behavior. Synth4bench attempts to shed light on the underlying principles of each variant caller by presenting them with data from a given range across the genomics data feature space and inspecting their response.

Results

Using synthetic dataset as ground truth provides an excellent approach for evaluating the performance of tumor-only somatic variant calling algorithms. Our findings are supported by an independent statistical analysis that was performed on the same data and output from all callers. Overall, synth4bench leverages the effort of benchmarking algorithms by offering the opportunity to utilize a generated ground truth dataset. This kind of framework is essential in the field of cancer genomics, where precision is an ultimate necessity, especially for variants of low frequency. In this context, our approach makes comparison of various algorithms transparent, straightforward and also enhances their comparability.

Availability: synth4bench is available at https://github.com/BiodataAnalysisGroup/synth4bench and all data available at https://zenodo.org/records/10683211.

Keywords: synthetic data, variant calling, bioinformatics framework

1. Introduction

It is commonly known that cancer is caused by alterations in the DNA of organisms. These alterations are induced by a plethora of factors such as aging, heredity, lifestyle, viral infections and more.

High throughput sequencing (HTS) technologies have greatly assisted in capturing the actual genomic complexity of cancer, through the identification of thousands of somatic mutations related to a variety of human cancers [1]. Furthermore, accurate detection of somatic variants plays a pivotal role in unraveling the genomic alterations that represent the actual drivers of this complex disease. To date, a large series of bioinformatics tools are available for the detection of single nucleotide variants (SNVs) and small insertions and deletions (indels). Yet, the challenge that rises with somatic variant calling is the absence of standards and criteria that would ensure its accuracy and consistency. Furthermore, in the case of variants identified at a low allele frequency (AF) (i.e. ≤10%), robust identification becomes even more challenging, requiring higher levels of sensitivity across the analytical workflow [2].

This open issue is reflected in the very low percentage of agreement of the results provided by different variant callers. To this date, several publications have dealt with the benchmarking and comparison of relevant algorithms [3] [4] [5]. In an effort to mend this outcome, improvements have been made and some tricks have been incorporated in variant calling pipelines to increase the reliability of their discordant and inconsistent results. To name a few of these ensemble approaches, consensus methods [6], majority voting [7], as well as methods from the domain of machine learning such as Bayesian inference [8], decision trees [9] and deep learning [10] have been applied.

Furthermore, the assessment of variant calling results has long been hindered by the scarcity of high-quality datasets that could serve as ground truth. With the implementation of such datasets, the process of benchmarking would be more efficient and fiducial [11]. To address this challenge, a solution has emerged from the field of synthetic data and specifically from the in-silico simulation of synthetic genomics that has been evolving for almost 30 years now [12] alongside sequencing technologies [13].

In this work we present synth4bench, a bioinformatics framework, in an effort to observe the behavior of variant calling algorithms. To this end, we generated a set of synthetic next-generation sequencing (NGS) datasets, strategically derived from the intricate genomic composition of the TP53 gene. At its heart lies NEATv3.3 simulator (NExt-generation sequencing Analysis Toolkit) [14] that was utilized to produce these datasets. Varying values of intrinsic parameters of NGS data were used in order to study the behavior of different variant callers and their response to these changes.

We focused primarily on intrinsic parameters such as sequencing depth, read length and position within each read. We benchmarked tumor-only somatic variant callers and supported our findings with independent statistical analyses. Our results demonstrate that our framework significantly assists in the process of variant reporting by shedding light on the behavior and decisions of the involved algorithms and outlines their patterns as they are presented with datasets generated based on different NGS-related parameters.

2. Experimental Design and Methods

Our framework takes as input a reference human genome, the NGS-related parameters as set by the user and outputs the comparison between each variant caller and the ground truth. The in-between steps include the preprocessing required before moving to the variant calling process, as well as the downstream analysis of reported variants that is necessary to extract the required information (Fig. 1).

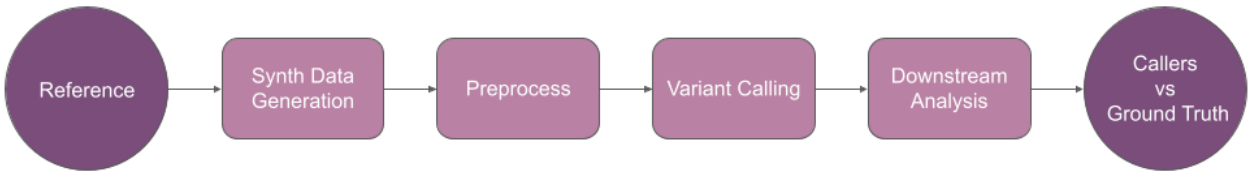

Fig. 1: Schematic representation of synth4bench. The circles represent the input and output files, while the rectangles represent the relevant processes.

2.1 Synthetic Data Generation

Due to its central role within our framework, the generation of synthetic human genomics was a task of high importance. The selected simulator was NEATv3.3, due to its adequate precision [13] as well as flexibility, due to its wide list of available parameters compared to other simulating algorithms [15]. A list of the parameters of NEATv3.3 and a short description are presented in Table 1. The reference human genome used during the data generation was hg38.

The primary data source for this endeavor was the TP53 gene. It is located on the short arm (p) of chromosome 17 and comprises 11 exons, of which 10 are coding [16]. The TP53 gene was strategically chosen for its variability in terms of variants as well as its significance in cancer genomics. This significance is highlighted by the high frequency of variants in a large series of patients with different types of cancer [17]. Furthermore, the encoded p53 protein is involved in tumor suppression through its activation after stress signals and crucial role in various cellular responses, such as DNA repair and control of cell proliferation [18]. For the generation of all datasets the parameter to determine a seed was used to ensure reproducibility.

For the needs of our task, we wanted to achieve consistently low frequencies, from 10% and below (i.e. AF<10%). In order to achieve this, a number N of individual datasets with the same NGS characteristics, each of coverage x were produced. Afterwards, these individual files were merged into a single file, yielding a final coverage X equal to:

$$X = N \cdot x. \tag{1}$$

On the contrary, the AF of each individual file is inversely related to the coverage of the final merged file. If the coverage and AF of a variant in one of the individual files is x and f respectively then the new AF, F of the same variant in the merged file will become:

$$F = \frac{f}{N \cdot x} = \frac{f}{X}. \tag{2}$$

Table 1: A subset of NEATv3.3 input parameters, their corresponding description taken from the documentation and their given value.

Parameter	Description	Value
coverage	Average coverage	300-5000
read length	The desired read length	50-300
avg mut rate	Rescale avg mutation rate to this (1/bp)	0.1
ploidy	Desired ploidy	2
rng	rng seed value	varying
pe	Paired-end fragment length mean and std	(300, 30)

After running a number of experiments to determine the number of individual files N, we concluded that the optimal value in our case is 10 because it helps reduce the overall AFs. Given that Eq. (1) now becomes $X = 10x$ and Eq. (2) becomes $F = \frac{f}{10x}$. The schematic representation of this methodology is shown in Fig. 2. The AF of a variant was originally equal to 100% in one of the individual files (i.e. one read with a yellow triangle) and, then, became 10% in the final merged file (group of a total of 10 reads). In practice, there is usually an overlap among the variants from the individual files. Different variants from the individual files end up in the same chromosomal position. In such cases if the variants happen to be similar, their AF are added. If the variants are different then we treat them as background noise.

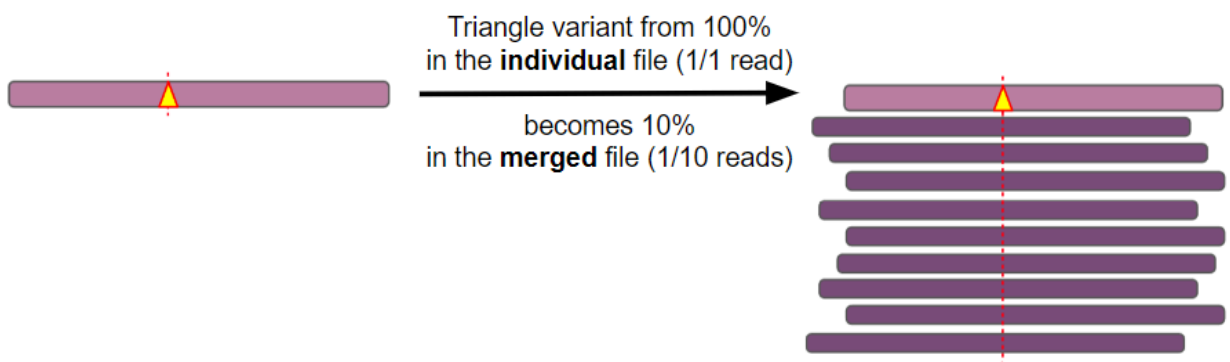

Fig. 2: Schematic representation of the methodology used to achieve consistently low frequencies within our ground truth datasets. When merging two individual groups of reads, their corresponding AF is readjusted in the final combined group of reads.

2.2 Exploration of genomic data feature space

Coverage. It has been observed that errors occur during the NGS process. One approach to overcome these errors and ensure the accurate base calling of each nucleotide is by increasing the coverage [19]. Coverage refers to the mean count of sequencing reads covering each base of the sequenced DNA fragments. For each

variant position, a minimum coverage of ≥1000× is recommended, with at least 50 reads for the allelic variant [20]. For the purposes of our study, the coverage values that were chosen were 300x, 700x, 1000x, 3000x and 5000x, in order to assess in-depth the behavior of variant callers.

Reads related parameters. As described by Illumina, a fragment consists of the insert (i.e. the DNA region of interest) together with the adapters, while a read refers to the sequenced part of a fragment. Furthermore, the recommended type of sequencing is that of paired-end (i.e. every DNA fragment is read from both ends) since it generates twice the number of sequencing reads and also facilitates more accurate read alignment. In order to minimize the risk of introducing false-positive results, it is recommended to ensure an overlap between the paired reads [21]. To explore this NGS data related parameter the following read lengths were put to test: 50, 75, 100, 150 and 300 nucleotides (nt). This approach was applied to cover all possible scenarios for a given fragment (in our case with a length of 300 nt). Although some of the given values are rarely found in wet lab experiments, one can easily test these cases when following the synthetic path.

2.3 Variant Calling Pipeline

One of the main pillars of the process of identifying genetic variants is that of accurately mapping reads to the reference using alignment tools. Thanks to the fact that NEATv3.3 produces aligned reads, we were able to skip the crucial alignment step that can be a source of errors, and focus merely on the variants' detection, presented directly in the Binary Alignment Map (BAM) format.

Next, all individual files from each independent run, both BAM and Variant Call Format (VCF) files, were combined (10 per run) into one merged file that served as ground truth as described in Section 2.1.

During the preprocessing phase, some steps were common for all callers, such as indexing. On the other hand, each variant caller has its own demands (for example Mutect2 requires the name of each sample to be also present within the BAM file). All these needed steps, both common and individual, were incorporated in our framework using SAMtools [22].

After the BAM files were formatted, the testing of the somatic variant callers took place. We chose those that fall under the category of non-matched, meaning that they can operate without the requirement of a matched normal sample file. The list of these callers included Mutect2 [23], FreeBayes [24], VarDict [25], VarScan2 [26] and LoFreq [27]. A candidate that was ultimately left out was Strelka2 [28], since it operates only in matched tumor-normal mode. One by one, the callers were presented with the same file for each dataset and each one produced a VCF file that included the identified variants.

Another major issue that emerged during the comparison of VCF files was the lack of a cohesive representation method for the identified variants, especially for indels. Different variant callers reported the same indels, but in a different way, thus hindering the comparison of their call sets [29]. The solution to this challenge involved the practice of indel normalization, where each variant was normalized based on two conditions; each variant was reported using the smallest base position possible, i.e. left-aligned, and the length of each reported variant was the shortest, i.e. parsimonious. All VCF files

produced from all variant callers were processed with BCFtools [22] to make sure everything was normalized and comparable.
Afterwards, the package bam-readcount [30] was implemented to acquire low-level information for all nt positions in the ground truth BAM files to compare the reported variants from the variant callers.
The final downstream analysis of our framework was incorporated in a series of R scripts. First of all, the variants with AF $f = 100\%$ in the ground truth individual BAM files were identified and their corresponding AF F in the ground truth merged file was calculated. These results were compared with the VCF file produced by each caller to assess the level of their performance. Some data reformatting was performed at this stage, to render all information in a common format to facilitate further analysis. In order to present our findings in the most informative way, two different methods of visualization were implemented. These included: (i) barplots of the coverage and AF and (ii) density plots of the AF deviation of the variants. All these visualizations depicted the comparison of the ground truth against the output of each variant caller.

2.4 Statistical Analysis

For descriptive statistics, frequencies and relative frequencies were used for categorical variables, whereas the median and interquartile ranges (IQR) were used for the numeric variables. Grouped barplots were generated for data visualization.
Regarding the inferential statistics, the primary aim was to examine the impact of the coverage and the read length on the capacity of each caller to detect variants. The secondary aim was to examine the exposure variables that are associated with this capacity, within a given coverage and read length.
To this end, scatterplots of the modified accuracy (mAC) shown in Eq. (3) with the coverage and the read length were constructed for the primary aim, and the Kendall rank correlation coefficient was applied, along with the p-value. For the secondary aim, we conducted both univariable and multivariable analyses. The exposure variables used in those analyses were the ground truth alterations (ALT) and reference (REF), the chromosomal position, the position within reads, and the number of sequences. The first two are categorical variables (with values A, G, T and C), whereas the rest are numeric.
It should be mentioned here that for this type of analysis there is no data in the True Negative (TN) group. In our case TN data would be translated to “those variants that were not present in the ground truth and were also not present in the caller’s report”. For the purposes of our study, we did not take into account the TN, thus redefining the AC as shown in Eq. (3):

$$mAC = \frac{TP}{TP + FP + FN} \ . \tag{3}$$

Concerning the univariable analysis, chi-squared and Fisher’s exact test (if necessary) were used to test the association of the secondary aim with the categorical exposure variables, whereas logistic regression was applied for the numeric variables. To perform bias reduction, Firth’s penalty was also applied in the logistic regression model. Furthermore, the Youden index was maximized to obtain the optimal cut-off for the numeric exposure variables for discriminating the primary aim. To better understand the behavior of each numeric exposure variable, we also displayed their distributions

between the two classes of the secondary outcome and constructed the receiver operating characteristic (ROC) curves.
As for the multivariable analyses, logistic regression was conducted, including all the above-mentioned exposure variables. For each analysis, two models were adjusted; one having only the main effects and one including the interaction terms as well. In order to select which interaction terms would be inserted in the model, we first constructed grouped (in terms of the secondary aim) barplots of each numeric exposure within a categorical variable. Hence, we excluded the interaction terms in the case that the association between the numeric exposure and the secondary aim remained relatively constant within the categorical exposure. To further select the significant variables for the secondary outcome, the Lasso penalty was applied in each model. The optimal tuning parameter was selected with cross-validation. Odds ratio (OR) was used to report the effect of each regression coefficient, along with their 95% Wald's confidence intervals. The performance of each model was evaluated through the area under the curve (AUC).

3. Results

In this section the results from our experiments are presented. All synthetically generated datasets are shown in subsection 3.1. For the exploration of the behavior of callers we focused on two aspects; the effect of the different NGS parameters on the detection of variants and the aspects of the behavior of each caller. These are presented in subsection 3.2 and discussed in section 4. Furthermore, the results from the independent statistical analysis are presented in subsection 3.3. Finally, some technical aspects about run time and memory usage are presented in subsection 3.4.

3.1 Synthetic Datasets

The produced datasets are presented in Table 2. For the exploration of the NGS parameters effect on variant calling 10 datasets were produced, 5 with different coverage (while keeping all other parameters fixed) and 5 with varying read length. Additionally, 4 datasets were created to test the effect of the individual files on the final AF in the merged file, as described in subsection 2.1.

Table 2: List of synthetically generated datasets. The first column contains the number of runs that resulted in a new dataset. The second column contains the parameter under study, while the third column contains its value.

Number of Datasets	Varying Parameter	Given Values
5	coverage	300-700-1000-3000-5000
9	read length	25-50-75-100-150-170-200-250-300
4	number of individual files	1-2-4-10

3.2 Data Feature Space Exploration

This subsection concerns the results of each variant caller for each dataset. Tables 3 and 4 show the level of agreement of each algorithm with the ground truth. Specifically, there are three groups of variants; those that are present only in the ground truth but not reported by the variant caller, i.e. the False Negative (FN), those that are present in the ground truth and also reported from the variant callers i.e. their overlap or True Positive (TP) and, finally, those that are only reported by the variant caller but do not exist in the ground truth i.e. False Positive (FP). All values are given in percentages, while the highest TP value of each caller is outlined.

Table 3: Results regarding the impact of coverage in all variant callers. All values for FN, TP and FP variants are given in percentages. For each caller the highest TP value is given in bold.

		Coverage				
Caller	**Variants**	**300x**	**700x**	**1000x**	**3000x**	**5000x**
Mutect2	FN	61.1	55.9	54.9	55.7	56.5
	TP	35.3	40.1	**40.9**	40.1	39.1
	FP	3.7	3.9	4.2	4.2	4.4
Freebayes	FN	41.1	35.9	32.7	26.1	22.6
	TP	**20.9**	16.2	13.2	3.0	0.6
	FP	38.0	47.9	54.1	70.9	76.8
VarDict	FN	20.2	19.0	18.5	18.6	18.6
	TP	66.1	**66.4**	66.2	63.3	63.4
	FP	13.8	14.6	15.3	18.0	18.0
VarScan2	FN	15.7	15.4	15.1	15.5	15.7
	TP	52.3	52.5	**52.8**	52.5	52.5
	FP	32.1	32.1	32.1	32.0	31.8
LoFreq	FN	8.2	3.1	2.8	1.8	1.7
	TP	91.7	96.9	97.1	98.2	**98.3**
	FP	0.1	0.1	0.1	0.1	0.1

Table 4: Results regarding the impact of read length in all variant callers. All values for FN, TP and FP variants are given in percentages. The highest TP value of each caller is given in bold.

		Read Length				
Caller	**Variants**	**50**	**75**	**100**	**150**	**300**
Mutect2	FN	43.2	47.1	48.6	54.9	64.9
	TP	**50.7**	47.1	45.7	40.9	31.8
	FP	6.1	5.8	5.7	4.2	3.3
Freebayes	FN	32.4	33.1	99.8	32.7	33.4
	TP	**13.3**	12.9	0.2	13.2	13.0
	FP	54.3	54.0	0.0	54.1	53.6
VarDict	FN	18.3	18.7	18.0	18.5	19.4
	TP	66.6	65.8	**66.9**	66.2	65.1
	FP	15.1	15.5	15.1	15.3	15.5
VarScan2	FN	15.2	15.3	15.4	15.1	15.7
	TP	52.7	52.6	52.4	**52.8**	52.4
	FP	32.1	32.1	32.1	32.1	31.9
LoFreq	FN	2.7	2.6	2.6	2.8	3.5
	TP	97.2	**97.4**	**97.4**	97.1	96.4
	FP	0.1	0.1	0.0	0.1	0.1

We define the deviation of the AF as the difference between the AF of the reported variants from each caller minus the AF of the ground truth as shown in Eq. (4).

$$\Delta AF = AF_{Caller} - AF_{GT} \tag{4}$$

It becomes apparent that the ideal scenario is when the deviation of the AF equals zero, $\Delta AF = 0$ which can be achieved when the reported AF from the callers is the same as the one of the ground truth. On the other hand, $\Delta AF > 0$ when the detected variants from the callers are falsely given higher AF than the real values present in the ground truth. Finally, $\Delta AF < 0$ when variants are falsely reported by the callers with lower AF. It should be noted that this comparison can be only applied in the TP variants; in the case

of the FP $AF_{GT} = 0$ and for the FN $AF_{Caller} = 0$. The overall study of the callers' response to the various values of the observed parameters is depicted in Fig. 4 for one of the datasets. The other multipanel figures for the other datasets from the ablation study are included in the Supplementary Material.

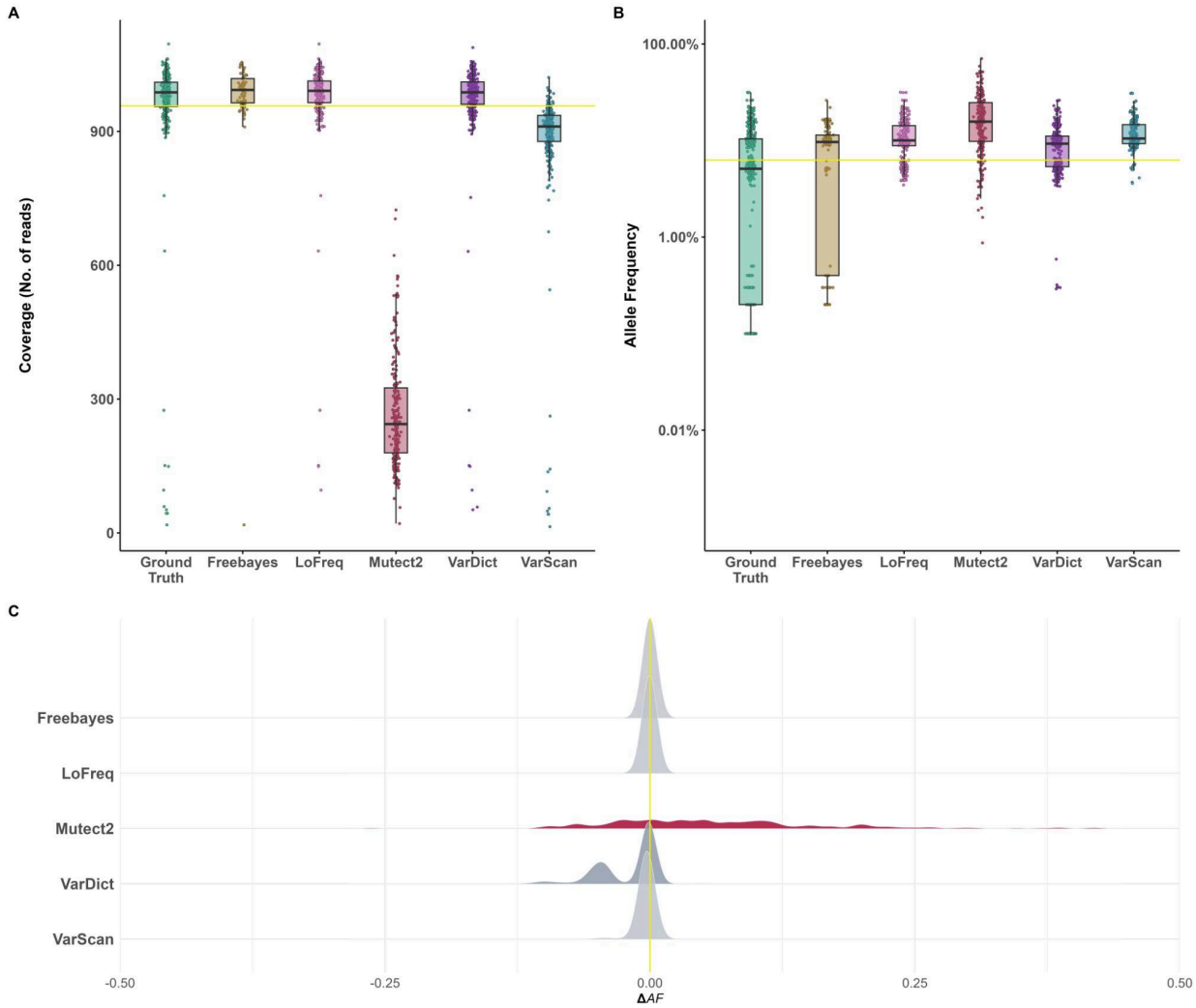

Fig. 4 A multipanel figure showing the behavior of all variant callers for SNVs for a given dataset (1000x coverage and 150 read length). **A** The coverage barplots of ALT for the ground truth and all TP variants (those present in the ground truth that were also reported from a specific algorithm) for all callers. The yellow line marks the mean value of the coverage in the ground truth. **B** The corresponding AF barplots for the same TP variants of each scenario. Note that the y axis is logarithmic to better outline the distribution of the order of magnitude of the variants' AF. The yellow line marks the mean value of the AF in the ground truth. **C** The density plots of the AF deviation $\Delta AF = AF_{Caller} - AF_{GT}$ for all TP variants. The distributions are near 0.00 when the callers' AF report is very close to the ground truth and they deviate when their reports are off according to the equation of ΔAF .

3.3 Statistical Analysis

For Freebayes, an extremely high and negative correlation was displayed between the mAC and the coverage (tau = -1, p-value = -0.017), whereas no correlation was observed between the mAC and the read length (tau = -0.2, p-value = 0.82). These findings indicate that the performance of Freebayes decreased as the coverage increased. In contrast, Mutect2 showed no correlation between the mAC and the coverage (tau = 0, p-value > 0.99), but an extremely high and negative correlation with read length (tau = -1, p-value = 0.017). Hence, the performance of Mutect2 improved as the read length decreased. VarDict and VarScan2 showed no correlation with either the coverage or the read length, whereas LoFreq displayed an extremely high and positive correlation with coverage (tau = 1, p-value = 0.017), but no correlation with read length (tau = -0.4, p-value = 0.48). Consequently, the performance of LoFreq increased as the coverage increased.

Regarding the secondary aim, which concerned the examination of the exposure variables that are associated with the callers' capacity for variant report for given coverage and read length values, although for each dataset some exposure variables displayed a statistically significant correlation with the primary outcome, the AUC was always relatively low as seen in the case ofLoFreq in Fig. 5. Besides, the significant exposure variables and the directions of their relationship with the secondary aim were not stable among datasets. Consequently, no specific association of any of the exposure variables with the primary aim prevailed. The ROC curve figures for the other callers are included in the Supplementary Material in Supplementary Figures 9-12.

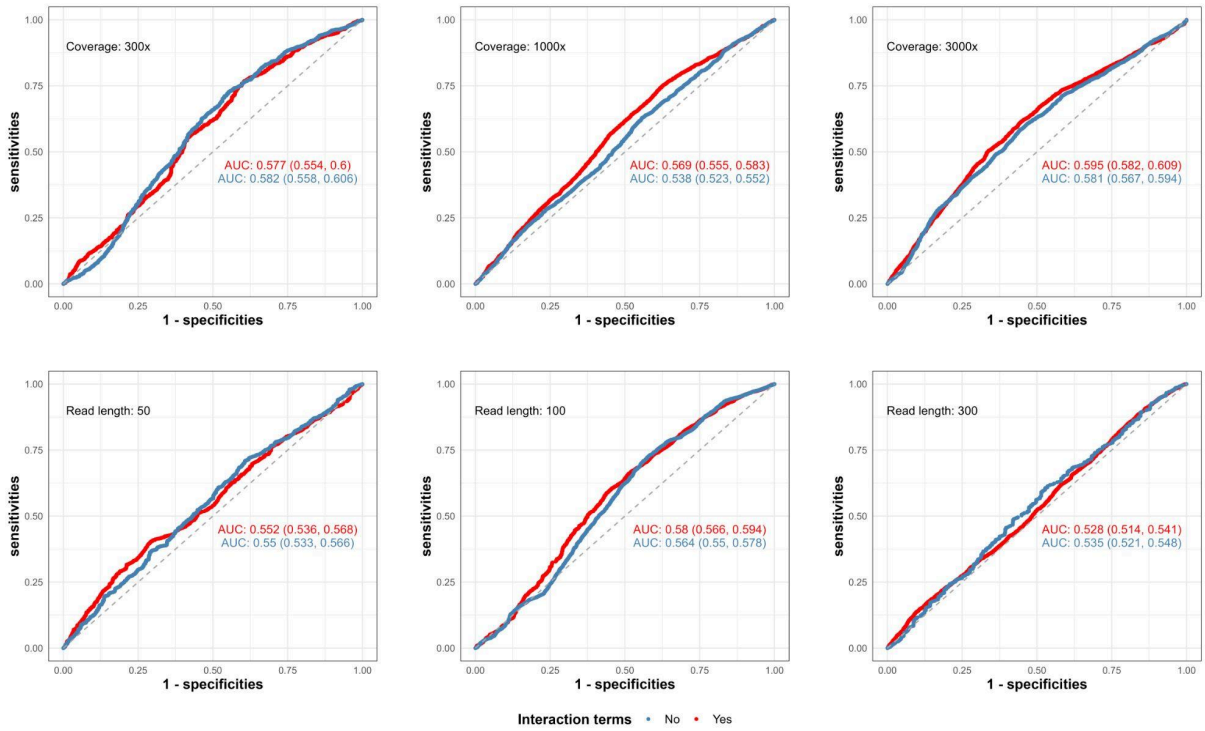

Fig. 5: ROC curves obtained from the logistic models adjusted for LoFreq. *Upper panel* the ROC curves for the statistical analysis regarding coverage. The datasets of 300x, 100x and 3000x and read length 150 are depicted. *Lower panel* the ROC curves for the

statistical analysis regarding read length. All datasets have coverage 1000x and those with length 50, 100 and 300 are depicted.

3.4 Run time

An indicative time measurement of synth4bench was performed to produce a dataset of x5000 coverage, which contained a total of 620,642 reads. Specifically the generation part lasted for about 6 minutes and for the variant calling algorithms the time that elapsed was 50 minutes for Mutect2, 2 minutes for Freebayes, 12 minutes for VarDict, 3 minutes for VarScan and 12 minutes for LoFreq. All experiments were performed on a desktop equipped with an Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz and 32 GB of RAM.

4. Discussion

In this article we present a framework, synth4bench, for benchmarking somatic variant callers through the use of synthetic genomics data as ground truth. Our approach showed that callers do not behave similarly due to their algorithmic nature that weighs the involved components differently. Synth4bench has been developed to shed light on the underlying principles that govern the performance of each variant caller by analyzing datasets across the NGS feature space and observing their response to the range of the applied parameters.

In terms of the impact of coverage, depicted in Table 3, LoFreq had the highest percentages of TP accompanied with the lowest FN and FP, followed by VarDict and VarScan. All three of them showed stability when presented with various values of coverage, leading to minor changes in their TP (range of 0.5-6.6%). On the other hand, Freebayes had the lowest overlap with the ground truth datasets and also the biggest variation in its performance. Mutect2 displayed a more stable overlap, of around 40%, for larger values of coverage. The overall callers' behavior in relation to the varying coverage was not consistent; LoFreq exhibited the highest overlap, but this was not the case for the rest of the callers. For example, Freebayes achieved the highest overlap for the dataset with the lowest coverage, i.e. 300x, as mentioned in subsection 3.3.

For the exploration of read length, depicted in Table 4, LoFreq was the caller with the highest overlap in all scenarios, again followed by VarDict and VarScan. In this case, Mutect2 was not so consistent, displaying a range of about 20%. Surprisingly, Mutect2 and Freebayes operated better when analyzing datasets with smaller read lengths. On the other hand, VarDict and VarScan provided the best results for larger read lengths (of 100 and 150 nt, respectively). LoFreq displayed its best performance when dealing with read lengths of 75 and 100 nt.

Regarding the duration that is required for each algorithm to operate, as presented in subsection 3.4, Mutect2 took the longest to run. On the other hand, although Freebayes and VarDict needed the same time, i.e a couple of minutes to process 620,642 reads, the former detected very few variants compared to the latter. VarScan had an overlap of around 50% with the ground truth and needed only 3 minutes for the whole process. Finally, LoFreq ran in 12 minutes for the given dataset and yielded the best results in terms of TP variant detection.

After comparing the performance of all variant callers, a few observations were evident. In regard to Mutect2, there was a systematic tendency for downsampling, depicted in Fig. 4 as well as in Supplementary Figures 1-8, which also affected the reported AF leading in great deviation from the ground truth. We believe that although downsampling is quite useful computationally, it can lead to misrepresentation of information, when not done appropriately. Freebayes showed the highest deviation in its behavior for both ablation studies of the coverage and the read length, as well as the lowest agreement with the ground truth. As far as VarDict and VarScan are concerned, they both demonstrated a more stable behavior for all datasets, yet VarDict had a higher overlap with the ground truth compared to VarScan as well aslower FP percentages. Finally, LoFreq seems to be able to handle adequately variants at low frequencies, since it outperformed all other callers for the given datasets with low AF.
Overall, synth4bench leverages the effort of benchmarking algorithms by offering the opportunity to utilize a generated ground truth dataset as reference for comparisons. Our framework can be of great importance in the field of cancer genomics, where precision is an ultimate necessity, especially in the low frequency domain, because it makes the comparison of different algorithms transparent, straightforward and it also enhances their comparability.

5. Conclusion and Future Work

Our next objectives concern the refinement and extension of the presented framework, focusing on both the synthetic data generation process and the variant calling methodology. Our main aim for the future is to explore more sophisticated algorithms from the domain of machine learning, such as generative adversarial networks (GANs). Regarding the data generation process, we plan to explore more components that may influence the results. This exploration entails an examination of additional technical parameters, such as the quality of the synthesized NGS data, that may influence the performance of variant callers.
Furthermore, there is the aspect of the quality of synthetic data. There are various types of evaluation metrics with respect to fidelity, privacy and applicability. On this matter there is a well organized effort by the ELIXIR Machine Learning Focus Group [31], which deals with this important issue. Finally, we aim at implementing a workflow system such as Nextflow [32] to ensure reproducibility of our framework and a more user friendly interface.
Regarding a more philosophical aspect of the usage of synthetic data, although they are very popular in life sciences, there are some aspects that should be carefully considered. Synthetic data are excellent both for the development of new tools as well as for benchmarking existing tools; they are easy to produce or acquire and can be implemented into extreme or peculiar scenarios that would be expensive and/or impossible to achieve in a wet lab. However, our actual aim is to test our hypothesis against real data, since they are the only true indicator of a well educated methodology.

Conflict of interest

No conflicts of interest to declare.

Funding

This work has been written with the support of the EOSC4Cancer project, funded by the Horizon Europe Framework Programme under grant agreement number 101058427. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them. This work was also supported by ELIXIR, the research infrastructure for life-science data.

References

1. Zhang L, Jia Z, Mao F *et al.* Whole-exome sequencing identifies a somatic missense mutation of NBN in clear cell sarcoma of the salivary gland. *Oncol Rep* 2016;**35**:3349–56.
2. Federici G, Soddu S. Variants of uncertain significance in the era of high-throughput genome sequencing: a lesson from breast and ovary cancers. *J Exp Clin Cancer Res* 2020;**39**:46.
3. Krøigård AB, Thomassen M, Lænkholm A-V *et al.* Evaluation of Nine Somatic Variant Callers for Detection of Somatic Mutations in Exome and Targeted Deep Sequencing Data. Jordan IK (ed.). *PLOS ONE* 2016;**11**:e0151664.
4. Sandmann S, De Graaf AO, Karimi M *et al.* Evaluating Variant Calling Tools for Non-Matched Next-Generation Sequencing Data. *Sci Rep* 2017;**7**:43169.
5. He X, Chen S, Li R *et al.* Comprehensive fundamental somatic variant calling and quality management strategies for human cancer genomes. *Brief Bioinform* 2021;**22**:bbaa083.
6. Trevarton AJ, Chang JT, Symmans WF. Simple combination of multiple somatic variant callers to increase accuracy. *Sci Rep* 2023;**13**:8463.
7. ICGC-TCGA DREAM Somatic Mutation Calling Challenge participants, Ewing AD, Houlahan KE *et al.* Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nat Methods* 2015;**12**:623–30.
8. Cantarel BL, Weaver D, McNeill N *et al.* BAYSIC: a Bayesian method for combining sets of genome variants with improved specificity and sensitivity. *BMC Bioinformatics* 2014;**15**:104.
9. Anzar I, Sverchkova A, Stratford R *et al.* NeoMutate: an ensemble machine learning framework for the prediction of somatic mutations in cancer. *BMC Med Genomics* 2019;**12**:63.
10. Ainscough BJ, Barnell EK, Ronning P *et al.* A deep learning approach to automate refinement of somatic variant calling from cancer sequencing data. *Nat Genet* 2018;**50**:1735–43.
11. Majidian S, Agustinho DP, Chin C-S *et al.* Genomic variant benchmark: if you cannot measure it, you cannot improve it. *Genome Biol* 2023;**24**:221.
12. Engle ML, Burks C. Artificially Generated Data Sets for Testing DNA Sequence Assembly Algorithms. *Genomics* 1993;**16**:286–8.
13. Alosaimi S, Bandiang A, van Biljon N *et al.* A broad survey of DNA sequence data simulation tools. *Brief Funct Genomics* 2020;**19**:49–59.
14. Stephens ZD, Hudson ME, Mainzer LS *et al.* Simulating Next-Generation Sequencing Datasets from Empirical Mutation and Sequencing Models. Parkinson J (ed.). *PLOS ONE* 2016;**11**:e0167047.
15. Milhaven M, Pfeifer SP. Performance evaluation of six popular short-read simulators. *Heredity* 2023;**130**:55–63.
16. Catherwood MA, Gonzalez D, Donaldson D *et al.* Relevance of TP53 for CLL diagnostics. *J Clin Pathol* 2019;**72**:343–6.
17. Levine AJ. The many faces of p53: something for everyone. *J Mol Cell Biol* 2019;**11**:524–30.
18. Levine AJ. p53: 800 million years of evolution and 40 years of discovery. *Nat Rev Cancer*

2020;**20**:471–80.
19. Sims D, Sudbery I, Ilott NE *et al.* Sequencing depth and coverage: key considerations in genomic analyses. *Nat Rev Genet* 2014;**15**:121–32.
20. Pandzic T, Ladenvall C, Engvall M *et al.* Five Percent Variant Allele Frequency Is a Reliable Reporting Threshold for TP53 Variants Detected by Next Generation Sequencing in Chronic Lymphocytic Leukemia in the Clinical Setting. *HemaSphere* 2022;**6**:e761.
21. Malcikova J, Tausch E, Rossi D *et al.* ERIC recommendations for TP53 mutation analysis in chronic lymphocytic leukemia—update on methodological approaches and results interpretation. *Leukemia* 2018;**32**:1070–80.
22. Danecek P, Bonfield JK, Liddle J *et al.* Twelve years of SAMtools and BCFtools. *GigaScience* 2021;**10**:giab008.
23. Cibulskis K, Lawrence MS, Carter SL *et al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat Biotechnol* 2013;**31**:213–9.
24. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. 2012, DOI: 10.48550/arXiv.1207.3907.
25. Lai Z, Markovets A, Ahdesmaki M *et al.* VarDict: a novel and versatile variant caller for next-generation sequencing in cancer research. *Nucleic Acids Res* 2016;**44**:e108–e108.
26. Koboldt DC, Zhang Q, Larson DE *et al.* VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res* 2012;**22**:568–76.
27. Wilm A, Aw PPK, Bertrand D *et al.* LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Res* 2012;**40**:11189–201.
28. Kim S, Scheffler K, Halpern AL *et al.* Strelka2: fast and accurate calling of germline and somatic variants. *Nat Methods* 2018;**15**:591–4.
29. Tan A, Abecasis GR, Kang HM. Unified representation of genetic variants. *Bioinformatics* 2015;**31**:2202–4.
30. Khanna A, Larson D, Srivatsan S *et al.* Bam-readcount - rapid generation of basepair-resolution sequence metrics. *J Open Source Softw* 2022;**7**:3722.
31. Machine Learning Focus Group. *ELIXIR*.
32. Di Tommaso P, Chatzou M, Floden EW *et al.* Nextflow enables reproducible computational workflows. *Nat Biotechnol* 2017;**35**:316–9.