






RESEARCH ARTICLE

NCBench: providing an open, reproducible, transparent, adaptable, and continuous benchmark approach for DNA-sequencing-based variant calling [version 1; peer review: 2 approved with reservations]

Friederike Hanssen¹, Gisela Gabernet¹, Nicholas H. Smith ², Christian Mertes²⁻⁴, Avirup Guha Neogi⁵, Leon Brandhoff^{5,6}, Anna Ossowski⁵, Janine Altmueller^{5,7,8}, Kerstin Becker⁵, Andreas Petzold⁹, Marc Sturm¹⁰, Tyll Stöcker¹¹, Sugirthan Sivalingam¹², Fabian Brand¹³, Axel Schmid¹⁴, Andreas Bunes¹⁵, Alexander J. Probst¹⁶, Susanne Motameny ^{5,6}, Johannes Köster ^{17,18}

¹Quantitative Biology Center, Eberhard Karls University Tübingen, Tübingen, Germany

²TUM School of Computation, Information and Technology, Technical University of Munich, Munich, Germany

³Munich Data Science Institute, Technical University of Munich, Munich, Germany

⁴Institute of Human Genetics, Klinikum rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany

⁵Cologne Center for Genomics, University of Cologne, Cologne, Germany

⁶West German Genome Center - Cologne, University of Cologne, Cologne, Germany

⁷Core Facility Genomics, Berlin Institute of Health at Charité - Universitätsmedizin Berlin, Berlin, Germany

⁸Max Delbrück Center for Molecular Medicine in the Helmholtz Association (MDC), Berlin, Germany

⁹DRESDEN-concept Genome Center, TUD Dresden University of Technology, Dresden, Germany

¹⁰Institute of Medical Genetics and Applied Genomics, University Hospital Tuebingen, Tübingen, Germany

¹¹Institute of Crop Science and Resource Conservation, University of Bonn, Bonn, Germany

¹²Institute of Human Genetics, Medical Faculty and University Hospital Düsseldorf, Heinrich-Heine-University Düsseldorf, Düsseldorf, Germany

¹³Institute for Genomic Statistics and Bioinformatics, Medical Faculty, University of Bonn, Bonn, Germany

¹⁴Institute of Human Genetics, University Hospital of Bonn, Bonn, Germany

¹⁵Core Unit for Bioinformatics Analysis, University Hospital Bonn, Bonn, Germany

¹⁶Environmental Metagenomics, Research Center One Health Ruhr, University Alliance Ruhr, Faculty of Chemistry, University of Duisburg-Essen, Essen, Germany

¹⁷German Cancer Consortium, Essen, Germany

¹⁸Bioinformatics and Computational Oncology, Institute for Artificial Intelligence in Medicine (IKIM), University Medicine Essen, University of Duisburg-Essen, Essen, Germany

V1 First published: 11 Sep 2023, 12:1125
<https://doi.org/10.12688/f1000research.140344.1>

Latest published: 11 Sep 2023, 12:1125
<https://doi.org/10.12688/f1000research.140344.1>


Abstract

We present the results of the human genomic small variant calling benchmarking initiative of the German Research Foundation (DFG) funded Next Generation Sequencing Competence Network (NGS-CN) and the German Human Genome-Phenome Archive (GHGA).

In this effort, we developed NCBench, a continuous benchmarking

Open Peer Review

Approval Status  

	1	2
version 1 11 Sep 2023	 view	 view

platform for the evaluation of small genomic variant callsets in terms of recall, precision, and false positive/negative error patterns. NCBench is implemented as a continuously re-evaluated open-source repository.

We show that it is possible to entirely rely on public free infrastructure (Github, Github Actions, Zenodo) in combination with established open-source tools. NCBench is agnostic of the used dataset and can evaluate an arbitrary number of given callsets, while reporting the results in a visual and interactive way.

We used NCBench to evaluate over 40 callsets generated by various variant calling pipelines available in the participating groups that were run on three exome datasets from different enrichment kits and at different coverages.

While all pipelines achieve high overall quality, subtle systematic differences between callers and datasets exist and are made apparent by NCBench. These insights are useful to improve existing pipelines and develop new workflows.

NCBench is meant to be open for the contribution of any given callset. Most importantly, for authors, it will enable the omission of repeated re-implementation of paper-specific variant calling benchmarks for the publication of new tools or pipelines, while readers will benefit from being able to (continuously) observe the performance of tools and pipelines at the time of reading instead of at the time of writing.

Keywords


continuous, benchmarking, NGS, variant calling



This article is included in the [Bioinformatics gateway](#).



This article is included in the [Genomics and Genetics gateway](#).

1. **Justin M. Zook** , National Institute of Standards and Technology (NIST), Gaithersburg, USA

2. **Kez Cleal**, Cardiff University, Cardiff, UK

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Johannes Köster (johannes.koester@uni-due.de)

Author roles: **Hanssen F:** Conceptualization, Data Curation, Resources, Writing – Original Draft Preparation, Writing – Review & Editing; **Gabernet G:** Conceptualization, Data Curation, Resources, Writing – Original Draft Preparation, Writing – Review & Editing; **Smith NH:** Data Curation, Resources, Writing – Review & Editing; **Mertes C:** Conceptualization, Data Curation, Resources, Writing – Review & Editing; **Neogi AG:** Data Curation, Resources, Writing – Review & Editing; **Brandhoff L:** Data Curation, Resources, Writing – Review & Editing; **Ossowski A:** Data Curation, Resources, Writing – Review & Editing; **Altmueller J:** Data Curation, Resources, Writing – Review & Editing; **Becker K:** Data Curation, Resources, Writing – Review & Editing; **Petzold A:** Conceptualization, Writing – Review & Editing; **Sturm M:** Conceptualization, Data Curation, Resources, Writing – Review & Editing; **Stöcker T:** Data Curation, Resources, Writing – Review & Editing; **Sivalingam S:** Data Curation, Resources, Writing – Review & Editing; **Brand F:** Data Curation, Resources, Writing – Review & Editing; **Schmid A:** Data Curation, Resources, Writing – Review & Editing; **Buness A:** Conceptualization, Data Curation, Resources, Writing – Review & Editing; **Probst AJ:** Conceptualization, Writing – Review & Editing; **Motameny S:** Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Köster J:** Conceptualization, Formal Analysis, Investigation, Methodology, Project Administration, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: NHS and CM are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) via the project NFDI 1/1 "GHGA - German Human Genome-Phenome Archive" (\#441914366). SM is supported by the DFG via the project "West German Genome Center" (\#407493903). The other authors declare that no grants were involved in supporting this work. *The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

Copyright: © 2023 Hanssen F *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Hanssen F, Gabernet G, Smith NH *et al.* **NCBench: providing an open, reproducible, transparent, adaptable, and continuous benchmark approach for DNA-sequencing-based variant calling [version 1; peer review: 2 approved with reservations]** F1000Research 2023, 12:1125 <https://doi.org/10.12688/f1000research.140344.1>

First published: 11 Sep 2023, 12:1125 <https://doi.org/10.12688/f1000research.140344.1>

Introduction

Genome sequencing is integral to many research and diagnostic procedures. For both pipeline and tool development, it is crucial to ensure that genomic variant calls are as accurate as possible. This can be achieved by testing tools and pipelines on datasets with a known set of true variants and correspondingly known sites where the genome is the same as the reference genome.

Several such benchmark datasets have been published. The Genome in a Bottle Consortium (GIAB) has released truth variant sets based on common calls across three variant callers on 14 different sequencing technologies and library preparation methods on a well-characterized genome (HG001 or NA12878), as well as an Ashkenazim trio (HG002-4) and a Han Chinese trio (HG005-7).^{1,2} The Platinum variant catalog provides consensus calls of six variant calling pipelines across two different sequencing platforms on a family of four grandparents, two parents and 11 children including the NA12878 genome, allowing an extended inheritance-based validation.³ In an alternative approach, Li et al.⁴ generated a synthetic diploid from two complete hydatidiform mole (CHM) cell lines (CHM1 and CHM13), which are almost completely homozygous across the whole genome, such that the known variants in this set are phased (their haplotype of origin is known). The synthetic diploid benchmark dataset has the advantage of not relying on a consensus callset across several variant callers, which limit the benchmark set to high-confidence regions and lead to an over-estimation of the true variant calling performance. Finally, the SEQC2/MAQC-IV initiative provides another extensive set of validated benchmarks, not only focussing on genomic DNA but also considering RNA-seq and single-cell sequencing.⁵

Several publications have utilized the aforementioned gold-standard callsets to benchmark variant calling tools and pipelines.⁶⁻⁹ However, the continuous development of variant calling tools and pipelines means that static, one-time benchmarks based on a specific pipeline or tool version can quickly become outdated.

In contrast, benchmarking platforms aim at providing a way to facilitate continuous benchmarking by pipeline and tool developers and users. Examples of such platforms are OpenEBench [1] and Omnibenchmark [2]. Both platforms run on their own dedicated computing infrastructure and utilize specialized frameworks for results reporting and dataset uploading.

In this work, we want to propose a different approach for hosting a continuous benchmark, which was developed by the human genomic small variant calling benchmarking initiative of the NGS-CN [3] and GHGA [4]. We show that it is possible to build a benchmarking platform by entirely relying on public free infrastructure, namely GitHub [5], GitHub Actions [6], and Zenodo [7]. Using these technologies as a basis and extending upon best practices,¹⁰ we developed a comprehensive and reproducible benchmarking workflow for small genomic variants that is agnostic of the used dataset and can evaluate an arbitrary number of given callsets, while reporting the results in a visual and interactive way.

Methods

Datasets

We have sequenced the NA12878 sample from the genome in a bottle (GIAB) [8] project with two exome sequencing kits at varying average coverages. The genomic DNA from NA12878 was obtained from the NIGMS Human Genetic Cell Repository at the Coriell Institute for Medical Research. The Agilent Human All Exon V7 kit was used to yield a dataset with 182 million paired-end reads sequenced on an Illumina Nova Seq 6000 (211 bp mean insert size and 2×101 bp read length). We used random subsampling to derive two datasets from this that were used in the benchmarking, one with 37.5 million and one with 100 million paired-end reads. The Twist Human Comprehensive Exome (Twist Bioscience, San Francisco, CA, USA) sequencing kit was used according to the manufacturer's protocol to generate 200 million paired-end reads on an Illumina NovaSeq 6000 (291 bp mean insert size and 2×101 bp read length). The raw reads of the two subsampled Agilent and Twist exome datasets are available via Zenodo.^{11,12}

Evaluation pipeline

To analyze the quality of the callsets yielded by each pipeline on the given datasets, we have developed a generic, reproducible Snakemake¹³ workflow, which conducts all steps from downloading benchmark data, preprocessing,

¹<https://openebench.bsc.es>

²<https://omnibenchmark.org>

³<https://ngs-kn.de/>

⁴<https://www.ghga.de>

⁵<https://github.com>

⁶<https://github.com/features/actions>

⁷<https://zenodo.org>

⁸<https://www.nist.gov/programs-projects/genome-bottle>

comparison with a known ground truth, plotting, and automatic deployment of the required software stacks via Snakemake's Conda/Mamba [9] integration: <https://github.com/snakemake-workflows/dna-seq-benchmark>. The workflow comes with predefined standard datasets like CHM-eval and GIAB, but can be additionally configured to use any other DNA-seq-based benchmark dataset consisting of a known set of true variants, confident regions where the reported true variants are considered to be complete (*i.e.* every non-variant position is assumed to homozygously have the reference allele), raw read data (as FASTQ files), and (optionally) sequenced target regions (*e.g.* in case of exome sequencing). The workflow uses BWA-mem,¹⁴ Picard tools [10] and Mosdepth¹⁵ for calculating the read coverage across the genome. We use Bedtools¹⁶ to limit the known true variants to the confident regions provided by the respective truth publishers and to stratify variants by coverage (see below). For interactive exploration of the results, we use Datavzrd [11] and Vega-Lite.¹⁷ The matching of calls and true variants in a haplotype-aware manner happens via RTG-tools vcfEval [12]. To ensure a fair and correct comparison of the different evaluated callsets, several key points had to be considered, which we outline below.

Read depth stratification and selection of regions of interest. The available read depth can naturally affect both the precision and recall of a pipeline. Hence, the read depth characteristics of a benchmark dataset can have an impact on the derived precision and recall, which can limit the generalizability of obtained results. In order to avoid this effect, we decided to stratify recall and precision by read depth. For any benchmark dataset, this workflow generates a quantized set of regions with low (0-9), medium (10-30), and high (> 30) read depth using Mosdepth, while considering only reads with mapping quality (MAPQ) ≥ 60 . Notably, this means that, for example, regions in the low read depth category have either only few reads or a lot of reads with uncertain alignments (*high mapping uncertainty*). We intersect these regions with the confidence regions of the benchmark sample (*e.g.* as provided by GIAB) using Bedtools. If the given dataset was generated using a capturing approach (*e.g.* exome sequencing) we further restrict the regions to the captured loci according to the manufacturer. Afterwards, any given callset is split into three subsets with low, medium, and high coverage using Bedtools.

Separating genotyping from calling performance At decreasing read depth or increasing mapping uncertainty, one can expect a callset to yield a decreasing recall: with less evidence, it will become harder to find variants. This is true for both genotyping (*i.e.* requiring that the variant caller detects the correct genotype) as well as when just requiring the variant allele to be correctly recognized without considering whether the variant is predicted to be homo- or heterozygous (*i.e.* plain variant *calling* without genotyping). In contrast, a variant callset's precision should ideally remain constant and unaffected by a decrease in read depth or increase in mapping uncertainty, if the method manages to correctly report the increasing uncertainty with decreasing depth or increasing mapping uncertainty. The latter behavior differs between measuring a callset's genotyping or calling precision. In order to make these differences visible, we therefore decided to calculate precision and recall for both genotyping and calling separately.

Variant atomization Some variant callers report complex variants as replacements of longer alleles (*i.e.*, both the reported reference and the alternative allele are longer than one base, *e.g.* ACCGCGT>ACGCT). While this is in general a good idea (*e.g.* in order to be able to properly assess the combined impact on proteins), we found this to introduce problems with vcfEval's internal comparison approach. This resulted in spurious false positives and false negatives in callsets having such variants. Similar to the approach implemented in the hap.py pipeline [13], we solved this issue by introducing a normalization step prior to vcfEval into our analysis workflow, which uses Bcftools¹⁸ to normalize variants, in a way that indels are moved to their left-most possible location, and complex replacements are split into their atomic components—*i.e.* single nucleotide variants (SNVs), insertions or deletions (indels)—while removing exact duplicates resulting from the atomization.

Reporting For reporting results, we employ Datavzrd to create interactive tabular reports for recall and precision, as well as individual false positive and false negative variants. Datavzrd enables us to just provide the required data as TSV or CSV files combined with a configuration file that defines the rendering of each column. For the latter, one can choose from automatic link-outs, heatmap plots, tick plots, bar plots, or custom complex Vega-Lite plots (which can also be used to define alternative visualizations for an entire table view). For the former, we report a table containing for each callset and each read depth category (low, medium, high) precision and recall (while ignoring whether the genotype was predicted correctly), the underlying counts of true positives (TP), false positives (FP), and false negatives (FN), as well as

⁹<https://github.com/mamba-org/mamba>

¹⁰<https://broadinstitute.github.io/picard>

¹¹<https://github.com/datavzrd/datavzrd>

¹²<https://github.com/RealTimeGenomics/rtg-tools>

¹³<https://github.com/Illumina/hap.py>

the fraction of wrongly predicted genotypes. It is important to note that it cannot be excluded that the same variant in the truthset is predicted multiple times by a callset, *e.g.* as part of several complex replacements (see “Variant atomization” above). We therefore report two TP counts TP_{query} (number of TPs in the callset with the same matching variant from the truth potentially counted multiple times) and TP_{truth} (number of variants in the truth set that occur in the callset, each variant counted once, regardless how often it occurs in the callset), with $TP_{\text{query}} \geq TP_{\text{truth}}$. Following the established definitions, precision is then calculated as

$$\frac{TP_{\text{query}}}{TP_{\text{query}} + FP}$$

while recall is calculated as

$$\frac{TP_{\text{truth}}}{TP_{\text{truth}} + FN}$$

An example can be seen in [Figure 2](#). For reporting of individual FP and FN variants, we provide a Datavzrd table view for each that has one row per variant and a column for each callset. In order to visualize systematic patterns arising from the properties of callsets (*e.g.*, using the same variant detection or mapping method), any kind of property can be annotated as a so-called “label” when registering a callset for evaluation with the pipeline. The labels are displayed using a categorical color coding in the header of the table views. Moreover, we perform a χ^2 test for the association of the FP or FN pattern of each variant against the different labels in order to detect systematic effects. The variant/label combinations for which this test yields a significant result are then displayed in a separate table view for each type of label. These allow, for example, to spot variants that only occur when callsets use a particular variant caller. Thereby, significance is determined by controlling the false discovery rate over the p-values of the χ^2 test using the Benjamini-Yekutieli procedure, as the variants could be both positively (*e.g.*, being on the same haplotype) or negatively (*e.g.*, being on different haplotypes) correlated. In order to combine the results with data provenance information we include the Datavzrd views into a Snakemake report [14], which automatically provides a menu structure for navigation between views, association with used parameters, code, and software versions as well as runtime statistics.

Continuous public evaluation

A central goal of the project was not to conduct a single benchmark and just publish the results, but rather provide a resource for continuous repeated and always up-to-date benchmarking, that is moreover open to any kind of contribution (callsets and code improvements, among others) from outside collaborators. In order to achieve this, we have developed the following approach (see [Figure 1](#) for an illustration). We deployed the benchmarking workflow [15] as a module [16] into another Snakemake workflow that in addition has the ability to download callsets from Zenodo, using Snakemake’s Zenodo integration [17]. Then, we deployed this workflow into the GitHub repository [18] and configured GitHub Actions [19] to continuously rerun the workflow upon every commit on the main branch or any pull request [20]. In order to ensure that the workflow runs sufficiently fast (GitHub Actions offers only limited runtime and resources per job), we have precomputed benchmark dataset-specific central intermediate results (read depth and confidence derived stratification regions) that are computationally intensive to obtain, and deployed them along with the workflow code into the GitHub repository.

Upon each completion of the evaluation pipeline, a Snakemake report [21] is generated. In case of pull requests (*e.g.*, contributing a feature or a new callset), the report is uploaded as a GitHub artifact [22], for inspection by the pull request author and the reviewer. In the case of the main branch, we utilize GitHub Actions to trigger the execution of a secondary GitHub Action pipeline in a repository that hosts the NCBench homepage [23]. This pipeline fetches the latest report artifact associated with the main branch and deploys it to the homepage. This way, the most recent results are automatically accessible on the homepage.

¹⁴<https://snakemake.readthedocs.io/en/stable/snakefiles/reporting.html>

¹⁵<https://github.com/snakemake-workflows/dna-seq-benchmark>

¹⁶<https://snakemake.readthedocs.io/en/stable/snakefiles/modularization.html#snakefiles-modules>

¹⁷https://snakemake.readthedocs.io/en/stable/snakefiles/remote_files.html#zenodo

¹⁸<https://github.com/ncbench/ncbench-workflow>

¹⁹<https://github.com/features/actions>

²⁰<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests>

²¹<https://snakemake.readthedocs.io/en/stable/snakefiles/reporting.html>

²²<https://docs.github.com/en/actions/using-workflows/storing-workflow-data-as-artifacts>

²³<https://ncbench.github.io>

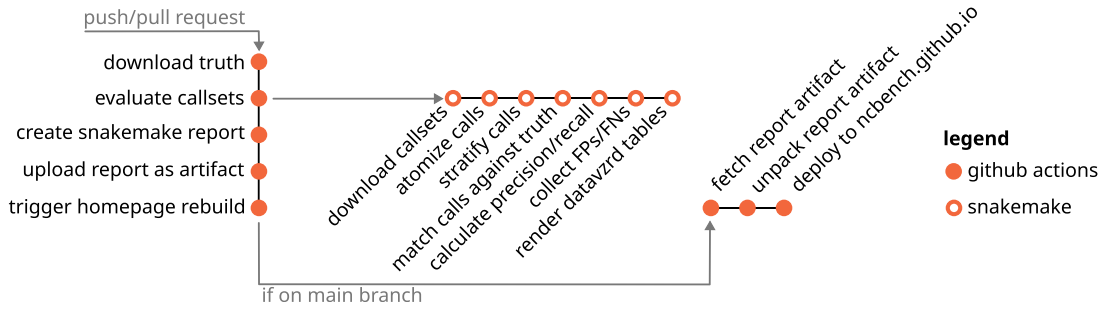


Figure 1. Continuous evaluation and reporting workflow. Upon pull requests or pushes, a GitHub Actions workflow is triggered. This downloads data, runs the Snakemake-based evaluation pipeline, creates the Snakemake report and uploads it as an artifact. If the workflow is triggered on the main branch, its finalization triggers a second Github Actions workflow that builds and deploys the homepage at <https://ncbench.github.io>.

Results

The always up-to-date results of the benchmark can be found and interactively explored under <https://ncbench.github.io>. At the time of writing, the benchmark consists of more than 40 callsets on three different benchmark datasets, the two NA12878 samples described in the Datasets section and CHM-eval.¹⁹ The callsets span various pipelines, read mapping, variant detection, and genotyping approaches.

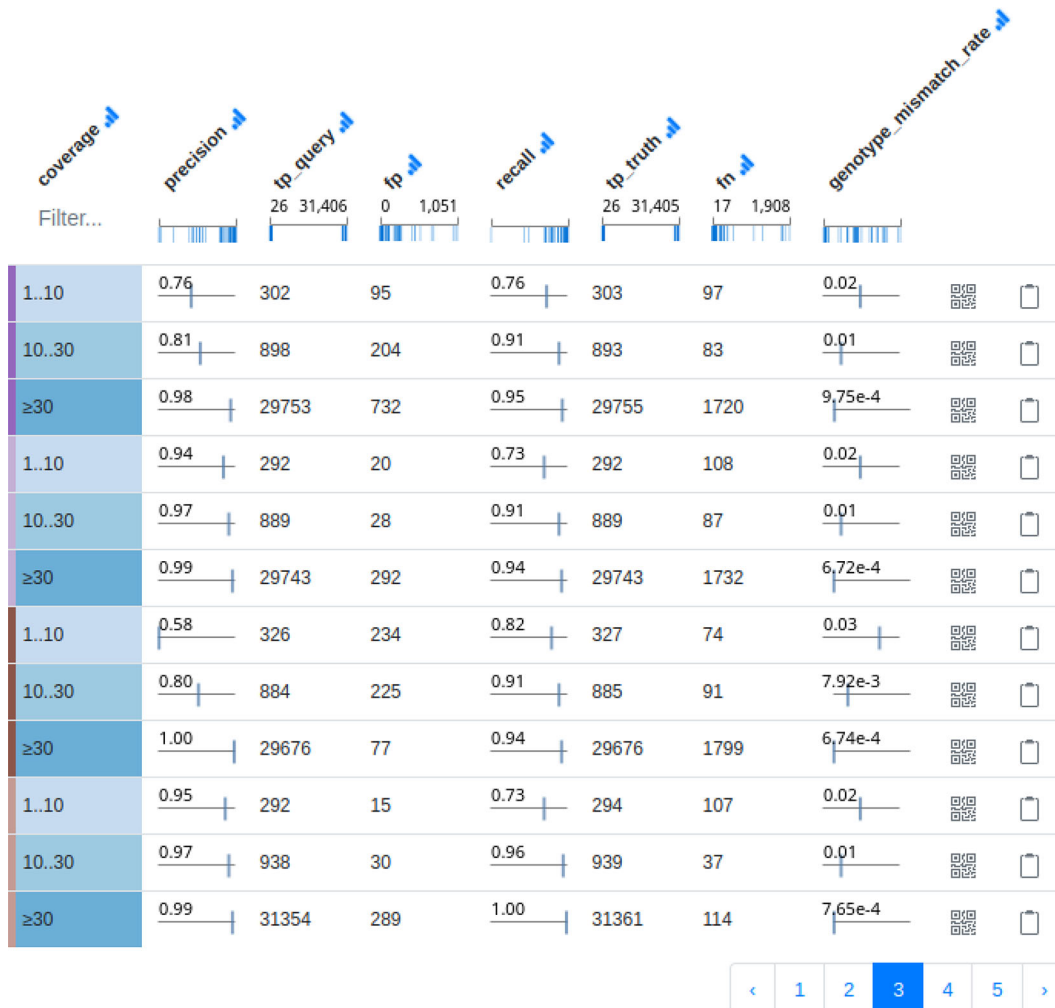


Figure 2. Exemplary screenshot of interactive tabular precision recall display. Each three rows display precision and recall together with underlying numbers and wrongly predicted genotypes stratified by read depth/coverage category. In the interactive report, callset/pipeline names would occur on the left. Here, they have been removed since results can be expected to change over time. For actual results please see the always up-to-date interactive report at <https://ncbench.github.io>.

Since the central idea of this project is to provide a continuous, standardized and open benchmark platform for DNA-seq, we strived to make the contribution of new callsets as straightforward as possible. The benchmark repository [24] shows the steps needed to perform variant calling on the supported datasets and describes how to pre-check the resulting callset locally. Once a contributor is convinced that the callset is ready for publication, we provide instructions for uploading the result to Zenodo and providing it via a pull request for continuous evaluation in the future.

Figure 2 shows an exemplary screenshot of the interactive tabular precision/recall display (see Pipeline section). This illustrates the importance of stratifying by read depth/coverage categories (see Pipeline). This is in contrast to the commonly seen practice, where GIAB and other benchmark datasets are evaluated on the entire set of variants, without stratification. While this generates realistic estimates for the prediction quality of a variant calling pipeline overall, the provided information is less generalizable, since a new dataset might have different read depth characteristics. Further, it tells little about the expected quality at an individual location, which might differ as well from the global characteristics.

Discussion and conclusions

So far, variant calling benchmark studies were often published once, in a single or multiple manuscripts that can only represent a snapshot at the time of writing. This holds both for studies evaluating multiple tools or pipelines, as well as the evaluations around newly published individual tools.

For continuous benchmarking, platforms like Omnibenchmark [25] or OpenEBench²⁰ are available. Both platforms run on their own dedicated computing infrastructure and utilize specialized frameworks for results reporting and dataset uploading.

In this work, we demonstrate that a continuous benchmarking platform can be set up without the need for dedicated computing infrastructure, and instead entirely relying on freely available and widely used resources.

- By basing the benchmark of DNA-seq variant calling pipelines on a public GitHub repository for code, configuration and result storage, GitHub Actions for analysis execution, and callsets hosted by Zenodo, we allow rapid and straightforward contributions by anybody used to these services.
- By implementing the analysis with Snakemake and Conda/Mamba, we decouple the analysis code and the reporting of results from the hosting platform: instead of relying on GitHub Actions, the benchmark analysis can easily be conducted locally, or on a different platform without any modifications of the code.
- By generating interactive visual presentations of the results with Datavzrd, we (a) allow for a modern and versatile exploration of results and comparisons between different methods and pipelines, and (b) to a large degree enable contributions and modifications to the way the data is presented by simply editing YAML based configuration files.
- By encapsulating all results in a Snakemake report that is portable and can be viewed and provided without any web service, we enable people to freely choose between relying on the online version of the report and providing snapshot-like versions of the report in their publications.

In the future, we will further extend upon this approach. For example, we will add a whole genome dataset of the NA12878 sample sequenced on an Illumina NovaSeq 6000 of *ca.* 400 million paired-end reads (mean insert size 473 and 2×151 bp reads length). Further, we will extend the pipeline to include the evaluation of structural and somatic variants and corresponding datasets. Finally, as the implemented comparison workflow is in principle agnostic to the considered species, we will evaluate the inclusion of benchmark datasets from non-human organisms. Particularly for natural microbial populations, whose species mostly exist as multiple genotypes in one ecosystem, variant calling can be a complex process²¹ and often not completely resolved due to the lack of complete and closed reference genomes from mono-cultures.

We hope that our approach will attract contributors beyond our initiative. Ideally, the combination of being continuous, simple to use, reproducible, and easy to integrate outside of the primary web service will change the way DNA-seq benchmarking is handled in the future. Instead of requiring every new tool and benchmark study manuscript to conduct its

²⁴<https://github.com/ncbench/ncbench-workflow>

²⁵<https://omnibenchmark.org>

own analysis for precision and recall on public resources like GIAB or CHM-eval as well as comparison with other tools or pipelines, authors can rather include their callsets in our benchmark. In turn, readers will be able to always see the performance of a tool in the context of the state of the art at the time of reading, instead of at the time of writing.

Author contributions

FH, GG, SM, and JK have written the manuscript. JK has implemented the benchmarking pipeline. SM has coordinated the benchmarking initiative. SM and KB provided the FastQ files for the Agilent Human All Exon v7 kit. MS has created the callset data for the megSAP pipeline and edited the manuscript. TS has created the callset data for the WEScrobio pipeline. LB has created the callset data for the Cologne exome pipeline. AGN analyzed callset data. JA and AO have sequenced the NA12878 sample at the WGGC Cologne. AJP contributed to discussion and manuscript writing. AP has provided advice on and reviewed the benchmark design. SS has created the call sets for the NVIDIA Parabricks pipeline. AB and FB have supported the NVIDIA Parabricks pipeline. AS has coordinated the sequencing of NA12878 at the NGS Core Facility Bonn. NHS and CM have created the callset data for the GHGA pipeline. GG and FH have created the callset data for the sarek pipeline. All authors have read and approved the manuscript.

Data availability

Underlying data

Twist Whole-Exome Sequencing Dataset of NA12878: <https://doi.org/10.5281/zenodo.7075040>

Agilent v7 exomes of NA12878: <https://doi.org/10.5281/zenodo.6513788>

Data are available under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/) (CC-BY 4.0).

CHM-eval public benchmark data: <https://github.com/lh3/CHM-eval>

Analysis code

NCBench code available from: <https://github.com/ncbench/ncbench-workflow>

Archived NCBench code available from: <https://doi.org/10.5281/zenodo.8268264>

Acknowledgements

We thank the NIST and the GIAB working group for their extraordinarily useful work. We thank the authors of CHM-eval for their amazing work and making their data publicly available. We acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

References

- Zook JM, Chapman B, Wang J, *et al.*: **Integrating human sequence data sets provides a resource of benchmark snp and indel genotype calls.** *Nat. Biotechnol.* Mar 2014; **32**(33): 246–251. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Zook JM, Catoe D, McDaniel J, *et al.*: **Ying Sheng, Karoline Bjarnesdatter Rypdal, and Marc Salit. Extensive sequencing of seven human genomes to characterize benchmark reference materials.** *Scientific Data.* Jun 2016; **3**(11): 160025. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Eberle MA, Fritzilas E, Krusche P, *et al.*: **A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree.** *Genome Res.* Jan 2017; **27**(1): 157–164. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Li H, Bloom JM, Farjoun Y, *et al.*: **A synthetic-diploid benchmark for accurate variant-calling evaluation.** *Nat. Methods.* Aug 2018; **15**(8): 595–597. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Wendell JSAS Cary's Russ Wolfinger, and MAQC As: **Sequencing benchmarked.**
- Barbitoff YA, Abasov R, Tvorogova VE, *et al.*: **Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery.** *BMC Genomics.* Feb 2022; **23**(1): 155. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chen J, Li X, Zhong H, *et al.*: **Systematic comparison of germline variant calling pipelines cross multiple next-generation sequencers.** *Sci. Rep.* Jun 2019; **9**(1): 9345–9345. MAG ID: 2953517386. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Supernat A, Vidarsson OV, Steen VM, *et al.*: **Comparison of three variant callers for human whole genome sequencing.** *Sci. Rep.* Dec 2018; **8**(11): 17851. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Zhao S, Agafonov O, Azab A, *et al.*: **Accuracy and efficiency of germline variant calling pipelines for human genome data.** *Sci. Rep.* Nov 2020; **10**(11): 20222. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Krusche P, Trigg L, Boutros PC, *et al.*: **Best practices for benchmarking germline small-variant calls in human genomes.** *Nat. Biotechnol.* May 2019; **37**(5): 555–560. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Motameny S: **Agilent v7 exomes of NA12878.** May 2022. [Publisher Full Text](#)
- Schmidt A, Sivalingam S, Bunes A, *et al.*: **Twist human comprehensive exome sequencing kit - high coverage - coriell -**

- NA12878**. September 2022.
[Publisher Full Text](#)
13. Mölder F, Jablonski KP, Letcher B, *et al.*: **Sustainable data analysis with Snakemake**. *F1000Res*. January 2021; **10**: 33.
[Publisher Full Text](#)
 14. Li H: **Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM**. *arXiv:1303.3997 [q-bio]* March 2013. arXiv: 1303.3997.
[Publisher Full Text](#)
 15. Pedersen BS, Quinlan AR: **Mosdepth: quick coverage calculation for genomes and exomes**. *Bioinformatics*. March 2018; **34**(5): 867–868.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
 16. Quinlan AR, Hall IM: **BEDTools: a flexible suite of utilities for comparing genomic features**. *Bioinformatics*. March 2010; **26**(6): 841–842.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
 17. Satyanarayan A, Moritz D, Wongsuphasawat K, *et al.*: **Vega-Lite: A Grammar of Interactive Graphics**. *IEEE Trans. Vis. Comput. Graph.* January 2017; **23**(1): 341–350.
[PubMed Abstract](#) | [Publisher Full Text](#)
 18. Danecek P, Bonfield JK, Liddle J, *et al.*: **Twelve years of SAMtools and BCFtools**. *GigaScience*. February 2021; **10**(2): giab008.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
 19. Li H, Bloom JM, Farjoun Y, *et al.*: **A synthetic-diploid benchmark for accurate variant-calling evaluation**. *Nat. Methods*. August 2018; **15**(8): 595–597.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
 20. Capella-Gutierrez S, de la Iglesia D, Haas J, *et al.*: **Lessons Learned: Recommendations for Establishing Critical Periodic Scientific Benchmarking, August 2017. Pages: 181677 Section: New Results**.
 21. Olm MR, Crits-Christoph A, Bouma-Gregson K, *et al.*: **instrain profiles population microdiversity from metagenomic data and sensitively detects shared microbial strains**. *Nat. Biotechnol.* Jun 2021; **39**: 727–736.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 28 February 2024

<https://doi.org/10.5256/f1000research.153686.r239648>

© 2024 Cleal K. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Kez Cleal

Cardiff University, Cardiff, UK

The article presents NCBench, a benchmarking platform for evaluating SNVs and indel variant calls from various gold-standard benchmark sets. NCBench makes use of public, free infrastructure like GitHub Actions and Zenodo to facilitate continuous, open-source benchmarking, which is a neat idea. NCBench supports evaluation across several datasets, emphasizing adaptability and reproducibility, and effectively addresses the need for ongoing evaluation in genomic research. However, the true utility of NCBench will depend on its adoption by the wider research community which itself could be enhanced by making the platform as easy to use as possible. On this front, I think NCBench could be improved by addressing the following:

1. The documentation of how to implement/run NCBench and how to contribute a new callset was pretty limited. A more detailed guide aimed at newcomers, detailing the pipeline steps, inputs and outputs, and ways to configure NCBench for other datasets would help with adoption.
2. It is unclear how to run NCBench locally on a custom benchmark dataset. The pipeline supports uploading a vcf to zenodo and running via github actions. Ideally, NCBench should be possible to run locally to test different tool parameters, or for developers to experiment with new tool implementations. This pattern would make NCBench more useful for general bioinformatics workflows. If this pattern is supported, it should be documented more clearly.
3. The output tables on github.io are useful, but I think they could be improved. For example, it would be useful if rows could be sorted by a column, ability to hide certain columns.
4. There doesn't appear to be a way to download the results of the benchmarking run, or if there is, I didn't find it. It would be nice to be able to download results in a table, in order to make custom tables for publication, for example.
5. Stratifying benchmark results by coverage categories is a nice idea. However, I think it would still be useful to include an 'any' category (any mapq), to provide a dataset summary. Additionally, if there was a way to filter some of the rows, this would be very useful, for example selecting only mapq 1-10 category.
6. Tables should probably include an F1 score, this would be useful to rank callsets, although

- which metric used (tp_query or tp_truth) should be evident.
7. The dark-blue colour of some of the cells on the tables makes the text very hard to read (see 'Show as plot' subpage).
 8. Some of the table numbers needs to be rounded, for example clicking on 'Show as plot', the numbers are rounded to >12dp.
 9. In some of the tables, the text doesn't fit on the page (fn variant page, the top row of labels stretched off the page using Safari).

Is the work clearly and accurately presented and does it cite the current literature?

Yes

Is the study design appropriate and is the work technically sound?

Yes

Are sufficient details of methods and analysis provided to allow replication by others?

Yes

If applicable, is the statistical analysis and its interpretation appropriate?

Not applicable

Are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions drawn adequately supported by the results?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Cancer genomics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 03 October 2023

<https://doi.org/10.5256/f1000research.153686.r207163>

© 2023 Zook J. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Justin M. Zook 

Material Measurement Laboratory, National Institute of Standards and Technology (NIST),
Gaithersburg, MD, USA

The authors present an open benchmarking platform for variant calling, with exome and genome variant call sets for two different benchmarks as examples. As the authors note, the ability to continuously benchmark variant calls is currently lacking, as existing efforts like precisionFDA Truth Challenges only reflect a point in time. Ongoing benchmarking should be very useful as both variant calling methods and benchmark sets evolve. I only have a few suggestions for clarifying the data and methods used.

1. Are FNs and FPs counted to include genotype errors, as is the default in vcfeval and hap.py, or do both exclude genotype errors from their counts, which is implied but not explicit in the text
2. I saw the bed file for Agilent but not for twist in zenodo
3. What were the benchmark versions used for GIAB and CHM-eval?
4. Do the authors expect that adding new versions of the benchmarks would be straightforward as GIAB develops these?
5. vcfeval generally should be able to compare different representations of the same variant, as long as they exactly match, even if they are not atomized. The only reason I have found this not to work is if the variant caller gets part of a haplotype wrong or the genotype wrong, in which case the whole haplotype is called wrong even if most of the variants in the region are correct. Is this what the authors' encountered? If so, it might be good to clarify this. A potential problem with left shifting variants is that occasionally this will cause a change in the haplotype, e.g., if an indel in a homopolymer is shifted past a SNV on the same haplotype, though is relatively rare.
6. The number of variants for CHM-eval is lower than I'd expect. Did the authors restrict to one chromosome?

Is the work clearly and accurately presented and does it cite the current literature?

Yes

Is the study design appropriate and is the work technically sound?

Yes

Are sufficient details of methods and analysis provided to allow replication by others?

Partly

If applicable, is the statistical analysis and its interpretation appropriate?

Yes

Are all the source data underlying the results available to ensure full reproducibility?

Partly

Are the conclusions drawn adequately supported by the results?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Variant calling and benchmarking

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

This text is made available via DuEPublico, the institutional repository of the University of Duisburg-Essen. This version may eventually differ from another version distributed by a commercial publisher.

DOI: 10.12688/f1000research.140344.1

URN: urn:nbn:de:hbz:465-20240802-100211-1



This work may be used under a Creative Commons Attribution 4.0 License (CC BY 4.0).