vcferr: Development, validation, and application of a single nucleotide polymorphism genotyping error simulation framework [version 1; peer review: 1 approved, 2 approved with reservations]

V.P. Nagraj¹, Matthew Scholz¹, Shakeel Jessa¹, Jianye Ge², August E. Woerner², Meng Huang², Bruce Budowle², Stephen D. Turner³¹

¹Signature Science LLC., Austin, TX, 78759, USA
²Center for Human Identification, Department of Microbiology, Immunology, and Genetics, University of North Texas Health Science Center, Fort Worth, TX, 76107, USA

Abstract

Motivation: Genotyping error can impact downstream single nucleotide polymorphism (SNP)-based analyses. Simulating various modes and levels of error can help investigators better understand potential biases caused by miscalled genotypes.

Methods: We have developed and validated vcferr, a tool to probabilistically simulate genotyping error and missingness in variant call format (VCF) files. We demonstrate how vcferr could be used to address a research question by introducing varying levels of error of different type into a sample in a simulated pedigree, and assessed how kinship analysis degrades as a function of the kind and type of error.

Software availability: vcferr is available for installation via PyPi (https://pypi.org/project/vcferr/) or conda (https://anaconda.org/bioconda/vcferr). The software is released under the MIT license with source code available on GitHub (https://github.com/signaturescience/vcferr)

Keywords
bioinformatics, kinship, python, genealogy, GWAS, simulation, benchmarking

This article is included in the Bioinformatics gateway.
Corresponding author: V.P. Nagraj (vpnagraj@signaturescience.com)

Author roles: Nagraj VP: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Scholz M: Conceptualization, Investigation, Methodology, Software, Validation, Writing – Review & Editing; Jessa S: Writing – Review & Editing; Ge J: Funding Acquisition, Writing – Review & Editing; Woerner AE: Conceptualization, Writing – Review & Editing; Huang M: Writing – Review & Editing; Budowle B: Conceptualization, Funding Acquisition, Writing – Review & Editing; Turner SD: Conceptualization, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported in part by award 2019-DU-BX-0046 (Dense DNA Data for Enhanced Missing Persons Identification) to B.B., awarded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice and by internal funds from the Center for Human Identification. The opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect those of the U.S. Department of Justice. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2022 Nagraj VP 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: Nagraj VP, Scholz M, Jessa S et al. vcferr: Development, validation, and application of a single nucleotide polymorphism genotyping error simulation framework [version 1; peer review: 1 approved, 2 approved with reservations] F1000Research 2022, 11:775 https://doi.org/10.12688/f1000research.122840.1

First published: 11 Jul 2022, 11:775 https://doi.org/10.12688/f1000research.122840.1
Introduction

Single nucleotide polymorphisms (SNPs) are inherited single base-pair substitutions in genomic DNA that are easily characterized by microarray, targeted, or whole-genome sequencing, and have a wide range of uses in personalized medicine, disease association studies, population genetics, genealogy, and forensics.\(^1,2\) Genotyping error occurs at known rates on genomic sequencing and array platforms. However, novel applications of sequencing or genotyping such as those encountered in forensics\(^3\) or ancient DNA,\(^4\) or novel downstream data analysis techniques such as imputation on low-pass whole-genome sequencing\(^5-8\) can result in genotyping errors that differ from well-established rates and types of errors that occur in typical genotyping and high-coverage sequencing in high-quality samples. Different applications or data analysis procedures can result in errors that can manifest in a variety of ways, for example, with differing rates of drop in or drop out of heterozygous calls.

There are several existing tools for simulating sequencing or genotype array data. Wgsim is a tool for simulating sequencing reads from a reference genome, allowing for a uniform user-defined substitution sequencing error.\(^9\) The ART sequencing read simulator allows for technology-specific read error models and base quality value profiles derived empirically from large sequencing datasets.\(^10\) Downstream of raw sequencing read error simulation, other tools provide methods for simulating genotyping datasets (as if variant calling had already been performed, or genotyping microarrays were being used to generate SNP genotypes). The ped-sim software allows for simulating large, arbitrarily complex pedigrees with genotypes drawn from founders in a variant call format (VCF) file,\(^11\) allowing for realistic familial data to be generated by employing sex-specific genetic maps and realistic crossover interference models.\(^12\) Ped-sim allows for specifying the rate of genotyping error and the rate of an opposite homozygous error conditional on a genotyping error at a marker, but does not allow for arbitrary specification of substitution rates in biallelic SNPs (such as different rates for heterozygous to homozygous reference drop-out versus homozygous reference to heterozygous drop-in), and it applies these universal error rates to all samples in a multi-sample VCF.

While there are tools available to simulate errors in sequencing reads, and other tools such as ped-sim which allow for broad specification of global error rate in genotype calls, to our knowledge there is no existing tool that allows for individual specification of the type and degree of genotyping error an arbitrary sample in a VCF (e.g., different rates for heterozygous to homozygous reference versus homozygous to heterozygous). The ability to precisely simulate the degree and kind of genotyping error (based on error probabilities discovered from prior upstream data analysis) is necessary to facilitate testing and evaluation of downstream applications which may be differentially sensitive to different types of genotyping errors. Here we describe our efforts to develop, validate, and demonstrate an application and usage of vcferr, a user-friendly python tool for probabilistically simulating arbitrarily specified genotyping error and missing data into VCF files. Source code, PyPI/bioconda package, and code to reproduce the results presented here are given in the Software availability section below.\(^2,7\)

Methods

Implementation

The vcferr python package

vcferr is written in Python and distributed to run directly from a command line interface. The tool is built using pysam,\(^13\) which internally manages reading and writing VCF file contents. When run on an input VCF, vcferr will load all variant records for a given sample. The tool checks each genotype, and randomly draws from a list of possible genotypes (heterozygous, homozygous for the alternate allele, homozygous for the reference allele, missing) with each element weighted by error rates specified by the user. The processing runs iteratively for every site in the input VCF, with the output streamed or optionally written to a new output VCF file. The following error and missingness models are available:

- **rarr** = Heterozygous drop out to homozygous reference: 0/1 or 1/0 to 0/0
- **aara** = Homozygous alternate to heterozygous: 1/1 to 0/1
- **rrra** = Homozygous reference to heterozygous drop in: 0/0 to 0/1
- **raaa** = Heterozygous to homozygous alternate: 0/1 or 1/0 to 1/1
- **aarr** = Homozygous alternate to homozygous reference: 1/1 to 0/0
- **rraa** = Homozygous reference to homozygous alternate: 0/0 to 1/1
ramm = Heterozygous to missing: 0/1 or 1/0 to ./.
rrmm = Homozygous reference to missing: 0/0 to ./.
aamm = Homozygous alternate to missing: 1/1 to ./.

We validated vcferr by reviewing concordance metrics between VCF files before and after error simulation. To perform this validation we used data from the 1000 Genomes Project\textsuperscript{14} (see Underlying data) and the companion error assessment tool described below. The code for accessing the 1000 Genomes Project data and for reproducing the analyses presented in this paper is available from Extended data.\textsuperscript{25} First, we retrieved all genotype calls for chromosome 22 from the 1000 Genomes file transfer protocol (FTP) site. Next, we simulated a set of error rates for one sample. We observed that the proportion of mismatches reported by our bcftools-based error assessment procedure aligned as expected to each of the specified error probabilities. We also confirmed that the simulation of error in one sample did not trigger any mismatches for other samples in the VCF. The chromosome 22 VCF includes 1,103,547 positions and 2504 samples, and genotype error simulation took approximately five minutes on a single CPU (central processing unit). The code used to fully reproduce this validation analysis is described in the Extended data section below.\textsuperscript{25}

Companion evaluation tool: nrc

As part of this analysis we developed a companion tool (called nrc) to measure all types of error types that can be observed with biallelic SNPs between a reference and a query sample. This tool also computes the nonreference concordance (NRC), which is defined as:

$$NRC = 1 - \frac{e_{rr} + e_{ra} + e_{aa}}{e_{rr} + e_{ra} + e_{aa} + m_{ra} + m_{aa}}$$

Where $e_{rr}$, $e_{ra}$, and $e_{aa}$ are defined as the counts of mismatches for the homozygous reference, heterozygous, and homozygous alternate genotypes, and $m_{ra}$ and $m_{aa}$ are the counts of the matches at the heterozygous and homozygous alternate genotypes (the number of homozygous reference matches, $m_{rr}$, is omitted from this analysis). We implemented this procedure as a Docker container, which first runs bcftools stats\textsuperscript{13} followed by post-processing results using R. The Dockerfile needed to build the nrc image is available in the Software availability section below,\textsuperscript{27} where the Docker image can be pulled directly from the GitHub container registry.

Operation

vcferr is packaged in Python and available to be installed from PyPi or Bioconda. Once installed, the tool can be used through the command line without calling the Python interpreter. vcferr requires a path to an input VCF from which error will be simulated as the first positional argument. The tool also requires a specification for --sample for the ID of the sample to be simulated as it appears in the VCF header. All other error flags are optional, as is --output-vcf which if used will define the path to which the VCF should be written. If no value is passed to a given error or missingness option, then the probability will default to 0. The following is a minimal example of usage, which will result in 5% of the heterozygous genotypes in "sampleX" to be erroneously called as homozygous reference, and 1% of that same individual’s heterozygous genotypes to be erroneously called as homozygous alternate, with output being written to a new bgzipped VCF file:

```bash
# Install via pip or conda
pip install vcferr
conda install --c bioconda vcferr

# Run vcferr
vcferr my.vcf.gz --sample='sampleX' --p_rarr=0.05 --p_raaa=0.01 --output_vcf=sampleX.err.vcf.gz
```

The companion tool nrc for assessing individual error rate probabilities and overall non-reference concordance is available as a Docker image from the GitHub container registry. Once installed, the container can be run through the command line with no dependencies other than Docker. The container expects two VCF files and a sample ID which exists in both VCFs to compare. Optionally, a set of sites at which to compare can be specified (default is to compare all sites). This is useful to restrict analyses to a subset of sites (e.g., those that have at least some minimum minor allele frequency from GnomAD, given that filtered site VCF). Example usage:
# Pull the image from the github container registry and tag:
docker pull ghcr.io/signaturescience/nrc
docker tag ghcr.io/signaturescience/nrc nrc

# Run the nrc tool on sampleX in two vcfs in the current working directory,
# restricting analyses to sites in targetsites.vcf.gz in the current directory.
docker run --rm -v $(pwd):$(pwd) -w $(pwd) nrc \
  -s sampleX \
  -T targetsites.vcf.gz \
  original.vcf.gz vcferr.vcf.gz

Use cases

Benchmarking SNP-based kinship analysis
To demonstrate how vcferr could be used to address a research question, we conducted an analysis of the impact of genotyping error on pairwise kinship estimation. For instance, a heterozygous drop-out to homozygous reference may be much better tolerated than a switch from homozygous reference to homozygous alternate. We first generated simulated pedigree data using ped-sim\textsuperscript{12} with founders drawn from the British in England and Scotland (GBR) population in the 1000 Genomes Project data. The simulated data was masked to a subset of 590,241 sites, including content from the Illumina Global Screening Array. With all data originally simulated to include no error, we next identified an individual for which we would inject genotyping error. We used vcferr to iterate through each error mode from 0 to 20% error at 1% steps, holding all other error modes constant at 0.

# Loop through 0, .01, .02, …, .19, .20 for each error mode, 
# keeping all other error modes at zero
vcferr pedigree.vcf.gz \
  --sample='sample_to_mutate' \
  --p_rarr=0.05 \
  --p_aara=0 \
  --p_rrra=0 \
  --p_raaa=0 \
  --p_aarr=0 \
  --p_rraa=0 \
  --output_vcf=pedigree.mutated.prarr.05.vcf.gz

Next, we obtained KING robust\textsuperscript{15} kinship coefficients from akt\textsuperscript{16} for all pairwise comparisons that included the error-simulated individual in the resulting simulation VCFs. We then used the kinship coefficients to infer relatedness out to the third-degree and compared accuracy of degree inference to truth from the original simulated pedigree. The results of this analysis are summarized in Figure 1. Code needed to reproduce this entire analysis, implemented as a Snakemake workflow,\textsuperscript{17} is available in the Zenodo archive described in the Extended data section below.\textsuperscript{25}

In short, certain kinds of error dramatically decrease accuracy of relatedness inference with the KING robust estimator. Note that the sample for which errors were simulated included 430,056 homozygous reference sites, 105,413 heterozygous sites, and 54,772 homozygous alternate site, resulting in varying absolute counts of errors being simulated at each 0 to 20% step (as expected in any sample with genotypes from microarray or sequencing). By leveraging vcferr to inject a range of genotyping error scenarios, we can identify the scenarios that are most likely to impact kinship analysis.

Genome-wide association studies and polygenic risk scores
Genome-wide association studies (GWAS) are a widely used approach for genetic epidemiology studies attempting to understand genetic underpinnings to complex human disease.\textsuperscript{18} Formalin-fixed, paraffin-embedded (FFPE) tissue fixation is a standard means to preserve tissues from clinical samples, and can result in genotyping error\textsuperscript{19} that can vary by type, with heterozygous errors being more commonly discordant.\textsuperscript{20} Power and sample size requirements for GWAS are impacted by genotyping error.\textsuperscript{21} If error probabilities for different sample types such as FFPE tissue samples are known, the impact of different types of genotyping error on GWAS power or sample size requirements can be estimated in simulation. One of many available tools can be used to simulate GWAS data with known sample size, effect size, SNP density, etc.\textsuperscript{22,23} Following data simulation, vcferr can be used to inject error in samples of interest, precisely controlling the degree and kind of error in each sample.
Similarly, a polygenic risk score (PRS) captures the additive effect of hundreds, or perhaps thousands of SNPs, and are computed on a genome-wide genotyping data using a prediction model based on summary statistics from external GWAS studies. PRS are first calculated by summing the risk alleles an individual has where each risk allele is weighted by the risk allele effect size as estimated by a large GWAS on that phenotype. The PRS is then applied to target genotype data to assess overall genetic liability to developing the trait or disease. If the target genotypes result from a process that introduces error at different levels for different zygosity (e.g., low-pass whole genome sequencing), the performance of a PRS will be impacted differently depending on the type and kind of error introduced. If this error profile is known by characterizing against high-quality samples, those error profiles can be used to inject error at those specified rates into samples simulated for assessing the predictive ability of a PRS in a new target genotype collection using the same approach above for GWAS.

Summary

`vcferr` provides a flexible, user-friendly interface to simulate genotyping error in VCF files. While tools exist to simulate entire VCFs with error (e.g. `ped-sim`), to our knowledge at the time of writing there are currently no other tools that can simulate arbitrary individuals in the VCF at varying types and levels of error while leaving all other samples as-is. We anticipate that `vcferr` will be useful for researchers in multiple areas, including forensic genomics and medical genetics.
**Data availability**

**Underlying data**

The 1000 Genomes Data\(^{14}\) used in this study is freely accessible to anyone without registration via the The International Genome Sample Resource website here: HTTPS://WWW.INTERNATIONALGENOME.ORG/CATEGORY/FTP/.

**Extended data**

- Archived analysis code at time of publication: HTTPS://DOI.ORG/10.5281/zenodo.6599134\(^{25}\)
- License: Creative Commons Attribution 4.0 International license (CC-BY 4.0)

**Software availability**

**vcferr**

The `vcferr` python tool is the primary software tool described and demonstrated in this paper.

- Software available from: PyPI (HTTPS://PYPI.ORG/PROJECT/VCFERR/) and Bioconda (HTTPS://ANAconda.ORG/BIOconda/VCFER)
- Source code available from: HTTPS://GITHUB.COM/SIGNATURESCIENCE/VCFERR
- Archived source code at time of publication: HTTPS://DOI.ORG/10.5281/zenodo.6611796\(^{26}\)
- License: MIT

**nrc**

The `nrc` companion tool was created for assessing error produced by `vcferr`

- Software (Docker image) available from: HTTPS://GHCR.IO/SIGNATURESCIENCE/NRC
- Source code available from: HTTPS://GITHUB.COM/SIGNATURESCIENCE/NRC
- Archived source code at time of publication: HTTPS://DOI.ORG/10.5281/zenodo.6611782\(^{27}\)
- License: MIT

**Acknowledgements**

The authors acknowledge Dr. Mike Coble for helpful discussions about the software and its use cases.

**References**

1. Nielsen R, Paul JS, Albrechtsen A, Song YS: Genotype and SNP calling from next-generation sequencing data. Nat. Rev. Genet. June 2011; 12(6): 463–481.
2. Kim S, Misra A: SNP genotyping: Technologies and biomedical applications. Annu. Rev. Biomed. Eng. 2007; 9: 289–320.
3. Pompanon F, Bonin A, Bellemain E, Taberlet P: Genotyping errors: causes, consequences and solutions. Nat. Rev. Genet. November 2005; 6(11): 847–859.
4. Gorden EM, Greytak EM, Stark-Andreaggi K, et al.: Extended kinship analysis of historical remains using SNP capture. Forensic Sci. Int. Genet. March 2022; 57: 102636.
5. Hui R, D'Anastasi E, Cassidy LM, et al.: Evaluating genotype imputation pipeline for ultra-low coverage ancient genomes. Sci. Rep. October 2020; 10(1): 16542.
6. Martin AR, Atkinson EG, Chapman SB, et al.: Low-coverage sequencing cost-effectively detects known and novel variation in underrepresented populations. bioRxiv. April 2020; page 2020.04.27.064832.
7. Li JH, Mazur CA, Berisa T, Pickrell JK: Low-pass sequencing increases the power of GWAS and decreases measurement error of polygenic risk scores compared to genotyping arrays. bioRxiv. May 2020; page 2020.04.29.068452.
8. Rubinacci S, Ribeiro DM, Hofmeister RJ, Delaneau O: Efficient phasing and imputation of low-coverage sequencing data using large reference panels. Nat. Genet. January 2021; 53(1): 120–126.
9. Li H: Wgsim. 2011.
10. Huang W, Li L, Myers JR, Marth GT: ART: A next-generation sequencing read simulator. Bioinformatics. February 2012; 28(4): 593–594.
11. Danecek P, Auton A, Abecasis G, et al.: The variant call format and VCFtools. Bioinformatics. 06 2011; 27(15): 2156–2158.
12. Caballero M, Seidman DN, Qiao Y, et al.: Crossover interference and sex-specific genetic maps shape identical by descent sharing in close relatives. *PLoS Genet.* December 2019; 15(12): e1007979. 1553-7404. Publisher: Public Library of Science. PubMed Abstract | Publisher Full Text

13. Danecek P, Bonfield JK, Liddle J, et al.: Twelve years of SAMtools and BCFtools. *GigaScience.* February 2021; 10(2): giab008. 2047-217X. PubMed Abstract | Publisher Full Text

14. Auton A, Abecasis GR, Altshuler DM, et al.: A global reference for human genetic variation. *Nature.* October 2015; 526(7571): 68–74. Publisher Full Text | Reference Source

15. Manichaikul A, Mychaleckyj JC, Rich SS, et al.: Robust relationship inference in genome-wide association studies. *Bioinformatics.* 10 2010; 26(22): 2867-2873. 1367-4803. PubMed Abstract | Publisher Full Text

16. Arthur R, Schulze-Trieglaff O, Cox AJ, et al.: AKT: ancestry and kinship toolkit. *Bioinformatics.* 09 2016; 33(1): 142-144. 1367-4803. PubMed Abstract | Publisher Full Text

17. Koster J, Rahmann S: SnakeMake—a scalable bioinformatics workflow engine. *Bioinformatics.* 08 2012; 28(19): 2520-2522. Publisher Full Text

18. Gallagher MD, Chen-Plotkin AS: The post-gwas era: From association to function. *Am. J. Hum. Genet.* 05 2018; 102(5): 717-730. PubMed Abstract | Publisher Full Text

19. Sikora MJ, Thibert JN, Salter J, et al.: High-efficiency genotype analysis from formalin-fixed, paraffin-embedded tumor tissues. *Pharmacogenomics.* 06 2010; 11(5): 348-358. PubMed Abstract | Publisher Full Text

20. Hertz DL, Kidwell KM, Thibert JN, et al.: Genotyping concordance in DNA extracted from formalin-fixed paraffin embedded (FFPE) breast tumor and whole blood for pharmacogenetic analyses. *Mol. Oncol.* 07 2015; 9(9): 1868-1876. PubMed Abstract | Publisher Full Text

21. Lin H, Sun N, Mane S, et al.: Impact of genotyping errors on statistical power of association tests in genomic analyses: A case study. *Genet. Epidemiol.* 12 2010; 41(2): 152–162. PubMed Abstract | Publisher Full Text

22. Fortune MD, Wallace C: simgwas: a fast method for simulation of large scale case-control GWAS summary statistics. *Bioinformatics.* 10 2018; 35(11): 1901–1906. PubMed Abstract | Publisher Full Text

23. Tang Y, Liu X: 6Zp: a genome-wide-association-study simulation tool for genotype simulation, phenotype simulation and power evaluation. *Bioinformatics.* 02 2019; 35(19): 3852–3854. PubMed Abstract | Publisher Full Text

24. Torkamani A, Weininger NE, Topol EJ: The personal and clinical utility of polygenic risk scores. *Nat. Rev. Genet.* 05 2018; 19(9): 581–590. PubMed Abstract | Publisher Full Text

25. Nagraj VP, Scholz M, Jessa S, et al.: Documentation and code for reproducing analyses presented in: vcferr: Development, Validation, and Application of a SNP Genotyping Error Simulation Framework [Software]. *Zenodo.* 2022. Publisher Full Text

26. Nagraj VP, Turner S, Scholz M: signaturescience/vcferr: v1.0.2-zenodo (v1.0.2-zenodo). *Zenodo.* 2022. Publisher Full Text

27. Turner S: signaturescience/nrc: v1.0.0 (v1.0.0). *Zenodo.* 2022. Publisher Full Text
Open Peer Review

Current Peer Review Status:  

Version 1

Reviewer Report 15 May 2024

https://doi.org/10.5256/f1000research.134883.r260018

© 2024 Taniguti C. 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.

Cristiane Taniguti  
Texas A&M University, College Station, Texas, USA

The software presents a method for simulating genotyping errors in specific samples in a VCF file. The novelty compared to other software available is that it can introduce different error types and intensities. Having the tool in Conda or Docker is also a great advantage, as some other tools created for the same purpose are currently outdated and difficult to install. The authors provide two use cases, one measuring the impact of the different error types in kinship analysis and the other in GWAS and polygenic risk score estimation. The authors provide links for all mentioned resources including the software by itself, a companion package called nrc, and a Snakemake workflow to reproduce the analysis of the first use case.

Results are presented for only the first of the cases. Despite not being necessary to understand the application, having access to results for the second use case would consolidate the statement. If not presently available, I recommend considering two additional features: incorporating a seed option for simulations and providing detailed information on the absolute number of modified genotypes and their final counts. Not knowing the absolute numbers, users can easily get lost in the interpretations of the results once the error rate parameters are based on frequencies and the absolute number will depend on the proportion of the initial genotypes, as emphasized in paragraph #12.

I noticed that the introduction doesn't mention amplicon sequencing techniques like genotyping-by-sequencing (GBS or RAD-seq) as a means to obtain SNPs. In the plant genetics field, GBS methods have been broadly used. Compared to the other methodologies mentioned in the text, GBS results have been showing a higher number of genotyping errors. Lots of efforts in terms of software have been made to try to overcome these errors (see Furuta et al. 2023 for an example Ref [1]). Briefly mentioning it in the text could expand the audience and the applications to this other domain.

Other than predicting the accuracy rate for certain doubt quality samples, do you think that the tool could be useful to also measure the capabilities of these software designed to fix genotyping errors?

Overall, the text is well-written and follows a logical description for a software release.
References
1. Furuta T, Yamamoto T, Ashikari M: GBScleanR: robust genotyping error correction using a hidden Markov model with error pattern recognition. *Genetics.* 2023; 224 (2). PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, statistical genetics, plant genetics, polyploids

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 02 May 2024

https://doi.org/10.5256/f1000research.134883.r266345

© 2024 Vedova G. 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.

Gianluca Della Vedova
University of Milano-Bicocca, Milan, Italy

The paper present a tool for simulating genotyping errors on top of a know vcf file. The tool has a clear scope and usefulness. The paper is well written and easy to follow. I have only a few, very minor, suggestions but I recommend to accept the paper.
Page 3, "specified genotyping error". I suggest to state if you accept some kind of error profile.
Page 3. "The tool is built using" -> "The tool depends on"
Page 4. "from which error will be simulated as the first positional argument". Please state explicitly that this is the ground truth you will use in your simulation steps.
Page 4. (last paragraph) "The companion tool nrc ... registry". You have already discussed nrc and it docker image about half a page before (just before Operation).
Page 5, Code snippet #1. I expect a single "docker run" to suffice. May you explain why you also have a docker pull and a docker tag?
Page 5, Code snippet #2. "Loop through ... error mode". I cannot find any loop in that code. Please make the code and the comments consistent.
Page 7, Underlying data. Now that web browsers do not support ftp, I think it's time to move away from it, especially since it provides no advantages. That's not much that the authors can do, so it's mostly a suggestion for future works.
Page 7. Please explain (here or somewhere else in the paper) why you have decided to release nrc as a docker image, instead of via conda (as you have done for vcferr). I find puzzling to use two different installation methods, so it's better to provide a rationale.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Algorithms

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.
Ardalan Naseri
University of Texas Health Science Center at Houston, Texas, USA

The paper presents a new tool called vcferr, which can be used to simulate genotyping errors and missing data in variant call format (VCF) files. The authors demonstrate how vcferr can help answer research questions by introducing various levels and types of errors into a simulated pedigree, and evaluating how kinship analysis is affected. While there are other similar tools available, the authors highlight the unique features of their work.

However, some minor clarifications are needed. The terms "degree" and "kind" of genotyping errors require further explanation. I assume "Kind" refers to different error models, such as RARR or AARA, while "degree" refers to the rates. In the manuscript, they have been referred to as type as well. A more coherent terminology would be helpful.

Regarding the implementation, "randomly draws from a list of possible genotypes (heterozygous, homozygous for the alternate allele, homozygous for the reference allele, missing) with each element weighted by error rates specified by the user.". It is unclear whether the values used for weighting are p_rarr, p_aar, etc., and if so, are the actual error rates produced by the tool the same? Would a p_rarr of 0.05 change exactly 5% of heterozygous sites to homozygous reference? Additionally, it would be useful if the tool provided a seed value for reproducibility (if not available yet) when benchmarking other tools.

Overall, the paper is well-written, and the tool provides a valuable resource for the research community.

Is the rationale for developing the new software tool clearly explained? Yes

Is the description of the software tool technically sound? Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others? Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool? Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article? Yes
**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** Computational biology, Bioinformatics, Population genetics, Identity by descent, Artificial Intelligence

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