vcferr: Development, Validation, and Application of a SNP Genotyping Error Simulation Framework

V. P. Nagraj[1], Matthew Scholz[1], Shakeel Jessa[1], Jianye Ge[2], August E. Woerner[2], Bruce Budowle[2], Meng Huang[2], and Stephen D. Turner[1]

[1]**Signature Science, LLC., Austin, TX 78759, USA.**
[2]**Center 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 SNP-based analyses. Simulating various modes and and levels of error can help investigators better understand potential biases caused by miscalled genotypes.

Results: We have developed and validated `vcferr`, a tool to probabilistically simulate genotyping error and missingness in 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 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

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 [3]. However, novel applications of sequencing or genotyping such as those encountered in forensics [4] or ancient DNA [5], or novel downstream data analysis techniques such as imputation on low-pass whole-genome sequencing [6, 7, 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 data (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 VCF [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.

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 [14] and the companion error assessment tool described below. First, we retrieved all genotype calls for chromosome 22 from the 1000 Genomes 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 ~5 minutes on a single CPU. The code used to fully reproduce this validation analysis is described in the Software Availability section below.

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:

$$NRC = 1 - (e_{rr} + e_{ra} + e_{aa})/(e_{rr} + e_{ra} + e_{aa} + m_{ra} + m_{aa})$$

Where e_{rr}, e_{ra}, and e_{aa}, are defined as the the counts of mismatches for the homozygous reference, heterozygous, and homozygous ALT genotypes, and m_{ra}, and m_{aa} are the counts of the matches at the heterozygous and homozygous ALT 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` [13] followed by post-processing results using R. The Dockerfile needed to build the `nrc` image is available under the MIT license at `https://github.com/signaturescience/nrc`, and the Docker image can be pulled directly from the GitHub container registry with `docker pull ghcr.io/signaturescience/nrc`.

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:

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

# Run vcrerr
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
```

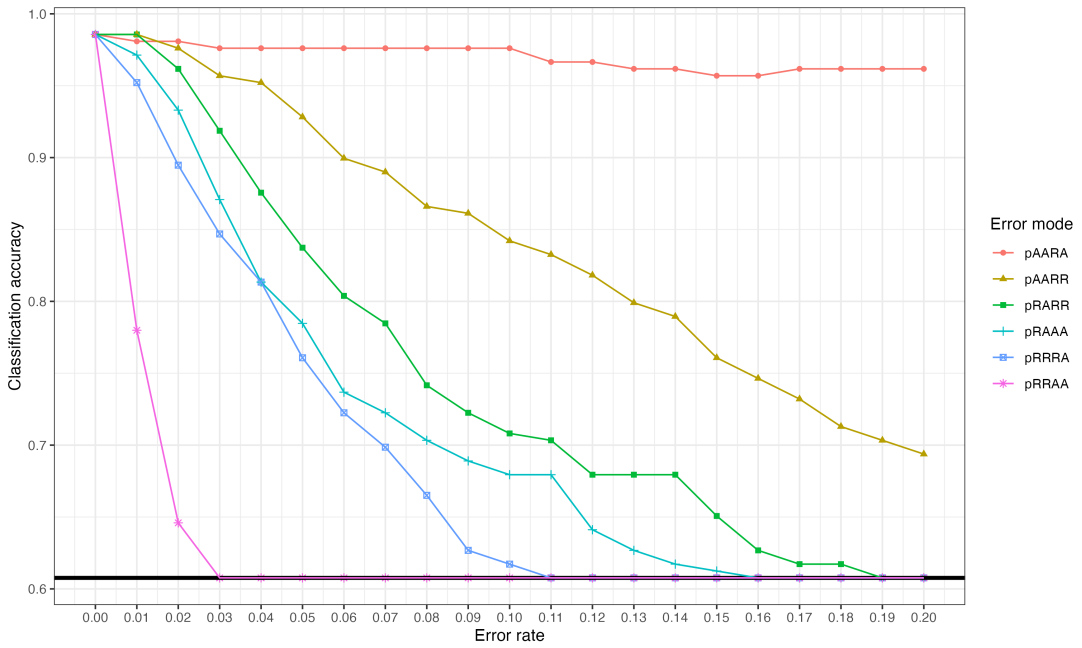


Figure 1. Impact of genotyping error on inferred degree classification accuracy. Each line corresponds to a type of error, with error rates from 0 to 0.2 stepped at 0.01 increments while holding all other types of error constant at 0. The solid black line at bottom represents the accuracy at guessing, which is the floor for relationship degree classification.

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` [12] with founders drawn from the 1000 Genomes GBR population. 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 [15] kinship coefficients from `akt` [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 [17], is available in the Zenodo archive described in the Software Availability section below.

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 [18]. Formalin-fixed, paraffin-embedded (FFPE) tissue fixation is a standard means to preserve tissues from clinical samples, and can result in genotyping error [19] that can vary by type, with heterozygous errors being more commonly discordant [20]. Power and sample size requirements for GWAS are impacted by genotyping error [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. [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.

```
# Loop through id_1, id_2, id_n for sample_to_mutate
vcferr gwas.vcf.gz \
  --sample='sample_to_mutate' \
  --p_rarr=0.05 \
  --p_aara=0.01 \
  --p_rrra=0.01 \
  --p_raaa=0.05 \
  --p_aarr=0.0001 \
  --p_rraa=0.0001 \
  --output_vcf=gwas.mutated.vcf.gz
```

Similarly, a polygenic risk score (PRS) captures the additive effect of hundreds, or perhaps thousands of SNPs, and are computed on genome-wide genotyping data using a prediction model based on summary statistics from external GWAS studies [24]. 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` [12]), 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.

Software availability

vcferr

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

1. Software available from: PyPI (`https://pypi.org/project/vcferr/`) and Bioconda (`https://anaconda.org/bioconda/vcferr`).
2. Source code available from: `https://github.com/signaturescience/vcferr`.
3. Archived source code at time of publication: `https://doi.org/10.5281/zenodo.6611796`.
4. Software license: MIT License.

nrc

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

1. Software (Docker image) available from: `https://ghcr.io/signaturescience/nrc`.
2. Source code available from: `https://github.com/signaturescience/nrc`.
3. Archived source code at time of publication: `https://doi.org/10.5281/zenodo.6611782`.
4. Software license: MIT License.

Code for reproducing the analyses in this paper

Code for reproducing the analyses presented in this paper is available and documented on Zenodo at `https://doi.org/10.5281/zenodo.6599133`.

Author information

VPN, SDT, and MBS developed the software.

All authors contributed to method development.

VPN wrote the first draft of the manuscript.

All authors assisted with manuscript revision.

All authors read and approved the final manuscript.

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.

Acknowledgments

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

References

[1] Rasmus Nielsen, Joshua S. Paul, Anders Albrechtsen, and Yun S. Song. Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics*, 12(6):443–451, June 2011. ISSN 1471-0064. doi: 10.1038/nrg2986.

[2] Sobin Kim and Ashish Misra. SNP genotyping: Technologies and biomedical applications. *Annu. Rev. Biomed. Eng.*, 9: 289–320, 2007.

[3] François Pompanon, Aurélie Bonin, Eva Bellemain, and Pierre Taberlet. Genotyping errors: causes, consequences and solutions. *Nature Reviews. Genetics*, 6(11):847–859, November 2005. ISSN 1471-0056. doi: 10.1038/nrg1707.

[4] Erin M. Gorden, Ellen M. Greytak, Kimberly Sturk-Andreaggi, Janet Cady, Timothy P. McMahon, Steven Armentrout, and Charla Marshall. Extended kinship analysis of historical remains using SNP capture. *Forensic Science International. Genetics*, 57:102636, March 2022. ISSN 1878-0326. doi: 10.1016/j.fsigen.2021.102636.

[5] Ruoyun Hui, Eugenia D'Atanasio, Lara M. Cassidy, Christiana L. Scheib, and Toomas Kivisild. Evaluating genotype imputation pipeline for ultra-low coverage ancient genomes. *Scientific Reports*, 10(1):18542, October 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-75387-w.

[6] Alicia R. Martin, Elizabeth G. Atkinson, Sinéad B. Chapman, Anne Stevenson, Rocky E. Stroud, Tamrat Abebe, Dickens Akena, Melkam Alemayehu, Fred K. Ashaba, Lukoye Atwoli, Tera Bowers, Lori B. Chibnik, Mark J. Daly, Timothy DeSmet, Sheila Dodge, Abebaw Fekadu, Steven Ferriera, Bizu Gelaye, Stella Gichuru, Wilfred E. Injera, Roxanne James, Symon M. Kariuki, Gabriel Kigen, Karestan C. Koenen, Edith Kwobah, Joseph Kyebuzibwa, Lerato Majara, Henry Musinguzi, Rehema M. Mwema, Benjamin M. Neale, Carter P. Newman, Charles R. J. C. Newton, Joseph K. Pickrell, Raj Ramesar, Welelta Shiferaw, Dan J. Stein, Solomon Teferra, Celia van der Merwe, Zukiswa Zingela, and NeuroGAP-Psychosis Consortium. Low-coverage sequencing cost-effectively detects known and novel variation in underrepresented populations. *bioRxiv*, page 2020.04.27.064832, April 2020. doi: 10.1101/2020.04.27.064832.

[7] Jeremiah H. Li, Chase A. Mazur, Tomaz Berisa, and Joseph K. Pickrell. Low-pass sequencing increases the power of GWAS and decreases measurement error of polygenic risk scores compared to genotyping arrays. *bioRxiv*, page 2020.04.29.068452, May 2020. doi: 10.1101/2020.04.29.068452.

[8] Simone Rubinacci, Diogo M. Ribeiro, Robin J. Hofmeister, and Olivier Delaneau. Efficient phasing and imputation of low-coverage sequencing data using large reference panels. *Nature Genetics*, 53(1):120–126, January 2021. ISSN 1546-1718. doi: 10.1038/s41588-020-00756-0.

[9] Heng Li. wgsim, 2011. URL `https://github.com/lh3/wgsim`.

[10] Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. ART: A next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, February 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr708.

[11] Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A. Albers, Eric Banks, Mark A. DePristo, Robert E. Handsaker, Gerton Lunter, Gabor T. Marth, Stephen T. Sherry, Gilean McVean, Richard Durbin, and 1000 Genomes Project Analysis Group. The variant call format and VCFtools. *Bioinformatics*, 27(15):2156–2158, 06 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr330. URL `https://doi.org/10.1093/bioinformatics/btr330`.

[12] Madison Caballero, Daniel N. Seidman, Ying Qiao, Jens Sannerud, Thomas D. Dyer, Donna M. Lehman, Joanne E. Curran, Ravindranath Duggirala, John Blangero, Shai Carmi, and Amy L. Williams. Crossover interference and sex-specific genetic maps shape identical by descent sharing in close relatives. *PLOS Genetics*, 15(12):e1007979, December 2019. ISSN 1553-7404. doi: 10.1371/journal.pgen.1007979. URL `https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1007979`. Publisher: Public Library of Science.

[13] Petr Danecek, James K. Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O. Pollard, Andrew Whitwham, Thomas Keane, Shane A. McCarthy, Robert M. Davies, and Heng Li. Twelve years of SAMtools and BCFtools. *GigaScience*, 10(2):giab008, February 2021. ISSN 2047-217X. doi: 10.1093/gigascience/giab008.

[14] Adam Auton, Gonçalo R. Abecasis, David M. Altshuler, Richard M. Durbin, Gonçalo R. Abecasis, David R. Bentley, Aravinda Chakravarti, Andrew G. Clark, Peter Donnelly, Evan E. Eichler, Paul Flicek, Stacey B. Gabriel, Richard A. Gibbs, Eric D. Green, Matthew E. Hurles, Bartha M. Knoppers, Jan O. Korbel, Eric S. Lander, Charles Lee, Hans Lehrach, Elaine R. Mardis, Gabor T. Marth, Gil A. McVean, Deborah A. Nickerson, Jeanette P. Schmidt, Stephen T. Sherry, Jun Wang, Richard K. Wilson, Richard A. Gibbs, Eric Boerwinkle, Harsha Doddapaneni, Yi Han, Viktoriya Korchina, Christie Kovar, Sandra Lee, Donna Muzny, Jeffrey G. Reid, Yiming Zhu, Jun Wang, Yuqi Chang, Qiang Feng, Xiaodong Fang, Xiaosen Guo, Min Jian, Hui Jiang, Xin Jin, Tianming Lan, Guoqing Li, Jingxiang Li, Yingrui Li, Shengmao Liu, Xiao Liu, Yao Lu, Xuedi Ma, Meifang Tang, Bo Wang, Guangbiao Wang, Honglong Wu, Renhua Wu, Xun Xu, Ye Yin, Dandan Zhang, Wenwei Zhang, Jiao Zhao, Meiru Zhao, Xiaole Zheng, Eric S. Lander, David M. Altshuler, Stacey B. Gabriel, Namrata Gupta, Neda Gharani, Lorraine H. Toji, Norman P. Gerry, Alissa M. Resch, Paul Flicek, Jonathan Barker, Laura Clarke, Laurent Gil, Sarah E. Hunt, Gavin Kelman, Eugene Kulesha, Rasko Leinonen, William M. McLaren, Rajesh Radhakrishnan, Asier Roa, Dmitriy Smirnov, Richard E. Smith, Ian Streeter, Anja Thormann, Iliana Toneva, Brendan Vaughan, Xiangqun Zheng-Bradley, David R. Bentley, Russell Grocock, Sean Humphray, Terena James, Zoya Kingsbury, Hans Lehrach, Ralf Sudbrak, Marcus W. Albrecht, Vyacheslav S. Amstislavskiy, Tatiana A. Borodina, Matthias Lienhard, Florian Mertes, Marc Sultan, Bernd Timmermann, Marie-Laure Yaspo, Elaine R. Mardis, Richard K. Wilson, Lucinda Fulton, Robert Fulton, Stephen T. Sherry, Victor Ananiev, Zinaida Belaia, Dimitriy Beloslyudtsev, Nathan Bouk, Chao Chen, Deanna Church, Robert Cohen, Charles Cook, John Garner, Timothy Hefferon, Mikhail Kimelman, Chunlei Liu, John Lopez, Peter Meric, Chris O'Sullivan, Yuri Ostapchuk, Lon Phan, Sergiy Ponomarov, Valerie Schneider, Eugene Shekhtman, Karl Sirotkin, Douglas Slotta, Hua Zhang, Gil A. McVean, Richard M. Durbin, Senduran Balasubramaniam, John Burton, Petr Danecek, Thomas M. Keane, Anja Kolb-Kokocinski, Shane McCarthy, James Stalker, Michael Quail, Jeanette P. Schmidt, Christopher J. Davies, Jeremy Gollub, Teresa Webster, Brant Wong, Yiping Zhan, Adam Auton, Christopher L. Campbell, Yu Kong, Anthony Marcketta, Richard A. Gibbs, Fuli Yu, Lilian Antunes, Matthew Bainbridge, Donna Muzny, Aniko Sabo, Zhuoyi Huang, Jun Wang, Lachlan J. M. Coin, Lin Fang, Xiaosen Guo, Xin Jin, Guoqing Li, Qibin Li, Yingrui Li, Zhenyu Li, Haoxiang Lin, Binghang Liu, Ruibang Luo, Haojing Shao, Yinlong Xie, Chen Ye, Chang Yu, Fan Zhang, Hancheng Zheng, Hongmei Zhu, Can Alkan, Elif Dal, Fatma Kahveci, Gabor T. Marth, Erik P. Garrison, Deniz Kural, Wan-Ping Lee, Wen Fung Leong, Michael Stromberg, Alistair N. Ward, Jiantao Wu, Mengyao Zhang, Mark J. Daly, Mark A. DePristo, Robert E. Handsaker, David M. Altshuler, Eric Banks, Gaurav Bhatia, Guillermo del Angel, Stacey B. Gabriel, Giulio Genovese, Namrata Gupta, Heng Li, Seva Kashin, Eric S. Lander, Steven A. McCarroll, James C. Nemesh, Ryan E. Poplin, Seungtai C. Yoon, Jayon Lihm, Vladimir Makarov, Andrew G. Clark, Srikanth Gottipati, Alon Keinan, Juan L. Rodriguez-Flores, Jan O. Korbel, Tobias Rausch, Markus H. Fritz, Adrian M. Stütz, Paul Flicek, Kathryn Beal, Laura Clarke, Avik Datta, Javier Herrero, William M. McLaren, Graham R. S. Ritchie, Richard E. Smith, Daniel Zerbino, Xiangqun Zheng-Bradley, Pardis C. Sabeti, Ilya Shlyakhter, Stephen F. Schaffner, Joseph Vitti, David N. Cooper, Edward V. Ball, Peter D. Stenson, David R. Bentley, Bret Barnes, Markus Bauer, R. Keira Cheetham, Anthony Cox, Michael Eberle, Sean Humphray, Scott Kahn, Lisa Murray, John Peden, Richard Shaw, Eimear E. Kenny, Mark A. Batzer, Miriam K. Konkel, Jerilyn A. Walker, Daniel G. MacArthur, Monkol Lek, Ralf Sudbrak, Vyacheslav S. Amstislavskiy, Ralf Herwig, Elaine R. Mardis, Li Ding, Daniel C. Koboldt, David Larson, Kai Ye, Simon Gravel, The 1000 Genomes Project Consortium, Corresponding authors, Steering committee, Production group, Baylor College of Medicine, BGI-Shenzhen, Broad Institute of MIT and Harvard, Coriell Institute for Medical Research, European Bioinformatics Institute European Molecular Biology Laboratory, Illumina, Max Planck Institute for Molecular Genetics, McDonnell Genome Institute at Washington University, US National Institutes of Health, University of Oxford, Wellcome Trust Sanger Institute, Analysis group, Affymetrix, Albert Einstein College of Medicine, Bilkent University, Boston College, Cold Spring Harbor Laboratory, Cornell University, European Molecular Biology Laboratory, Harvard University, Human Gene Mutation Database, Icahn School of Medicine at Mount Sinai, Louisiana State University, Massachusetts General Hospital, McGill University, and NIH National Eye Institute. A global reference for human genetic variation. *Nature*, 526(7571):68–74, October 2015. ISSN 1476-4687. doi: 10.1038/nature15393. URL `https://www.nature.com/articles/nature15393`.

[15] Ani Manichaikul, Josyf C. Mychaleckyj, Stephen S. Rich, Kathy Daly, Michèle Sale, and Wei-Min Chen. Robust relationship inference in genome-wide association studies. *Bioinformatics*, 26(22):2867–2873, 10 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq559. URL `https://doi.org/10.1093/bioinformatics/btq559`.

[16] Rudy Arthur, Ole Schulz-Trieglaff, Anthony J Cox, and Jared O'Connell. AKT: ancestry and kinship toolkit. *Bioinformatics*, 33(1):142–144, 09 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw576. URL `https://doi.org/10.1093/bioinformatics/btw576`.

[17] J. Köster and S. Rahmann. Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 08 2012. doi: 10.1093/bioinformatics/bts480. URL `http://dx.doi.org/10.1093/bioinformatics/bts480`.

[18] Michael D. Gallagher and Alice S. Chen-Plotkin. The post-gwas era: From association to function. *The American Journal of Human Genetics*, 102(5):717–730, 05 2018. doi: 10.1016/j.ajhg.2018.04.002. URL `http://dx.doi.org/10.1016/j.ajhg.2018.04.002`.

[19] M J Sikora, J N Thibert, J Salter, M Dowsett, M D Johnson, and J M Rae. High-efficiency genotype analysis from formalin-fixed, paraffin-embedded tumor tissues. *The Pharmacogenomics Journal*, 11(5):348–358, 06 2010. doi: 10.1038/tpj.2010.50. URL `http://dx.doi.org/10.1038/tpj.2010.50`.

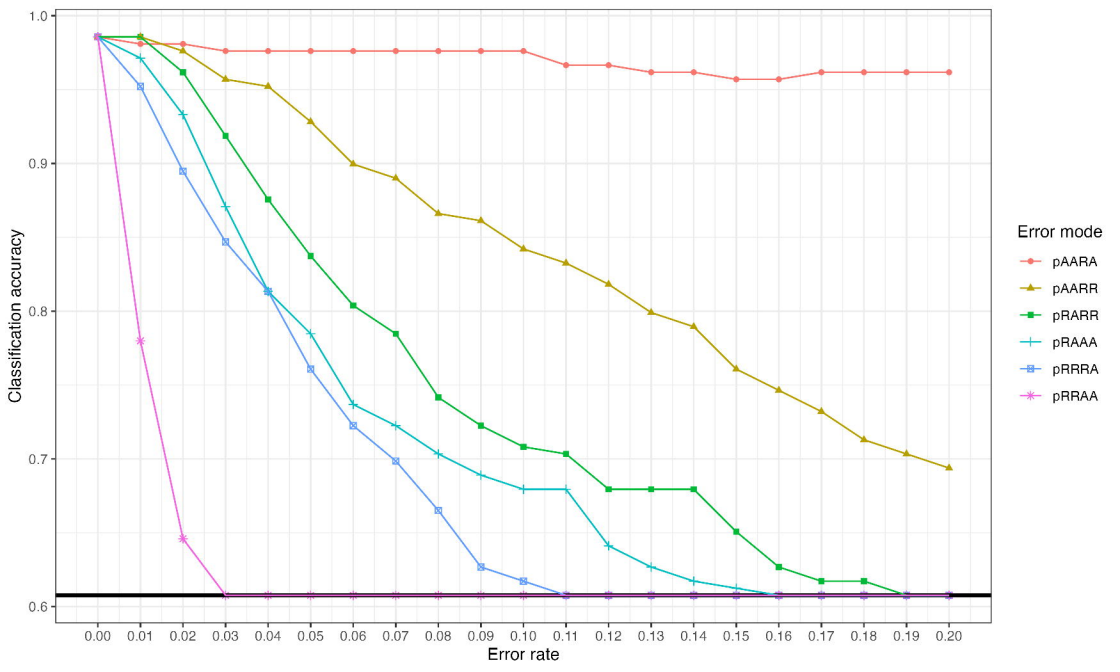
[20] Daniel L. Hertz, Kelley M. Kidwell, Jacklyn N. Thibert, Christina Gersch, Meredith M. Regan, Todd C. Skaar, N. Lynn Henry, Daniel F. Hayes, Catherine H. Van Poznak, and James M. Rae. Genotyping concordance in dna extracted from formalin-fixed paraffin embedded (ffpe) breast tumor and whole blood for pharmacogenetic analyses. *Molecular Oncology*, 9(9):1868–1876, 07 2015. doi: 10.1016/j.molonc.2015.07.002. URL `http://dx.doi.org/10.1016/j.molonc.2015.07.002`.

[21] Lin Hou, Ning Sun, Shrikant Mane, Fred Sayward, Nallakkandi Rajeevan, Kei-Hoi Cheung, Kelly Cho, Saiju Pyarajan, Mihaela Aslan, Perry Miller, Philip D. Harvey, J. Michael Gaziano, John Concato, and Hongyu Zhao. Impact of genotyping errors on statistical power of association tests in genomic analyses: A case study. *Genetic Epidemiology*, 41(2): 152–162, 12 2016. doi: 10.1002/gepi.22027. URL `http://dx.doi.org/10.1002/gepi.22027`.

[22] Mary D Fortune and Chris Wallace. simgwas: a fast method for simulation of large scale case–control gwas summary statistics. *Bioinformatics*, 35(11):1901–1906, 10 2018. doi: 10.1093/bioinformatics/bty898. URL `http://dx.doi.org/10.1093/bioinformatics/bty898`.

[23] You Tang and Xiaolei Liu. G2p: a genome-wide-association-study simulation tool for genotype simulation, phenotype simulation and power evaluation. *Bioinformatics*, 35(19):3852–3854, 02 2019. doi: 10.1093/bioinformatics/btz126. URL `http://dx.doi.org/10.1093/bioinformatics/btz126`.

[24] Ali Torkamani, Nathan E. Wineinger, and Eric J. Topol. The personal and clinical utility of polygenic risk scores. *Nature Reviews Genetics*, 19(9):581–590, 05 2018. doi: 10.1038/s41576-018-0018-x. URL `http://dx.doi.org/10.1038/s41576-018-0018-x`.


Classification accuracy
1.0
0.9
0.8
0.7
0.6
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
0.15
0.16
0.17
0.18
0.19
0.20
Error rate
Error mode
pAARA
pAARR
pRARR
pRAAA
pRRRA
pRRAA