Lessons Learned from Bugs in Models of Human History

Aaron P. Ragsdale,[1] Dominic Nelson,[1] Simon Gravel,[1,3,*] and Jerome Kelleher[2,3,*]

Simulation plays a central role in population genomics studies. Recent years have seen rapid improvements in software efficiency that make it possible to simulate large genomic regions for many individuals sampled from large numbers of populations. As the complexity of the demographic models we study grows, however, there is an ever-increasing opportunity to introduce bugs in their implementation. Here, we describe two errors made in defining population genetic models using the *msprime* coalescent simulator that have found their way into the published record. We discuss how these errors have affected downstream analyses and give recommendations for software developers and users to reduce the risk of such errors.

In the effort to build more realistic simulations of genetic diversity, scientific software developers often focus on computational speed and biological realism. As the models simulated become more realistic, however, they also become more complex and difficult to specify. The interface through which users define their models is therefore increasingly important. Without an intuitive and thoroughly documented interface, it is very difficult to simulate complex population models correctly.

The *msprime* coalescent simulator[1–3] is now widely used in genetics studies. Much of its appeal is the large increase in efficiency over the classical *ms* program,[4] which makes it feasible to simulate large samples of whole chromosomes for the first time. Another distinct advantage of *msprime* is its Python application programming interface (API), which greatly increases the flexibility and ease of use over the standard approach of text-based command line interfaces. In particular, programs like *ms* require users to specify cryptic command line options to describe demographic models. For example, the Gutenkunst et al.[5] demographic model (which is the subject of this note), as written in *ms* syntax, is shown in Figure 1A. This model is relatively simple, and models with many more populations and parameters are increasingly common. Descriptions of such models in *ms* syntax are not easy to comprehend. The Python interface for *msprime*, by contrast, allows the user to state models in a more human-readable and programmatic manner and has many advantages over *ms*' command line interface.

Even when using a high-level programming language like Python, however, implementing multi-population models of demographic history is difficult and prone to error. In this note, we discuss two implementation errors that arose through unfortunate design decisions in *msprime*'s demography API and that then found their way into the scientific record. The first error has relatively mild effects on genetic diversity but was used in many publications, whereas the second error was used only once but had a large impact on the simulation results. In light of these implementation errors, we discuss improvements to *msprime*'s API motivated by these discoveries and, more generally, best practices for implementing and simulating complex multi-population demography.

Case 1: A Misspecified Model in *msprime*'s Documentation

To illustrate the demography API, *msprime* included a description of a widely used three-population Out-of-Africa model[5] as part of its tutorial documentation. In this model (Figure 2A), Eurasian (CEU and CHB) and African (YRI) populations split from each other in the deep past, followed by a more recent split of European and Asian populations with variable rates of continuous migration between each of the populations. Regrettably, the implementation in the *msprime* tutorial was incorrect. Before the time of the split of African and Eurasian populations, when there should have been just a single randomly mating population, migration was allowed to occur between the ancestral population and a second population with size equal to the Eurasian bottleneck size for all time into the past (Figure 2B). This incorrect model was introduced into the tutorial for *msprime* version 0.3.0 and remained in the documentation for around 4 years.

Fortunately, the effects of this error are subtle. Population sizes and structure since the time of the earliest split are unaffected, so differences in expected F_{ST} are negligible between the correct and incorrect models. However, the ancient structure distorts the distribution of early TMRCAs (times to most recent common ancestor) (Figures 2E–2G). The extraneous ancient population increases the long-term effective population size, resulting in roughly 4% excess heterozygosity in contemporary populations compared to the intended model, but the overall effects

[1] McGill Genome Centre and Department of Human Genetics, McGill University, Montréal, QC H2X 3C9, Canada; [2] Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, Oxford OX3 7LF, UK
[3] Senior author
*Correspondence: simon.gravel@mcGill.ca (S.G.), jerome.kelleher@bdi.ox.ac.uk (J.K.)

https://doi.org/10.1016/j.ajhg.2020.08.017.

on patterns of diversity are minimal (Figures 2C and 2D).

Even though the error has a limited effect on simulated data, the tutorial code has been copied many times and used in publications. By searching for some identifying strings from the model definition on GitHub, we found 32 repositories containing either direct copies of the erroneous model code or code that was obviously derived from it (we have opened issues on each of these repositories to alert the authors). In most cases, the publications used simulations to test a non-demographic inference method, such as inferring gene genealogies from sequencing data,[7] determining the age of a genetic variant,[8] or studying the power of variant association tests.[9] In each of these studies, this model was used as an example of a complex population history. Zhou et al.[10] used the incorrect model as an example of how their method for visualizing demographic models can support *msprime* input. Finally, Pfaffelhuber et al.[11] used simulations of the incorrect model demography to evaluate their method for choosing ancestry-informative markers. Given the very subtle effect of the incorrect model on demography (and the fact the method was evaluated with other simulations and real data), it seems unlikely that the model details had any qualitative effect on their conclusions.

This long-standing error could have been prevented by better API design. To model a population split currently in *msprime*, a user must specify a "MassMigration" event that moves lineages from one population to another and then must also remember to turn off migration between those populations at the same time. The demographic events for the correct model are given in Figure 1B. Note that the migration rate change on line 17 was missing in the incorrect model. The release of *msprime* version 1.0 will introduce a "PopulationSplit" event, which more intuitively links the movement of lineages with appropriate changes in migration rates at the time of the split.

A

```
-n 1 1.682020 -n 2 3.736830 -n 3 7.292050
-eg 0 2 116.010723 -eg 0 3 160.246047
-ma x 0.881098 0.561966 0.881098 x 2.797460 0.561966 2.797460 x
-ej 0.028985 3 2 -en 0.028985 2 0.287184
-ema 0.028985 3 x 7.293140 x 7.293140 x x x x x
-ej 0.197963 2 1 -en 0.303501 1 1
```

B

```
1  demographic_events = [
2      # CEU and CHB merge into B with rate changes at T_EU_AS
3      msprime.MassMigration(
4          time=T_EU_AS, source=2, destination=1, proportion=1.0),
5      msprime.MigrationRateChange(time=T_EU_AS, rate=0),
6      msprime.MigrationRateChange(
7          time=T_EU_AS, rate=m_AF_B, matrix_index=(0, 1)),
8      msprime.MigrationRateChange(
9          time=T_EU_AS, rate=m_AF_B, matrix_index=(1, 0)),
10     msprime.PopulationParametersChange(
11         time=T_EU_AS, initial_size=N_B, growth_rate=0,
12         population_id=1),
13     # Population B merges into YRI at T_B
14     msprime.MassMigration(
15         time=T_B, source=1, destination=0, proportion=1.0),
16     # Missing in msprime documentation
17     msprime.MigrationRateChange(time=T_B, rate=0),
18     # Size changes to N_A at T_AF
19     msprime.PopulationParametersChange(
20         time=T_AF, initial_size=N_A, population_id=0)
21 ]
```

C

```
dbg = msprime.DemographyDebugger(
    population_configurations=population_configurations,
    demographic_events=demographic_events,
    migration_matrix=migration_matrix)
dbg.print_history()
ts = msprime.simulate(
    ..., # parameters defining samples, sequence length, etc
    population_configurations=population_configurations,
    demographic_events=demographic_events, # Missing from Ref. 6
    migration_matrix=migration_matrix)
```

D

```
demography = msprime.Demography(
    populations=populations,
    events=demographic_events,
    migration_matrix=migration_matrix)
dbg = demography.debug()
dbg.print_history()
ts = msprime.simulate(
    ..., # parameters defining samples, sequence length, etc
    demography=demography)
```

Figure 1. Implementations of Demography in Coalescent Simulations

(A) The Gutenkunst et al. (2009)[5] human expansion model as written in *ms* syntax.

(B) The same human expansion model with demographic events written in *msprime* syntax. In earlier versions of the *msprime* documentation, the highlighted line 17 was missing, so population structure erroneously continued into the past.

(C) A simulation of this model in Martin et al. (2017)[6] omitted demographic events in the simulation function (highlight). This omission meant that populations remained distinct in the distant past and had excess F_{ST}.

(D) In *msprime* version 1.0, models will be specified so that population configurations and demographic events are coupled, reducing the possibility of making such errors of omission.

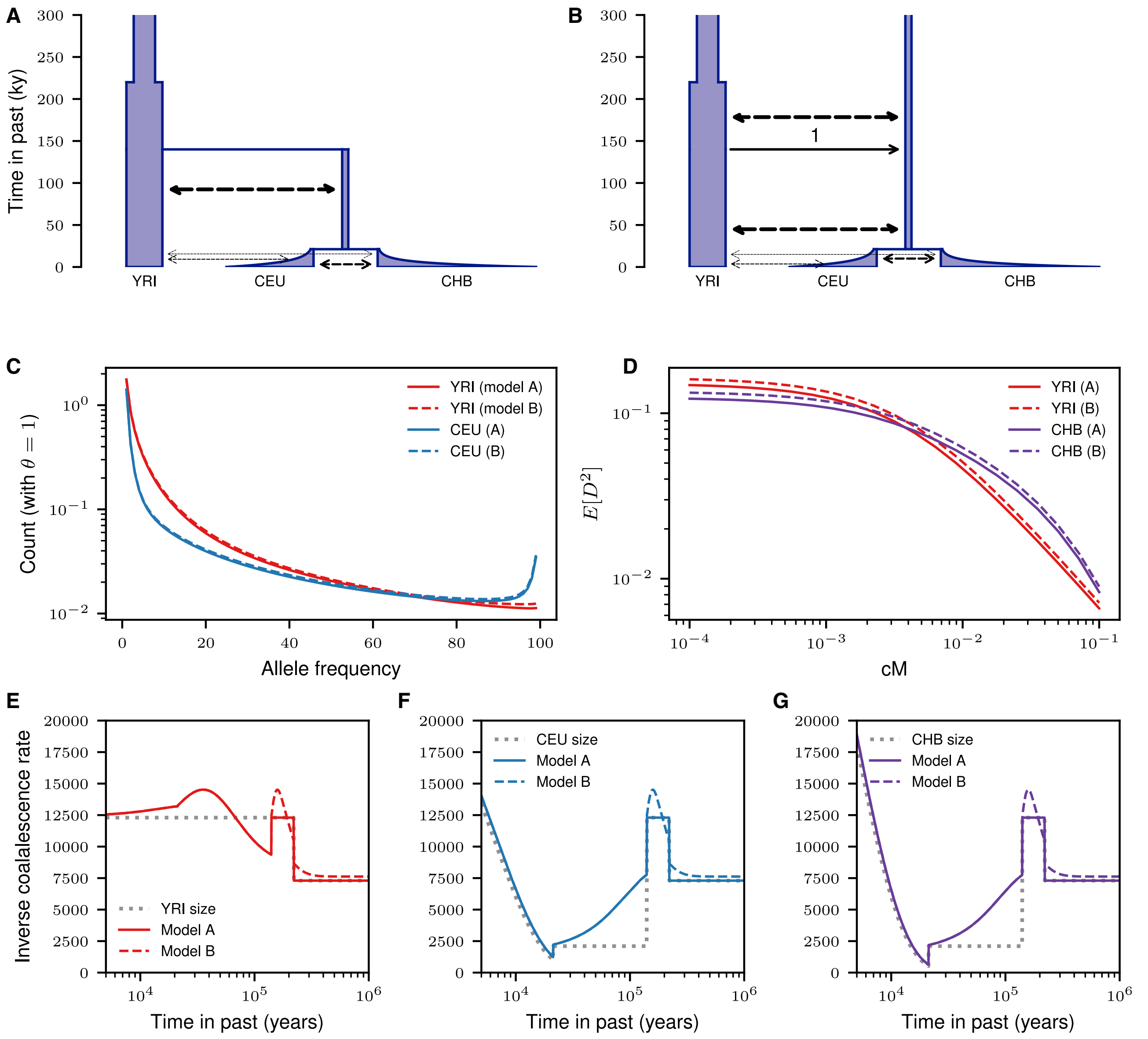

Figure 2. Expected Diversity Statistics under the Gutenkunst et al. Model

(A) The correctly implemented model. Dashed arrows depict continuous migration.

(B) The incorrectly implemented model from the *msprime* tutorial with migration continuing into the past beyond the mass migration event with proportion 1 from the ancestral population to the bottleneck population.

(C) Marginal allele frequency spectra under the two models. Heterozygosity in the incorrect model is inflated by ~ 3.5%, although the general shapes of the distributions are qualitatively similar.

(D) Similarly, the increased heterozygosity leads to excess D^2, although the LD-decay is qualitatively similar between models.

(E–G) We compared the true size history for each population under this model to their expected inverse coalescent rates, which are often interpreted as N_e. The correct and incorrect models (A and B) are equivalent in the recent history, so the inverse coalescence rates only differ in the more distant past. For multi-population demographic history, the inverse coalescence rates are not expected to match the "true" historical size along each branch because population structure and migration between populations change coalescent probabilities over time.

Case 2: Incorrect Model Parameters in an Analysis Pipeline

In another publication using this model,[6] a separate error was introduced: the model itself was defined as suggested in the documentation (with updated parameters from Gravel et al.[12]) and inspected with the *msprime* debugging tools. Despite these initial checks, the simulation was performed without passing the list of demographic events, such as historical changes in size or population splits, so that the three populations never merged and remained separated with low levels of migration (Figure 3A). This leads to a vast overestimate of the divergence across human populations. Although the correct model predicts a mean F_{ST} of $0.05 - 0.10$ across the three populations, the simulated model generated F_{ST} ranging between $0.3 - 0.6$, depending on the populations considered.

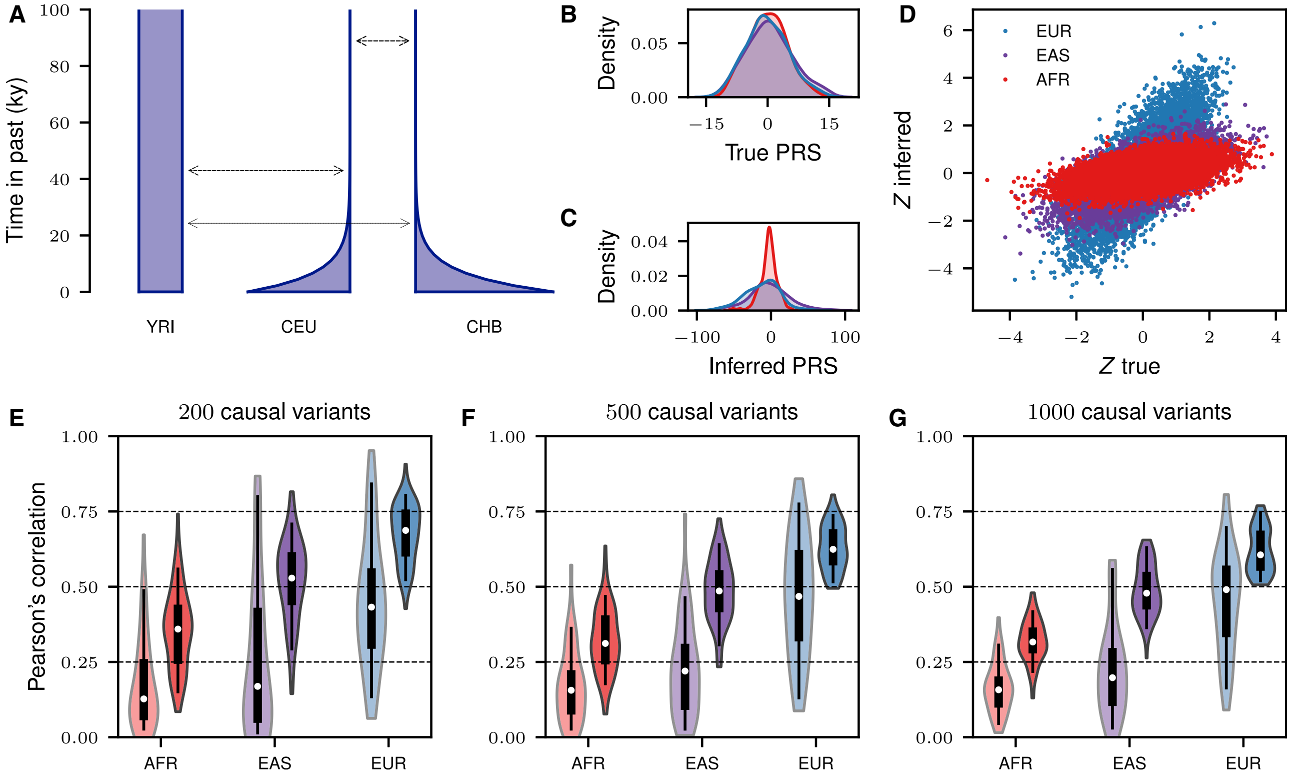

Figure 3. The Transferability of Polygenic Risk Scores (PRSs) under Neutrality
(A) In Martin et al.,[6] the simulated demographic model did not apply demographic events in the past, so continental populations were simulated as isolated with low levels of migration for all time. The correct model has historical events as shown in Figure 2A.
(B–G) We repeated the simulation experiment in Martin et al.[6] by using the correct demographic model. GWAS summary statistics were computed from 10,000 case and control subjects in the European population, and the distribution of inferred PRSs was compared across the three simulated populations. Correlations between true and inferred polygenic scores were computed over 100 simulation replicates. Unlike the original study, we do not observe large differences in mean inferred PRS across the three populations (C and D), and although risk prediction in the African and East Asian populations is still reduced compared to the European population, the reduction in prediction accuracy is not as large as reported in the original study (E–G). (B) and (C) show the distribution of true and inferred PRSs after standardizing PRSs across all individuals in the simulation (this is in contrast to Figure 5 in Martin et al., which plotted unstandardized PRS[6]). Because the polygenic trait was simulated as a threshold trait, it is the relative distribution of PRSs across individuals that determined cases and controls instead of absolute true or inferred PRS. (D) shows a comparison of PRS distributions for a simulation replicate with 1,000 causal variants. For each population in the violin plots, the original correlations from Martin et al.[6] are shown on the left, and correlations via the correct model are shown on the right. For direct comparison to the original study, see Figure 5 in Martin et al.[6] Box plots show median, first and third quartiles, and 95% intervals.

Overall diversity was also strongly affected: heterozygosity was more than doubled in African populations and reduced by more than half in Eurasian populations relative to the correct model.

This simulation was performed to assess the transferability of polygenic risk scores across human populations. In other words, it sought to explore how human demographic history and population structure affect our ability to predict genetic risk in diverse populations given the well-documented unequal representation in medical genetic studies.[13] The resulting publication has been influential in the discussion of health inequalities and genomics: it has been cited over 350 times since 2017. The large excess in divergence under the incorrect model here exaggerated the role of demography and genetic drift in limiting the transferability of genetic risk scores across populations.

Difficulties in transferability remain in the corrected model (Figure 3), although risk prediction in each population is significantly improved (compare to Figure 5 in Martin et al.[6]). Simulations under the correct model indicate that the accuracy of genetic risk scores is still substantially reduced in understudied populations (Figures 3E–3G), supporting one of the main conclusions of Martin et al.[6] However, the reduction is much less pronounced than reported. In particular, we do not observe large differences in mean predicted risk across populations (Figures 3C and 3D) that were present in data and simulations from Martin et al.[6] Thus, a model with continental-scale demographic structure, highly polygenic architecture, and neutral evolution does not appear to explain the large directional biases in mean predicted risk identified in Martin et al.[6]

From a software perspective, this error was easy to make and could have been prevented by better API design. The original *msprime* API requires the user to pass three separate parameters to specify a demographic model (see example in Figure 1C). Note that the same three parameters must be passed to both the debugger and the simulate function. To help prevent such errors, *msprime* version 1.0 will introduce a demography class. The above snippet would then be rewritten (approximately) as shown in Figure 1D. This simple change to the interface makes it much less likely that different models are passed to the debugger and simulator, reducing the potential for error.

Conclusions

The implementation of complex demographic models is error prone, and such errors can have a large impact on downstream analyses and interpretation. The discovery and correction of the demographic models discussed here underscore how API design choice can lead to the propagation of mistakes that are difficult to notice. There are myriad other ways to get models slightly or very wrong. Migration rates can be set in the wrong direction, for example, and parameters can be rescaled incorrectly when a new mutation rate is used. We therefore recommend the following steps to ensure more robust simulations.

First, if possible, we recommend using the *stdpopsim*[14] library, which removes the need to re-implement demographic models from scratch. This is a "standard library" of quality-controlled models and simulation resources for a growing number of commonly studied species. The resource is built around an open-source community development model with rigorous code review and quality control (QC) procedures. For example, when a contributor adds a new model, it is not fully integrated into the catalog until a second, entirely independent, implementation of the model is provided by another developer. If the original and "QC" versions of the model are precisely equal, then we can have a reasonable degree of confidence in correctness. It is through this QC process that we discovered the misspecified model in *msprime*'s documentation. However, *stdpopsim* has QC-verified models available for only a handful of published models and species, so it might not meet the needs of a particular simulation study. If the required model is not present in *stdpopsim*, then extra care should be taken to validate the implementation. This could include verification through code review or comparison with an independent implementation of the model. If the model is of wider interest to the community, we would encourage developers to contribute it to *stdpopsim*; at the very least, the QC procedures in place provide a strong reassurance of correctness.

Second, regardless of whether the model has been implemented locally or obtained from a resource like *stdpopsim*, basic statistical validation should *always* be performed on the results. Errors are all too easy to make, and an analysis of the basic statistical properties of the simulations is essential due diligence. For example, a brief analysis of F_{ST} values most likely would have prevented the error in Martin et al.[6] It is understandable that such analyses were not undertaken in this case because processing such a huge dataset (600,000 samples) was highly challenging. Recent progress has made statistical analysis at this scale much easier[15] so that F_{ST}, heterozygosity, the allele frequency spectrum, or the distribution of coalescence times can be rapidly computed. However, the problems would have been apparent if pilot simulations with smaller sample sizes were undertaken and analyzed because simulations and analysis of small samples sizes are very efficient. Graphical inspection of demographic models can also help identify issues, especially if visualization can be automated. The demography debugging tool in *msprime* summarizes demographic events occurring over time, and various efforts are underway to facilitate the specification and visualization of population genetics models, such as Zhou et al.[10] and the *demography* package used in this paper.

Finally, openness is essential to the self-correcting nature of science. We only know about these errors because of open code and open-source development processes. By making their entire pipeline available, Martin et al.[6] not only enabled other research teams to build upon their findings, but they also made it possible for errors to be found and corrected. There must be many, many more mistakes out there, and we need both pre- and post-publication vigilance from users and developers to ensure the soundness of the large body of simulation-based analyses.

Data and Code Availability

We computed the expected allele frequency spectrum (AFS) by using *moments* version 1.0.3[16] and LD-decay curves by using *moments.LD*.[17] F_{ST} and other diversity statistics were computed from the expected AFS and verified with branch statistics from the output of *msprime* simulations using *tskit* version 0.2.3.[15] Demographic models were plotted with the *demography* package written for Python (https://github.com/apragsdale/demography, version 0.0.3). The *demography* package allows users to define a demographic model as a directed acyclic graph using *networkx*,[18] which can be plotted as in Figures 2A, 2B, and 3A. *Demography* also can translate the graph-based demography into *msprime* input commands or simulate the AFS or two-locus statistics by using *moments*[16,17] or *dadi*.[5] For the analysis of polygenic risk, we used the original pipeline from Martin et al.[6] available from https://github.com/armartin/ancestry_pipeline/blob/master/simulate_prs.py. We updated the pipeline to run with the correct demographic parameters and more recent versions of *msprime* and *tskit*, and the updated pipeline is available at https://github.com/apragsdale/PRS. Data and Python scripts to recreate Figures 2 and 3 can be found at https://github.com/jeromekelleher/msprime-model-errors. A full list of the GitHub repositories containing copies of the erroneous model are also given here.

Acknowledgments

We would like to thank Alicia Martin for comments on the manuscript and for providing the simulation data from their study reproduced in Figure 3. We would also like to thank Chris Gignoux, Ilan Gronau, and John Novembre for helpful comments on the manuscript. J.K. would like to apologize to those affected by the error in the example Out-of-Africa model and by the poor design decisions in *msprime*'s demography API. This research is funded in part by the Canada Research Chair Program, CIHR project grant MOP 136855, and an NSERC Discovery grant to S.G. J.K. is supported by the Robertson Foundation.

References

1. Kelleher, J., Etheridge, A.M., and McVean, G. (2016). Efficient coalescent simulation and genealogical analysis for large sample sizes. PLoS Comput. Biol. *12*, e1004842.
2. Nelson, D., Kelleher, J., Ragsdale, A.P., Moreau, C., McVean, G., and Gravel, S. (2020). Accounting for long-range correlations in genome-wide simulations of large cohorts. PLoS Genet. *16*, e1008619.
3. Kelleher, J., and Lohse, K. (2020). Coalescent simulation with msprime. In Statistical Population Genomics, J.Y. Dutheil, ed. (New York, NY: Springer US), pp. 191–230.
4. Hudson, R.R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics *18*, 337–338.
5. Gutenkunst, R.N., Hernandez, R.D., Williamson, S.H., and Bustamante, C.D. (2009). Inferring the joint demographic history of multiple populations from multidimensional snp frequency data. PLoS Genet. *5*, e1000695.
6. Martin, A.R., Gignoux, C.R., Walters, R.K., Wojcik, G.L., Neale, B.M., Gravel, S., Daly, M.J., Bustamante, C.D., and Kenny, E.E. (2017). Human demographic history impacts genetic risk prediction across diverse populations. Am. J. Hum. Genet. *100*, 635–649.
7. Kelleher, J., Wong, Y., Wohns, A.W., Fadil, C., Albers, P.K., and McVean, G. (2019). Inferring whole-genome histories in large population datasets. Nat. Genet. *51*, 1330–1338.
8. Albers, P.K., and McVean, G. (2020). Dating genomic variants and shared ancestry in population-scale sequencing data. PLoS Biol. *18*, e3000586.
9. Tong, D.M.H., and Hernandez, R.D. (2020). Population genetic simulation study of power in association testing across genetic architectures and study designs. Genet. Epidemiol. *44*, 90–103.
10. Zhou, Y., Tian, X., Browning, B.L., and Browning, S.R. (2018). POPdemog: visualizing population demographic history from simulation scripts. Bioinformatics *34*, 2854–2855.
11. Pfaffelhuber, P., Grundner-Culemann, F., Lipphardt, V., and Baumdicker, F. (2020). How to choose sets of ancestry informative markers: A supervised feature selection approach. Forensic Sci. Int. Genet. *46*, 102259.
12. Gravel, S., Henn, B.M., Gutenkunst, R.N., Indap, A.R., Marth, G.T., Clark, A.G., Yu, F., Gibbs, R.A., and Bustamante, C.D. (2011). 1000 Genomes Project. Demographic history and rare allele sharing among human populations. Proc. Natl. Acad. Sci. USA *108*, 11983–11988.
13. Popejoy, A.B., and Fullerton, S.M. (2016). Genomics is failing on diversity. Nature *538*, 161–164.
14. Adrion, J.R., Cole, C.B., Dukler, N., Galloway, J.G., Gladstein, A.L., Gower, G., Kyriazis, C.C., Ragsdale, A.P., Tsambos, G., Baumdicker, F., et al. (2020). A community-maintained standard library of population genetic models. eLife *9*, e54967.
15. Ralph, P., Thornton, K., and Kelleher, J. (2020). Efficiently summarizing relationships in large samples: a general duality between statistics of genealogies and genomes. Genetics *215*, 779–797.
16. Jouganous, J., Long, W., Ragsdale, A.P., and Gravel, S. (2017). Inferring the joint demographic history of multiple populations: beyond the diffusion approximation. Genetics *206*, 1549–1567.
17. Ragsdale, A.P., and Gravel, S. (2019). Models of archaic admixture and recent history from two-locus statistics. PLoS Genet. *15*, e1008204.
18. Hagberg, A., Swart, P., and Schult, D. (2008). Exploring network structure, dynamics, and function using networkx (Los Alamos National Lab).