The Journal of Open Source Software

REGENS: an open source Python package for simulating realistic autosomal genotypes

John T. Gregg[1], **Trang T. Le**[1], **and Jason H. Moore**[2]

1 Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, Philadelphia, PA 19087, USA **2** Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19087, USA

DOI: 10.21105/joss.02743

Software

- Review
- Repository
- Archive

Editor: Frederick Boehm

Reviewers:

- @raivivek
- @dwinter

Submitted: 01 October 2020
Published: 04 March 2021

Summary

REcombinatory Genome ENumeration of Subpopulations (REGENS) is an open source Python package that simulates autosomal genotypes by concatenating real individuals' genomic segments in a way that preserves their linkage disequilibrium (LD), which is defined as statistical associations between alleles at different loci (Slatkin, 2008). Recombining segments in a way that preserves LD simulates autosomes that closely resemble those of the real input population (Shi, 2018) because real autosomal genotypes can be accurately modeled as genomic segments from a finite pool of heritable association structures (LD haplotypes) (Druet, 2009). REGENS can also simulate mono-allelic and epistatic single nucleotide variant (SNV) effects of any order without perturbing the simulated LD pattern. The SNVs involved in an effect can contribute additively, dominantly, recessively, only if heterozygous, or only if homozygous. All simulated effects contribute to the value of either a binary or continuous biological trait (phenotype) with a specified mean value and a specified amount of random noise.

Statement of need

The goal of most genome-wide association studies (GWAS) is to identify associations between single nucleotide variants (SNVs) and a phenotype to inform researchers and clinicians about potentially causative genetic factors. Completing this task will require overcoming numerous challenges such as insufficient sample sizes and over-representation of European ancestries (Torkamani et al., 2018). Computational biologists build machine learning models that look for genetic associations in such unconventional datasets, but the majority of genetic associations have yet to be discovered (Nolte, 2017). Researchers can use simulated datasets with known ground truths to assess the effectiveness of an algorithm, such as the power to detect epistatic effects with dimensionality reduction techniques (Moore, 2017). The more closely simulated data matches real-world data, the more accurate such test results will be. Since humans of different ancestry have different LD patterns (Eberle, 2006), a simulation that can replicate those patterns from a small number of real samples is desirable. Therefore, intended users of REGENS are computational biologists who aim to test a statistical learning model on simulated GWAS data with precise realistic LD patterns.

Algorithm overview

Two genomic segments are said to be in low LD if alleles are approximately uncorrelated between the two segments, which is guaranteed to occur if the boundary separating the

segments has a sufficiently high recombination rate. If two genomic segments from randomly sampled individuals are concatenated in silico at a boundary with a high recombination rate (the position of which is referred to as a breakpoint from here on), then the LD pattern of the resultant in-silico autosomal genotypes will change minimally (Shi, 2018). To illustrate this point, let us let $P(R_i = 1)$ be the probability of *observing* a recombination event at the i^{th} genomic position. The following holds:

$$P(R_i = 1) = 1 \times P(R_i = 1) + 0 \times P(R_i = 0) = E[R_i], \tag{1}$$

hence,

$$\frac{P(R_i = 1)}{\sum_i P(R_i = 1)} = \frac{E[R_i]}{\sum_i E[R_i]}. \tag{2}$$

Drawing simulated breakpoints from the right hand side of (Equation 2) is like drawing differently colored marbles from a jar. Just as the color composition inferred from drawing (with replacement) a marble from a jar many times approaches the true distribution of colors, the sample of simulated segment recombinations learned from drawing breakpoints for many simulated individuals approaches the input population's empirical distribution of real recombination events. Genomic segments that only contain alleles in high LD are rarely separated by breakpoints, which retains the original LD pattern (Figure 1).

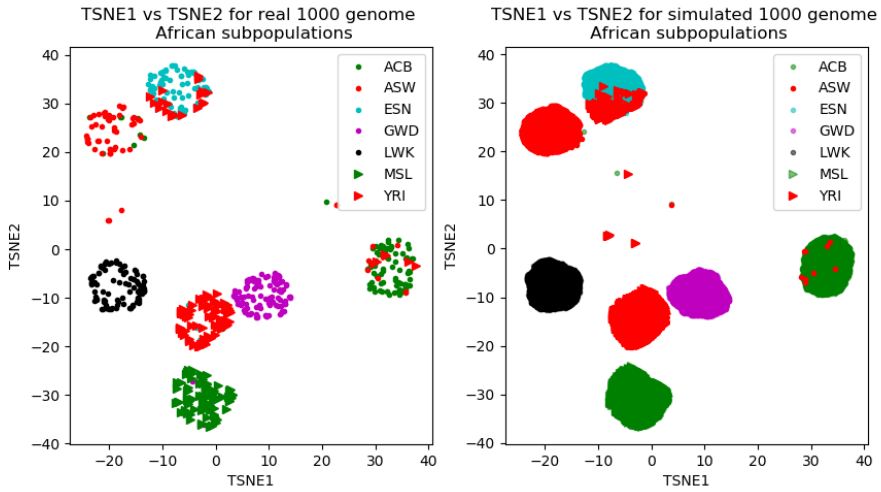

Figure 1: Comparison of population whole genomes in 2 dimensional TSNE space.

Differentiating attributes

Many packages were built to simulate genetic data with different goals in mind. Genetic Architecture Model Emulator for Testing and Evaluating Software (GAMETEs) simulates simple and epistatic SNV/phenotype associations quickly but ignores LD patterns (Urbanowicz, 2012). Genome Simulation of Linkage and Association (GenomeSIMLA) uses forward time simulation to produce broadly realistic LD patterns. However, these patterns do not exactly match those of a particular dataset (Ritchie, 2015). Triadsim (Shi, 2018) replicates exact LD patterns, but it requires (mother, father, kin) trios and takes an average CPU-time of 6.8 hours and an average peak RAM of 54.6 GB to simulate 10000 trios (20000 unrelated GWAS samples) with 4 breakpoints. REGENS uses the same recombination principles that Triadsim relies on, but it is 88.5 times faster (95% CI (75.1, 105.0) via bootstrapping) and requires 6.2 times lower peak RAM (95% CI (6.04, 6.33) via bootstrapping) on average over 10 replicate simulations (Intel(R) Xeon(R) CPU E5-2690 v4 2.60GHz processor). REGENS also recombines individuals

instead of trios to simulate GWAS data with small publicly available genomic datasets, such as those in the 1000 Genomes project. This fact allows REGENS to accurately simulate the full genetic diversity of the world's population (representative figures are in the supplementary analysis). Finally, REGENS can simulate continuous and binary phenotypes that depend on any linear combination of products of f(SNV) values, where f transforms the standard SNP values of $\{0, 1, 2\}$ to represent nonlinear monoallelic effects (such as dominance). Example implementations of these features are in REGENS' GitHub repository.

Supplementary analysis

Figures that demonstrate the similarity between real and simulated populations for all twenty-six 1000 genomes populations, as well as the methods that were used to create those figures, are here https://github.com/EpistasisLab/regens-analysis

Inspiration and dependencies

REGENS was inspired by Triadsim's idea to draw simulated breakpoints at locations with higher recombination rates, as well as well as by GAMETE's objective of simulating data quickly. REGENS relies on bed-reader, a spinoff of PySnpTools's core .bed file code (Nicolo Fusi., https://fastlmm.github.io/), to optimally read re-sampled rows from plink bed files as 8 bit integers and then write the 8 bit integer simulated autosomal genotypes into new bed files. REGENS also relies on the 1000 genomes project's whole genomes from 26 distinct sub-populations (Gibbs, 2015), and it relies on those populations' corresponding genome-wide sex-averaged recombination rates inferred by the pyrho algorithm (Spence, 2019).

Acknowledgements

We acknowledge contributions from Carl Kadie, who developed PySnpTools, for implementing its ability to read and write plink bed files as 8 bit integers. This work was supported by NIH grant LM010098.

References

Druet, T. et al. (2009). A hidden markov model combining linkage and linkage disequilibrium information for haplotype reconstruction and quantitative trait locus fine mapping. *Genetics*, *184*, 789–798. https://doi.org/10.1534/genetics.109.108431

Eberle, M. et al. (2006). Allele frequency matching between SNPs reveals an excess of linkage disequilibrium in genic regions of the human genome. *PLoS Genetics*, *2*, e142. https://doi.org/10.1371/journal.pgen.0020142

Gibbs, R. et al. (2015). A global reference for human genetic variation. *Nature*, *526*, 68–74. https://doi.org/doi:10.1038/nature15393

Moore, J. et al. (2017). Grid-based stochastic search for hierarchical gene-gene interactions in population-based genetic studies of common human diseases. *BioData Mining*, *10*. https://doi.org/10.1186/s13040-017-0139-3

Nicolo Fusi., C. K. &. (https://fastlmm.github.io/). *PySnpTools & bed-reader, 2020*.

Nolte, I. et al. (2017). Missing heritability: Is the gap closing? An analysis of 32 complex traits in the lifelines cohort study. *European Journal of Human Genetics*, *25*, 877–885. https://doi.org/10.1038/ejhg.2017.50

Ritchie, M. D. et al. (2015). Generating linkage disequilibrium patterns in data simulations using genomeSIMLA. *Lecture Notes in Computer Science*, *526*, 24–35. https://doi.org/10.1007/978-3-540-78757-0_3

Shi, M. et al. (2018). Simulating autosomal genotypes with realistic linkage disequilibrium and a spiked-in genetic effect. *BMC Bioinformatics*, *19*. https://doi.org/10.1186/s12859-017-2004-2

Slatkin, M. (2008). Linkage disequilibrium—understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, *9*(6), 477–485. https://doi.org/10.1038/nrg2361

Spence, J. et al. (2019). Inference and analysis of population-specific fine-scale recombination maps across 26 diverse human populations. *Science Advances*, *5*. https://doi.org/10.1126/sciadv.aaw9206

Torkamani, A., Wineinger, N. E., & Topol, E. J. (2018). The personal and clinical utility of polygenic risk scores. *Nature Reviews Genetics*, *19*(9), 581–590. https://doi.org/10.1038/s41576-018-0018-x

Urbanowicz, R. et al. (2012). Epistatic models with random architectures. *BioData Mining*, *5*. https://doi.org/10.1186/1756-0381-5-16