

# GWAsimulator: A rapid whole-genome simulation program

Chun Li<sup>1,\*</sup>, Mingyao Li<sup>2</sup>

<sup>1</sup>Department of Biostatistics, Vanderbilt University School of Medicine, Nashville, TN 37232, USA; <sup>2</sup>Department of Biostatistics and Epidemiology, University of Pennsylvania School of Medicine, Philadelphia, PA 19104, USA.

## ABSTRACT

**Summary:** GWAsimulator implements a rapid moving-window algorithm to simulate genotype data for case-control or population samples from genomic SNP chips. For case-control data, the program generates cases and controls according to a user-specified multi-locus disease model, and can simulate specific regions if desired. The program uses phased genotype data as input and has the flexibility of simulating genotypes for different populations and different genomic SNP chips. When the HapMap phased data are used, the simulated data have similar local LD patterns as the HapMap data. As genome-wide association (GWA) studies become increasingly popular and new GWA data analysis methods are being developed, we anticipate that GWAsimulator will be an important tool for evaluating performance of new GWA analysis methods.

**Availability:** The C++ source code, executables for Linux, Windows, and MacOS, manual, example data sets and analysis program are available at <http://biostat.mc.vanderbilt.edu/GWAsimulator>.  
**Contact:** chun.li@vanderbilt.edu

## 1 INTRODUCTION

Genome-wide association (GWA) studies have become an important tool for discovering susceptibility genes for complex diseases. This has led to great needs in development and evaluation of new methodologies for GWA studies. As association mapping relies on linkage disequilibrium (LD) between disease and marker loci, a key issue in method evaluation is to simulate data that have similar LD patterns as seen in practice. A popular algorithm for simulating genomic data is based on the coalescent approach (Kingman 1982; Hudson 1983, 1990; Donnelly and Tavaré 1995) in which DNA sequences are simulated from a theoretical population. Programs that implement this algorithm have been extensively used for method evaluation (Hudson 2002; Schaffner et al. 2005; Liang et al. 2007). However, these programs are generally slow for simulating whole genome genotype data such as those from SNP chips. To improve simulation speed, we implement a moving-window algorithm (Durrant et al. 2004) that is different from the coalescent and simulates whole-genome data based on a set of phased input data.

## 2 METHODS

The program can generate unrelated case-control (sampled retrospectively conditional on affection status) or population (sampled randomly) data of genome-wide SNP genotypes with patterns of LD similar to the input data.

### 2.1 Phased input data and control file

The program requires phased data as input. If the HapMap data are used, the number of phased autosomes and X chromosomes are 120 and 90 for both CEU and YRI, 90 and 68 for CHB, and 90 and 67 for JPT. Additional

parameters needed by the program should be provided in a control file, including disease model (see 2.2), window size (see 2.3), whether to output the simulated data (see 2.4), and the number of subjects to be simulated.

### 2.2 Determination of disease model

For simulations of case-control data, a disease model is needed. The program allows the user to specify disease model parameters, including disease prevalence, the number of disease loci, and for each disease locus, its location, risk allele, and genotypic relative risk. If the user wants to simulate specific regions, the start and end positions need to be specified. The risk allele frequencies can then be calculated based on the input data. For a disease model with  $m$  disease loci, let  $g_i = 0, 1, 2$  denote the number of copies of the risk allele at SNP  $i$  ( $i = 1, \dots, m$ ). For genotype  $G = \{g_1, \dots, g_m\}$ , let  $f(G) = \Pr(\text{affected} | G)$  denote the penetrance. The program assumes the penetrance is a function of the genotypes such that  $\text{logit}[f(G)] = \alpha + \beta_1 g_1 + \dots + \beta_m g_m$ , where  $\text{logit}(x) = \ln[x/(1-x)]$ . The values of  $\alpha$  and  $\beta_i$  will be determined by the program so that the model's genotypic relative risks and prevalence agree with those specified by the user. The current version of the program allows one disease variant per chromosome.

### 2.3 Simulation algorithm

For simulations of population samples, no disease model is involved, and the program assumes all chromosomes are non-disease chromosomes (see below). For simulations of case-control data, once the disease model is determined, the program calculates the conditional probabilities  $\Pr(G | \text{case})$  and  $\Pr(G | \text{control})$  over all disease locus genotypes given the subject's affection status, and then generates disease locus genotypes for cases and controls according to these conditional probabilities. This retrospective approach is different from prospective simulation schemes in which a joint genotype  $\{g_1, \dots, g_m\}$  is simulated and is kept or discarded depending on whether a random number is smaller or larger than the penetrance of the genotype. Therefore, compared to prospective simulation schemes, especially for disease models with a small prevalence, our retrospective sampling approach is more efficient.

After the disease locus genotypes are generated, the program then simulates genotypes for the other SNPs on the disease chromosomes using a moving-window algorithm (Durrant et al. 2004). We assume all SNPs follow Hardy-Weinberg equilibrium in the general population. For each disease chromosome, the two alleles at the disease locus, say  $d$ , serve as the starting points for growing the two copies of the chromosome. For each copy, the program randomly selects a five-SNP haplotype at loci  $[d-2, d+2]$  from the input phased data that has the same allele as the already simulated allele at  $d$ . The program then gradually grows the whole chromosome as follows: for SNPs on the right side of the disease locus, it generates an allele at locus  $d+i$  given the haplotype at  $[d+i-4, d+i-1]$  for  $i \geq 3$ ; the conditional probabilities for the alleles at locus  $d+i$  given the haplotype at  $[d+i-4, d+i-1]$  are determined based on the input phased data. Similarly, for SNPs on the left side of the disease locus, it generates an allele at locus  $d-i$  given the haplotype at  $[d-i+1, d-i+4]$  for  $i \geq 3$ . Genotypes for non-disease chromosomes are generated similarly except that a randomly selected SNP is designated as the starting SNP. In this algorithm, every four consecutive SNPs are used to determine the allele at the next SNP, but the window size can be modified by the user in the control file.

The simulated chromosomes generated by this algorithm are not exact

\*To whom correspondence should be addressed.

copies of those in the original input data. Rather, the input phased chromosomes are used to generate plausible haplotypes in a wider population that have a similar local LD structure as the input phased data.

## 2.4 Output options

The program can be easily built upon with user's programs for further data analysis. This avoids saving the simulated data to files, which can be time consuming. If the user chooses to output data, 23 files, one for each chromosome will be generated and then compressed to save disk space. The current version of the program offers three data output options: genotype format (each row is a person), phased data format (a person has two rows, each being a phased chromosome) or linkage format (each row is a person, with six columns for pedigree information followed by genotype data).

## 3 RESULTS

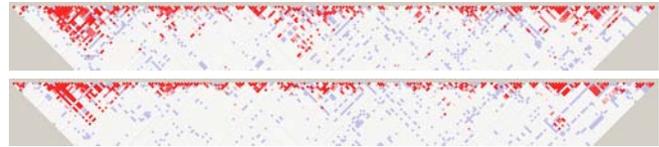
To evaluate our simulation algorithm, we used HapMap phased data as input and compared with the simulated data. We obtained SNP names and positions for the Illumina HumanHap300 chip. After discarding SNPs that are not in the HapMap CEU phased data, 314,174 SNPs remained and were used for simulations. As an example, Figure 1 shows LD patterns of the HapMap CEU dataset and a simulated dataset of 60 unrelated individuals for a 200-SNP region on chromosome 22. The figure clearly indicates the similarity of short-range LD patterns between the two data sets. We also compared LD patterns using LD unit (LDU) maps (Maniatis et al. 2002). Using the SNPs on the HumanHap300, we constructed LDU maps for the HapMap CEU samples and for five simulated data sets of 60 unrelated individuals. The profiles of the LDU maps were very similar, although the LDU maps for the simulated data sets were longer in overall length (Supplementary Figure 1).

Our results indicate that the simulation algorithm can well preserve short-range LD but lacks the capability to preserve long-range LD. However, we note that power of association analysis mainly depends on the local LD around the disease loci, which are similar between the HapMap and the simulated data.

GWAsimulator is fast. For example, when simulating genotype data for Illumina HumanHap300 for  $N$  cases and  $N$  controls, on an Intel Xeon E5345 CPU (2.33 GHz, 32-bit Linux 2.6.18, g++ 4.1.1), it took 9.2 minutes (202 Mb memory) when  $N=1000$ , and 36.7 minutes (665 Mb) when  $N=4000$ . For the HumanHap550, it took 13.9 (307 Mb) and 55.7 minutes (1081 Mb), respectively.

## 4 DISCUSSION

GWAsimulator is written in C++ and can be ported to a variety of operating systems. Executables are available for Windows, Linux, and Mac OS X. Our simulation algorithm faithfully follows the local LD structure of the input phased data. Switching to a different population or SNP chip requires a simple change of input files. The program is efficient in several aspects. Simulated data are internally stored as bit vectors, which minimizes the amount of memory and allows simulation of a large sample size. Unlike prospective simulation algorithms, GWAsimulator samples cases and controls retrospectively and avoids throwing data away. In addition, there is no need to store a large pool of population chromosomes to sample from. Compared to prospective and coalescent based algorithms, GWAsimulator is faster, making it feasible for evaluating the performance of GWA analysis methods through realistic simulations. The program is also easy to be built upon with user's data analysis functions; this avoids saving whole-



**Figure 1.** Comparison of LD between HapMap CEU samples (top) and a simulated data set of 60 unrelated individuals (bottom). Displayed are Haploview (Barrett et al. 2005) plots on 200 SNPs on chromosome 22.

genome data to files, which can be time consuming.

Because the program relies on large-scale genotyping data to provide local LD patterns, any limitations of the input data may be passed on to the simulated data, such as ascertainment bias (Clark et al. 2005) if the HapMap data are used, despite its demonstrated similarity of LD patterns with other samples (Willer et al. 2006). Although the program can use other sources of input data when available, currently it might not be useful for populations that have not been extensively genotyped. If the input data are not variable enough due to bias or small sample size, the generated data might not show enough variability for the population under study. The program also requires the disease loci to be known. They can be selected from the source database or created by the user. Creating loci, however, requires complete phase information between the disease loci and other markers, which may not be available.

The program has been successfully used in methodology development (Li et al. 2008). As GWA studies become increasingly popular, we anticipate that GWAsimulator will become an important tool for evaluating performance of GWA analysis methods.

## ACKNOWLEDGEMENTS

We thank Weihua Guan for helping write an earlier version and three anonymous reviewers for helpful critiques. This work was supported by the University Research Foundation grant and the McCabe Pilot Award from the University of Pennsylvania (to ML).

*Conflict of Interest: none declared.*

## REFERENCES

- Barrett, J.C. et al. (2005) Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, 21, 263-265.
- Clark, A.G. et al. (2005) Ascertainment bias in studies of human genome-wide polymorphism. *Genome Res.*, 15, 1496-1502.
- Donnelly, P. and Tavaré, S. (1995) Coalescents and genealogical structure under neutrality. *Annu. Rev. Genet.*, 29, 401-421.
- Durrant, C. et al. (2004) Linkage disequilibrium mapping via clastic analysis of single-nucleotide polymorphism haplotypes. *Am. J. Hum. Genet.*, 75, 35-43.
- Hudson, R.R. (1983) Properties of a neutral allele model with intragenic recombination. *Theor. Popul. Biol.*, 23, 183-201.
- Hudson, R.R. (1990) Gene genealogies and the coalescent process. *Oxford Surveys in Evolutionary Biology*, 7, 1-44.
- Hudson, R.R. (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18, 337-378.
- Kingman, J.F.C. (1982) On the genealogy of large populations. *J Appl Probab.* 19A, 27-43.
- Li, C. et al. (2008) Prioritized subset analysis: Improving power in genome-wide association studies. *Hum. Hered.*, 65, 129-141.
- Liang, L. et al. (2007) GENOME: a rapid coalescent-based whole genome simulator. *Bioinformatics*, 23, 1565-1567.
- Maniatis, N. et al. (2002) The first linkage disequilibrium (LD) maps: Delineation of hot and cold blocks by diplotype analysis. *Proc Natl Acad Sci USA* 99, 2228-2233.
- Schaffner, S.F. et al. (2005) Calibrating a coalescent simulation of human genome sequence variation. *Genome Res.*, 15, 1576-1583.
- Willer, C.J. et al. (2006) Tag SNP selection for Finnish individuals based on the CEPH Utah HapMap database. *Genet. Epidemiol.*, 30, 180-190.