Data Simulation Software for Whole-Genome Association and Other Studies in Human Genetics

Scott M. Dudek, Alison A. Motsinger, Digna R. Velez, Scott M. Williams, and Marylyn D. Ritchie

Pacific Symposium on Biocomputing 11:499-510(2006)

DATA SIMULATION SOFTWARE FOR WHOLE-GENOME ASSOCIATION AND OTHER STUDIES IN HUMAN GENETICS

SCOTT M. DUDEK, ALISON A. MOTSINGER, DIGNA R. VELEZ, SCOTT M. WILLIAMS, MARYLYN D. RITCHIE

Center for Human Genetics Research, Vanderbilt University, 519 Light Hall, Nashville, TN 37232, USA

Genome-wide association studies have become a reality in the study of the genetics of complex disease. This technology provides a wealth of genomic information on patient samples, from which we hope to learn novel biology and detect important genetic and environmental factors for disease processes. Because strategies for analyzing these data have not kept pace with the laboratory methods that generate the data it is unlikely that these advances will immediately lead to an improved understanding of the genetic contribution to common human disease and drug response. Currently, no single analytical method will allow us to extract all information from a whole-genome association study. Thus, many novel methods are being proposed and developed. It will be vital for the success of these new methods, to have the ability to simulate datasets consisting of polymorphisms throughout the genome with realistic linkage disequilibrium patterns. Within these datasets, we can embed genetic models of disease whereby we can evaluate the ability of novel methods to detect these simulated effects. This paper describes a new software package, genomeSIM, for the simulation of large-scale genomic data in population based case-control samples. It allows for single SNP, as well as gene-gene interaction models to be associated with disease risk. We describe the algorithm and demonstrate its utility for future genetic studies of whole-genome association.

1. Introduction

The identification and characterization of susceptibility genes for common complex human diseases, such as cardiovascular disease, is a difficult challenge for genetic epidemiologists. This is because many disease susceptibility genes exhibit effects that are partially or solely dependent on interactions with other genes. In addition, selection of the appropriate candidate genes limits our ability to identify novel genetic factors associated with disease. Whole-genome association has been proposed as a solution to these problems; however, the appropriate analytical methods for this type of data are unknown. To deal with this issue, many groups, including our own, are in the process of developing new computational approaches for the analysis of whole-genome association studies, but without a priori knowledge of the genetic model underlying the phenotype it is unclear whether a given method is accurate.

Strategies for analyzing datasets on the scale of whole genome association studies data have not kept pace with the laboratory methods that generate the

data. Because of this it is unlikely that technological advances will immediately lead to an improved understanding of the genetic contribution to common human disease and drug response. Currently, no single analytical method will allow us to extract all information from a whole-genome association study. In fact, no single method can be optimal for all datasets, especially if the genetic architecture for disease is substantially different.

One way to better design analytical protocols is to have datasets with known answers, but this is not possible using real data. When real data are used to test new methods, and significant results are found, it is impossible to know if they are false positives or true positives. Similarly, if nothing significant is detected, one cannot know if this is a lack of power, or the data had no true signal. Thus, it will be vital for the success of genome-wide association methods, to have the ability to simulate datasets consisting of polymorphisms throughout the genome on the scale of what is technically feasible. Having simulated data allows one to evaluate whether a methodology can detect known effects, and if the simulations are well-designed one can potentially embed a variety of genetic models of disease, making the evaluation of methods robust to genetic architecture.

Data simulations are often criticized because they are much cleaner than real data. However, simulating data remains an important component of most new methods development projects. To this end, any advances to improve the complexity of the data simulations will permit investigators to better assess new analytical methods. The present study was motivated by this lack of appropriately complex simulated data for association studies.

Several data simulation packages are currently available for family based study designs. SIMLINK[1,2], SIMULATE, and SLINK[3] will simulate pedigrees from an existing dataset. SIMLA[4] is a very nice software package for simulating both linkage and association in pedigree data. However, it does not allow for epistasis models or population-based simulations. Coalescent-based methods[5] have been used for population based simulation in genetic studies, however they do not allow for the tracking of ancestral information. In recent years, forward-time population simulations have been developed including easyPOP[6], FPG[7], and simuPOP[8]. simuPOP is the newest simulation package. It performs forward-time population simulations and allows the user to manipulate the evolutionary features. simuPOP is implemented in Python and provides flexibility for the user to run interactively using a Python shell or writing batch files[8]. The main weakness of simuPOP is the inability to simulate data based on complex gene-gene interaction penetrance functions. In addition, the programming environment is specific to Python, therefore, may not be user-friendly for all users. This paper describes a new software package,

genomeSIM, for the simulation of large-scale genomic data in population based case-control samples. It is a forward-time population simulation algorithm that allows the user to specify many evolutionary parameters and control evolutionary processes. It allows for single SNP, as well as gene-gene interaction models to be associated with disease risk. We describe the algorithm and demonstrate its utility for future genetic studies of whole-genome association.

2. Methods

2.1. *Algorithm*

genomeSIM utilizes two different methods to generate datasets. An initial population can be generated on the basis of allele frequencies of the SNPs and then further generations are created by crossing the members of successive generations. The simulator assigns affection status only after a specified number of generations. Alternatively, the simulator can construct a case-control dataset by generating individuals as above, assigning affection status, and selecting cases and controls until the dataset is complete.

Fig. 1 illustrates the general steps involved in producing a simulated dataset utilizing successive generations. As a first step, genomeSIM establishes the genome based on the parameters passed to it. The total number of SNPs is not limited except by hardware considerations. The user specifies the number of SNPs per gene and the total number of genes in the genome. The simulator randomly determines the number of SNPs per gene based on the minimum and maximum parameters. The simulator then randomly determines the recombination fraction between adjacent SNPs within each gene based on maximum and minimum recombination fraction parameters. The recombination fraction between any pair of SNPs is independent of the recombination fraction between other pairs of SNPs within a given gene. Similarly, recombination fractions between genes are independent. Thus, all recombination fractions are random and independent. SNPs are unlinked across genes. Finally, the allele frequencies are randomly set for each SNP based on preset maximum and minimum allele frequency parameters. For all these parameters, when the minimum is set equal to the maximum, the values across the simulated genome will be identical. Specific SNPs can also be set so that the disease SNPs allele frequencies will match the expected frequencies for the model used.

genomeSIM then generates an initial population based on the genome established in the previous step. Each individual in the population has two binary chromosomes. For each SNP in the genome, the simulator randomly

assigns an allele to each chromosome based on the allele frequencies of the SNP. The dual chromosome representation allows for an efficient representation of the genome and for crossover between chromosomes during the mating process. The genotype at any SNP can be determined simply by adding the values of the two chromosomes at that position. As a result, the genotypes range from 0 to 2 at any SNP.

The initial population forms the basis for the second generation in the simulation. For each cross two individuals are randomly selected with replacement to be the parents for a member of the new generation. Each parent contributes one haploid genome to the child. genomeSIM creates the gametic genotype by recombining the parent's chromosomes. The total number of individuals in each population is constant so the number of crosses conducted equals the number of individuals in the population for each generation.

A crossover is conducted as follows. genomeSIM selects one chromosome to be the start chromosome and begins copying allele values from that chromosome into the new chromosome. At every interval between SNPs, the simulator checks the recombination fraction against a randomly generated number. When the number is less than or equal to the fraction, the simulator switches chromosomes (assuming independent assortment) and begins taking allele values from the second chromosome. The simulator continues to check each interval and copies the allele values for the current chromosome until it reaches the end of the genome or another crossover takes place.

genomeSIM continues producing generations for the number specified and then assigns affection status to the final generation. Affection status is determined by the penetrance table for the simulation. To determine status, the simulator determines the genotype of the individual at the disease SNPs. The simulation then determines the penetrance for that genotype and generates a random number to determine if this individual is affected.

Alternatively, genomeSIM can produce the final dataset by producing individuals using the allele frequencies. The simulator's goal in this case is to generate the desired number of cases and controls. Each individual is checked against the penetrance table and then kept if there are not enough individuals with that affection status in the dataset. Additional individuals with that status are discarded. For example, if 500 cases and controls are needed, the simulator will take the first 500 controls that are generated but will then ignore any more while continuing to select the cases as needed. The simulator initially only generates the disease SNPs for each individual. If the simulator then needs to keep the individual based on its status, the rest of the alleles for the individual are generated.

genomeSIM can produce genetic heterogeneity by utilizing multiple penetrance tables. Each table is used for a portion of the final population. Datasets can also be produced with no disease model. If no penetrance table is used, then the individual has an equal chance of being a case or control. In addition, the simulator can generate phenocopies by assigning a fraction of the unaffected population to be affected at random. Finally, the simulator can introduce genotyping errors into the final population. The rate determines the expected number of errors per SNP. For each SNP, individuals are randomly selected and their genotypes are adjusted in a direction specified by the user if possible. For example, a selected individual may have a genotype of 2 and the error direction is specified as -1. In this case, the reported genotype for the individual will be 1. If the individual had a genotype of 0, no error would occur and another individual would be selected.

2.2. *Implementation*

genomeSIM is written in **ANSI-C++** and compiled using the GNU compiler into a library that can be linked to programs to generate datasets without the need for intermediate files. For the analyses done in this paper, the library was linked to a simple driver program that created input files for the Multifactor Dimensionality Reduction (MDR)[9] analysis software. The library provides simulation classes to be accessed by the main program for simulating both generational-based and frequency-based datasets.

The analysis can be run using functions in the library classes or the library can accept a configuration file as input for easy linkage with existing programs. The simulator accepts keywords and values as the configuration format. Table 1 displays the keywords that control the dataset production. Some keywords (POPSIZE, GENES, NUMGENS, MAXSNP, MINSNP, MAXRECOMB, MINRECOMB) are only used when simulating multiple generations to produce the final population. Other keywords (AFFECTED, UNAFFECTED, SIMLOCI) are only used when simulating a case-control set based on allele frequencies without crossing individuals. The differences arise from optimization of the process in the two cases. When simulating a case-control dataset without generations, the individuals can be set without regard to recombination rates between SNPs. In addition, the final dataset can be set to produce the desired number of cases and controls and the simulator will continue until it generates those numbers. The simulator only produces bi-allelic SNPs. This limitation allows the simulator to represent each chromosome as a series of bits and reduces the memory requirements.

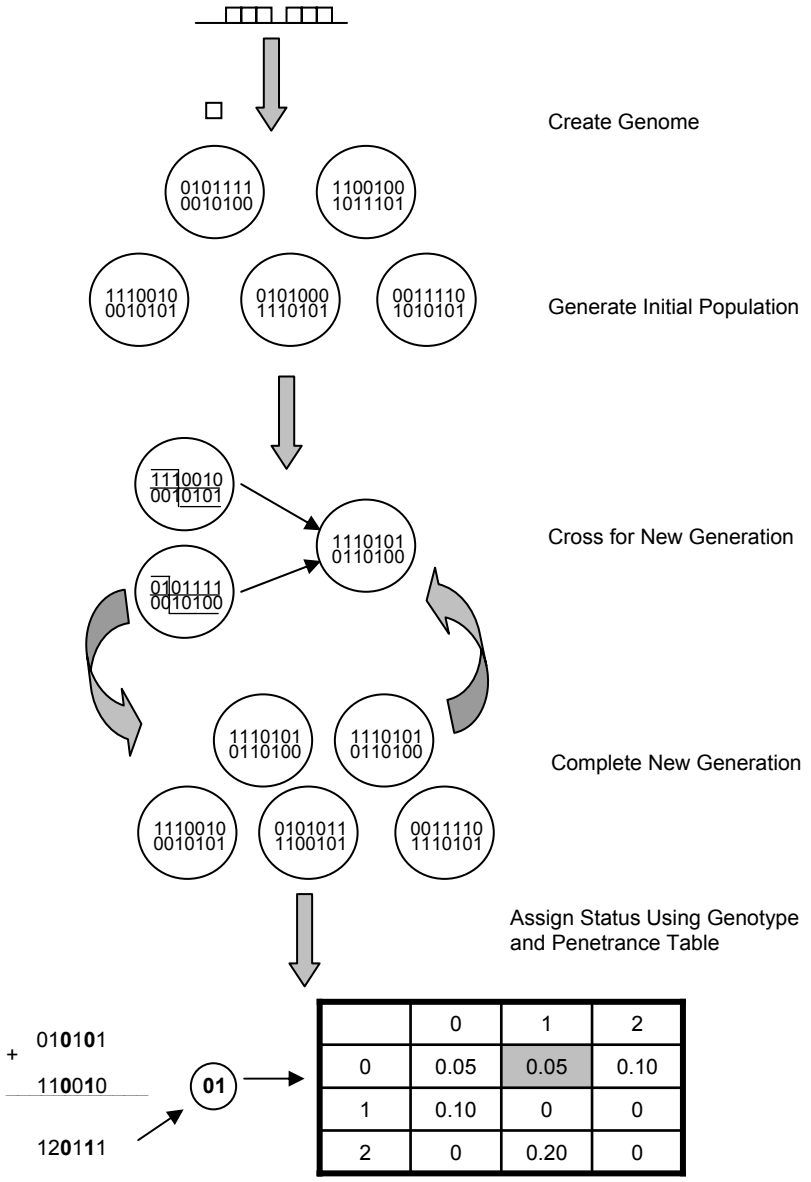

Figure 1. Summary of process involved in producing a simulated dataset. After the genome is constructed, an initial population of individuals is created and individuals cross by contributing one chromosome each to the offspring. These crosses create the next generation and the process repeats until the specified number of generations has occurred. In the last generation, the genotypes for the individual are produced by summing the chromosomes at each position. The genotype at the disease SNPs is used to find the penetrance value in the penetrance table.

Memory requirements vary with both the number of individuals and number of SNPs in the set being produced. For example, 10,000 individuals and 100,000 SNPs require slightly less than 400 MB of RAM. Total numbers of individuals and SNPs are only limited by the memory of the system running the software. We have successfully simulated 10,000 individuals and 400,000 SNPs on a system with 2 GB of RAM.

The library outputs data in a simple text format. Each line consists of one individual with the first column being the case or control status of the individual. Each additional column lists the genotype of the individual (0, 1, 2). This information is available through accessor functions of the library so that the output can be easily formatted to meet the needs of multiple software packages.

Table 1. Descriptive list of simulator parameters

Parameter	Example	Description
RAND	712	Sets random seed for creating dataset
MODELFILES	Model1.smod 0.7 Model2.smod 0.3	Lists model files that detail the penetrance table. Also indicates fraction of population that uses indicated model.
GENOTYPEERROR	.02	Per SNP error rate
PHENOCOPY	.05	Phenocopy rate in final population
AFFECTED	200	Number of cases in final population when only generating case-control set without crossing
UNAFFECTED	200	Number of controls in final population when only generating case-control set without crossing
SIMLOCI	500	Number of SNPs to simulate in a case-control set without crossing
ALLELELIMITS	0.05 0.50	Sets the range for the minor allele frequency of the SNPs in the simulation
ALLELEFREQS	1 0.7 0.3	Specifies allele frequencies for specific SNPs (overrides ALLELELIMITS for the SNP)
POPSIZE	1000	Size of simulated population
NUMGENS	100	Number of generations to simulate
GENES	100	Number of genes to simulate
MINSNP	5	Minimum number of SNPs per gene
MAXSNP	10	Maximum number of SNPs per gene
MINRECOMB	0.005	Minimum recombination rate between adjacent SNPs
MAXRECOMB	0.05	Maximum recombination rate between adjacent SNPs

2.3. *Benchmarks*

To test the genomeSIM's performance we simulated a dataset with 10,000 individuals and varying numbers of SNPs. The population underwent 100 generations of mating. We ran the tests on a PC with Intel Xeon 3.06 GHz CPUs and 2GB of RAM running Red Hat Enterprise Linux WS release 3 (Taroon Update 5). The simulator produces the dataset in 2 hours 48 minutes

when simulating 100,000 SNPs and 12 hours 17 minutes when simulating 400,000 SNPs.

We also tested the data simulator's performance in producing a set of 500 cases and 500 controls without mating generations. For 100,000 SNPs, the simulator produced the dataset in 11.7 seconds on the system listed above. For 400,000 SNPs the simulator produced the set in 48.8 seconds.

2.4. *Data Simulations*

For this paper, we performed several data simulations to demonstrate the utility of our new data simulation software. First, we simulated a single SNP recessive model with the penetrance table shown in Table 2. The allele frequency of the functional SNP was p=0.7, q=0.3, where p is the frequency of the A allele. Next, we simulated the two SNP gene-gene interaction model shown in Table 3. For this model, the allele frequencies of both functional SNPs were p=0.6, q=0.4.

Table 2. Single SNP recessive model with reduced penetrance

Genotype	Probability(disease|genotype)
AA	0.0
Aa	0.0
aa	0.9

Table 3. Two SNP gene-gene interaction model

	BB	Bb	bb
AA	0.177	0.080	0.005
Aa	0.074	0.150	0.017
aa	0.014	0.013	0.569

Table 4. Parameters for data simulations for MDR analysis

Population size	10,000
Total SNPs	50,000
Genes	5,000
SNPs per gene	10
Generations	150
Minimum recombination between SNPs	0.0
Maximum recombination between SNPs	0.10
Minimum minor allele frequency	0.05
Maximum minor allele frequency	0.5

We used one set of simulation parameters for these simulations (shown in Table 4). A total of 500 cases and 500 controls were extracted from the simulated population.

3. Results

To validate that the data simulations are indeed functioning as expected, we analyzed the datasets simulated to determine if statistical methodologies are able to detect the effects simulated. We applied the Multifactor Dimensionality Reduction (MDR)[9-11]approach to detect all single SNP and two-SNP models. We performed the MDR analysis without cross-validation due to the computation time required to analyze 50,000 SNPs. We selected the model with the minimum classification error and calculated a chi-square test for association. The results of the analyses are shown in Table 5. These results show uncorrected chi-square p-values. In the dataset with a recessive model simulated, MDR identified the correct model, SNP 5, as the optimal model. In the dataset with a two-SNP model simulated, MDR identified the optimal model as the two SNP model (SNP 5 and SNP 10). The best single SNP model in that dataset was not as significant as the two-SNP model. Thus, we would select the two-SNP model as the best model.

Table 5. Results of MDR analysis on simulated data

Model	SNPs	Classification error	Chi Square p-value
Recessive	5	1.10	0.00000000
Two SNP	7792	42.9	0.00001822
Two SNP	5 10	30.50	0.00000000

In addition to demonstrating the ability to simulate known effects, we also wanted to determine if our simulation algorithm was able to simulate linkage disequilibrium across the genome. Figure 2 shows a Haploview plot generated on one dataset simulated with our new software[12]. The data simulation parameters used for this particular dataset are shown in Table 6. There are several blocks of strong LD across this particular area of the genome. This indicates that this software is able to simulate LD in addition to specified genetic models.

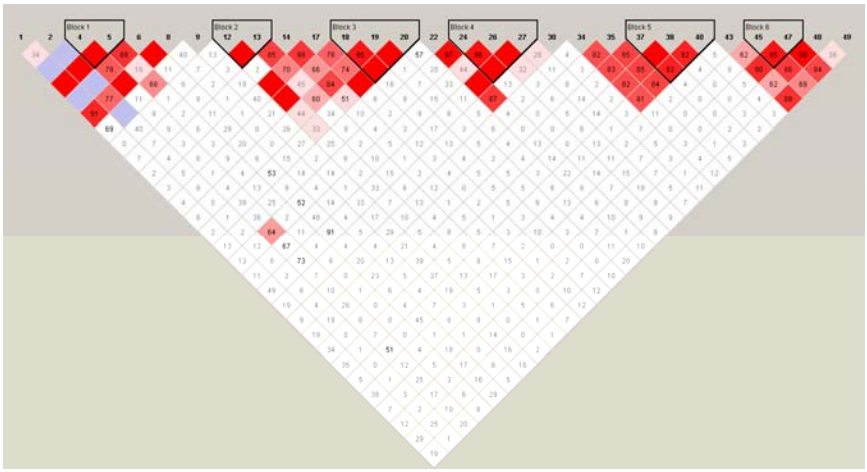

Figure 2. Plot generated by Haploview on one simulated dataset.

Table 6. Parameters for data simulations for Haploview plot

Population size	500
Total SNPs	50
Genes	5
SNPs per gene	10
Generations	500
Minimum recombination between SNPs	0.001
Maximum recombination between SNPs	0.001
Minimum minor allele frequency	0.05
Maximum minor allele frequency	0.5

4. Discussion

Detecting disease susceptibility genes for common disease is a major focus of study in human genetics. The ability to achieve success in this endeavor is dependent upon intelligent study design, accurate genotyping, and efficient algorithms for analysis. Methodology development in statistical and computational genetics continues to advance the field, and novel approaches are being developed in an attempt to keep pace with the development of genotyping technology. Evaluating and comparing these methods requires the ability to perform complex data simulations to efficiently test the new algorithms. Several methods currently exist for the simulation of family-based data including SIMLINK, SIMULATE, and SIMLA. Coalescent-based and forward-time population based algorithms have also been developed, however, to our knowledge, none have the flexibility of genomeSIM. genomeSIM is a new data

simulation package that uses forward-time population based simulations, user-specified evolutionary features, and the ability to specify simple or complex penetrance functions to assign disease status, including gene-gene interaction models. We believe that since interactions are likely to be an important component of complex disease[13,14] having the capability of evaluating new methods in this type of data will be a true test of the method's success.

While we believe that genomeSIM is an advance over current data simulation methods, we will continue to add additional features. There are currently no family based simulation algorithms that allow for the simulation of complex gene-gene interaction models. We are in the process of allowing genomeSIM to generate pedigree data under such penetrance functions. We plan on simulating larger sets more quickly by parallelizing the algorithm. In addition, there are many evolutionary features that could be parameter options in the algorithm including random genetic drift, population bottlenecks, and selection that we plan to implement. genomeSIM is freely available from the authors upon request. It will also be available via the internet at http://chgr.mc.vanderbilt.edu/ritchielab.

5. Acknowledgements

This work was supported by National Institutes of Health grants GM31304, AG20135, and in part by HL65962, the Pharmacogenomics of Arrhythmia Therapy U01 site of the Pharmacogenetics Research Network.

.

6. References

1. Boehnke,M. Estimating the power of a proposed linkage study: a practical computer simulation approach. *Am. J. Hum. Genet*. **39**, 513-527, (1986)
2. Ploughman,L.M. and Boehnke,M. Estimating the power of a proposed linkage study for a complex genetic trait. *Am. J. Hum. Genet*. **44**, 543-551, (1989)
3. Weeks, D. E, Ott, J, and Lathrop G.M. SLINK: A general simulation program for linkage analysis. *American Journal of Human Genetics* **47**, A204. (1990)
4. Bass,M.P. et al. Pedigree generation for analysis of genetic linkage and association. Pac. Symp. *Biocomput*., 93-103, (2004)
5. Kingman, J. The coalescent. *Stochastic Processes Appl* **13**, 235-248. (1982)
6. Balloux,F. EASYPOP (version 1.7): a computer program for population genetics simulations. *J. Hered*. **92**, 301-302, (2001)

7. Hey, J. Nielsen, R. Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of Drosophila pseudoobscura and D. persimilis. *Genetics* **167**, 747-60. (2004)
8. Peng,B. and Kimmel,M. simuPOP: a forward-time population genetics simulation environment. *Bioinformatics*. **21**,3686-7 (2005)
9. Ritchie,M.D. et al. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *Am J Hum. Genet* **69**, 138-147, (2001)
10. Hahn,L.W. et al. Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. *Bioinformatics*. **19**, 376-382, (2003)
11. Ritchie,M.D. et al. Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. *Genet. Epidemiol*. **24**, 150-157, (2003)
12. Barrett,J.C. et al. Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*. **21**, 263-265, (2005)
13. Moore,J.H. The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Hum. Hered.* **56**, 73-82, (2003)
14. Sing,C.F. et al. Genes, environment, and cardiovascular disease. *Arterioscler. Thromb. Vasc. Biol*. **23**, 1190-1196, (2003)