vGWAS-Simu Documentation

A. Kawam
June 8^{th}, 2017

Introduction

This document describes vGWAS-Sim, a tool for simulating phenotypes from genotype data. vGWAS-Sim reads the output of the coalescent genotype simulators and produces quantitative and qualitative phenotype values. vGWAS-Sim provides the user with the ability to induce both: mean and variance shifts resulting from on one or more causal loci. vGWAS-Sim supports both homozygous and heterozygous diploid genotypes. The phenotype could be generated using either a co-dominance or a complete dominance model. Finally, the resulting phenotypes are outputted in PLINK (Purcell et al, 2007) format to allow further analysis of the data.

Setup

The software was coded in Python 2.6 and uses some of Python's standard libraries. It was tested under both Linux and MacOS, but since Python is portable, it should run on Windows as well. To use the tool:

1. Unpack the vGWAS-Sim package.
2. Add execution permission to vGWAS-Sim by typing: chmod +x vgwas-sim.py
3. Make vGWAS-Sim available across your entire system through typing: export PATH=$PATH:$PWD. If you want to make this setting permanent, go to your home-directory and add the line export PATH=$PATH:¡path to simwas¿ to your .bashrc file.

For Windows:

1. Make sure Python is installed on your computer.Python can be downloaded from http://www.python.org/.

2. Unpack the vGWAS-Sim archive.

3. Add execution permission to vGWAS-Sim by right-clicking on the file, select 'Properties' and then uncheck the box 'Read-only'

4. In order to make vGWAS-Sim available across your entire system, you have to add vGWAS-Sim's directory to your system's PATH.

Tool Description

Input

vGWAS-Sim reads an input file defined using the –file option. This file can be the output of ms (Hudson, 2002), msHOT (Hellenthal and Stephens, 2007), msms (Ewing and Hermisson, 2010), and GENOME (Liang et al. 2007). The flag -i specifies the format of the input file by specifying 'M' for ms, msHOT, and msms, and 'G' for GENOME. Under the default setting, the individuals are treated as homozygous, by setting -h 1 they are treated as heterozygous. Heterozygous individuals are created by combining two simulated chromosomes to a joint genotype. A detailed list of inputs along with their default values is given in Table S1.

Output

vGWAS-Sim writes the genotypes and phenotypes in file formats compatible for PLINK (Purcell et al, 2007). The prefix of the output files can be defined by the user (–outfile). The output consists of three files: 1) A '.map' file containing each genotype's chromosome location, marker ID, genetic distance, and physical position. 2) A '.ped' file that describes the pedigree information of the genotypes and their individuals. Currently, our simulator uses this file to assign phenotype values to each genotype profile. However, we plan to incorporate pedigree information in the subsequent versions of our simulator. 3) The third file our simulator outputs is a '.causal' file which includes the position, index, minor allele frequency (MAF) and the effect of the causal marker(s).

Phenotype Generation

Our tool provides the user with the ability to introduce both: mean and variance discrepancies based on one or more loci. The tool is also capable of supporting both homozygous and het-

erozygous genotypes. For heterozygotes, the phenotype could be generated using either the co-dominance or the complete dominance model. The simulator uses Eq. 1 and Eq. 2 to calculate the phenotype value of the homozygous genotypes and complete dominant heterozygous genotypes respectively. The simulator also uses Eq. 1 to calculate the phenotype value in the case of co-dominance for heterozygous genotypes.

$$y = \mu + \boldsymbol{g^T}\boldsymbol{\beta} + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 + \boldsymbol{g^T}\boldsymbol{\theta}) \tag{1}$$

$$y = \mu + (\boldsymbol{g_1} + \boldsymbol{g_2})^{\boldsymbol{T}}\boldsymbol{\beta} + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 + (\boldsymbol{g_1} + \boldsymbol{g_2})^{\boldsymbol{T}}\boldsymbol{\theta}) \tag{2}$$

Using the default settings, the genotypes are treated as heterozygotes under co-dominance. By unsetting the heterozygous flag, (-h 0) the genotypes are treated as homozygotes. When running the simulator in the heterozygous case, the simulator reads the genotype input file and considers every two consecutive genotype strings as the two chromosomes of one individual. However, in the homozygous case and since the same allele exists on both chromosomes, the simulator considers every genotype string as a separate entry.

The user can control the value of several important parameters through the command line inputs, such as the dominance model, the number of loci, their minor allele frequencies, and their respective effect sizes. For example, complete dominance could be simulated by setting the dominance flag to one (- -dominance 1). A detailed list of inputs along with their default values is given in Table S1. In order to account for causal loci that are in LD, our simulator implements the mean and variance shift adjustments described in the previous section. In such a case, the simulator requires an r^2 matrix in a square format, as outputted by PLINK. This matrix could be specified using the (- -ldfile) flag.

The simulator is capable of generating qualitative phenotypes for case/control studies using the liability model described above. Finally, in the qualitative phenotype simulation, our tool provides the user with the option of sampling the case and control population to produce samples with a specified size. In typical scenarios, prevalence rates are usually low, which results in a substantially larger control population as compared to the case population. The sampling option gives the user the ability to set the size of the output samples.

To illustrate the usage of our tool, we provide two examples based on typical scenarios: a quantitative phenotype simulation example, and a qualitative phenotype simulation example.

Usage Example 1: Quantitative Simulation

vgwas-simu.py - -file input.genotypes -i M -n 2
- -snpfile snps.txt - -maf_r 0.2,0.5 - -outfile out

snps.txt:
542 0.1 0.0
-1 0.0 0.1

This example reads a file input.genotypes in ms format and simulates a quantitative phenotype with two loci as shown in the loci file $snp.txt$. The two loci have a combined effect of 20% on the total variance. The first locus has a mean effect size of 10% and no variance effect size, while the second locus has no mean effect and a 10% variance effect. The first column of $snp.txt$ specifies the genotype index at which the locus is created. Alternatively, when a '-1' value is specified, the simulator chooses a locus index randomly from the list of loci that have a minor allele frequency within the minor allele frequency range. The minor allele frequency range is specified using the '–maf_r' option which is between 0.2 and 0.5 in this example. Finally, the simulator will produce three files: 'out.ped', 'out.map', and 'out.causal' as explained earlier.

Usage Example 2: Qualitative Simulation

vgwas-simu.py - -file input.genotypes -i G -n 1
- -snpfile snps.txt - -maf_r 0.2,0.5 - - quant 1
- - sample 500,500 - -outfile out

snps.txt:
-1 0.1 0.1

In this example, the simulator reads a file input.genotypes in GENOME format and simulates a quanlitative phenotype with one locus as shown in the $snp.txt$ loci file. The locus has a mean effect size of 10% and a variance effect size of 10%. The first column of $snp.txt$ specifies that the simulator should randomly choose a locus from the list of loci that have a minor allele frequency within the specified minor allele frequency range. The minor allele frequency range is specified using the '–maf_r' option which is between 0.2 and 0.5 in this example.

The main distinction between this example and the first example is that the $quant$ flag is set, which means the simulated output will either have a '0' or '1' label representing 'control' and 'case' respectively. Furthermore, the $sample$ flag indicates that the simulator should extract a sample of 500 controls and a sample of 500 cases from the outputted phenotypes according to a uniform distribution. However, if the specified sample size is larger than the number of available phenotypes from that class, an error will be issued. It is advisable that the input genotype population be larger than the size of the case sample divided by the prevalence rate. Finally, the simulator will produce three files: 'out.ped', 'out.map', and 'out.causal' as explained earlier.

Table 1: *

Supplementary Table 1 Usage options for vGWAS-Simu.

Option	Description
–file	name of input file (In the format of ms or GENOME)
-i	type of input file ("G" for GENOME, "M" for ms) (default: G)
-h	a binary value: either homozygous (0) or heterozygous (1) (default: 0)
-q	quantitative phenotypes (0) or qualitative case/control phenotypes (1)
–outfile	prefix for the output files in PLINK format (default: name of input file)
–snpfile	file with tab delimited effect sizes of SNPs on total variance
–maf_r	MAF range for causal markers if SNP not specified in snpfile (upper and lower bound, separated by a comma, no space) (default: 0.05,0.45)
–ldfile	file with tab delimited linkage disequilibrium in square format as outputted by PLINK
–dominance	co-dominance (0) or complete dominant (1) model used (default: 0)
–prev	disease prevalence in case/control studies (default: 0.01)
–sample	sample case/control outputs (control then case size, separated by a comma e.g 500,500)
–base_avg	baseline average for phenotype (default: 0.0)
–tot_var	total variance for phenotype (default: 1.0)
–min	minimum possible value for phenotype (default: -inf)
–max	maximum possible value for phenotype (default: +inf)

References

1. Ewing G, Hermisson J (2010) MSMS: A Coalescent simulation program including recombination, demographic structure, and selection at a single locus. Bioinformatics 26(16):2064-2065, DOI 10.1093/bioinformatics/btq322

2. Hellenthal G, Stephens M (2007) msHOT: modifying Hudson's ms simulator to incorporate crossover and gene conversion hotspots. Bioinformatics 23(4):520-1, DOI 10.1093/bioinformatics/btl622

3. Hudson RR (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics 18:337-338

4. Purcell S, Neale B, Toddbrown K, Thomas L, Ferreira M, Bender D, Maller J, Sklar P, Debakker P, Daly M (2007) PLINK: A tool set for whole-genome association and population-based linkage analyses. The American Journal of Human Genetics 81(3):559-575, DOI 10.1086/519795

5. Van Rossum G (1995) Python Reference manual. CWI (Centre for Mathematics and Computer Science), Amsterdam