POPSIM: a general population simulation program

Jochen Hampe1,4, Thomas Wienker3, Stefan Schreiber1 and Peter Nürnberg2

1IVth Department of Medicine and 2Institute of Medical Genetics, Charite Medical School Berlin, Germany and 3MDC Center for Molecular Medicine, Berlin, Germany

Received on July 22, 1997; revised on December 22, 1997; accepted on December 23, 1997

Abstract

Motivation: The investigation of common disorders of polygenic inheritance using both population-based association designs and non-parametric linkage analysis within families is gaining increasing importance. We have created a program that allows for the flexible simulation of populations as a tool to investigate the properties of population-based mapping approaches.

Results: We have created a population simulation program, POPSIM, that (i) creates a virtual representation of every individual, (ii) makes no prior assumptions but the Mendelian rules and (iii) allows populations of several million individuals in size to be generated and to be followed over hundreds of generations. The parameters of the disease model, population structure and population expansion rate can be specified. Flexible sampling options exist that allow samples of families and individuals to be drawn at any given point during the population history. The program may be a useful tool in the study of the influence of genetic drift, recombination and admixture on the generation and maintenance of linkage disequilibrium in populations, as well as the evaluation of stochastic sampling characteristics of families and individuals conditional on a complex genetic phenotype from homogeneous and heterogeneous populations.

Availability: The source code as well as Sun and Windows NT4.0 console executables of the program are available under http://www.ukrv.de/ch/medgen/html/benutzer/j.hampe/popsim.html.

Contact: J.Hampe@mucosa.de

Introduction

Simulation techniques play an important role in applied genetics and in the development of new statistical methods for linkage and association analysis. Currently available programs include SIMULATE (Terwilliger and Ott, 1994), SIMLINK (Boehnke, 1986) and SLINK (Weeks et al., 1990), and the more general GASP (Wilson et al., 1996) programs which simulate the genotypes of families under a defined structure and preset linkage parameters. A simulation of a given pedigree using SLINK is now a common prerequisite before embarking on a genome-wide linkage analysis in disorders where reasonable parametric estimates on the disease model exist. However, for complex genetic disorders like diabetes, arteriosclerosis, bipolar disorder or inflammatory bowel disease, the genotype to phenotype relationship is much more complex (Risch, 1990; Cox et al., 1992; Orholm et al., 1993; Risch and Botstein, 1996). Possible approaches for the ultimate mapping of susceptibility genes responsible for these conditions include non-parametric linkage analysis (using recombination events within the families under investigation) and linkage disequilibrium mapping (association studies—using recombination events over the past generations in the population). In these approaches, population structure and sampling effects may have profound effects on the results (Spielman et al., 1993, 1994; Suarez et al., 1994; Thompson and Neel, 1997). Several analytical models for defined scenarios of population history and structure have been developed for the prediction of allele frequencies and disequilibrium parameters (Kaplan and Weir, 1995; Kaplan et al., 1995), which have been applied successfully in isolated populations (Hästbacka et al., 1992, 1994). Simulation-based statistics have been used to assess the sensitivity and specificity of non-parametric linkage analysis (Almasy et al., 1995; Devlin and Risch, 1995; Davis et al., 1996) which are each directed at a specific family structure. The use of predefined statistical distributions for the prediction of certain population parameters in the mentioned parametric simulation approaches may not be generally applicable.

Another approach uses direct simulation, whereby a representation of every individual is generated. The first implementation of this approach by MacCluer (1967; POPGEN) has evolved into a set of Fortran programs that
have been used by the ‘Genetic Analysis Workshops’ (MacCluer et al., 1995) and allow—by using different components from these programs—a variety of scenarios to be implemented. LINKERS, SNAPPERS and GRASPERS (Yang et al., 1990; Prasad, 1992; Ackerman et al., 1993; Goay et al., 1995) for the VMS environment allow very detailed simulation scenarios, but are limited to two major loci and two alleles per locus. The underlying simulation concept has been ported to a C++ library ‘SIMEX’ that enables the user to build custom models in C++ which may also be applied to infectious disorders like spread of influenza (Peterson et al., 1993). SMALLPOP (Fournet et al., 1995) is limited to small populations and is directed at animal breeding using external selection criteria.

The population-based programs are very resource intensive and often require additional programming for efficient use. We therefore implemented a program under the following aims: (i) to enable the direct simulation of populations of a realistic size (i.e. several million individuals); (ii) making no prior assumptions but the Mendelian rules; (iii) to enable the specification of relevant population parameters; (iv) to deliver the output in a format that is easily processed by analysis programs for both parametric and non-parametric linkage analysis; (v) to include all experimental information into a set-up file that is suitable for batch processing.

**System and methods**

POPSIM was written in ANSI C and uses C standard libraries. System-specific versions for Sun Solaris and a Windows NT4.0 console application are also available. Efforts to minimize the memory requirements of the program have included the use of half-bytes for the storage of genotype information and the immediate replacement of the parents by their children in the population space. The central ‘breeding’ routines use exclusively integer and bit operations, all float point operations have been moved into initialization routines.

**Specification of the experiment**

In order for the program to be as flexible as possible and to be easily used in batch processing, all information on the simulation experiment is specified in a text file (extension *.pop_init). Here, a variety of settings pertaining to the structure of the population, susceptibility locus and marker information, disease model, population propagation and sampling characteristics can be specified (Table 1). These data are read into an ‘experiment’ structure during the experimental set-up routines.

<table>
<thead>
<tr>
<th>Table 1. Variables specified in the experiment set-up file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Population generation</strong></td>
</tr>
<tr>
<td>Number of population replicates</td>
</tr>
<tr>
<td>Initial size of the population</td>
</tr>
<tr>
<td>Number of subpopulations and their ratio in the final population</td>
</tr>
<tr>
<td>Random seed for population generation</td>
</tr>
<tr>
<td><strong>Population propagation</strong></td>
</tr>
<tr>
<td>Number of generations</td>
</tr>
<tr>
<td>Average number of kids per family and the proportion of the population attempting to reproduce—a method of specifying population growth or shrinkage</td>
</tr>
<tr>
<td>Random seed for propagation</td>
</tr>
<tr>
<td><strong>Locus structure and disease model</strong></td>
</tr>
<tr>
<td>Number of susceptibility loci (all unlinked to each another)</td>
</tr>
<tr>
<td>Initial allele frequencies of the risk allele for each locus and subpopulation</td>
</tr>
<tr>
<td>Recombination fraction for the marker loci linked to each susceptibility locus (no interference)</td>
</tr>
<tr>
<td>Initial association of a founder haplotype with the risk allele of the susceptibility locus</td>
</tr>
<tr>
<td>Disease model (additive polygenic, classical inheritance, 2 locus penetrance matrix)</td>
</tr>
<tr>
<td>Weighting factors for each locus, weight of the environmental factor for the additive disease model</td>
</tr>
<tr>
<td>Cut-off value for affection</td>
</tr>
<tr>
<td><strong>Sampling and I/O streams</strong></td>
</tr>
<tr>
<td>For each Checkpoint: (i) number of generations after which sampling is performed; (ii) type of sample to be drawn</td>
</tr>
<tr>
<td>Sample types: Sib-pairs and nuclear families with the number of affected individuals per family to be specified</td>
</tr>
<tr>
<td>Output either on standard output or in a dedicated file for each type of sample</td>
</tr>
</tbody>
</table>
Fig. 1. Implementation of a virtual individual: The internal storage format is symbolized in boxes, giving the type of information and the format of storage (0/1: bit; 0..15: half-byte; 0..255: byte) on the top and bottom, respectively. Risk locus information contains the genotype of the susceptibility gene itself and the genotypes of the linked marker loci (up to four in the standard compilation). The thetas between marker and risk locus are specified in the experimental set-up file.

**Population implementation**

The program uses non-overlapping populations. The population is stored in three principal data structures: person, family and population. The central data structure of the program is a person, for which the sex, a random environmental factor and genetic information on risk genes and linked markers are stored. In order to minimize memory requirements, the allele numbers of the genotypes are packed in half-bytes (Figure 1). The family is a temporary dataset of pointers, for which a male and a female are drawn from the population, and children are generated as determined by a random draw from a binomial distribution, the mean of which is given in the experimental set-up (see below). Since the program uses non-overlapping generations, memory requirements can be minimized by immediately replacing the parents with their children. Thus, the next generation grows in the same population space at the expense of the old one (Figure 2).

**Population replicates**

Two streams of pseudo random numbers are used in the program. One stream provides all random numbers for mating, meiosis, etc., and the second stream is used for population set-up. The latter is reset every time a population replicate is required, giving the opportunity to subject the same population to different mating scenarios repeatedly. After the experimental set-up, the complete experiment structure is put on the stack and reconstituted for each population replicate.

**Sampling during simulation**

Sample information is either directed to the standard output or to files with characteristic extensions (Figure 3). The sampling timetable, i.e. after which generation what type of sample is to be drawn, is given in the experimental set-up file and read into the ‘checkpoints’ structure. No analysis functions are incorporated into the POPSIM program itself. Rather, independent analysis programs like the LINKAGE (Lathrop *et al.*, 1984) package or the companion program CHISQUARE (which calculates $\chi^2$ statistics from case–control data) and the TDT (transmission disequilibrium test)
FIG. 3. Flow of information in a simulation experiment using POPSIM. As specified in the experimental set-up file, the flow of sample information is directed into files with specific extensions.

(Spielman et al., 1993, 1994) statistic from triplet data are used for data analysis.

**Family creation and meiosis**

A stream of integer random numbers is used to make all decisions in the mating and meiosis routines, and is designated as ‘random’ in the pseudo code below. In order to avoid floating point operations, the cut-off values were normalized during the experimental set-up routines on the intervals of the equally distributed pseudo random numbers. In the case of the number of offspring to be created for a given family, the mean of a binomial distribution is specified in the experimental set-up file. The density curve of this distribution is translated into values of an integer vector and a random number used as the index for an individual family.

We now give the pseudo code of the central mating and meiosis routines.

```c
breed (struct *experiment, struct *population, struct *checkpoints) {
  for (all generations) {
    reset_population_pointers;
    total_sampling(population, checkpoints);
    // total population sampling happens here
    while (still families needed) {
      mother = get_male(population, random);
      father = get_female(population, random);
      if (get_number_of_kids(binomial distribution, random)) != 0) {
        make_space_for_kids(population experiment, kidnumber);
        create_kids(mother, father, population, experiment);
        ascertain(mother, father, population, checkpoints);
        // family sampling is performed here
        destroy_parents(mother, father, population);
      }
    }
    create_kids(struct *population, struct *experiment, kidnumber) {
      while (still kids to create) {
        assign_sex(random);
        assign_environmental_factor(random);
        assign_paternal_haplotype(random);
        assign_maternal_haplotype(random);
      }
    }
    assign_maternal_haplotype(random) {
      for (all markers) {
        assign_principal_chromo
        some(first_bit_of_random);
        // grandfather or grandmother
        if (last_15_bits_of_random >= normalized_theta)
          recombination;
        else no_recombination;
      }
    }
}
```

**Results and discussion**

**Efficiency of the implementation**

The direct simulation approach leads to a heavy load on both main memory and processor capacities. The bulk of processor resources are used during the simulation of meiosis. Here, the packaging of genotype information into half-bytes results in a slowdown of ~50% (data not shown). However, in contrast to the use of integer data types, this reduces (de-
Fig. 4. Decline of initial linkage disequilibrium in an at random mating population of 200,000 individuals. It is evident that the recombination fraction $\theta$ determines the rate of this process. The analytical prediction is plotted in parallel to the simulation results. A population with an initial haplotype frequency ($1D$) of 0.87 and a population allele frequency of marker 1 of 1/16 was chosen. Fifty replicates of this population were generated and subjected to successive generations of random mating. The mean of the observed haplotype frequency ($1D$) was plotted as a function of time. Note the relative discrepancy of simulation and theoretical prediction in very small thetas (see the text).

Fig. 5. Distribution of the haplotype frequency $1D$ in an at random mating population ($\theta = 1$ cM). The same data as in the experiment in Figure 4. Instead of the mean of $f_{1D}$, the distribution of observations from the 50 experiments (N on the Z-axis) is shown.
and allows for a theoretically unlimited number of susceptibility loci. Specialized simulations like changing population expansion patterns, ongoing admixture, conditional mating, differential environmental factors, interference and the use of linked susceptibility loci are presently only possible by modifying the respective routines in the source code. As a future development, the respective procedures will be made available as options in the experimental set-up files.

**Error elimination**

Errors are an inevitable part of all software from a certain degree of complexity. However, in simulations of large datasets and complex problems, errors are very hard to trace back and may corrupt the integrity of the whole computer experiment. The present version of the program has been used as a tool with gradual enhancement of functionality since May 1996 in our laboratory. Pedigrees and recombinational events were traced by hand in several simulations. Furthermore, the program was used to simulate data under known parametric models, under which the expected results were obtained (comparison with SLINK under recessive and dominant models; data not shown). It was our policy to leave the central breeding routines untouched in the evolution of the program, since they had been extensively debugged.

**Application example: decay of linkage disequilibrium in a population**

We investigated the decay of linkage disequilibrium in a large population, which can be well described by mathematical models (Robbins, 1918; Thompson and Neel, 1978) in order to validate the simulation (Figure 4). Linkage disequilibrium $\Delta_0$ between a marker allele $I$ of a biallelic marker $M$ ($I,2$) and a disease locus ($D,d$), that existed at a certain point in population history, is expected to decline under random mating during consecutive generations ($n$) at a rate that is dependent on the recombination fraction ($\theta$) between marker $M$ and the disease locus. In a population of indefinite size, this relationship is given by $\Delta_n = (1 – \theta)^n \Delta_0$, where $\Delta_n$ was calculated as $\Delta = f_{I2}f_{D2} – f_{I2}f_{D2}$. The figure shows the mean of 50 replicate simulation experiments in good concordance with theoretical estimates. As shown in Figure 4, the relative discordance of simulation and prediction in very small recombination fractions ($0.1 \text{cM}$) shows that genetic drift rather than recombination determines the haplotype frequency. Here, the simulation offers the possibility to look at the distribution of haplotype frequency $1D$ in replicate experiments (Figure 5), which are difficult to derive analytically (Neel and Thompson, 1978; Weir and Hill, 1980; Hill and Weir, 1988, 1994).

**Possible applications**

The program allows for a very flexible specification of population simulation experiments. Special options exist that allow the disease model to be fitted empirically as close as possible to the known epidemiological data. Once this model is specified, the program can serve as a tool for the study of genetic drift, recombination and admixture on the generation and maintenance of linkage disequilibrium in populations, as well as the evaluation of stochastic sampling characteristics of families and individuals conditional on a complex genetic phenotype from homogeneous and heterogeneous populations. The program may, therefore, serve both as an instrument in theoretical research and applied genetics in allowing the generation of datasets for power estimations of mapping projects, as well as the evaluation of genetic study designs in complex genetic disorders.

**References**


