



FAMILY-SIM

A program for simulating parent-offspring, full-sibs, half-sibs, and unrelated individuals based on an allele frequency file.

ver 1.0

by:
Tim Frasier

Last Updated:
22-Sep-2009

Department of Biology
Saint Mary's University
923 Robie Street
Halifax, NS B3H 3C3
Canada

Table of Contents

Overview	3
Legal Stuff	3
Installation	4
Windows Installation.....	4
LINUX Installation #1 (No Source Code).....	4
LINUX Installation #2 (From the Source Code).....	5
Infiles	6
Allele Frequency File.....	6
What the Program Does	6
References	8

Overview

FAMILY-SIM is a program written in C that will take an allele frequency file and use those data to generate a user-defined number of individuals of known relatedness. Specifically, the program will generate the given number of pairs of parent-offspring, full-sibs, half-sibs, and unrelated individuals. These data can then be used to assess the expected resolution you can expect from your data in terms of differentiating between individuals of different relatedness.

The generation of individuals, as well as the process of Mendelian inheritance within the simulated data sets, require random numbers. FAMILY-SIM uses random number generators from the GNU Scientific Library (GSL, Galassi *et al.* 2006).

Legal Stuff

I have only briefly tested FAMILY-SIM so far. It appears to be functioning properly, and I am currently using it to analyze data from projects that I am working on. However, I cannot guarantee that it does not contain any errors, or that its results will always be reliable.

The program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details (<http://www.gnu.org/licenses/gpl.html>).

Installation

There are three ways to install FAMILY-SIM: one way for Windows; and two for LINUX - one method without the source code, and the other with the source code that you can compile yourself. Details are below.

Windows Installation

Step 1: Download the zipped file from the website.

Step 2: Unzip it wherever you want to keep the program

- The folder will include:

- (a) The executable file;
- (b) A folder containing the example file; and
- (c) Other files necessary for the program to run (don't move or delete them!!!)

Step 3: You should place your infiles in the same folder as the executable file prior to running the program.

LINUX Installation #1 (No Source Code)

Step 1: Download the appropriate file from the website.

Step 2: Move the folder to wherever you want to keep the program.

Step 3: Unzip the file by typing:

```
>gunzip FAM-SIM_v1.0_LINUX_1.tar.gz
```

Then:

```
>tar -xvf FAM-SIM_v1.0_LINUX_1.tar
```

- This will make a folder called FAM-SIM_v1.0_LINUX_1

Step 4: Go into that folder

```
>cd FAM-SIM_v1.0_LINUX_1
```

- There will be a folder called "Example_Files", and a file called "sim"

Step 5: Turn this file into an executable by typing:

```
>chmod a+x sim
```

Step 6: You can now start the program by typing:

```
>./sim
```

LINUX Installation #2 (From the Source Code)

If you choose to download the source code and compile it yourself, you will need to have the GNU Scientific Library (GSL) installed. There are a few ways to get this. One is from the GNU website (<http://www.gnu.org/software/gsl/>). If you are using Ubuntu, you can also install the GSL using Synaptic (install both the “gsl-bin” and the “libgsl0-dev” packages). Note that you will also have to have some general programming packages installed as well: “libc6” and “build-essential”, which can also be installed via Synaptic.

Once that is installed, you can install FAMILY-SIM.

Step 1: Download the appropriate file from the website.

Step 2: Move the folder to wherever you want to keep the program.

Step 3: Unzip the file by typing:

```
>gunzip FAM-SIM_v1.0_LINUX_2.tar.gz
```

Then:

```
>tar -xvf FAM-SIM_v1.0_LINUX_2.tar
- This will make a folder called FAM-SIM_v1.0_LINUX_2
```

Step 4: Go into that folder

```
>cd FAM-SIM_v1.0_LINUX_2
- There will be a folder called “Example_Files”, and a file called “FSIM.c”
```

Step 5: You can compile the source code by typing:

```
>gcc FSIM.c -lgsl -lgslcblas -lm -o sim
- This will create and executable file called “sim”
```

Step 6: You can now run the program by typing:

```
>./sim
```

Infiles

There is only one infile that you will need for the analyses conducted by FAMILY-SIM: your allele frequency file. Your file should be in a tab-delimited text format. The name of the file should not be longer than 40 characters (including extension), and it **should not contain any spaces**. When entering the name of your file, be sure to include the extension (if it has one).

Allele Frequency File

Your allele frequency file should have one column for each locus, with each column containing the allele frequencies for that locus. A zero (0) needs to be present at the top of each column, and you cannot include locus names. ALL columns must have values for the same number of rows. This means that the number of rows will equal the number of alleles in your most polymorphic locus. You **MUST** fill these rows with integers > 1 as placeholders for those loci with fewer alleles. An example allele frequency file is shown below for four loci. The first locus has two alleles, the second has four alleles, the third has five alleles, and the fourth has three alleles. In this case, 5 was used as the placeholder.

0	0	0	0
0.2	0.1	0.1	0.4
0.8	0.2	0.3	0.1
5	0.5	0.15	0.5
5	0.2	0.2	5
5	5	0.25	5

What the Program Does

The program should be straightforward to use. Once opened, it will ask you a few questions:

1. How many pairs of individuals (for each set of relatedness values) do you want to generate?
- Note that this is just one number (e.g. 100), and the program creates the same number of pairs of individuals for each type of relatedness.
2. How many alleles do you have in your most polymorphic locus?
3. How many loci did you use?
4. What is the name of your allele frequency file (including extension)?

It will then generate the given number of pairs of individuals for each type of relationship. The output is a file called "Fam_Sim" that contains the individual

genotypes for the desired number of pairs. The order of relationships simulated, and in the output file, are parent-offspring, full-sibs, half-sibs, an unrelated.

For example, suppose that you generated 50 pairs of individuals representing each type of relationship. Then the output would look like this:

Individual	Locus 1		Locus 2		...
	Allele 1	Allele 2	Allele 1	Allele 2	...
1	100	102	325	325	} First P-O pair
2	100	104	325	323	
3	104	104	321	325	} Second P-O pair
4	104	106	325	325	
...					
99	98	106	325	327	} Last P-O pair
100	98	98	321	325	
101	100	106	319	327	} First full-sib pair
102	100	106	323	325	
...					
199	98	100	319	321	} Last full-sib pair
200	100	100	319	325	
201	100	104	321	321	} First half-sib pair
202	98	98	319	321	
...					
299	106	108	325	325	} Last half-sib pair
300	102	104	327	327	
301	100	100	321	325	} First unrelated pair
302	104	106	325	327	
...					
399	100	102	321	327	} Last unrelated pair
400	98	108	325	325	

This format has its good and bad sides. On the good side, it is very clear who is related to who, and how. The bad side is that it is *these specific pairs* that have the desired relatedness values, but most relatedness programs give you ALL pairwise relatedness values. However, that is not really what you're interested in with these data.

For example, a program calculating all pairwise relatedness values will give you the relationships of:

- 1 & 2
- 1 & 3
- 1 & 4
- ...
- 2 & 3
- 2 & 4
- 2 & 5
- ...

But all that you are interested in is the value for the specific pairs 1 & 2, 3 & 4, etc...So, this means that you will need to severely edit your outfile from whatever relatedness program that you use, by removing all non-relevant comparisons, which will be most of them. Hopefully this makes sense. If not, let me know and I will explain it differently.

References

Galassi M, Davies J, Theiler J, Gough B, Jungman G, Booth M, Rossi F (2006) GNU Scientific Library Reference Manual - Revised Second Edition (v 18). Network Theory Ltd. Bristol UK.