# invertFREGENE: Software for simulating inversion polymorphisms in population genetic data

DOCUMENTATION

November 12, 2009

## 1 Overview

The package contains two programs: invertFREGENE and SAMPLE. invertFREGENE is the forward–in–time simulator of inversions in population genetic data, while SAMPLE samples genotype and haplotype data from the output of invertFREGENE simulations based on specified individual and marker ascertainment criteria. invertFREGENE has been developed from a beta version of the population genetic simulator FREGENE, and has a few missing features – the main ones being that loci cannot be subject to natural selection and there is no automatic time rescaling when simulating a smaller population for computational efficiency [1]. So although invertFREGENE retains much of the functionality of FREGENE, we provide self-contained documentation for invertFREGENE here.

## 2 Installation

invertFREGENE is a `C++` program that uses packages from the GNU Scientific Library (GSL), and has been developed and tested using the `g++` compiler on several Linux platforms.

1. Ensure that both `g++` and GSL are installed on your machine.
2. Download `invertfregene.tar.gz` and unpack it by typing
   `tar -zxf invertfregene.tar.gz`
   to generate the `invertFREGENE` directory.
3. This creates two sub–directories: `source_code` which has the invertFREGENE and SAMPLE source code and `analysis` which contains example command lines and a `data` directory which has input files (discussed below) for running invertFREGENE and SAMPLE.
4. invertFREGENE and SAMPLE are compiled by typing `make` in the `source_code` directory.

5. The installation can be checked by running `./run_example.sh` in the `analysis` directory.

# 3 invertFREGENE Quickstart

invertFREGENE simulates genetic data in essentially the same way as the forward–in–time simulator FREGENE [1, 2] but also simulates a single inversion polymorphism of a specified length, location and population frequency. Before simulating the inversion polymorphism, a population without an inversion should be simulated until homozygosity levels and pairwise $r^2$ (reflecting LD) values have reached equilibrium.

## 3.1 Simulating a population to equilibrium

Once invertFREGENE has been unpacked and the source files compiled, running the following line in the `analysis` directory simulates a population for 10000 generations (population size is $N = 1000$ here; equilibruim is reached in approx. $10N$ generations) without an inversion and writes it to the file `data/rin.xml`.

```
../source_code/invertFREGENE -i data/in.xml -p data/par.xml -gn 10000 -sd 446
-recombsd 235 -o data/rin.xml -s -freq > data/log.txt
```

The recombination and mutation rates are defined in the file `data/par.xml` and the population size and chromosome length in the file `data/in.xml`. The mutation rate, recombination model, population size and number of generations in the example are chosen such that the final population has linkage disequilibrium and homozygosity similar to that observed in humans. The recombination model is the same as that described in [3]; for more details see section 4 below. To simulate multiple populations with the same recombination map (hotspots of the same intensity in the same position etc) keep the recombination seed argument `-recombsd` fixed and change the seed argument `-sd`. To simulate multiple populations with different recombination maps both arguments should be changed. Local recombination rates are written to the file `rin.xml_RecombSummary.txt`. The `-s` switch writes additional summaries of the simulation to `rin.xml_Summary.txt`. The `-freq` option writes the minor (or mutant if `-NoAlleleFlip` option is used; see below) allele frequencies in the final population to `rin.xml_MAF.txt`.

## 3.2 Simulating an inversion from the equilibrium population

The output population from above, `rin.xml`, can now be used as the input population for a subsequent invertFREGENE simulation (a general feature of FREGENE). Running the following line in the `analysis` directory simulates an inversion in the equilibrium population `data/rin.xml`. The inversion starts at 0.75Mb, ends at 1.25Mb, and has a target frequency

(proportion) in the population of 0.4.

```
../source_code/invertFREGENE -i data/rin.xml -p data/par.xml -gn 50000 -sd 684
-recombsd 235 -o data/rin_inv.xml -StartOfInv 750000 -EndOfInv 1250000
-StopFreqOfInv 0.4 -MaxFreqOfLostInv 0.1 -s -freq > data/log_inv.txt
```

Using the same seed argument for `-recombsd` as used to generate the equilibrium population, `rin.xml`, ensures that the recombination map is the same. If the required inversion frequency is not reached by `-gn` generations or the frequency of a lost inversion exceeds `-MaxFreqOfLostInv` then the simulation automatically restarts with a new seed.

### 3.3 Sampling haplotype and genotype data from a population

The following line samples 500 individuals (`-controls`) from the population and writes out their haplotype and genotype data at all loci with population allele count greater than 100 (`-PopAlleleCount`), i.e. all SNPs with population MAF>5%, to the files `haplotype_0.txt` and `genotypes_0.txt`.

```
../source_code/SAMPLE -i data/rin_inv.xml -oh data/haplotypes -og data/genotypes
-controls 500 -PopAlleleCount 100 > data/log_sample.txt
```

The first row of the output files contains the chromosomal position of each SNP, and each subsequent row corresponds to a chromosome in the haplotype file or individual in the genotype file. Each element is a count of the mutant allele at the corresponding position.

## 4 invertFREGENE: Details and further options

For a population at equilibrium, average $r^2$ and mean homozygosity are a function of $N\mu$ and $N\rho$, where $N$ is the population size, $\mu$ is the per base mutation rate and $\rho$ is the per base recombination rate [2]. Observed homozygosity in humans reflects an effective population size of approximately 10000 [2]. In the provided example, computational time is saved by decreasing the population size by a factor of ten to 1000 and increasing the mutation and recombination rates accordingly by a factor of ten. Computation time is saved since not only is the population size reduced, but equilibrium, which is attained after approximately $10N$ generations [2], is reached in one tenth of the number of generations. For confirmation see the `output_file_name_Homozygosity.txt` output file which shows the mean homozygosity across all SNPs periodically throughout the simulation. In humans, the average per base mutation and recombination crossover rate have been estimated as $2.3 \times 10^{-8}$ [4] and $1.25 \times 10^{-8}$ [5],

respectively. Thus in our simulations, which had a population size of 1000, the rates used were ten times these (see the `par.xml` file).

The variability in recombination follows the model described in [3], which was developed using knowledge of recombination patterns (from the the genetic map [6] and other studies [7]) and other genetic parameters, with the aim of reflecting the variability in LD and allele frequencies observed in humans. In the model the crossover rates follow a hierarchical model consisting of four levels, with variation in rates at different scales of physical distance. So while recombination is modelled as varying over a range of megabases, the final level of the model consists of 2 kb recombination hotspots in which 88% of crossover recombinations occur. The model also incorporates gene conversions of length 500 bases with a constant per base mutation rate of $4.5 \times 10^{-9}$. Again, due to the rescaling of the population size the rate used in the example simulations was ten times this.

The program checks allele frequencies periodically, and SNPs with allele frequency greater than 0.5 have ancestral and mutant alleles exchanged. This saves computational time and memory by reducing the number of mutant sites stored (see [1, 2] for further details). This feature can be disabled by adding the flag `-NoAlleleFlip` to the command line, if, for example, one wants to inspect the derived allele frequencies at the SNPs.

## 4.1 Command line options

`-i file_name` – input population file (see details below)

`-p file_name` – parameter file (see details below)

`-o file_name` – output population file

`-gn int` – number of generations to run simulation

`-sd int` – random seed for simulation

`-recombsd int` – seed for recombination map (can be used to regenerate the recombination map)

`-StartOfInv int` – start position (in bases) of inversion

`-EndOfInv int` – end position (in bases) of inversion

`-StopFreqOfInv int` – target population frequency of inversion (a proportion between 0 and 1)

`-MaxFreqOfLostInv float` – maximum frequency (proportion between 0 and 1) a lost inversion can have reached before simulation is discarded and restarted.

`-sub int ... int` – specifies subpopulation structure. First argument is the number of subpopulations ($K$), the following $K$ arguments correspond to the population sizes (number of chromosomes) in each subpopulation. Default is for a single panmictic population.

`-mg float` – migration rate between subpopulations.

`-NoAlleleFlip` – switch to turn off exchange of ancestral and mutant alleles that occurs when the mutant allele frequency $> 0.5$.

`-seq` – switch to output sequence–like data. That is, alleles in the inverted sequence are not flipped back to their original position (i.e. that of a 'reference genome').

`-s` – switch for additional summary details of the simulation, writes to file `output_file_name_Summary.txt`.

`-freq` – switch to output minor (mutant) allele frequencies, writes to file `output_file_name_MAF.txt`.

## 4.2 Options passed by `-p` file

`<MUTAT_RATE>` – per base mutation rate.

`<RECOM_RATE>` – average per base recombination rate.

`<PROP_RECOM_HS>` – proportion of crossover. recombinations in hotspots

`<HS_LENGTH>` – length of crossover recombinations hotspots.

`<GC_RATE>` – per base gene–conversion rate.

`<GC_LENGTH>` – gene–conversion length.

The following options affect the efficiency and memory allocation of the simulation but not the final output population:

`<DELETION_INTERVAL>` – frequency, in number of generations, that allele frequencies are checked and flipped when $> 0.5$. Set to 500 in example file.

`<MATRIX_SIZE>` – controls memory allocation. Set to 1.0e+8 in example file. Particularly large simulations may require this number to be increased.

### 4.3 Input -i file

This file can be either the output from a previous invertFREGENE simulation or can be a new, homogeneous, population in which the user specifies the population size and length of the chromosome. The `in.xml` provided with the package defines an initial homogeneous population with 2000 chromosomes (1000 individuals), defined by `<SEGMENTS>2000</SEGMENTS>`, and length 2Mb, defined by `<CHROMO_LENGTH>2</CHROMO_LENGTH>`

### 4.4 Output files

The **screen** output, which provides any error messages and the progress of the simulation in generations, can be sent to a file by adding > `screen.out` to the end of the command line.

**Homozygosity** file: `output_file_name_Homozygosity.txt`. Output of homozygosity at fixed intervals during the simulation, corresponding in generation–spacing to that of the deletion interval defined in `par.xml` file. The homozygosity of the final population is also given. The file has two columns: the 1st is the generation number and the 2nd is the homozygosity. The expected homozygosity at equilibrium is given by $1/(1 + 4N\mu)$.

**Recombination** summary file: `output_file_name_RecombSummary.txt`. File has two columns: the 1st is chromosomal position and the 2nd is the recombination rate between that chromosomal position and the next (ie. that at the subsequent SNP).

**Inversion** summary file: `output_file_name_InvSummary.txt`. File has four columns: 1st – current number of chromosomes with the inverted sequence; 2nd – number of mutant alleles on the original inverted sequence; 3rd – maximum frequency of the inverted sequence of the current inversion; 4th – maximum frequency of an inversion that was lost during present simulation.

**Allele frequency** file: `output_file_name_MAF.txt`. File has two columns: the 1st is the chromosomal position and the 2nd is the minor allele frequency.

## 5 SAMPLE

SAMPLE generates haplotype and/or genotype data from an invertFREGENE output population. All options are set on the command line.

`-i file_name` **(required)**:
Input file (the output file of a invertFREGENE run, e.g. `data/rin_inv.xml`).

`-controls int` **(required)**:
Number of individuals to sample.

`-og file_name` option:
Records genotypes (in rows) of sampled individuals – coded as 0, 1, 2 (minor/mutant allele count).

`-oh file_name` option:
Records haplotypes (in rows) of sampled individuals – coded as 0 (major) and 1 (minor/mutant). Either `-og` or `-oh` must be specified; if both are specified then both output files are generated.

`-scan file_name` option:
File containing list of locations of SNPs to be output to the genotype and/or haplotype files.

`-PopAlleleCount int` option:
Specifies the minimum allele count in the population of SNPs that are output. If neither this option nor `-scan` are selected then all polymorphisms are written to the genotype and/or haplotype files.

`-chromolength float` option:
This option specifies the length of chromosome to output, in megabases, from the start of the chromosome. The default is to output the entire length.

`-LD file_name` option:
Provides file that lists all pairs of SNPs within 200kb of each other along with their pairwise $r^2$ value. (see `./data_sample/LD_example.txt`).

`-sd int` option:
Seed for random number generator.

`-samples int` option:
Specifies the number of sampled populations (default = 1).

SAMPLE can also sample individuals from various disease models. However, since this application focuses on inversionsm, this functionality is not described here. For details of the full functionality of SAMPLE see the related documentation of FREGENE.

# References

[1] M. Chadeau-Hyam, C. J. Hoggart, P. F. O'Reilly, J. C. Whittaker, M. De Iorio, and D. J. Balding. Fregene: simulation of realistic sequence-level data in populations and ascertained samples. *BMC Bioinformatics*, 9:364, 2008.

[2] C. J. Hoggart, M. Chadeau-Hyam, T. G. Clark, R. Lampariello, M. De Iorio, J. Whittaker, and D. J. Balding. Sequence-level population simulations over large genomic region. *Genetics*, 177(3):1725–31, 2007.

[3] S. F. Schaffner, C Foo, S Gabriel, D. Reich, M. J. Daly, and D. Altshuler. Calibrating a coalescent simulation of human genome sequence variation. *Genome Research*, 15:1576–1583, 2005.

[4] M. A. Jobling, M. E. Hurles, and C. Tyler-Smith. *Human Evolutionary Genetics: Origins, Peoples & Disease*. Garland Science, 2004.

[5] T. C. Matise, F. Chen, W. Chen, De La Vega F. M., M. Hansen, He C., F.C. Hyland, G. C. Kennedy, X. Kong, S. S. Murray, and et al. A second-generation combined linkage physical map of the human genome. *Genome Res.*, 17:1783Â1786, 2007.

[6] A. Kong, D. F. Gudbjartsson, J. Sainz, G. M. Jonsdottir, S. A. Gudjonsson, B. Richardsson, S. Sigurdardottir, J. Barnard, B. Hallbeck, G. Masson, A. Shlien, S. T. Palsson, M. L. Frigge, T. E. Thorgeirsson, J. R . Gulcher, and K. Stefansson. A high-resolution recombination map of the human genome. *Nature Genetics*, 31:241–7, 2002.

[7] A. J. Jeffreys and C. A. May. Intense and highly localized gene conversion activity in human meiotic crossover hot spots. *Nature Genetics*, 36:151–6, 2004.