FEST: Getting started manual

Jenny Welander [1]

Øivind Skare [2,3]

Thore Egeland [4,5]

[1]Department of Forensic Genetics and Forensic Toxicology, National Board of Forensic Medicine, 587 58 Linköping, Sweden
[2]Norwegian Institute of Public Health, 0403 Oslo, Norway
[3]Department of Public Health and Primary Health Care, University of Bergen, 5018 Bergen, Norway
[4]Institute of Forensic Medicine, University of Oslo, 0027 Oslo, Norway
[5]Oslo University College

Version: July 30, 2009

Contents

1 Introduction

The software FEST is an R package for simulations and likelihood calculations of pair-wise family relationships using DNA marker data. FEST is freely available and accompanies the paper by Skare, Egeland and Sheehan (2009). The software basically provides a front-end to Merlin (Abecasis et al., 2002), in which pair-wise relationships can be easily specified. The program converts the provided input to the appropriate format and then calls Merlin to simulate data and compute likelihoods. To avoid memory problems in R due to large input files, PERL scripts are used to create input files for the simulations and to organize the outputs before computing likelihoods.

FEST also contains a function for plotting pedigrees based on pair-wise family relationships (as used for the figures in this manual).

This manual briefly describes the different functions of FEST and gives examples of their usage. The complete R documentation for all functions and classes can be downloaded from the FEST webpage.

R commands are written in `Courier New` and comments are indicated by `##`, for example:

```
library(FEST)  ## This command loads the package FEST.
```

All commands used in this manual are also available in a .txt format, which can easily be copy-pasted into R.

Further help for the specific functions of FEST are available within R, for instance `help(PlotPedigree)` or by using the web interface of R starting with HELP> htm help.

2 Installation

The latest version of R can be downloaded from the R Project website (http://www.r-project.org/index.html).

The R package FEST and its documentation is available at http://folk.uio.no/thoree/FEST. It requires the following freeware to be installed and path variables to be set:

- Merlin (http://www.sph.umich.edu/csg/abecasis/merlin/index.html)
- Perl (http://www.activestate.com/activeperl/)

To set the path variables in Windows, right-click “My Computer”, select Properties>Advanced>Environment Variables, and add the locations of Merlin and Perl to the variable named Path.

When you load FEST into R for the first time you may be prompted to install other R packages (such as “kinship” and “hapsim”).

3 Definition of family relationships in FEST

In FEST, pair-wise family relationships can be of three different types:

1. An extended half-sibling relationship where two individuals A and B share one common ancestor. Such a relationship is denoted HS-n_1-n_2 if there are n_1 generations between the common ancestor and A, and n_2 generations between the common ancestor and B. Half-siblings thus constitute a HS-1-1 relationship, while two individuals sharing a single grandparent are denoted as HS-2-2. If $n_1 = n_2 = n$, the notation can be abbreviated as HS-n.

2. An extended sibling relationship where two individuals A and B share two common ancestors. Such a relationship is denoted S- n_1-n_2. Siblings are thus denoted as S-1-1, first-cousins as S-2-2, an uncle-niece relation as S-1-2, etc. If $n_1 = n_2 = n$, the notation can be abbreviated as S-n.

3. An extended parent-child relationship where A is the ancestor of B. Such a relationship is denoted PC-n if there are n generations between A and B. Parent-child is thus denoted PC-1 whereas grandparent-grandchild is denoted PC-2.

Examples of the different relation types are given in Figure 1.

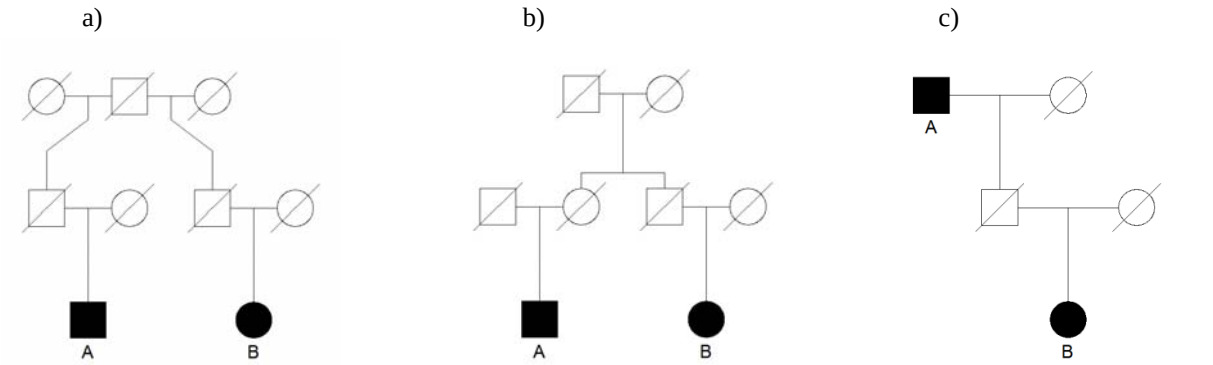

Figure 1 Family relationships of three different types. a) A HS-2-2 relation (two individuals sharing one grandparent). b) A S-2-2 relation (a pair of first-cousins). c) A PC-2 relation (grandparent and grandchild).

4 Plotting pedigrees

Pedigrees for given family relations can be plotted using the function `PlotPedigree`. Relationships should be on the form “HS-n_1-n_2”,” S-n_1-n_2” or “PC-n”. Pedigrees can be saved to pdf or postscript files using the arguments `plotfile` and `devtype`. Examples:

```
PlotPedigree("S-1-2")  ## Plots the pedigree of an uncle-niece
relation
title("Uncle-niece")  ## Adds a title to the pedigree



PlotPedigree("HS-3-3", plotfile="pedigree.pdf", devtype="pdf")
## Saves the pedigree of two individuals sharing a great-grandparent
to the file pedigree.pdf
```

5 Simulations

Genotype data for pairs of individuals can be simulated for given sets of true and alternative family relationships, using the function `SimulationStudy`. The relationships are specified using the function `SetModels` and should be on the form "HS-n_1-n_2"," S-n_1-n_2", "PC-n" or "unrelated". `SimulationStudy` returns an object of type "SimStudyObject-class", for which likelihoods can be calculated using `ComputeSummaryStatistcs`. An example of a simple simulation study with 50 biallelic markers is given below:

```
## 50 SNP simulation study

set.seed(17)  ## Random seed
nsim <- 5  ## Number of simulations
nmarker <- 50  ## Number of markers
maf <- 0.5  ## Minor allele frequency
chr <- c(1:22)  ## Number of chromosomes
mymodels <- c("S-1", "unrelated")  ## Family relations
models <- SetModels(trueModels=mymodels, altModels=mymodels)
simObj <- SimulationStudy(models, chr=chr, nmarker=nmarker, nsim=nsim,
maf=maf)


## Statistics computation

stat<-ComputeSummaryStatistics(simObj)


## Output

Sibs=stat$posterior[1,1,]  ## Posteriors
Unrelated=stat$posterior[1,2,]
res=rbind(Sibs, Unrelated)
loglik=simObj@logLik[[1]][[1]]  ## Likelihoods when "S-1" is true
row.names(loglik)=c("Sibs", "Unrelated")
colnames(loglik)=paste("Sim:", 1:nsim)
loglik
res
```

Results should look like:

```
Minor allele frequency  0.5
Number of markers  50      -  Tue Jun 09 17:07:23 2009
Simulation study finished  -  Tue Jun 09 17:07:26 2009
```

```
> loglik
            Sim: 1   Sim: 2   Sim: 3   Sim: 4   Sim: 5
Sibs       -84.628  -80.417  -80.417  -80.609  -82.626
Unrelated  -90.112  -90.805  -90.805  -88.726  -88.033


> res
                  S-1    unrelated
Sibs       0.9982079  0.001792099
Unrelated  0.1996913  0.800308676
```

The variable `loglik` contains natural logarithms of the likelihoods of the DNA data for each of the sibling simulations. The variable `res` contains the average posterior probabilities for the different true and alternative hypotheses, for each of the simulated relationships. Posterior probabilities are derived using a Bayesian approach with flat priors, resulting in

$$\Pr(Hypothesis \mid DNA) = \frac{LR}{LR+1},$$ where LR is the likelihood ratio.

Based on the above data we find for the first simulation

$$LR_1 = \frac{\exp(-84.628)}{\exp(-90.112)} = 240.808.$$

This is the likelihood ratio comparing alternatives ‘sibs’ and ‘unrelated’. For the first simulation we find

$$\Pr('sibs' \mid DNA) = \frac{240.808}{240.808+1} = 0.9959.$$

By averaging the results of the five simulations we get the posterior probability 0.9982 in the above table. Further details on the calculations are provided in the paper by Skare, Egeland and Sheehan (2009).

5.1 Simulations with real frequency information

Allele frequency data for the Affymetrix 500k chip, based on HapMap data (The International HapMap Consortium, 2005), can be downloaded from the FEST webpage as a file named *affy.RData*. The data contains frequency information for 500 000 biallelic markers spread over the 22 autosomal chromosomes. Below follows an example of a simulation with 2200 evenly spaced markers (make sure that the file *affy.RData* is located in your working directory). A frequency threshold can be used to remove markers with low minor allele frequencies:

```
## Simulation using HapMap frequencies
```

```
load("affy.RData")  ## Loads frequency data
set.seed(17)
models <- SetModels(trueModels=c("S-1", "HS-1", "unrelated"),
altModels=c("S-1", "HS-1", "unrelated"))  ## Family relations
nsim <- 5
nmarker <- 2200
chr <- c(1:22)
simObj2 <- SimulationStudy(models, chr=chr, nmarker=nmarker,
nsim=nsim, frequencyData=affy, freqThreshold=0.1)
stat2 <- ComputeSummaryStatistics(simObj2)

Sibs=stat2$posterior[1,1,]
Halfsibs=stat2$posterior[1,2,]
Unrelated=stat2$posterior[1,3,]
res2=rbind(Sibs, Halfsibs, Unrelated)
loglik2=simObj2@logLik[[1]][[1]]
row.names(loglik2)=c("Sibs", "Halfsibs", "Unrelated")
colnames(loglik2)=paste("Sim:", 1:nsim)
loglik2
res2
```

Results should look like:

```
> loglik2
               Sim: 1     Sim: 2     Sim: 3     Sim: 4     Sim: 5
Sibs       -3386.362  -3319.174  -3322.653  -3452.397  -3406.940
Halfsibs   -3539.260  -3532.190  -3549.165  -3535.744  -3558.693
Unrelated  -3829.298  -3831.368  -3876.338  -3741.351  -3822.084
> res2
                   HS-1           S-1      unrelated
Sibs       1.270246e-37  1.000000e+00   6.455090e-127
Halfsibs   1.000000e+00  4.357900e-13   3.282577e-25
Unrelated  1.296518e-14  1.374824e-35   1.000000e+00
```

If you use your own frequency data, it should be on the same format as the *affy.RData* (see R documentation for `affy`).

6 Analysis of real data

6.1 Input file formats

Input files should be on the formats presented by the Merlin webpage:

- One .dat file containing the marker identifiers (e.g. rsIDs)
- One .map file containing the genetic position of each marker (chromosome and position in cM)
- One .freq file containing the allele frequencies of each marker
- One .ped file containing family information and genotype data for all individuals

Examples of input files (*example.dat*, *example.map*, *example.freq* and *example.ped*) can be downloaded from the FEST webpage.

6.2 Likelihood calculations

Likelihoods and posterior probabilities can be computed for different relationship hypotheses using the function `RealStudy`. Input parameters are set using the function `SetDataPars`. Below follows an example using the example input files provided at the FEST website (make sure that they are located in your working directory):

```
## RealStudy with hypotheses "siblings" and "unrelated"

mypath <- "."
chrdirs <- NULL
suffixPed <- ".ped"
format <- "qtdt"
famList <- 1  ## Family number in .ped file
individualsTyped <- rbind(c(3,4))  ## Individuals to be analysed
(according to individual numbers in .ped file)
prefixInputFiles <-  "example"  ## Prefix of input files
dataPars <- SetDataPars(path=mypath, chrdirs=chrdirs,
suffixPed=suffixPed, prefixInputFiles=prefixInputFiles, format=format,
famList=famList, individualsTyped=individualsTyped)

realObj <- RealStudy(altModels=c("S-1", "unrelated"), dataPars)
realObj
```

Results should look like:

```
> realObj

$logLiks
           S-1   unrelated
3-4   -145.719   -156.616

$posterior
           S-1   unrelated
3-4  0.9999815   1.851335e-05

$nmarker
[1] 82
```

The variable `realObj` thus contains natural logarithms of the likelihoods, posterior probabilities of the different hypotheses, and the number of markers that have been analysed.

6.3 Sorting and thinning of data sets

If you have a large data set, you may want to thin it before analysis using the function `ThinMerlinInputFiles`. This is useful to:

- Remove very closely situated markers (using a genetic distance threshold)
- Remove markers with zero or very low minor allele frequencies (using a frequency threshold)
- Reduce the size of the data set

The use of very tightly linked markers might make unrelated persons appear related if linkage equilibrium (LE) is assumed (Huang et al., 2004). Experiences with our own data sets indicate that the assumption of LE is valid when map distances are larger than about 0.05 cM. If distances are smaller, linkage disequilibrium (LD), which is a dependency between alleles at a population level, appears to affect the results. Thinning is a practical, perhaps somewhat ad-hoc attempt to escape problems with LD.

If your data set is not already sorted according to chromosome order and increasing map distances, this should be performed before the thinning, using the function `SortMerlinInputFiles` ("file" should be replaced by the prefix of your files):

```
## Sort Merlin input files according to chromosome order and
increasing map distances

SortMerlinInputFiles("file.map", "file.dat", "file.freq", "file.ped",
prefix = "sorted_")  ## Sorts the markers and saves the sorted files
with the prefix "sorted_"
```

The sorting may take several minutes. When data has been sorted, it can be thinned using `ThinMerlinInputFiles`. Here you can select thresholds for the minor allele frequency and the minimal genetic distance (in cM) between markers:

```
## Thin Merlin input files based on frequencies and map distances

nMarkerThinned <- ThinMerlinInputFiles("sorted_file.map",
"sorted_file.dat", "sorted_file.freq", "sorted_file.ped",
limitCentiMorgan=0.1, freqThreshold=0.1, suffix="_thinned")
## Thins Merlin input files by selecting markers at distances >0.1 cM
and removing markers with minor allele frequencies <0.1

file.rename("sorted_file.map_thinned", "sorted_file2.map")
file.rename("sorted_file.dat_thinned", "sorted_file2.dat")
file.rename("sorted_file.freq_thinned", "sorted_file2.freq")
file.rename("sorted_file.ped_thinned", "sorted_file2.ped")
## Renames the thinned files (returns "TRUE" if succeeded)
```

The thinning may take several minutes as well. The sorted and thinned data set can now be analysed using `RealStudy`:

```
## RealStudy with hypotheses "siblings" and "unrelated"

mypath <- "."
chrdirs <- NULL
suffixPed <- ".ped"
format <- "qtdt"
famList <- 1  ## Family number in .ped file
individualsTyped <- rbind(c(3,4))  ## Individuals to be analysed
(according to individual numbers in the .ped file)
prefixInputFiles <-  "sorted_file2"  ## Prefix of input files
dataPars <- SetDataPars(path=mypath, chrdirs=chrdirs,
suffixPed=suffixPed, prefixInputFiles=prefixInputFiles, format=format,
famList=famList, individualsTyped=individualsTyped)

realObj <- RealStudy(altModels=c("S-1", "unrelated"), dataPars)
realObj
```

If sorted, a data set can also be thinned directly in `RealStudy`. Below follows a simple example using the example input files:

```
## RealStudy using cM and frequency thresholds

mypath <- "."
chrdirs <- NULL
suffixPed <- ".ped"
format <- "qtdt"
famList <- 1
individualsTyped <- rbind(c(3,4))
prefixInputFiles <-  "example"
dataPars <- SetDataPars(path=mypath, chrdirs=chrdirs,
suffixPed=suffixPed, prefixInputFiles=prefixInputFiles, format=format,
famList=famList, individualsTyped=individualsTyped)

realObj <- RealStudy(altModels=c("S-1", "unrelated"), dataPars,
limitCentiMorgan=60, freqThreshold=0.4) ## cM and frequency thresholds
realObj
```

Results:

```
> realObj

$logLiks
          S-1   unrelated
3-4   -49.894   -58.192

$posterior
          S-1   unrelated
3-4  0.999751   0.0002489524

$nmarker
[1] 28
```

7 References

Abecasis GR., Cherny SS., Cookson WO and Cardon LR. (2002). Merlin – rapid analysis of dense genetic maps using sparse gene flow trees. *Nature genetics* 30: 97-101.

Huang Q., Shete S. and Amos CI. (2004). Ignoring linkage disequilibrium among tightly linked markers induces false-positive evidence for affected sib pair analysis. *American journal of human genetics* 75: 1106-1112.

Skare Ø, Sheehan N and Egeland T. (2009). Identification of distant family relationships. *Bioinformatics*. Advance Access published online July 6, 2009.

The International HapMap Consortium. (2005). A haplotype map of the human genome. *Nature* 437: 1299-1320.