

Package ‘FEST’

January 2, 2012

Version 0.08

Date 24 August 2010 version of FEST, R version 2.11.0

Author Øivind Skare <oivind.skare@medisin.uio.no>

Maintainer Øivind Skare <oivind.skare@medisin.uio.no>

Description Identification of Family Relations using linked markers.

Title Identification of Family Relations using linked markers

Depends R (>= 2.4.0), methods, kinship, MASS, hapsim

SystemRequirements merlin, perl

License GPL (>= 2)

LazyLoad yes

Encoding UTF-8

URL <http://folk.uio.no/thoree/FEST>,
<http://www.sph.umich.edu/csg/abecasis/merlin/index.html>

Repository CRAN

Date/Publication 2010-08-25 13:11:23

R topics documented:

affy	2
affy.subset	3
ComputeSummaryStatistics	3
Model-class	4
Pedigree-class	5
PlotPedigree	6
RealStudy	7
SetDataPars	9
SetModels	10

SimStudyObject-class	11
SimulationStudy	12
SortMerlinInputFiles	14
ThinMerlinInputFiles	15

Index	17
--------------	-----------

affy	<i>Affymetrix 500K frequency data</i>
------	---------------------------------------

Description

Allele frequency data from the Affymetrix 500K chip derived from the HapMap data. It contains frequency information for all 22 autosomal chromosomes. See also [affy.subset](#).

Usage

```
## download file from 'http://folk.uio.no/thoree/FEST/affy.RData'
```

Format

List of length 22 where element *i* (`affy[[i]]`) is a data frame with the following columns:

```
SNP Name of SNP
cM Distance in centi Morgan
A Frequency of allele A
C Frequency of allele C
```

Source

<http://www.stats.ox.ac.uk/~marchini/software/gwas/chiamo.html>

Examples

```
## download file from 'http://folk.uio.no/thoree/FEST/affy.RData'
## Not run: load("affy.RData")
## Average frequencies for chromosome 1:
## Not run: f <- apply(affy[[1]][, -c(1,2)], 2, mean)
## Not run: print(f)

## Minor allele frequency for chromosome 1
## Not run: maf1 <- pmin(affy[[1]][,3], affy[[1]][,4])
## Empirical cumulative distribution function
## Not run: plot(ecdf(maf1), xlim=c(0,0.5))
## Some statistics
## Not run: stat <- c(length(maf1), mean(maf1>0.01), mean(maf1>0.02), mean(maf1>0.05))
```

`affy.subset`*Subset of Affymetrix 500K frequency data*

Description

A subset of the allele frequency data from the Affymetrix 500K chip derived from the HapMap data. It contains frequency information of 100 markers for all 22 autosomal chromosomes.

Usage

```
data(affy.subset)
```

Format

See [affy](#).

Examples

```
data(affy.subset)
## Average frequencies for chromosome 1:
f <- apply(affy.subset[[1]][,-c(1,2)],2,mean)
print(f)

## Minor allele frequency for chromosome 1
maf1 <- pmin(affy.subset[[1]][,3], affy.subset[[1]][,4])
## Empirical cumulative distribution function
plot(ecdf(maf1), xlim=c(0,0.5))
## Some statistics
stat <- c(length(maf1), mean(maf1>0.01), mean(maf1>0.02), mean(maf1>0.05))
```

`ComputeSummaryStatistics`*Computes summary likelihood statistics*

Description

The function computes summary likelihood statistics for an object of type [SimStudyObject-class](#) that is returned from the function [SimulationStudy](#).

Usage

```
ComputeSummaryStatistics(simObj, altHyp = "unrelated")
```

Arguments

`simObj` Object of type [SimStudyObject-class](#)
`altHyp` The alternative hypothesis to be used when doing likelihood ratio statistics

Value

posterior	Three dimensional array, where element (i,j,k) is the average posterior probability for simulation sub study i, true model j and alternative model k. The sub studies correspond to the different number of markers specified in SimulationStudy
posterior.sd	Standard deviation of the average posterior values. Same data structure as for posterior.
likrat	Average likelihood ratio between true models and altHyp. Same data structure as for posterior.
likrat.sd	Standard deviation of the average likelihood ratio. Same data structure as for posterior.
classrat	Average number of times the likelihood ratio is positive. Same data structure as for posterior.
classrat.sd	Standard deviation of the classrat statistics. Same data structure as for posterior.

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[SimStudyObject-class](#), [SimulationStudy](#)

Model-class

Class "Model"

Description

Contains true and alternative pairwise family relations. An object of class `Model` is used as input to [SimulationStudy](#).

Details

Models are set by the function [SetModels](#).

Objects from the Class

Objects can be created by calls of the form `new("Model", ...)`.

Slots

true: True family relations.

alternative: Alternative family relations. Specified for each true family relation.

Methods

No methods defined with class "Model" in the signature.

Author(s)

Oivind Skare <oivind.skare@medisin.uio.no>

See Also

[SetModels](#), [SimulationStudy](#)

Examples

```
models1 <- SetModels(trueModels=paste("HS-", 1:6, sep=""),
                    altModels=c("true", "unrelated"))
modellist <- paste("HS-", 1:5, sep="")
models2 <- SetModels(trueModels=modellist, altModels=c(modellist, "unrelated"))
```

Pedigree-class	<i>Class "Pedigree"</i>
----------------	-------------------------

Description

Contains information about pedigree structure.

Objects from the Class

Objects can be created by calls of the form `new("Pedigree", ...)`. Internal class used in FEST.

Slots

fam: Integer vector with family numbers.

subj: Integer vector with subject numbers.

fatherID: Integer vector with ID of father.

motherID: Integer vector with ID of mother.

gender: Integer vector specifying gender (1=father, 2=mother).

affected: Integer vector with affected status (0=unaffected, 1=affected).

typed: Integer vector with indices of typed persons.

Methods

initialize signature(.Object = "Pedigree"): ...

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[PlotPedigree](#)

Examples

```
showClass("Pedigree")
```

PlotPedigree

Plot of a pedigree

Description

Given a family relation or a pedigree file, the function plots the pedigree.

Usage

```
PlotPedigree(reltype, plotfile, devtype = NULL)
```

Arguments

reltype	Character string specifying the family relation. Either 'reltype' or 'pedfile' must be specified. 'reltype' must be one of the family relations listed in SetModels .
plotfile	The plot is written to this file if 'devtype' is different from NULL.
devtype	Device type. Possible values: 'pdf', 'postscript', NULL. If NULL, the plot is not written to file 'plotfile'.

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

Examples

```
PlotPedigree("S-1-4")  
PlotPedigree("HS-3-4")  
PlotPedigree("PC-3")
```

RealStudy	<i>Analysis of alternative hypothesized family relations for observed marker data</i>
-----------	---

Description

Based on observed marker data, likelihood and posterior values are computed for alternative hypothesized family relations.

Usage

```
RealStudy(altModels, dataPars, saveMerlinFiles, limitCentiMorgan=0,  
          freqThreshold=0)
```

Arguments

altModels	Vector of strings that gives alternative family relations. See SetModels for a list of the family relations that can be used.
dataPars	List: Output from function SetDataPars
saveMerlinFiles	If TRUE the files used as input to the likelihood computations in merlin are saved. Default value is FALSE.
limitCentiMorgan	Markers are thinned such that the distance between two consecutive markers are larger than this limit.
freqThreshold	Includes only markers with minor allele frequency larger than this threshold.

Value

logLiks	log likelihood values
posterior	posterior values

Author(s)

Øivind Skare <ovind.skare@medisin.uio.no>

References

<http://folk.uio.no/thoree/FEST>

Øivind Skare, Nuala Sheehan, and Thore Egeland Identification of distant family relationships Bioinformatics Advance Access published on July 6, 2009.

Examples

```

## The example can not be run, user must supply data
## Not run: pathDataDir <- "../Data/RealData/" ## path to data directory
## Not run: chrdirs <- paste("Chr", 1:22, sep="")
## Not run: dataPars <- SetDataPars(pathDataDir, chrdirs=chrdirs,
prefixInputFiles="", format="linkage", famList=32,
individualsTyped=rbind(c(32,33), c(31,33), c(30,33), c(26,33)))
## End(Not run)
## Not run: realObj <- RealStudy(altModels=c("S-1", "HS-1", "S-2", "S-3","unrelated"),
dataPars)
## End(Not run)

## A simple test example: Two persons assumed to be half-sibs, 2 markers
## Make first data files (qtdt format)
ped <- rbind(c(1, 1, 0, 0, 2, 0, 0, 0, 0, 0),
             c(1, 2, 0, 0, 1, 0, 0, 0, 0, 0),
             c(1, 3, 0, 0, 2, 0, 0, 0, 0, 0),
             c(1, 4, 2, 1, 1, 1, 2, 1, 1, 1),
             c(1, 5, 2, 3, 1, 1, 1, 2, 2, 2))
dat <- rbind("M locus1", "M locus2")
freq <- rbind("M locus1", "F 0.4 0.6", "M locus2", "F 0.7 0.3")
map <- rbind("CHROMOSOME MARKER POSITION",
            "1 locus1 123.4",
            "1 locus2 136.2")

write.table(ped, file="test1.ped", col.names=FALSE, row.names=FALSE)
write.table(dat, file="test1.dat", col.names=FALSE, row.names=FALSE, quote=FALSE)
write.table(freq, file="test1.freq", col.names=FALSE, row.names=FALSE, quote=FALSE)
write.table(map, file="test1.map", col.names=FALSE, row.names=FALSE, quote=FALSE)

## Analysis of this data set
mypath <- "."
chrdirs <- NULL
suffixPed <- ".ped"
format <- "qtdt"
famList <- 1
individualsTyped <- cbind(4,5)
prefixInputFiles <- "test1"
dataPars <- SetDataPars(path=mypath, chrdirs=chrdirs,
                        suffixPed=suffixPed,
                        prefixInputFiles=prefixInputFiles, format=format,
                        famList=famList, individualsTyped=individualsTyped)
realObj <- RealStudy(altModels=c("HS-1", "HS-2", "HS-3",
                                "S-1", "unrelated"), dataPars)

## realObj$posterior
##           HS-1      HS-2      HS-3      S-1  unrelated
##4-5 0.1440720 0.2432792 0.2669072 0.07134156 0.2743999

## Take away the second locus and compare three family relations that
## should give equal likelihood for 1 marker (and more generally for
## unlinked markers)

dat <- rbind("M locus1", "S2 locus2") # S2: skips the locus

```

```

write.table(dat, file="test1.dat", col.names=FALSE, row.names=FALSE, quote=FALSE)
realObj <- RealStudy(altModels=c("HS-1", "S-1-2", "PC-2",
                                "unrelated"), dataPars)

## realObj$logLiks
##      HS-1      S-1-2      PC-2 unrelated
##4-5 -2.448768 -2.448768 -2.448768 -2.566551
## realObj$posterior
##      HS-1      S-1-2      PC-2 unrelated
##4-5 0.2571429 0.2571429 0.2571429 0.2285714

```

SetDataPars

Set input parameters for function RealStudy

Description

Set input parameters for function RealStudy.

Usage

```

SetDataPars(path, chrdirs = NULL, prefixInputFiles = NULL, suffixPed = ".pre",
            format = c("qtdt", "linkage"), famList = NULL, individualsTyped = NULL)

```

Arguments

path	The data directory.
chrdirs	Sub directories relative to the data directory: the data directories for the different chromosomes. A sub directory may contain data on several chromosomes. Default is NULL, i.e. no sub directories, the data files are in the directory given by 'path'.
prefixInputFiles	Prefix for int input files. If prefixInputFiles=NULL, then it is assumed that there are unique input files corresponding to the default suffixes: .dat (linkage format), .dat, .freq, .map (qtdt format).
suffixPed	Suffix for the pedigree files.
format	'qtdt' or 'linkage'
famList	Vector: families included in the analysis
individualsTyped	Matrix: Number of rows equal number of families included in the analysis. Row i: the two individuals which relationship is to be tested

Value

chrdirs	Data directories for the different chromosomes
suffixPed	Suffix for the pedigree files.
format	'qtdt' or 'linkage'

fam Vector: families included in the analysis
 individualsTyped Matrix: Number of rows equal number of families included in the analysis. Row
 i: the two individuals which relationship is to be tested

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[RealStudy](#)

SetModels

Specify true and alternative family relations

Description

Specify true and alternative family relations.

Usage

```
SetModels(trueModels, altModels = c("true", "unrelated"))
```

Arguments

trueModels a character array of the true family relations. See details.
 altModels a character array of alternative hypothesised family relations. See details.

Details

The true and alternative pairwise family relations are among the following:

HS-k-l: Generalised half-sib relationships where k=1, 2, ... and l=1, 2,... 'HS-k-l' specifies a relation where two persons sharing one common ancestor; one person is k generation apart and the other is l generation apart. Examples: 'HS-1-1' is a half-sib relation, 'HS-2-2' specifies two persons sharing a grandparent.

HS-k: Same as HS-k-k

S-k-l: Generalised sibling relationships where k=1, 2 ... and l=1, 2,... 'S-k-l' specifies a relation where two persons are sharing one common ancestor; one person is k+1 generation apart and the other is l+1 generation apart. Examples: 'S-1-1' specifies sibs, 'S-2-2' first cousins, 'S-1-2' an uncle-niece relation.

S-k: Same as S-k-k

PC-k: Generalised sibling relationships where k=1, 2 ... 'PC-k' specifies a relation where the second person is the ancestor of first person, k generations apart. Examples: 'PC-1' specifies a parent-child relation, 'PC-2' a grandparent-grandchild relation.

unrelated: Two unrelated persons

In addition altModels may also contain the following specifications:

true: Same models as specified in trueModels

upper: A generation-less relation than the true model specified in trueModels

lower: A generation-more relation than the true model specified in trueModels

The available family relations are constrained by default setting in merling for maximal allowed pedigree complexity allowed by merlin <http://www.sph.umich.edu/csg/abecasis/Merlin/index.html>. For the moment we allow HS-k-l and S-k-l where $k+l \leq 25$, and PC-k where $k \leq 25$.

Value

An object of class `Model`.

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[Model](#)

Examples

```
## Example 1
modellist <- paste("HS-", 1:5, sep="")
models <- SetModels(trueModels=modellist, altModels=c(modellist, "unrelated"))
## Example 2
models <- SetModels(trueModels=paste("HS-", 1:6, sep=""),
                    altModels=c("true", "unrelated"))
```

SimStudyObject-class *Class "SimStudyObject"*

Description

An object returned by the 'SimulationStudy' function. Used as input to [ComputeSummaryStatistics](#).

Objects from the Class

Objects can be created by calls of the form `new("SimStudyObject", ...)`. describe objects here

Slots

posterior: Two dimensional list structure, where `posterior[[i]][[j]]` is a matrix of posterior probabilities for data set *i* and true model *j*. Element (*k*,*l*) of the matrix contains the posterior probability of alternative model *k* for simulation *j*.

logLik: Same data structure as for `posterior` containing the log likelihood values.

nsim: Number of simulations.

nmarker: Number of markers.

maf: Minor allele frequency

freqThreshold: A frequency threshold that have been used in selecting markers. Only markers with minor allele frequency larger than this threshold have been considered for selection.

model: Object of class `Model-class`.

Methods

initialize signature(.Object = "SimStudyObject"): ...

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[SimulationStudy](#)

SimulationStudy

Simulation study of family relationships

Description

Assume a given set of true and alternative family relationships. Use simulations and exact likelihood computations to compute likelihood and posterior values. These computations are done for different number of markers.

Usage

```
SimulationStudy(models, chr=c(1:22), nmarker = c(22 * c(1, 10, 100, 1000), 5e+05), nsim = c(1000, 1000, 1000, 1000, 400), maf = numeric(), frequencyData = NULL, freqThreshold = c(rep(0.1, 4), 0), saveMerlinFiles=FALSE, verbose=TRUE)
```

Arguments

<code>models</code>	Object of type <code>Model</code> that specifies true and alternative family relations.
<code>chr</code>	Either a vector or a list. If <code>chr</code> is a vector it contains the chromosomes included in the simulation study. If it is list it should have same length as the <code>nmarker</code> vector. Each element of the list is a vector containing chromosomes included in the simulation study.
<code>nmarker</code>	Vector of number of markers. A simulation study is done for each number of markers.
<code>nsim</code>	Number of simulations. A vector with same length as the <code>nmarker</code> vector.
<code>maf</code>	Minor allele frequency. Same for all SNPs, must be specified if <code>frequencyData</code> is not specified.
<code>frequencyData</code>	A list with frequency information for each chromosome. See affy for a description of the format.
<code>freqThreshold</code>	Selects a sub set of the SNPs in <code>frequencyData</code> : only SNPs with minor allele frequency > <code>freqThreshold</code> are retained. Vector with same length as the <code>nmarker</code> vector.
<code>saveMerlinFiles</code>	If TRUE the files used as input to the likelihood computations in merlin are saved. Default value is FALSE.
<code>verbose</code>	If TRUE, information about simulations are output to screen. Default TRUE.

Value

An object of type `SimStudyObject-class`.

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

References

<http://folk.uio.no/thoree/FEST>

Øivind Skare, Nuala Sheehan, and Thore Egeland Identification of distant family relationships *Bioinformatics Advance Access published on July 6, 2009.*

See Also

[SetModels](#)

Examples

```
set.seed(17)
models <- SetModels(trueModels=paste("HS-", 1:6, sep=""),
                    altModels=c("true", "unrelated"))
nsim <- rep(10, 2)
nmarker <- 22*c(1, 10)
chr <- c(1:22)
```

```

simObj1 <- SimulationStudy(models, chr=chr, nmarker=nmarker, nsim=nsim, maf=0.5)
stat1 <- ComputeSummaryStatistics(simObj1)
## Average posterior results for no of markers=22:
print(round(stat1$posterior[1,,],4)) # rows: true models, columns: alternative models
###      HS-1  HS-2  HS-3  HS-4  HS-5 HS-6 unrelated
### HS-1 0.5284   NA   NA   NA   NA   NA   0.4716
### HS-2   NA 0.4821   NA   NA   NA   NA   0.5179
### HS-3   NA   NA 0.4945   NA   NA   NA   0.5055
### HS-4   NA   NA   NA 0.5004   NA   NA   0.4996
### HS-5   NA   NA   NA   NA 0.5001   NA   0.4999
### HS-6   NA   NA   NA   NA   NA 0.5   0.5000
## No of markers=220:
print(round(stat1$posterior[2,,],4))

## Simulation study using Affymetrix frequency data
## The complete 500K data file may be downloaded from
## 'http://folk.uio.no/thoree/FEST/affy.RData'
## load("affy.RData")

## A small subset of the Affymetrix frequency data. 100 markers on
## each chromosome
data(affy.subset)
simObj2 <- SimulationStudy(models, chr=chr, nmarker=nmarker, nsim=nsim,
                          frequencyData=affy.subset, freqThreshold=c(0.1,0.1))
stat2 <- ComputeSummaryStatistics(simObj2)
## Average posterior results (Affymetrix):
## No of markers=22:
print(round(stat1$posterior[1,,],4)) # rows: true models, columns: alternative models
## No of markers=220:
print(round(stat1$posterior[2,,],4))

```

SortMerlinInputFiles *Sorting of Merlin input files*

Description

Merlin input files are sorted according to chromosome order and increasing genetic map distance in centiMorgans (cM).

Usage

```
SortMerlinInputFiles(mapfile, datfile, freqfile, pedfile, nNotMarker = 5,
                    prefix = "sorted_", chr = NULL, excludeSNP = NULL)
```

Arguments

mapfile	File of the type filename.map which contains the genetic position of each marker (chromosome and position in cM).
datfile	File of the type filename.dat which contains the marker identifiers (e.g. rsIDs).

freqfile	File of the type filename.freq which contains the allele frequencies of each marker.
pedfile	File of the type filename.ped which contains family information and genotype data for all individuals.
nNotMarker	Number of columns in the .ped file that precedes the marker columns (containing genotypes). Default value is 5.
prefix	Prefix given to the sorted files. Default is "sorted_".
chr	Vector describing the chromosomes to be included, for example c(1:10). Default is NULL (meaning all chromosomes).
excludeSNP	Identifiers (e.g. rsIDs) of SNPs that should be excluded. Default is NULL.

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[ThinMerlinInputFiles](#)

Examples

```
## The example can not be run, user must supply data.
## Replace "filename" with the name of your files.

## SortMerlinInputFiles("filename.map", "filename.dat", "filename.freq",
##                       "filename.ped", prefix = "sorted_")
```

ThinMerlinInputFiles *Thinning of Merlin Input files*

Description

Merlin input files are thinned based on thresholds for minor allele frequency and minimal distance (in cM) between markers.

Usage

```
ThinMerlinInputFiles(mapfile, datfile, freqfile, pedfile, nNotMarker = 5,
                     limitCentiMorgan = 0, freqThreshold = 0, suffix = "_thinned")
```

Arguments

mapfile	File of the type filename.map which contains the genetic position of each marker (chromosome and position in cM).
datfile	File of the type filename.dat which contains the marker identifiers (e.g. rsIDs).
freqfile	File of the type filename.freq which contains the allele frequencies of each marker.
pedfile	File of the type filename.ped which contains family information and genotype data for all individuals.
nNotMarker	Number of columns in pedfile that precedes the marker columns. Default value equal 5.
limitCentiMorgan	The data set is thinned such that the distance between two consecutive markers is larger than this limit. Default value is 0.
freqThreshold	Markers with a minor allele frequency smaller or equal to this limit will be removed. Default value is 0.
suffix	Suffix that will be added to the filenames of the thinned files.

Author(s)

Øivind Skare <oivind.skare@medisin.uio.no>

See Also

[SortMerlinInputFiles](#)

Examples

```
## The example can not be run, user must supply data.
## Replace "file" with the name of your files.

## nMarkerThinned <- ThinMerlinInputFiles("file.map", "file.dat", "file.freq", "file.ped",
##                                     limitCentiMorgan=0.1, freqThreshold=0.1,
##                                     suffix="_thinned")
## file.rename("file.map_thinned", "file2.map")
## file.rename("file.dat_thinned", "file2.dat")
## file.rename("file.freq_thinned", "file2.freq")
## file.rename("file.ped_thinned", "file2.ped")
```

Index

*Topic **classes**

- Model-class, 4
- Pedigree-class, 5
- SimStudyObject-class, 11

*Topic **datasets**

- affy, 2
- affy.subset, 3

*Topic **data**

- SetDataPars, 9
- SortMerlinInputFiles, 14
- ThinMerlinInputFiles, 15

*Topic **hplot**

- PlotPedigree, 6

*Topic **htest**

- RealStudy, 7
- SimulationStudy, 12

*Topic **iteration**

- RealStudy, 7
- SimulationStudy, 12

*Topic **models**

- SetModels, 10

SetDataPars, 7, 9

SetModels, 4–7, 10, 13

SimStudyObject-class, 3, 4, 13

SimStudyObject-class, 11

SimulationStudy, 3–5, 12, 12

SortMerlinInputFiles, 14, 16

ThinMerlinInputFiles, 15, 15

affy, 2, 3, 13

affy.subset, 2, 3

ComputeSummaryStatistics, 3, 11

initialize, Pedigree-method

(Pedigree-class), 5

initialize, SimStudyObject-method

(SimStudyObject-class), 11

Model, 11

Model-class, 12

Model-class, 4

Pedigree-class, 5

PlotPedigree, 6, 6

RealStudy, 7, 10