

Package ‘fam2r’

March 29, 2017

Type Package

Title From 'Familias' to R

Version 1.2

Date 2017-03-29

Author Thore Egeland and Magnus Dehli Vigeland.

Maintainer Thore Egeland <Thore.Egeland@nmbu.no>

Depends R (>= 3.2.0), Familias, paramlink

Description Functionality provided for conditional simulation, likelihoods and plotting of pedigrees, mostly as a wrapper for 'paramlink'. Users typically start by exporting from the Windows version of 'Familias'.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-29 12:17:33 UTC

R topics documented:

fam2r-package	2
conditionalLR	3
FamiliasConditional	8
grandmother	10
LRparamlink	11
missing.person.plot	12
paramlinkConditional	13
PE	16
Index	19

 fam2r-package

 From 'Familias' to R

Description

Conditional simulation of marker data on pedigrees, plotting and likelihood calculations, using [Familias](#) or [paramlink](#), on pedigrees. The package is specifically tailored to work on export of data from windows [Familias](#).

Details

The main function is `conditionalLR`. Marker data for a specified individual is simulated conditionally on pedigrees and previously typed individuals using `markerSim`. Then likelihoods are calculated using [Familias](#) or [paramlink](#).

Author(s)

Thore Egeland <thore.egeland@gmail.com> and Magnus Dehli Vigeland.

Maintainer: Thore Egeland

References

Kling et al (2017)

Examples

```
# Example. There is one marker. A grand mother (GM=1) is
# is genotyped and we simulate the genotype of the grand son (GS=POI).
data(grandmother)
pedigrees = grandmother$pedigrees
datamatrix = grandmother$datamatrix
loci = grandmother$loci
x = Familias2linkdat(pedigrees, datamatrix, loci)
missing.person.plot(x[[1]], missing=3, marker = 1, newdev=TRUE)
Nsim = 5 #Increase to 1000
# Calculations can be done using 'Familias' or 'paramlink':
res1 = FamiliasConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
  truePed = 1, available = "GS", ref=2, seed=17)
res2 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
  truePed = 1, available = "GS", ref=2, seed=17)
LR1 = data.frame(LR.H1.Familias=res1[[1]][,1], LR.H1.paramlink=res2[[1]][,1])
# Checks that 'Familias' and 'paramlink' gives the same result:
stopifnot(all(apply(LR1,1, function(x) abs(x[1]-x[2])<1e-10)))

# Alternatively, there is a wrapper function
res3 = conditionalLR(Nsim = Nsim, datamatrix, loci, pedigrees,
  available = "GS", ref=2, seed=17, verbose=FALSE,
  program="Familias", simplify=TRUE)
```

conditionalLR	<i>Simulates marker data on pedigrees conditionally on typed individuals and calculates likelihoods</i>
---------------	---

Description

Marker data are simulated on pedigrees, conditional on existing genotypes and likelihoods are calculated using [FamiliasConditional](#) or [paramlinkConditional](#). `transferMarkerdata2` transfers markerdata from a `linkdat` object to a `linkdat` object or a list thereby generalising [transferMarkerdata](#).

Usage

```
conditionalLR(Nsim=5, datamatrix, loci, pedigrees, file = NULL , program = "Familias",
prior=NULL, available=NULL, seed=NULL, ref=NULL, truePeds = NULL,
verbose = TRUE, simplify = FALSE)
```

```
# Remove persons, e.g., singletons for all hypotheses
removePersons(pedigrees, datamatrix, ids=NULL)
```

```
label2num(label, familiasped)
```

Arguments

Nsim	Integer. Number of simulations.
datamatrix	A data frame. The row names must be the names of the persons you have data for. The columns contain the alleles, two columns for each marker, in the same order used in the loci list.
loci	A <code>FamiliasLocus</code> object or a list of such objects.
pedigrees	An object of type <code>'FamiliasPedigree'</code> or <code>'pedigree'</code> , or a list of such objects.
familiasped	An object of type <code>'FamiliasPedigree'</code> or <code>'pedigree'</code> , or a list of such objects.
file	Character. First part of name for output file. If NULL nothing is written to files.
program	Character. Specifies program used for likelihood calculation. Either <code>'Familias'</code> (default) or <code>'paramlink'</code> .
prior	Double vector. Not currently relevant as only LRs are reported.
available	Character or integer identifying person to be simulated.
seed	Integer.
ref	Integer. Index of pedigree in numerator of LR. Set to last if NULL.
truePeds	Indices of pedigrees to be simulated from. If NULL, all.
simplify	Logical. <code>simplify=TRUE</code> is used for the standard cases, those with two hypotheses.
verbose	logical. If TRUE output is explained.
label	A character vector, <code>'Familias'</code> ID label.
ids	A character or integer vector.

Details

If truePeds is a subset of all pedigrees, only the files corresponding to truePeds are written. In this case, LR[, , i] contains missing values if i is not in truePeds

Value

The number of hypotheses corresponds to the number of pedigrees. In many cases, there will only be two hypotheses. The output is then simplified if one specifies simplify = TRUE. By default the reference hypothesis is number 2. i.e., ref=2) and the likelihood ratio is $LR = \Pr(\text{data}|H1)/\Pr(\text{data}|H2)$. The output will then be a matrix with columns. The first column consists of the simulated LR-s when H1 is assumed true, the second one when H2 is true. When there are more than two hypotheses, an array is returned. LR[, , i] are the LR values. when simulations are conditioned on pedigree i. There is one row for each simulation and one column for each pedigree. In other words LR[k, l, i] is the k-th simulated value of $\Pr(\text{data}|H1)/\Pr(\text{dat}|H\text{ref})$ when hypothesis Hi is true, the one simulated from. File(s) with simulated values are written (if variable file is not NULL). If simplify=TRUE one file is written, otherwise there will be one one for each pedigree i.

Author(s)

Thore Egeland <Thore.Egeland@gmail.com>

References

Kling et al. (2017).

Examples

```
# Example
# Computational details for (currently) Example 1 of Kling et al (2017) are provided.
# The purpose is to explain the code and also check results against exact formulae and compare
# to a previous implementation, i.e., library(famr). # There is one marker with alleles 1, 2
# and 3 having frequencies p1, p2, and p3. # We disregard complicating factors like mutation
# (see next example), # and silent alleles. # One person, the grandmother GM, is genotyped as 1/1.
# The grandson GS is to be simulated.

# The dataset 'grandmother' loaded below is a list with three components explained
# in the documentation of FamiliasPosterior, i.e, pedigrees, datamatrix and loci.
# The paramlink function 'Familias2linkdat' converts from 'Familias' format to a format
# suitable for plotting and conditional simulation, likelihood calculations etc,
# using 'paramlink'; below 'plotPedList' is used to plot.

data(grandmother)
pedigrees = grandmother$pedigrees
datamatrix = grandmother$datamatrix
loci = grandmother$loci
persons = rownames(datamatrix)
## Not run:
# Plot with newdev=TRUE, resize plot window and then plot with newdev=FALSE
x = Familias2linkdat(pedigrees, datamatrix, loci)
plotPedList(x, newdev=TRUE, frametitles=c("H1", "H2"),
  available = "shaded", marker=1, skip.empty.genotypes = TRUE)
```

```

## End(Not run)
# In this case there are two (=length(pedigrees))
# hypotheses H1 and H2 or equivalently two pedigrees.
# We will be interested in the likelihood ratio (LR) defined by  $\Pr(\text{data}|H1)/\Pr(\text{data}|H2)$ 
# or rather the random variables LR(H1) and LR(H2) where H1 and H2 indicate the true hypotheses.
# In other words, we estimate the distribution of LR(H1) by simulating assuming H1 to be true
# and similarly for H2. Obviously, we also condition on genotyped individuals, GM in this case.
# Assume first H1 to be true. Then, as explained in the Kling et al. (2017),
# there are three possible values for the likelihood ratios,
# namely  $y1=1/2$ ,  $y2=0.5+1/(4*p1)$ , and  $y3=0.5+0.5/p1$ 
# occurring with the probabilities  $py1$ ,  $py2$  and  $py3$  calculated below

p = loci[[1]]$alleles
p1 = p[1]; p2=p[2]; p3=p[3]
py2 = (1-p1)*(p1+0.5)
py3 = 0.5*p1*(1+p1)
py1 = 1-py2-py3
y1 = 0.5; y2 = 0.5+1/(4*p1); y3 =0.5+0.5/p1; LRs = c(y1, y2, y3)
LR.H1.exact = c(py1,py2,py3)
names(LR.H1.exact) = paste(LRs)

# The above probability distribution, LR.H1.exact, can be approximated by simulation,
# using 'markerSim' followed by likelihood calculation in 'Familias' or 'paramlink'.

Nsim = 5; seed = 17; avail = "GS"
res1 = conditionalLR(Nsim=Nsim, datamatrix, loci, pedigrees, available=avail,
                    seed=seed, program = "Familias", simplify=TRUE)
res2 = conditionalLR(Nsim=Nsim, datamatrix, loci, pedigrees, available=avail,
                    seed=seed, program = "paramlink", simplify=TRUE) #Change

LR.H1.Familias = table(res1[,1])/Nsim
LR.H1.paramlink = table(res2[,1])/Nsim
stopifnot(round(LR.H1.Familias-LR.H1.paramlink,12)==0)
# We see that the two implementations give the same result and agree well
# with the theoretical result. The previous implementation also agrees well.
## Not run:
#Try old code
install.packages("http://familias.name/famr_1.0-zip")
library(famr)
res3 <- conditionalSimulationWrite(nsim = Nsim, datamatrix, persons,
    loci, pedigrees, available = 3, seed = 1482659436, ref = 2, file = NULL)
LR.H1.old = table(res3[,1])/Nsim
LR.H2.old = table(res3[,2])/Nsim
# The above code is limited in some respects: It assumes
# that there are two pedigrees and that there
# are no mutations. With ref=2, H2 is the denominator of the LR. Column 1 of the output,
# res3[,1] above is simulated assumed H1 to be true, res[,2] assuming H2 to be true.

## End(Not run)

# Consider next simulation under H2 calculated for H2
LR.H2.Familias = table(res1[,2])/Nsim

```

```

LR.H2.paramlink = table(res2[,2])/Nsim
stopifnot(round(LR.H2.Familias-LR.H2.paramlink,12)==0)

# Obviously, the possible values for LR are the same for H1 and H2.
# Note that the largest value of LR occurs with probability  $py3=p1^2= 1e-04$ 
# This value may not be reached in the simulations. Again results agree well and
# also with the previous implementation if Nsim=1000. The probability distribution of LR
# conditionally on H2 is calculated as

py3 = p1^2
py1 = (1-p1)^2
py2 = 2*p1*(1-p1)
LR.H2.exact = c(py1,py2,py3)
names(LR.H2.exact) = paste(LRs)

# The previous example continues, but we will now
# model mutations and for simplicity assume a SNP marker.
## Not run:
# Let the mutation rate be 0.05
# The mutation rate is chosen (too) high to see some impact.
# in a 'proportional mutation model, i.e.,
p = c(0.2, 0.8); R=0.05
loci = list(FamiliasLocus(p, 1:2, "L1", MutationModel = "Proportional",
                        MutationRate=R))
x = Familias2linkdat(pedigrees[[1]], datamatrix, loci)
m = marker(x, 1, c(1,1), alleles=1:2, afreq=p)
x = addMarker(x,m)
p.GS = oneMarkerDistribution(x, 3, partialmarker=1, verbose = FALSE)
# For instance
p22.one = p.GS["2/2"]
# is the probability of the grandson being 2/2 when
# the grandmother is 1/1; without mutation this would be
p22.ind = 0.5*p[2]^2
# We next check the exact result by an exact formula
# Egeland, Pinto and Amorim (2017, submitted) and also
# using simulation. Let
H = 1-sum(p^2); k = R/H
p22 = 0.5*p[2]^2*(2-(1-k)^2) # LR = 0.5+0.5*(1-(1-k)^2) with
stopifnot(round(p22.one-p22, 10)==0)
# probability p22. Below we simulate to check
Nsim = 1000; seed=177
res = conditionalLR(Nsim=Nsim, datamatrix, loci, pedigrees, available="GS",
                    seed=seed, program = "Familias", verbose = FALSE, simplify=TRUE)
LR.H1.mut = table(res[,1])/Nsim
(LR.H1.mut[1]-p22)/p22 #relative difference

## End(Not run)

# Example Missing grandchild example
## Not run: #Takes 3-4 minutes
data(F21)
pedigrees = F21$pedigrees
datamatrix = F21$datamatrix

```

```

loci = F21$loci
persons = rownames(datamatrix)
x = Familias2linkdat(pedigrees, datamatrix, loci)
Nsim = 1000
res1 = conditionalLR(Nsim = Nsim, datamatrix, loci, pedigrees, program = "Familias",
                    available = "Missing Person", seed=17, verbose = FALSE, simplify=TRUE)
LR = data.frame(LR.H1=res1[,1], LR.H2=res1[,2])
length(LR[,1][LR[,1]>10^5])/Nsim
length(LR[,2][LR[,2]==0])/Nsim #PE estimate
res1 = FamiliasConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                          available = "Missing Person", seed=17)

res2 = conditionalLR(Nsim = Nsim, datamatrix, loci, pedigrees, program = "Familias",
                    available = "Missing Person", seed=17, verbose = FALSE)
res = cbind(res1[[1]][,1], res2[,1][,1])

boxplot(log(res)); title ("log LR(H1), Familias and paramlink (right)")

## End(Not run)

## Not run:
# Example
data(Demo3Markers)
pedigrees = Demo3Markers$pedigrees
datamatrix = Demo3Markers$datamatrix
loci = Demo3Markers$loci
persons = rownames(datamatrix)
Nsim = 5
res.Familias = conditionalLR(Nsim = Nsim, datamatrix, loci, pedigrees,
                            file = NULL, program = "Familias", truePed = NULL,
                            available = "Mother", ref=NULL, seed=177, simplify = TRUE)
res.paramlink = conditionalLR(Nsim = Nsim, datamatrix, loci, pedigrees,
                              file = NULL, program = "paramlink", truePed = NULL,
                              available = "Mother", ref=NULL, seed=177, simplify=TRUE)

# Always the same LR as the probabilities of the genotypes of the person simulated ("Mother")
# is the same for both hypotheses:
stopifnot(round(res.Familias[,1]-res.paramlink[1,1],12)==0)
stopifnot(round(res.Familias[,2]-res.paramlink[1,2],12)==0)

# Example
data(symmetric)
pedigrees = symmetric$pedigrees
datamatrix = symmetric$datamatrix
loci = symmetric$loci
persons = rownames(datamatrix)
truePeds = 1:3
res.Familias = conditionalLR(Nsim = 5, datamatrix, loci, pedigrees,
                            file = NULL, program = "Familias", truePeds = truePeds,
                            available = NULL, ref=2, seed=177, verbose = FALSE)
res.paramlink = conditionalLR(Nsim = 5, datamatrix, loci, pedigrees,
                              file = NULL, program = "paramlink", truePeds = truePeds, #Change
                              available = NULL, ref=2, seed=177, verbose = FALSE)

```

```

stopifnot(round(res.Familias[, , truePeds]-res.paramlink[, , truePeds],12)==0)

# Example
ped = list(singleton(5,1), nuclearPed(2))
x = markerSim(ped, N=5, alleles=1:5, verbose=FALSE, available=5)
y = nuclearPed(3)
y2 = transferMarkerdata(x, y)
y2

## End(Not run)

```

FamiliasConditional *Conditional simulation of marker data on pedigrees and 'Familias' likelihood*

Description

Marker data is simulated for a specified person and several markers using `FamiliasConditionalOne` for each marker. The only difference between `FamiliasConditional` and `paramlinkConditional` is that the former uses the C implementation of `FamiliasPosterior` for likelihood calculation while the latter is based on the R implementation in `paramlink`.

Usage

```

FamiliasConditional(Nsim = 5, datamatrix, loci, pedigrees, truePed = 1, available = NULL,
                    ref = 2, prior = NULL, seed = NULL)
FamiliasConditionalOne(Nsim = 5, mark = 1, ref = 2, datamatrix,
                       persons, loci, pedigrees, truePed = 1, available = NULL,
                       prior = NULL, seed = NULL)

```

Arguments

<code>Nsim</code>	Integer. Number of simulations.
<code>ref</code>	Integer Denominator of LR.
<code>datamatrix</code>	A data frame or a matrix. The row names must be the names of the persons you have data for. The columns contain the alleles, two columns for each marker, in the same order used in the loci list.
<code>loci</code>	A list of <code>FamiliasLocus</code> objects.
<code>pedigrees</code>	An list with elements of type 'FamiliasPedigree'.
<code>truePed</code>	Integer. Index of pedigree from which marker data are simulated.
<code>available</code>	A character giving the name of the person to be simulated or the integer ID.
<code>prior</code>	Double vector. The prior on pedigrees.
<code>seed</code>	Integer used to fix simulations.
<code>mark</code>	Integer. Index of <code>\[link[paramlink]{marker}</code> used in <code>FamiliasConditionalOne</code>
<code>persons</code>	Character vector. Names of persons.

Details

Marker data is simulated for a specified person and one specified marker using `FamiasConditional` which calls `markerSim` of `paramlink`. The marker data is then loaded into a datamatrix and likelihoods calculated using `Famias`.

Value

`LR.All.Markers` One LR for each simulation for each pedigree.

`lik.All.Markers` One likelihood for each simulation for each pedigree.

`LR.Per.Marker` One LR for each simulation and marker for each pedigree.

`lik.Per.Marker` One likelihood for each simulation and marker for each pedigree.

`first.Sim` Marker data for first marker.

Author(s)

Thore Egeland <Thore.Egeland@gmail.com> and Magnus Dehli Vigeland

References

Kling et al. (2017)

See Also

See also `markerSim`

Examples

```
# Example. Mutation.
# See documentation of conditionalLR for examples without mutation
data(grandmother)
pedigrees = grandmother$pedigrees
datamatrix = grandmother$datamatrix
persons = rownames(datamatrix)
Nsim = 100
p = c(0.5,0.5)
loci = list(FamiasLocus(p, 1:2, "L1", MutationModel = "Proportional",
                      MutationRate=0.005))
res1 = FamiasConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                        truePed = 1, available = "GS", ref=2, seed=17)
LR1 = table(res1$LR.All.Markers[,1])/Nsim
# Next, we calculate genotype probabilities for GS exactly
x = Famias2linkdat(pedigrees[[1]], datamatrix, loci)
m = marker(x, 1:2)
x = addMarker(x,m)
p.GS = oneMarkerDistribution(x, 3, partialmarker=1, verbose = FALSE)
# The difference between simulated and exact probabilities are
LR1 - p.GS[c(2,3,1)]
```

```

### Example
data(Demo3Markers)
pedigrees = Demo3Markers$pedigrees
datamatrix = Demo3Markers$datamatrix
loci = Demo3Markers$loci
x = Familias2linkdat(pedigrees, datamatrix, loci)
P.mother.H1 = oneMarkerDistribution(x[[1]], ids=3, partialmarker=3,
  ignore.affection=TRUE)
P.mother.H2 = oneMarkerDistribution(x[[2]][[1]], ids=3,
  partialmarker=3, ignore.affection=TRUE)
all(round(P.mother.H1/P.mother.H2,10) == 1)
# The probability distribution of the mother is the same for both hypotheses
# and therefore we always get the same LR.

# Example
data(symmetric)
pedigrees = symmetric$pedigrees
datamatrix = symmetric$datamatrix
loci = symmetric$loci
persons = rownames(datamatrix)
## Not run:
# Plot with newdev=TRUE, resize plot window and then plot with newdev=FALSE
x = Familias2linkdat(pedigrees, datamatrix, loci)
plotPedList(x, newdev=FALSE, frametitles=c("H1", "H2", "H3"),
  available="shaded", marker=1:2, skip.empty.genotypes = TRUE)

## End(Not run)
res1 = FamiliasConditional(Nsim = 2, datamatrix, loci, pedigrees,
  truePed = 1, available = "MO", ref = 2, seed = 17)
# Without mutation, all LRs 1. With mutation:
res1$LR.All.Markers

```

grandmother

Dataset for missing grandchildren paper

Description

The datasets are used for examples in Kling et al. (2017) or to validate conditional simulation of marker data followed by likelihood calculation for alternative pedigrees. Further brief description is given for the examples below, see [FamiliasConditional](#) or [paramlinkConditional](#) for examples involving computations.

Usage

```

data(grandmother)
data(Demo3Markers)
data(symmetric)
data(F21)
data(dc)
data(adoption)

```

Format

A list with components pedigrees, datamatrix, loci, see [FamiliasLocus](#)

Examples

```

data(grandmother) #One marker
data(Demo3Markers) #Three markers, one with mutation
#Three pedigrees, Half-sibs, avuncular and grandparent:
data(symmetric)
data(E004) # E zero zero four
pedigrees = E004$pedigrees
datamatrix = E004$datamatrix
loci = E004$loci
avail = 7
persons = rownames(datamatrix)
x = Familias2linkdat(pedigrees, datamatrix, loci)
missing.person.plot(x[[2]][[1]], missing=avail, marker=1:2, newdev=TRUE,
                    frametitle=c("H1", "H2"), dev.height=3.3, dev.width=9)
x=Familias2linkdat(pedigrees, datamatrix, loci)
missing.person.plot(x[[2]][[1]], missing=avail, marker=1:2, newdev=TRUE,
                    frametitle=c("H1", "H2"))
## Not run: #Takes a few minutes
Nsim=1000
res.new = conditionalLR(Nsim=Nsim, datamatrix, loci, pedigrees, available=avail,
                       seed=1483118516, program = "Familias", ref=1 )
LRnew.mean =cbind(res.new[,1][,2],res.new[,2][,2])
apply(LRnew.mean,2,quantile)

## End(Not run)

```

LRparamlink

Calculates likelihoods and likelihood ratios using 'paramlink'

Description

Typically export from windows 'Familias' is converted to [linkdat](#) using [Familias2linkdat](#) prior to calling the function.

Usage

```
LRparamlink(x, ref, markers)
```

Arguments

x	A list of pedigrees, in the form of linkdat objects.
ref	Integer. Index of the pedigree to be used in the denominator of LR.
markers	Integer vector. Index for markers for which calculations should be done.

Value

LR	Likelihood ratios.
LRperMarker	Likelihood ratios for each marker.
likelihoodsPerSystem	Likelihoods for each marker.
time	User, system and elapsed time.

Author(s)

Magnus Dehli Vigeland and Thore Egeland

Examples

```
data(adoption)
x = Familias2linkdat(adoption$pedigrees, adoption$datamatrix, adoption$loci)
result = LRparamlink(x, ref=2)
# Only marker 11 and 33
result33 = LRparamlink(x, ref=2, marker=c(11,33))
```

missing.person.plot *Plots hypotheses for family reunion*

Description

A panel with two plots are generated. To the left, the one corresponding to the person of interest (POI) being identical to the Missing Person (MP). To the right these persons are unrelated.

Usage

```
missing.person.plot(ped_related, missing, id.labels=NULL,
  available="shaded", marker=NULL, width=c(4,4,1), newdev=TRUE,
  frametitles=c("H1: POI related", "H2:POI unrelated"), ...)

internalID(x, orig.ids)
getSex(x, orig.ids)
```

Arguments

ped_related	A <code>linkdat</code> object.
missing	An integer giving the MP.
id.labels	Character vector naming individuals in plot.
available	Character determining how genotyped individuals are displayed.
marker	Integer vector indicating markers to be displayed.
width	A numeric vector of relative widths of the subplots. Recycled to <code>length(plot.arg.list)</code> if necessary, before passed on to layout. Note that the vector does not need to sum to 1.

newdev	logical. If TRUE, new plot window is generated.
frametitles	A character vector of length two giving titles.
...	Further arguments passed on to each call to plotPedList
x	linkdat object.
orig.ids	Character or integer vector,

Details

See example in the Examples section below.

Author(s)

Magnus Dehli Vigeland and Thore Egeland

See Also

[plotPedList](#)

Examples

```
data(grandmother)
pedigrees = grandmother$pedigrees
datamatrix = grandmother$datamatrix
datamatrix[3,]=1
loci = grandmother$loci
x1 = Familias2linkdat(pedigrees, datamatrix, loci)
missing.person.plot(x1[[1]],2, available="shaded", frametitles=c("", ""),
  newdev=TRUE, width=c(1,2), marker=1)
```

paramlinkConditional *Conditional simulation of marker data on pedigrees and [paramlink](#) likelihood*

Description

Marker data is simulated for a specified person and several markers using [paramlinkConditionalOne](#) for each marker. The only difference between [FamiliasConditional](#) and [paramlinkConditional](#) is that the former uses the C implementation of [FamiliasPosterior](#) for likelihood calculation while the latter is based on the R implementation in [paramlink](#).

Usage

```
paramlinkConditional(Nsim = 5, datamatrix, loci, pedigrees,
  truePed = 1, ref=NULL, available = NULL, prior=NULL, seed = NULL)
paramlinkConditionalOne(Nsim = 5, mark = 1, ref = 2, datamatrix, loci,
  pedigrees, truePed = 1, available = NULL, prior = NULL, seed = NULL)
```

Arguments

Nsim	Integer. Number of simulations.
ref	Integer Denominator of LR.
datamatrix	A data frame or a matrix. The row names must be the names of the persons you have data for. The columns contain the alleles, two columns for each marker, in the same order used in the loci list.
loci	A list of FamiliasLocus objects.
pedigrees	An list with elements of type FamiliasPedigree .
truePed	Integer. Index of pedigree from which marker data are simulated.
available	A character giving the name of the person to be simulated or the integer ID.
prior	Double vector. The prior on pedigrees.
seed	Integer used to fix simulations.
mark	Integer. Index of marker used in <code>paramlinkConditionalOne</code> .

Details

Marker data is simulated for a specified person and one specified marker using `paramlinkConditional` which calls `markerSim`. The marker data is then loaded into a `datamatrix` and likelihoods calculated using 'Familias'.

Value

LR.All.Markers	One LR for each simulation for each pedigree.
lik.All.Markers	One likelihood for each simulation for each pedigree.
LR.Per.Marker	One LR for each simulation and marker for each pedigree.
lik.Per.Marker	One likelihood for each simulation and marker for each pedigree.
first.Sim	NULL

Author(s)

Thore Egeland <Thore.Egeland@gmail.com> and Magnus Dehli Vigeland

References

Kling et al. (2017)

See Also

See also [FamiliasConditional](#)

Examples

```

# Example
data(grandmother)
pedigrees = grandmother$pedigrees
datamatrix = grandmother$datamatrix
loci = grandmother$loci
persons = rownames(datamatrix)
Nsim = 5 #Increase to 1000
res1 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                           truePed = 1, available = "GS", ref=2, seed=17)
LR.H1 = table(res1$LR.All.Markers[,1])/Nsim
# LR(H1) distribution, agrees well with theory:
# Pr(LR=0.5|H1) = 0.49005, Pr(LR=25.5|H1) = 0.50490, Pr(LR=50.5|H1) = 0.00505
# Simulate from unrelated alternative:
res2 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                           truePed = 2, available = "GS", ref=2, seed=17)
LR.H2 = table(res2$LR.All.Markers[,1])/Nsim
# Try mutation
p = as.double(loci[[1]]$alleles)
loci = list(FamiliasLocus(p, 1:3, "L1", MutationModel = "Proportional",
                        MutationRate=0.005))
x = Familias2linkdat(pedigrees, datamatrix, loci)
res1 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                           truePed = 1, available = "GS", ref=2, seed=17)
# Simulate father instead
res1 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                           truePed = 1, available = "FAT", ref=2, seed=17)
LR.H1 = table(res1$LR.All.Markers[,1])/Nsim

res2 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                           truePed = 2, available = "FAT", ref=2, seed=17)
LR.H2 = table(res2$LR.All.Markers[,1])/Nsim

### Example
data(Demo3Markers)
pedigrees = Demo3Markers$pedigrees
datamatrix = Demo3Markers$datamatrix
loci = Demo3Markers$loci
persons = rownames(datamatrix)
Nsim = 5
res1 = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                           truePed = 1, available = "Mother", ref=2, seed=177)
res1[[1]][,1] #Always the same LR! Why?:
x = Familias2linkdat(pedigrees, datamatrix, loci)
P.mother.H1 = oneMarkerDistribution(x[[1]],3,3, verbose=FALSE)
P.mother.H2 = oneMarkerDistribution(x[[2]][[1]],3,3, verbose=FALSE)
round(P.mother.H1/P.mother.H2,10) == 1
# The probability distribution of the mother is the same for both hypotheses
# and therefore we always get the same LR.

# Example

```

```

data(symmetric)
pedigrees = symmetric$pedigrees
datamatrix = symmetric$datamatrix
loci = symmetric$loci
persons = rownames(datamatrix)
res1 = paramlinkConditional(Nsim = 2, datamatrix, loci, pedigrees,
                           truePed = 1, available = NULL, ref=2, seed=17)
# Without mutation, all LRs 1. With mutation:
res1$LR.All.Markers

# Example Mariana's F21 example
## Not run: #Takes a few minutes; compares 'Familias' and 'paramlink'.
#Results and paramlink execution times
data(F21)
pedigrees = F21$pedigrees
datamatrix = F21$datamatrix
loci = F21$loci
persons = rownames(datamatrix)
Nsim = 1000
start.time <- Sys.time()
res1.paramlink = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                                     truePed = 1, available = "Missing Person", ref=2, seed=17)
res2.paramlink = paramlinkConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                                     truePed = 2, available = "Missing Person", ref=2, seed=17)
end.time <- Sys.time()
paramlink.time <- end.time - start.time
start.time <- Sys.time()
res1.familias = FamiliasConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                                   truePed = 1, available = "Missing Person", ref=2, seed=17)
res2.familias = FamiliasConditional(Nsim = Nsim, datamatrix, loci, pedigrees,
                                   truePed = 2, available = "Missing Person", ref=2, seed=17)
end.time <- Sys.time()
familias.time <- end.time - start.time
familias.time - paramlink.time #around -0.7 mins
LR1.familias = res1.familias[[1]][,1]
LR1.paramlink = res1.paramlink[[1]][,1]
aa=cbind(LR1.familias, LR1.paramlink)
foo=apply(aa,2, quantile)
foo = LR1.familias - LR1.paramlink
max(foo/(0.5*(LR1.familias+LR1.paramlink)))
LR1.familias = res2.familias[[1]][,1]
LR1.paramlink = res2.paramlink[[1]][,1]
aa=cbind(LR1.familias, LR1.paramlink)
foo=apply(aa,2, quantile)
foo = LR1.familias - LR1.paramlink
max(foo)

## End(Not run)

```


Description

This is a wrapper for [exclusionPower](#). The probability of exclusion (PE) is calculated for each marker and combined for all markers.

Usage

```
PE(pedigrees, datamatrix, loci, claim = 1, true = 2, available = NULL,  
   file = NULL, ignore = FALSE)
```

Arguments

pedigrees	A FamiliasPedigree object or a list of such.
datamatrix	A data frame, each row gives the genotype of one individual, a pair of columns represents a marker.
loci	A FamiliasLocus object or a list of such.
claim	Integer. Index of the claimed pedigree, typically the one where MP=POI.
true	Integer. Index of the assumed true pedigree for PE calculation, typically the one where POI is assumed to be an unrelated individual.
available	Character or integer identifying person to be simulated.
file	Character. If not NULL, a file is written.
ignore	Logical. If TRUE, mutations are ignored.

Details

Wrapper for [exclusionPower](#)

Value

A data frame. The first column gives the name of the markers, the second the PE. The last line gives the combined result for all markers.

Author(s)

Thore Egeland <Thore.Egeland@gmail.com>

References

Egeland, Pinto and Vigeland (2014).

See Also

[exclusionPower](#)

Examples

```
data(F21)
pedigrees = F21$pedigrees
datamatrix = F21$datamatrix
loci = F21$loci
available = "Missing Person"
PE(pedigrees, datamatrix, loci, claim = 1, true = 2,
  available = available, file = NULL)
```

Index

adoption (grandmother), [10](#)

conditionalLR, [2, 3](#)

dc (grandmother), [10](#)

Demo3Markers (grandmother), [10](#)

E004 (grandmother), [10](#)

exclusionPower, [17](#)

F21 (grandmother), [10](#)

fam2r (fam2r-package), [2](#)

fam2r-package, [2](#)

Familias, [2](#)

Familias2linkdat, [11](#)

FamiliasConditional, [3, 8, 10, 13, 14](#)

FamiliasConditionalOne
(FamiliasConditional), [8](#)

FamiliasLocus, [8, 11, 14, 17](#)

FamiliasPedigree, [14, 17](#)

FamiliasPosterior, [8, 13](#)

getSex (missing.person.plot), [12](#)

grandmother, [10](#)

internalID (missing.person.plot), [12](#)

label2num (conditionalLR), [3](#)

linkdat, [3, 11–13](#)

LRparamlink, [11](#)

LRwrap (conditionalLR), [3](#)

marker, [14](#)

markerSim, [2, 9, 14](#)

missing.person.plot, [12](#)

paramlink, [2, 8, 9, 13](#)

paramlinkConditional, [3, 8, 10, 13](#)

paramlinkConditionalOne
(paramlinkConditional), [13](#)

PE, [16](#)

plotPedList, [13](#)

removePersons (conditionalLR), [3](#)

symmetric (grandmother), [10](#)

transferMarkerdata, [3](#)