

Package ‘fbnet’

July 1, 2023

Title Forensic Bayesian Networks

Version 1.0.3

Description

Open-source package for computing likelihood ratios in kinship testing and human identification cases (Chernomoretz et al. (2021) <[doi:10.1016/j.fsir.2020.100132](https://doi.org/10.1016/j.fsir.2020.100132)>). It has the core function of the software GENis, developed by Fundación Sadosky. It relies on a Bayesian Networks framework and is particularly well suited to efficiently perform large-size queries against databases of missing individuals (Darwiche (2009) <[doi:10.1017/CBO9780511811357](https://doi.org/10.1017/CBO9780511811357)>).

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL <https://github.com/MarsicoFL/fbnet>

Imports graphics, grDevices, igraph, stats, Rsolnp, assertthat, utils, paramlink

NeedsCompilation no

Author Franco Marsico [aut, cre],
Ariel Chernomoretz [aut]

Maintainer Franco Marsico <franco.lmarsico@gmail.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-07-01 18:20:02 UTC

R topics documented:

Argentina_STRs	2
buildBN	3
buildCPTs	3
convertPedformat	4
evidencePruning	5

factorHeteroFounders	5
FamiliasLocus	6
FamiliasPedigree	7
fbnet	8
getConditional	8
getGenotypeTables	9
getLocusCPT	9
getMAP	10
getQSetRMP	10
getValuesOut	11
imposeEvidence	11
initBN	12
initBN.fromPed	12
initBN.fromVars	13
minOrdering	13
preparePed	14
prodFactor	14
pruneNodes	15
removeEvidenceFromPed	15
reportLR	16
reportPQ	16
reverseSplit	17
setOrdering	17
stateRemoval	18
stateRemoval2	18
stateRemovalSubnucs	19
sumFactor	19
toybase	20
toyped	20
velim.bn	20

Index	22
--------------	-----------

Argentina_STRs	<i>STRs allelic frequencies from Argentina.</i>
----------------	---

Description

STRs allelic frequencies from Argentina.

Usage

Argentina_STRs

Format

A data frame with allele frequencies

buildBN	<i>buildBN: a function for building the bayesian network.</i>
---------	---

Description

buildBN: a function for building the bayesian network.

Usage

```
buildBN(pbn, QP)
```

Arguments

pbn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
QP	Query Persons Ids

Value

A bayesian network based on pedigree evidence and QP definition.

Examples

```
pbn <- initBN(toyped)
bnet <- buildBN(pbn, QP=3)
```

buildCPTs	<i>buildCPTs: a function for building conditional probability tables based on pedigree bayesian network.</i>
-----------	--

Description

buildCPTs: a function for building conditional probability tables based on pedigree bayesian network.

Usage

```
buildCPTs(
  bn,
  bNodePrunning = TRUE,
  bStateRemoval = TRUE,
  bStateRemoval2 = TRUE,
  lumpingParameter = NULL,
  renorm = "row-wise",
  verbose = FALSE
)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
bNodePruning	Standard pruning.
bStateRemoval	State based pruning.
bStateRemoval2	State based pruning (model 2).
lumpingParameter	Used for stepwise mutational model.
renorm	If "row-wise" is selected, zero probability is assigned for transitions out of range.
verbose	Computations output.

Value

A bayesian network based on pedigree evidence and QP definition.

Examples

```
pbn <- initBN(toyped)
bnet <- buildBN(pbn, QP=3)
bn1 <- buildCPTs(bnet)
```

convertPedformat	<i>convertPedformat: a function for converting a pedtools ped object to a famlink ped object.</i>
------------------	---

Description

convertPedformat: a function for converting a pedtools ped object to a famlink ped object.

Usage

```
convertPedformat(x, verbose = FALSE)
```

Arguments

x	A pedtools ped object.
verbose	Function output.

Value

A dataframe with LRs.

evidencePruning *evidencePruning: a fuction for pruning instantiated variables.*

Description

evidencePruning: a fuction for pruning instantiated variables.

Usage

```
evidencePruning(bn)
```

Arguments

bn A bayesian network (output of buildBN function).

Value

A preprocessed bayesian network.

factorHeteroFounders *factorHeteroFounders: a function for multiplying probabilities in case of heterocigote founders.*

Description

factorHeteroFounders: a function for multiplying probabilities in case of heterocigote founders.

Usage

```
factorHeteroFounders(rresQ, bn)
```

Arguments

rresQ List of CPTs.

bn A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.

Value

A dataframe with genotype probabilities.

 FamiliasLocus

FamiliasLocus: a function for reading Familias locus data.

Description

FamiliasLocus: a function for reading Familias locus data.

Usage

```
FamiliasLocus(
  frequencies,
  allelenames,
  name,
  MutationModel = "Stepwise",
  MutationRate = 0,
  MutationRange = 0.5,
  MutationRate2 = 0,
  MutationMatrix,
  Stabilization = "None",
  MaxStabilizedMutrate = 1,
  femaleMutationModel,
  femaleMutationRate,
  femaleMutationRange,
  femaleMutationRate2,
  femaleMutationMatrix,
  maleMutationModel,
  maleMutationRate,
  maleMutationRange,
  maleMutationRate2,
  maleMutationMatrix
)
```

Arguments

frequencies	allele frequencies
allelenames	names
name	system name
MutationModel	model
MutationRate	rates
MutationRange	range
MutationRate2	rate two, applied for extended stepwise
MutationMatrix	matrix
Stabilization	stabilization factor
MaxStabilizedMutrate	mix factor

```

femaleMutationModel
    for females
femaleMutationRate
    rate
femaleMutationRange
    range
femaleMutationRate2
    rate 2
femaleMutationMatrix
    females matrix
maleMutationModel
    male matrix
maleMutationRate
    male rate
maleMutationRange
    male range
maleMutationRate2
    rate 2
maleMutationMatrix
    matrix

```

Value

Locus analysis.

Examples

```

frequencies <- c(0.1, 0.2, 0.3, 0.4)
allelenames <- c("A", "B", "C", "D")
marker <- FamiliasLocus(frequencies, allelenames)

```

FamiliasPedigree	<i>FamiliasPedigree: a function for constructing Familias pedigree format.</i>
------------------	--

Description

FamiliasPedigree: a function for constructing Familias pedigree format.

Usage

```
FamiliasPedigree(id, dadid, momid, sex)
```

Arguments

id	individual id
dadid	father id
momid	mother id
sex	biological sex

Value

A dataframe with probabilities.

Examples

```
persons <- c("mother", "child", "AF")
sex <- c("female", "female", "male")
ped1 <- FamiliasPedigree(id = persons, dadid = c(NA, "AF", NA), momid = c(NA, "mother", NA), sex=sex)
```

 fbnet

fbnet: Forensic Bayesian Networks

Description

'fbnet' is an open source software package written in R statistical language. It relies on a Bayesian Networks framework doi: [10.1017/CBO9780511811357](https://doi.org/10.1017/CBO9780511811357). It is particularly well suited to efficiently perform large-size queries against databases of missing individuals. It could interact with the main functionalities of other packages for pedigree analysis. In particular, 'fbnet' imports the 'Familias' software doi: [10.1016/S03790738\(00\)00147X](https://doi.org/10.1016/S03790738(00)00147X). In addition 'pedtools', a software for creating and manipulating pedigrees and markers, is supported. 'fbnet' allows computing LR's and obtaining genotype probability distributions for query individual, based on the pedigree data. 'fbnet' implements the complete GENis functionality, a recently published open-source multi-tier information system developed to run forensic DNA databases to perform kinship analysis based on DNA profiles doi: [10.1016/j.fsir.2020.100132](https://doi.org/10.1016/j.fsir.2020.100132).

 getConditional

getConditional: a function for obtaining the conditional probability tables based on a given evidence.

Description

getConditional: a function for obtaining the conditional probability tables based on a given evidence.

Usage

```
getConditional(lf)
```

Arguments

lf A list of joint probabilities.

Value

A list of conditioned probabilities.

getGenotypeTables	<i>getGenotypeTables: a function for obtaining genotypetables after variable elimination and using available genetic evidence.</i>
-------------------	--

Description

getGenotypeTables: a function for obtaining genotypetables after variable elimination and using available genetic evidence.

Usage

```
getGenotypeTables(bn, resQ, geno = NULL, lqp = NULL)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
resQ	List of CPTs.
geno	data.frame with genotypes.
lqp	list of individuals genotypes.

Value

A dataframe with genotype probabilities.

getLocusCPT	<i>getLocusCPT: a function for obtaining the conditional probability table from a specific locus.</i>
-------------	---

Description

getLocusCPT: a function for obtaining the conditional probability table from a specific locus.

Usage

```
getLocusCPT(bn, locus, lumpingParameter = NULL, renorm = "row-wise")
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
locus	Specified locus.
lumpingParameter	Used for stepwise mutational model.
renorm	If "row-wise" is selected, zero probability is assigned for transitions out of range.

Value

A bayesian network based on pedigree evidence and QP definition.

Examples

```
pbn <- initBN(toyped)
bnet <- buildBN(pbn, QP=3)
bn1 <- buildCPTs(bnet)
locCPT <- getLocusCPT(bn1, "M1")
```

getMAP	<i>factorHeteroFounders: a function for multiplying probabilities in case of heterocigote founders.</i>
--------	---

Description

factorHeteroFounders: a function for multiplying probabilities in case of heterocigote founders.

Usage

```
getMAP(resQ, topn = 3)
```

Arguments

resQ	List of CPTs.
topn	Format parameter.

Value

A MAP from the probability table.

getQSetRMP	<i>getGenotypeTables: a function for obtaining genotypetables after variable elimination and using available genetic evidence.</i>
------------	--

Description

getGenotypeTables: a function for obtaining genotypetables after variable elimination and using available genetic evidence.

Usage

```
getQSetRMP(bn, lqp)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
lqp	list of individuals genotypes.

Value

A dataframe with genotype probabilities.

getValuesOut	<i>getValuesOut: a function for getting out variables with zero probability in the bayesian network</i>
--------------	---

Description

getValuesOut: a function for getting out variables with zero probability in the bayesian network

Usage

```
getValuesOut(cpt, condVar = c())
```

Arguments

cpt	conditional probability table from the bayesian network
condVar	variables from the conditioning table

Value

A processed conditional probability table

imposeEvidence	<i>imposeEvidence: a fuction for imposing evidence in the bayesian network.</i>
----------------	---

Description

imposeEvidence: a fuction for imposing evidence in the bayesian network.

Usage

```
imposeEvidence(bn)
```

Arguments

bn	A bayesian network (output of buildBN function).
----	--

Value

A preprocessed bayesian network.

<code>initBN</code>	<i>initBN: a function to initialize the bayesian network.</i>
---------------------	---

Description

`initBN`: a function to initialize the bayesian network.

Usage

```
initBN(ped = NULL, bplotped = FALSE)
```

Arguments

<code>ped</code>	A ped object with information of the genotyped members. The ped object must be in Familias format.
<code>bplotped</code>	An alternative ped object to be compared.

Value

A bayesian network.

Examples

```
pbn <- initBN(toyped)
```

<code>initBN.fromPed</code>	<i>initBN.fromPed: a function to initialize the bayesian network.</i>
-----------------------------	---

Description

`initBN.fromPed`: a function to initialize the bayesian network.

Usage

```
initBN.fromPed(ped, bplotped)
```

Arguments

<code>ped</code>	A ped object in Familias format.
<code>bplotped</code>	An alternative ped object to be compared.

Value

A bayesian network.

initBN.fromVars	<i>initBN.fromVars: a function to initialize the bayesian network.</i>
-----------------	--

Description

initBN.fromVars: a function to initialize the bayesian network.

Usage

```
initBN.fromVars(bplotped)
```

Arguments

bplotped An alternative ped object to be compared.

Value

A bayesian network.

minOrdering	<i>minOrdering: a function for getting an ordering of bayesian network variables not in Q using min fill criteria on interaction graphs.</i>
-------------	--

Description

minOrdering: a function for getting an ordering of bayesian network variables not in Q using min fill criteria on interaction graphs.

Usage

```
minOrdering(bn, vars = NULL, method = c("min_degree", "min_fill")[1])
```

Arguments

bn A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.

vars Subset of tables where the order is calculated

method Elimination method, min_degree or min_fill

Value

A bayesian network after ordering process.

preparePed	<i>preparePed: a function for simulating genetic data from untyped individuals conditioned on known genotypes.</i>
------------	--

Description

preparePed: a function for simulating genetic data from untyped individuals conditioned on known genotypes.

Usage

```
preparePed(ped, available, lLociFreq, rseed = NULL)
```

Arguments

ped	A ped object with information of the genotyped members. The ped object must be in Familias format.
available	Genotyped individuals IDs.
lLociFreq	Allele frequencies.
rseed	Seed used for simulations.

Value

A ped object.

prodFactor	<i>prodFactor: a function for performing product between probability tables.</i>
------------	--

Description

prodFactor: a function for performing product between probability tables.

Usage

```
prodFactor(laux)
```

Arguments

laux	probability distribution aux
------	------------------------------

Value

A dataframe with probabilities.

pruneNodes	<i>pruneNodes: a fuction for clasical pruning in bayesian networks.</i>
------------	---

Description

pruneNodes: a fuction for clasical pruning in bayesian networks.

Usage

```
pruneNodes(bn)
```

Arguments

bn	A bayesian network (output of buildBN function).
----	--

Value

A preprocessed bayesian network.

removeEvidenceFromPed	<i>removeEvidenceFromPed: a function for removing evidence from specific individuals in a ped object.</i>
-----------------------	---

Description

removeEvidenceFromPed: a function for removing evidence from specific individuals in a ped object.

Usage

```
removeEvidenceFromPed(pped, idNotEv)
```

Arguments

pped	A ped object with information of the genotyped members. The ped object must be in Familias format.
idNotEv	A set of individuals whom evidence should be removed.

Value

A ped object.

reportLR	<i>reportLR: a function for calculating the LRs of specified genotypes in a pedigree.</i>
----------	---

Description

reportLR: a function for calculating the LRs of specified genotypes in a pedigree.

Usage

```
reportLR(bn, resQ, geno = NULL)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
resQ	List of CPTs.
geno	data.frame with genotypes.

Value

A dataframe with LRs.

reportPQ	<i>reportPQ: a function for calculating the probability of specified genotypes in a pedigree.</i>
----------	---

Description

reportPQ: a function for calculating the probability of specified genotypes in a pedigree.

Usage

```
reportPQ(bn, resQ, geno = NULL)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
resQ	List of CPTs.
geno	data.frame with genotypes.

Value

A dataframe with genotype probabilities.

reverseSplit	<i>reverseSpit: a function for formatting.</i>
--------------	--

Description

reverseSpit: a function for formatting.

Usage

```
reverseSplit(inList)
```

Arguments

inList	input for formatting.
--------	-----------------------

Value

A bayesian network.

setOrdering	<i>setOrdering: a function for selecting the ordering method in the elimination process.</i>
-------------	--

Description

setOrdering: a function for selecting the ordering method in the elimination process.

Usage

```
setOrdering(bn, ordMethod, vars = NULL, orderElim = NULL)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
ordMethod	Ordering method.
vars	Vars
orderElim	Order elimination criteria.

Value

A bayesian network after ordering process.

stateRemoval	<i>stateRemoval: a function for processing the bayesian network.</i>
--------------	--

Description

stateRemoval: a function for processing the bayesian network.

Usage

```
stateRemoval(bn)
```

Arguments

bn A bayesian network (output of buildBN function).

Value

A preprocessed bayesian network.

stateRemoval2	<i>stateRemoval2: a function for processing the bayesian network. It implements another approach from the described in stateRemoval function.</i>
---------------	---

Description

stateRemoval2: a function for processing the bayesian network. It implements another approach from the described in stateRemoval function.

Usage

```
stateRemoval2(bn, verbose = FALSE)
```

Arguments

bn A bayesian network (output of buildBN function).
verbose Computation output.

Value

A preprocessed bayesian network.

stateRemovalSubnuucs *stateRemovalSubnuucs: a fuctiong for variable state pruning.*

Description

stateRemovalSubnuucs: a fuctiong for variable state pruning.

Usage

```
stateRemovalSubnuucs(bn, verbose = FALSE)
```

Arguments

bn	A bayesian network (output of buildBN function).
verbose	Computation output.

Value

A preprocessed bayesian network.

sumFactor *prodFactor: a function for performing sum between probability tables.*

Description

prodFactor: a function for performing sum between probability tables.

Usage

```
sumFactor(cpt, Z)
```

Arguments

cpt	Conditional probability table
Z	factor

Value

A dataframe with probabilities.

toybase *Toy allele frequency database.*

Description

Toy allele frequency database.

Usage

toybase

Format

A data frame two markers allele frequencies

toyped *STRs allelic frequencies from specified country.*

Description

STRs allelic frequencies from specified country.

Usage

toyped

Format

A toy pedigree. Nuclear family.

velim.bn *velim.bn: a function for variable elimination in a bayesian network.*

Description

velim.bn: a function for variable elimination in a bayesian network.

Usage

```
velim.bn(  
  bn,  
  ordMethod = c("id", "min_degree", "min_fill", "fixed")[2],  
  orderElim = NULL,  
  verbose = FALSE  
)
```

Arguments

bn	A bayesian network for pedigree object with information of the genotyped members. The ped object must be in Familias format.
ordMethod	Selected ordering method between id, min_degree, min_fill and fixed.
orderElim	Elimination order.
verbose	Computation output.

Value

Variable elimination result.

Examples

```
pbn <- initBN(toyped)
bnet <- buildBN(pbn,QP=3)
bn1 <- buildCPTs(bnet)
resQ <- velim.bn(bn1,ordMethod="min_fill",verbose=FALSE)
```

Index

* datasets

- Argentina_STRs, [2](#)
- toybase, [20](#)
- toyped, [20](#)

Argentina_STRs, [2](#)

buildBN, [3](#)
buildCPTs, [3](#)

convertPedformat, [4](#)

evidencePruning, [5](#)

factorHeteroFounders, [5](#)
FamiliasLocus, [6](#)
FamiliasPedigree, [7](#)
fbnet, [8](#)

getConditional, [8](#)
getGenotypeTables, [9](#)
getLocusCPT, [9](#)
getMAP, [10](#)
getQSetRMP, [10](#)
getValuesOut, [11](#)

imposeEvidence, [11](#)
initBN, [12](#)
initBN.fromPed, [12](#)
initBN.fromVars, [13](#)

minOrdering, [13](#)

preparePed, [14](#)
prodFactor, [14](#)
pruneNodes, [15](#)

removeEvidenceFromPed, [15](#)
reportLR, [16](#)
reportPQ, [16](#)
reverseSplit, [17](#)

setOrdering, [17](#)
stateRemoval, [18](#)
stateRemoval2, [18](#)
stateRemovalSubnucls, [19](#)
sumFactor, [19](#)

toybase, [20](#)
toyped, [20](#)

velim.bn, [20](#)