

# Package ‘fbnet’

February 27, 2023

**Title** Forensic Bayesian Networks

**Version** 1.0.2

## 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** 7.2.1

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

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

**NeedsCompilation** no

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

**Maintainer** Franco Marsico <[franco.lmarsico@gmail.com](mailto:franco.lmarsico@gmail.com)>

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2023-02-27 22:02:35 UTC

## R topics documented:

Argentina_STRs . . . . .	2
bnet . . . . .	3
buildBN . . . . .	3
buildCPTs . . . . .	4
evidencePruning . . . . .	5
factorHeteroFounders . . . . .	5

fbnet . . . . .	6
getConditional . . . . .	6
getGenotypeTables . . . . .	7
getLocusCPT . . . . .	7
getMAP . . . . .	8
getQSetRMP . . . . .	8
getValuesOut . . . . .	9
imposeEvidence . . . . .	9
minOrdering . . . . .	10
pbn . . . . .	10
prodFactor . . . . .	11
pruneNodes . . . . .	11
removeEvidenceFromPed . . . . .	12
reportLR . . . . .	12
reportPQ . . . . .	13
reverseSplit . . . . .	13
setOrdering . . . . .	14
stateRemoval . . . . .	14
stateRemoval2 . . . . .	15
stateRemovalSubnucs . . . . .	15
sumFactor . . . . .	16
toybase . . . . .	16
toyped . . . . .	17
velim.bn . . . . .	17
<b>Index</b>	<b>18</b>

---

Argentina\_STRs

*STRs allelic frequencies from Argentina.*

---

### Description

STRs allelic frequencies from Argentina.

### Usage

Argentina\_STRs

### Format

A data frame with allele frequencies

---

bnet	<i>Initialized bayesian network.</i>
------	--------------------------------------

---

**Description**

Initialized bayesian network.

**Usage**

bnet

**Format**

Initialized bayesian network, based on toyped.

---

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**

```
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,
  bNodePruning = 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

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

---

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.

---

fbnet	<i>fbnet: Forensic Bayesian Networks</i>
-------	--

---

### 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	<i>getConditional: a function for obtaining the conditional probability tables based on a given evidence.</i>
----------------	---

---

### 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**

```
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.

---

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.

---

pbn	<i>Prepared pedigree for bayesian network trimming.</i>
-----	---

---

**Description**

Prepared pedigree for bayesian network trimming.

**Usage**

```
pbn
```

**Format**

A data.matrix.

---

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

---

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

---

### 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.

---

stateRemovalSubnucs	<i>stateRemovalSubnucs: a fuctiong for variable state pruning.</i>
---------------------	--

---

**Description**

stateRemovalSubnucs: a fuctiong for variable state pruning.

**Usage**

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

**Arguments**

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

**Value**

A preprocessed bayesian network.

---

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

---

**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	<i>Toy allele frequency database.</i>
---------	---------------------------------------

---

**Description**

Toy allele frequency database.

**Usage**

```
toybase
```

**Format**

A data frame two markers allele frequencies



---

toyped	<i>STRs allelic frequencies from specified country.</i>
--------	---

---

**Description**

STRs allelic frequencies from specified country.

**Usage**

```
toyped
```

**Format**

A toy pedigree. Nuclear family.

---

velim.bn	<i>velim.bn: a function for variable elimination in a bayesian network.</i>
----------	---

---

**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**

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

# Index

## \* datasets

- Argentina\_STRs, 2
- bnet, 3
- pbn, 10
- toybase, 16
- toyped, 17

Argentina\_STRs, 2

- bnet, 3
- buildBN, 3
- buildCPTs, 4

evidencePruning, 5

- factorHeteroFounders, 5
- fbnet, 6

- getConditional, 6
- getGenotypeTables, 7
- getLocusCPT, 7
- getMAP, 8
- getQSetRMP, 8
- getValuesOut, 9

imposeEvidence, 9

minOrdering, 10

- pbn, 10
- prodFactor, 11
- pruneNodes, 11

- removeEvidenceFromPed, 12
- reportLR, 12
- reportPQ, 13
- reverseSplit, 13

- setOrdering, 14
- stateRemoval, 14
- stateRemoval2, 15

- stateRemovalSubnucs, 15

- sumFactor, 16

- toybase, 16

- toyped, 17

- velim.bn, 17