Package: pedFamilias (via r-universe)

September 6, 2024

Type Package

Title Import and Export 'Familias' Files

Version 0.2.4

Description Tools for exchanging pedigree data between the 'pedsuite' packages and the 'Familias' software for forensic kinship computations (Egeland et al. (2000) <doi:10.1016/s0379-0738(00)00147-x>). These functions were split out from the 'forrel' package to streamline maintenance and provide a lightweight alternative for packages otherwise independent of 'forrel'.

License GPL (>= 3)

URL https://github.com/magnusdv/pedFamilias

BugReports https://github.com/magnusdv/pedFamilias/issues

Depends pedtools, R (>= 4.1.0)

Imports pedmut

Suggests testthat

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://magnusdv.r-universe.dev

RemoteUrl https://github.com/magnusdv/pedfamilias

RemoteRef HEAD

RemoteSha e9430cbc3231e5f77d144c002f9ff9dc29ab0e8c

Contents

`Familias2ped` *Convert* `Familias` *R objects to* `ped`

Description

Convert pedigrees and marker data from the `Familias` R package into the `ped` format used by the `pedsuite`.

Usage

```
Familias2ped(
  familiasped,
  datamatrix,
  loci,
  matchLoci = FALSE,
  prefixAdded = "added_"
)

readFamiliasLoci(loci)
```

Arguments

`familiasped`	A `FamiliasPedigree` object or a list of such.
`datamatrix`	A data frame with two columns per marker (one for each allele) and one row per individual.
`loci`	A `FamiliasLocus` object or a list of such.
`matchLoci`	A logical, by default FALSE. If TRUE, the column names of `datamatrix` are matched against `names(loci)`, or, if these are missing, against the `name` entries of `loci`. The column names of `datamatrix` are assumed to come in pairs with suffixes ".1" and ".2", e.g. "TH01.1", "TH01.2", etc. If FALSE, the `loci` are assumed to be in correct order, and no matching on marker name is done.
`prefixAdded`	A string used as prefix when adding missing parents.

Details

The definition of a *pedigree* in Familias is more liberal than that implemented in the `pedsuite`, which requires that each `ped` object is a connected pedigree, and that each member has either 0 or 2 parents. The conversion function `Familias2ped` takes care of all potential differences. Specifically, it converts each `FamiliasPedigree` object into a list of connected `ped` components, and adds missing parents when needed.

Value

A `ped` object, or a list of such.

References

Familias is freely available from https://familias.name.

See Also

readFam().

Examples

```
famPed = list(id = c('mother', 'daughter', 'AF'),
              findex = c(0, 3, 0),
              mindex = c(0, 1, 0),
              sex = c('female', 'female', 'male'))
class(famPed) = "FamiliasPedigree"

datamatrix = data.frame(
  M1.1 = c(NA, 8, NA),
  M1.2 = c(NA, 9.3, NA),
  row.names = famPed$id)

famLoc = list(locusname = "M1",
              alleles = c("8" = 0.2, "9" = 0.5, "9.3" = 0.3))
class(famLoc) = "FamiliasLocus"

Familias2ped(famPed, datamatrix, loci = famLoc, matchLoci = TRUE)
```

---

readFam *Read Familias* .fam *files*

Description

Parses the content of a .fam file exported from Familias, and converts it into suitable ped objects. This function does not depend on the Familias R package.

Usage

```
readFam(
  famfile,
  useDVI = NA,
  Xchrom = FALSE,
  prefixAdded = "added_",
  fallbackModel = c("equal", "proportional"),
  simplify1 = TRUE,
  deduplicate = TRUE,
  includeParams = FALSE,
  verbose = TRUE
)
```

Arguments

famfile	Path (or URL) to a .fam file.
useDVI	A logical, indicating if the DVI section of the .fam file should be identified and parsed. If NA (the default), the DVI section is included if it is present in the input file.
Xchrom	A logical. If TRUE, the chrom attribute of all markers will be set to "X". Default = FALSE.
prefixAdded	A string used as prefix when adding missing parents.
fallbackModel	Either "equal" or "proportional"; the mutation model to be applied (with the same overall rate) when a specified model fails for some reason. Default: "equal".
simplify1	A logical indicating if the outer list layer should be removed in the output if the file contains only a single pedigree.
deduplicate	A logical, only relevant for DVI. If TRUE (default), redundant copies of the reference pedigrees are removed.
includeParams	A logical indicating if various parameters should be read and returned in a separate list. See Value for details. Default: FALSE.
verbose	A logical. If TRUE, various information is written to the screen during the parsing process.

Value

The output of readFam() depends on the contents of the input file, and the argument includeParams. This is FALSE by default, giving the following possible outcomes:

- If the input file only contains a database, the output is a list of information (name, alleles, frequencies, mutation model) about each locus. This list can be used as locusAttributes in e.g. pedtools::setMarkers().
- If the input file describes pedigree data, the output is a list of ped objects. If there is only one pedigree, and simplify1 = TRUE, the output is a ped object.
- If useDVI = TRUE, or useDVI = NA *and* the file contains DVI data, then the Reference Families section of the file is parsed and converted to ped objects. Each family generally describes multiple pedigrees, so the output gets another layer in this case.

If includeParams = TRUE, the output is a list with elements main (the main output, as described above) and params, a list with some or all of the following entries:

- version: The version of Familias
- dvi: A logical indicating if a DVI section was read
- dbName: The name of the database
- dbSize: A named numeric vector containing the DatabaseSize reported for each marker
- dropoutConsider: A named logical vector indicating for each person if dropouts should be considered
- dropoutValue: A named numeric vector containing the dropout value for each marker
- maf: A named numeric vector containing the "Minor Allele Frequency" given for each marker
- theta: The Theta/Kinship/Fst value given for the marker database

References

Egeland et al. (2000). *Beyond traditional paternity and identification cases. Selecting the most probable pedigree.* Forensic Sci Int 110(1): 47-59.

See Also

`writeFam()`.

Examples

```
# Using example file "paternity.fam" included in the package
fam = system.file("extdata", "paternity.fam", package = "pedFamilias")

# Read and plot
peds = readFam(fam)
plotPedList(peds, hatched = typedMembers, marker = 1)

# Store parameters
x = readFam(fam, includeParams = TRUE)
x$params

stopifnot(identical(x$main, peds))
```

`writeFam` *Export* ped *objects to .fam*

Description

This function produces a `.fam` file readable by the Familias software (Egeland et al., 2000), containing all input pedigrees, their marker data and mutation models. The option `openFam = TRUE` calls `openFamilias()` to open a fresh Familias session with the produced file loaded.

Usage

```
writeFam(
  ...,
  famfile = "ped.fam",
  params = NULL,
  dbOnly = FALSE,
  openFam = FALSE,
  FamiliasPath = NULL,
  verbose = TRUE
)

openFamilias(famfile = NULL, FamiliasPath = NULL, verbose = TRUE)
```

Arguments

...	One or several pedigrees. Each argument should be either a single ped object or a list of such. If the pedigrees are unnamed, they are assigned names "Ped 1", "Ped 2", etc.
famfile	The name or path to the output file to be written. The extension ".fam" is added if missing.
params	A list of further parameters; see readFam() for valid entries. See also Details for default values.
dbOnly	A logical. If TRUE, no pedigree information is included; only the frequency database.
openFam	A logical. If TRUE, an attempt is made to open the produced .fam file in an external Familias session. Only available on Windows systems with a working Familias installation.
FamiliasPath	The path to the Familias executable. If empty, the following are tried in order: "Familias3.exe", "C:/Program Files (x86)/Familias3/Familias3.exe".
verbose	A logical, by default TRUE.

Details

The following parameters are applied by default, but may be adjusted with the params argument:

- version = "3.2.8"
- dvi = FALSE (for now the only valid option)
- dbName = "unknown"
- dbSize = 1000
- dropout = 0
- maf = 0
- theta = 0

The params argument should be a list similar to the params slot produced by readFam() with includeParams = TRUE. Single entries are recycled if needed. If params contains a vector dropout with dropout probabilities for certain pedigree members, it is converted into corresponding dropoutConsider and dropoutValue vectors (see Examples).

Value

The file name is returned invisibly.

References

Egeland et al. (2000). *Beyond traditional paternity and identification cases. Selecting the most probable pedigree.* Forensic Sci Int 110(1): 47-59.

See Also

readFam().

Examples

```
# Create pedigree with 2 markers
x = nuclearPed() |>
  addMarker(geno = c("2/2", "1/3", "2/3"), alleles = 1:3,
            afreq = c(.3,.3,.4), name = "M1")

# Write to .fam
tmp = writeFam(x, famfile = tempfile())

# Read back in
y = readFam(tmp)

stopifnot(identical(x, y))


### With stepwise mutation model
x2 = setMutmod(x, model = "stepwise",
               rate = list(male = 0.001, female = 0.002),
               range = 0.1, rate2 = 0.0001)

# Write and read
y2 = x2 |>
  writeFam(famfile = tempfile()) |>
  readFam()

stopifnot(identical(x2, y2))


### Read/write including detailed parameters
params = list(theta = 0.1, dbName = "myDB", dropout = c("3" = 0.01))
fam = writeFam(x2, famfile = tempfile(), params = params)

dat = readFam(fam, includeParams = TRUE)

# Pedigree is now in the `main` slot
stopifnot(identical(x2, dat$main))

# The `dropout` parameter is converted to (and is equivalent to):
dat$params$dropoutConsider
dat$params$dropoutValue


### Read/write frequency database

# Write database as fam file
dbfam = writeFam(x2, famfile = tempfile(), dbOnly = TRUE)

# Read back in: The result is a list of marker attributes
a = readFam(dbfam)

# Attach to a pedigree and write to a new file
z = singleton(1) |> setMarkers(locusAttributes = a)
```

```
dbfam2 = writeFam(z, famfile = tempfile(), dbOnly = TRUE)

stopifnot(identical(readLines(dbfam), readLines(dbfam2)))
```

Index