Package: KLINK (via r-universe)

November 2, 2024

Title Kinship Analysis with Linked Markers

Version 1.0.3

Description A 'shiny' application for forensic kinship testing, based on the 'pedsuite' R packages. 'KLINK' is closely aligned with the (non-R) software 'Familias' and 'FamLink', but offers several unique features, including visualisations and automated report generation. The calculation of likelihood ratios supports pairs of linked markers, and all common mutation models.

License GPL (>= 3)

URL `https://github.com/magnusdv/KLINK`

BugReports `https://github.com/magnusdv/KLINK/issues`

Depends R (>= 4.1)

Imports forrel (>= 1.7.0), gt (>= 0.10.0), openxlsx, pedFamilias, pedmut (>= 0.7.1), pedprobr (>= 0.9.3), pedtools (>= 2.7.1), shiny (>= 1.9.1), shinydashboard, shinyjs, verbalisr (>= 0.7.1), xml2

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

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

RemoteRef HEAD

RemoteSha 83c70ffb26e8f71701a52004a1391d0a153eb137

Contents

getLinkedPairs 2
launchApp 3
LINKAGEMAP 3
linkedLR 4
loadFamFile 5
markerSummary 5
parseXML 6
paternity 6
writeResult 7

Index 9

`getLinkedPairs` *Identify linked marker pairs*

Description

This function returns a "maximal" set of disjoint pairs of linked markers, given a genetic marker map and a subset of the markers included in the map. The pairs are identified in a greedy manner, successively choosing the closest markers on each chromosome.

Usage

```
getLinkedPairs(markers, linkageMap, maxdist = Inf)
```

Arguments

`markers`	A character vector containing marker names.
`linkageMap`	A data frame with columns including `Marker`, `Chrom` and `PosCM`.
`maxdist`	A positive number indicating the maximum linkage distance (in cM). Markers further apart than this are considered unlinked.

Value

A list of character vectors, each containing two marker names.

Examples

```
# Example using the built-in map of 50 STR markers
map = KLINK::LINKAGEMAP

getLinkedPairs(map$Marker, map, maxdist = 25)
```

`launchApp` *Launch KLINK*

Description

This launches the KLINK app. `runKLINK()` is a synonym for `launchApp()`, but with an additional argument `version`.

Usage

```
launchApp()

runKLINK(version = NULL)
```

Arguments

`version`	A character, e.g. "1.0.0". If the installed version of KLINK differs from this, the program aborts with an error.

Value

No return value, called for side effects.

Examples

```
## Not run:
launchApp()

## End(Not run)
```

`LINKAGEMAP` *Built-in linkage map*

Description

A genetic map including 50 autosomal STR markers.

Usage

```
LINKAGEMAP
```

Format

A data frame with 50 rows and 4 columns: `Marker`, `Kit`, `Chrom`, `PosCM`.

`linkedLR` *LR with pairwise linked markers*

Description

This function does the main LR calculations of the KLINK app.

Usage

```
linkedLR(
  pedigrees,
  linkageMap,
  linkedPairs = NULL,
  maxdist = Inf,
  markerData = NULL,
  mapfun = "Kosambi",
  lumpSpecial = FALSE,
  verbose = FALSE
)
```

Arguments

`pedigrees`	A list of two pedigrees.
`linkageMap`	A data frame with columns including `Marker`, `Chrom` and `PosCM`.
`linkedPairs`	A list of marker pairs. If not supplied, calculated as `getLinkedPairs(markerData$Marker, linkageMap, maxdist = maxdist)`.
`maxdist`	A number, passed onto `getLinkedMarkers()` if `linkedPairs` is NULL.
`markerData`	A data frame with marker data, typically the output of `markerSummary(pedigrees)`.
`mapfun`	Name of the map function to be used; either "Haldane" or "Kosambi" (default).
`lumpSpecial`	A logical, by default FALSE.
`verbose`	A logical, by default FALSE.

Value

A data frame with detailed LR results.

Examples

```
linkedLR(paternity, KLINK::LINKAGEMAP)

# For testing
# .linkedLR(paternity, markerpair = c("SE33", "D6S474"), linkageMap = LINKAGEMAP)
```

`loadFamFile` *Load `.fam` file*

Description

Load `.fam` file

Usage

```
loadFamFile(path, fallbackModel = "equal", withParams = FALSE)
```

Arguments

`path`	The path to a `.fam` file.
`fallbackModel`	The name of a mutation model; passed on to `pedFamilias::readFam()`.
`withParams`	A logical indicating if the Familias parameters should be included in the output. (See `pedFamilias::readFam()`.)

Value

A list of two ped objects.

Examples

```
fam = system.file("extdata/halfsib-test.fam", package = "KLINK")
peds = loadFamFile(fam)
pedtools::plotPedList(peds)
```

`markerSummary` *Generate table of marker data*

Description

Generate table of marker data

Usage

```
markerSummary(pedigrees, replaceNames = FALSE)
```

Arguments

`pedigrees`	A list of 2 pedigrees.
`replaceNames`	A logical, indicating if IDs should be changed to Person1, Person2, ...

Value

A data frame.

Examples

```
markerSummary(paternity)
```

`parseXML` *Parse `XML` file associated with `.fam` file*

Description

Parse `XML` file associated with `.fam` file

Usage

```
parseXML(xml)
```

Arguments

`xml`	Path to a file with extension `.xml`.

Examples

```
# (No example included)
```

`paternity` *Dataset for a paternity case*

Description

A list of two pedigrees forming the hypotheses in a paternity case: H1 (`AF` is the father of `CH`) and H2 (unrelated). `AF` and `CH` are genotyped with 11 markers, with allele frequencies from `forrel::NorwegianFrequencies`.

Usage

```
paternity
```

Format

A list of two pedigrees, named H1 and H2.

Examples

```
pedtools::plotPedList(paternity, marker = "SE33")
markerSummary(paternity)

forrel::kinshipLR(paternity)
```

`writeResult` *Write data and results to Excel*

Description

This function produces an Excel document containing the genotype data and various LR tables.

Usage

```
writeResult(
  resultTable,
  pedigrees,
  linkageMap,
  markerData,
  outfile,
  notes = NULL,
  famname = NULL,
  settings = NULL,
  XML = NULL
)
```

Arguments

`resultTable`	A data frame.
`pedigrees`	A list of two ped objects.
`linkageMap`	A data frame.
`markerData`	A data frame.
`outfile`	The output file name.
`notes`	A character vector.
`famname`	The name of the input `.fam` file.
`settings`	A list of KLINK settings to be included in the output
`XML`	Optional data from .xml file.

Examples

```
# Built-in dataset `paternity`
peds = paternity
map = LINKAGEMAP
mdata = markerSummary(peds)

# Result table
LRtab = linkedLR(pedigrees = peds, linkageMap = map, markerData = mdata)

# Write to excel
tmp = paste0(tempfile(), ".xlsx")
writeResult(LRtab,
            pedigrees = peds,
            linkageMap = map,
            markerData = mdata,
            outfile = tmp)

# openxlsx::openXL(tmp)
```

Index

∗ **datasets**
    `LINKAGEMAP`, 3
    `paternity`, 6

`getLinkedPairs`, 2

`launchApp`, 3
`LINKAGEMAP`, 3
`linkedLR`, 4
`loadFamFile`, 5

`markerSummary`, 5

`parseXML`, 6
`paternity`, 6
`pedFamilias::readFam()`, *5*

`runKLINK` *(*`launchApp`*)*, 3

`writeResult`, 7