

Package: KLINK (via r-universe)

October 13, 2026

Title Kinship Analysis with Linked Markers

Version 1.3.1

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. The program is described in Vigeland and Gilfillan (2026) <[doi:10.1016/j.fsigen.2026.103578](https://doi.org/10.1016/j.fsigen.2026.103578)>.

License GPL (>= 3)

URL <https://github.com/magnusdv/KLINK>,
<https://doi.org/10.1016/j.fsigen.2026.103578>

BugReports <https://github.com/magnusdv/KLINK/issues>

Depends R (>= 4.2)

Imports forrel (>= 1.9.0), gt, norSTR (>= 0.3.0), openxlsx,
pedFamilias, pedmut, pedprobr (>= 1.1.0), pedtools (>= 2.11.0),
scales, shiny (>= 1.9.0), shinyBS, shinydashboard, shinyjs,
verbalisr, xml2

Config/roxygen2/version 8.1.0

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev zlib1g-dev

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

Date/Publication 2026-10-09 13:09:13 UTC

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

RemoteRef HEAD

RemoteSha 48aade7a37eb9583cc6e8467e548a7a9959731ea

Contents

getTotals	2
halfsib	3
launchApp	3
linkedLR	4
loadFamFile	5
loadMap	6
map51	7
markerSummary	7
parseXML	8
paternity	8
sibship	9
writeResult	10
Index	12

getTotals	<i>Get LR totals</i>
-----------	----------------------

Description

Find, or add, the LR totals in a data frame with marker-wise LR values. The totals are found by multiplying the values in each column, after removing NA's.

Usage

```
getTotals(x, cols = c("LRlinked", "LRnolink", "LRnomut"))  
  
addTotals(x, cols = c("LRlinked", "LRnolink", "LRnomut"))
```

Arguments

- x A data frame with LR results, typically the output of [linkedLR\(\)](#).
- cols A vector of column names, by default c("LRlinked", "LRnolink", "LRnomut")

Details

Note that for the multiplication to respect linked markers, the input table should include LR values for only one marker in each linkage group, and NA's elsewhere. This is the format used by [linkedLR\(\)](#).

Value

getTotals() returns a named numeric. addTotals returns a data frame equal to the input, but with a row of totals added at the bottom.

Examples

```
res = linkedLR(paternity)
getTotals(res)
addTotals(res)
```

halfsib

Dataset for a case involving a putative half sibling.

Description

A list of two pedigrees forming opposing hypotheses about three individuals:

Usage

```
halfsib
```

Format

A list of two ped objects, named H1 and H2.

Details

- H1: A and B are full siblings, and C is their half brother
- H2: A and B are full siblings, and C is unrelated to them

The individuals are typed with 50 markers, but some genotypes are missing.

Examples

```
library(pedtools)
plotPedList(halfsib, hatched = typedMembers)
markerSummary(halfsib)
linkedLR(halfsib)
```

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)
```

linkedLR	<i>LR with pairwise linked markers</i>
----------	--

Description

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

Usage

```
linkedLR(
  pedigrees,
  linkageMap = map51,
  linkedPairs = NULL,
  maxdist = Inf,
  markerData = NULL,
  mapfun = "Kosambi",
  lumpSpecial = TRUE,
  alleleLimit = 10,
  verbose = TRUE,
  debug = FALSE
)
```

Arguments

pedigrees A list of two pedigrees.

linkageMap A data frame with columns including Marker, Chr and cM. By default, a built-in map `norSTR::map51` of 51 STR markers is used.

linkedPairs A list of marker pairs. If not supplied, calculated internally with [getLinkedPairs\(\)](#).

maxdist A number, passed onto [getLinkedPairs\(\)](#) 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 indicating if special lumping should be activated. This is strongly recommended in all cases with linked STR markers.
alleleLimit	A number, by default 10, passed on to <code>pedprobr::likelihood2()</code> .
verbose	A logical, by default TRUE.
debug	A logical, by default FALSE.

Value

A data frame with detailed LR results.

Examples

```
linkedLR(paternity)

# Detailed messages, including reports on lumping
linkedLR(paternity, debug = TRUE)

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

loadFamFile	<i>Load .fam file</i>
-------------	-----------------------

Description

Load .fam file

Usage

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

Arguments

path	The path to a .fam file.
fallbackModel	The name of a mutation model; passed on to <code>pedFamiliars::readFam()</code> .
checkLumping	A logical. If TRUE (default), and if the pedigree disallows special lumping, unlumpable models are replaced with the fallback model.
withParams	A logical indicating if the Familias parameters should be included in the output. (See <code>pedFamiliars::readFam()</code> .)

Value

If withParams = FALSE, a list of two pedigree hypotheses in the form of ped objects. If withParams = TRUE, a list with components peds, params and hasUnlumpable.

Examples

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

loadFamFile(fam, withParams = TRUE)$params
```

loadMap

*Genetic map utilities***Description**

Functions for loading, subsetting and pairing markers in genetic linkage maps.

Usage

```
loadMap(path)

getLinkedPairs(linkageMap = map51, maxdist = Inf)

subsetMap(linkageMap, markers)
```

Arguments

path	Path to a genetic map file, either a tab-separated text file or an Excel file with extension .xlsx.
linkageMap	A data frame with columns Marker, Chr, cM and (optionally) Mb, sorted by chromosome and position.
maxdist	A positive number indicating the maximum distance (in cM) between paired markers. Markers further apart are considered unlinked.
markers	A character vector of marker names to retain.

Details

loadMap() reads and sorts a genetic map from a text or Excel file, with required columns Marker, Chr, cM, and optionally a column Mb with physical positions used in karyograms.

subsetMap() restricts a map to a given set of markers. Marker names are matched flexibly, ignoring case and the separators "-", ".", "_" and space. The returned map uses the marker spelling supplied in markers. If multiple map entries match the same marker, only the first is retained.

getLinkedPairs() identifies disjoint marker pairs from the supplied map. Within each chromosome, pairs are selected greedily by successively choosing the closest available markers. Each marker occurs in at most one pair.

Value

loadMap() and subsetMap() return data frames. getLinkedPairs() returns a list of character vectors, each containing two marker names.

Examples

```
map = norSTR::map51

# Keep only markers present in a pedigree
mvec = pedtools::name(sibship[[1]])
map2 = subsetMap(map, mvec)

# Subsetting is insensitive to case and separators
mvec2 = sub(" ", ".", mvec) |> tolower()
subsetMap(map, mvec2)

# Pair the remaining markers
getLinkedPairs(map2, maxdist = 50)
```

map51	<i>Genetic marker maps</i>
-------	----------------------------

Description

These objects are re-exported from [norSTR::map51](#) and [norSTR::map50](#). map51 is the current map, while map50 is retained for reproducibility.

Usage

```
map51

map50
```

markerSummary	<i>Generate table of marker data</i>
---------------	--------------------------------------

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	<i>Parse XML file associated with .fam file</i>
----------	---

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	<i>Dataset for a paternity case</i>
-----------	-------------------------------------

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 norSTR: : norwayDB.

Usage

```
paternity
```

Format

A list of two pedigrees, named H1 and H2.

Examples

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

forrel::kinshipLR(paternity)
```

sibship

Dataset for a full vs half sibship case

Description

A list of two pedigrees forming opposing hypotheses about two individuals NN1 and NN2: FS (full siblings) vs. HS (half siblings). Both individuals are genotyped with the 23 markers included in the Fusion 6C kit. Among the 23 markers, four pairs are linked:

Usage

```
sibship
```

Format

A list of two ped objects, named FS and HS.

Details

- TPOX and D2S441 (86.0 cM, 9 & 13 alleles)
- D5S818 and CSF1PO (26.2 cM, 9 & 11 alleles)
- vWA and D12S391 (11.4 cM, 12 & 23 alleles)
- D21S11 and Penta D (45.7 cM, 26 & 24 alleles)

(Genetic distances taken from the norSTR::map51 linkage map.)

Examples

```
library(pedtools)
plotPedList(sibship, hatched = typedMembers)
markerSummary(sibship)
linkedLR(sibship)
```

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,
  hideEmpty = FALSE,
  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.
hideEmpty	A logical, indicating if untyped markers should be dropped.
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 = norSTR::map51
mdata = markerSummary(peds)

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

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

# openxlsx::openXL(tmp)
```

Index

* datasets

- halfsib, [3](#)
- paternity, [8](#)
- sibship, [9](#)

addTotals (getTotals), [2](#)

getLinkedPairs (loadMap), [6](#)

getLinkedPairs(), [4](#)

getTotals, [2](#)

halfsib, [3](#)

launchApp, [3](#)

linkedLR, [4](#)

linkedLR(), [2](#)

loadFamFile, [5](#)

loadMap, [6](#)

map50 (map51), [7](#)

map51, [7](#)

maptools (loadMap), [6](#)

markerSummary, [7](#)

norSTR::map50, [7](#)

norSTR::map51, [7](#)

parseXML, [8](#)

paternity, [8](#)

pedFamiliias::readFam(), [5](#)

pedprobr::likelihood2(), [5](#)

runKLINK (launchApp), [3](#)

sibship, [9](#)

subsetMap (loadMap), [6](#)

writeResult, [10](#)