

# Package ‘KLINK’

July 2, 2024

**Title** Kinship Analysis with Linked Markers

**Version** 1.0.0

**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 (>= 1.6.0), gt (>= 0.10.0), openxlsx, pedFamilias, pedmut (>= 0.7.1), pedprobr (>= 0.9.3), pedtools (>= 2.6.0), shiny (>= 1.8.0), shinydashboard, shinyjs, verbalisr (>= 0.6.0), xml2

**Encoding** UTF-8

**Language** en-GB

**LazyData** true

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Magnus Dehli Vigeland [aut, cre]  
(<https://orcid.org/0000-0002-9134-4962>)

**Maintainer** Magnus Dehli Vigeland <m.d.vigeland@medisin.uio.no>

**Repository** CRAN

**Date/Publication** 2024-07-02 17:20:07 UTC

## Contents

getLinkedPairs . . . . .	2
launchApp . . . . .	3
LINKAGEMAP . . . . .	3
linkedLR . . . . .	4
loadFamFile . . . . .	5
markerSummary . . . . .	5
parseXML . . . . .	6
paternity . . . . .	6
writeResult . . . . .	7
<b>Index</b>	<b>9</b>

---

getLinkedPairs	<i>Identify linked marker pairs</i>
----------------	-------------------------------------

---

### 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()`.

**Usage**`launchApp()``runKLINK()`**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 = TRUE
)
```

### 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 <code>getLinkedPairs(markerData\$Marker, linkageMap, maxdist = maxdist)</code> .
maxdist	A number, passed onto <code>getLinkedMarkers()</code> if <code>linkedPairs</code> is NULL.
markerData	A data frame with marker data, typically the output of <code>markerSummary(pedigrees)</code> .
mapfun	Name of the map function to be used; either "Haldane" or "Kosambi" (default).
lumpSpecial	A logical, by default TRUE.

### Value

A data frame with detailed LR results.

### Examples

```
library(forrel)

ped1 = nuclearPed(fa = "AF", child = "CH") |>
  profileSim(markers = NorwegianFrequencies)

ped2 = singletons(c("AF", "CH")) |>
  transferMarkers(from = ped1, to = _)

pedigrees = list(ped1, ped2)
```

```

linkedLR(pedigrees, KLINK::LINKAGEMAP)

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

```

---

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

---

### 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 <a href="#">pedFamilies::readFam()</a> .
withParams	A logical indicating if the Families parameters should be included in the output. (See <a href="#">pedFamilies::readFam()</a> .)

### Value

A list of two ped objects.

### Examples

```

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

```

---

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 `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	<i>Write data and results to Excel</i>
-------------	--

---

**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

pedFamilies::readFam(), 5

runKLINK (launchApp), 3

writeResult, 7