

Package: linkmapper (via r-universe)

June 22, 2026

Title Linkmapper: A GUI for Linkage Mapping and QTL Visualisation

Version 0.1.0

Description A Shiny web application that wraps the 'onemap' R package to let users perform linkage mapping and QTL visualisation on biparental mapping populations (F2 intercross and backcross) without writing any code. Provides a graphical workflow for data upload, segregation distortion testing, two-point linkage analysis, marker grouping, linkage group ordering, and final linkage map generation and download. Designed for students and researchers in molecular genetics and plant breeding who require a reproducible, open-source alternative to commercial linkage mapping software.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports shiny (>= 1.7.0), onemap, tools

Suggests shinyjs (>= 2.1.0), waiter, bslib (>= 0.5.0), bsicons, qtl, ggplot2, plotly, DiagrammeR, testthat (>= 3.0.0), knitr, rmarkdown

URL <https://github.com/ebenogoe/linkmapper>

BugReports <https://github.com/ebenogoe/linkmapper/issues>

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-06-22 22:39:40 UTC

RemoteUrl <https://github.com/ebenogoe/linkmapper>

RemoteRef main

RemoteSha 5eae583906672077cd01857e64a73a713d215f13

Contents

group_markers	2
run_linkmapper	3
validate_lm_file	4

Index	5
--------------	----------

group_markers	<i>Group markers into linkage groups</i>
---------------	--

Description

Performs two-point linkage analysis with `onemap::rf_2pts()` and groups markers into linkage groups with `onemap::group()`. This is Module 2 of the Linkmapper analytical workflow.

Usage

```
group_markers(onemap_obj, lod = 3, max_rf = 0.5, map_fun = "kosambi")
```

Arguments

onemap_obj	An onemap object returned by <code>validate_lm_file()</code> or <code>onemap::read_mapmaker()</code> .
lod	Numeric. LOD score threshold for declaring linkage. Passed to <code>onemap::rf_2pts()</code> . Default 3.
max_rf	Numeric in [0, 0.5]. Maximum recombination frequency for declaring linkage. Passed to <code>onemap::rf_2pts()</code> . Default 0.5.
map_fun	Character. Mapping function. One of "kosambi" (default) or "haldane". Passed to <code>onemap::set_map_fun()</code> .

Details

To obtain a data-suggested LOD threshold instead of supplying a fixed value, call `onemap::suggest_lod()` on the onemap object first and pass the result as `lod`.

Value

An object of class "sequence" (as returned by `onemap::group()`) with field `$n.groups` giving the number of linkage groups detected.

Examples

```
## Not run:
f2data <- validate_lm_file("path/to/mydata.txt")

# Use a fixed LOD threshold
lgs <- group_markers(f2data, lod = 3, max_rf = 0.5)
lgs$n.groups
```

```
# Or let onemap suggest the LOD
suggested <- onemap::suggest_lod(f2data)
lgs <- group_markers(f2data, lod = suggested, max_rf = 0.5)

## End(Not run)
```

run_linkmapper

Run the Linkmapper Shiny application

Description

Launches the Linkmapper GUI in the default web browser (or RStudio Viewer). The application provides a point-and-click workflow for linkage mapping and QTL visualisation on biparental mapping populations using the onemap package as the computational back-end.

Usage

```
run_linkmapper(...)
```

Arguments

... Additional arguments passed to `shiny::runApp()`, such as `port`, `launch.browser`, `host`, or `display.mode`.

Value

Does not return a value; called for its side effect of launching the Shiny application. The function blocks until the app is stopped.

Examples

```
## Not run:
# Launch on a random available port and open in the default browser
run_linkmapper()

# Launch on a specific port without opening a browser
run_linkmapper(port = 4321, launch.browser = FALSE)

## End(Not run)
```

validate_lm_file	<i>Validate and read a MAPMAKER linkage mapping input file</i>
------------------	--

Description

Attempts to read a MAPMAKER-format text file using `onemap::read_mapmaker()`, then validates that the result is a non-empty onemap object. This is the first step in the Linkmapper analytical workflow; all downstream functions expect the object this function returns.

Usage

```
validate_lm_file(path)
```

Arguments

path	Character string. Path to the .txt MAPMAKER input file. The file must have a .txt extension and be a valid MAPMAKER/onemap dataset (F2 intercross or backcross format).
------	---

Value

An onemap object as returned by `onemap::read_mapmaker()`. Stops with an informative error if (a) the file cannot be found, (b) the extension is not .txt, (c) `onemap::read_mapmaker()` throws an error, (d) the result is not an onemap object, or (e) the dataset contains zero individuals or zero markers.

Examples

```
## Not run:  
f2data <- validate_lm_file("path/to/mydata.txt")  
f2data # prints onemap summary: n individuals, n markers, cross type  
  
## End(Not run)
```

Index

`group_markers`, [2](#)

`onemap::group()`, [2](#)

`onemap::read_mapmaker()`, [2](#), [4](#)

`onemap::rf_2pts()`, [2](#)

`onemap::set_map_fun()`, [2](#)

`onemap::suggest_lod()`, [2](#)

`run_linkmapper`, [3](#)

`shiny::runApp()`, [3](#)

`validate_lm_file`, [4](#)

`validate_lm_file()`, [2](#)