

# Package ‘MapRtools’

February 17, 2025

**Title** Tools for genetic mapping

**Version** 0.35

**Author** Jeffrey B. Endelman

**Maintainer** Jeffrey Endelman <endelman@wisc.edu>

**Description** Tools for genetic mapping

**Depends** R (>= 4.0)

**License** GPL-3

**LazyData** true

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**Imports** ggplot2, scam, seriation, CVXR, Matrix, HMM, splines2

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

## Contents

EGQ . . . . .	2
genetic_map . . . . .	2
interpolate_cM . . . . .	3
inverse_map_fn . . . . .	4
LDbin . . . . .	4
LG . . . . .	5
LGtrim . . . . .	5
LL . . . . .	6
map_fn . . . . .	6
MLEL . . . . .	7
order_markers . . . . .	7
plot_coverage . . . . .	8
plot_geno . . . . .	8
plot_genoprob . . . . .	9
plot_haplofreq . . . . .	10
plot_LD . . . . .	10
plot_map . . . . .	11
plot_square . . . . .	12
rabbit_diallel . . . . .	12
rabbit_read . . . . .	13

S1_haplo . . . . .	13
S1_selection . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

EGQ	<i>Expected Genotype Quality</i>
-----	----------------------------------

---

### Description

Expected Genotype Quality for Binomial Model

### Usage

EGQ(DP, error, prior)

### Arguments

DP	read depth
error	allelic error
prior	numeric vector of length ploidy + 1

### Details

Expected GQ conditional on the true allele dosage (0,1,2,...ploidy)

### Value

numeric vector of length ploidy + 1

---

genetic_map	<i>Multi-point estimation of a genetic map</i>
-------------	--

---

### Description

Multi-point estimation of a genetic map

### Usage

genetic\_map(x, LOD, n.point = 5)

### Arguments

x	matrix of pairwise map distances (cM) between the marker-bins for one chromosome
LOD	matrix of LOD scores between marker-bins
n.point	Number of points used for estimation

**Details**

Uses LOD-score weighted least-squares regression method described by Stam (1993). Markers must be binned (e.g., using [LDbins](#)) for this function to work properly. Argument `n.point` controls how many pairwise distances are used in the linear regression. `n.point=2` means only adjacent bins; `n.point=3` means adjacent bins and bins with one intervening marker, etc. Marker names taken from the `rownames` attribute of `x`.

**Value**

data frame with columns `marker`, `position` (in cM)

---

<code>interpolate_cM</code>	<i>Interpolate genetic distances</i>
-----------------------------	--------------------------------------

---

**Description**

Creates monotone spline between physical (bp) and genetic (cM) distance

**Usage**

```
interpolate_cM(map, df = 8, max.extend = 5)
```

**Arguments**

<code>map</code>	Data frame with four columns: <code>marker</code> , <code>chrom</code> , <code>bp</code> , <code>cM</code>
<code>df</code>	Degrees of freedom for the spline
<code>max.extend</code>	Maximum distance (in cM) to extrapolate beyond the end of the input map

**Details**

The input map can be generated by merging an existing genetic map with positions in bp and cM with additional markers with only bp information. Interpolation is based on minimizing the mean-squared error between the original and interpolated positions in cM. For the `df` and `max.extend` parameters, use either a single integer (if same for all chromosomes) or a vector of length equal to the number of chromosomes.

**Value**

List containing

**map** Map data frame with additional column named `cM.spline`

**plot** ggplot2 object

---

inverse_map_fn	<i>Inverse map function</i>
----------------	-----------------------------

---

**Description**

Computes recombination frequency from map distance

**Usage**

```
inverse_map_fn(x, model)
```

**Arguments**

x	map distance (cM)
model	Either "Haldane" or "Kosambi"

**Value**

recombination frequency

---

LDbin	<i>Create marker bins based on LD</i>
-------	---------------------------------------

---

**Description**

Create marker bins based on LD

**Usage**

```
LDbin(geno, r2.thresh = 0.99)
```

**Arguments**

geno	matrix of haplotype dosages (markers x indiv)
r2.thresh	threshold for binning

**Details**

Bins are created based on hierarchical clustering with `hclust` and `method='single'`, using  $1 - r^2$  as the dissimilarity metric. The argument `r2.thresh` controls the height for cutting the dendrogram to create the bins. The marker with the least missing data for each bin is chosen to represent it.

**Value**

List containing

**bins** data frame with two columns: marker,bin

**geno** genotype matrix for the bins

**r2** r2 matrix for the bins

---

LG	<i>Make linkage groups based on clustering</i>
----	--

---

**Description**

Make linkage groups based on clustering

**Usage**

```
LG(LODmat, thresh = seq(2, 20, by = 2))
```

**Arguments**

LODmat	matrix of LOD scores for the marker bins
thresh	numeric vector of thresholds for clusterings

**Details**

If thresh is a numeric vector with multiple LOD thresholds, the function returns a plot showing the number of markers per LG. If thresh is a single value, the function returns a data frame with the LG assignment for each marker. LGs are numbered from the largest to smallest group.

**Value**

Either a ggplot2 object or data frame of linkage groups (see Details)

---

LGtrim	<i>Trim a linkage group based on genotype frequencies</i>
--------	---

---

**Description**

Trim a linkage group based on genotype frequencies

**Usage**

```
LGtrim(geno, LODmat, thresh)
```

**Arguments**

geno	matrix of haplotype dosages (markers x samples)
LODmat	matrix of LOD scores for the markers
thresh	numeric vector of thresholds for clusterings

**Details**

This function should only be run on a single linkage group (to form the linkage groups, use [LG](#)). If thresh is a numeric vector with multiple LOD thresholds, the function returns a plot showing the impact of the threshold on genotype frequencies. If thresh is a single value, the function returns a vector of the marker names that are retained. The rownames of geno and LODmat must match.

**Value**

Either a ggplot2 object or a vector of marker names (see Details)

---

LL	<i>Log-likelihood for mapping populations</i>
----	---

---

**Description**

Log-likelihood for mapping populations

**Usage**

```
LL(r, counts, pop.type)
```

**Arguments**

r	recombination frequency
counts	3x3 contingency table for haplotype dosages 0,1,2
pop.type	One of the following: "DH", "BC", "F2", "S1r", "RIL.self", "RIL.sib"

**Details**

The argument counts can be constructed using the table function for two markers. Genotype coding must represent dosage of a founder haplotype. For BC populations, possible allele dosages are 0,1. For DH and RIL pops, it is 0,2. For F2 and S1 pops, it is 0,1,2. S1r is an S1 population with the 1 alleles in repulsion phase.

**Value**

log-likelihood

---

map_fn	<i>Map functions</i>
--------	----------------------

---

**Description**

Computes cM map distance from recombination frequency

**Usage**

```
map_fn(r, model)
```

**Arguments**

r	recombination frequency
model	Either "Haldane" or "Kosambi"

**Value**

Map distance in cM

MLEL

*Max Likelihood Estimation of Linkage***Description**

Max Likelihood Estimation of Linkage

**Usage**

MLEL(geno, pop.type, LOD, n.core = 1, adjacent = FALSE)

**Arguments**

geno	Matrix of haplotype dosages (markers x indiv)
pop.type	One of the following: "DH", "BC", "F2", "S1", "RIL.self", "RIL.sib"
LOD	Logical, whether to return LOD (TRUE) or recomb freq (FALSE)
n.core	For parallel execution on multiple cores
adjacent	Logical, should calculation be done for all pairs (FALSE) or adjacent (TRUE) markers

**Details**

Can be used to estimate either the LOD score or recombination frequency, depending on the value of LOD. Genotype coding must represent dosage of a founder haplotype. For BC populations, possible allele dosages are 0,1. For DH and RIL pops, it is 0,2. For F2 and S1 pops, it is 0,1,2.

**Value**

If adjacent is FALSE, a matrix of recombination frequencies or LOD scores; otherwise, a three-column data frame with marker, the LOD or r value, and the phase ("c","r") with the previous marker

order\_markers

*Order markers by solving the TSP***Description**

Order markers by solving the TSP

**Usage**

order\_markers(x)

**Arguments**

x	distance matrix
---	-----------------

**Details**

Uses R package `seriation` to minimize the distance between adjacent markers. For example, `x` could be a matrix of recombination frequencies or monotone decreasing transformation of LOD scores.

**Value**

a list containing

**order** optimized order as a vector of integers

**distance** sum of adjacent distances

---

plot_coverage	<i>Plot marker coverage of the genome</i>
---------------	---

---

**Description**

Plot marker coverage of the genome

**Usage**

```
plot_coverage(map, limits = NULL)
```

**Arguments**

`map` data frame with columns `chrom` & `position`

`limits` optional data frame with columns `chrom` & `position`, with the maximum length for each chromosome

**Details**

If `limits` not provided, then the maximum values in `map` are used.

**Value**

ggplot2 variable

---

plot_genotype	<i>Graphical genotyping</i>
---------------	-----------------------------

---

**Description**

Graphical genotyping

**Usage**

```
plot_genotype(genotype, map = NULL)
```



**Arguments**

geno	genotype matrix (markers x indiv)
map	optional data frame with 3 columns (marker, chrom, position)

**Details**

Input matrix geno should have rownames attribute that matches marker names in the first column of map (when present).

**Value**

ggplot object

---

plot_genoprob	<i>Plot genotype probabilities for one chromosome</i>
---------------	---

---

**Description**

Plot genotype probabilities for one chromosome

**Usage**

```
plot_genoprob(genoprob, map = NULL)
```

**Arguments**

genoprob	matrix (markers x genotypes) of probabilities for one individual
map	optional data frame with 3 columns (marker, chrom, position)

**Details**

Names for the genotypes are taken from the colnames of genoprob.

**Value**

ggplot object

---

plot_haplofreq	<i>Plot and filter markers based on haplotype frequency vs position</i>
----------------	---

---

### Description

Plot and filter markers based on haplotype frequency vs position

### Usage

```
plot_haplofreq(geno, thresh = 0.1, span = 0.3)
```

### Arguments

geno	haplotype dosage matrix (markers x indiv)
thresh	threshold for removing markers (see Details)
span	parameter to control degree of smoothing for spline (higher = less smooth)

### Details

This functions works for mapping populations derived from two inbred founders. The geno matrix must be coded based on the dosage (0,1,2) of one founder haplotype.

Markers are removed if their residual to the fitted spline exceeds thresh. Markers are assumed to be ordered. Function designed to be used for one chromosome.

### Value

List containing

**outliers** index of outlier markers

**plot** ggplot2 variable

---

plot_LD	<i>Plot LD vs distance</i>
---------	----------------------------

---

### Description

Plot LD vs distance

### Usage

```
plot_LD(r2, map, max.pair = 10000, dof = 8)
```

### Arguments

r2	squared correlation matrix
map	data frame with 3 columns (marker, chrom, position)
max.pair	maximum number of r2 pairs for the spline
dof	degrees of freedom for the spline

**Details**

A monotone decreasing, convex spline is fit using R package `scam`. The input matrix `r2` should have rownames attribute that matches marker names in the first column of `map`.

**Value**

List containing

**plot** ggplot object

**spline** data frame with fitted values for the spline

---

plot_map	<i>Plot data against position and curate based on trend</i>
----------	---

---

**Description**

Plot data against position and curate based on trend

**Usage**

```
plot_map(data, thresh = NULL, span = 0.3)
```

**Arguments**

<code>data</code>	data frame with 3 columns: chrom, position, plotting variable
<code>thresh</code>	threshold for removing markers (see Details)
<code>span</code>	smoothing parameter for loess spline (higher = less smooth)

**Details**

For data from a single chromosome, markers are removed when their residual to the fitted spline exceeds `thresh`. If `NULL`, no curation occurs.

**Value**

ggplot

List containing

**outliers** outlier positions

**plot** ggplot2 variable

---

plot_square	<i>Plot square (dis)similarity matrix</i>
-------------	---

---

**Description**

Plot square (dis)similarity matrix

**Usage**

```
plot_square(data, lims = NULL)
```

**Arguments**

data	square matrix
lims	numeric 3-vector with the low,mid,high points for the colors

**Details**

Can be used to plot squared correlation, recomb frequency, LOD and more. By default, lims equals (0,median,max)

**Value**

ggplot2 variable

---

rabbit_diallel	<i>Make RABBIT input files for diploid diallel population</i>
----------------	---

---

**Description**

Make RABBIT input files for diploid diallel population

**Usage**

```
rabbit_diallel(ped, geno, geno.founder, map, outstem)
```

**Arguments**

ped	data frame with pedigree (pop,parent1,parent2)
geno	list of genotype matrices (markers x indiv), one for each population in ped
geno.founder	matrix of genotype data for the founders (markers x indiv)
map	genetic map (marker,chromosome,position)
outstem	name for output files

**Details**

Populations must be numbered in ped corresponding to their position in geno. Founders are not included in ped. All genotype matrices must have identical markers. Genetic map position should be in cM. Genotypes need to be coded according to RABBIT format.

---

rabbit_read	<i>Parse output from RABBIT MagicReconstruct</i>
-------------	--

---

### Description

Parse output from RABBIT MagicReconstruct

### Usage

```
rabbit_read(rabbit.file, ML.file = NULL, diaQTL.file = NULL)
```

### Arguments

rabbit.file	name of RABBIT output file
ML.file	name of most likely genotype file to create
diaQTL.file	name of diaQTL genotype file to create

### Details

Two different file formats can be created. The `ML.file` contains the most likely (i.e., posterior maximum) genotype for each individual at each marker. The `diaQTL.file` contains the full distribution of genotype probabilities in the format required by the diaQTL R package (`diaQTL.file`). The default value for each filename is `NULL`, which generates no file.

### Value

data frame defining the genotypes

---

S1_haplo	<i>Phase S1 parent and reconstruct progeny in terms of parental haplotypes</i>
----------	--

---

### Description

Phase S1 parent and reconstruct progeny in terms of parental haplotypes

### Usage

```
S1_haplo(geno, r, error)
```

### Arguments

geno	ordered genotype matrix (markers x indiv) for one chromosome
r	average recombination frequency to use for the HMM
error	average genotype error to use for the HMM

**Details**

It is assumed that only segregating markers are present. Progeny reconstruction occurs using an HMM with a uniform transition probability matrix, based on an average recombination frequency *r*, and a uniform model for the genotype error.

**Value**

- List containing
  - parent** two column matrix (rows = markers) with the haplotypes for the parent
  - progeny** matrix with progeny reconstructed based on dosage of the second parental haplotype

---

S1_selection	<i>Signatures of selection in S1 populations</i>
--------------	--

---

**Description**

Signatures of selection in S1 populations

**Usage**

```
S1_selection(data, alpha = 0.05)
```

**Arguments**

- data** data frame with columns: marker, chrom, position, AA, AB, BB. Columns 4-6 have count data.
- alpha** significance level

**Details**

Genotypes must be coded based on the S1 parental haplotypes, not markers.

The null hypothesis is no selection, in which case the expected frequency of genotypes is (AA = 1/4, AB = 1/2, BB = 1/4). Two alternate hypotheses are tested for gametic selection: 1.selection in one sex, 2.selection in both sexes. Two models of zygotic selection are also tested: 1.selection against one homozygote, 2.selection against both homozygotes. The selection coefficient equals the sum of the absolute differences between the observed and expected frequencies. Positive values correspond to selection against A or AA, negative values for selection against B or BB. For zygotic2, positive (negative) values represent selection against (for) homozygotes.

P-values are computed based on the likelihood ratio test; in other words, the change in deviance is assumed to be chi-squared distributed under the null hypothesis.

**Value**

- list with "plot" and "table" of results:
  - model** name of best model
  - s** selection coefficient
  - score** -log10(p) value

# Index

EGQ, [2](#)

genetic\_map, [2](#)

interpolate\_cM, [3](#)

inverse\_map\_fn, [4](#)

LDbin, [3](#), [4](#)

LG, [5](#), [5](#)

LGtrim, [5](#)

LL, [6](#)

map\_fn, [6](#)

MLEL, [7](#)

order\_markers, [7](#)

plot\_coverage, [8](#)

plot\_geno, [8](#)

plot\_genoprob, [9](#)

plot\_haplofreq, [10](#)

plot\_LD, [10](#)

plot\_map, [11](#)

plot\_square, [12](#)

rabbit\_diallel, [12](#)

rabbit\_read, [13](#)

S1\_haplo, [13](#)

S1\_selection, [14](#)