Package 'MDSMap'

July 21, 2025

Type Package

Title High Density Genetic Linkage Mapping using Multidimensional Scaling

Version 1.3

Date 2025-04-28

Description Estimate genetic linkage maps for markers on a single chromosome (or in a single linkage group) from pairwise recombination fractions or intermarker distances using weighted metric multidimensional scaling. The methods are suitable for autotetraploid as well as diploid populations. Options for assessing the fit to a known map are also provided. Methods are discussed in detail in Preedy and Hackett (2016) <doi:10.1007/s00122-016-2761-8>.

License GPL-3 | file LICENSE

Depends R (>= 3.2.0)

Imports smacof (>= 1.9.0), princurve (>= 2.0.2), rgl, reshape

RoxygenNote 7.2.3

Suggests qtl, R.rsp

VignetteBuilder R.rsp

Encoding UTF-8

NeedsCompilation no

Author Katharine F. Preedy [aut],
Christine A. Hackett [aut],
Bram Boskamp [cre]

Maintainer Bram Boskamp <bram.boskamp@bioss.ac.uk>

Repository CRAN

Date/Publication 2025-04-28 14:50:02 UTC

Contents

`MDSMap-package` *High density Genetic Linkage Mapping using Multidimensional Scaling*

Description

MDSmap provides functions for estimating genetic linkage maps for markers from a single linkage group from pairwise intermarker map distances using the Haldane or Kosambi map function; or recombination fractions. It either uses constrained weighted metric multidimensional scaling (cMDS) in 2 dimensions or unconstrained weighted metric multidimensional scaling (MDS) followed by fitting a principal curve (PC) in either 2 or 3 dimensions. Pairwise distances can be weighted either by the LOD score or LOD2. There are functions for diagnostic plots, estimating the difference between the observed and estimated difference between points and their nearest informative neighbour, which may be useful in deciding which weights to use and also for testing estimated maps against a map estimated externally.

Details

The main top level functions to use: `calc.maps.pc` and `calc.maps.sphere`, and use `plot.pcmap`, `plot.spheremap` or `plot.pcmap3d` to visualize the result.

Author(s)

Katharine F. Preedy <Katharine.preedy@bioss.ac.uk>

Examples

```
map<-calc.maps.pc(system.file("extdata", "lgI.txt", package="MDSMap"),
ndim=2,weightfn='lod',mapfn='haldane')
plot(map)
```

`calc.maps.pc` *Estimate marker positions using Principal Curves*

Description

Reads a text file of pairwise recombination fractions and LOD scores, reduces to 2 or 3 dimensions using wMDS and projects onto a single dimension using principal curves to estimate marker positions.

Usage

```
calc.maps.pc(
  fname,
  spar = NULL,
  n = NULL,
  ndim = 2,
  weightfn = "lod2",
  mapfn = "haldane"
)
```

Arguments

`fname`	Character string the name of the file of recombination fractions and scores it should not contain any suffices (the file should be a .txt file as described below).
`spar`	Integer - the smoothing parameter for the principal curve. If NULL this will be done using leave one out cross validation.
`n`	Vector of integers or character strings containing markers to be omitted from the analysis.
`ndim`	Number of dimensions in which to perform the wMDS and fit the curve - can be 2 or 3.
`weightfn`	Character string specifying the values to use for the weight matrix in the MDS `'lod2'` or `'lod'`.
`mapfn`	Character string specifying the map function to use on the recombination fractions `'haldane'` is default, `'kosambi'` or `'none'`.

Details

Reads a file of the form described below and casts the data into matrices of pairwise recombination fractions and weights determined by the `weightfn` parameter (`LOD` or `LOD^2^`) calculates a distance matrix from the map function. Haldane is the default map function, none just uses recombination fractions and the other alternative is Kosambi (see `dmap` for details).

Performs both an weighted MDS on the distance matrix using `smacofSym` and `smacofSphere` (*de Leeuw & Mair 2009*) and fits a principal curve to map this to an interval (`principal_curve` for details).

File names should be of the form `fname.txt` and it is assumed that they are in a tab or space separated file of the format displayed below. The first entry on the first row is the number of markers to be analysed. Underneath this is a table in which the first two columns contain marker names, the third column contains the pairwise recombination fractions between the markers and the fourth column the associated lod score. Note that marker names in the first column vary more slowly than in the second column. Missing recombination pairs are acceptable. Recombination fractions greater than 0.499999 are set to that value.

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
1          2          .                        .
1          3          .                        .
1          4          .                        .
.          .          .                        .
.          .          .                        .
.          .          .                        .
2          3          .                        .
2          4          .                        .
.          .          .                        .
```

Value

A list (S3 class pcmap or pcmap3d depending on ndim) with the following elements:

`smacofsym`	The unconstrained wMDS results.
`pc`	The results from the principal curve fit.
`distmap`	A symmetric matrix of pairwise distances between markers where the columns are in the estimated order.
`lodmap`	A symmetric matrix of lod scores associated with the distances in distmap.
`locimap`	A data frame of the markers containing the name of each marker, the number in the configuration plot if that is being used, the position of each marker in order of increasing distance and the nearest neighbour fit of the marker.
`length`	Integer giving the total length of the segment.
`removed`	A vector of the names of markers removed from the analysis.
`locikey`	A data frame showing the number associated with each marker name for interpreting the wMDS configuration plots.
`meannnfit`	The mean across all markers of the nearest neighbour fits.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31: 1-30 https://www.jstatsoft.org/v31/i03/

Hastie T, Weingessel A (2013) princurve: Fits a Principal Curve in Arbitrary Dimension.) R package version 1.1-12. https://CRAN.R-project.org/package=princurve

See Also

calc.maps.sphere, calc.pair.rf.lod, smacofSym, smacofSphere, map.to.interval, dmap

Examples

```
map<-calc.maps.pc(system.file("extdata", "lgI.txt", package="MDSMap"),
ndim=2,weightfn='lod',mapfn='haldane')
plot(map)
```

`calc.maps.sphere` *Estimate marker positions using spherically constrained weighted MDS*

Description

Reads a text file of pairwise recombination fractions and LOD scores, estimates marker positions using spherically constrained weighted MDS

Usage

```
calc.maps.sphere(
  fname,
  p = 100,
  n = NULL,
  weightfn = "lod2",
  mapfn = "haldane"
)
```

Arguments

`fname`	Character string specifying the base name of the file fname.txt which contains the data to be analysed. The file should be white space or tab separated.
`p`	Integer - the penalty for deviations from the sphere - higher p forces points more closely onto a sphere.
`n`	Vector of integers or strings containing markers to be omitted from the analysis.
`weightfn`	Character string specifying the values to use for the weight matrix in the MDS 'lod2' or 'lod'.
`mapfn`	Character string specifying the map function to use on the recombination fractions 'haldane' is default, 'kosambi' or 'none'.

Details

This can be very slow with large sets of markers, in which case it may be better to consider `calc.maps.pc`.

Reads a file of the form described below and casts the data into matrices of pairwise recombination fractions and weights determined by the `weightfn` parameter (`LOD` or `LOD^2^`) calculates a distance matrix from the map function. Haldane is the default map function, None just uses recombination fractions and the other alternative is Kosambi (see `link{dmap}` for details).

Performs both an unconstrained and dual spherically constrained weighted MDS on the distance matrix using `smacofSym` and `smacofSphere` (*de Leeuw & Mair 2009*) and maps this to an interval (see `map.to.interval` for details).

Inevitably the constrained MDS has higher stress than the unconstrained MDS and a good rule of thumb is that this should not be more than about 10

File names should be of the form `fname.txt` and it is assumed that they are in a tab or space separated file of the format displayed below. The first entry on the first row is the number of markers to be analysed. Underneath this is a table in which the first two columns contain marker names, the third column contains the pairwise recombination fractions between the markers and the fourth column the associated LOD score. Note that marker names in the first column vary more slowly than in the second column. Missing recombination pairs are acceptable. Recombination fractions greater than 0.499999 are set to that value.

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
1          2          .                        .
1          3          .                        .
1          4          .                        .
.          .          .                        .
.          .          .                        .
.          .          .                        .
2          3          .                        .
2          4          .                        .
.          .          .                        .
```

Value

A list (S3 class 'spheremap') with the following elements:

`smacofsym`	The unconstrained wMDS results.
`smacofsphere`	The spherically constrained wMDS results.
`mapsphere`	Map of the markers onto an interval containing order-the rank of each marker.
`distmap`	A symmetric matrix of pairwise distances between markers where the columns are in the estimated order.
`lodmap`	A symmetric matrix of lod scores associated with the distances in distmap.
`locimap`	A data frame of the markers containing the name of each marker, the number in the configuration plot if that is being used, the position of each marker in order of increasing distance and the nearest neighbour fit of the marker.

length	Integer giving the total length of the segment.
removed	A vector of the names of markers removed from the analysis.
locikey	A data frame showing the number associated with each marker name for interpreting the wMDS configuration plots.
stressratio	The ratio of the constrained to unconstrained stress.
ssphere	The stress per point of the spherically constrained wMDS.
ssym	Stress per point of the unconstrained wMDS.
meannnfit	The mean across all markers of the nearest neighbour fits.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31: 1-30 https://www.jstatsoft.org/v31/i03/

See Also

calc.maps.pc, calc.pair.rf.lod, smacofSym, smacofSphere, map.to.interval, dmap, calc.nnfit

Examples

```
smap<-calc.maps.sphere(system.file("extdata", "lgI.txt", package="MDSMap"),
weightfn='lod',mapfn='kosambi')
plot(smap)
```

---

calc.nnfit *Calculate the nearest neighbour fit.*

Description

Calculates the total, mean and individual differences between the observed and estimated distances from all loci and their nearest neighbours with non-zero LOD scores.

Usage

```
calc.nnfit(distmap, lodmap, estmap)
```

Arguments

distmap	Symmetric matrix of pairwise inter-marker distances with columns and rows corresponding to the estimated map order.
lodmap	Symmetric matrix of pairwise lod scores with columns and rows corresponding to the estimated map order.
estmap	Vector of estimated marker positions.

Details

The nearest neighbour fit for a marker is the sum of the difference between the observed and estimated distances between the marker and its nearest informative neighbour. A neighbour is informative if the LOD score for the inter-marker distance is greater than zero. This function calculates the nearest neighbour fit for each marker and returns the fit for each point and the sum of all the fits.

Value

A list with the elements:

`fit`	Sum of the nearest neighbour fits over all markers.
`pointfits`	Vector of nearest neighbour fits for each marker.
`meanfit`	Mean of the nearest neighbour fits over all markers.

See Also

`calc.nnfit.loci`

---

calc.nnfit.from.file *Nearest neighbour fit from estimated map and file of pairwise recombination fractions.*

Description

Calculates a nearest neighbour fit based from an estimated map and a file containing pairwise recombination fractions and LOD scores.

Usage

```
calc.nnfit.from.file(
  estmap,
  fname,
  mapfn = "haldane",
  n = NULL,
  header = FALSE
)
```

Arguments

`estmap`	A character string indicating the name of a comma separated value file with the first column containing marker names in the order of their estimated position.
`fname`	A character string specifying the base name of the file `fname.txt` which contains the data to be analysed the file should be white space or tab separated.
`mapfn`	Character string, `'haldane'`, `'kosambi'` or `'none'` specifying the values to use to estimate the map distance from the recombination fractions. Default is `'haldane'`.

n	Vector of character strings or numbers specifying the markers to be omitted from the analysis. Default is NULL.
header	Logical argument indicating whether the .csv file estmap contains headers - default is TRUE

Details

Reads in two files fname.txt and estmap.

The data is cast the data into symmetric matrices of pairwise recombination fractions and LOD scores with the order of columns and rows in the matrix determined by the order specified in estmap. A distance matrix is calculated according to the method specified by mapfn. Haldane is the default map function, None just uses recombination fractions and the other alternative is Kosambi (see dmap for details). The nearest neighbour fit is then calculated (see calc.nnfit for details)

estmap should contain marker names in the first column in the order of the estimated map.

fname should be of the form fname.txt and it is assumed that they are in a tab or space separated file of the format displayed below. The first entry on the first row is the number of markers to be analysed. Underneath this is a table in which the first two columns contain marker names, the third column contains the pairwise recombination fractions between the markers and the fourth column the associated LOD score. Note that marker names in the first column vary more slowly than in the second column. Missing recombination pairs are acceptable. Recombination fractions greater than 0.499999 are set to that value.

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
1          2          .                        .
1          3          .                        .
1          4          .                        .
.          .          .                        .
.          .          .                        .
.          .          .                        .
2          3          .                        .
2          4          .                        .
.          .          .                        .
```

Value

A list with the following elements:

fit	Sum over all markers of the nearest neighbour fits.
pointfits	The nearest neighbour fit for each marker.
meanfit	Mean of the nearest neighbour fits over all markers.

See Also

dmap, calc.nnfit, calc.pair.rf.lod

`calc.nnfit.loci` *Calculates the nearest neighbour fit for an individual marker.*

Description

Calculates the nearest neighbour fit for an individual marker.

Usage

```
calc.nnfit.loci(loci, distmap, lodmap, estmap)
```

Arguments

`loci`	Scalar indicating the estimated rank position of the marker.
`distmap`	Symmetric matrix of pairwise inter-marker distances with columns and rows corresponding to the estimated map order.
`lodmap`	Symmetric matrix of pairwise lod scores with columns and rows corresponding to the estimated map order.
`estmap`	Vector of estimated marker positions.

Details

The nearest neighbour fit for a marker is the sum of the difference between the observed and estimated distances between the marker and its nearest informative neighbour. A neighbour is informative if the LOD score for the inter-marker distance is non zero. This function finds the nearest markers with a non-zero LOD score (this may be one or two markers). Calculates the estimated distances between these markers and the marker of interest and returns the sum of the absolute values of the difference between the observed and estimated distances.

Value

Scalar corresponding to the difference between the observed and estimated intermarker differences.

`calc.nswaps` *Calculates the number of swaps required to move from one order to another.*

Description

Calculates the number of swaps required to move from one order to another.

Usage

```
calc.nswaps(map1, map2)
```

Arguments

map1	Vector of marker positions or ranks.
map2	Vector of marker positions or ranks.

Details

This is intended to be used when comparing an estimated marker ordering to some perceived "truth". It is most likely to be useful when dealing with simulated data where the concept of truth makes most sense. It calculates the minimum number of single place swaps that would be needed to move from map1 to map2 and it does this by reverse engineering kendall's tau b correlation coefficient

$$\tau = \frac{2(C-D)}{N}$$

where N is the total number of pairs of markers, C the number of concordant pairs and D the number of discordant pairs. If there are n markers then the total number of pairs $N = \binom{n}{2}$ and $C = N - D$ so $D = 0.5\binom{n}{2}(1-\tau)$ and the minimum number of swaps is the minimum of D and $N - D$

Value

Scalar giving the number of swaps.

---

calc.pair.rf.lod *Create recombination matrix from pairwise data file.*

Description

Reads a text file of pairwise recombination fractions and LOD scores and casts it into a matrix of recombination fractions and weights.

Usage

```
calc.pair.rf.lod(fname, weightfn = "lod", ...)
```

Arguments

fname	Character string specifying the base name of the file fname.txt which contains the data to be analysed the file should be white space or tab separated.
weightfn	Character string specifiying the values to use for the weight matrix 'lod2' or 'lod'.
...	read.table arguments.

Details

File names should be of the form `fname.txt` and it is assumed that they are in a tab or space separated file of the format displayed below. The first entry on the first row is the number of markers to be analysed. Underneath this is a table in which the first two columns contain marker names, the third column contains the pairwise recombination fractions between the markers and the fourth column the associated LOD score. Note that marker names in the first column vary more slowly than in the second column. Missing recombination pairs are acceptable. Recombination fractions greater than 0.499999 are set to that value

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
1          2          .                        .
1          3          .                        .
1          4          .                        .
.          .          .                        .
.          .          .                        .
.          .          .                        .
2          3          .                        .
2          4          .                        .
.          .          .                        .
```

Value

A list with the following elements:

`rf`	A symmetric matrix of recombination fractions.
`nloci`	The number of markers in the analysis.
`locinames`	The names of the markers in the analysis.

Examples

```
lodrf<-calc.pair.rf.lod(system.file("extdata", "lgI.txt", package="MDSMap"),
"lod2")
```

`convert.polar` *Convert Cartesian coordinates from wMDS coordinates to polar coordinates.*

Description

Converts the coordinates of points in the final configuration of a spherically constrained wMDS from Cartesian to polar coordinates.

Usage

```
convert.polar(mdsobject, nloci)
```

Arguments

mdsobject	Output from smacofSphere using the dual method in the **smacof** package.
nloci	The number of markers in the configuration.

Details

Centres the circle on zero if necessary, finds a the most natural break in the points to start as 0, then calculates the angle of each point relative to this. The radius is the median distance of points from the centre.

Value

theta	A vector of angles one for each point.
radius	A scalar the radius of sphere.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31:1-30 https://www.jstatsoft.org/v31/i03/

See Also

smacofSphere

Examples

```
#M and lod should be n x n symmetric matrices of the same dimensions where n
#is the number markers to be analysed
## Not run:
mds1<-smacofSphere(M,ndim=2,algorithm="dual",weightmat=lod,penalty=100)
pol<-convert.polar(mds1,n)

## End(Not run)
```

---

dmap *Calculates pairwise map distances from the recombination fraction.*

Description

Calculates pairwise map distances from the recombination fraction.

Usage

```
dmap(rf, mapfn = "haldane")
```

Arguments

rf	A symmetric matrix of pairwise recombination fractions.
mapfn	A character string specifying the map function to be used in calculated the distance 'haldane', 'kosambi', 'none'.

Details

The default is the 'haldane' map function $0.5\ln(1-2rf)$, 'kosambi' returns $0.25\ln((1+2rf)/(1-2rf))$ and 'none' returns rf, the recombination fraction.

Value

a symmetric matrix of pairwise map distances in the same format as the recombination matrix supplied.

Examples

```
lodrf<-calc.pair.rf.lod(system.file("extdata", "lgI.txt", package="MDSMap"))
mdist=dmap(lodrf$rf,mapfn="haldane")
```

---

dmap.check *Reorders a distance map by a new marker order.*

Description

Reorders a distance map by a new marker order.

Usage

```
dmap.check(distmap, newrank)
```

Arguments

distmap	A symmetric matrix of pairwise inter-marker distances.
newrank	A vector of scalars giving the new rank of each marker, markers should appear in the same order as in the distmap.

Details

The rows and columns in distmap are reordered such that if entry i in newrank has value j then row j and column j in the new matrix are row i and column i from distmap.

Value

Matrix of pairwise inter-marker distances.

Examples

```
s<-matrix(1:25,nrow=5)
s<-0.5*(s+t(s))
rank<-c(1,3,4,2,5)
dmap.check(s,rank)
```

`estimate.map` *Load data, estimate a linkage map and plot diagnostics for the fit.*

Description

Load data, estimate a linkage map and plot diagnostics for the fit.

Usage

```
estimate.map(
  fname,
  p = NULL,
  n = NULL,
  ispc = TRUE,
  ndim = 2,
  weightfn = "lod2",
  mapfn = "haldane",
  D1lim = NULL,
  D2lim = NULL,
  D3lim = NULL,
  displaytext = TRUE
)
```

Arguments

`fname`	Character string containing the base file from which the data should be read - should contain the complete file name excluding the suffix which should be `.txt`
`p`	Smoothing parameter.
`n`	Vector of integers or character strings containing the name or position in the input list of loci to be excluded from the analysis.
`ispc`	Logical determining the method to be used to estimate the map. By default this is `TRUE` and the method of principal curves will be used. If `FALSE` then the constrained MDS method will be used.
`ndim`	Integer the number of dimensions to use if the Principal curves method is used. By default this is 2, but it can also be 3.
`weightfn`	Character string specifying the values to use for the weight matrix in the MDS `lod2` or `lod`.
`mapfn`	Character string specifying the map function to use on the recombination fractions `'haldane'` is default, `'kosambi'` or `'none'`.

D1lim	Numeric vector specifying the limits of the axis relating to dimension 1 of the wMDS used to estimate the map.
D2lim	Numeric vector specifying the limits of the axis relating to dimension 2 of the wMDS used to estimate the map.
D3lim	Numeric vector specifying the limits of the axis relating to dimension 3 of the wMDS used to estimate the map.
displaytext	Logical argument determining how markers should be labelled in the wMDS configuration plot. If TRUE then marker names are used. If FALSE then numbers are used.

Details

Data is read from a text file which should be of the form described below. By default, ispc=TRUE, in which case maps are estimated using unconstrained weighted MDS followed by fitting a principal curve. Details can be found in the description of the function calc.maps.pc. If ispc=FALSE maps are estimated using spherically constrained weighted MDS. Details can be found in the description of the function calc.maps.sphere.

ndim is only relevant if ispc=TRUE, in which case it specifies the number of dimensions to be used, the default is 2 but it can also be 3 dimensions.

Diagnostic plots are then produced using plot.pcmap for the method of principal curves in 2 dimensions, plot.pcmap3d for the method of principal curves in 3 dimensions and plot.spheremap for the method using spherically constrained MDS.

n specifies markers to be omitted from the analysis. It can be a vector of character strings specifying makers to be omitted, or a vector of integers specifying the markers to omit. The latter method is likely to be useful when removing outliers after inspection of the diagnostic plot, because the output contains a dataframe, locikey, which associates each marker with its identifying number. By default this is NULL and all markers in the file will be analysed.

p is a smoothing parameter which operates quite differently depending on whether map estimation is performed using Principal Curves or Constrained MDS. If the PC method is used, p determines the smoothing parameter spar in the function principal_curve from the package **princurve**. If NULL then the most appropriate value will be determined using leave one out cross validation. If Constrained MDS is used then p must be set to a number which specifies the penalty for deviations from the sphere in the function smacofSphere from the **smacof** package. Something between 50 and 100 is generally appropriate and this penalty can be decreased if stress from the constrained analysis is more than about 10 for details)

File names should be of the form fname.txt and it is assumed that they are in a tab or space separated file of the format displayed below. The first entry on the first row is the number of markers to be analysed. Underneath this is a table in which the first two columns contain marker names, the third column contains the pairwise recombination fractions between the markers and the fourth column the associated LOD score. Note that marker names in the first column vary more slowly than in the second column. Missing recombination pairs are acceptable. Recombination fractions greater than 0.499999 are set to that value.

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
```

1	2	.	.
1	3	.	.
1	4	.	.
.	.	.	.
.	.	.	.
.	.	.	.
2	3	.	.
2	4	.	.
.	.	.	.

Value

map (s3 class pcmap, pcmap3d or spheremap) from `calc.maps.pc` if `ispc=TRUE` or `calc.maps.sphere` if `ispc=FALSE`.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31: 1-30 https://www.jstatsoft.org/v31/i03/

Hastie T, Weingessel A (2013) princurve: Fits a Principal Curve in Arbitrary Dimension.) R package version 1.1-12. https://CRAN.R-project.org/package=princurve

See Also

`smacofSphere`, `principal_curve`, `calc.maps.pc`, `calc.maps.sphere`, `plot.pcmap`, `plot.pcmap3d`, `plot.spheremap`

Examples

```
estimate.map(system.file("extdata", "lgI.txt", package="MDSMap"),
ndim=2)
```

`get.dist.loci` *Calculates the distance of a marker from some objective "truth".*

Description

Calculates the distance of a marker from some objective "truth".

Usage

```
get.dist.loci(loci, estmap, realmap)
```

Arguments

loci	Character string or number specifying the marker name.
estmap	Data frame in which has a column called "names" containing marker names and a column called "position" containing marker positions.
realmap	Data frame in which the first column contains marker names and the second column marker positions. Column names are not necessary.

Details

Both the first column of realmap and estmap$name must contain markername, but aside from this they do not have to have identical entries.

Value

position in estmap-position in realmap

---

get.nearest.informative

For a given marker finds the nearest neighbours with LOD scores > 0.

Description

Finds the nearest neighbours of a marker with LOD scores > 0.

Usage

```
get.nearest.informative(loci, lodmap)
```

Arguments

loci	Scalar indicating a marker number
lodmap	Symmetric matrix of pairwise LOD scores

Details

The columns and rows of the matrix should be in the order corresponding to the estimated map order. The function then returns the ranks of first markers to the left and right of the marker of interest with non-zero lod scores.

Value

A vector of length 1 or 2 containing the rank of the nearest informative markers.

invert.map *Invert the order of locimap from an estimated map.*

Description

Takes the locimap from estimate.maps a dataframe containing names and positions and any other information in increasing order of distance and inverts the order.

Usage

```
invert.map(locimap)
```

Arguments

locimap	a data frame containing the markers names and positions

Details

The map should be a data frame with a column called 'position'. It should have a starting marker a position zero. The function then inverts the distances from so that the marker at maximum distance from the starting marker (the end marker) is at distance 0 and the original starting marker is now at the maximum distance. It also inverts the order of the rows in the data frame. Thus if the markers were originally in order of increasing distance from the starting marker they will now be in order of increasing distance from the end marker.

Value

The original data frame in inverted order with the distances inverted so that the end marker is now the starting marker.

lgI.txt *Dataset lgI.txt: pairwise recombination fractions for 143 markers.*

Description

A dataset containing the pairwise recombinations fractions for 143 SNP markers from linkage group I of potato. These are derived from the genotypes of 190 offspring from a cross between potato cultivar Stirling and the breeding line 12601ab1. Further details are available in Hackett et al. (2013).

Format

An ascii text file in the format described in `calc.maps.pc` and `calc.maps.sphere`. The first line contains the number of markers and the number of combinations. Then follow the space-separated combinations with their recombination fractions and LOD scores:

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
1          2          .                        .
1          3          .                        .
1          4          .                        .
.          .          .                        .
.          .          .                        .
.          .          .                        .
2          3          .                        .
2          4          .                        .
.          .          .                        .
```

lgI.txt

NA

Source

Hackett, C.A., McLean, K. and Bryan, G.J. (2013). Linkage analysis and QTL mapping using SNP dosage data in a tetraploid potato mapping population. PLoS ONE 8, e63939

Examples

```
system.file("extdata", "lgI.txt", package="MDSMap")
```

`lgV.txt` *Dataset lgV.txt: pairwise recombination fractions for 238 markers.*

Description

A dataset containing the pairwise recombinations fractions for 238 SNP markers from linkage group V of potato. These are derived from the genotypes of 190 offspring from a cross between potato cultivar Stirling and the breeding line 12601ab1. Further details are available in Hackett et al. (2013).

Format

An ascii text file in the format described in `calc.maps.pc` and `calc.maps.sphere`. The first line contains the number of markers and the number of combinations. Then follow the space-separated combinations with their recombination fractions and LOD scores:

```
nmarkers
marker_1   marker_2   recombination fraction   LOD
1          2          .                        .
1          3          .                        .
1          4          .                        .
.          .          .                        .
.          .          .                        .
.          .          .                        .
2          3          .                        .
2          4          .                        .
.          .          .                        .
```

lgV.txt

NA

Source

Hackett, C.A., McLean, K. and Bryan, G.J. (2013). Linkage analysis and QTL mapping using SNP dosage data in a tetraploid potato mapping population. PLoS ONE 8, e63939

Examples

```
system.file("extdata", "lgV.txt", package="MDSMap")
```

`map.to.interval` *Map points from MDS final configuration to interval starting at 0.*

Description

Maps points from the final configuration of a 2-dimensional spherically constrained wMDS to an interval with a starting point at 0.

Usage

```
map.to.interval(mdsobject, nloci)
```

Arguments

mdsobject	The output from smacofSphere.
nloci	The number of markers in the configuration.

Details

Centres the configuration on zero and calculates the median distance of the points from the origin. Finds the largest gap in the spherical configuration and assigns the marker on the right hand side of it angle 0. Converts Cartesian coordinates to polar coordinates and projects points onto the arc centred on 0 with radius the median distance from the origin.

Value

A list with the elements:

chromlength	A named vector giving the position of each marker.
order	A named vector giving the rank order of the markers.
locilength	A named vector giving the position of each marker in order of increasing distance along the segment.
maporder	A named vector of the position in the input list of each marker in order of increasing distance along the segment.

See Also

convert.polar

Examples

```
# M and lod should be n x n symmetric matrices of the same dimensions where
# n is the number markers to be analysed
## Not run:
mds1<-smacofSphere(M,ndim=2,algorithm=dual,weightmat=lod,penalty=100)
pol<-map.to.interval (m1,n)

## End(Not run)
```

---

meandist.from.truth *Calculates mean square distance between marker positions in two different maps.*

Description

Calculates mean square distance of markers in the analysis from some objective "truth".

Usage

```
meandist.from.truth(estmap, realmap)
```

Arguments

estmap	Estimated map with 2 columns, name and position contain marker names and positions.
realmap	Map in which the first column contains marker names and the second contains marker positions. Column names are not necessary.

Details

The first column of realmap must contain identical entries to estmap$name. However, the order of entries can be different.

For every marker the difference between the position stated in estmap and in realmap is calculated (see get.dist.loci).

Every difference is squared and the mean of the square differences is returned.

Note that where different weights are used in estimating maps, it is valid to compare the mean distance from the truth. However, if different map functions are used then the distances are not comparable.

Therefore, if there is some knowledge of markers on a chromosome and data is simulated so that there is some objective knowledge of the truth then this function could be used to decide whether to use lod or lod2 weightings to estimate maps attempting to locate additional markers. However, it is not suitable for deciding on the map functions used to calculated the pairwise marker distances.

Value

A list with the following elements:

pointdist	Data frame containing marker names and the distance between the estimated position and the "real" position.
meansquaredist	mean square distance between the estimated real position of markers.

---

plot.pcmap *Diagnostic plots for the map estimation using calc.maps.pc with 2 dimensions.*

Description

Diagnostic plots for the map estimation using calc.maps.pc with 2 dimensions.

Usage

```
## S3 method for class 'pcmap'
plot(x, D1lim = NULL, D2lim = NULL, displaytext = TRUE, ...)
```

Arguments

x	Map object from calc.maps.pc() with 2 dimensions.
D1lim	Numeric vector specifying the limits of the horizontal axis.
D2lim	Numeric vector specifying the limits of the vertical axis.
displaytext	Logical argument determining how markers should be labelled in the wMDS configuration plot. If TRUE then marker names are used. If FALSE then numbers are used.
...	Further arguments are ignored. (accepted for compatibility with generic plot)

Details

Plots 2 panels:

Panel 1 the final MDS configuration and the fitted principal curve from the calc.maps.pc() in 2 dimensions. If D1lim or D2lim is not specified, then limits are defined by plot.smacof.

Panel 2 the pointwise nearest neighbour fits in order of the position in the estimated map.

Markers are assigned numbers according to the order in which they occur in the input file. The locikey output of the map object is a data frame associating marker names with their numbers. This can be accessed using pcmap$locikey. If displaytext=FALSE then markers will be labelled by these numbers. By default displaytext=TRUE and markers are labelled by marker name.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31: 1-30 https://www.jstatsoft.org/v31/i03/

See Also

plot.pcmap3d, plot.spheremap,plot.smacof, calc.maps.pc

Examples

```
map<-calc.maps.pc(system.file("extdata", "lgI.txt", package="MDSMap"),
ndim=2,weightfn='lod',mapfn='haldane')
plot(map)
```

---

plot.pcmap3d *Diagnostic plots for the map estimation using calc.maps.pc with 3 dimensions.*

Description

Diagnostic plots for the map estimation using calc.maps.pc with 3 dimensions.

Usage

```
## S3 method for class 'pcmap3d'
plot(
  x,
  D1lim = NULL,
  D2lim = NULL,
  D3lim = NULL,
  displaytext = TRUE,
  as2d = FALSE,
  ...
)
```

Arguments

x	Map object from calc.maps.pc() with 3 dimensions.
D1lim	Numeric vector specifying the limits of the axis relating to dimension 1 of the wMDS used to obtain pcmap3d.
D2lim	Numeric vector specifying the limits of the axis relating to dimension 2 of the wMDS used to obtain pcmap3d.
D3lim	Numeric vector specifying the limits of the axis relating to dimension 3 of the wMDS used to obtain pcmap3d.
displaytext	Logical argument determining how markers should be labelled in the wMDS configuration plot. If TRUE then marker names are used. If FALSE then numbers are used.
as2d	If TRUE: plot this pcmap3d as if it were a pcmap (2d).
...	Further arguments are ignored. (accepted for compatibility with generic plot)

Details

Plots 4 panels

Panels 1-3 show the final MDS configuration and the fitted principal curve from the `calc.maps.pc()` in 3 dimensions. plots `D1` vs `D2`, `D1` vs `D3` and `D2` vs `D3`. If `D1lim`, `D2lim` or `D3lim` is not specified, then limits are defined by `plot.smacof`.

Panel 4 shows the pointwise nearest neighbour fits in order of the position in the estimated map.

Also plots a 3 dimensional scatterplot of the final MDS configuration and the fitted principal curve in a new window using `plot3d` from the **rgl** package.

Markers are assigned numbers according to the order in which they occur in the input file. The locikey output of the map object is a data frame associating marker names with their numbers. This can be accessed using `pcmap3d$locikey`. If `displaytext=FALSE` then markers will be labelled by these numbers. By default `displaytext=TRUE` and markers are labelled by marker name.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31: 1-30 https://www.jstatsoft.org/v31/i03/

See Also

plot.pcmap, plot.spheremap,plot.smacof, calc.maps.pc, plot3d

Examples

```
map<-calc.maps.pc(system.file("extdata", "lgI.txt", package="MDSMap"),
ndim=3,weightfn='lod',mapfn='haldane')
plot(map)
```

---

plot.spheremap *Produces diagnostic plots for the estimated map using* calc.maps.sphere.

Description

Produces diagnostic plots for the estimated map using calc.maps.sphere.

Usage

```
## S3 method for class 'spheremap'
plot(x, displaytext = TRUE, ...)
```

Arguments

x	Map object from calc.maps.sphere
displaytext	Logical argument determining how markers should be labelled in the MDS configuration plot. If TRUE then marker names are used. If FALSE then numbers are used.
...	Further arguments are ignored. (accepted for compatibility with generic plot)

Details

Produces a figure with 3 panels from a map object produced by calc.maps.sphere.

Panel one shows the stress of the unconstrained MDS, the stress of the constrained MDS and the ratio of the two. A good rule of thumb is the stress from the constrained MDS should not be more than 10

Panel 2 shows the final configuration of the unconstrained MDS which can be used to identify outliers.

Panel 3 shows the final configuration of the constrained MDS in black and the unconstrained MDS in red. This can be used to check that the constrained fit is not distorting the data - large changes in the rank of a point in either dimension 1 or dimension 2 are indications of a problem with the fit.

Panel 4 shows the pointwise nearest neighbour fits in order of the position in the estimated map.

If D1lim or D2lim is not specified, then limits of panels 2 and 3 are defined by plot.smacof.

Markers are assigned numbers according to the order in which they occur in the input file. The locikey output of the map object is a data frame associating marker names with their numbers. This

can be accessed using pcmap3d$locikey. If displaytext=FALSE then in panels 2 and 3 markers will be labelled by these numbers. By default displaytext=TRUE and markers are labelled by marker name.

References

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31: 1-30 https://www.jstatsoft.org/v31/i03/

See Also

plot.pcmap, plot.pcmap3d, plot.smacof, calc.maps.sphere

Examples

```
map<-calc.maps.sphere(system.file("extdata", "lgI.txt", package="MDSMap"),
weightfn='lod', mapfn='kosambi')
plot(map)
```

---

recalc.nnfit.from.map *Calculate a nearest neighbour fit from an estimated map object.*

Description

Calculates a new nearest neighbour fit based on a new order from a map object generated by calc.maps.pc, calc.maps.sphere or estimate.map

Usage

```
recalc.nnfit.from.map(estmap, mapobject, header = TRUE)
```

Arguments

estmap	A character string indicating the name of a comma separated value file with the first column containing marker names in the order of their estimated position.
mapobject	A map object generated by calc.maps.pc, calc.maps.sphere or estimate.map.
header	Logical argument indicating whether the .csv file estmap contains headers - default is TRUE

Details

Reads in a new estimated order, reorders the distance map and LOD scores by the new order and recalculates the nearest neighbour fit.

Value

A list with the elements:

`fit`	Sum over all markers of the nearest neighbour fits.
`pointfits`	The nearest neighbour fit for each marker.
`meanfit`	Meanv of the nearest neighbour fits over all markers.

See Also

`calc.maps.pc`, `calc.maps.sphere`, `estimate.map`, `calc.nnfit`

`sim.bc.rflod.file` *Simulate a backcross population from homozygous parents.*

Description

Simulates a backcross population from homozygous parents and writes a file containing the number of markers and observed pairwise distances, the pairwise recombination fractions and LOD scores in a text file suitable for analysis by other functions in the package.

Usage

```
sim.bc.rflod.file(fname)
```

Arguments

`fname`	a character string specifying the base name of the file fname.txt to which the data should be written

Details

This function simply generates data for use with the vignette. The R/qtl package is used to simulate a backcross #'population of 200 individuals from homozygous parents with 200 markers in a single linkage group of length #'100cM. The recombination fractions and LOD scores are calculated. The data is written to a text file in the #'format of output from JoinMap 4. In particular, the data is cast into a data frame with marker names in the #'first two columns, pairwise recombination fractions in the third column and associated LOD scores in the fourth #'column. The data is written to a text file 'fname.txt' where the first row contains two entries - the number of #'markers and the number of pairwise observations. Below this the data frame containing the distance data is #'appended with no column headings.

```
nmarkers
marker_1   marker_2   recombination fraction   lod
1          2          .                        .
1          3          .                        .
```

1	4	.	.
.	.	.	.
.	.	.	.
.	.	.	.
2	3	.	.
2	4	.	.
.	.	.	.

Value

No output - just the text file as above

References

Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. Bioinformatics. 189: 889-890 Van Ooijen JW (2006) JoinMap 4; Software for the calculation of genetic linkage maps in experimental populations. Wageningen; Netherlands: Kyazma B.V

Index