Package ‘onemap’

February 20, 2015

Title Software for constructing genetic maps in experimental crosses: full-sib, RILs, F2 and backcrosses

Version 2.0-4

Date September, 2013

Depends tcltk, tkrplot, R (>= 2.12.0)

Suggests qtl

Description Analysis of molecular marker data from model (backcrosses, F2 and recombinant inbred lines) and non-model systems (i. e. outcrossing species). For the later, it allows statistical analysis by simultaneously estimating linkage and linkage phases (genetic map construction). All analysis are based on multipoint approaches using hidden Markov models.

License GPL-3

Author Gabriel Margarido [aut], Marcelo Mollinari [aut, cre], Karl Broman [ctb], Augusto Garcia [ctb]

Maintainer Marcelo Mollinari <mmollina@usp.br>

Repository CRAN

NeedsCompilation yes

Date/Publication 2013-09-09 20:03:53

R topics documented:

`onemap-package` *Software for constructing genetic maps*

Description

In its earlier version, the software implemented the methodology proposed by *Wu et al.* (2002a), which uses the maximum likelihood approach to estimate linkage and linkage phases for a mixed set of different marker types, to building genetic maps in outcrossing species. After, it was modified to also using a Hidden Markov Model approach for constructing multipoint maximum likelihood linkage maps (Wu et al. 2002b). It was then applied in several studies, such as Garcia et al. (2006), Oliveira et al. (2007) and Oliveira et al. (2008). Nowadays, the latest versions (2.0-3) was fully modified to also handle with backcrosses, F_2 and recombinant inbred lines (RIL) populations, allowing fully integration with software for QTL mapping, such as R/qtl and QTL Cartographer. OneMap can read and export files to this packages, and also from the widely used software MAPMAKER.

Details

Package:	onemap
Type:	Package
Version:	2.0-3
Date:	2012-12-06
License:	GNU GENERAL PUBLIC LICENSE (Version 3)

Usage of **onemap** is completely described in a tutorial distributed with the package. You can look for it in directory `/doc` of the package distribution.

The most important functions are:

- `read.outcross` for data importing.
- `read.mapmaker` for importing data from a MAPMAKER raw file.
- `rf.2pts` to perform two-point analyses between all pairs of markers.
- `marker.type` to check the segregation type of a marker.
- `make.seq` to define a sequence of markers, which are the input of most mapping functions.
- `group` to assign markers to linkage groups.
- `compare` to compare all possible orders of markers in a sequence.
- `try.seq` to try a given marker in every position of a framework of mapped markers.
- `order.seq` to automate the process of mapping a sequence of markers, combining compare and try.seq functions.
- `ripple.seq` to check a set of mapped markers, looking for (plausible) alternative orders.
- `map` to construct a map for a sequence in a given order.

Note

See the complete tutorial distributed along with this package for complete examples (located in directory /doc of the package distribution).

Author(s)

Gabriel R A Margarido, `<gramarga@gmail.com>`, Marcelo Mollinari, `<mmollina@usp.br>` and Antonio Augusto F Garcia, `<augusto.garcia@usp.br>`

References

Basten, C. J., Weir, B. S. and Zeng, Z.-B. (2005) *QTL Cartographer Version 1.17: A Reference Manual and Tutorial for QTL Mapping*.

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Buetow KH, Chakravarti A (1987). Multipoint gene mapping using seriation. I. General methods. *Am J Hum* Genet 41: 180-188.

Doerge, R. W. (1996) Constructing genetic maps by rapid chain delineation. *Journal of Quantitative Trait Loci* 2: 121-132.

Garcia, A. A. F., Kido, E. A., Meza, A. N., Souza, H. M. B., Pinto, L. R., Pastina, M. M., Leite, C. S., Silva, J. A. G., Ulian, E. C., Figueira, A. V. O. and Souza, A. P. (2006) Development of an integrated genetic map of a sugarcane (*Saccharum* spp.) commercial cross, based on a maximum-likelihood approach for estimation of linkage and linkage phases. *Theoretical and Applied Genetics* 112: 298-314.

Haldane, J. B. S. (1919) The combination of linkage values and the calculation of distance between the loci of linked factors. *Journal of Genetics* 8: 299-309.

Jiang, C. and Zeng, Z.-B. (1997). Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica* 101: 47-58.

Kosambi, D. D. (1944) The estimation of map distance from recombination values. *Annuaire of Eugenetics* 12: 172-175.

Lander, E. S. and Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA* 84: 2363-2367.

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

Lincoln, S. E., Daly, M. J. and Lander, E. S. (1993) Constructing genetic linkage maps with MAPMAKER/EXP Version 3.0: a tutorial and reference manual. *A Whitehead Institute for Biomedical Research Technical Report*.

Margarido, G. R. A., Souza, A.P. and Garcia, A. A. F. (2007) OneMap: software for genetic mapping in outcrossing species. *Hereditas* 144: 78-79.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetic maps. *Heredity* 103: 494-502

Oliveira, K. M., Pinto, L. R., Marconi, T. G., Margarido, G. R. A., Pastina, M. M., Teixeira, L. H. M., Figueira, A. V. O., Ulian, E. C., Garcia, A. A. F. and Souza, A. P. (2007) Functional genetic linkage map based on EST-markers for a sugarcane (*Saccharum* spp.) commercial cross. *Molecular Breeding* 20: 189-208.

Oliveira, E. J., Vieira, M. L. C., Garcia, A. A. F., Munhoz, C. F.,Margarido, G. R.A., Consoli, L., Matta, F. P., Moraes, M. C., Zucchi, M. I., and Fungaro,M. H. P. (2008) An Integrated Molecular Map of Yellow Passion Fruit Based on Simultaneous Maximum-likelihood Estimation of Linkage and Linkage Phases *J. Amer. Soc. Hort. Sci.* 133: 35-41.

Tan, Y., Fu, Y. (2006). A novel method for estimating linkage maps. *Genetics* 173: 2383-2390.

Van Os H, Stam P, Visser RGF, Van Eck HJ (2005). RECORD: a novel method for ordering loci on a genetic linkage map. *Theor Appl Genet* 112: 30-40.

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002a) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Wu, R., Ma, C.-X., Wu, S. S. and Zeng, Z.-B. (2002b). Linkage mapping of sex-specific differences. *Genetical Research* 79: 85-96

`add_drop` *Add or Drop Markers From a Sequence*

Description

Creates a new sequence by adding or dropping markers from a predetermined one. The markers are added in the end of the sequence.

Usage

```
add.marker(input.seq, mrks)
drop.marker(input.seq, mrks)
```

Arguments

`input.seq` an object of class `sequence`.

`mrks` a vector containing the markers to be added or removed from the `sequence`.

Value

An object of class `sequence`, which is a list containing the following components:

`seq.num` a `vector` containing the (ordered) indices of markers in the sequence, according to the input file.

`seq.phases` a `vector` with the linkage phases between markers in the sequence, in corresponding positions. `-1` means that there are no defined linkage phases.

`seq.rf` a `vector` with the recombination fractions between markers in the sequence. `-1` means that there are no estimated recombination fractions.

`seq.like` log-likelihood of the corresponding linkage map.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

Examples

```
data(example.out)
twopt <- rf.2pts(example.out)
all.mark <- make.seq(twopt,"all")
groups <- group(all.mark)
(LG1 <- make.seq(groups,1))
(LG.aug<-add.marker(LG1, c(4,7)))
(LG.red<-drop.marker(LG1, c(1,2,3,5,6)))
```

`compare` *Compare all possible orders (exhaustive search) for a given sequence of markers*

Description

For a given sequence with n markers, computes the multipoint likelihood of all $\frac{n!}{2}$ orders.

Usage

```
compare(input.seq,n.best=50,tol=10E-4,verbose=FALSE)
```

Arguments

`input.seq` an object of class `sequence`.

`n.best` the number of best orders to store in object (defaults to 50).

`tol` tolerance for the C routine, i.e., the value used to evaluate convergence.

`verbose` if `FALSE` (default), simplified output is displayed. if `TRUE`, detailed output is displayed.

Details

Since the number $\frac{n!}{2}$ is large even for moderate values of n, this function is to be used only for sequences with relatively few markers. If markers of types D1, D2 and C (specially in repulsion phase) are mixed in the sequence, this function can be used with up to 5 or 6 markers; otherwise, up to 10 markers will not take a very long time. The multipoint likelihood is calculated according to Wu et al. (2002b) (Eqs. 7a to 11), assuming that the recombination fraction is the same in both parents. Hidden Markov chain codes adapted from Broman et al. (2008) were used. For backcross, F_2 and RIL populations the linkage phases are known *a priori* and are not calculated.

Value

An object of class `compare`, which is a list containing the following components:

`best.ord` a `matrix` containing the best orders.

`best.ord.rf` a `matrix` with recombination frequencies for the corresponding best orders.

`best.ord.phase` a `matrix` with linkage phases for the best orders.

`best.ord.like` a `vector` with log-likelihood values for the best orders.

`best.ord.LOD` a `vector` with LOD Score values for the best orders.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Jiang, C. and Zeng, Z.-B. (1997). Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica* 101: 47-58.

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. _Heredity_ 103: 494-502.

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002a) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Wu, R., Ma, C.-X., Wu, S. S. and Zeng, Z.-B. (2002b). Linkage mapping of sex-specific differences. *Genetical Research* 79: 85-96

See Also

marker.type for details about segregation types and make.seq.

Examples

```
## Not run:
  #outcrossing example
  data(example.out)
  twopt <- rf.2pts(example.out)
  markers <- make.seq(twopt,c(12,14,15,26,28))
  (markers.comp <- compare(markers))
  (markers.comp <- compare(markers,verbose=TRUE))

  #F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  markers <- make.seq(twopt,c(17,26,29,30,44,46,55))
  (markers.comp <- compare(markers))
  (markers.comp <- compare(markers,verbose=TRUE))


## End(Not run)
```

def.rf.3pts *Three-point analysis of genetic markers*

Description

Due to the limitations of the method, this function is defunct and kept only for historical reasons. Performs the three-point analysis for outcrosses in the way proposed by *Wu et al. (2002)* for a triplet of markers in a given order.

Usage

```
def.rf.3pts(w, mrk1name = NULL, mrk2name = NULL, mrk3name = NULL, LOD =
5, max.rf = 0.35, max.nolink = 0.55)
```

Arguments

`w` an object of class `outcross`.

`mrk1name` a character string indicating the name of the first marker, corresponding to any marker in the input file stored in object `w`.

`mrk2name` a character string indicating the name of the second marker.

`mrk3name` a character string indicating the name of the third marker.

`LOD` minimum LOD Score to declare linkage (defaults to `5`).

`max.rf` maximum recombination fraction between *adjacent* markers to declare linkage (defaults to `0.35`).

`max.nolink` maximum recombination fraction between markers on the *edge* of the triplet to declare linkage (defaults to `0.55`).

Details

The three markers are analyzed in the order they are given as input, i.e., `mrk1name - mrk2name - mrk3name`.

Value

An object of class `rf.3pts`, which is a list containing the following components:

`LOD` minimum LOD Score to declare linkage.

`max.rf` maximum recombination fraction between *adjacent* markers to declare linkage.

`max.nolink` maximum recombination fraction between markers on the *edge* of the triplet to declare linkage.

`marnames` names of the three markers.

`recomb` a vector with the three-point estimates of recombination fraction between markers `mrk1name - mrk2name` and `mrk2name - mrk3name`, under the most probable assignment.

`phase` a character string indicating the most probable assignment (linkage phases) for the three markers.

`analysis` complete results of the three-point analysis for the triplet of markers.

`goodness` a vector with character strings indicating the “goodness” of each assignment, i.e., if the LOD Score and the estimates of recombination fraction are consistent with the criteria defined. Possible values are: `"****"` if the test is significant, the estimates are below the thresholds and the order seems to be right; `"*"` if the test is significant, but some estimates are above the thresholds and/or the order seems to be wrong; `"-"` if the test is not significant or all estimates are above the thresholds.

`flag` a number indicating if there is more than one equally probable assignment for the triplet of markers. Possible values are: `1` if positive, `0` if negative and `NA` if linkage is not significant.

Author(s)

Gabriel R A Margarido, `<gramarga@gmail.com>`

References

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Examples

```
## Not run:
  data(example.out)

  threepts <- def.rf.3pts(example.out,”M1”,”M2”,”M14”) # correct order
  threepts

  threepts <- def.rf.3pts(example.out,”M1”,”M14”,”M2”) # wrong order
  threepts

## End(Not run)
```

draw.map *Draw a genetic map*

Description

Provides a simple draw of a genetic map.

Usage

```
draw.map(map.list, horizontal=FALSE, names=FALSE, grid=FALSE, cex.mrk=1, cex.grp=.75)
```

Arguments

`map.list` a map, i.e. an object of class `sequence` with a predefined order, linkage phases, recombination fraction and likelihood; also it could be a list of maps.

`horizontal` if `TRUE`, indicates that the map should be plotted horizontally. Default is `FALSE`

`names` if `TRUE`, displays the names of the markers. Default is `FALSE`

`grid` if `TRUE`, displays a grid in the background. Default is `FALSE`

`cex.mrk` the magnification to be used for markers.

`cex.grp` the magnification to be used for group axis annotation.

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

Examples

```
## Not run:
 #outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  lg<-group(make.seq(twopt, "all"))
  maps<-vector("list", lg$n.groups)
  for(i in 1:lg$n.groups)
     maps[[i]]<- make.seq(order.seq(input.seq= make.seq(lg,i),twopt.alg =
   "rcd"), "force")
  draw.map(maps, grid=TRUE)
  draw.map(maps, grid=TRUE, horizontal=TRUE)

  #F2 example
  data(fake.f2.onemap)
  twopt<-rf.2pts(fake.f2.onemap)
  lg<-group(make.seq(twopt, "all"))
  maps<-vector("list", lg$n.groups)
  for(i in 1:lg$n.groups)
     maps[[i]]<- make.seq(order.seq(input.seq= make.seq(lg,i),twopt.alg =
   "rcd"), "force")
  draw.map(maps, grid=TRUE)
  draw.map(maps, grid=TRUE, horizontal=TRUE)


## End(Not run)
```

`example.out` *Data from a full-sib family derived from two outbred parents*

Description

Simulated data set for an outcross, i.e., an F1 population obtained by crossing two non-homozygous parents.

Usage

```
data(example.out)
```

Format

An object of class `outcross`.

Details

A total of 100 F1 individuals were genotyped for 30 markers. The data currently contains only genotype information (no phenotypes). It is included to be used as a reference in order to understand how a data file needs to be. Also, it is used for the analysis in the tutorial that comes with OneMap.

Author(s)

Gabriel R A Margarido, <gramarga@gmail.com>

See Also

read.outcross for details about objects of class outcross.

Examples

```
data(example.out)

# perform two-point analyses
twopts <- rf.2pts(example.out)
twopts
```

---

fake.bc.onemap *Simulated data from a backcross population*

Description

Simulated data set from a backcross population.

Usage

```
data(fake.bc.onemap)
```

Format

An object of class bc.onemap.

Details

A total of 150 individuals were genotyped for 67 markers with 15% of missing data. There is one quantitative phenotype to show how to use onemap output as R\qtl input.

Author(s)

Marcelo Mollinari, <mmollina@usp.br>

See Also

read.mapmaker for details about objects of class bc.onemap.

Examples

```
data(fake.bc.onemap)

# perform two-point analyses
twopts <- rf.2pts(fake.bc.onemap)
twopts
```

fake.f2.onemap *Simulated data from a F2 population*

Description

Simulated data set from a F2 population.

Usage

```
data(fake.f2.onemap)
```

Format

An object of class `f2.onemap`.

Details

A total of 200 individuals were genotyped for 66 markers (36 co-dominant, i.e. AA, AB or BB and 30 dominant i.e. Not AA or AA and Not BB or BB) with 15% of missing data. There is one quantitative phenotype to show how to use `onemap` output as `R\qtl` and `QTL Cartographer` input. Also, it is used for the analysis in the tutorial that comes with OneMap.

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

See Also

`read.mapmaker` for details about objects of class `f2.onemap`.

Examples

```
data(fake.f2.onemap)

# perform two-point analyses
twopts <- rf.2pts(fake.f2.onemap)
twopts
```

group *Assign markers to linkage groups*

Description

Identifies linkage groups of markers, using results from two-point (pairwise) analysis and the *transitive* property of linkage.

Usage

```
group(input.seq, LOD=NULL, max.rf=NULL)

## S3 method for class 'group'
print(x, detailed=TRUE, ...)
```

Arguments

`input.seq`	an object of class `sequence`.
`LOD`	a (positive) real number used as minimum LOD score (threshold) to declare linkage.
`max.rf`	a real number (usually smaller than 0.5) used as maximum recombination fraction to declare linkage.
`x`	an object of class `group`.
`detailed`	logical. If `FALSE`, only a small summary of the linkage groups is printed. If `TRUE` (default), the names of markers in each linkage group are also displayed.
`...`	further arguments, passed to other methods. Currently ignored.

Details

If the arguments specifying thresholds used to group markers, i.e., minimum LOD Score and maximum recombination fraction, are `NULL` (default), the values used are those contained in object `input.seq`. If not using `NULL`, the new values overridden the ones in object `input.seq`.

Value

Returns an object of class `group`, which is a list containing the following components:

`data.name`	name of the object of class `outcross` that contains the raw data.
`twopt`	name of the object of class `rf.2ts` used as input, i.e., containing information used to assign markers to linkage groups.
`marnames`	marker names, according to the input file.
`n.mar`	total number of markers.
`LOD`	minimum LOD Score to declare linkage.
`max.rf`	maximum recombination fraction to declare linkage.
`n.groups`	number of linkage groups found.
`groups`	number of the linkage group to which each marker is assigned.

Author(s)

Gabriel R A Margarido, <gramarga@gmail.com>

References

Lincoln, S. E., Daly, M. J. and Lander, E. S. (1993) Constructing genetic linkage maps with MAPMAKER/EXP Version 3.0: a tutorial and reference manual. *A Whitehead Institute for Biomedical Research Technical Report*.

See Also

rf.2pts and make.seq

Examples

```
data(example.out)
twopts <- rf.2pts(example.out)

all.data <- make.seq(twopts,"all")
link_gr <- group(all.data)
link_gr
```

---

make.seq *Create a sequence of markers*

Description

Makes a sequence of markers based on an object of another type.

Usage

```
make.seq(input.obj, arg=NULL, phase=NULL, twopt=NULL)
```

Arguments

input.obj an object of class rf.2pts, group, compare, try or order.

arg its value depends on the type of object input.obj. For an object rf.2pts, arg can be the string "all", resulting in a sequence with all markers in the raw data (generally done for grouping markers); otherwise, it must be a vector of integers specifying which markers comprise the sequence. For an object of class group, arg must be an integer specifying the group. For a compare object, arg is an integer indicating the corresponding order (arranged according to the likelihood); if NULL (default), the best order is taken. For an object of class try, arg must be an integer less than or equal to the length of the original sequence plus one; the sequence obtained will be that with the additional marker in the position indicated by arg. Finally, for an order object, arg is a string: "safe" means the order that contains only markers mapped with the provided threshold; "force" means the order with all markers.

phase its value is also dependent on the type of input.obj. For an rf.2pts object, phase can be a vector with user- defined linkage phases (its length is equal to the number of markers minus one); if NULL (default), other functions will try to find the best linkage phases. For example, if phase assumes the vector c(1,2,3,4), the sequence of linkage phases will be couple/couple, couple/repulsion, repulsion/couple and repulsion/repulsion for a sequence of five markers. If input.obj is of class compare or try, this argument indicates which combination of linkage phases should be chosen, for the particular order given by argument arg. In both cases, NULL (default) makes the best combination to be taken. If input.obj is of class group or order, this argument has no effect.

twopt a string indicating the name of the object which contains the two-point information. This does not have to be defined by the user: it is here for compatibility issues when calling make.seq from inside other functions.

Value

An object of class sequence, which is a list containing the following components:

seq.num a vector containing the (ordered) indices of markers in the sequence, according to the input file.

seq.phases a vector with the linkage phases between markers in the sequence, in corresponding positions. -1 means that there are no defined linkage phases.

seq.rf a vector with the recombination frequencies between markers in the sequence. -1 means that there are no estimated recombination frequencies.

seq.like log-likelihood of the corresponding linkage map.

data.name name of the object of class outcross with the raw data.

twopt name of the object of class rf.2pts with the 2-point analyses.

Author(s)

Gabriel Margarido, <gramarga@gmail.com>

References

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

See Also

compare, try.seq, order.seq and map.

Examples

```
## Not run:
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,"all")
  all.mark <- make.seq(twopt,1:30) # same as above, for this data set
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.ord <- order.seq(LG1)
  (LG1.final <- make.seq(LG1.ord)) # safe order
  (LG1.final.all <- make.seq(LG1.ord,"force")) # forced order

  markers <- make.seq(twopt,c(2,3,12,14))
  markers.comp <- compare(markers)
  (base.map <- make.seq(markers.comp))
  base.map <- make.seq(markers.comp,1,1) # same as above
  (extend.map <- try.seq(base.map,30))
  (base.map <- make.seq(extend.map,5)) # fifth position is the best

## End(Not run)
```

---

map *Construct the linkage map for a sequence of markers*

Description

Estimates the multipoint log-likelihood, linkage phases and recombination frequencies for a sequence of markers in a given order.

Usage

```
map(input.seq, tol)
```

Arguments

input.seq	an object of class sequence.
tol	tolerance for the C routine, i.e., the value used to evaluate convergence.

Details

Markers are mapped in the order defined in the object input.seq. If this object also contains a user-defined combination of linkage phases, recombination frequencies and log-likelihood are estimated for that particular case. Otherwise, the best linkage phase combination is also estimated. The multipoint likelihood is calculated according to Wu et al. (2002b)(Eqs. 7a to 11), assuming that the recombination fraction is the same in both parents. Hidden Markov chain codes adapted from Broman et al. (2008) were used.

Value

An object of class sequence, which is a list containing the following components:

seq.num	a vector containing the (ordered) indices of markers in the sequence, according to the input file.

`seq.phases`	a `vector` with the linkage phases between markers in the sequence, in corresponding positions. `-1` means that there are no defined linkage phases.
`seq.rf`	a `vector` with the recombination frequencies between markers in the sequence. `-1` means that there are no estimated recombination frequencies.
`seq.like`	log-likelihood of the corresponding linkage map.
`data.name`	name of the object of class `outcross` with the raw data.
`twopt`	name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Adapted from Karl Broman (package 'qtl') by Gabriel R A Margarido, `<gramarga@gmail.com>` and Marcelo Mollinari, `<mmollina@usp.br>`

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Jiang, C. and Zeng, Z.-B. (1997). Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica* 101: 47-58.

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002a) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Wu, R., Ma, C.-X., Wu, S. S. and Zeng, Z.-B. (2002b). Linkage mapping of sex-specific differences. *Genetical Research* 79: 85-96

See Also

`make.seq`

Examples

```
data(example.out)
twopt <- rf.2pts(example.out)

markers <- make.seq(twopt,c(30,12,3,14,2)) # correct phases
map(markers)

markers <- make.seq(twopt,c(30,12,3,14,2),phase=c(4,1,4,3)) # incorrect phases
map(markers)
```

map_func *Mapping functions Haldane and Kosambi*

Description

Functions to convert recombination fractions to distance in `cM` (centiMorgans).

Usage

```
haldane(rcmb)
kosambi(rcmb)
```

Arguments

`rcmb`	A recombination fraction between two markers, i.e., a number between `0` and `0.5`.

Details

Haldane mapping function is defined as

$$d_M = -\frac{1}{2}\ln(1-2r),$$

for $0 \leq r \leq 0.5$, where `r` stands for the recombination fraction in `rcmb`. Kosambi mapping function is

$$d_M = \frac{1}{4}\ln\left[\frac{1+2r}{1-2r}\right],$$

for $0 \leq r \leq 0.5$, where `r` is defined as above.

Value

Both functions return a number with a distance measured in `cM`.

Author(s)

Gabriel R A Margarido, <gramarga@gmail.com>

References

Haldane, J. B. S. (1919) The combination of linkage values and the calculation of distance between the loci of linked factors. *Journal of Genetics* 8: 299-309.

Kosambi, D. D. (1944) The estimation of map distance from recombination values. *Annuaire of Eugenetics* 12: 172-175.

Examples

```
# little difference for small recombination fractions
haldane(0.05)
kosambi(0.05)

# greater difference as recombination fraction increases
haldane(0.35)
kosambi(0.35)
```

`marker.type` *Informs the segregation patterns of markers*

Description

Informs the type of segregation of all markers from an object of class sequence. For outcross populations it uses the notation by *Wu et al., 2002*. For backcrosses, F_2 and RILs, it uses the traditional notation from MAPMAKER i.e. AA, AB, BB, not AA and not BB.

Usage

```
marker.type(input.seq)
```

Arguments

`input.seq`	an object of class `sequence`.

Details

The segregation types are (*Wu et al., 2002*):

Type	Cross	Segregation
A.1	ab x cd	1:1:1:1
A.2	ab x ac	1:1:1:1
A.3	ab x co	1:1:1:1
A.4	ao x bo	1:1:1:1
B1.5	ab x ao	1:2:1
B2.6	ao x ab	1:2:1
B3.7	ab x ab	1:2:1
C8	ao x ao	3:1
D1.9	ab x cc	1:1
D1.10	ab x aa	1:1
D1.11	ab x oo	1:1
D1.12	bo x aa	1:1
D1.13	ao x oo	1:1
D2.14	cc x ab	1:1
D2.15	aa x ab	1:1
D2.16	oo x ab	1:1
D2.17	aa x bo	1:1
D2.18	oo x ao	1:1

Value

Nothing is returned. Segregation types of all markers in the sequence are displayed on the screen.

Author(s)

Gabriel R A Margarido, <gramarga@gmail.com>

References

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

See Also

make.seq

Examples

```
data(example.out)
twopts <- rf.2pts(example.out)
markers.ex <- make.seq(twopts,c(3,6,8,12,16,25))
marker.type(markers.ex) # segregation type for some markers

data(fake.f2.onemap)
twopts <- rf.2pts(fake.f2.onemap)
all.mrk<-make.seq(twopts, "all")
lgs<-group(all.mrk)
lg1<-make.seq(lgs,1)
marker.type(lg1) # segregation type for linkage group 1
```

order.seq *Search for the best order of markers combining compare and try.seq functions*

Description

For a given sequence of markers, this function first uses the compare function to create a framework for a subset of informative markers. Then, it tries to map remaining ones using the try.seq function.

Usage

```
order.seq(input.seq, n.init=5, subset.search=c("twopt", "sample"),
          subset.n.try=30, subset.THRES=3,
          twopt.alg=c("rec", "rcd", "ser", "ug"),
          THRES=3, touchdown=FALSE, draw.try=FALSE,
          wait=0, tol=10E-2)
```

Arguments

input.seq an object of class sequence.

n.init the number of markers to be used in the compare step (defaults to 5).

subset.search a character string indicating which method should be used to search for a subset of informative markers for the compare step. It is used for backcross, F_2 or RIL populations, but not for outcrosses. See the Details section.

subset.n.try integer. The number of times to repeat the subset search procedure. It is only used if subset.search=="sample". See the Details section.

subset.THRES numerical. The threshold for the subset search procedure. It is only used if subset.search=="sample". See the Details section.

twopt.alg a character string indicating which two-point algorithm should be used if subset.search=="twopt". See the Details section.

THRES threshold to be used when positioning markers in the try.seq step.

touchdown logical. If FALSE (default), the try.seq step is run only once, with the value of THRES. If TRUE, try.seq runs with THRES and then once more, with THRES-1. The latter calculations take longer, but usually are able to map more markers.

draw.try if TRUE, a diagnostic graphic for each try.seq step is displayed. See Details section in try.seq function.

wait the minimum time interval in seconds to display the diagnostic graphic for each try.seq step. Defaults to 0.00

tol tolerance number for the C routine, i.e., the value used to evaluate convergence of the EM algorithm.

Details

For outcrossing populations, the initial subset and the order in which remaining markers will be used in the try.seq step is given by the degree of informativeness of markers (i.e markers of type A, B, C and D, in this order).

For backcrosses, F_2 or RILs, two methods can be used for choosing the initial subset: i) "sample" randomly chooses a number of markers, indicated by n.init, and calculates the multipoint log-likelihood of the $\frac{n.init!}{2}$ possible orders. If the LOD Score of the second best order is greater than subset.THRES, than it takes the best order to proceed with the try.seq step. If not, the procedure is repeated. The maximum number of times to repeat this procedure is given by the subset.n.try argument. ii) "twopt" uses a two-point based algorithm, given by the option "twopt.alg", to construct a two-point based map. The options are "rec" for RECORD algorithm, "rcd" for Rapid Chain Delineation, "ser" for Seriation, and "ug" for Unidirectional Growth. Then, equally spaced markers are taken from this map. The "compare" step will then be applied on this subset of markers.

In both cases, the order in which the other markers will be used in the try.seq step is given by marker types (i.e. co-dominant before dominant) and by the missing information on each marker.

After running the compare and try.seq steps, which result in a "safe" order, markers that could not be mapped are "forced" into the map, resulting in a map with all markers positioned.

Value

An object of class `order`, which is a list containing the following components:

`ord` an object of class `sequence` containing the "safe" order.

`mrk.unpos` a `vector` with unpositioned markers (if they exist).

`LOD.unpos` a `matrix` with LOD-Scores for unmapped markers, if any, for each position in the "safe" order.

`THRES` the same as the input value, just for printing.

`ord.all` an object of class `sequence` containing the "forced" order, i.e., the best order with all markers.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Gabriel R A Margarido, `<gramarga@gmail.com>` and Marcelo Mollinari, `<mmollina@usp.br>`

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Jiang, C. and Zeng, Z.-B. (1997). Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica* 101: 47-58.

Lander, E. S. and Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA* 84: 2363-2367.

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. *Heredity* 103: 494-502.

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002a) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Wu, R., Ma, C.-X., Wu, S. S. and Zeng, Z.-B. (2002b). Linkage mapping of sex-specific differences. *Genetical Research* 79: 85-96

See Also

`make.seq`, `compare` and `try.seq`.

Examples

```
## Not run:
  #outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG2 <- make.seq(groups,2)
  LG2.ord <- order.seq(LG2,touchdown=TRUE)
  LG2.ord
  make.seq(LG2.ord) # get safe sequence
  make.seq(LG2.ord,"force") # get forced sequence

  #F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG3 <- make.seq(groups,3)
  LG3.ord <- order.seq(LG3, subset.search = "twopt", twopt.alg = "rcd", touchdown=TRUE)
  LG3.ord
  make.seq(LG3.ord) # get safe sequence
  ord.1<-make.seq(LG3.ord,"force") # get forced sequence

  LG3.ord.s <- order.seq(LG3, subset.search = "sample", touchdown=TRUE)
  LG3.ord.s
  make.seq(LG3.ord) # get safe sequence
  ord.2<-make.seq(LG3.ord,"force") # get forced sequence

  rbind(ord.1$seq.num, ord.2$seq.num) # probably, the same order for
  this dataset

  #Now showing diagnostic graphics for each try.seq step.
  LG3.ord.dg <- order.seq(LG3, subset.search = "sample", touchdown=TRUE,
                          draw.try=TRUE, wait=3)


## End(Not run)
```

rcd *Rapid Chain Delineation*

Description

Implements the marker ordering algorithm *Rapid Chain Delineation* (*Doerge, 1996*).

Usage

```
rcd(input.seq, LOD=0, max.rf=0.5, tol=10E-5)
```

Arguments

`input.seq` an object of class `sequence`.

`LOD` minimum LOD-Score threshold used when constructing the pairwise recombination fraction matrix.

`max.rf` maximum recombination fraction threshold used as the LOD value above.

`tol` tolerance for the C routine, i.e., the value used to evaluate convergence.

Details

Rapid Chain Delineation (*RCD*) is an algorithm for marker ordering in linkage groups. It is not an exhaustive search method and, therefore, is not computationally intensive. However, it does not guarantee that the best order is always found. The only requirement is a matrix with recombination fractions between markers. Next is an excerpt from QTL Cartographer Version 1.17 Manual describing the *RCD* algorithm (*Basten et al., 2005*):

The linkage group is initiated with the pair of markers having the smallest recombination fraction. The remaining markers are placed in a "pool" awaiting placement on the map. The linkage group is extended by adding markers from the pool of unlinked markers. Each terminal marker of the linkage group is a candidate for extension of the chain: The unlinked marker that has the smallest recombination fraction with either is added to the chain subject to the provision that the recombination fraction is statistically significant at a prespecified level. This process is repeated as long as markers can be added to the chain.

After determining the order with *RCD*, the final map is constructed using the multipoint approach (function `map`).

Value

An object of class `sequence`, which is a list containing the following components:

`seq.num` a `vector` containing the (ordered) indices of markers in the sequence, according to the input file.

`seq.phases` a `vector` with the linkage phases between markers in the sequence, in corresponding positions. `-1` means that there are no defined linkage phases.

`seq.rf` a `vector` with the recombination frequencies between markers in the sequence. `-1` means that there are no estimated recombination frequencies.

`seq.like` log-likelihood of the corresponding linkage map.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Gabriel R A Margarido, <`gramarga@gmail.com`>

References

Basten, C. J., Weir, B. S. and Zeng, Z.-B. (2005) *QTL Cartographer Version 1.17: A Reference Manual and Tutorial for QTL Mapping*.

Doerge, R. W. (1996) Constructing genetic maps by rapid chain delineation. *Journal of Quantitative Trait Loci* 2: 121-132.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. *Heredity* 103: 494-502.

See Also

make.seq, map

Examples

```
## Not run:
  #outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.rcd <- rcd(LG1)

  #F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.rcd <- rcd(LG1)
  LG1.rcd

## End(Not run)
```

---

read.mapmaker *Read data from a Mapmaker raw file*

Description

Imports data from a Mapmaker raw file.

Usage

```
read.mapmaker(dir, file)
```

Arguments

dir	directory where the input file is located.
file	the name of the input file which contains the data to be read.

Details

For details about MAPMAKER files see *Lincoln et al.* (1993). The current version supports backcross, F_2 and RIL populations. The file can contain phenotypic data, but it will not be used in the analysis.

Value

An object of class `bc.onemap`, `f2.onemap`, `riself.onemap` or `risib.onemap` i.e., a list with the following components:

`geno` a matrix with integers indicating the genotypes read for each marker in `onemap` fashion. Each column contains data for a marker and each row represents an individual.

`geno.mmk` a list containing the type of cross and a matrix with integers indicating the genotypes read for each marker in `MAPMAKER/EXP` fashion, i.e., 1, 2, 3: AA, AB, BB, respectively; 3, 4: BB, not BB, respectively; 1, 5: AA, not AA, respectively. Each column contains data for a marker and each row represents an individual.

`n.ind` number of individuals.

`n.mar` number of markers.

`segr.type` a vector with the segregation type of each marker, as `strings`. Segregation types were adapted from outcross segregation types, using the same notation. For details see read.outcross.

`segr.type.num` a vector with the segregation type of each marker, represented in a simplified manner as integers. Segregation types were adapted from outcross segregation types. For details see read.outcross.

`phase` a numeric vector containing the linkage phase information between markers, i.e., 1 for coupling and -1 for repulsion, which is trivial for backcrosses, F_2 and RILs.

`input` the name of the input file.

`n.phe` number of phenotypes.

`pheno` a matrix with phenotypic values. Each column contains data for a trait and each row represents an individual. Currently ignored.

Author(s)

Adapted from Karl Broman (package **qtl**) by Marcelo Mollinari, `<mmollina@usp.br>`

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Lincoln, S. E., Daly, M. J. and Lander, E. S. (1993) Constructing genetic linkage maps with MAPMAKER/EXP Version 3.0: a tutorial and reference manual. *A Whitehead Institute for Biomedical Research Technical Report*.

See Also

`fake.bc.onemap` and `fake.f2.onemap` directory in the package source.

Examples

```
  ## Not run: 
    map_data <-read.mapmaker(dir="work_directory",file="data_file.txt")
    #Checking 'fake.f2.onemap'
    data(fake.f2.onemap)
    names(fake.f2.onemap)

## End(Not run)
```

`read.outcross` *Read data from a segregating full-sib population*

Description

Imports data from a full-sib family derived from the cross of two outbred parents and creates an object of class `outcross`.

Usage

```
read.outcross(dir, file)
```

Arguments

`dir`	directory where the input file is located.
`file`	the name of the input file which contains the data to be read.

Details

The file format is quite similar to that used by `MAPMAKER/EXP` (*Lincoln et al.*, 1993). The first line contains two integers: the number of individuals and the number of markers.

Next comes the genotype data for all markers. Each new marker is initiated with a "*" (without the quotes) followed by the marker name, without any space between them. Each marker name is followed by the corresponding segregation type, which may be: `"A.1"`, `"A.2"`, `"A.3"`, `"A.4"`, `"B1.5"`, `"B2.6"`, `"B3.7"`, `"C.8"`, `"D1.9"`, `"D1.10"`, `"D1.11"`, `"D1.12"`, `"D1.13"`, `"D2.14"`, `"D2.15"`, `"D2.16"`, `"D2.17"` or `"D2.18"` (without quotes) [see `marker.type` and *Wu et al.* (2002) for details].

Finally, after the segregation type comes the genotype data for the corresponding marker. Depending on the segregation type, genotypes may be denoted by `ac`, `ad`, `bc`, `bd`, `a`, `ba`, `b`, `bc`, `ab` and `o`, in several possible combinations. To make things easier, we have followed **exactly** the notation used by *Wu et al.* (2002). Genotypes *must* be separated by commas. Missing values are denoted by `"-"`

The `example` directory in the package distribution contains an example data file to be read with this function. Further instructions can be found at the tutorial distributed along with this package.

Value

An object of class `outcross`, i.e., a list with the following components:

`geno` a matrix with integers indicating the genotypes read for each marker. Each column contains data for a marker and each row represents an individual.

`n.ind` number of individuals.

`n.mar` number of markers.

`segr.type` a vector with the segregation type of each marker, as `strings`.

`segr.type.num` a vector with the segregation type of each marker, represented in a simplified manner as integers, i.e. 1 corresponds to markers of type `"A"`; 2 corresponds to markers of type `"B1.5"`; 3 corresponds to markers of type `"B2.6"`; 4 corresponds to markers of type `"B3.7"`; 5 corresponds to markers of type `"C.8"`; 6 corresponds to markers of type `"D1"` and 7 corresponds to markers of type `"D2"`

`input` the name of the input file.

Author(s)

Adapted from Karl Broman (package **qtl**) by Gabriel R A Margarido, `<gramarga@gmail.com>`

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Lincoln, S. E., Daly, M. J. and Lander, E. S. (1993) Constructing genetic linkage maps with MAPMAKER/EXP Version 3.0: a tutorial and reference manual. *A Whitehead Institute for Biomedical Research Technical Report*.

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

See Also

`example` directory in the package source.

Examples

```
  ## Not run:
    outcr_data <-
read.outcross(dir="work_directory",file="data_file.txt")

## End(Not run)
```

record *Recombination Counting and Ordering*

Description

Implements the marker ordering algorithm *Recombination Counting and Ordering* (*Van Os et al., 2005*).

Usage

```
record(input.seq, times=10, LOD=0, max.rf=0.5, tol=10E-5)
```

Arguments

`input.seq` an object of class `sequence`.

`times` integer. Number of replicates of the RECORD procedure.

`LOD` minimum LOD-Score threshold used when constructing the pairwise recombination fraction matrix.

`max.rf` maximum recombination fraction threshold used as the LOD value above.

`tol` tolerance for the C routine, i.e., the value used to evaluate convergence.

Details

Recombination Counting and Ordering (*RECORD*) is an algorithm for marker ordering in linkage groups. It is not an exhaustive search method and, therefore, is not computationally intensive. However, it does not guarantee that the best order is always found. The only requirement is a matrix with recombination fractions between markers. Next is an adapted excerpt from *Mollinari et al (2009)* describing the *RECORD* algorithm:

Based on the expected number of recombination events, an S *matrix is constructed,* $S = [S_{M_iM_j}]_{m\times m}$ *(for* $M_i = M_j$*,* $S_{M_iM_j} = 0$*), where* m *is the number of markers. The procedure to obtain* S *is based on the expected number of crossovers between marker pairs, conditioned by the observation of the markers' phenotype. The optimization criterion* COUNT *for a sequence of* m *markers may be calculated by* $COUNT = \sum_{i=1}^{m-1} S_{M_iM_{i+1}}$*, where smaller* COUNT *values correspond to better orders. Map building is carried out by randomly taking two markers and positioning a third one at the beginning, at the end and between them. The marker is fixed at the position that gives a smaller value of* COUNT*. Similarly, the remaining markers are positioned at pre-established orders until completion of the map. Subsequently, a search for smaller values of* COUNT *is performed, inverting the position on the map of subsequences of size* $m' = 2, \ldots, 20$*. If the map resulting from the inverted positions presents a* COUNT *value smaller than the previous one, it is kept. The procedure is repeated* `times` *times and the sequence presenting the smallest* COUNT *value is chosen.*

After determining the order with *RECORD*, the final map is constructed using the multipoint approach (function `map`).

Value

An object of class `sequence`, which is a list containing the following components:

`seq.num` a `vector` containing the (ordered) indices of markers in the sequence, according to the input file.

`seq.phases` a `vector` with the linkage phases between markers in the sequence, in corresponding positions. `-1` means that there are no defined linkage phases.

`seq.rf` a `vector` with the recombination frequencies between markers in the sequence. `-1` means that there are no estimated recombination frequencies.

`seq.like` log-likelihood of the corresponding linkage map.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

References

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. *Heredity* 103: 494-502.

Van Os, H., Stam, P., Visser, R.G.F. and Van Eck, H.J. (2005) RECORD: a novel method for ordering loci on a genetic linkage map. *Theoretical and Applied Genetics* 112: 30-40.

See Also

`make.seq` and `map`

Examples

```
## Not run:
  ##outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.rec <- record(LG1)

  ##F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.rec <- record(LG1)
  LG1.rec

## End(Not run)
```

rf.2pts *Two-point analysis between genetic markers*

Description

Performs the two-point (pairwise) analysis proposed by *Wu et al. (2002)* between all pairs of markers.

Usage

```
rf.2pts(input.obj, LOD = 3, max.rf = 0.50, verbose = TRUE)

## S3 method for class 'rf.2pts'
print(x, mrk1 = NULL, mrk2 = NULL, ...)
```

Arguments

`input.obj` an object of class `outcross`, `bc.onemap`, `f2.onemap`, `riself.onemap` or `risib.onemap`.

`LOD` minimum LOD Score to declare linkage (defaults to `3`).

`max.rf` maximum recombination fraction to declare linkage (defaults to `0.50`).

`verbose` logical. If `TRUE`, current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `rf.2pts`.

`mrk1, mrk2` optionally, two markers can be specified. If so, detailed results of the two-point analysis will be printed for this pair. Both arguments can be numeric or character strings indicating the numbers/names corresponding to any markers in the input file.

`...` further arguments, passed to other methods. Currently ignored.

Details

For `n` markers, there are

$$\frac{n(n-1)}{2}$$

pairs of markers to be analyzed. Therefore, completion of the two-point analyses can take a long time.

Value

An object of class `rf.2pts`, which is a list containing the following components:

`data.name` name of the object with the raw data.

`n.mar` total number of markers.

`marnames` marker names, according to the input file.

`LOD` minimum LOD Score to declare linkage.

`max.rf` maximum recombination fraction to declare linkage.

`input` the name of the input file.

`analysis` an array with the complete results of the two-point analysis for each pair of markers.

Note

The thresholds used for `LOD` and `max.rf` will be used in subsequent analyses, but can be overriden.

Author(s)

Gabriel R A Margarido, `<gramarga@gmail.com>`

References

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Examples

```
data(example.out)

twopts <- rf.2pts(example.out,LOD=3,max.rf=0.5) # perform two-point analyses
twopts

print(twopts,"M1","M2") # detailed results for markers 1 and 2
```

`rf.graph.table` *Plots pairwise recombination fractions and LOD Scores using a color scale.*

Description

Plots a matrix of pairwise recombination fractions (under the diagonal) and LOD Scores (upper the diagonal) using a color scale. Any value of the matrix can be easily accessed using an interactive Tcl-Tk interface, helping the user to check for possible problems.

Usage

```
rf.graph.table(input.seq, scale=1, axis.cex=1, main, inter=TRUE)
```

Arguments

`input.seq` an object of class `sequence` with a predefined order.

`scale` controls the plot size. If `inter == FALSE` this value is not used.

`axis.cex` the magnification to be used for axis annotation.

`main` the title for no interactive plot, i.e. it is only used if `inter == FALSE`.

`inter` logical. If `TRUE`, an interactive graphic is plotted. Otherwise, a default graphic device is used.

Details

The color scale varies from red (small distances or big LODs) to dark blue. When clicking on a cell, a dialog box is displayed with some information about corresponding markers for that cell (line × column). The informations are: i) the name of the markers; ii) the number of the markers on the data set; iii) the segregation types; iv) the recombination fraction between the markers and v) the LOD-Score for each possible linkage phase calculated via two-point analysis. For neighbor markers, the multipoint recombination fraction is printed; otherwise, the two-point recombination fraction is printed. For markers of type `D1` and `D2`, it's impossible to calculate recombination fraction via two-point analysis and, therefore the corresponding cell will be empty. For cells on the diagonal of the matrix, the name, the number and the type of the marker are printed, as well as the percentage of missing data for that marker.

Author(s)

Marcelo Mollinari, <mmollina@usp.br>

Examples

```
##outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.rcd <- rcd(LG1)
  rf.graph.table(LG1.rcd, inter=FALSE)
## Not run:
  ##Now, using interactive Tcl-Tk
  rf.graph.table(LG1.rcd, scale=1.5, inter=TRUE)

  ##F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)

  ##"pre-allocate" an empty list of length groups$n.groups (3, in this case)
  maps.list<-vector("list", groups$n.groups)

  for(i in 1:groups$n.groups){
    ##create linkage group i
    LG.cur <- make.seq(groups,i)
    ##ordering
    map.cur<-order.seq(LG.cur, subset.search = "sample")
    ##assign the map of the i-th group to the maps.list
    maps.list[[i]]<-make.seq(map.cur, "force")
  }
  ##Plot LOD/recombination fraction matrices for each group
  op <- par(mfrow = c(1, 3))
  for(i in 1:groups$n.groups)
    rf.graph.table(maps.list[[i]], axis.cex=.7, main=paste("Group", i),inter=FALSE)
  par(op)

## End(Not run)
```

`ripple.seq` *Compares and displays plausible alternative orders for a given linkage group*

Description

For a given sequence of ordered markers, computes the multipoint likelihood of alternative orders, by shuffling subsets (windows) of markers within the sequence. For each position of the window, all possible $(ws)!$ orders are compared.

Usage

```
ripple.seq(input.seq,ws=4,LOD=3,tol=10E-2)
```

Arguments

`input.seq`	an object of class `sequence` with a predefined order.
`ws`	an integer specifying the length of the window size (defaults to 4).
`LOD`	threshold for the LOD-Score, so that alternative orders with LOD less then or equal to this threshold will be displayed.
`tol`	tolerance for the C routine, i.e., the value used to evaluate convergence.

Details

Large values for the window size make computations very slow, specially if there are many partially informative markers.

Value

This function does not return any value; it just produces text output to suggest alternative orders.

Author(s)

Gabriel R A Margarido, `<gramarga@gmail.com>`

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Jiang, C. and Zeng, Z.-B. (1997). Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica* 101: 47-58.

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. *Heredity* 103: 494-502.

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002a) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Wu, R., Ma, C.-X., Wu, S. S. and Zeng, Z.-B. (2002b). Linkage mapping of sex-specific differences. *Genetical Research* 79: 85-96

See Also

`make.seq`, `compare`, `try.seq` and `order.seq`.

Examples

```
## Not run:
  data(example.out)
  twopt <- rf.2pts(example.out)

  markers <- make.seq(twopt,c(27,16,20,4,19,21,23,9,24,29))
  markers.map <- map(markers)
  ripple.seq(markers.map)

## End(Not run)
```

`seriation` *Seriation*

Description

Implements the marker ordering algorithm *Seriation* (*Buetow & Chakravarti, 1987*).

Usage

```
seriation(input.seq, LOD = 0, max.rf = 0.5, tol=10E-5)
```

Arguments

`input.seq` an object of class `sequence`.

`LOD` minimum LOD-Score threshold used when constructing the pairwise recombination fraction matrix.

`max.rf` maximum recombination fraction threshold used as the LOD value above.

`tol` tolerance for the C routine, i.e., the value used to evaluate convergence.

Details

Seriation is an algorithm for marker ordering in linkage groups. It is not an exhaustive search method and, therefore, is not computationally intensive. However, it does not guarantee that the best order is always found. The only requirement is a matrix with recombination fractions between markers. Next is an adapted excerpt from *Mollinari et al (2009)* describing the *Seriation* algorithm:

The map is initiated with each of the m markers and the matrix R(recombination fraction matrix). Considering M_i as the initial marker, M_j is positioned to the right if M_i if the recombination fraction between them is the smallest fraction between M_i and the other $m-1$ markers. From the remaining $m-1$ markers, M_k is chosen if it has the smallest recombination fraction with M_i. The recombination fractions of M_k and both external loci of the positioned markers, M_{left} (the most external marker to the left) and M_{right} (the most external marker to the right) are compared. If $\hat{r}_{M_kM_{right}} > \hat{r}_{M_kM_{left}}$, M_k is positioned to the left of the group of markers, and if the relationship is inverse, to the right. In the case of ties, the internal loci of the group already positioned are considered. The procedure is repeated until all markers are positioned, therefore providing m orders (one for each marker at the initial position). For each order, the continuity index is calculated as $CI = \sum_{i<j} \frac{\hat{r}_{M_iM_j}}{(i-j)^2}$. The best order is considered the one that gives the smallest CI value.

NOTE: When there are to many pairs of markers with the same value in the recombination fraction matrix, it can result in ties during the ordination process and the *Seriation* algorithm may not work properly. This is particularly relevant for outcrossing populations with mixture of markers of type `D1` and `D2`. When this occurs, the function shows the following error message: `There are too many ties in the ordination process - please, consider using another ordering algorithm`.

After determining the order with *Seriation*, the final map is constructed using the multipoint approach (function `map`).

Value

An object of class `sequence`, which is a list containing the following components:

`seq.num` a `vector` containing the (ordered) indices of markers in the sequence, according to the input file.

`seq.phases` a `vector` with the linkage phases between markers in the sequence, in corresponding positions. `-1` means that there are no defined linkage phases.

`seq.rf` a `vector` with the recombination frequencies between markers in the sequence. `-1` means that there are no estimated recombination frequencies.

`seq.like` log-likelihood of the corresponding linkage map.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Gabriel R A Margarido, <gramarga@gmail.com>

References

Buetow, K. H. and Chakravarti, A. (1987) Multipoint gene mapping using seriation. I. General methods. *American Journal of Human Genetics* 41: 180-188.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. *Heredity* 103: 494-502.

See Also

make.seq, map

Examples

```
## Not run:
  ##outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG3 <- make.seq(groups,3)
  LG3.ser <- seriation(LG3)

  ##F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.ser <- seriation(LG1)
  LG1.ser

## End(Not run)
```

set.map.fun *Defines the default mapping function*

Description

Defines the function that should be used to display the genetic map through the analysis.

Usage

```
set.map.fun(type=c("kosambi", "haldane"))
```

Arguments

`type`	Indicates the function that should be used, which can be `"kosambi"` or `"haldane"`

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

References

Haldane, J. B. S. (1919) The combination of linkage values and the calculation of distance between the loci of linked factors. *Journal of Genetics* 8: 299-309.

Kosambi, D. D. (1944) The estimation of map distance from recombination values. *Annuaire of Eugenetics* 12: 172-175.

See Also

`kosambi` and `haldane`

try.seq *Try to map a marker into every possible position between markers*

Description

For a given linkage map and an additional unpositioned marker, this function estimates parameters for all possible maps including the new marker, while keeping the original linkage map unaltered.

Usage

```
try.seq(input.seq, mrk, tol=10E-2, draw.try=FALSE, pos=NULL, verbose=FALSE)

## S3 method for class 'try'
print(x,j=NULL, ...)
```

Arguments

`input.seq`	an object of class `sequence` with a predefined order.
`mrk`	the index of the marker to be tried, according to the input file.
`tol`	tolerance for the C routine, i.e., the value used to evaluate convergence.
`draw.try`	if `TRUE`, a diagnostic graphic is displayed. See `Details` section.
`pos`	defines in which position the new marker `mrk` should be placed for the diagnostic graphic. If `NULL` (default), the marker is placed on the best position i.e. the one which results LOD = 0.00
`verbose`	if `FALSE` (default), simplified output is displayed. if `TRUE`, detailed output is displayed.
`x`	an object of class `try`.

j — if NULL (default), output is a summary of the results for all possible positions of the additional marker. Otherwise, an integer makes detailed output to be printed for the corresponding position. This integer must be less than or equal to the length of the original sequence plus 1.

... — further arguments, passed to other methods. Currently ignored.

Details

The diagnostic graphic is made of three figures: i) the top figure represents the new genetic map obtained with the insertion of the new marker mrk on position pos. If pos = NULL (default), the marker is placed on the best position i.e. the one which results LOD = 0.00, which is indicated by a red triangle; ii) the left bottom figure represents the base map (contained in input.seq) on x-axis and the LOD-Scores of the linkage maps obtained with the new marker mrk tested at the beginning, between and at the end of the base map. Actually, it is a graphic representation of the LOD vector (see Value section). The red triangle indicates the best position where the new marker mrk should be placed; iii) the right bottom figure is the non-interactive rf.graph.table function for the new genetic map. It plots a matrix of pairwise recombination fractions (under the diagonal) and LOD Scores (upper the diagonal) using a color scale.

Value

An object of class try, which is a list containing the following components:

ord — a list containing results for every linkage map estimated. These results include linkage phases, recombination frequencies and log-likelihoods.

LOD — a vector with LOD-Scores for each position where the additional marker is placed. This Score is based on the best combination of linkage phases for each map.

try.ord — a matrix with the orders of all linkage maps.

data.name — name of the object of class outcross with the raw data.

twopt — name of the object of class rf.2pts with the 2-point analyses.

Author(s)

Marcelo Mollinari, <mmollina@usp.br>

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Jiang, C. and Zeng, Z.-B. (1997). Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. *Genetica* 101: 47-58.

Lander, E. S., Green, P., Abrahamson, J., Barlow, A., Daly, M. J., Lincoln, S. E. and Newburg, L. (1987) MAPMAKER: An interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1: 174-181.

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetic maps. *Heredity* 103: 494-502

Wu, R., Ma, C.-X., Painter, I. and Zeng, Z.-B. (2002a) Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical Population Biology* 61: 349-363.

Wu, R., Ma, C.-X., Wu, S. S. and Zeng, Z.-B. (2002b). Linkage mapping of sex-specific differences. *Genetical Research* 79: 85-96

See Also

make.seq and compare.

Examples

```
## Not run:
  #outcrossing example
  data(example.out)
  twopt <- rf.2pts(example.out)
  markers <- make.seq(twopt,c(2,3,12,14))
  markers.comp <- compare(markers)
  base.map <- make.seq(markers.comp,1)

  extend.map <- try.seq(base.map,30)
  extend.map
  print(extend.map,5) # best position
  print(extend.map,4) # second best position

  #F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,"all")
  groups <- group(all.mark)
  LG3 <- make.seq(groups,3)
  LG3.ord <- order.seq(LG3, subset.search = "twopt", twopt.alg = "rcd", touchdown=TRUE)
  LG3.ord
  safe.map<-make.seq(LG3.ord,"safe")
  extend.map <- try.seq(safe.map,64)
  extend.map
  (new.map<-make.seq(extend.map,14)) # best position

  #Display diagnostic graphics
  try.seq(safe.map,64,draw.try=TRUE) #best position (default)
  try.seq(safe.map,64,draw.try=TRUE,pos=13) #second best position
  try.seq(safe.map,64,draw.try=TRUE,pos=4) #wrong position

  #Trying to position an unliked marker
  try.seq(safe.map,66,draw.try=TRUE) #note the inconsistencies in the graphic


## End(Not run)
```

ug *Unidirectional Growth*

Description

Implements the marker ordering algorithm *Unidirectional Growth* (*Tan & Fu, 2006*).

Usage

```
ug(input.seq, LOD=0, max.rf=0.5, tol=10E-5)
```

Arguments

`input.seq`	an object of class `sequence`.
`LOD`	minimum LOD-Score threshold used when constructing the pairwise recombination fraction matrix.
`max.rf`	maximum recombination fraction threshold used as the LOD value above.
`tol`	tolerance for the C routine, i.e., the value used to evaluate convergence.

Details

Unidirectional Growth (*UG*) is an algorithm for marker ordering in linkage groups. It is not an exhaustive search method and, therefore, is not computationally intensive. However, it does not guarantee that the best order is always found. The only requirement is a matrix with recombination fractions between markers. Next is an adapted excerpt from *Mollinari et al (2009)* describing the *UG* algorithm:

Based on the R *(recombination fraction) matrix, the distance between all* m *loci is calculated by* $d_{ij} = \hat{r}_{ij} + (\frac{2}{n_{ij}}) \sum_k \hat{r}_{ik}\hat{r}_{jk}$*, for every* k*, with* $\hat{r}_{ij} > \hat{r}_{ik}, \hat{r}_{ij} > \hat{r}_{jk}$*, and* n_{ij} *individuals. The value* $T_{ij} = 2d_{ij} - (\sum_{k \neq i} d_{ik} + \sum_{k \neq j} d_{jk})$ *is calculated for every* $i < j$*. The terminal end of the map is defined by taking the pair of markers* (f, g) *that presents the smallest value of* T*. The pair* (f, g) *is then denoted locus* $m + 1$ *and its distance to the remaining markers is determined by* $d_{im+1} = \frac{1}{2}(d_{if} + d_{ig} - d_{fg})$ *if* $(d_{if} + d_{ig}) > d_{fg}$*, if not,* $d_{im+1} = 0$*. The calculation* $W_{im+1} = (m - 2)d_{im+1} - \sum_{k \neq i} d_{ik}$ *is also performed and the locus that minimizes the value* W_{im+1} *(called locus* h*) is placed on the map. The partial resultant map is* f-g-h *if* $d_{fh} > d_{gh}$ *or* h-f-g *otherwise. Considering* $k = 2$*, the partial distance of the map with the remaining markers is updated:* $d_{im+k} = min(d_{im+k-1}, d_{ij})$*. The value* $W_{im+k} = (m - k - 1)d_{im+k} - \sum_{k \neq i} d_{ik}$ *is calculated and the locus that minimizes* W *is added to the map. The last two steps are repeated, taking* $k = 3, \ldots, m - 1$ *to obtain the complete map.*

After determining the order with *UG*, the final map is constructed using the multipoint approach (function `map`).

Value

An object of class `sequence`, which is a list containing the following components:

`seq.num` a `vector` containing the (ordered) indices of markers in the sequence, according to the input file.

`seq.phases` a `vector` with the linkage phases between markers in the sequence, in corresponding positions. `-1` means taht there are no defined linkage phases.

`seq.rf` a `vector` with the recombination frequencies between markers in the sequence. `-1` means that there are no estimated recombination frequencies.

`seq.like` log-likelihood of the corresponding linkage map.

`data.name` name of the object of class `outcross` with the raw data.

`twopt` name of the object of class `rf.2pts` with the 2-point analyses.

Author(s)

Marcelo Mollinari, `<mmollina@usp.br>`

References

Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. (2009) Evaluation of algorithms used to order markers on genetics maps. *Heredity* 103: 494-502.

Tan, Y. and Fu, Y. (2006) A novel method for estimating linkage maps. *Genetics* 173: 2383-2390.

See Also

`make.seq`, `map`

Examples

```
## Not run:
  #outcross example
  data(example.out)
  twopt <- rf.2pts(example.out)
  all.mark <- make.seq(twopt,”all”)
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.ug <- ug(LG1)

  #F2 example
  data(fake.f2.onemap)
  twopt <- rf.2pts(fake.f2.onemap)
  all.mark <- make.seq(twopt,”all”)
  groups <- group(all.mark)
  LG1 <- make.seq(groups,1)
  LG1.ug <- ug(LG1)
  LG1.ug

## End(Not run)
```

write.map *Write a genetic map to a file*

Description

Write a genetic map to a file, base on a given map, or a list of maps. The output file can be used as an input to QTL mapping using the R package R/qtl. It is also possible to create an output to be used with QTLCartographer.

Usage

```
write.map(map.list, file.out)
```

Arguments

`map.list`	a map, i.e. an object of class `sequence` with a predefined order, linkage phases, recombination fraction and likelihood or a list of maps.
`file.out`	output map file.

Details

This function is avaliable only for backcross, F2 and RILs.

Author(s)

Marcelo Mollinari, <mmollina@usp.br>

References

Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. (2008) *qtl: Tools for analyzing QTL experiments* R package version 1.09-43

Wang S., Basten, C. J. and Zeng Z.-B. (2010) Windows QTL Cartographer 2.5. Department of Statistics, North Carolina State University, Raleigh, NC.

Examples

```
## Not run:
data(fake.f2.onemap)
twopt<-rf.2pts(fake.f2.onemap)
lg<-group(make.seq(twopt, "all"))

##"pre-allocate" an empty list of length lg$n.groups (3, in this case)
  maps.list<-vector("list", lg$n.groups)

  for(i in 1:lg$n.groups){
    ##create linkage group i
    LG.cur <- make.seq(lg,i)
    ##ordering
    map.cur<-order.seq(LG.cur, subset.search = "sample")
    ##assign the map of the i-th group to the maps.list
    maps.list[[i]]<-make.seq(map.cur, "force")
  }

##write maps.list to "fake.f2.onemap.map" file
write.map(map.list, "fake.f2.onemap.map")

##Using R/qtl
##you must install the package  'qtl'
##install.packages("qtl")

require(qtl)
file<-paste(system.file("example",package="onemap"),"fake.f2.onemap.raw", sep="/")
dat1 <- read.cross("mm", file=file, mapfile="fake.f2.onemap.map")
newmap <- est.map(dat1, tol=1e-6, map.function="kosambi")

(logliks <- sapply(newmap, attr, "loglik"))
plot.map(dat1, newmap)

##Using R/qtl to generate QTL Cartographer input files (.map and .cro)
write.cross(dat1, format="qtlcart", filestem="fake.f2.onemap")


## End(Not run)
```

Index