

Package ‘dlmap’

January 25, 2012

Type Package

Title Detection Localization Mapping for QTL

Version 1.11

Date 2012-01-24

Author Emma Huang and Andrew George

Maintainer Emma Huang <Emma.Huang@csiro.au>

Description QTL mapping in a mixed model framework with separate detection and localization stages. The first stage detects the number of QTL on each chromosome based on the genetic variation due to grouped markers on the chromosome; the second stage uses this information to determine the most likely QTL positions. The mixed model can accommodate general fixed and random effects, including spatial effects in field trials and pedigree effects. Applicable to backcrosses, doubled haploids, recombinant inbred lines, F2 intercrosses, and association mapping populations.

License GPL-2

Depends qtl, ibdreg, wgaim, nlme, mgcv

Suggests asreml

Collate

`'calc.genprob2.R' 'calcpos.R' 'cintern.R' 'dlcross.cross.R' 'dlcross.dlmap.R' 'dlcross.other.R' 'dlcross.R' 'dlldetect.R' 'dlldetect.R'`

Repository CRAN

Date/Publication 2012-01-25 07:53:49

R topics documented:

dlmap-package	2
BSdat	3
BSphe	4
data access	5
dlcross	6
dlmap	9

Index	14
--------------	-----------

dlmap-package	<i>DLMapping for QTL detection</i>
---------------	------------------------------------

Description

QTL mapping in a mixed model framework with separate detection and localization stages. The former detects the number of QTL on each chromosome based on the genetic variation due to the grouped markers on the chromosome, while the latter stage uses this information to determine the most likely QTL positions. The mixed model can accommodate general fixed and random effects, including spatial effects in field trials and random pedigree effects.

Details

Package:	dlmap
Type:	Package
Version:	1.10
Date:	2012-01-24
License:	GPL 2

The function `dlcross` is a constructor function for `dlcross` objects to be input to `dlmap`. It will read in files in several different formats, including the **qtl** cross format and two new formats to accommodate association mapping populations and designs with extensive phenotypic data.

The primary function is `dlmap`, which performs the iterative algorithm to detect and position QTL on all chromosomes with significant genetic variation. This can accommodate sophisticated mixed models for phenotypic variation in addition to the genetic modeling.

Because ASReml-R is proprietary, we provide the option of using the **nlme** package to fit mixed models via the `algorithm` argument. However, there are some features which are only accessible when the `algorithm` used is `asreml`.

The vignette included in this package gives more background on the methodology, input file structure, and examples of how to use each of the important functions in the package.

Author(s)

Emma Huang and Andrew George
 Maintainer: Emma Huang <Emma.Huang@csiro.au>

References

Huang, B.E. and George, A.W. 2009. Look before you leap: A new approach to QTL mapping. TAG 119:899-911

See Also

[ibdreg](#), [lme](#)

BSdat

Simulated data for a backcross

Description

Dataset simulated according to Broman and Speed (2002).

Usage

```
data(BSdat)
```

Format

An object of class `cross`. List with two components:

- `geno` is a list with elements corresponding to chromosomes. `names(geno)` contains the names of the chromosomes. There are two components for each chromosome: `data`, a matrix whose rows are individuals and whose columns are markers, and `map`, a vector of marker positions in cM. There is no missing data, and genotypes are coded as 1=AA, 2=AB
- `pheno` is a data frame of size (250 x 2) containing the trait data. The first trait is generated from a random normal distribution with variance 1.0 and mean determined by the QTL genotypes as described below. The second trait is an ID for each individual

Details

The data was generated for a sample size of 250 from a map with 9 chromosomes. Each chromosome had length 100 cM and contained 11 equally spaced markers (spaced 10 cM apart). The background phenotypic variation was 1.0 and there was no missing data. The QTL were located as follows:

- Chr 1: 2 QTL located at 30 and 70 cM, both with effect size of 0.76
- Chr 2: 2 QTL located at 30 and 70 cM with effect size of 0.76 and -0.76
- Chr 3: 1 QTL located at 50 cM with effect size of 0.76
- Chr 4: 1 QTL located at 30 cM with effect size of 0.76
- Chr 5: 1 QTL located at 0 cM with effect size of 0.76

References

Broman, KW and Speed TP. 2002. A model selection approach for the identification of quantitative trait loci in experimental crosses. *JRSS-B* 64:641-656.

Examples

```
data(BSdat)
library(qtl)

# Summary of chromosomes and markers
nchr(BSdat)
nmar(BSdat)

# linkage map of data
plot.map(BSdat)

# interval mapping
BSgp <- calc.genoprob(BSdat, step=2)
BSim <- scanone(BSgp)

# composite interval mapping
BScim <- cim(BSgp, n.marcov=5, method="hk")

# LOD profile from CIM
plot(BScim)

# LOD threshold for 5 cofactors from paper
abline(h=3.56)
```

BSphe

Simulated phenotypic data

Description

Phenotypic data simulated with BSdat which has multiple observations per genotype

Usage

```
data(BSphe)
```

Format

A data frame with three columns:

ID: the ID for each individual. These are the same 250 individuals appearing in BSdat, each with 4 replicates

Block: a factor with 4 levels indicating the replicate

phenotype: the phenotype value. It is generated from the same QTL effects as in BSdat, with the additional block effect and random error

References

Broman, KW and Speed TP. 2002. A model selection approach for the identification of quantitative trait loci in experimental crosses. *JRSS-B* 64:641-656.

Examples

```
data(BSphe)

boxplot(BSphe$phenotype~BSphe$Block)
```

data access

Data summaries of dlcross and dlmap objects

Description

Access the number of unique genotyped individuals; unique phenotyped individuals; and number of markers on each map chromosome

Usage

```
ngen(object, ...)

## S3 method for class 'dlmap'
ngen(object, ...)

## S3 method for class 'dlcross'
ngen(object, ...)

nphen(object, ...)

## S3 method for class 'dlmap'
nphen(object, ...)

## S3 method for class 'dlcross'
nphen(object, ...)

nmrk(object, ...)

## S3 method for class 'dlmap'
nmrk(object, ...)

## S3 method for class 'dlcross'
nmrk(object, ...)
```

Arguments

object	Object of type dlcross or dlmap
...	Additional arguments

Value

ngen returns the number of unique genotyped individuals. nphen returns the number of unique phenotyped individuals - generally greater than or equal to ngen because of replicates. nmrk returns a vector indicating the number of markers on each chromosome.

Author(s)

Emma Huang and Andrew George

References

Huang, BE and George, AW. 2009. Look before you leap: A new approach to QTL mapping. TAG 119:899-911

Examples

```
# load dataset
data(BSdat)
data(BSphe2)

## Not run:
# convert data to dlmmap format
dl.in1 <- dlcross(format="rqt1", genobj=BSdat, idname="ID", fixpos=1)

ngen(dl.in1)
nphen(dl.in1)
nmrk(dl.in1)

# convert data with separate phenotypic trait file
dl.in2 <- dlcross(format="rqt1", genobj=BSdat, pheobj=BSphe2, idname="ID", step=5)

ngen(dl.in2)
nphen(dl.in2)
nmrk(dl.in2)

## End(Not run)
```

dlcross

Constructor, summary and plotting functions for dlcross format

Description

Reads in objects in cross format and files or objects in dlmmap format and converts them to dlmmap input format.

Usage

```

dlcross(format = c("rqtl", "dlmap", "other"), genobj, pheobj, mapobj, idname="ID", genfile, phefile, ma
## S3 method for class 'dlcross'
plot(x, chr, pheno.col, ...)
## S3 method for class 'dlcross'
summary(object, ...)

```

Arguments

format	See documentation for read.cross . Also supports the input of an object of class <code>cross</code> ("rqtl" format), "dlmap" format (described below) and continuous data for association mapping populations ("other" format)
genobj	if format="rqtl", object of class <code>cross</code> ; if format="dlmap" or "other" data frame containing genotypes
pheobj	Data frame or matrix containing supplementary phenotypic or environmental data
mapobj	if format="other" or "dlmap", 2/3-columned data-frame containing marker names, chromosomes and marker positions
idname	Unique identifier variable name; will be used to match phenotypic and marker data
genfile	if format="dlmap" see below
mapfile	if format="dlmap" see below
phefile	if format="dlmap" see below
type	Type of experimental cross; only necessary if format="dlmap" or "other"; if not input, will be assumed that data is from unrelated individuals
step	Step size for localization stage, i.e. if step=2, grid of positions spaced 2 cM apart are considered for QTL locations. If step=0 (default) positions are only located at markers.
fixpos	Alternative to specifying a step size - if fixpos=2, 2 evenly spaced positions between each marker are considered as QTL locations. If fixpos=0 (default) positions are only located at markers.
estmap	Flag for whether to re-estimate the linkage map. Cannot be done for format="other"
...	additional arguments
x	input to plot function, object of class <code>dlcross</code>
chr	character string naming the subset of chromosomes to plot; if numeric, a vector of chromosome indices
pheno.col	vector of phenotypes for which to show barplots/histograms
object	input to summary function, object of class <code>dlcross</code>

Details

The main function constructs `dlcross` objects for input into [dlmap](#). The `format` argument allows for compatibility with the `cross` format as supported by [read.cross](#) in R/qtl. In addition, `format="dlmap"` or "other" takes three files or objects in the following form.

- `genfile`: First row contains an identifier variable and marker names. Following rows contain genotype values for each individual
- `phefile`: First row contains names of phenotypic and environmental traits. One of these traits must be an identifier for each individual. Following rows contain values of the traits for each individual. Note: This file may contain more observations than the file containing the marker data
- `mapfile`: Contains marker names, chromosome groupings and positions

If a single set of trait values is available for each genotype, then phenotypic data will be input through the arguments `genobj` or `phefile` (depending on the file format). The argument `pheobj` allows for input of phenotypic data on replicates or additional individuals which are not necessarily genotyped.

Choosing `format="other"` allows for association mapping populations to be analyzed, and in this case the data can be input as with `format="dlmap"`, but a genetic linkage map is not required. Hence the `mapfile` only needs the first two columns of marker names and chromosome groupings.

The `plot` function plots diagnostics summarizing the data. If the type of cross is not "other", will plot the genetic map for the cross. Will also plot a histogram or barplot of the first few phenotypic variables. Will only plot a maximum of three phenotypic variables; `pheno.col` can be used to select which are plotted.

The `summary` function outputs a summary of data stored in `dlcross` object, including number of genotypes, number of phenotypes, number of phenotypic variables, number of chromosomes and markers per chromosome. For experimental crosses, based off of `cross` object summary; similar output for association mapping populations.

Value

Object with class "dlcross" which can be input to `dlmap.asreml` or `dlmap.lme`. Contains the following elements:

<code>dfMerged</code>	Data frame to be used in <code>dlMapping</code> analysis
<code>map</code>	Original genetic map
<code>nphe</code>	Number of phenotypic traits
<code>loc</code>	A flag for whether to run the localization stage
<code>idname</code>	the ID name input to the function
<code>mapp</code>	If format is not "other", genetic map augmented by imputed genotypes at grid of positions
<code>genCross</code>	if format is not "other", <code>rqt1</code> cross object. CAUTION: if there are replicates, i.e., more phenotypic data than genotypic, this object will not contain all of the phenotypic data for the sample

Author(s)

Emma Huang and Andrew George

References

Huang, BE and George, AW. 2009. Look before you leap: A new approach to QTL mapping. TAG 119:899-911

See Also

[plot.cross](#)

Examples

```
## Not run:
# load dataset
data(BSdat)
data(BSphe)

# convert data to dimap format
dl.in1 <- dlcross(format="rqt1", genobj=BSdat, idname="ID", fixpos=1)

# convert data with separate phenotypic trait file
dl.in2 <- dlcross(format="rqt1", genobj=BSdat, pheobj=BSphe, idname="ID", step=5)

plot(dl.in2)
summary(dl.in2)

## End(Not run)
```

dimap

Perform DLMapping

Description

Fits the iterative algorithm for DLMapping. Reads in data, performs detection and localization stages and outputs summary of selected QTL effects.

Usage

```
dimap(object, phename, baseModel, algorithm=c("asreml", "lme"), fixed = NULL,
random = NULL, rcov = NULL, sparse = NULL, pedigree,
seed = 1, maxit=60, n.perm = 0, multtest=c("holm", "bon"),
alpha = 0.05, filestem = "dl", ...)
```

S3 method for class 'dimap'

```
plot(x, chr, max.dist, qcol="light blue", mcol="red", pcol="purple", ...)
```

profileplot(object, ...)

S3 method for class 'dimap'

```
profileplot(object, chr, marker.names=TRUE, QTLpos=TRUE, pch=20, ...)
```

S3 method for class 'dimap'

```
summary(object, ...)
```

Arguments

object	For input to dimap, dlcross object created by create.dlcross; for plotting functions, object of class dimap
phename	Response variable name
algorithm	Indicates whether to fit mixed models using asreml function or lme function (using packages asreml or nlme)
fixed	A formula object specifying the fixed effects part of the base model, with the terms, separated by + operators, on the right of a ~ operator. There is no left side to the ~ expression. If no fixed effect is specified, the model defaults to ~1, i.e. intercept only.
random	A formula object specifying the random effects part of the base model, with the terms, separated by + operators, on the right of a ~ operator. See asreml documentation for more detail.
rcov	A formula object specifying the error structure of the model, with the terms, separated by + operators, on the right of a ~ operator. See asreml documentation for more detail.
sparse	A formula object specifying the fixed effects to be absorbed, with the terms, separated by + operators, on the right of a ~ operator. See asreml documentation for more detail.
baseModel	An alternative to specifying fixed, random, sparse, and rcov separately. If a base model has already been fit in asreml-R for the phenotypic variation, this can be input directly
pedigree	A pedigree object consisting of three columns. The first column is the individual ID, then the mother's ID and the father's ID. The name of the ID variable in the first column must match the idname variable
seed	Random number seed. Default=1
n.perm	Number of permutations used to get adjusted p-values at each iteration of detection. If n.perm=0 (default) the Holm correction is used.
multtest	Correction used for multiple testing. If n.perm>0 will use permutation, but otherwise can choose between Holm and Bonferroni
alpha	Significance level for testing
filestem	Stem to add to names of any files generated in DL Mapping process. Default="dl"
maxit	Maximum number of iterations to attempt for convergence of lme
...	additional arguments
x	object of class dimap
chr	character string naming the subset of chromosomes to plot; or, if numeric, a vector of chromosome indices
max.dist	a numerical value in cM determining the distance the genetic map should be subsetted by
qcol	colour of intervals surrounding QTL (see par for colour options)
mcol	colour of QTL flanking markers (see par for colour options)

pcol	colour of QTL positions (see par for colour options)
marker.names	logical value. if TRUE then marker names are plotted along the top of the profileplot. Defaults to TRUE
QTLpos	logical value. if TRUE then QTL positions are indicated with vertical lines in profileplot. Defaults to TRUE
pch	Character to be used for points in plotting; default is solid circle

Details

There are two versions of the main function, which use different engines to fit the linear mixed models which form the framework of the algorithm. Which is used depends on the value of the argument `algorithm`. `algorithm="asreml"` provides a much more general implementation of the DLMapping algorithm and is the preferred method of analysis. `algorithm="lme"` is more restricted in its capabilities, in that it cannot model random effects or covariance structure, cannot handle more than 200 markers, and only allows for a single phenotypic observation per genotype. Also, permutation has not been implemented for this function because it is very slow. However, this version will fit the basic algorithm and is useful should a license for ASReml not be available.

In studies where the number of genetic markers is much larger than the number of phenotyped individuals, we reduce the dimension of the analysis to the number of genetic lines used in the analysis multiplied by the number of chromosomes in the genetic map. This is done in a similar manner to **wgaim**, with thanks to Julian Taylor and Ari Verbyla for the suggestion. The transformation of the genetic data reduces the time for computational analysis for high-dimensional data and is particularly useful in association analysis.

This version of **wgaim** allows high dimensional marker information to be analysed. A simple transformation of the collated high dimensional marker set shows that it may be reduced to the number of genetic lines used in the analysis. This transformation is internal to the `wgaim.asreml` call and users can now expect a considerably large acceleration in the performance of **wgaim**.

In the `asreml` version, there are two options for specifying the model for phenotypic variation. The individual model components can either be input directly as they would be in an ASReml call, or a previous model (`baseModel`) output from ASReml can be input and the components will be retrieved from it. The latter formulation may be useful if prior phenotypic modelling has taken place. Note that in either case, variables appearing in the `rcov` statement must be ordered appropriately in the dataset. For example, if `rcov=~ar1(Column):ar1(Row)` the data must be sorted as *Row* within *Column*.

Missing values in `asreml` are replaced with zeros, so it is important to centre the covariate in question. This is done for all genotypes when `algorithm="asreml"`. Thus individuals with phenotypic but not genotypic data, which play important roles in field trials, may be included safely. When `algorithm="lme"` these individuals cannot be included, so the default behavior is to omit observations with missing values.

It is recommended that `n.perm` be set to 0 for initial exploratory analysis, as the permutation analysis may be lengthy. The Holm correction is used to adjust for the number of chromosomes under consideration at each detection stage. While this is a conservative measure it seems to perform well in practice.

Two files are output with names set by the argument `filestem`, which has a default value of "dl". The file "`filestem.trace`" contains ASReml licensing and likelihood convergence output which otherwise would be dumped to the screen and possibly obscure other messages. Errors, warnings and

other messages will still appear on the screen. Some warnings which appear may be passed through from an ASReml call and output on exit. These may generally be ignored. This file is not created if `algorithm="lme"` is used.

The file "filestem.det.log" is a record of iterations in the detection stage. For each iteration the REMLRT testing for genetic variation on each chromosome is output, along with adjusted p-values, genomewide threshold and markers selected as fixed effects. The p-values are corrected for the number of chromosomes tested either by the Holm correction or by permutation. If the number of permutations (`n.perm`) is greater than 0, then for the Xth iteration an additional file "filestem.permX" will be created which contains the test statistics for the permuted datasets. See the accompanying vignette for an example of how to interpret the ".det.log" file.

If the type of cross is not "other", the plotting function plots the genetic linkage map for a selection of chromosomes. Indicates marker locations, marker names, and detected QTL positions and associated flanking markers obtained from a dlmap fit. This function relies upon `link.map.cross`, which was written by Julian Taylor for the `wgaim` package. It is built upon here by adding QTL regions and estimated positions to the map.

The function `plot.dlmap` provides a neat visual display of chromosomes. If no QTL are detected, only the linkage map will be plotted; otherwise detected QTL will be placed at their estimated positions and the intervals around them (and flanking markers) will be highlighted. If a subset of chromosomes are plotted and detected QTLs exist outside that subset a warning will be given that QTLs have been omitted from the display.

The arguments `mcol`, `qcol` and `pcol` have been added for personal colour highlighting the flanking markers, QTL regions and QTL positions respectively. The procedure may also be given the usual `col` argument which will be passed on to the other markers.

In order to ensure that all marker names are displayed without vertical overlap, the default value of the "cex" parameter passed to "text" should be manipulated. For large maps with many chromosomes, marker names and adjacent chromosomes will overlap horizontally. In this case it is suggested that the user horizontally maximize the plotting window to remove overlap, or subset the chromosomes displayed.

The `profileplot` function plots the Wald statistic profile for each chromosome with detected QTL on first interval mapping scan. Indicates marker locations, marker names, and detected QTL positions obtained from a dlmap fit. It provides a neat visual display of the Wald profile for chromosomes with detected QTL. If no QTL are detected, nothing will be plotted. Otherwise, the Wald profile will be plotted by cM position of points on the interval mapping grid. Marker names will be displayed at the appropriate positions along the top of the plot. Vertical lines will mark the position of detected QTL.

The summary function outputs a summary of a dlmap object and detected QTL. It primarily prints the summary table computed from dlmap. This includes the chromosome QTL are detected on, estimated positions, flanking markers, QTL effects and standard deviations, Z-ratio and p-value.

Value

Summary	Table with one row per QTL detected, columns for which chromosome the QTL is on, its position (cM), flanking markers, additive (dominant) effect and standard deviation, Z-ratio and p-value.
<code>no.qtl</code>	Total number of QTL detected on all chromosomes

final.model	Object of class <code>asreml</code> for final model containing all terms in the base model, as well as effects for every QTL detected at the appropriate locations. No random effects for markers are fit
profile	If QTL are detected on C chromosomes, this is a list with C elements, each a matrix with 2 rows and a column for each position on the chromosome. The first row contains the cM position; the second row contains the Wald statistic for the model fit in the localization stage
input	Original <code>dldcross</code> input to the analysis

Author(s)

Emma Huang and Andrew George; Julian Taylor

References

Huang, BE and George, AW. 2009. Look before you leap: A new approach to QTL mapping. TAG 119:899-911

See Also

[dldcross](#)

Examples

```
## Not run:
data(BSdat)
data(BSphe)

# Convert cross object to DL Mapping format
dl.in1 <- dldcross(format="rqt1", genobj=BSdat, idname="ID", fixpos=1)

# Analyze data
BSdl <- dldmap(object=dl.in1, algorithm="lme", phename="phenotype", filestem="BS")

plot(BSdl)

# With additional phenotypic data
dl.in2 <- dldcross(format="rqt1", genobj=BSdat, pheobj=BSphe, idname="ID", step=5)
BSph <- dldmap(object=dl.in2, algorithm="asreml", phename="phenotype", env=TRUE, random=~Block)

profileplot(BSph)
summary(BSph)

## End(Not run)
```

Index

*Topic **datasets**

BSdat, [3](#)

BSphe, [4](#)

BSdat, [3](#)

BSphe, [4](#)

data access, [5](#)

dlcross, [2](#), [6](#), [13](#)

dmap, [2](#), [7](#), [9](#)

dmap-package, [2](#)

dmap-pkg (dmap-package), [2](#)

ibdreg, [3](#)

lme, [3](#)

ngen (data access), [5](#)

nmrk (data access), [5](#)

nphen (data access), [5](#)

plot.cross, [9](#)

plot.dlcross (dlcross), [6](#)

plot.dmap (dmap), [9](#)

profileplot (dmap), [9](#)

read.cross, [7](#)

summary.dlcross (dlcross), [6](#)

summary.dmap (dmap), [9](#)