

Package ‘diaQTL’

October 25, 2020

Title QTL Analysis in Diallel Populations

Version 0.80

Author Jeffrey B. Endelman and Rodrigo R. Amadeu

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Description QTL analysis of diploid and autotetraploid diallel populations. Phenotypes are regressed on genotype probabilities, and the regression coefficients are random effects.

Depends R (>= 3.5.0)

License GPL-3

LazyData true

RoxygenNote 7.1.1

Roxygen list(markdown = TRUE)

Encoding UTF-8

Imports BGLR, ggplot2, methods, coda, Matrix, scam, parallel, arrangements, tidyr

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

diallel_genoclass	2
diallel_genophenoclass	2
diplo_freq	3
diplo_get	3
F1codes	4
fitQTL	4
haplo_freq	6
haplo_get	7
haplo_plot	7
haplo_switch	8
IBD	9
LODthresh	10
read_data	10
read_polyancestry	12
S1codes	12
scan1	13
scan1_permute	14
scan1_summary	15
set_params	16

Index**18**

diallel_geno-class *S4 class with genotype data*

Description

S4 class with genotype data

Slots

ploidy Either 2 or 4

dominance Integer 1-4 indicating 1 = additive, 2 = digenic dominance, 3 = trigenic dominance, 4 = quadrigenic dominance.

X.GCA Incidence matrix for GCA effects

map data frame with marker,chrom, position (cM and/or bp) and bin

geno list of length equal to the number of marker bins. Each element is a list of length dominance. The elements in the nested list are sparse matrices with dimensions (id x effects).

diallel_geno_pheno-class
S4 class with genotype and phenotype data

Description

S4 class with genotype and phenotype data

Slots

ploidy Either 2 or 4

dominance Integer 1-4 indicating 1 = additive, 2 = digenic dominance, 3 = trigenic dominance, 4 = quadrigenic dominance.

X.GCA Incidence matrix for GCA effects

map data frame with marker,chrom, position (cM and/or bp) and bin

geno list of length equal to the number of marker bins. Each element is a list of length ploidy corresponding to additive, digenic, trigenic, and quadrigenic effects. The elements in the nested list are sparse matrices with dimensions (id x effects).

pheno data frame of phenotypes

X incidence matrix for fixed effects

Z incidence matrix for individuals

diplo_freq	<i>Diplotype frequencies</i>
------------	------------------------------

Description

Plot the frequency of individuals with diplotype dosage above a threshold

Usage

```
diplo_freq(data, diplotypes, dosage, position, chrom = NULL)
```

Arguments

data	Variable inheriting from class <code>diallel_genotype</code>
diplotypes	Names of diplotypes
dosage	Dosage threshold
position	Either "cM" or "bp" for plotting
chrom	Names of chromosomes (default is all)

Details

Useful for visualizing selection in selfed populations.

Value

List containing

- result** Data frame with the map and frequency
- plot** ggplot object

diplo_get	<i>Dosage of parental diplotypes</i>
-----------	--------------------------------------

Description

Dosage of parental diplotypes

Usage

```
diplo_get(data, marker = NULL, id = NULL)
```

Arguments

data	Variable inheriting from class <code>diallel_genotype</code>
marker	Name of marker
id	Name of individual

Details

Function can be used to get parental diplotype dosage estimates at a single marker for all individuals (in which case id should be NULL) or for a single individual for all markers (in which case marker should be NULL)

Value

Matrix of (id or markers) x parental diplotypes

Examples

```
## Not run:
diplo_example = diplo_get(data = diallel_example,
                          marker = "solcap_snp_c2_25522")
diplo_example = diplo_get(data = diallel_example,
                          id = "W15263-8R")

## End(Not run)
```

F1codes	<i>Genotype codes for F1 populations</i>
---------	--

Description

Character vector with the 100 possible tetraploid genotypes for a F1 population. Maternal haplotypes are denoted 1,2,3,4 and paternal haplotypes 5,6,7,8.

Usage

```
data(F1codes)
```

Format

character vector

fitQTL	<i>Fit a single QTL model</i>
--------	-------------------------------

Description

Fit a single QTL model

Usage

```
fitQTL(data, trait, marker, params, dominance = 1, CI = 0.9, cofactor = NULL)
```

Arguments

<code>data</code>	Variable of class <code>diallel_geno_pheno</code>
<code>trait</code>	Name of trait
<code>marker</code>	Name of marker to fit as QTL
<code>params</code>	List containing the number of burn-in (<code>burnIn</code>) and total iterations (<code>nIter</code>)
<code>dominance</code>	Dominance degree
<code>CI</code>	Probability for Bayesian credible interval
<code>cofactor</code>	Name of marker to fit as cofactor (optional)

Details

For quantitative traits, R^2 is the percent of variation explained by the regression (MSS/TSS). For binary traits, R^2 is the squared phi correlation (as a percentage). LOD score is the difference between the log10-likelihood for the QTL model vs. no QTL model; higher values are better. `deltaDIC` is the difference between the Deviance Information Criterion for the QTL model vs. no QTL model; lower values are better. Parameter `dominance` controls the genetic model: 1 = additive, 2 = digenic dominance, 3 = trigenic dominance, 4 = quadrigenic dominance. MCMC params can be estimated using `set_params`. Parameter `CI` sets the size of the Bayesian credible interval for the estimated effects at $(0.5 - CI/2, 0.5 + CI/2)$. The returned list `effects` contains the additive (and when included) digenic dominance effects. The proportion of variance for each effect is returned in `var`, labeled `h2` (additive), `d2` (digenic dominance), `t2` (trigenic dominance), `q2` (quadrigenic dominance).

Value

List containing

`R2` Coefficient of determination

`deltaDIC` Deviance Information Criterion relative to null model

`resid` Residuals

`var` Matrix with proportion of variance for additive and higher order effects

`effects` List of matrices containing the additive and higher order effects

`plots` List of ggplot objects for the effects

Examples

```
## Not run:
## additive effects
params1 <- set_params( diallel_example, trait = "tuber_shape" ,q=0.05,r=0.05)

fit1 <- fitQTL( data = diallel_example,
               trait = "tuber_shape",
               params = params1,
               marker = "solcap_snp_c2_25522",
               CI = 0.9)

## additive + dominance effects
params2 <- set_params( diallel_example, trait = "tuber_shape", dominance=2,q=0.05,r=0.05)

fit2 <- fitQTL( data = diallel_example,
               trait = "tuber_shape",
```

```

      params = params2,
      marker = "solcap_snp_c2_25522",
      dominance = 2,
      CI=0.9)

## End(Not run)

```

haplo_freq	<i>Haplotype frequencies</i>
------------	------------------------------

Description

Plot the frequency of individuals with haplotype dosage above a threshold

Usage

```
haplo_freq(data, haplotypes, dosage, position, chrom = NULL)
```

Arguments

data	Variable inheriting from class <code>diallel_geno</code>
haplotypes	Names of haplotypes
dosage	Dosage threshold
position	Either "cM" or "bp" for plotting
chrom	Names of chromosomes (default is all)

Details

Useful for visualizing selection in selfed populations.

Value

List containing

result Data frame with the map and frequency

plot ggplot object

haplo_get	<i>Dosage of parental haplotypes</i>
-----------	--------------------------------------

Description

Dosage of parental haplotypes

Usage

```
haplo_get(data, marker = NULL, id = NULL)
```

Arguments

data	Variable inheriting from class <code>diallel_geno</code>
marker	Name of marker
id	Name of individual

Details

Function can be used to get parental haplotype dosage estimates at a single marker for all individuals (in which case `id` should be `NULL`) or for a single individual for all markers (in which case `marker` should be `NULL`)

Value

Matrix of (id or markers) x parental haplotypes

Examples

```
## Not run:
haplo_example = haplo_get(data = diallel_example,
                          marker = "solcap_snp_c2_25522")
haplo_example = haplo_get(data = diallel_example,
                          id = "W15263-8R")

## End(Not run)
```

haplo_plot	<i>Plot parental haplotype dosage</i>
------------	---------------------------------------

Description

Plot parental haplotype dosages across the chromosome for one individual

Usage

```
haplo_plot(data, id, chrom, position, markers = NULL)
```

Arguments

data	Variable inheriting from class <code>diallel_geno</code>
id	Name of individual
chrom	Name of chromosome
position	Either "cM" or "bp"
markers	Optional, markers to indicate with dashed line

Details

For "cM" plotting, only one marker per bin is displayed. For "bp" plotting, all markers are included.

Value

ggplot object

Examples

```
## Not run:
haplo_plot(data = diallel_example,
            id = "W15263-8R",
            chrom = 10)

haplo_plot(data = diallel_example,
            id = "W15263-8R",
            chrom = 10,
            marker = "solcap_snp_c2_25522")

## End(Not run)
```

haplo_switch	<i>Find haplotype switches</i>
--------------	--------------------------------

Description

Find haplotype switches

Usage

```
haplo_switch(data, marker, haplotype, position, jump = 0.8)
```

Arguments

data	Variable inheriting from class <code>diallel_geno</code>
marker	Name of focal marker
haplotype	Name of parental haplotype
position	Either "cM" or "bp"
jump	Change in dosage to designate haplotype switch

Details

Designed to help with fine mapping of QTL. Function returns the location of the nearest haplotype switch on both sides (left, right) of a marker and haplotype of interest that represent the presumed location of a QTL allele.

Value

Data frame with the locations of the nearest haplotype switch on the left and right side of the focal marker for all individuals with a haplotype switch

IBD	<i>Realized IBD relationship</i>
-----	----------------------------------

Description

Calculates realized relationship matrices (additive and dominance) from founder genotype probabilities

Usage

```
IBD(data, dominance = 1, chrom = NULL)
```

Arguments

data	Variable inheriting from class <code>diallel_geno</code>
dominance	One of 1,2,3,4
chrom	Optional, vector of chromosome names to include

Details

Parameter dominance refers to 1 = additive, 2 = digenic, 3 = trigenic, 4 = quadrigenic (Gallais 2003). Can specify to use only a subset of the chromosomes (by default, all chromosomes are used). Calculated based on the marker bins.

Value

Relationship matrix

References

Gallais, A. 2003. Quantitative Genetics and Breeding Methods in Autopolyploid Plants. Institut National de la Recherche Agronomique, Paris.

Examples

```
## Not run:
IBmat_example = IBD(data = diallel_example, dominance=1) #additive
IBD_example = IBD(data = diallel_example, dominance=2) #digenic dominance

## End(Not run)
```

LODthresh	<i>LOD thresholds for scan1</i>
-----------	---------------------------------

Description

LOD thresholds for scan1

Usage

```
LODthresh(genome.size, num.parents, ploidy)
```

Arguments

genome.size	Genome size in Morgans (not centiMorgans)
num.parents	Number of parents
ploidy	2 or 4

Details

LOD thresholds to control the genome-wide false positive rate at 0.05 were determined via simulation for up to 20 parents and genome sizes up to 12 Morgans. A monotone increasing concave curve was fit to these results using R package *scam* and is used for prediction. (The LOD threshold does not depend on population size.)

Value

LOD threshold

read_data	<i>Read data files</i>
-----------	------------------------

Description

Reads genotype, pedigree, and phenotype data files

Usage

```
read_data(
  genofile,
  ploidy = 4,
  pedfile,
  phenofile = NULL,
  fixed = NULL,
  bin.markers = T,
  dominance = 2,
  n.core = 1
)
```

Arguments

genofile	File with map and genotype probabilities
ploidy	Either 2 or 4
pedfile	File with pedigree data (id,parent1,parent2)
phenofile	File with phenotype data (optional)
fixed	If there are fixed effects, this is a character vector of "factor" or "numeric"
bin.markers	TRUE/FALSE whether to bin markers with the same cM position
dominance	Maximum value of dominance that will be used for analysis (1-4). See Details.
n.core	Number of cores for parallel execution (only available for UNIX/Linux/macOS command line)

Details

Genotype and pedigree input files can be created from PolyOrigin output using [read_polyancestry](#). The first 3 or 4 columns of the genotype file are the map (marker, chrom, bp and/or cM), followed by the members of the population. The genotype data for each marker x individual combination is a string with the format "statestatestate...=>probprobprob...", where "state" refers to the genotype state and "prob" is the genotype probability in decimal format. Only states with nonzero probabilities need to be listed. The encoding for the states in tetraploids is described in the documentation for the F1codes and S1codes datasets that come with the package. For diploids, there are 4 F1 genotype codes, 1,2,3,4, which correspond to haplotype combinations 1-3,1-4,2-3,2-4, respectively; the S1 genotype codes 1,2,3 correspond to 1-1,1-2,2-2, respectively. For the phenotype file, first column is id, followed by traits, and then any fixed effects. Pass a character vector for the function argument "fixed" to specify whether each effect is a factor or numeric covariate. The number of traits is deduced based on the number of columns. Binary traits must be coded N/Y and are converted to 0/1 internally for analysis by probit regression. Missing data in the phenotype file should be coded as NA. The parameter dominance specifies the maximum value of dominance that can be used in subsequent analysis: 1 = additive, 2 = digenic dominance, 3 = trigenic dominance, 4 = quadrigenic dominance. For maximum flexibility, use dominance = 4, but more memory is required. This will allow you to use any value of dominance (from 1 to 4) in functions such as [scan1](#) and [fitQTL](#). Output files from the BGLR package are stored in a folder named 'tmp' in the current directory.

Value

Variable of class [diallel_geno](#) if phenofile is NULL, otherwise [diallel_geno_pheno](#)

Examples

```
## Not run:
## Get the location of raw csv files examples
genocsv = system.file( "vignette_data", "potato_geno.csv", package = "diaQTL" )
pedcsv = system.file( "vignette_data", "potato_ped.csv", package = "diaQTL" )
phenocsv = system.file( "vignette_data", "potato_pheno.csv", package = "diaQTL" )

## Check their location in the system
print(genocsv)
print(pedcsv)
print(phenocsv)

## Load them in R
diallel_example <- read_data(genofile = genocsv,
                             ploidy = 4,
```

```
pedfile = pedcsv,  
phenofile = phenocsv)  
  
## End(Not run)
```

read_polyancestry	Create diaQTL input files from polyancestry file
-------------------	--

Description

Create diaQTL input files from polyancestry file

Usage

```
read_polyancestry(filename, remove.outliers = TRUE)
```

Arguments

- filename Name of polyancestry file
- remove.outliers Should offspring flagged as outliers be removed (default is TRUE)

Details

Creates the pedigree (diaQTL_pedfile.csv) and genotype (diaQTL_genofile.csv) input files needed for [read_data](#) from the polyancestry output file generated by the PolyOrigin software.

S1codes	Genotype codes for S1 populations
---------	-----------------------------------

Description

Character vector with the 35 possible tetraploid genotypes for a S1 population. Haplotypes are denoted 1,2,3,4.

Usage

```
data(S1codes)
```

Format

character vector

scan1	<i>Single QTL scan</i>
-------	------------------------

Description

Performs a linear regression for each position in the map.

Usage

```
scan1(  
  data,  
  trait,  
  params,  
  dominance = 1,  
  chrom = NULL,  
  cofactor = NULL,  
  n.core = 1  
)
```

Arguments

data	Variable of class diallel_geno_pheno
trait	Name of trait
params	List containing burnIn and nIter
dominance	Dominance degree (1-4)
chrom	Names of chromosomes to scan (default is all)
cofactor	Optional name of marker to include as cofactor in the scan
n.core	Number of cores for parallel execution (only available from Linux or Mac command line)

Details

For quantitative traits, R2 is the percent of variation explained by the regression (MSS/TSS). For binary traits, R2 is the squared phi correlation (as a percentage). LOD score is the difference between the log10-likelihood for the QTL model vs. no QTL model; higher values are better. deltaDIC is the difference between the Deviance Information Criterion for the QTL model vs. no QTL model; lower values are better. Parameter dominance controls the genetic model: 1 = additive, 2 = digenic dominance, 3 = trigenic dominance, 4 = quadrigenic dominance. MCMC params can be estimated using [set_params](#).

Value

Data frame containing the map, LOD, R2 and deltaDIC results.

Examples

```
## Not run:
par1 <- set_params(data = diallel_example,
                   trait = "tuber_shape")

scan1_example <- scan1(data = diallel_example,
                      chrom = 10,
                      trait = "tuber_shape",
                      params = par1)

## End(Not run)
```

scan1_permute	<i>Permutation test for scan1</i>
---------------	-----------------------------------

Description

Permutation test for scan1

Usage

```
scan1_permute(
  data,
  trait,
  params,
  n.permute = 1000,
  chrom = NULL,
  dominance = 1,
  cofactor = NULL,
  n.core = 1
)
```

Arguments

data	Variable of class diallel_geno_pheno
trait	Name of trait
params	List containing burnIn and nIter
n.permute	Number of permutations
chrom	Names of chromosomes to scan (default is all)
dominance	Dominance degree (1-4)
cofactor	Optional name of marker to include as cofactor in the scan
n.core	Number of cores for parallel execution (only available from Linux or Mac command line)

Value

Data frame with maximum LOD and minimum deltaDIC for each iteration

Examples

```
## Not run:
par1 <- set_params(data = diallel_example,
                   trait = "tuber_shape")

ans1_permut <- scan1_permute(data = diallel_example,
                           chrom = 10,
                           trait = "tuber_shape",
                           params = par1,
                           n.permute = 100)

## End(Not run)
```

scan1_summary	<i>Summary of scan1 result</i>
---------------	--------------------------------

Description

Summary of scan1 result

Usage

```
scan1_summary(scan1_data, thresh = NULL, chrom = NULL, position = "cM")
```

Arguments

scan1_data	output from scan1
thresh	optional, LOD threshold for plotting
chrom	optional, subset of chromosomes to plot
position	Either "cM" or "bp"

Value

List containing

peaks Data frame of the markers with the highest LOD score per chromosome

plot ggplot object

Examples

```
## Not run:
scan1_summary( scan1_example )
scan1_summary( scan1_example, chrom = "10" )
scan1_summary( scan1_example, chrom = c( "10", "12" ) )

## End(Not run)
```

set_params

*Determine burn-in and total number of iterations***Description**

Determine burn-in and total number of iterations

Usage

```
set_params(
  data,
  trait,
  dominance = 1,
  marker = NULL,
  q = 0.5,
  r = 0.1,
  nIter = 2000
)
```

Arguments

data	variable of class <code>diallel_geno_pheno</code>
trait	name of trait
dominance	dominance degree
marker	name of marker (optional)
q	quantile to estimate
r	tolerance for quantile
nIter	number of iterations

Details

Determines the burn-in and total number of iterations using the Raftery and Lewis diagnostic from R package coda, based on a 95% probability that the estimate for quantile q of the additive effects is within the interval $(q-r, q+r)$. The 90th percentile for burn-in and total iterations across the additive effects is returned. Parameter dominance specifies which genetic model (1 = additive, 2 = digenic dominance, 3 = trigenic dominance, 4 = quadrigenic dominance) to use when determining the number of iterations, but this must be parameter must still be specified when calling functions such as `scan1` or `fitQTL`. Suggested values for `scan1` are $q=0.5$ and $r=0.1$. For `fitQTL`, the values depend on the desired Bayesian credible interval. For a 90% CI, suggested values are $q=0.05$ and $r=0.05$. If `marker=NULL` (default), the first marker of every chromosome is analyzed to generate parameters suitable for `scan1`. Parameter `nIter` sets the number of iterations used to apply the Raftery and Lewis diagnostic; the default value is 2000, and if a larger number is needed, an error will be generated with this information.

Value

List containing

burnIn Number of burn-in iterations

nIter Total number of iterations

Examples

```
## Not run:  
# Parameters for scan1  
par1 <- set_params(data = diallel_example,  
                   trait = "tuber_shape", q=0.5, r=0.1)  
  
# Parameters for fitQTL  
par2 <- set_params(data = diallel_example,  
                   trait = "tuber_shape", q=0.05, r=0.05, marker="solcap_c2_25522")  
  
## End(Not run)
```

Index

* datasets

F1codes, 4
S1codes, 12

diallel_genotype, 3, 6–9, 11
diallel_genotype (diallel_genotype-class), 2
diallel_genotype-class, 2
diallel_genotype_phenotype, 5, 11, 13, 14, 16
diallel_genotype_phenotype
 (diallel_genotype_phenotype-class), 2
diallel_genotype_phenotype-class, 2
diplo_freq, 3
diplo_get, 3

F1codes, 4
fitQTL, 4, 11, 16

haplo_freq, 6
haplo_get, 7
haplo_plot, 7
haplo_switch, 8

IBD, 9

LODthresh, 10

read_data, 10, 12
read_polyancestry, 11, 12

S1codes, 12
scan1, 11, 13, 16
scan1_permute, 14
scan1_summary, 15
set_params, 5, 13, 16