Package ‘qtlpoly’

March 24, 2019

Type Package

Title Random-effect multiple QTL mapping in autopolyploids

Version 0.1.0

Author Guilherme da Silva Pereira [aut], Marcelo Mollinari [ctb], Zhao-Bang Zeng [ctb]

Maintainer Guilherme da Silva Pereira <gdasilv@ncsu.edu>

Description Performs random-effect multiple interval mapping (REMIM) in full-sib families of autopolyploid species based on REML estimation and score statistics.

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.5)

Imports varComp (== 0.2), sommer (== 3.6), ggplot2 (>= 3.1), abind (>= 1.4), MASS (>= 7.3), parallel, stats

Suggests mappoly, knitr, rmarkdown, devtools

VignetteBuilder knitr

Remotes cran/varComp

RoxygenNote 6.1.1

R topics documented:

breeding_values *Prediction of QTL-based breeding values from REMIM model*

Description

Computes breeding values for each genotyped individual based on multiple QTL models

Usage

```
breeding_values(data, fitted)

## S3 method for class 'qtlpoly.bvalues'
plot(x, pheno.col = NULL)
```

Arguments

data	an object of class qtlpoly.data.
fitted	an object of class qtlpoly.fitted.
x	an object of class qtlpoly.bvalues to be plotted.
pheno.col	a numeric vector with the phenotype column numbers to be plotted; if NULL, all phenotypes from 'data' will be included.

Value

An object of class qtlpoly.bvalues which is a list of results for each trait containing the following components:

pheno.col	a phenotype column number.
y.hat	a column matrix of breeding value for each individual.

A **ggplot2** histogram with the distribution of breeding values.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi:`.

See Also

`read_data`, `fit_model`

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
    d.sint = 1.5, n.clusters = 4, plot = "remim")

  # fit model
  fitted.mod <- fit_model(data = data, model = remim.mod, probs = "joint",
    polygenes = "none")

  # predict genotypic values
  y.hat <- breeding_values(data = data, fitted = fitted.mod)
  plot(y.hat)

## End(Not run)
```

`feim` *Fixed-effect interval mapping (FEIM)*

Description

Performs interval mapping using the single-QTL, fixed-effect model proposed by Hackett et al. (2001).

Usage

```
feim(data = data, pheno.col = NULL, w.size = 15, sig.lod = 7,
  d.sint = 1.5, plot = "feim", verbose = TRUE)

## S3 method for class 'qtlpoly.feim'
print(x, pheno.col = NULL, sint = NULL)
```

Arguments

`data` an object of class `qtlpoly.data`.

`pheno.col` a numeric vector with the phenotype columns to be analyzed; if `NULL` (default), all phenotypes from `'data'` will be included.

`w.size` a number representing the window size (in centiMorgans) to be avoided on either side of QTL already in the model when looking for a new QTL, e.g. 15 (default).

`sig.lod` the vector of desired significance LOD thresholds (usually permutation-based) for declaring a QTL for each trait, e.g. 5 (default); if a single value is provided, the same LOD threshold will be applied to all traits.

`d.sint` a d value to subtract from logarithm of the odds $(LOD - d)$ for support interval calculation, e.g. $d = 1.5$ (default) represents approximate 95% support interval.

`plot` a suffix for the file's name containing plots of every algorithm step, e.g. "remim" (default); if `NULL`, no file is produced.

`verbose` if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `qtlpoly.feim` to be printed.

`sint` whether `"upper"` or `"lower"` support intervals should be printed; if `NULL` (default), QTL peak information will be printed.

Value

An object of class `qtlpoly.feim` which contains a list of `results` for each trait with the following components:

`pheno.col` a phenotype column number.

`LRT` a vector containing LRT values.

`LOD` a vector containing LOD scores.

`AdjR2` a vector containing adjusted R^2.

`qtls` a data frame with information from the mapped QTL.

`lower` a data frame with information from the lower support interval of mapped QTL.

`upper` a data frame with information from the upper support interval of mapped QTL.

Author(s)

Guilherme da Silva Pereira, `<gdasilv@ncsu.edu>`

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

Hackett, C.A., Bradshaw, J.E., McNicol, J.W. (2001) Interval mapping of quantitative trait loci in autotetraploid species, *Genetics* 159: 1819-1832. http://www.genetics.org/content/159/4/1819

See Also

permutations

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  feim.mod <- feim(data = data, sig.lod = 7, plot = "feim")

## End(Not run)
```

fit_model *Fits multiple QTL models*

Description

Fits alternative multiple QTL models by performing variance component estimation using REML.

Usage

```
fit_model(data, model, probs = c("joint", "marginal"),
  polygenes = c("none", "most", "all"), keep = TRUE, verbose = TRUE)

## S3 method for class 'qtlpoly.fitted'
summary(x, pheno.col = NULL)
```

Arguments

`data` an object of class `qtlpoly.data`.

`model` an object of class `qtlpoly.profile` or `qtlpoly.remim`.

`probs` a character string indicating if either `"joint"` (genotypes) or `"marginal"` (parental gametes) conditional probabilities should be used.

`polygenes` a character string indicating if either `"none"`, `"most"` or `"all"` QTL should be used as polygenes.

`keep` if `TRUE` (default), stores all matrices and estimates from fitted model; if `FALSE`, nothing is stored.

`x` an object of class `qtlpoly.fitted` to be summarized.

`pheno.col` a numeric vector with the phenotype column numbers to be summarized; if `NULL`, all phenotypes from `'data'` will be included.

Value

An object of class `qtlpoly.fitted` which contains a list of `results` for each trait with the following components:

`pheno.col` a phenotype column number.

`fitted` a **sommer** object of class `mmer`.

`qtls` a data frame with information from the mapped QTL.

Author(s)

Guilherme da Silva Pereira, <`gdasilv@ncsu.edu`>

References

Covarrubias-Pazaran G (2016) Genome-assisted prediction of quantitative traits using the R package sommer. *PLoS ONE* 11 (6): 1–15. `doi:10.1371/journal.pone.0156744`.

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi:`.

See Also

`read_data`, `remim`

Examples

```
## Not run:
# load raw data
data(maps)
data(pheno)

# estimate conditional probabilities using mappoly package
```

```
    library(mappoly)
    genoprob <- lapply(maps, calc_genoprob)

    # prepare data
    data <- read_data(ploidy = 6, geno.prob = geno.prob, pheno = pheno, step = 1)

    # perform remim
    remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
      d.sint = 1.5, n.clusters = 4, plot = "remim")

    # fit model
    fitted.mod <- fit_remim(data=data, model=remim.mod, probs="joint", polygenes="none")

  ## End(Not run)
```

maps *Simulated autohexaploid map*

Description

A simulated map containing three homology groups of a hypotetical cross between two autohexaploid individuals.

Usage

```
data(maps)
```

Format

An object of class "mappoly.map" from the package **mappoly**, which is a list of three linkage groups (LGs):

LG 1 538 markers distributed along 112.2 cM

LG 2 329 markers distributed along 54.6 cM

LG 3 443 markers distributed along 98.2 cM

Author(s)

Marcelo Mollinari, <mmollin@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

Mollinari M, Garcia AAF (2019) Linkage analysis and haplotype phasing in experimental autopolyploid populations with high ploidy level using hidden Markov models, *bioRxiv*. https://doi.org/10.1101/415232

See Also

hexafake, pheno

Examples

```
## Not run:
data(maps)
library(mappoly)
plot(maps)

## End(Not run)
```

modify_qtl *Modify QTL model*

Description

Adds or removes QTL manually from a given model.

Usage

```
modify_qtl(model, pheno.col = NULL, add.qtl = NULL, drop.qtl = NULL,
  verbose = TRUE)

## S3 method for class 'qtlpoly.modify'
print(x, pheno.col = NULL)
```

Arguments

`model` an object of class `qtlpoly.model` containing the QTL to be modified.

`pheno.col` a phenotype column number whose model will be modified or printed.

`add.qtl` a marker position number to be added.

`drop.qtl` a marker position number to be removed.

`verbose` if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `qtlpoly.modify` to be printed.

Value

An object of class `qtlpoly.modify` which contains a list of `results` for each trait with the following components:

`pheno.col` a phenotype column number.

`stat` a vector containing values from score statistics.

`pval` a vector containing *p*-values from score statistics.

`qtls` a data frame with information from the mapped QTL.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

read_data, remim

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
    d.sint = 1.5, n.clusters = 4, plot = "remim")

  # modify model
  modified.mod <- modify_qtl(model = remim.mod, pheno.col = 3, add.qtl = 184)

## End(Not run)
```

null_model *Null model*

Description

Creates a null model (with no QTL) for each trait.

Usage

```
null_model(data, pheno.col = NULL, n.clusters = NULL, plot = "null",
  verbose = TRUE)

## S3 method for class 'qtlpoly.null'
print(x, pheno.col = NULL)
```

Arguments

`data`	an object of class `qtlpoly.data`.
`pheno.col`	a numeric vector with the phenotype columns to be analyzed; if `NULL`, all phenotypes from `'data'` will be included.
`n.clusters`	number of parallel processes to spawn.
`plot`	a suffix for the file's name containing simple plots of every QTL search round, e.g. "null" (default); if `NULL`, no file is produced.
`verbose`	if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.
`x`	an object of class `qtlpoly.null` to be printed.

Value

An object of class `qtlpoly.null` which contains a list of `results` for each trait with the following components:

`pheno.col`	a phenotype column number.
`stat`	a vector containing values from score statistics.
`pval`	a vector containing *p*-values from score statistics.
`qtls`	a data frame with information from the mapped QTL (`NULL` at this point).

Author(s)

Guilherme da Silva Pereira, <`gdasilv@ncsu.edu`>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi:`.

Qu L, Guennel T, Marshall SL (2013) Linear score tests for variance components in linear mixed models and applications to genetic association studies. *Biometrics* 69 (4): 883–92. `doi:10.1111/biom.12095`.

See Also

`read_data`

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using 'mappoly' package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # build null models
  null.mod <- null_model(data = data, n.clusters = 4, plot = "null")

## End(Not run)
```

optimize_qtl *Model optimization*

Description

Tests each QTL at a time and updates its position (if it changes) or drops the QTL (if non-significant).

Usage

```
optimize_qtl(data, model, sig.bwd = 1e-04, polygenes = FALSE,
  n.clusters = NULL, plot = "optimize", verbose = TRUE)

## S3 method for class 'qtlpoly.optimize'
print(x, pheno.col = NULL)
```

Arguments

`data`	an object of class `qtlpoly.data`.
`model`	an object of class `qtlpoly.model` containing the QTL to be optimized.
`sig.bwd`	the desired score-based *p*-value threshold for backward elimination, e.g. 0.0001 (default).
`polygenes`	if `TRUE` all QTL but the one being tested are treated as a single polygenic effect, if `FALSE` (default) all QTL effect variances have to estimated.
`n.clusters`	number of parallel processes to spawn.
`plot`	a suffix for the file's name containing plots of every QTL optimization round, e.g. "optimize" (default); if `NULL`, no file is produced.
`verbose`	if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

x	an object of class qtlpoly.optimize to be printed.
pheno.col	a numeric vector with the phenotype columns to be printed; if NULL, all phenotypes from 'data' will be included.

Value

An object of class qtlpoly.optimize which contains a list of results for each trait with the following components:

pheno.col	a phenotype column number.
stat	a vector containing values from score statistics.
pval	a vector containing *p*-values from score statistics.
qtls	a data frame with information from the mapped QTL.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

Qu L, Guennel T, Marshall SL (2013) Linear score tests for variance components in linear mixed models and applications to genetic association studies. *Biometrics* 69 (4): 883–92. doi:10.1111/biom.12095.

See Also

read_data, null_model, search_qtl

Examples

```
## Not run:
# load raw data
data(maps)
data(pheno)

# estimate conditional probabilities using 'mappoly' package
library(mappoly)
genoprob <- lapply(maps, calc_genoprob)

# prepare data
data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

# build null models
null.mod <- null_model(data = data, n.clusters = 4, plot = "null")

# perform forward search
```

```
  search.mod <- search_qtl(data = data, model = null.mod, w.size = 15, sig.fwd = 0.01,
    n.clusters = 4, plot = "search")

  # optimize model
  optimize.mod <- optimize_qtl(data = data, model = search.mod, sig.bwd = 0.0001,
    n.clusters = 4, plot = "optimize")

## End(Not run)
```

`permutations` *Fixed-effect interval mapping (FEIM) model permutations*

Description

Stores maximum LOD scores for a number of permutations of given phenotypes.

Usage

```
permutations(data, pheno.col = NULL, n.sim = 1000, probs = c(0.9,
  0.95), n.clusters = 1, seed = 123, verbose = TRUE)

## S3 method for class 'qtlpoly.perm'
print(x, pheno.col = NULL, probs = c(0.9, 0.95))
```

Arguments

`data` an object of class `qtlpoly.data`.

`pheno.col` a numeric vector with the phenotype columns to be analyzed; if `NULL` (default), all phenotypes from `'data'` will be included.

`n.sim` a number of simulations, e.g. 1000 (default).

`probs` a vector of probability values in [0, 1] representing the quantiles, e.g. c(0.90, 0.95) for the 90% and 95% quantiles.

`n.clusters` a number of parallel processes to spawn.

`seed` an integer for the `set.seed()` function; if `NULL`, no reproducible seeds are set.

`verbose` if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `qtlpoly.perm` to be printed or plotted.

Value

An object of class `qtlpoly.perm` which contains a list of `results` for each trait with the maximum LOD score per permutation.

LOD score thresholds for given quantiles for each trait.

A **ggplot2** histogram with the distribution of ordered maximum LOD scores and thresholds for given quantiles for each trait.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Churchill GA, Doerge RW (1994) Empirical threshold values for quantitative trait mapping, *Genetics* 138: 963-971. http://www.genetics.org/content/138/3/963

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

feim

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform permutations
  perm <- permutations(data = data, n.sim = 1000, n.clusters = 4)

## End(Not run)
```

pheno *Simulated phenotypes*

Description

A simulated data set of phenotypes for a hipotetical autohexaploid species map.

Usage

```
data(pheno)
```

Format

A data frame of phenotypes with 300 named individuals in rows and three named phenotypes in columns, which are:

T32 3 QTLs, with heritabilities of 0.20 (LG 1 at 32.03 cM), 0.15 (LG 1 at 95.02 cM) and 0.30 (LG 2 at 40.01 cM).

T17 1 QTL, with heritability of 0.15 (LG 3 at 34.51 cM).

T45 no QTLs.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

simulate_qtl, pheno

Examples

```
## Not run:
data(pheno)
head(pheno)

## End(Not run)
```

plot_profile *Logarithm of* p*-value (LOP) profile plots*

Description

Plots profiled logarithm of score-based *p*-values (LOP) from individual or combined traits.

Usage

```
plot_profile(data = data, model = model, pheno.col = NULL,
  main = NULL, ylim = NULL, grid = FALSE)
```

Arguments

data	an object of class qtlpoly.data.
model	an object of class qtlpoly.profile or qtlpoly.remim.
pheno.col	a numeric vector with the phenotype column numbers to be plotted; if NULL, all phenotypes from 'data' will be included.
main	a character string with the main title; if NULL, no title will be shown.
ylim	a numeric value pair supplying the limits of y-axis, e.g. c(0,10); if NULL (default), limits will be provided automatically.
grid	if TRUE, profiles will be organized in rows (one per trait); if FALSE (default), profiles will appear superimposed. Only effective when plotting profiles from more than one trait.

Value

A **ggplot2** with the LOP profiles for each trait.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

profile_qtl, remim

Examples

```
## Not run:
# load raw data
data(maps)
data(pheno)

# estimate conditional probabilities using mappoly package
library(mappoly)
genoprob <- lapply(maps, calc_genoprob)

# prepare data
data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

# perform remim
remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
  d.sint = 1.5, n.clusters = 4, plot = "remim")
```

```
    # plot profiles
    for (p in remim.mod$pheno.col) {
      plot_profile(data = data, model = remim.mod, pheno.col = p, ylim = c(0, 10))
    } # separate plots

    plot_profile(data = data, model = remim.mod, grid = FALSE) # combined plots

  ## End(Not run)
```

`plot_qtl` *QTL heritability and significance plot*

Description

Creates a plot where dot sizes and colors represent the QTLs heritabilities and their *p*-values, respectively.

Usage

```
plot_qtl(data = data, model = model, fitted = fitted,
  pheno.col = NULL, main = NULL, drop = FALSE)
```

Arguments

`data`	an object of class `qtlpoly.data`.
`model`	an object of class `qtlpoly.profile` or `qtlpoly.remim`.
`fitted`	an object of class `qtlpoly.fitted`.
`pheno.col`	the desired phenotype column numbers to be plotted. The order here specifies the order of plotting (from top to bottom.)
`main`	plot title; if `NULL` (the default), no title is shown.
`drop`	if `FALSE`, shows the names of all traits from `pheno.col`, even of those with no QTLs; if `TRUE` (the default), shows only the traits with QTL(s).

Value

A **ggplot2** with dots representing the QTLs.

Author(s)

Guilherme da Silva Pereira, <`gdasilv@ncsu.edu`>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi:`.

See Also

read_data, remim, fit_model

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
    d.sint = 1.5, n.clusters = 4, plot = "remim")

  # fit model
  fitted.mod <- fit_model(data=data, model=remim.mod, probs="joint", polygenes="none")

  # plot qtls
  plot_qtl(data = data, model = remim.mod, fitted = fitted.mod)

## End(Not run)
```

plot_sint *QTLs with respective support interval plots*

Description

Creates a plot where colored bars represent the support intervals for QTL peaks (black dots).

Usage

```
plot_sint(data, model, pheno.col = NULL, main = NULL, drop = FALSE)
```

Arguments

data	an object of class qtlpoly.data.
model	an object of class qtlpoly.profile or qtlpoly.remim.
pheno.col	a numeric vector with the phenotype column numbers to be plotted; if NULL, all phenotypes from 'data' will be included.
main	a character string with the main title; if NULL, no title will be shown.
drop	if TRUE, phenotypes with no QTL will be dropped; if FALSE (default), all phenotypes will be shown.

Value

A **ggplot2** with QTL bars for each linkage group.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

read_data, remim, profile_qtl

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
    d.sint = 1.5, n.clusters = 4, plot = "remim")

  # plot support intervals
  plot_sint(data = data, model = remim.mod)

## End(Not run)
```

profile_qtl *QTL profiling*

Description

Generates the score-based genome-wide profile conditional to the selected QTL.

Usage

```
profile_qtl(data, model, d.sint = 1.5, polygenes = FALSE,
  n.clusters = NULL, plot = "profile", verbose = TRUE)

## S3 method for class 'qtlpoly.profile'
print(x, pheno.col = NULL, sint = NULL)
```

Arguments

`data`	an object of class `qtlpoly.data`.
`model`	an object of class `qtlpoly.model` containing the QTL to be profiled.
`d.sint`	a d value to subtract from logarithm of p-value ($LOP - d$) for support interval calculation, e.g. $d = 1.5$ (default) represents approximate 95% support interval.
`polygenes`	if `TRUE` all QTL but the one being tested are treated as a single polygenic effect, if `FALSE` (default) all QTL effect variances have to estimated.
`n.clusters`	number of parallel processes to spawn.
`plot`	a suffix for the file's name containing plots of every QTL profiling round, e.g. "profile" (default); if `NULL`, no file is produced.
`verbose`	if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.
`x`	an object of class `qtlpoly.profile` to be printed.
`pheno.col`	a numeric vector with the phenotype column numbers to be plotted; if `NULL`, all phenotypes from `'data'` will be included.
`sint`	whether `"upper"` or `"lower"` support intervals should be printed; if `NULL` (default), only QTL peak information will be printed.

Value

An object of class `qtlpoly.profile` which contains a list of `results` for each trait with the following components:

`pheno.col`	a phenotype column number.
`stat`	a vector containing values from score statistics.
`pval`	a vector containing p-values from score statistics.
`qtls`	a data frame with information from the mapped QTL.
`lower`	a data frame with information from the lower support interval of mapped QTL.
`upper`	a data frame with information from the upper support interval of mapped QTL.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

Qu L, Guennel T, Marshall SL (2013) Linear score tests for variance components in linear mixed models and applications to genetic association studies. *Biometrics* 69 (4): 883–92. doi:10.1111/biom.12095.

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using 'mappoly' package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # build null models
  null.mod <- null_model(data = data, n.clusters = 4, plot = "null")

  # perform forward search
  search.mod <- search_qtl(data = data, model = null.mod, w.size = 15, sig.fwd = 0.01,
    n.clusters = 4, plot = "search")

  # optimize model
  optimize.mod <- optimize_qtl(data = data, model = search.mod, sig.bwd = 0.0001,
    n.clusters = 4, plot = "optimize")

  # profile model
  profile.mod <- profile_qtl(data = data, model = optimize.mod, d.sint = 1.5,
    polygenes = FALSE, n.clusters = 4, plot = "profile")

## End(Not run)
```

qtl_effects *QTL allele effect estimation*

Description

Computes allele specific and allele combination additive effects from multiple QTL models.

Usage

```
qtl_effects(ploidy = 6, fitted)

## S3 method for class 'qtlpoly.effects'
plot(x, pheno.col = NULL, p1 = "P1",
  p2 = "P2")
```

Arguments

`ploidy` a numeric value of ploidy level of the cross (currently, only 4 or 6).

`fitted` a fitted multiple QTL model of class `qtlpoly.fitted`.

`x` an object of class `qtlpoly.effects` to be plotted.

`pheno.col` a numeric vector with the phenotype column numbers to be plotted; if `NULL`, all phenotypes from `'fitted'` will be included.

`p1` a character string with the first parent name, e.g. `"P1"` (default).

`p2` a character string with the second parent name, e.g. `"P2"` (default).

Value

An object of class `qtlpoly.effects` which is a list of `results` for each containing the following components:

`pheno.col` a phenotype column number.

`y.hat` a vector with the predicted values.

A **ggplot2** barplot with parental allele and allele combination effects.

Author(s)

Guilherme da Silva Pereira, <`gdasilv@ncsu.edu`>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi:`.

See Also

`read_data`, `remim`, `fit_model`

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
    d.sint = 1.5, n.clusters = 4, plot = "remim")

  # fit model
  fitted.mod <- fit_model(data=data, model=remim.mod, probs="joint", polygenes="none")

  # estimate effects
  est.effects <- qtl_effects(ploidy = 6, fitted = fitted.mod)
  plot(est.effects)

## End(Not run)
```

---

read_data *Read data*

Description

Reads files in specific formats and creates a `qtlpoly.data` object to be used in subsequent analyses.

Usage

```
read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno,
  step = 1)

## S3 method for class 'qtlpoly.data'
print(x, detailed = FALSE)
```

Arguments

`ploidy`	a numeric value of ploidy level of the cross.
`geno.prob`	an object of class `mappoly.genoprob` from **mappoly**.
`pheno`	a data frame of phenotypes (columns) with individual names (rows) identical to individual names in `geno.prob` object.

`step` a numeric value of step size (in centiMorgans) where tests will be performed, e.g. 1 (default); if `NULL`, tests will be performed at every marker.

`x` an object of class `qtlpoly.data` to be printed.

`detailed` if `TRUE`, detailed information on linkage groups and phenotypes in shown; if `FALSE`, no details are printed.

Value

An object of class `qtlpoly.data` which is a list containing the following components:

`ploidy` a scalar with ploidy level.

`nlgs` a scalar with the number of linkage groups.

`nind` a scalar with the number of individuals.

`nmrk` a scalar with the number of marker positions.

`nphe` a scalar with the number of phenotypes.

`lgs.size` a vector with linkage group sizes.

`cum.size` a vector with cumulative linkage group sizes.

`lgs.nmrk` a vector with number of marker positions per linkage group.

`cum.nmrk` a vector with cumulative number of marker positions per linkage group.

`lgs` a list with selected marker positions per linkage group.

`lgs.all` a list with all marker positions per linkage group.

`step` a scalar with the step size.

`pheno` a data frame with phenotypes.

`G` a list of relationship matrices for each marker position.

`Z` a list of conditional probability matrices for each marker position for genotypes.

`X` a list of conditional probability matrices for each marker position for alleles.

`Pi` a matrix of identical-by-descent shared alleles among genotypes.

Author(s)

Guilherme da Silva Pereira, `<gdasilv@ncsu.edu>`

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

`read_data`, `maps`, `pheno`

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

## End(Not run)
```

`remim` *Random-effect multiple interval mapping (REMIM)*

Description

Automatic function that performs REMIM algorithm using score statistics.

Usage

```
remim(data, pheno.col = NULL, w.size = 15, sig.fwd = 0.01,
  sig.bwd = 0.001, d.sint = 1.5, polygenes = FALSE,
  n.clusters = NULL, n.rounds = Inf, plot = "remim",
  verbose = TRUE)

## S3 method for class 'qtlpoly.remim'
print(x, pheno.col = NULL, sint = NULL)
```

Arguments

`data` an object of class `qtlpoly.data`.

`pheno.col` a numeric vector with the phenotype columns to be analyzed or printed; if `NULL` (default), all phenotypes from `'data'` will be included.

`w.size` the window size (in centiMorgans) to avoid on either side of QTL already in the model when looking for a new QTL, e.g. 15 (default).

`sig.fwd` the desired score-based significance level for forward search, e.g. 0.01 (default).

`sig.bwd` the desired score-based significance level for backward elimination, e.g. 0.001 (default).

`d.sint` a d value to subtract from logarithm of p-value ($LOP - d$) for support interval calculation, e.g. $d = 1.5$ (default) represents approximate 95% support interval.

`polygenes` if `TRUE` all QTL already in the model are treated as a single polygenic effect; if `FALSE` (default) all QTL effect variances have to estimated.

`n.clusters` number of parallel processes to spawn.

`n.rounds` number of search rounds; if `Inf` (default) forward search will stop when no more significant positions can be found.

`plot` a suffix for the file's name containing plots of every algorithm step, e.g. "remim" (default); if `NULL`, no file is produced.

`verbose` if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `qtlpoly.remim` to be printed.

`sint` whether `"upper"` or `"lower"` support intervals should be printed; if `NULL` (default), only QTL peak information will be printed.

Value

An object of class `qtlpoly.remim` which contains a list of `results` for each trait with the following components:

`pheno.col` a phenotype column number.

`stat` a vector containing values from score statistics.

`pval` a vector containing *p*-values from score statistics.

`qtls` a data frame with information from the mapped QTL.

`lower` a data frame with information from the lower support interval of mapped QTL.

`upper` a data frame with information from the upper support interval of mapped QTL.

Author(s)

Guilherme da Silva Pereira, <`gdasilv@ncsu.edu`>

References

Kao CH, Zeng ZB, Teasdale RD (1999) Multiple interval mapping for quantitative trait loci. *Genetics* 152 (3): 1203–16. `www.genetics.org/content/152/3/1203`.

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi`.

Qu L, Guennel T, Marshall SL (2013) Linear score tests for variance components in linear mixed models and applications to genetic association studies. *Biometrics* 69 (4): 883–92. `doi:10.1111/biom.12095`.

See Also

`read_data`

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # perform remim
  remim.mod <- remim(data = data, w.size = 15, sig.fwd = 0.01, sig.bwd = 0.0001,
    d.sint = 1.5, n.clusters = 4, plot = "remim")

## End(Not run)
```

---

search_qtl *QTL forward search*

Description

Searches for QTL and adds them one at a time to a multiple random-effect QTL model based on score statistics.

Usage

```
search_qtl(data, model, w.size = 15, sig.fwd = 0.01,
  polygenes = FALSE, n.rounds = Inf, n.clusters = NULL,
  plot = "search", verbose = TRUE)

## S3 method for class 'qtlpoly.search'
print(x, pheno.col = NULL)
```

Arguments

`data`	an object of class `qtlpoly.data`.
`model`	an object of class `qtlpoly.model` from which a forward search will start.
`w.size`	the window size (in cM) to avoid on either side of QTL already in the model when looking for a new QTL.
`sig.fwd`	the desired score-based *p*-value threshold for forward search, e.g. 0.01 (default).
`polygenes`	if `TRUE` all QTL but the one being tested are treated as a single polygenic effect; if `FALSE` (default) all QTL effect variances have to estimated.

`n.rounds` number of search rounds; if `Inf` (default) forward search will stop when no more significant positions can be found.

`n.clusters` number of parallel processes to spawn.

`plot` a suffix for the file's name containing plots of every QTL search round, e.g. "search" (default); if `NULL`, no file is produced.

`verbose` if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `qtlpoly.search` to be printed.

`pheno.col` a numeric vector with the phenotype column numbers to be printed; if `NULL`, all phenotypes from `'data'` will be included.

Value

An object of class `qtlpoly.search` which contains a list of `results` for each trait with the following components:

`pheno.col` a phenotype column number.

`stat` a vector containing values from score statistics.

`pval` a vector containing p-values from score statistics.

`qtls` a data frame with information from the mapped QTL.

Author(s)

Guilherme da Silva Pereira, <`gdasilv@ncsu.edu`>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. `doi:`.

Qu L, Guennel T, Marshall SL (2013) Linear score tests for variance components in linear mixed models and applications to genetic association studies. *Biometrics* 69 (4): 883–92. `doi:10.1111/biom.12095`.

See Also

`read_data`, `null_model`

Examples

```
## Not run:
# load raw data
data(maps)
data(pheno)

# estimate conditional probabilities using 'mappoly' package
library(mappoly)
genoprob <- lapply(maps, calc_genoprob)
```

```
  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # build null models
  null.mod <- null_model(data = data, n.clusters = 4, plot = ”null”)

  # perform forward search
  search.mod <- search(data = data, model = null.mod, w.size = 15, sig.fwd = 0.01,
    n.clusters = 4, plot = ”search”)

## End(Not run)
```

`simulate_qtl` *Simulations of multiple QTL*

Description

Simulate new phenotypes with a given number of QTL and creates new object with the same structure of class `qtlpoly.data` from an existing genetic map.

Usage

```
simulate_qtl(data, mu = 0, h2.qtl = c(0.3, 0.2, 0.1), var.error = 1,
  n.sim = 1000, w.size = 20, seed = 123, verbose = TRUE)

## S3 method for class 'qtlpoly.simul'
print(x, detailed = FALSE)
```

Arguments

`data` an object of class `qtlpoly.data`.

`mu` simulated phenotype mean, e.g. 0 (default).

`h2.qtl` vector with QTL heritabilities, e.g. c(0.3, 0.2, 0.1) for three QTL (default).

`var.error` simulated error variance, e.g. 1 (default).

`n.sim` number of simulations, e.g. 1000 (default).

`w.size` the window size (in centiMorgans) to avoid on either side of QTL already in the model when looking selecting new QTL, e.g. 20 (default).

`seed` integer for the `set.seed()` function.

`verbose` if `TRUE` (default), current progress is shown; if `FALSE`, no output is produced.

`x` an object of class `qtlpoly.sim` to be printed.

`detailed` if `TRUE`, detailed information on linkage groups and phenotypes in shown; if `FALSE`, no details are printed.

Value

An object of class qtlpoly.sim which contains a list of results with the same structure of class qtlpoly.data.

Author(s)

Guilherme da Silva Pereira, <gdasilv@ncsu.edu>

References

Pereira GS, Gemenet DC, Mollinari M, Olukolu BA, Wood JC, Mosquera V, Gruneberg WJ, Khan A, Buell CR, Yencho GC, Zeng ZB (2019) Multiple QTL mapping in autopolyploids: a random-effect model approach with application in a hexaploid sweetpotato full-sib population, *bioRxiv*. doi:.

See Also

read_data

Examples

```
  ## Not run:
  # load raw data
  data(maps)
  data(pheno)

  # estimate conditional probabilities using mappoly package
  library(mappoly)
  genoprob <- lapply(maps, calc_genoprob)

  # prepare data
  data <- read_data(ploidy = 6, geno.prob = genoprob, pheno = pheno, step = 1)

  # simulate new phenotypes
  sim.dat <- simulate_qtl(data = data, n.sim = 1000)

## End(Not run)
```

Index