Package 'qtlbim'

July 2, 2014

Version 2.0.7

Date 2013-04-15

Author Brian S. Yandell <yandell@stat.wisc.edu> and Nengjun Yi
<nyi@ms.soph.uab.edu>, with contributions from Tapan Mehta,Samprit Banerjee, Daniel Shriner and Ramprasad Venkataraman
(UA-Birmingham), Jee Young Moon, W. Whipple Neely (UW-Madison)
and Hao Wu and Randy von Smith (Jackson Laboratory).

Title QTL Bayesian Interval Mapping

Description Functions for model selection for genetic architecture.

Maintainer Brian S. Yandell <yandell@stat.wisc.edu>

Depends R (>= 2.10), stats, qtl (>= 1.9.43), lattice, coda, tools,MASS

License GPL (>= 2)

URL http://www.qtlbim.org

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-04-16 07:43:26

R topics documented:

`Bnapus` *Cross structure for complete Brassica napus data*

Description

Contains genotypes and phenotypes for Brassica napus study, including 0- 4- and 8-week vernalization, survival, and 19 chromosomes.

Usage

```
data(Bnapus)
```

Format

See `read.cross` in `library(qtl)` for format.

Details

Traits included are percent winter survival for 1992-3, 1993-4, 1994-5, 1997-8, and 1999-2000 (surv92, surv93, surv94, surv97, surv99, respectively), and days to flowering after no vernalization (flower0), 4 weeks vernalization (flower4) or 8 weeks vernalization (flower8). Percents are of plants alive in the Fall, taken from the middle of rows (totals unavailable). Days to flowering after transplant are averages over four replicates from a RCB design (values by block unavailable). First column has the trait name. The remaining columns identify individual DH line (302-455).

Marker genotype data for Major x Stellar double haploid (DH) population. Double haploids have the same relation of recombination to distance as backcrosses as there is just one meiosis tracked (in F1). However, DH are homozygous at every locus (usually mapped as RI0 lines). Marker genotypes are coded as M = Major, S = Stellar, - = missing. Data columns are `chrom` = B. napus chromosome (N1, N2, etc.); `order` = along chromosome; `cM` = istance from proximal end; `marker` = marker name (E = AFLP; *ec, *tg, *wg = RFLP; *xxx = other markers from Arabidopsis: Lem, eru1, eru2, fad3, isoDia, isoIdh, isoPgi, isoLap, pr2, slg6, Aca1, cor15). Remaining columns are for individual DH lines (identifier 302-455).

Source

Thomas C. Osborn, Department of Agronomy, UW-Madison (http://agronomy.wisc.edu).

References

http://www.stat.wisc.edu/~yandell/qtl/data/osborn/Bnapus

Ferreira ME, Satagopan J, Yandell BS, Williams PH, Osborn TC (1995) Mapping loci controlling vernalization requirement and flowering time in Brassica napus. Theor Appl Genet 90: 727-732. [original source and analysis]

JM Satagopan, BS Yandell, MA Newton and TC Osborn (1996) Markov chain Monte Carlo approach to detect polygene loci for complex traits. Genetics 144: 805-816. http://www.genetics.org/cgi/content/abstract/144/2/805 [first MCMC for experimental crosses; analysis of B. napus N2=LG9; see `vern` data]

Kole C, Thorman CE, Karlsson BH, Palta JP, Gaffney P, Yandell BS, Osborn TC (2001) Comparative mapping of loci controlling winter survival and related traits in oilseed Brassica rapa and B. napus. Molecular Breeding 1: 329-339. [refined map and reanalysis]

See Also

`read.cross`,`plot.qb`

Examples

```
data(Bnapus)
summary(Bnapus)
```

fisch *Eight QTL Stephens and Fisch simulated data*

Description

Contains genotypes and phenotypes for data simulated using model in Stephens and Fisch (1998) but with 90 percent heritability.

Usage

```
data(fisch)
```

Format

`fisch` is f2 (see `read.cross` for format).

Author(s)

Brian S. Yandell, mailto:yandell@stat.wisc.edu

Source

Patrick J. Gaffney (mailto:paga@lubrizol.com), Lubrizol Corp.

References

http://www.qtlbim.org

See Also

`read.cross`, `plot.qb`, `qb.mcmc`

Examples

```
data(fisch)
summary(fisch)
```

plot.qb *Diagnostics plots for Bayesian interval mapping*

Description

Diagnostic plots highlight putative QTL loci and effects as well as providing graphical model assessment tools.

Usage

```
## S3 method for class 'qb'
plot(x, ask = dev.interactive(), verbose = TRUE, ...)
## S3 method for class 'qb'
print(x, ...)
## S3 method for class 'qb'
summary(object, cutoff = 1, ...)
```

Arguments

x	An object of class qb.
object	An object of class qb.
verbose	Verbose summaries if TRUE.
ask	Ask before each plot if TRUE.
cutoff	Cutoff passed to qb.BayesFactor.
...	graphical parameters can be given as arguments to plot

Details

This generic plot routine takes an object of class qb created by qb.mcmc and produces plots via calls to several other plot routines. The generic summary produces a summary, while the generic print passes through to summary.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

plot, print, summary, qb.mcmc, qb.coda, qb.loci, qb.BayesFactor, qb.hpdone, qb.epistasis, qb.diag

Examples

```
data(qbExample)

plot(qbExample)
summary(qbExample)
```

`plot.qb.diag` *Marginal and model-conditional summaries of Bayesian interval mapping diagnostics*

Description

A density histogram is drawn for model-averaged summary diagnostics such as LOD, variance, or heritability.

Usage

```
qb.diag(qbObject, items= c("mean","envvar","var","herit"), ...)
## S3 method for class 'qb.diag'
plot(x, ... )
## S3 method for class 'qb.diag'
print(x, ... )
## S3 method for class 'qb.diag'
summary(object, digits = 5, ... )
```

Arguments

`qbObject`	Object of class `qb`.
`object`	Object of class `qb.diag`.
`x`	Object of class `qb.diag`.
`items`	Diagnostics to be summarized; must be name of a column in `element`.
`digits`	Number of significant digits.
`...`	Parameters to methods. Not used for `qb.diag`.

Details

Model-averaged density is smooth kernel estimate similar to ordinary histogram. A `boxplot` (without outliers) is overlaid for comparison with conditional boxplots. Conditional boxplots by number of QTL may show indication of model bias for small number of QTL. This and `qb.BayesFactor` can help suggest the minimal model. Diagnostic items that make sense to plot are `"LOD"`, `"envvar"` (environmental variance), `"herit"` (heritability), `"mean"` (grand mean), `"addvar"` (variance of add), `"domvar"` (variance of add). Marginal and conditional medians are printed.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

plot.qb, density, boxplot, qb.BayesFactor

Examples

```
data(qbExample)

temp <- qb.diag(qbExample)
summary(temp)
plot(temp)
```

---

plot.qb.epistasis *Density Plots for Models Showing Epistasis and GxE Interactions.*

Description

Produces density plots of models showing epsistasis (qb.epistasis) or GxE interactions (qb.intcov). The vertical axis shows magnitude of effect, horizontal axis shows chromosomes in epistatic pairs or covariate by chromosome. Paralell plots are produced for each of the entries in the effects parameter.

Usage

```
qb.epistasis(qbObject, effects = c("aa", "ad", "da", "dd"),
  cutoff = 1, maxpair = 5, pairs, ...)
qb.intcov(qbObject, covar, effects = c("add","dom"),
  cutoff = 1, nmax = 5, cov.chr, ...)
## S3 method for class 'qb.epistasis'
plot(x, effects, cex = 0.5, main, ... )
## S3 method for class 'qb.epistasis'
print(x, ... )
## S3 method for class 'qb.epistasis'
summary(object, ... )
```

Arguments

qbObject	An object of class qb.
object	Object of class qb.epistasis.
x	Object of class qb.epistasis.
cutoff	The cutoff parameter for number of epistatic pairs.
maxpair	Maximum number of epistatic pairs shown.

pairs	A character vector of chromosome pairs to examing for epistatic pairs. Chromosome names are separated by a dot.
covar	Covariate(s) to include; default is seq(nfixcov) where nfixcov is taken from qb.data.
nmax	Maximum number of covariate chromosomes shown.
cov.chr	A character vector of covariate by chromosome pairs to examing for GxE effects. Covariate names and chromosome names are separated by a dot.
effects	Character string of model effects.
cex	Horizontal expansion factor for characters in the plot. See par.
main	Main titles for plots; default is effects.
...	Arguments passed to generic plot.

Value

Returns a table of counts of epistatic pairs with counts above the cutoff value.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

Examples

```
data(qbExample)

temp <- qb.epistasis(qbExample)
summary(temp)
plot(temp)
temp <- qb.intcov(qbExample)
summary(temp)
plot(temp)
```

---

plot.qb.loci *Jittered plot of Bayesian QTL loci samples by chromosome*

Description

Each point is one locus from the Bayesian QTL estimates, plotted vertically by chromosome, jittered to give a sense of density. Separate colored vertical bands by loci element.

Usage

```
qb.loci(qbObject, loci = c("main", "epistasis", "GxE"), covar, ...)
## S3 method for class 'qb.loci'
plot(x, loci, labels = FALSE, amount = 0.35, cex, col, ...)
## S3 method for class 'qb.loci'
print(x, ...)
## S3 method for class 'qb.loci'
summary(object, digit = 1, ...)
```

Arguments

qbObject	Object of class qb.
loci	Character string identifying MCMC sample elements; may include "main", "epistasis", "GxE" and "all".
covar	Fixed covariate(s) for "GxE" loci; default is all fixed covariates involved in GxE interactions.
x	Object of class qb.loci.
object	Object of class qb.loci.
labels	Include marker labels if TRUE.
amount	Amount of jitter (between 0 and .45)
cex	Character expansion (may be invisible if too small–default based on number of MCMC samples).
col	Character string with colors named by loci; also includes color for marker lines.
digit	Number of digits for roundoff of loci quantiles.
...	Graphical parameters can be given as arguments to plot. Not used in qb.loci.

Details

Focuses attention on chromosome lengths and concentration of QTL loci estimates. Horizontal lines at markers. Separate bands by loci for each chromosome. Adjust amount and cex to modify look of plot. Most useful when looking at multiple chromosomes.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

jitter,subset.qb

Examples

```
data(qbExample)

temp <- qb.loci(qbExample)
plot(temp)
summary(temp)
temp <- qb.loci(qbExample, "all")
plot(temp)
summary(temp)
```

`plot.qb.mainmodes` *Summaries of QTL modes by chromosome.*

Description

Determine number of QTL per chromosome and estimate peaks and valleys.

Usage

```
qb.split.chr(qbObject, split, ...)
qb.epimodes(qbObject, cutoff = 1, nqtl, n.iter, pairloci, ...)
qb.mainmodes(qbObject, cutoff = 25, nqtl, n.iter, mainloci, ...)
## S3 method for class 'qb.mainmodes'
summary(object, digits = 4, ...)
```

Arguments

`qbObject`	Object of class `qb`.
`split`	List of split locations; names of list must correspond to chromosome names in `qbObject`. Default set by call to `qb.mainmodes` with addition arguments in `....`
`object`	Object of class `qb.mainmodes`.
`cutoff`	Cutoff for neglible number of QTL.
`nqtl`	Vector of number of QTL per chromosome.
`n.iter`	Number of iterations (for internal use only).
`mainloci`	Object containing `mainloci` data (for internal use only).
`pairloci`	Object containing `pairloci` data (for internal use only).
`digits`	Number of significant digits.
`...`	Parameters to `qb.mainmodes` or to methods.

Details

Cut off histogram of number of QTL per chromosome such that cumulative percent above number is less than `cutoff`. Once `nqtl` is determined or provided in call, divide MCMC samples using linear discriminant analysis (`lda`) and find peak locations per class. Use these peak locations to find locations of valleys between peaks. These valleys are used to divide MCMC samples into separate QTL for analysis. Currently this is used by `summary.qb.scanone` and `qb.multloci`.

`qb.split.chr` sets up a `split` for chromosomes for at valleys between inferred multiple QTL as an attribute of a returned `qbObject`. This is done by default when `qbObject` is created, and is stored as an attribute available as `qb.get(qbObject, "split.chr")`.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

`qb.multloci`, `summary.qb.scanone`

Examples

```
data(qbExample)

temp <- qb.mainmodes(qbExample)
summary(temp)
```

---

plot.qb.pairloci *Summaries of epistatic pairs of loci.*

Description

Summaries and detailed scatterplot showing all MCMC samples for epistatic pairs for selected chromosomes.

Usage

```
qb.pairloci(qbObject, chr, ...)
## S3 method for class 'qb.pairloci'
plot(x, main, cex = 0.75, ... )
## S3 method for class 'qb.pairloci'
print(x, ... )
## S3 method for class 'qb.pairloci'
summary(object, ... )
```

Arguments

qbObject	Object of class qb.
object	Object of class qb.pairloci.
x	Object of class qb.pairloci.
chr	Identifiers for one or two chromosomes.
main	Main title for plot.
cex	Character expansion of plot symbols.
...	Parameters to methods.

Details

Find pairs of loci in MCMC samples. Produce scatter plot with generic plot or show numerical summary. The plot provides position detail complementary to qb.multloci and qb.scantwo.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

plot.qb, qb.scantwo, qb.multloci

Examples

```
data(qbExample)

tmpar <- par(mfrow = c(2,2))
temp <- qb.pairloci(qbExample, c(1,2))
summary(temp)
plot(temp)
temp <- qb.pairloci(qbExample, c(1,3))
summary(temp)
plot(temp)
par(tmpar)
```

`plot.qb.scanone` *Plot or print qb.scanone object.*

Description

Plot or print marginal diagnostics of effects from a qb.scanone object.

Usage

```
## S3 method for class 'summary.qb.scanone'
print(x, digits = 3, ...)
## S3 method for class 'qb.scanone'
print(x, digits = 3, ...)
## S3 method for class 'qb.scanone'
plot(x, chr, scan, ylim, scan.name, ...)
```

Arguments

`x`	An object of class `qb.scanone`.
`digits`	Significant digits to round with print.
`chr`	Vector of chromosomes to plot. Must be integer.
`scan`	The model effects to include, the default is all those included in the `scanone` object `x`.
`ylim`	Vector of length 2 with vertical limits.
`scan.name`	Name of `scan` for plot; default is `"effects"` or comma-separated catenation of `scan`.
`...`	Other values passed to the generic plot function.

Details

This plot method uses `plot.scanone` as the engine to plot marginal posterior diagnostics created with `qb.scanone`. When there are multiple effects in `x`, these may be organized into one or several stacked plots using `scan`. The default for most diagnostics except counts is `scan = c("sum", "main", "epis")`. Counts and posterior diagnostics are typically plotted in two stacked plots. Individual columns from the `x` object can be plotted by specifying their names as a vector to option `scan`.

Arguments `col` and `lty` use keywords to names in `scan` argument. Default `main` and `sub` arguments are provided.

Value

Data frame with colors and line types used in plots.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

qb.scanone, summary.qb.scanone, plot.scanone

Examples

```
example(qb.scanone)
```

---

plot.qb.scantwo *Two dimensional LOD Plot of Epistasis/joint*

Description

Plots joint LOD for chromosomes on a two dimensional grid.

Usage

```
## S3 method for class 'qb.scantwo'
plot(x, chr, smooth = 3, main, offset,
  nodiag, slice = NULL, show.locus = TRUE,
  weight = c("sqrt","count","none","atten","ratten"), verbose = FALSE,
  split.chr, ...)
```

Arguments

x	An object of class qb.scantwo.
chr	Vector of chromosomes to plot. Must be integer.
smooth	Perform smoothing if > 0 using weighted average over smooth adjacent points.
main	Main title.
offset	Offset to make all values non-negative (see below).
nodiag	If TRUE do not include diagonal in plot.
slice	Take 1-D slice through 2-D surface is not NULL (see below).
show.locus	If a slice, show locus estimate if TRUE.
weight	Weights to use for nearest neighbor smoothing. sqrt is square root of count per locus. Used only if smooth > 0.
verbose	Give verbose feedback if TRUE.
split.chr	Split summary by multiple QTL per chromosome (see details for plot.qb.scanone).
...	Other parameters passed to generic plot function.

Details

The `offset` is used only if `qb.scantwo` used `type = "estimate"` to make values for plotting all non-negative. Values are rescaled by `offset` so that the origin is at 1 and, by default, the min and max are at 0 and 2, respectively, for each half of the plot. We need this at this time because `plot.scantwo` does not allow negative values.

The `plot.scantwo` argument `nodiag` is set to ensure values are all shown and not modified by `plot.scantwo`. Plots with different values for `nodiag` or `lower` than the defauls may be non-sensical. For instance, passing `lower = "cond-int"` produces much white area on the image.

A non-null `slice` yields a 1-D view of the 2-D surface. The plots for slices use `plot.scantwo`. The elements of the slice vector are: `chr` = Chromosome number to slice on; `upper` = Focus on upper triangle of 2-D if `TRUE`; `start` = Start position in chromosome `chr` (default = 0); `end` = End position in chromosome `chr` (default = end of chromosome); `weight` = Type of weighted mean across `chr` by number of MCMC samples: 0 = unweigthed; 1 = uniform weighting; 2 = position-specific weighting (default).

Value

The scantwo object being plotted.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

`qb.scantwo`, `plot.scantwo`

Examples

```
example(qb.scantwo)
```

`qb-internal` *Internal qtlbim routines*

Description

These are internal qtlbim routines that are made visible in the namespace for technical use.

Usage

```
covar.mean(qbObject, adjust.covar, verbose = FALSE, pheno.col)
pull.grid(qbObject, offset, spacing, mask.region, cross, step,
  off.end, stepwidth, drop.duplicates, ...)
pull.loci(cross, step, off.end, stepwidth, region)
qb.cross(qbObject, genoprob = TRUE, ...)
qb.cross.class(qbObject)
qb.demo()
qb.get(qbObject, element, pheno.col, warn = TRUE, ...)
qb.load(cross, qbObject, dir, file)
qb.save(cross, qbObject, dir, Name)
qb.reorder(qbObject, warn = FALSE, pheno.col)
```

Arguments

qbObject	Object of class qb.
adjust.covar	Adjustments to covariates. Default is NA, which adjusts by covariate mean values. Values are assumed to be in order of fixed covariates.
verbose	Verbose mode if TRUE.
cross	Object of class cross (see read.cross).
offset	Offset by first marker if TRUE.
step	Argument to create.map.
off.end	Argument to create.map.
stepwidth	Argument to create.map.
drop.duplicates	Drop duplicate chr.pos after roundoff if TRUE.
region	Argument to create.map.
genoprob	Compute genotype probabilities with qb.genoprob if TRUE.
spacing	Add columns for map, eq.spacing and xchr if TRUE. This corresponds to map element of a scantwo object.
mask.region	Subset genome regions if TRUE (see subset.qb).
element	Character string for element of qbObject to get. Typically this is a parameter to qb.data, qb.model or qb.mcmc, or it is one of the MCMC sample files in output.dir, from c("iterdiag", "mainloci", "pairloci", "covariates", "gbye").
pheno.col	Numeric or character identifier for phenotype. Useful eventually for multiple traits.
warn	Warn user if qbObject is legacy format if TRUE.
dir	Character string name of directory for load if qbObject does not exist.
file	Character string name of file for load if qbObject does not exist.
Name	Character string name of suffix for save if qbObject exists.
...	Additional arguments.

Details

These are all internal routines. But some may be useful beyond.

qb.demo is called in demo(qb.tour) and provides an interactive selection of the R/qtlbim demos.

qb.cross extracts the cross object associated with qbObject. qb.get is the internal main routine for extracting information from a qbObject. As stated elsewhere, currently qbObject refers to objects that are critical to it but not part of it: the cross object used to create it and the MCMC samples in files in output.dir.

covar.mean finds covariate means or adjusts them to user-supplied values.

pull.grid pulls the grid of pseudomarkers from the cross object associated with qbObject. The option spacing determines whether this is in a format similar to scanone (FALSE) or scantwo (TRUE). It is used qb.get when accessing external MCMC sample files and by several other routines that require pseudomarker information, notably genotype probabilities.

qb.reorder is called by qb.mcmc to create pointers to reorder the MCMC samples so that chromosome numbers and positions within chromosomes are in increasing order. It creates the subset element of a qb object.

qb.save and qb.load are depricated. They were used to save and retrieve MCMC samples used in examples and vignettes. We now use save and data.

Examples

```
data(qbExample)

covar.mean(qbExample)
qb.get(qbExample, "output.dir")
summary(qb.cross(qbExample))
temp <- qb.get(qbExample, "iterdiag")
dim(temp)
names(temp)

## Not run:
## The following should have no effect.
qbExample <- qb.reorder(qbExampleb)

## You can call the following rather than demo() to get a tour.
qb.demo()

## End(Not run)
```

---

qb.arch *Build genetic architecture with chromosomes, positions and epistatic pairs.*

Description

These routines work in conjunction with qb.hpdone, qb.scantwo, qb.sliceone and step.fitqtl to infer the number, pattern and position of QTL from MCMC samples.

Usage

```
qb.arch(object, ...)
## Default S3 method:
qb.arch(object, chr, pos, tolerance = 10, ...)
## S3 method for class 'step.fitqtl'
qb.arch(object, main, epistasis, ...)
## S3 method for class 'qb.BestPattern'
qb.arch(object, ...)
## S3 method for class 'qb.arch'
summary(object, ...)
## S3 method for class 'qb.arch'
print(x, ...)
```

Arguments

object	Object for appropriate method: summary of object of class qb.scantwo for default; object of class step.fitqtl.
x	Object of class qb.arch.
chr	Vector of chromosome numbers.
pos	Vector of positions on chromosomes (much be same length as chr).
tolerance	Minimum distance for two QTL to be considered distinct.
main	Vector of chromosome identifiers with only main effects.
epistasis	Data frame with a 2-element vector of chromosome identifiers for each epistatic pair.
...	Not used here.

Details

Extract architecture in terms of chromosomes and positions of main QTL and identifiers of epistatic pairs of QTL. The step.fitqtl approach is used to compare an automatic fit to a user-defined set of main chromosomes and epistatic pairs.

Value

qtl	Data frame with main QTL as chr and pos.
by.num	Data frame with epistatic pairs indexed by chromosome number, labeled qtla and qtlb.
by.chr	List with elements chr and pos showing epistatic pairs. These elements are data frames with chromosomes and positions for each epistatic pair: rows are QTL number, columns are qtla and qtlb.
by.set	List of connected sets of epistatic chromosomes.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

step.fitqtl, qb.sweave, qb.best

Examples

```
data(qbExample)

## Run qb.scantwo and get summary to use in qb.arch
temp <- summary(qb.scantwo(qbExample, type = "2logBF"),
  threshold = c(upper = 10))
## qb.arch default use.
cross.arch <- qb.arch(temp, chr = c(1,1,2,3), pos = c(15,45,12,15))
cross.arch
```

---

qb.BayesFactor *Bayesian model selection via Bayes factors.*

Description

Model-averaged posteriors and Bayes factors computed for number and pattern of QTL, chromosomes and pairs of chromosomes showing epistasis.

Usage

```
qb.bf(...)
qb.BayesFactor(qbObject, items = c("nqtl","pattern","chrom","pairs"),
  cutoff.pattern, cutoff.pairs = 1, nmax = 15, epistasis = TRUE, ...)
## S3 method for class 'qb.BayesFactor'
plot(x, ...)
## S3 method for class 'qb.BayesFactor'
summary(object, sort = TRUE, digits = 3, ...)
## S3 method for class 'qb.BayesFactor'
print(x, ...)
```

Arguments

qbObject	An object of class qb.
object	Object of class qb.BayesFactor.
x	Object of class qb.BayesFactor.
items	Items to include in model selection assessment. These are nqtl = Number of QTLs; pattern = Pattern of QTL across chromosomes as comma-separated chromosome numbers and colon-separated chromosome pairs; chrom = Chromosome; pairs = Epistatic pairs of chromosomes.

cutoff.pattern	Percent cutoff for pattern inclusion in model selection. Default is 0.25 (0.5) if epistasis is TRUE (FALSE).
cutoff.pairs	Percent cutoff for epistatic pair inclusion in model selection.
nmax	Maximum number of model terms included per item (for items "pattern" and "pairs" only).
epistasis	Include epistasis in patterns if TRUE.
sort	Sort by Bayes factor if TRUE.
digits	Number of significant digits for summary.
...	Additional arguments passed to generic plot, summary or print.

Details

qb.BayesFactor (or qb.bf for short) creates model selection results for selected items. These are based on marginal posteriors and priors, averaged over all other model parameters. The posterior may be influenced by prior, while Bayes factors are empirically less sensitive for QTL model selection. The Bayes factors are computed relative to the smallest term for each item, using the ratios of posterior/prior. Any pair of model terms can be compared as the ratio of their Bayes factors.

Value

List with items, each containing:

posterior	Posterior frequency of MCMC samples.
prior	Prior frequency.
bf	Rank-ordered Bayes factors relative to smallest value.
bfse	Approximate standard error for bf computed using binomial variance of MCMC samples.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

plot.qb, qb.mcmc

Examples

```
data(qbExample)

temp <- qb.BayesFactor(qbExample)
summary(temp)
plot(temp)
```

qb.BestPattern *Proximity of common genetic architecture patterns.*

Description

Multidimensional scaling and hierarchical clustering of most common patterns of genetic architecture.

Usage

```
qb.best(...)
qb.BestPattern(qbObject, epistasis = TRUE,
  category = c("pattern", "nqtl"), cutoff, score.type =
  c("sq.atten","attenuation","variance","recombination","distance"),
  include = c("nested","all","exact"),
  center = c("median","mean"), level = 5, ...)
## S3 method for class 'qb.BestPattern'
plot(x, type = c("mds", "hclust"),
  main, xlab, method = "complete", cluster = 3, cexmax = 5,
  colmax = 75, cex, col,
  symbol = c("pattern","nqtl","cluster","c@n","c@p","n@p","c@n@p"), ...)
## S3 method for class 'qb.BestPattern'
summary(object, method = "complete",
  cluster = 3, n.best = 1, ...)
```

Arguments

qbObject	Object of class qb.
x,object	Object of class qb.BestPattern.
epistasis	Include epistasis in patterns if TRUE.
category	Distances indexed by nqtl or pattern.
cutoff	Percent cutoff for pattern inclusion in model selection. Default is 0.25 (0.5) if epistasis is TRUE (FALSE).
score.type	Type of score to use as distance. See qb.close.
type	Plot dendrogram for hclust or 2-D multidimensional scaling projection for mds.
main	Main plot title as character string.
xlab	Character string for horizontal (x) axis.
method	Method for hierarchical clustering.
cluster	Number of clusters desired.
n.best	Number of better models to display.
cexmax	Maximum font size (minimum is set to 1); patterns are displayed in mds plot proportional to their posterior probability.
colmax	Maximum number of colors.

cex	Manual override of font size for mds plot; should be length 1 or the number of patterns exceeding cutoff.
col	Colors for plotting.
symbol	Plot symbol for mds plot. Shorthand using at sign @ signifies catenation of two or more symbols into one.
include	Action for model averaging of chromosome-specific locus and explained variance: use all MCMC samples that match the chromosome; use only MCMC samples for patterns that have the target pattern nested within them; or use only MCMC samples with the exact same target pattern.
center	Method of estimating the center for locus and explained variance.
level	Confidence level as percent between 0 and 100 for loci and variance contributions.
...	Parameters to methods.

Details

This uses the closeness measure from qb.close to compute a similarity matrix among patterns whose posterior probabilities exceed cutoff. Distance = 1 - similarity is used for hierarchical clustering or multidimensional scaling.

The best pattern is chosen as the one with highest posterior mean; all other patterns are compared to that pattern in terms of the score.type. This best pattern is a natural target for qb.close.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

qb.close

Examples

```
data(qbExample)

temp <- qb.BestPattern(qbExample)
summary(temp, n.best = 3)
plot(temp, type = "hclust")
plot(temp)
plot(temp, symbol = "c@n")

best <- summary(temp)$best
temp <- qb.close(qbExample, best)
summary(temp)
plot(temp)
```

qb.close *Measures closeness of genetic architectures to target.*

Description

Boxplots and summaries of how close MCMC samples of genetic architectures are to target architecture.

Usage

```
qb.close(qbObject, target = NULL, epistasis = TRUE, signed = FALSE,
  score.type = c("sq.atten","attenuation","variance","recombination",
  "distance"), ...)
## S3 method for class 'qb.close'
plot(x, category = c("pattern", "nqtl"), xlab,
  cutoff, sort.pattern = c("percent","score"), ...)
## S3 method for class 'qb.close'
summary(object, cutoff, digits = 0,
  show = "score", ... )
```

Arguments

qbObject	Object of class qb.
x,object	Object of class qb.close.
target	Target architure as data frame with columns chrom and locus. Extracted from summary if target is a qb.scanone object. If target = NULL, score.type is set to "variance" to be compared with null model.
epistasis	Include epistasis in patterns if TRUE.
signed	Sign score. Most useful to examine single chromosome.
score.type	Type of score to use as distance.
category	Boxplots indexed by nqtl or pattern.
xlab	Label for X axis (default taken from x object).
cutoff	Percent cutoff for pattern inclusion in model selection. Default is 0.25 (0.5) if epistasis is TRUE (FALSE).
sort.pattern	If type = "pattern", sort by percent posterior or by median of score.
digits	Number of digits displayed for locus.
show	Character string with name from object to show.
...	Parameters to methods.

Details

Closeness for each loci is measured as `1-2r`, with `r` the recombination rate. Thus unlinked loci have measure 0. Loci between a MCMC sample architecture and the target architecture on the same chromosome are matched by closest distance in cM between subsets of the the same length (if target has 2 QTL on chr 3 and sample has 3, consider all pairs from sample to find closest pair in 2-D). Measure per sample is sum across all loci. A quick way to generate a target is to use `qb.BestPattern`.

The `score.type` is `"recombination"` = r, the recombination rate; `"attenuation"` = 1-2r; `"sq.atten"` = squared attenuation, `"distance"` in cM, or genetic `"variance"`.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

`qb.BestPattern`, `qb.hpdone`

Examples

```
data(qbExample)

## Here target is known for simulated data.
target <- data.frame(chrom = c(1,1,2,3),
  locus = c(15,45,12,15))
temp <- qb.close(qbExample, target)
summary(temp)
plot(temp)
plot(temp, category = "nqtl")

## Or use target from qb.BestPattern
target <- summary(qb.BestPattern(qbExample))$best
qb.close(qbExample, target)
```

---

qb.coda *Coerce to an MCMC object for use with the coda package.*

Description

This function creates an object of class `mcmc` from an object of class qb produced by the package `R/qtlbim`.

Usage

```
qb.coda(qbObject, element, variables, ...)
```

Arguments

qbObject	An object of class qb returned by calling the function qb.mcmc.
element	A character string which has to one of "iterdiag", "mainloci", "pairloci", "covariates", or "gbye"; default is "iterdiag".
variables	A vector of integers specifying the columns or column names of element to be considered. Details about the columns can be found in qb.mcmc
...	Additional arguments to pass along (mostly ignored).

Details

This package requires the package coda.

Value

An object of class mcmc. This object could be used to analyze the MCMC output using R/coda.

Author(s)

Dr. Nengjun Yi, et al., nyi@ms.ssg.uab.edu

References

http://www.qtlbim.org

See Also

mcmc, qb.mcmc, plot.qb

Examples

```
data(qbExample)

## Default plots for iteration diagnostics "iterdiag".
temp <- qb.coda(qbExample)
plot(temp)

## Summaries for some "mainloci" elements.
temp <- qb.coda(qbExample, "mainloci")
plot(temp)
```

qb.confound *Examine confounding of covariate with pseudomarkers.*

Description

Covariates used in gene mapping may be correlated with covariates. These routines examine the pattern of confounding.

Usage

```
qb.confound(qbObject, covar = 1)
## S3 method for class 'qb.confound'
plot(x, ylim, main, ...)
## S3 method for class 'qb.confound'
print(x, ...)
## S3 method for class 'qb.confound'
summary(object, ...)
```

Arguments

qbObject	Object of class qb.
x	Object of class qb.confound.
object	Object of class qb.confound.
covar	Index to covariate
ylim	Limits for y (vertical) plotting axis.
main	Title for plot.
...	Additional parameters passed alone.

Details

This examines possible confounding between a covariate and pseudomarkers across the genome. Confouding, evidenced by large correlation with a marker, would raise suspicions about mapping in a genomic region, unless of course the covariate is a marker in that region. Blue curves are correlation with additive effect; red curves are correlation with dominance effect. Dashed lines at 5 percent significance limits.

Value

qb.confound returns a matrix with columns for:

coradd	Correlation with additive pseudomarker effect.
cordom	Correlation with dominance pseudomarker effect (if F2).
chr	Chromosome identifier.

The object inherits from scanone objects.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

qb.mcmc

Examples

```
data(qbExample)

temp <- qb.confound(qbExample)
plot(temp)
```

---

qb.covar *Examine GxE effect of covariates on main genetic effects.*

Description

Compare main effects with GxE effects to address correlation of estimates.

Usage

```
qb.covar(qbObject, element = "add", covar = 1, adjust.covar, chr, ...)
## S3 method for class 'qb.covar'
summary(object, percent = 5, digits = 3, ...)
## S3 method for class 'qb.covar'
print(x, ...)
## S3 method for class 'qb.covar'
plot(x, percent = 5, cex, include.zero = TRUE, ...)
```

Arguments

qbObject	Object of class qb.
object	Object of class qb.covar.
x	Object of class qb.covar.
element	Main effect to examine ("add" or "dom").
covar	Index to covariates used in MCMC samples.
adjust.covar	Adjustments to covariates. Default is NA, which adjusts by covariate mean values. Values are assumed to be in order of fixed covariates.
chr	Subset of chromosomes as integer vector.

percent	Percentile (0 to 100) for summaries.
digits	Number of significant digits to print.
cex	Character expansion for plots (default decreases with MCMC sample size).
include.zero	Include zero values in plot when TRUE.
...	Arguments passed through to inherited routines.

Details

The diagonal dark green line of points on plots by chromosome indicate adjustment for covariates that have not been centered. Main effects are generally less correlated with GxE when covariates are first centered to have mean zero.

Value

Objects of class qb.covar have three columns: main effect, GxE effect and chromosome. Summary objects have eight columns, three for main effect and GxE (mean, lower and upper percentile), followed by correlation and p-value. Summaries are done by chromosome.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

qb.mcmc

Examples

```
data(qbExample)

temp <- qb.covar(qbExample)
summary(temp)
plot(temp)
```

---

qb.data *Prepares data for qb.mcmc*

Description

This function selects trait(s) and covariates from a cross object to build a model (qb.model) for MCMC (qb.mcmc).

Usage

```
qb.data(cross, pheno.col = 1, trait = c("normal","binary","ordinal"),
  censor = NULL, fixcov = c(0), rancov = c(0), boxcox = FALSE,
  standardize = FALSE, ...)
```

Arguments

cross	An object of class cross. See read.cross for details.
pheno.col	the column number for the phenotype used by model. Currently, only one phenotype can be analyzed at a time.
trait	Type of the quatitative trait or dependent variable: "normal" or "binary" or "ordinal".
censor	Matrix of censor values with 2 columns and nind(cross) rows. Details needed here.
fixcov	list of fixed covariates. The column number(s) in cross$pheno which is(are) considered as fixed covariates.
rancov	list of random covariates.The column number(s) in cross$pheno which is(are) considered as random covariates.
boxcox	Indicates whether to use a Boxcox transformation for the dependent variable or not: TRUE or FALSE. Note: trait has to be "normal" and all phenotypic values have to be positive for using this option.
standardize	Indicates whether to standardize the dependent variable or not: TRUE or FALSE. Note: trait has to be "normal" to use this option.
...	Extra terms not used.

Details

This function picks the relevant part of the data from the cross object and prepares data for qb.model and qb.mcmc It can also standardize or transform continuous data if specified.

Value

yvalue	vector of the values of the dependent variable.
ncategory	number of category type if it is non-normal data.
envi	environment effect: TRUE or FALSE.
nfixcov	number of fixed covariates.
nrancov	number of random covariates.
fixcoef	values of the fixed covariate(s) for all individuals.
rancoef	values of the random covariate(s) for all individuals.
nran	number of categories defining the random covariate.
lamda	value of lamda, the transformation parameter for the boxcox transformation.

Note

This function returns a list and hence should have a differenct name from that of the cross object.

Author(s)

Dr. Nengjun Yi, et al., nyi@ms.ssg.uab.edu

References

http://www.qtlbim.org

See Also

qb.genoprob, qb.model, qb.mcmc

Examples

```
qbData <- qb.data(cross, pheno.col = 3, rancov = 2, fixcov = 1)
```

---

qb.genoprob *Grid point and genotype probability computation method*

Description

This function is used to compute putative QTL positions and genotypic probabilities at these positions. The genotypic probabilities for missing marker genotypes are also computed.

Usage

```
qb.genoprob(cross, map.function, step = 2,
  tolerance = 1e-6, stepwidth = "variable", ...)
```

Arguments

cross	An object of class cross. See read.cross for details.
map.function	Indicates what map function to use when converting genetic distances to recombination fractions. See calc.genoprob.
step	Distance (in cM) between positions at which putative QTL positions and their genotype probabilities are calculated. However, specifying step = 0 would assume marker positions as putative QTL locations and genotypic probabilities would be calculated only for markers with missing genotype.
tolerance	Minimum separation of markers enforced by jittermap.
stepwidth	Type of stepwidth for calc.genoprob; "variable" works best with MCMC sampling.
...	Extra arguments to pass to calc.genoprob.

Value

qb.genoprob first ensures marker separation is at least tolerance, and then computes genotype probabilities at pseudomarkers spaced approximately step units apart using calc.genoprob. See calc.genoprob for value.

Author(s)

Dr. Nengjun Yi, et al., nyi@ms.ssg.uab.edu

References

http://www.qtlbim.org

See Also

jittermap, calc.genoprob.

Examples

```
## calculate grids and genotypic probabilites
cross <- qb.genoprob(cross,map.func="haldane",step=2)
```

---

qb.hpdchr *Find highest probability density (HPD) region by Chromosome.*

Description

Find area under posterior by chromosome for genome-wide HPD region.

Usage

```
qb.hpdchr(qbObject, level = 0.5, height, hpd, chr, smooth = 3)
```

Arguments

qbObject	Object of class qb.
level	Value between 0 and 1 of HPD coverage. Ignored if height is supplied.
height	Height of posterior corresponding to level. Determined from level and hpd by default.
hpd	Object of class qb.hpdone. Created internally be default.
chr	Chromosomes to include; default determined by genome-wide HPD region.
smooth	Degree of smoothing.

Details

Determine 100*level percent HPD coverage by chromosome for a given height threshold.

Value

A List with the following elements:

`hpd.height`	Height of HPD threshold. Name of `hpd.height` is the `level`.
`chr.posterior`	Posterior probability by chromosome as percent.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

`qb.scanone`

Examples

```
data(qbExample)

qb.hpdchr(qbExample)
```

---

qb.hpdone *Highest probability density (HPD) region.*

Description

Determine HPD region across genome, including position of posterior mode.

Usage

```
qb.hpdone(qbObject, level = 0.5, profile = "2logBF",
  effects = "cellmean", scan = "sum", chr, smooth = 3, ...)
## S3 method for class 'qb.hpdone'
summary(object, chr, digits = 3, ...)
## S3 method for class 'qb.hpdone'
print(x, ...)
## S3 method for class 'qb.hpdone'
plot(x, chr, ...)
```

Arguments

qbObject	Object of class qb.
object	Object of class qb.hpdone.
x	Object of class qb.hpdone.
level	Value between 0 and 1 of HPD coverage.
scan	Elements to scan; usually one of "sum", "mean", "epistasis", "GxE".
smooth	Degree of smoothing.
chr	Chromosomes to include; default determined by HPD region.
effects	Effects are "cellmean" for means by genotype; "estimate" for estimates of Cockerham main effects.
profile	Objective profile for plot; default is "2logBF"; other choices found in option type for qb.scanone.
digits	Number of digits for round.
...	Extra parameters passed along to plot.

Details

Determine 100*level percent HPD region. Subset chromosomes based on HPD region. Create genome scans for profile and effects.

Value

qb.hpdone is a list with a hpd.region summary matrix and qb.scanone objects for the profile and effects. A summary of a qb.hpdone object yields a matrix with columns for

chr	chromosome number
n.qtl	estimated number of QTL on chromosome
pos	estimated position of QTL
lo.nn%	lower nn% HPD limit
hi.nn%	upper nn% HPD limit
profile	Peak of profile, identifed by the profile type.
effects	Columns for the effects, appropriately labeled.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

qb.scanone, qb.hpdchr

Examples

```
data(qbExample)

temp <- qb.hpdone(qbExample)
summary(temp)
plot(temp)
```

---

qb.mcmc *Bayesian Mutiple Interacting QTL mapping using MCMC*

Description

A computationally efficient MCMC algorithm using the Gibbs sampler or Metropolis-Hastings algorithm is used to produce posterior samples for QTL mapping.

Usage

```
qb.mcmc(cross, data, model, mydir = ".", n.iter = 3000, n.thin = 20,
  n.burnin = 0.01*n.iter*n.thin,
  genoupdate = TRUE, seed = 0, verbose = TRUE, ...)
```

Arguments

cross	An object of class cross. See read.cross for details.
data	List returned by calling the function qb.data .
model	List returned by calling the function qb.model.
mydir	A directory to save output from qb.mcmc in several '*.dat' files. A directory is created using the trait name and the system time and date. If no directory is specified, the default directory is the current working directory.
n.iter	number of iterations to be saved in mydir, the default being 3000. Note that, n.iter is not the total number of iterations performed but the number iterations saved or considered as posterior samples for future analysis. The actual number of iterations would be n.burnin + n.iter*n.thin
n.thin	the thinning number which must be a positive number (default=40)
n.burnin	the initial burn-in period, i.e number of iterations to discard at the beginning of the MCMC run default being 0.01*n.iter*n.thin.
genoupdate	=TRUE will update QTL genotypes and =FALSE will not do so and use the expected value of the QTL genotypes.
seed	Specifies the seed for the random number generator. Using the same seed for two runs of the qb.mcmc function will generate the exact same output. The seed needs to be an integer. The default value for seed is the system time.
verbose	=TRUE will force periodic output of the number of MCMC iterations saved. The location of the output directory where results are stored and the time taken for the MCMC run will also be displayed to the user.
...	Paramters passed to qb.data or qb.model if data or model, respectively, is not provided.

Details

A composite model space approach to develop a Bayesian model selection framework for identifying interacting QTL for complex traits in experimental crosses from two inbred lines. By placing a liberal constraint on the upper bound of the number of detectable QTL we restrict attention to models of fixed dimension. Either Gibbs sampler or Metroplis-Hastings algorithm can be applied to sample from the posterior distribution.

The following data frames in the `mcmc.samples` element of the `qb` object contain the MCMC samples. They are used by many other routines.

The iterations data frame `iterdiag` has `n.iter` rows and 5 major columns: `niter` = iteration number; `nqtl` = number of putative QTLs included; `mean` = overall mean; `envvar` = residual variance; `var` = total genetic variance. Depending on the type of cross, presence of covariates and epistatic effects there would be more columns in the following order: `varadd` = variance of all additive effects; `vardom` = variance of all dominant effects; `varaa` = variance of all additive-additive interactions; `varad` = variance of all additive-dominant interactions; `varda` = variance of all dominant-additive interactions; `vardd` = variance of all dominant-dominant interactions. Values for variance of environment-additive interaction, variance of environment-dominant interaction, and variance of environment effect have names that encode the covariate.

Covariates are in data frame `covariates`, with `n.iter` rows and L+M(`length(fixcov)+length(rancov)`) columns: `L columns` : Coefficient of the fixed effect. `M columns` : Variance of the random effect. If an ordinal trait is analyzed, the cutoff points for the threshold model are also included in additional columns. There would be C-3 bounded threshold values for an ordinal phenotype with C categories.

The mainloci data frame has N rows (N=sum of number of QTLs detected in `n.iter` iterations) and 6-8 columns: `niter` = iteration number; `nqtl` = number of putative QTLs included; `chrom` = chromosome number; `locus` = locus in cM; `add` = additive effect; `dom` = dominance effect (if included); `varadd` = variance of additive effect; `vardom` = variance of dominant effect (if included).

The pairloci data frame has N rows (N=sum of number of pairs of QTLs with epistatic effect detected) and 8-14 columns: `niter` = iteration number; `n.epis` = number of epistatic pairs included; `chrom1` = first chromosome number; `locus1` = first locus in cM; `chrom2` = second chromosome number; `locus2` = second locus in cM; `aa` = additive-additive effect; `ad` = additive-dominant effect (if included); `da` = dominant-additive effect (if included); `dd` = dominant-dominant effect (if included); `varaa` = variance of additive-additive interaction; `varad` = variance of additive-dominant interaction (if included); `varda` = variance of dominant-additive interaction (if included); `vardd` = variance of dominant-dominant interaction (if included).

The gbye (Gene by Environment) data frame has 7-9 columns: `niter` = iteration number; `n.gbye` = number of GxE terms included; `covar` = fixed covariate identifier; `chrom` = chromosome number; `locus` = locus in cM; `add` = additive effect; `dom` = dominance effect (if included); `varadd` = variance of additive effect; `vardom` = variance of dominant effect (if included).

The deviance data frame has 1 column with the posterior deviance. There is one deviance value for each iteration, or `n.iter` values. The last value is the deviance calculated at the posterior means, known as Dhat.

Value

Returns a list of class `qb`, including:

`args`	Arguments passed to `qb.mcmc`, `qb.data` and `qb.model`. An additional element for `subset` may be added by `subset.qb` if called.

cross.object	A clean-ed version of the original cross object, but only with phenotypes used by qb.mcmc.
mcmc.samples	A list containing the MCMC samples for each phenotype. There are always data frames for iterdiag and mainloci, with optional data frames for pairloci, covariates and gbye.

Author(s)

Nengjun Yi, nyi@ms.ssg.uab.edu

References

http://www.qtlbim.org

See Also

qb.sim.cross, qb.data, qb.model, qb.mcmc

Examples

```
## Not run:
example(qb.sim.cross)

## Calculate grids and genotypic probabilites.
cross <- qb.genoprob(cross, step=2)

## Create MCMC samples
## First line as qb.data options; second line has qb.model options.
qbExample <- qb.mcmc(cross, pheno.col = 3, rancov = 2, fixcov = 1,
  chr.nqtl = rep(3,nchr(cross)), intcov = 1, interval = rep(10,3),
  n.iter = 1000, n.thin = 20)

## End(Not run)
```

---

qb.meancomp *Examine grand mean and covariate MCMC samples.*

Description

Examine grand mean and covariate Monte Carlo samples to glean estimates of data center and importance of covariates.

Usage

```
qb.meancomp(qbObject, adjust.covar, ...)
## S3 method for class 'qb.meancomp'
summary(object, percent = 5, ...)
## S3 method for class 'qb.meancomp'
print(x, ...)
```

```
## S3 method for class 'qb.meancomp'
plot(x, covar, percent = 5, cex, ...)
```

Arguments

qbObject	Object of class qb.
adjust.covar	Adjustments to covariates. Default is NA, which adjusts by covariate mean values. Values are assumed to be in order of fixed covariates.
object	Object of class qb.meancomp.
x	Object of class qb.meancomp.
percent	Percentile between 0 and 100 for summaries.
covar	Sequence of covariate identifiers for plot.
cex	Character expansion for plot symbols. Default shrinks with number of MCMC iterations.
...	Extra parameters passed along.

Details

Grand mean is adjusted to mean level of covariates. Diagonal of scatterplot matrix includes density plot. Setting `covar = 0` yields a density plot for the grand mean alone.

Value

qb.meancomp is a matrix with columns for the grand mean and for each fixed covariate. Summaries show mean and upper and lower percentiles.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

qb.mcmc

Examples

```
data(qbExample)

temp <- qb.meancomp(qbExample)
summary(temp)
plot(temp)
```

qb.model *Set up interacting QTL model for qb.mcmc*

Description

This function sets up a genome-wide interacting QTL model by specifying global constraints on models and priors on unknowns.

Usage

```
qb.model(cross, epistasis = TRUE, main.nqtl = 3,
  mean.nqtl = main.nqtl + 3, max.nqtl = NULL, interval = NULL,
  chr.nqtl = NULL,intcov = c(0), depen = FALSE,
  prop = c(0.5, 0.1, 0.05), contrast = TRUE, ...)
```

Arguments

cross	An object of class cross. See read.cross for details.
epistasis	indicates if epistasis is included in the model: TRUE or FALSE
main.nqtl	prior expected number of main effect QTLs.
mean.nqtl	prior expected number for all QTLs on all chromosomes including QTLs with main effects, epistatic effects and gene-environment interactions.
max.nqtl	maximum number of QTLs allowed in the model. Default is $l + 3\sqrt{l}$ where l is main.qtl for non-epistatic model and mean.qtl for epistatic model.
interval	minimum distance between any two flanking QTLs for all chromosomes. Default is the average distance between markers in each chromosome.
chr.nqtl	list of the maximum number of QTLs allowed to be detected on each chromosome. Default is the length of the chromosome divided by interval.
intcov	logical or 0/1 vector for fixed covariates indicating which gene-environment interaction will be considered (default is all FALSE, no GxE).
depen	=TRUE will use dependent prior for indicator variables of epistatic effects.
prop	prior inclusion probabilities for epistatic effects in three different scenarios: when both (default 0.5), one (0.1) or none (0.05) of the main effects of the two interacting QTL are included in the model. Note that the sum of the probabilities need not be equal to 1 and prop should be specified only when depen=TRUE.
contrast	Use Cockerham model if TRUE; otherwise estimate genotypic values.
...	Not used.

Details

This function defines the model for Bayesian QTL mapping using qb.mcmc. This model considers two-way interaction as the highest level of both gene-gene and gene-environment interactions.

Value

qtl_envi	Indicates if there is an interaction between the QTLs and environmental variables: TRUE or FALSE.

Note

This function returns a list and hence should have a differenct name from that of the cross object.

Author(s)

Dr. Nengjun Yi, et al., nyi@ms.ssg.uab.edu

References

http://www.qtlbim.org

See Also

qb.data, qb.genoprob, qb.mcmc

Examples

```
qbModel <- qb.model(cross, chr.nqtl = rep(3,nchr(cross)), intcov = 1,
  interval = rep(10,3))
```

---

qb.multloci *Summaries of multiple loci on a chromosome.*

Description

Summaries and up to four plots showing loci found in MCMC samples for a chromosome.

Usage

```
qb.multloci(qbObject, chr = 1, cutoff = 25, nqtl, ...)
## S3 method for class 'qb.multloci'
plot(x, amount = 0.5, cex,
  split = TRUE, contour = TRUE, weight = TRUE, merge = TRUE, ...)
## S3 method for class 'qb.multloci'
summary(object, merge = TRUE, ...)
```

Arguments

qbObject	Object of class qb.
x,object	Object of class qb.multloci.
chr	Identifier for one chromosome.
cutoff	Smallest posterior probability for nqtl (ignored if nqtl provided).
nqtl	Number of QTL on chromosome (inferred by default).
amount	Amount to jitter points.
cex	Character expansion of plot symbols.
split	Split plots into four panels on one page if TRUE. Otherwise plot each panel separately. The split may be a numeric vector with values 1:4 signifying which panels to show. See details.
contour	Contour plot overlaid on pairs if TRUE.
weight	Inversely weight loci in density plot by number of QTL if TRUE.
merge	Merge across number of QTL if TRUE. Otherwise, show separate summary or plot by number of QTL. See details.
...	Parameters to qb.mainmodes or to methods.

Details

Find multiple loci in MCMC samples for chromosome chr. The number of QTL, nqtl is inferred from the histogram as the largest number of QTL above the percent cutoff.

The generic plot command produces the following plots: (1) density plot of main QTL grouped by QTL; (2) histogram of number of QTL; (3) density plot of epistatic pairs; (4) scatter plot of pairs of QTL. The density plots are divided into nqtl groups. The scatter plot shows pairs of main loci below diagonal and epistatic pairs above using codes corresponding to the number of QTL per sample; note that 3 QTL have 3 pairs, 4 QTL have 6, etc., and that solitary QTL are displayed along the diagonal.

split and merge control the manner of plotting. Setting merge to FALSE yields only density plots for main loci conditioned on the number of QTL per sample. Setting split to FALSE or to numbers between 1 and 4 yields plots on separate pages.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

plot.qb, qb.scantwo, qb.mainmodes

Examples

```
data(qbExample)

temp <- qb.multloci(qbExample, 1)
summary(temp)
plot(temp)
plot(temp, merge = FALSE)
summary(temp, merge = FALSE)
```

`qb.remove` *Legacy update, remove or recover of qb object and associated MCMC samples*

Description

Old (2006, R/qtlbim version 2.6) style qb objects had important external objects, namely the `cross` object and the MCMC samples in flat files that were loaded as needed. `qb.legacy` upgrades to the new (2007) style qb object. The commands `qb.remove` and `qb.recover` are useful to remove and restore old style qb objects.

Usage

```
qb.legacy(qbObject, remove = FALSE, ...)
qb.remove(qbObject, verbose = TRUE, external.only = FALSE)
qb.recover(cross, traitName, output.dir, n.thin = 40, n.burnin,
  algorithm = "M-H", genoupdate = FALSE, ...)
```

Arguments

`qbObject`	Object of class qb (see `qb.mcmc`).
`remove`	Remove external MCMC samples if `TRUE`. This will not remove the `cross` object associated with the `qbObject`. Be sure to remove the `qbObject` itself separately.
`verbose`	Print warning if `TRUE`.
`external.only`	Remove only external MCMC samples if `TRUE`.
`cross`	Object of class `cross` (see `read.cross`).
`traitName`	Character string name of trait to recover.
`output.dir`	Character string with name of output directory (inferred if missing).
`n.thin`	Thining of MCMC chain used in `qb.mcmc`.
`n.burnin`	Burnin of MCMC chain used in `qb.mcmc`.
`algorithm`	Algorithm of MCMC chain used in `qb.mcmc`.
`genoupdate`	Genotype update flag for MCMC chain used in `qb.mcmc`.
`...`	Options passed to `qb.data` and `qb.model`.

Details

At the present time, qb.mcmc stores MCMC samples in external files located in directory output.dir, whose name is typically the traitName followed by the date. qb.remove removes this directory along with the qbObject. qb.recover attempts to recover the use of an orphaned output.dir after a crash of R. These are fragile routines.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

qb.mcmc, qb.genoprob

Examples

```
## Not run:
## Upgrade legacy qb object.
myqbObject <- qb.legacy(myqbObject)

## Recover qbExample for trait "bp" of cross "hyper" using default output.dir.
qbExample <- qb.recover(hyper, "bp")

## Remove internal qbExample and external output.dir.
qb.remove(qbExample)

## End(Not run)
```

---

qb.scanone *Genome Scan for Main Loci Involved in Phenotypic Trait*

Description

This method extracts iteration diagnostics and mainloci from the qb object and returns a data frame (of class qb.scanone) containing information on environmental variance, explained variance components, non-epistatic variance components.

Usage

```
qb.scanone(qbObject, epistasis = TRUE, scan, type.scan, covar, adjust.covar,
  chr, sum.scan = "yes", min.iter = 1, aggregate = TRUE, smooth = 3,
  weight = c("sqrt","count","none","atten","ratten"), split.chr,
  center.type = c("mode","mean","scan"), half = FALSE, verbose = FALSE, ...)
```

Arguments

qb0bject	An object of class qb.
epistasis	If TRUE then information about epistasis is included.
scan	Vector of diagnostics to scan (see below).
type.scan	Type of scan; default is "heritability" (see below).
covar	Covariate(s) to include; default is seq(nfixcov) where nfixcov is taken from qb.data. Set to 0 to exclude any covariates.
adjust.covar	Adjustments to covariates. Default is NA, which adjusts by covariate mean values. Values are assumed to be in order of fixed covariates.
chr	Chromosomes to subset on if not NULL.
sum.scan	Sum over scan diagnostics if "yes" or "only"; only report sum if "only".
min.iter	Include only samples at loci if minimum number of iterations is at least min.iter; default is to include all (min.iter = 1).
aggregate	Aggregate effects into main, epis, gbye if TRUE.
half	Cut epistatic effects in half if TRUE.
smooth	Degree of nearest neighbor smoothing to determine maxima.
weight	Weights to use for nearest neighbor smoothing. sqrt is square root of count per locus. Used only if smooth > 0.
center.type	Method to find QTL loci. See details.
split.chr	Split summary by multiple QTL per chromosome (see details).
verbose	Give verbose feedback if TRUE.
...	Additional arguments mostly ignored.

Details

The type.scan specifies what type of scan is performed. Scan produces marginal estimates of diagnostics at each potential loci across the genome. That is, values are adjusted for other possible QTL simply by taking the marginal average over MCMC samples. Choices of type.scan are "heritability", "LPD", "LR", "deviance", "detection", "variance", "estimate", "cellmean", "count", "log10", "posterior", "logposterior" (i.e. log10(posterior)), "BF", "2logBF" (i.e. 2*ln(BF)), and "nqtl" (number of linked QTL). Default is "LPD".

Type "heritability" is actually R-squared at this point, not the theoretical heritability. Types "LPD", "LR" and "deviance" are all proportional to each other in the usual sense; "LPD" is computed to agree with lod from scanone if models were restricted to one QTL and missing genotypes are imputed. Detection is the marginal posterior probability of detectio of a QTL at a locus. Types "variance" and "estimate" yield, respectively, the marginal variance components and the marginal parameter estimates at each loci. Type "cellmean" gives marginal estimates for A, H, B genotypes (these are single character codes for AA, AB, BB, respectively). The remaining count types provide diagnostics. Types "count" and "log10" report on number of MCMC samples in raw or logged scale. Type "posterior" ("logposterior") yields the marginal (log) posterior probability. Type "BF" ("2logBF") gives the marginal Bayes factor per loci; both are proportional to "count". Type "nqtl" gives the average number of linked loci, which can be useful in sorting out multiple linked loci.

The scan specifies the model effects to include for all types except the counts. Aggregated effects (default except for type "cellmean") are "main", "epistasis" and "GxE" (genotype by environment). Individual model effects can be requested as "add", "dom", "aa", "ad", "da", "dd". In addition, GxE terms, if present are included automatically if covar is not 0. For type "estimate", main effects for "add" and "dom" are adjusted for any covariate GxE effects. The sum.scan is used for all types but the counts to get a summary across scan effects.

The "mode" and "mean" centering rely on the mode and mean, respectively, of the posterior; the "scan" centering uses the mode of the actual type used to create the qb.scanone object. The "scan" agrees with an scanone summary method for "pos" and "sum" columns. However, the mode for a "scan" could be in a region of low posterior mass and may not be reliable as such. Note that mean can be biased when there are linked loci. Only used in qb.scanone summary.

Evidence for linked loci leads to multiple summary lines per chromosome. Be default, a qbObject has inferred chromosome splits based on MCMC samples (see qb.split.chr). This is determined in a similar manner to qb.multloci. In particular, the arguments cutoff and nqtl documented in qb.multloci would adjust whether and how many linked loci may be considered. These apply across all chromosomes being summaried.

Value

Returns an object of class qb.scanone (a data frame) containing effects selected according to type.scan and scan.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

summary.qb.scanone, plot.qb.scanone

Examples

```
data(qbExample)

temp <- qb.scanone(qbExample)
summary(temp)
plot(temp)
```

qb.scantwo *Genome Scan for Pairs of Loci Involved in Phenotypic Trait*

Description

This method extracts iteration diagnostics and pair loci from the qb object and returns a data frame (of class qb.scanone) containing information on environmental variance, explained variance components, epistatic and non-epistatic variance components.

Usage

```
qb.scantwo(qbObject, epistasis = TRUE, scan, type.scan,
  upper.scan = "epistasis", lower.scan = "full", covar,
  adjust.covar, chr, min.iter = 1, verbose = FALSE, ...)
```

Arguments

qbObject	An object of class qb.
epistasis	If TRUE information on epistasis is included in the return value.
scan	List of diagnostics to scan (see below).
type.scan	Vector of two scan types; default is "heritability" (see below).
upper.scan	Vector of diagnostics to scan for upper triangle (see below).
lower.scan	Vector of diagnostics to scan for lower triangle (see below).
covar	Covariate(s) to include; default is seq(nfixcov) where nfixcov is taken from qb.data. Set to 0 to exclude any covariates.
adjust.covar	Adjustments to covariates. Default is NA, which adjusts by covariate mean values. Values are assumed to be in order of fixed covariates.
chr	Chromosomes to subset on if not NULL.
min.iter	Include only samples at loci if minimum number of iterations is at least min.iter; default is to include all (min.iter = 1).
verbose	Give verbose feedback if TRUE.
...	Additional arguments mostly ignored.

Details

The scan and type.scan are similar to those used in qb.scanone. However, here scan is a list and type.scan is a vector, each with elements "lower" and "upper". You can either specify scan as a list, or profide upper.scan and lower.scan separately.

The scan defaults for types other than counts to list(upper = "epistasis", lower = "full"); you can modify the list scan or the separate options upper.scan and lower.scan. The string "epistasis" is short-hand for the epistatic effects, c("aa", "ad", "da", "dd"). The string "full" is shorthand for the epistatics effects plus main effects, c("add", "dom"), plus any GxE terms.

The type.scan defaults to c(upper = "LPD", lower = "LPD"). See qb.scanone for the range of possible types. Mostly the 2-D version of type.scan provides marginal summaries for pairs of loci. However, for type "nqtl", the marginal summaries involving main effects (e.g. with scan values "full" or "main" or "add" or "dom") show, for each pair of chromosomes, the average number of QTL at both chromosomes.

Value

Returns an object of class qb.scantwo (a data frame) containing effects selected according to type.scan and scan.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

plot.qb.scanone

Examples

```
data(qbExample)

temp <- qb.scantwo(qbExample)
summary(temp)
plot(temp)
```

---

qb.sim.cross *Simulates QTL related data for an F2 or BC cross.*

Description

This function is used to simulate genotypic, phenotypic and covariate data for BC and F2 populations. The underlying genetic model is Cockerham's model and data for both continuous (normally distributed only) and ordinal traits can be generated.

Usage

```
qb.sim.cross(len = rep(100,20), n.mar = 11, eq.spacing = TRUE,
  n.ind = 400, type = c("f2","bc","riself","risib"), missing.geno = 0.0,
  missing.pheno = 0.0, ordinal = c(0.5,0.5),
  qtl.pos = NULL, qtl.main = NULL, qtl.epis = NULL,
  covariate = NULL, gbye = NULL, seed = NULL )
## S3 method for class 'qb.sim'
summary(object, ...)
```

Arguments

len	defines the length (in cM) of each chromosome and number of chromosomes. Thus `len = c(80,90,44)` would represent a model with three chromosomes of lengths 80, 90, and 44 respectively.
n.mar	The number of markers per chromosome. This can be specified as a single number or as a vector. If a single number is specified, all the chromosomes will have the same number of markers. If n.mark is a vector then it must have the same number of entries as there are chromosomes. For example, if n.mar = c(10,11,9) then we have a three chromosome model in which the first chromosome has 10 markers, the second has 11 and the third has 9.a vector specifying the number of markers per chromosome.
eq.spacing	if TRUE, markers will be equally spaced. Default is TRUE. If FALSE, markers are generated uniformly over the chromosome.
n.ind	specifies the number of individuals.
type	indicates whether to simulate an intercross ("f2") or a backcross ("bc").
missing.geno	the frequency of missing genotypes.
missing.pheno	the frequency of missing phenotypic values.
ordinal	define the probabilities of each ordinal category and the number of elements in the vector will determine the number of categories. The elements must be positive and the should sum up to 1
qtl.pos	This parameter specifies the positions of qtl as a matrix with dimensions (number of qtl) x 2. Note that the row dimension is the number of qtl and is not the number of chromosomes. Each row identifies a qtl, the first column entries represent the chromosome's index, the second column entries represent the location on the chromosome of the qtl. The (row) order in which qtl are listed in this parameter is the index by which they are identified later on in the parameters `qtl.main` and `qtl.epi`.
qtl.main	The parameter `qtl.main` is a matrix specifying the main effects of QTLs. The first column gives the qtl-index (the row index of the qtl in the `qtl.pos` parameter.), the second and third column gives the additive and dominance effects, respectively. There are two or three columns depending on `type` being "bc" or "f2".
qtl.epis	It is a matrix specifying epistatic effects. There are 3 or 6 columns depending on `type` being "bc" or "f2". Each row gives an epistatic pair. The first entry in a row gives the first qtl index, the second entry represents the index of the second qtl. The other entries give the value of the epistatic effects (additive-additive, additive-dominance, dominance-additive and dominance-dominance) of the two qtls. The indices used to represent the qtl are the row indices of the `qtl.pos` matrix which correspond to the first and second qtl in each epistatic pair.
covariate	A vector of two elements, the first being the true value of the coffecient for the fixed covariate and the second the true value for the standard deviation of the random covariate.

gbye	A matrix specifying the interaction between the fixed covariate and QTL main effect. The first column is the index of the QTL, the other column(s) is(are) the value(s) of interaction(s).
seed	Set pseudo-random number seed with set.seed if not NULL.
object	An object of class qb.sim, typically the qtl element of a cross object created by qb.sim.cross.
...	Not used here.

Details

The most important difference of this simulation function from others is that it computes phenotype values with full genetic model. i.e. both additive, dominance, and espistatic effects are considerred. Furthermore, environmental effects and gene-environment interactions can be included in the model to simulate phenotypes.

The outputted genotypes for markers and qtls will be coded as 1 and 2 for BC and 1,2, and 3 for F2. Missing data will be coded as NA.

Value

qb.sim.cross returns an object of class cross (see read.cross for details) with two additional components, qtl and gvalue. gvalue is a vector of predicted values of the same length as the phenotype pheno.normal. qtl is of class qb.sim is added which is a list of atmost 6 components depending on the options specified.

geno	is a matrix of true QTL genotypes for every individual and each locus. The genotypes are defined following read.cross.
pos	is a matrix of true QTL position. Same as qtl.pos.
herit.main	is a matrix of the heritability of main effects. nrow(qtlherit.main)=no. of QTLs and ncol(qtlherit.main)=2 or 3 depending on the type of genetic cross ("bc" or "f2"). The first column being the QTL index and the others being additive and dominant heritability respectively.
herit.epis	is a matrix of the heritability of epistatic effects. nrow(qtlherit.epis)=no. of QTLs pairs interacting and ncol(qtlherit.main)=3 or 6 depending on the type of genetic cross ("bc" or "f2"). The first column being the QTL index and the others being additive-additive, additive-dominant, dominant-additive and dominant-dominant heritability respectively.
herit.cov	is a vector of length 2 containing the heritability of the fixed and random covariate.
herit.gbye	is a matrix of heritability of GxE interactions. nrow(qtlherit.gbye)= no. of GxE interactions and ncol(qtlherit.gbye)= 2 or 3 depending on the type of genetic cross ("bc" or "f2"). The first column being the GxE index and the others being additive and dominant GxE interaction heritability.

Author(s)

Dr. Nengjun Yi, et al., nyi@ms.ssg.uab.edu

References

http://www.qtlbim.org

See Also

qb.genoprob, qb.data qb.model, qb.mcmc, sim.cross

Examples

```
## Not run:
## Simulate large intercross.
cross <- qb.sim.cross(len = rep(100,20), n.mar = 11, eq.spacing =FALSE,
  n.ind = 500, type = "f2", ordinal = c(0.3,0.3,0.2,0.2),
  missing.geno = 0.03, missing.pheno = 0.03,
  qtl.pos = rbind(qtl.1=c(chr=1,pos=15),qtl.2=c(1,45),qtl.3=c(3,12),
    qtl.4=c(5,15),qtl.5=c(7,15),qtl.6=c(10,15),qtl.7=c(12,35),qtl.8=c(19,15)),
  qtl.main = rbind(main.1=c(qtl=1,add=0.5,dom=0),main.2=c(2,0,0.7),
    main3=c(3,-0.5,0),main4=c(4,0.5,-0.5)),
  qtl.epis = rbind(epis1=c(qtl.a=4,qtl.b=5,aa=-0.7,ad=0,da=0,dd=0),
    epis2=c(6,8,0,1.2,0,0)),
  covariate = c(fix.cov=0.5,ran.cov=0.07),
  gbye = rbind(GxE.1=c(qtl=7,add=0.8,dom=0)) )

## Examine simulation information.
summary(cross$qtl)

## End(Not run)

## Simulate small backcross.
cross <- qb.sim.cross(len = rep(60,3), n.mar = 7, eq.spacing =FALSE,
  n.ind = 100, type = "bc", ordinal = c(0.3,0.3,0.2,0.2),
  missing.geno = 0.03, missing.pheno = 0.03,
  qtl.pos = rbind(qtl.1=c(chr=1,pos=15), qtl.2=c(1,45),
                  qtl.3=c(2,12), qtl.4=c(3,15)),
  qtl.main = rbind(main.1=c(qtl=1,add=1.5), main.2=c(2,0),
                   main3=c(3,-1), main4=c(4,0)),
  qtl.epis = rbind(epis1=c(qtl.a=2,qtl.b=3,aa=-2), epis2=c(2,4,3)),
  covariate = c(fix.cov=0.5,ran.cov=0.07),
  gbye = rbind(GxE.1=c(qtl=3,add=2)))
summary(cross$qtl)
```

---

qb.sliceone *Genome Slice to detect QTL for Phenotypic Trait*

Description

This method extracts iteration diagnostics and mainloci from the qb object and returns a data frame (of class qb.sliceone). Generic summary and plot can be used for display.

Usage

```
qb.sliceone(qbObject, slice, epistasis = TRUE, scan, type.scan, covar,
  adjust.covar, chr, sum.scan = "yes", min.iter = 1,
  aggregate = TRUE, smooth = 3, weight = c("sqrt","count","none","atten","ratten"),
  split.chr, center.type = c("mode","mean","scan"), verbose = FALSE, ...)
## S3 method for class 'qb.sliceone'
summary(object, chr, ...)
## S3 method for class 'qb.sliceone'
print(x, ...)
## S3 method for class 'qb.sliceone'
plot(x, ..., scan, auto.par = TRUE)
```

Arguments

qbObject	An object of class qb.
object	Object of class qb.sliceone.
x	Object of class qb.sliceone.
slice	Chromosomes to slice upon.
epistasis	If TRUE then information about epistasis is included.
scan	Vector of diagnostics to scan (see below).
type.scan	Type of scan; default is "heritability" (see below).
covar	Covariate(s) to include; default is seq(nfixcov) where nfixcov is taken from qb.data. Set to 0 to exclude any covariates.
adjust.covar	Adjustments to covariates. Default is NA, which adjusts by covariate mean values. Values are assumed to be in order of fixed covariates.
chr	Chromosomes to subset on if not NULL.
sum.scan	Sum over scan diagnostics if "yes" or "only"; only report sum if "only".
min.iter	Include only samples at loci if minimum number of iterations is at least min.iter; default is to include all (min.iter = 1).
aggregate	Aggregate effects into main, epis, gbye if TRUE.
smooth	Degree of nearest neighbor smoothing to determine maxima.
weight	Weights to use for nearest neighbor smoothing. sqrt is square root of count per locus. Used only if smooth > 0.
split.chr	Split summary by multiple QTL per chromosome (see details for plot.qb.scanone).
center.type	Method to find QTL loci. See details.
verbose	Give verbose feedback if TRUE.
auto.par	Automatic setting of plot parameters for multiple plots if TRUE.
...	Arguments to be passed along.

Details

All arguments except slice agree with qb.scanone. The slice specifies a chromosome upon which to slice, yielding a 1-D scan of what might be seen on a 2-D scan using qb.scantwo. One advantage of qb.sliceone is that you can get 2-QTL cell means for the slice chromosome and the scanned chromosomes.

The summary invokes summary.qb.scanone to summarize slice by chromosome. The plot will by default give separate plots for each slice genotype and use plot.qb.scanone to scan the chromosomes. If scan is specified for plot.qb.sliceone, then those elements will be plotted. For instance, plot(x,scan="slice") will plot the running average locus on the slice chromosome with respect to the other chromosomes.

Value

qb.sliceone returns an object of class qb.sliceone (a data frame) containing effects selected according to type.scan and scan.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

summary.qb.scanone, plot.qb.scanone

Examples

```
data(qbExample)

## Get profile of heritability.
temp <- qb.sliceone(qbExample, slice = 1, chr = 2:3)
summary(temp)
plot(temp)

## Get profile of cell means.
temp <- qb.sliceone(qbExample, slice = 1, chr = 2:3, type.scan = "cellmean")
summary(temp)
plot(temp)
```

qb.slicetwo *Slices for epistatic pairs.*

Description

These routines refine QTL positions for epistatic pairs and show plots to reveal the nature of epistasis.

Usage

```
qb.slicetwo(qbObject, chr, pos, type.scan = "2logBF", width = 10, ...)
## S3 method for class 'qb.slicetwo'
summary(object, ...)
## S3 method for class 'qb.slicetwo'
print(x, ...)
## S3 method for class 'qb.slicetwo'
plot(x, byrow = TRUE, figs, auto.par = TRUE, col, lty, ...)
```

Arguments

qbObject	Object of class qb.
object	Object of class qb.slicetwo.
x	Object of class qb.slicetwo.
chr	Chromosome vector.
pos	Position vector corresponding to chr.
type.scan	Type of profile scan; see qb.scanone.
width	Width of slice.
byrow	Arrange plots by row (for slides) if TRUE.
figs	Plot only selected figures. Full set of c("profile", "effects", "cellmean", "effectplot") is default.
auto.par	Automatic setting of plot parameters for multiple plots if TRUE.
col,lty	colors and line types for cellmean and interaction plots
...	Extra plot options.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

qb.scantwo, qb.sliceone

Examples

```
data(qbExample)

temp <- qb.slicetwo(qbExample, chr = c(1,2), pos = c(45,12))
summary(temp)
plot(temp)
```

---

qb.sweave *Run sweave to automate QTL search with MCMC samples.*

Description

This routine runs a separate Sweave file (*.Rnw) of commands, making substitutions for the user-supplied data and thresholds. It can be used to automate the search for genetic architecture.

Usage

```
qb.sweave(cross, pheno.col = 1, n.iter = 3000, n.draws = 64,
  scan.type = "2logBF", hpd.level = 0.5,
  upper.threshold, SweaveFile, SweaveExtra, PDFDir, remove.qb = TRUE)
```

Arguments

cross	Object of class cross.
pheno.col	Phenotype column in object cross.
n.iter	Number of MCMC iterations to be stored.
n.draws	Number of MC draws to use for fitqtl.
scan.type	Type of 1-D and 2-D scan to perform; see qb.scanone.
hpd.level	Highest probability density level for scan; see qb.hpdone.
upper.threshold	Threshold for upper triangle (epistasis) in 2-D scan; see qb.scantwo.
SweaveFile	Name of Sweave file (default is system.file("doc", "prototype_qtl_hyper_data.Rnw", package =
SweaveExtra	Name of user-supplied extra Sweave file (default is NULL).
PDFDir	Name of directory to store PDF files (default is phenoPDF, where pheno is the name associated with phenotype pheno.col).
remove.qb	Remove constucted objects if TRUE.

Details

This is a simple shell around the Sweave routine to create customized documents with embedded QTL analysis. The default file system.file("doc", "prototype.qtl.hyper.slides.Rnw", package = "qtlbim") creates a "beamer" style PDF slide show. An alternative file system.file("external", "prototype.qtl.hyper.paper.Rn creates a preprint document. Both require post-processing with pdflatex.

A user-defined section can be added to the automated documents, using the SweaveExtra option. We have provided system.file("external", "hyper.slide.extra.Rnw", package = "qtlbim") for the slide version and system.file("external", "hyper.paper.extra.Rnw", package = "qtlbim") for the preprint version.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

Sweave

Examples

```
## Not run:
data(hyper)

## Create default slide show LaTeX source without extra section.
qb.sweave(hyper)

## Turn LaTeX into PDF. Run twice to get outline correct.
## Need pdflatex on your system.
system("pdflatex prototype.qtl.hyper.slides")
system("pdflatex prototype.qtl.hyper.slides")

## Create document form, with extra section.
qb.sweave(hyper,
  SweaveFile = system.file("external", "prototype.qtl.hyper.paper.Rnw", package = "qtlbim"))
  SweaveExtra = system.file("external", "hyper.paper.extra.Rnw",
    package = "qtlbim"))
system("pdflatex prototype.qtl.hyper.paper")
system("pdflatex prototype.qtl.hyper.paper")

## End(Not run)
```

---

qb.varcomp *Variance components for Bayesian multiple QTL*

Description

These routines extract and summarize variance components for Bayesian multiple QTL. Variance components are averaged over genome loci. Covariates and GxE may be included.

Usage

```
qb.varcomp(qbObject, scan, aggregate = TRUE, ...)
## S3 method for class 'qb.varcomp'
summary(object, ...)
## S3 method for class 'qb.varcomp'
print(x, ...)
```

```
## S3 method for class 'qb.varcomp'
plot(x, log = TRUE, percent = 5, cex, ...)
```

Arguments

qbObject	Object of class qb.
object	Object of class qb.varcomp.
x	Object of class qb.varcomp.
scan	Aggregated terms to include in created object (see below).
aggregate	Sum over individual components of aggregated terms if TRUE.
log	Use log10 of variances in plot if TRUE.
percent	Percentile between 0 and 100 for summaries.
cex	Character expansion for plot symbols. Default shrinks with number of MCMC iterations.
...	Arguments to pass along.

Details

Variance components are organized as "main" ("add" and "dom"), "epistasis" ("aa", etc.), "fixcov" (for all fixed covariate terms), "rancov" (random covariates), and "GxE" (genotype by environment, including additive and dominance terms). Any subset of these may be chosen.

Value

qb.varcomp creates a matrix with columns of samples for the variance components. Each row represents an MCMC iteration. Values are averaged over loci.

Author(s)

Brian S. Yandell

References

http://www.qtlbim.org

See Also

qb.mcmc

Examples

```
data(qbExample)

temp <- qb.varcomp(qbExample)
summary(temp)
plot(temp)
```

qbExample *MCMC sample for fake BC example*

Description

Retrieve or recreate MCMC samples used in qtlbim.pdf document and demos.

Usage

```
data(qbExample)
```

Details

This example is used in vignettes and demos. See vignette `qtlbim.pdf` or see `qtlbim.Rnw` in doc folder of package.

See Also

`qb.genoprob`, `qb.mcmc`, `qb.sim.cross`

Examples

```
data(qbExample)
summary(qbExample)

## Not run:
## Fake BC Example.
set.seed(1234)
cross <- qb.sim.cross(len = rep(60,3), n.mar = 7, eq.spacing =FALSE,
       n.ind = 100, type = "bc", ordinal = c(0.3,0.3,0.2,0.2),
       missing.geno = 0.03, missing.pheno = 0.03,
       qtl.pos = rbind(qtl.1=c(chr=1,pos=15),qtl.2=c(1,45),qtl.3=c(2,12),qtl.4=c(3,15)),
       qtl.main = rbind(main.1=c(qtl=1,add=1.5),main.2=c(2,0),
         main3=c(3,-1),main4=c(4,0)),
       qtl.epis = rbind(epis1=c(qtl.a=2,qtl.b=3,aa=-2),
         epis2=c(2,4,3)),
       covariate = c(fix.cov=0.5,ran.cov=0.07),
       gbye = rbind(GxE.1=c(qtl=3,add=2)))
summary(cross$qtl)
cross <- qb.genoprob(cross, step=2)
qbExample <- qb.mcmc(cross, pheno.col = 3, rancov = 2, fixcov = 1,
                     chr.nqtl = rep(3,nchr(cross)), intcov = 1, seed = 1216,
                     interval = rep(10,3), n.iter = 1000, n.thin = 20)

save("qbExample", file = "qbExample.RData")

## End(Not run)
```

qbHyper *MCMC samples for hyper dataset from R/qtl*

Description

Retrieve MCMC samples for hyper dataset from R/qtl. Code provided to recreate qbHyper if desired. These samples are used in vignettes and demos.

Usage

```
data(qbHyper)
```

See Also

hyper, qb.genoprob, qb.mcmc

Examples

```
data(qbHyper)
summary(qbHyper)
## Not run:
## Here is code to generate qbHyper.

## Get data from R/qtl.
data(hyper)

## Restrict to autosomes.
hyper <- subset(hyper, chr = 1:19)

## Calculate genotype probabilities.
hyper <- qb.genoprob(hyper, step=2)

## Create MCMC samples.
qbHyper <- qb.mcmc(hyper, n.thin = 40, seed = 1616)

## The next line saves qbHyper as an external binary file.
save("qbHyper", file = "qbHyper.RData")

## End(Not run)
```

qbSim *MCMC samples used in scan.pdf document*

Description

Retrieve or recreate MCMC samples used in scan.pdf document.

Usage

```
data(qbSimMain)
data(qbSimEpi)
```

Details

Both calls to data create qb objects names qbSim. See vignette scan.pdf or see scan.Rnw in doc folder of package.

See Also

qb.genoprob, qb.mcmc, qb.sim.cross

Examples

```
data(qbSimMain)
summary(qbSim)

data(qbSimEpi)
summary(qbSim)

## Not run:
## Setup for Simulated Data used in scan.pdf.
n.ind <- 100 ## number of individuals
n.mark <- 200 ## number of markers
by.mark <- 1 ## cM spacing between markers
qtl.positions <- n.mark / 2 ## position of QTL
markers <- seq(0, n.mark, by = by.mark)
names(markers) <- paste("M", markers, sep = "")
sim.map <- list(ch1 = markers)
sim.model <- matrix(c(1, qtl.positions, qtl.effect / 2), 1, 3)
colnames(sim.model) <- c("chromosome","qtl-position","effect-size")
n.iter <- 1000 ## number of iterations for MCMC
qb.random.seed <- 1626  ## random seed for MCMC

## Genetic architecture for scan simulations: 3 QTL.
qtl.positions <- rbind(qtl1 = c(chromosome = 1, locus = 5),
                       qtl2 = c(chromosome = 1, locus = 50),
                       qtl3 = c(chromosome = 2, locus = 33) )

qtl.positions
qtl.main.model <-
  rbind(qtl1.main.effect = c(qtl = 1, main.effect.size = 0),
        qtl2.main.effect = c(qtl = 2, main.effect.size = 0),
        qtl3.main.effect = c(qtl = 3, main.effect.size = 0))

qtl.main.model
qtl.epi.model <- rbind(qtl1.and.qtl3.epi.effect =
  c(qtl1 = 1, qtl2 = 3, epi.effect.size = 10))
qtl.epi.model

## SimEpi
```

```
set.seed(1234)
sim <- qb.sim.cross(len = rep(100, 2), n.mar = 10, eq.spacing = TRUE,
                    n.ind = 100, type = "bc", missing.geno = 0.03,
                    qtl.pos = qtl.positions,
                    qtl.main = qtl.main.model,
                    qtl.epis = qtl.epi.model)
sim <- qb.genoprob(sim)
qbSim <- qb.mcmc(sim, n.iter = n.iter, verbose = FALSE, n.thin = 40,
                 seed = qb.random.seed)

## The next line saves qbSim as an external binary file.
save("qbSim", file = "qbSimEpi.RData")

## SimMain
qtl.main.model[2, "main.effect.size"] = 10
set.seed(1234)
sim <- qb.sim.cross(len = rep(100, 2), n.mar = 10, eq.spacing = TRUE,
                    n.ind = 100, type = "bc", missing.geno = 0.03,
                    qtl.pos = qtl.positions,
                    qtl.main = qtl.main.model,
                    qtl.epis = NULL)

## After the data is simulated call qb.genoprob to fill in
## missing data.
sim <- qb.genoprob(sim, step = 2)

## Call qb.mcmc and then analysis code.
qbSim <- qb.mcmc(sim, n.iter = n.iter, verbose = FALSE, n.thin = 40,
                 seed = qb.random.seed)

## The next line saves qbSim as an external binary file.
save("qbSim", file = "qbSimMain.RData")

## End(Not run)
```

`step.fitqtl` *Stepwise backward elimination and anova comparison.*

Description

These functions mimic `step` and `anova` but have reduced functionality. They are not truly methods, but can help study qtl model fits.

Usage

```
step.fitqtl(cross, qtl, pheno.col = 1, arch, cutoff = 0.05,
  trace = 1, steps = 100)
## S3 method for class 'step.fitqtl'
anova(object, object2, ...)
```

Arguments

cross	Object of class cross.
qtl	Object of class qtl, as output of makeqtl.
pheno.col	Column of phenotype (numeric).
arch	Object of class qb.arch from qb.arch.
cutoff	Significance cutoff for dropping terms.
trace	If positive, information is printed during the run. Values 1, 2, 3 give gradually more detailed information.
steps	Maximum number of steps to be considered.
object	Object of class step.fitqtl from step.fitqtl.
object2	Object of class step.fitqtl from step.fitqtl.
...	Currently not used.

Details

step.fitqtl is analogous to step applied to analysis with fitqtl. anova.step.fitqtl is an S3 method for anova. anova.step.fitqtl with one argument calls summary.fitqtl; with two arguments it attempts to conduct a general F comparison of anova fits.

Value

step.fitqtl returns an object of class step.fitqtl with

fit	Object of class fitqtl.
arch	Object of class qb.arch.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

qb.arch, fitqtl, summary.fitqtl, makeqtl

Examples

```
cross <- sim.geno(cross, n.draws = 8, step = 2, error = 0.01)
qtl <- makeqtl(cross, chr = c(1,1,2,3), pos = c(15,45,12,15))
cross.step <- step.fitqtl(cross, qtl, pheno.col = 3, arch = cross.arch)
anova(cross.step)
cross.step$arch
```

`subset.qb` *Subsetting Bayesian interval mapping data*

Description

Subset Bayesian interval mapping iterations on number of QTL and/or chromosome pattern of QTL, using exact match or inclusive subsetting.

Usage

```
## S3 method for class 'qb'
subset(x, nqtl=1, pattern=NULL, exact=FALSE, chr,
  region, offset = TRUE, restrict.pair = TRUE, pheno.col, ...)
```

Arguments

`x`	object of class `qb`
`nqtl`	subset on number of QTL
`pattern`	subset on chromosome pattern of QTL
`exact`	subset on exact pattern or number of QTL if true
`chr`	subset of chromosomes to plot (numerical indices, logical or chromosome names)
`region`	list containing `chr`, and `start` and `end` positions, for regions to include
`offset`	indicates that `start` and `end` are in cM position if `TRUE`; otherwise they are in distance from first marker
`restrict.pair`	Restrict `chr` and `region` selection to linked loci all in selected subset.
`pheno.col`	Numeric or character identifier for phenotype. Useful eventually for multiple traits.
`...`	additional arguments to `subset`

Details

Subset to include only iterations with at least `nqtl` number of QTL and at least the `pattern` across chromosomes. `pattern` is a vector of chromosome identices, with repeats for multiple linked QTL on a chromosome. If `exact=FALSE`, then all iterations with at least the given `pattern` and `nqtl` are included. `nqtl` will be reset to `length(pattern)` if it is smaller than this value. Note that `pattern` should be number codes corresponding to those used in the `x` object. Further subsets to only include QTL from these iterations that are on chromosomes `chr`.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

read.cross

Examples

```
data(qbExample)

## Subset to chr 1,2, and to within 10 cM of QTL on chr 1,2.
qbSubset <- subset(qbExample, chr = c(1,2),
  region = data.frame(chr = c(1,2), start = c(35,2), end = c(55,22)))
```

---

summary.qb.scanone *Summary of qb.scanone or qb.scantwo object.*

Description

Summary of a qb.scanone or qb.scantwo object.

Usage

```
## S3 method for class 'qb.scanone'
summary(object, chr, threshold = 0,
  sort = "no", n.qtl = 0.05, ...)
## S3 method for class 'qb.scantwo'
summary(object, chr, threshold = 0,
  sort = "no", which.pos = "upper", min.iter,
  refine = FALSE, width = 10, smooth = 3, n.qtl = 0.05,
  weight = c("sqrt","count","none","atten","ratten"), ...)
## S3 method for class 'summary.qb.scantwo'
print(x, digits = 3, ...)
```

Arguments

object	A qb.scanone or qb.scantwo object.
x	An object of class qb.scantwo.
chr	Chromosomes to include in summary (must be integers for now).
threshold	Threshold(s) for inclusion in summary (see below).
sort	Sort by selected column of object ("no" indicates sort by chromosome).
which.pos	Base position estimate on this summary for maximal statistics such as LOD.
min.iter	Minimum number of iterations included at each position (default gleaned from object).
refine	Refine estimates if TRUE.
width	Window width for refinement.
smooth	Degree of nearest neighbor smoothing to determine maxima.

n.qtl	Minimum number of estimated QTL per chromosome or chromosome pair.
weight	Weights to use for nearest neighbor smoothing. sqrt is square root of count per locus. atten and ratten are attenuation based on (1-2r). Used only if smooth > 0.
digits	Significant digits to round with print.
...	Additional arguments for multiple linked loci (see details).

Details

These summary method report estimates by chromosome (or chromosome pair) at the maximum poster. Threshold can be used to condense summary to a subset of chromosomes (or chromosome pairs). Threshold is a vector with names corresponding to a subset of column names of object. Positive threshold values select chromosomes where that column average is above given value; negative threshold values select chromosomes with mean value within that value of the maximum across chromosomes. Thresholding is inclusive rather than exclusive.

It can be helpful to use summary.qb.scanone as an initial screen of chromosomes worth a further look. Since marginal summaries can include effects of multiple QTL and epistasis. Subsets based on 1-D scans can be used for 2-D subsequent screens. See qb.demo for an example.

Value

Matrix with chromosome chr, estimated position pos (or chromosome pairschr1 and chr2 and two columns for pos1 and pos2 in the case of summary.qb.scantwo) and means or modes of each column of object. Means are weighted by a smooth average of the number of MCMC sample iterations.

Author(s)

Brian S. Yandell, yandell@stat.wisc.edu

References

http://www.qtlbim.org

See Also

qb.scanone, plot.qb.scanone, qb.split.chr

Examples

```
data(qbExample)

temp <- qb.scanone(qbExample)
summary(temp, threshold = c(sum=15), sort = "sum")

temp <- qb.scantwo(qbExample)
summary(temp, threshold = c(upper=3), sort = "upper")
```

vern *Eight week vernalization data for Brassica napus*

Description

Contains genotypes and phenotypes for 8-week vernalization study used in Satagopan et al. (1996).

Usage

```
data(vern)
```

Format

See read.cross for format of vern.

Source

Thomas C. Osborn (mailto:tcosborn@facstaff.wisc.edu), Department of Agronomy, UW-Madison.

References

http://www.stat.wisc.edu/~yandell/qtl/data/osborn/Bnapus

Ferreira ME, Satagopan J, Yandell BS, Williams PH, Osborn TC (1995) Mapping loci controlling vernalization requirement and flowering time in Brassica napus. Theor Appl Genet 90: 727-732. [original source and analysis]

Kole C, Thorman CE, Karlsson BH, Palta JP, Gaffney P, Yandell BS, Osborn TC (2001) Comparative mapping of loci controlling winter survival and related traits in oilseed Brassica rapa and B. napus. Molecular Breeding 1: 329-339. [refined map and reanalysis]

JM Satagopan, BS Yandell, MA Newton and TC Osborn (1996) Markov chain Monte Carlo approach to detect polygene loci for complex traits. Genetics 144: 805-816. http://www.genetics.org/cgi/content/abstract/144/2/805 [first MCMC for experimental crosses; analysis of B. napus N2=LG9; see vern data]

See Also

read.cross, plot.qb, qb.mcmc

Examples

```
data(vern)
summary(vern)
```

Index