The qtlbim Package

December 12, 2006

Version 1.6.5

Date 12/11/2006

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 stats, qtl (>= 1.0.3), lattice, coda, tools, MASS

License GPL version 2 or newer

URL http://www.qtlbim.org

R topics documented:

qb.BayesFactor ......................................................... 2
Bnapus ............................................................... 4
qb.arch ............................................................. 5
qb.coda ............................................................. 7
qb.covar ............................................................ 8
qb.confound ......................................................... 9
qb.data ............................................................. 11
plot.qb.diag ....................................................... 12
plot.qb.epistasis .................................................. 13
fisch ............................................................... 15
step.fitql .......................................................... 16
qb.genoprob ....................................................... 17
qb.hpdone .......................................................... 18
plot.qb.loci ....................................................... 20
qb.mcmc ............................................................ 21
Bayesian model selection via Bayes factors.

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

Usage

```r
qb.BayesFactor(qbObject, items = c("nqtl","pattern","chrom","pairs"),
               cutoff.pattern = 0.2, cutoff.pairs = 1, nmax = 15)
```

# 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.
qb.BayesFactor

cutoff.pattern  Percent cutoff for pattern inclusion in model selection.
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).
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 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 \( \text{posterior} / \text{prior} \). Any pair of model terms can be compared as the ratio of their Bayes factors. The items evaluated are:

nqtl  Number of QTLs.

pattern Pattern of QTL across chromosomes. Identifiers are comma-separated chromosome numbers, with asterisk \( n^* \) for multiple QTL per chromosome.

chrom Chromosome.

pairs Epistatic pairs of chromosomes.

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

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

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 = <em>B. napus</em>` chromosome (N1, N2, etc.)
- `order = along chromosome`
- `cM d= istance from proximal end`
- `marker = marker name: E = AFLP; *ec, *tg, *wg = RFLP; *xxx = other markers from Arabidopsis: Lem, eru1, eru2, fad3, isoDia, isoldh, isoPgi, isoLap, pr2, slg6, Aca1, cor15`

Remaining columns are for individual DH lines (identifier 302-455).

Source

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

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


See Also

read.cross, plot.qb

Examples

data(Bnapus)
summary(Bnapus)

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.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 (must 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.coda

Examples

```r
## 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
```

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/qb.

Usage

```r
qb.coda( object, element, variables)
```

Arguments

- `object` 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`.

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](http://www.qtlbim.org)

See Also

`mcmc`, `qb.mcmc`, `plot.qb`
Examples

```r
## 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

```r
qb.confound(qbObject, covar = 1)
```

## S3 method for class 'qb.confound':
```r
plot(x, ylim, main, ...)
```

## S3 method for class 'qb.confound':
```r
print(x, ...)
```

## S3 method for class 'qb.confound':
```r
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. Confounding, 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.
qb.covar

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

```r
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

```r
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](http://www.qtlbim.org)

See Also

- `qb.mcmc`

Examples

```r
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**

```r
qb.data( cross, pheno.col = 1, trait = c("normal","binary","ordinal"),
        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 quantitative trait or dependent variable: "normal" or "binary" or "ordinal".
- **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.
plot.qb.diag

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 different 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)

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, ...)

plot.qb.epistasis

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

```r
  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 epistasis (`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. Parallel 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

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

Description


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)
**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**

```r
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
qb.genoprob

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)
qt1 <- makeqtl(cross, chr = c(1,1,2,3), pos = c(15,45,12,15))
cross.step <- step.fitqtl(cross, qt1, pheno.col = 3, arch = cross.arch)
anova(cross.step)
cross.step$arch

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.func = c("Haldane","Kosambi"), step = 2, tolerance = 1e-6, ...)

Arguments

cross An object of class cross. See read.cross for details.
map.func Indicates whether to use the "Haldane" or "Kosambi" map function when converting genetic distances to recombination fractions.
step Distance (in cM) between positions at which putative QTL positions and their genotypic 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.
... 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.gtlbim.org

See Also
jittermap, calc.genoprob.

Examples

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

##Highest probability density (HPD) region.

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

### Usage

```r
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, smooth = 3, ...)
```

### 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 $100\times$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, identified 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**

### Examples

```r
temp <- qb.hpdone(qbExample)
supply(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

temp <- qb.loci(qbExample)
plot(temp)
summary(temp)
temp <- qb.loci(qbExample, "all")
plot(temp)
summary(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 = 40, n.burnin = 0.01*n.iter*n.thin, algorithm = c("M-H","Gibbs"), genoupdate = TRUE, seed = 0, verbose = TRUE, ...)

Arguments

cross An object of class cross. See read.cross for details.
data the list returned by calling the function qb.data.
model the 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.
### qb.mcmc

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>algorithm</strong></td>
<td>Specifies the sampling algorithm for MCMC: Gibbs sampler (&quot;Gibbs&quot;) or Metropolis-Hastings algorithm (&quot;M-H&quot;).</td>
</tr>
<tr>
<td><strong>genoupdate</strong></td>
<td>=TRUE will update QTL genotypes and =FALSE will not do so and use the expected value of the QTL genotypes.</td>
</tr>
<tr>
<td><strong>seed</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>verbose</strong></td>
<td>=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.</td>
</tr>
</tbody>
</table>

*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.

*Value*

The qb.mcmc function returns a list of class qb, the components of which contain input parameters from the cross object, qb.data and qb.model. Since, the parameters have already been described in their respective man pages we only include the components which have been added on top of these. However, the qboject$subset component is a way to manipulate the Monte Carlo samples to make it ready for the high-end plotting routines and might not be of much importance to the average user.

- **output.dir** directory used for saving all outputs generated by qb.mcmc.
- **subset**
  - **iterdiag** is a vector of integers from 1 to n.iter.
  - **mainloci** is a vector of length equaling the sum of the number of QTLs detected in each iteration. This vector is actually a permutation vector of the mainloci.dat file stored in mydir sorted with respect to the iteration number and ties are broken with the chromosome number and the locus of the putative QTL.
  - **pairloci** is a list of
    - **order** is a vector of integers from 1 to the total number of epistatic effects in all iterations.
    - **flip** is a logical vector = TRUE if chromosome no. 1 for the paired loci is greater than chromosome no.2 or if they are the same on the same chromosome then =TRUE when position of the first is greater than the position of the second.
    - **left** is a character vector containing "chrom1" and "locus1" if there is a single TRUE in flip.
- right is a character vector containing "chrom2" and "locus2" if there is a single TRUE in flip

region a data frame storing the first and last position of the marker map for each chromosome.

Output Files

The following files are saved in output.dir:

1. ‘iterdiag.dat’
The iteration file saved in mydir has n.iter rows and 4 major columns:
   column no 1: iteration number.
   column no 2: number of putative QTLs included.
   column no 3: the overall mean.
   column no 4: the residual variance.

Depending on the type of cross, presence of covariates and epistatic effects there would be more columns in the following order: variance of additive effect, variance of dominant effect, variance of additive-additive interaction, variance of additive-dominant interaction, variance of dominant-additive interaction, variance of dominant-dominant interaction, variance of environment-additive interaction, variance of environment-dominant interaction, variance of environment effect, total genetic variance.

2. ‘covariates.dat’
The covariate file saved in mydir has 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.

3. ‘mainloci.dat’
The mainloci file saved in mydir has the N rows (N=sum of number of QTLs detected in n.iter iterations) and 4 major columns:
   column no 1: iteration number.
   column no 2: number of putative QTLs included in the model.
   column no 3: the chromosome number on which a putative QTL has been detected.
   column no 4: the QTL position indicator in the grid.

Depending on the type of cross there would be more columns in the following order: additive effect, dominant effect, variance of additive effect, and variance of dominant effect.
4. ‘pairloci.dat’
The pairloci file saved in mydir has the N rows (N=sum of number of pairs of QTLs with epistatic effect detected) and 6 major columns:

column no 1: iteration number.
column no 2: number of pairs of QTLs detected to have epistatic effect.
column no 3: the chromosome number for the first one of each pair.
column no 4: the QTL position for this one.
column no 5: the chromosome number for the second one of each pair.
column no 6: the QTL position for this one.

Depending on the type of cross there would be more columns in the following order: additive-additive interaction effect, additive-dominant interaction effect, dominant-additive interaction effect, dominant-dominant interaction effect, variance of additive-additive interaction, variance of additive-dominant interaction, variance of dominant-additive interaction, variance of dominant-dominant interaction.

5. ‘gbye.dat’
The gbye (Gene by Environment) file saved in mydir has 5 major columns:

column no 1: iteration number.
column no 2: number of GxE interactions.
column no 3: fixed covariate number.
column no 4: chromosome number of the putative QTL in the GxE interaction.
column no 5: position of the corresponding QTL.

Depending on the type of cross there would be more columns in the following order: additive effect, dominant effect, variance of additive effect, and variance of dominant effect.

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:
extample(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

```r
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**

```r
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), ...)
```

**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.
The function `qb.model` 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 different 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

```r
qbModel <- qb.model(cross, chr.nqtl = rep(3,nchr(cross)), intcov = 1, interval = rep(10,3))
```
plot.qb.multloci  *Summaries of multiple loci on a chromosome.*

**Description**

Summaries and detailed scatterplot showing all loci found in MCMC samples for a chromosome.

**Usage**

```r
qb.multloci(qbObject, chr)
## S3 method for class 'qb.multloci':
plot(x, amount = 0.75, cex, ... )
## S3 method for class 'qb.multloci':
print(x, ... )
## S3 method for class 'qb.multloci':
summary(object, ... )
```

**Arguments**

- `qbObject`: Object of class `qb`.
- `object`: Object of class `qb.multloci`.
- `x`: Object of class `qb.multloci`.
- `chr`: Identifier for one chromosome.
- `amount`: Amount to jitter points.
- `cex`: Character expansion of plot symbols.
- `...`: Parameters to methods.

**Details**

Find multiple loci in MCMC samples for chromosome `chr`. Produce scatter plot, density plots and histogram of counts with generic `plot` or show numerical `summary`. The plot provides position detail complementary to `plot.qb.scantwo`.

**Author(s)**

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

**References**

[http://www.qtlbim.org](http://www.qtlbim.org)

**See Also**

- `plot.qb`, `qb.scantwo`
Examples

```r
temp <- qb.multloci(qbExample, 1)
summary(temp)
plot(temp)
```

---

```r
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

```r
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](http://www.qtlbim.org)
See Also

plot.qb, qb.scantwo, qb.multloci

Examples

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  
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

plot(qbExample)
summary(qbExample)

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, smooth = 3, scan, ylim,
    scan.name, col, main, sub, verbose = FALSE, ...)

Arguments

x    An object of class qb.scanone.
digits Significant digits to round with print.
chr   Vector of chromosomes to plot. Must be integer.
smooth Perform smoothing if > 0 using weighted average over smooth adjacent points.
scan   The model effects to include, the default is all those included in the scanone object x.
ylim   Limits for vertical (y) axis; default uses data limits.
scan.name Name used in automatically generated main title.
plot.qb.scantwo

col
Named vector of colors for plot. Names of colors correspond to effects to be plotted. Unnamed colors will be made "black".

main
Main title for the plot

sub
Subtitle for the plot; default is color names if not too long

verbose
Give verbose feedback if TRUE.

...
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.

Value
Colors used in plots as character vector.

Author(s)
Brian S. Yandell, yandell@stat.wisc.edu

References
http://www.qtlbim.org

See Also
`qb.scanone`, `summary.qb.scanone`, `plot.scanone`

Examples

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

Description
Plots joint LOD for chromosomes on a two dimensional grid.
Usage

```r
## S3 method for class 'summary.qb.scantwo':
print(x, digits = 3, ...)
## S3 method for class 'qb.scantwo':
print(x, digits = 3, ...)
## S3 method for class 'qb.scantwo':
plot(x, chr, smooth = 3, main, offset,
     nodiag, slice = NULL, show.locus = TRUE, verbose = FALSE, ...)
```

Arguments

- `x`: An object of class `qb.scantwo`.
- `digits`: Significant digits to round with `print`.
- `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 if not `NULL` (see below).
- `show.locus`: If a `slice`, show locus estimate if `TRUE`.
- `verbose`: Give verbose feedback if `TRUE`.
- `...`: Other parameters passed to generic plot function.

Details

The offset is used only if `line(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 defaults 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 = unweighted; 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

eexample(qb.scantwo)

qb.get  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)
pull.grid(qbObject, offset, spacing, mask.region)
qb.cross(qbObject)
qb.demo()
qb.get(qbObject, element, sub)
qb.load(cross, qbObject, dir, file)
qb.save(cross, qbObject, dir, file)
qb.reorder(qbObject)

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.
qb.get

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").

sub Character string for subelement of qbObject to get.

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.

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 used to speed up examples and vignettes by saving and loading an external workspace containing a cross and a qb object. qb.save returns TRUE if the save was successful. qb.load returns TRUE if the objects exist already or they were loaded properly.

Examples

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(qbExample)
## End(Not run)
## You can call the following rather than demo() to get a tour.
qb.demo()

qb.remove

### Remove or recover qb object and associated MCMC samples

#### Description

This removes the object and the directory that contains the MCMC samples. It is not enough to remove the R object.

#### Usage

```r
qb.remove(qbObject, verbose = TRUE)
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`).
- `verbose`: Print warning 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
### qb.scanone

**Genome Scan for Main Loci Involved in Phenotypic Trait**

**Description**

This method extracts iteration diagnostics and main loci 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**

```r
qb.scanone(qbObject, epistasis = TRUE, scan, type, covar, adjust.covar, chr, sum.scan = "yes", min.iter = 1, aggregate = TRUE, half = FALSE, verbose = FALSE)
```

**Arguments**

- `qbObject`: An object of class qb.
- `epistasis`: If TRUE then information about epistasis is included.
- `scan`: Vector of diagnostics to scan (see below).
- `type`: 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.
- `verbose`: Give verbose feedback if TRUE.

**Examples**

```r
## Not run:
## 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)
```
Details

The `type` 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` are "heritability", "LPD", "LR", "deviance", "detection", "variance", "estimate", "cellmean", "count", "log10", "posterior", "logposterior" (i.e. log10(posterior)), "BF", "2logBF" (i.e. 2*log(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.

Value

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

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

Details

The scan and type are similar to those used in qb.scanone. However, here scan is a list and type is a vector, each with elements "lower" and "upper". You can either specify scan as a list, or provide 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 epistatic effects plus main effects, c("add", "dom"), plus any GxE terms.

The type defaults to c(upper = "LPD", lower = "LPD"). See qb.scanone for the range of possible types. Mostly the 2-D version of type 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 and scan.

**Author(s)**

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

**References**

http://www.qtlbim.org

**See Also**

- `plot.qb.scanone`

**Examples**

```r
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**

```r
qb.sim.cross(len = rep(100,20), n.mar = 11, eq.spacing = TRUE,  
n.ind = 400, type = c("f2","bc"), 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 )
```

---

```r
# S3 method for class 'qb.sim':
summary(object, ...)
```
Arguments

len defines the length (in cM) of each chromosome and number of chromosomes. Thus \( \text{len} = c(80,90,44) \) 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 \( \text{n мар} \) is a vector then it must have the same number of entries as there are chromosomes. For example, if \( \text{n мар} = c(10,11,9) \) 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 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, 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 coefficient for the fixed covariate and the second the true value for the standard deviation of the random covariate.
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).

Set pseudo-random number seed with `set.seed` if not NULL.

An object of class `qb.sim`, typically the `qtl` element of a `cross` object created by `qb.sim.cross`.

Not used here.

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 epistatic effects are considered. 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.

`qb.sim.cross` will return an object of class `cross`. See `read.cross` for details. In addition, a component `qtl` of class `qb.sim` is added which is a list of at most 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($qtl$herit.main)=no. of QTLs and ncol($qtl$herit.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($qtl$herit.epis)=no. of QTLs pairs interacting and ncol($qtl$herit.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($qtl$herit.gbye)=no. of GxE interactions and ncol($qtl$herit.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.

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

References

http://www.qtlbim.org

See Also

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

Examples

```r
## 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),
                 main.3 = c(3, -0.5, 0),
                 main.4 = c(4, 0.5, -0.5)),
qtl.epis = rbind(epis.1 = c(qtl.a = 4, qtl.b = 5, aa = -0.7, ad = 0, da = 0, dd = 0),
                 epis.2 = 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),
                 main.3 = c(3, -1),
                 main.4 = c(4, 0)),
qtl.epis = rbind(epis.1 = c(qtl.a = 2, qtl.b = 3, aa = -2),
                 epis.2 = 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)
```
Usage

qb.sliceone(qbObject, slice, epistasis = TRUE, scan, type, covar,
    adjust.covar, chr, sum.scan = "yes", min.iter = 1,
    aggregate = TRUE, 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).
$type$ 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.
$verbose$ 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.
qb.slicetwo

**Value**

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

```r
## 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 = "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**

```r
qb.slicetwo(qbObject, chr, pos, type = "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, ...)
```
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**: 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`.
- **...**: Extra plot options.

Author(s)

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

References

[http://www.qtlbim.org](http://www.qtlbim.org)

See Also

- `qb.scantwo`
- `qb.sliceone`

Examples

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

Description

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

Usage

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

\texttt{x} \quad \text{object of class qb}

\texttt{nqtl} \quad \text{subset on number of QTL}

\texttt{pattern} \quad \text{subset on chromosome pattern of QTL}

\texttt{exact} \quad \text{subset on exact pattern or number of QTL if true}

\texttt{chr} \quad \text{subset of chromosomes to plot (numerical indices or chromosome names)}

\texttt{region} \quad \text{list containing \texttt{chr}, \texttt{start} and \texttt{end} positions, for regions to include}

\texttt{offset} \quad \text{indicates that \texttt{start} and \texttt{end} are in cM position if TRUE; otherwise they are in distance from first marker}

\texttt{restrict.pair} \quad \text{Restrict \texttt{chr} selection to epistatic pairs with both in selected subset.}

\ldots \quad \text{additional arguments to \texttt{subset}}

Details

Subset to include only iterations with at least \texttt{nqtl} number of QTL and at least the \texttt{pattern} across chromosomes. \texttt{pattern} is a vector of chromosome identices, with repeats for multiple linked QTL on a chromosome. If \texttt{exact=FALSE}, then all iterations with at least the given \texttt{pattern} and \texttt{nqtl} are included. \texttt{nqtl} will be reset to \texttt{length(pattern)} if it is smaller than this value. Note that \texttt{pattern} should be number codes corresponding to those used in the \texttt{x} object. At present, chromosome names are not allowed. Further subsets to only include QTL from these iterations that are on chromosomes \texttt{chr}.

Author(s)

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

References

http://www.qtlbim.org

See Also

\texttt{read.cross}

Examples

## 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 of a qb.scanone object.

Usage

```r
## S3 method for class 'qb.scanone':
summary(object, chr, threshold = 0,
         sort = "no", smooth = 3, n.qtl = 0.05,
         min.iter, ...)
## 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, ...)
```

Arguments

- `object`: A `qb.scanone` object.
- `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.
- `...`: Not used.

Details

These summary method report estimates by chromosome (or chromosome pair) at the maximum posterior. 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 `demo(qb.qb.scan.tour)` for an example.
Value

Matrix with chromosome chr, estimated position pos (or chromosome pairs chr1 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 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

Examples

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

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

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", "hyperslide.Rnw", package = "qtlbim")`).

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 constructed 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", "hyperslide.Rnw", package = "qtlbim")` creates a "beamer" style PDF slide show. An alternative file `system.file("doc", "hyperpaper.Rnw", package = "qtlbim")` 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", "hyperslideextra.Rnw", package = "qtlbim")` for the slide version and `system.file("external", "hyperpaperextra.Rnw", package = "qtlbim")` for the preprint version.

Author(s)

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

References

[http://www.qtlbim.org](http://www.qtlbim.org)

See Also

Sweave

Examples

```r
## Not run:
data(hyper)
qb.sweave(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 hyperslide")
system("pdflatex hyperslide")

## Create document form, with extra section.
qb.sweave(hyper,
```
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

\texttt{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

\texttt{qb.mcmc}

Examples

\begin{verbatim}
  temp <- qb.varcomp(qbExample)
  summary(temp)
  plot(temp)
\end{verbatim}

---

\texttt{vern} \hspace{1cm} \textit{Eight week vernalization data for Brassica napus}

\textbf{Description}

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

\textbf{Usage}

\texttt{data(vern)}

\textbf{Format}

See \texttt{read.cross} for format of \texttt{vern}.

\textbf{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


See Also

read.cross, plot.qb, qb.mcmc

Examples

data(vern)
summary(vern)
Index

*Topic datagen
  qb.sim.cross, 40
*Topic datasets
  Bnapus, 3
  fisch, 14
  vern, 52
*Topic data
  qb.data, 10
*Topic hplot
  plot.qb, 30
  plot.qb.epistasis, 13
  plot.qb.scanone, 31
  plot.qb.scantwo, 32
  qb.confound, 7
  qb.covar, 9
  qb.hpdone, 17
  qb.meancomp, 24
  qb.scanone, 37
  qb.scantwo, 38
  qb.sliceone, 43
  qb.slicetwo, 45
  qb.sweave, 49
  qb.varcomp, 51
*Topic manip
  qb.coda, 6
*Topic models
  plot.qb, 30
  plot.qb.diag, 12
  plot.qb.epistasis, 13
  plot.qb.loci, 19
  plot.qb.multloci, 27
  plot.qb.pairloci, 28
  plot.qb.scanone, 31
  plot.qb.scantwo, 32
  qb.BayesFactor, 1
  qb.genoprob, 16
  qb.mcmc, 20
  qb.model, 26
  qb.scanone, 37
  qb.scantwo, 38
  qb.sliceone, 43
  step.fitqtl, 15
*Topic regression
  qb.mcmc, 20
*Topic utilities
  plot.qb, 30
  plot.qb.diag, 12
  plot.qb.epistasis, 13
  plot.qb.scanone, 31
  plot.qb.scantwo, 32
  qb.confound, 7
  qb.covar, 9
  qb.hpdone, 17
  qb.meancomp, 24
  qb.scanone, 37
  qb.scantwo, 38
  qb.sliceone, 43
  qb.slicetwo, 45
  qb.sweave, 49
  qb.varcomp, 51
  Bnapus, 3
  boxplot, 12, 13
  calc.genoprob, 17
  covar.mean (qb.get), 34
  density, 13
  fisch, 14
  fitqtl, 16
  jitter, 20, 28
  jittermap, 17
  makeqtl, 15, 16
  mcmc, 6, 7
  par, 14
  plot, 14, 20, 30
  plot.qb, 3, 4, 7, 13, 15, 28, 29, 30, 53
  plot.qb.BayesFactor (qb.BayesFactor), 1
  plot.qb.confound (qb.confound), 7
  plot.qb.covar (qb.covar), 9
  plot.qb.diag, 12
plot.qb.epistasis, 13
plot.qb.hpdone (qb.hpdone), 17
plot.qb.loci, 19
plot.qb.meancomp (qb.meancomp), 24
plot.qb.multioci, 27
plot.qb.pairloci, 28
plot.qb.scanone, 31, 38, 40, 44, 48
plot.qb.scantwo, 28, 32
plot.qb.sliceone (qb.sliceone), 43
plot.qb.slicetwo (qb.slicetwo), 45
plot.qb.varcomp (qb.varcomp), 51
plot.scanone, 32
plot.scantwo, 33, 34
print, 30
print.qb (plot.qb), 30
print.qb.arch (qb.arch), 5
print.qb.BayesFactor
qb.BayesFactor, 1
print.qb.confound (qb.confound), 7
print.qb.covar (qb.covar), 9
print.qb.diag (plot.qb.diag), 12
print.qb.epistasis
plot.qb.epistasis, 13
print.qb.hpdone (qb.hpdone), 17
print.qb.loci (plot.qb.loci), 19
print.qb.meancomp (qb.meancomp), 24
print.qb.multioci
plot.qb.multioci, 27
print.qb.pairloci
plot.qb.pairloci, 28
print.qb.scanone
plot.qb.scanone, 31
print.qb.scantwo
plot.qb.scantwo, 32
print.qb.sliceone (qb.sliceone), 43
print.qb.slicetwo (qb.slicetwo), 45
print.qb.varcomp (qb.varcomp), 51
print.summary.qb.scanone
plot.qb.scanone, 31
print.summary.qb.scantwo
plot.qb.scantwo, 32
pull.grid (qb.get), 34
qb.arch, 5, 15, 16
qb.arch.step.fitqtl (qb.arch), 5
qb.BayesFactor, 1, 12, 13, 30
qb.coda, 6, 30
qb.confound, 7
qb.covar, 9
qb.cross (qb.get), 34
qb.data, 10, 13, 21, 24, 27, 34, 36, 37, 39, 42, 44
qb.demo (qb.get), 34
qb.diag, 30
qb.diag (plot.qb.diag), 12
qb.epistasis, 30
qb.epistasis (plot.qb.epistasis), 13
qb.genoprob, 11, 16, 27, 36, 42
qb.get, 34
qb.hpdone, 5, 17, 30, 49
qb.intcov (plot.qb.epistasis), 13
qb.load (qb.get), 34
qb.loci, 30
qb.loci (plot.qb.loci), 19
qb.mcmc, 3, 7, 8, 10, 11, 15, 20, 24–27, 30,
34–36, 42, 52, 53
qb.meancomp, 24
qb.model, 10, 11, 21, 24, 26, 34, 36, 42
qb.multiploici, 29
qb.multiploici (plot.qb.multiploici), 27
qb.pairloci (plot.qb.pairloci), 28
qb.recover (qb.remove), 35
qb.remove, 35
qb.reorder (qb.get), 34
qb.save (qb.get), 34
qb.scanone, 18, 19, 32, 37, 39, 44, 45, 48, 49
qb.scantwo, 5, 28, 29, 34, 38, 44, 46, 49
qb.sim.cross, 24, 40
qb.sliceone, 5, 43, 46
qb.slicetwo, 45
qb.sweave, 6, 49
qb.varcomp, 51
read.cross, 4, 10, 14, 15, 17, 21, 26, 34,
36, 42, 47, 52, 53
round, 18
scantwo, 8, 31, 35, 38
scanone, 8, 31, 35, 38
set.seed, 41
sim.cross, 42
sim.data (qb.sim.cross), 40
step, 16
step.fitqtl, 5, 6, 15
subset, 46
subset.qb, 20, 34, 46
summary, 30
summary.fitqtl, 16
summary.qb(plot.qb), 30
summary.qb.arch(qb.arch), 5
summary.qb.BayesFactor
  (qb.BayesFactor), 1
summary.qb.confound
  (qb.confound), 7
summary.qb.covar(qb.covar), 9
summary.qb.diag(plot.qb.diag), 12
summary.qb.epistasis
  (plot.qb.epistasis), 13
summary.qb.hpdone(qb.hpdone), 17
summary.qb.loci(plot.qb.loci), 19
summary.qb.meancomp
  (qb.meancomp), 24
summary.qb.multloci
  (plot.qb.multloci), 27
summary.qb.pairloci
  (plot.qb.pairloci), 28
summary.qb.scanone, 32, 38, 44, 47
summary.qb.scantwo
  (summary.qb.scanone), 47
summary.qb.sim(qb.sim.cross), 40
summary.qb.sliceone
  (qb.sliceone), 43
summary.qb.slicetwo
  (qb.slicetwo), 45
summary.qb.varcomp(qb.varcomp), 51
Sweave, 49, 50
vern, 4, 52, 53