Package 'vqtl'

March 7, 2016

Type Package

Title Genome Scans to Accommodate and Target Genetic and Non-Genetic Effects on Trait Variance

Version 1.0

Date 2016-03-04

Author Robert Corty

Maintainer Robert Corty <rcorty@gmail.com>

Description In recognition that there are many factors (genetic loci, macro-genetic factors such as sex, and environmental factors) that influence the environmental variation, the 'vqtl' package conducts genome
scans that accommodate and target these factors. The main functions of this package, scanonevar() and
scanonevar.perm() take as input a cross object from the popular 'qtl' package.

Depends R (>= 2.14.0), dglm, evd, dplyr, qtl

Imports RColorBrewer, gtools, stringr, plyr, scales, graphics, grDevices

License GPL-3

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-07 01:01:40

R topics documented:

`get.additive.coef.from.2.genoprobs`
Compute Additive Coefficient From Two Genotype Probabilities

Description

`get.additive.coef.from.2.genoprobs` should not typically be called by a user. It is used to reliably set up model coefficients from genotype probabilities.

Usage

```
get.additive.coef.from.2.genoprobs(pa, pb)
```

Arguments

`pa`	A vector of length n giving the probability of A-type for each individual, or a matrix or data.frame of dimension n-by-2 giving probability of A-type and the probability of B-type for each individual.
`pb`	A vector of length n giving the probability of B-type for each individual, or missing if `pa` is a data.frame or matrix.

Details

none

Value

Additive coefficient vector.

Author(s)

Robert Corty <rcorty@gmail.com>

get.additive.coef.from.3.genoprobs

Compute Additive Coefficient From Three Genotype Probabilities

Description

`get.add.coef.from.3.genoprobs` should not typically be called by a user. It is used to reliably set up model coefficients from genotype probabilities.

Usage

```
get.additive.coef.from.3.genoprobs(paa, pab, pbb)
```

Arguments

`paa`	A vector of length n giving the probability of AA-type for each individual, or a matrix or data.frame of dimension n-by-3 giving probability of AA-type and the probability of AB-type and the probability of BB-type for each individual.
`pab`	A vector of length n giving the probability of AB-type for each individual, or missing if `pa` is a data.frame or matrix.
`pbb`	A vector of length n giving the probability of BB-type for each individual, or missing if `pa` is a data.frame or matrix.

Details

none

Value

Additive coefficient vector.

Author(s)

Robert Corty <rcorty@gmail.com>

get.dom.coef.from.3.genoprobs

Compute Dominance Deviation Coefficient From Three Genotype Probabilities

Description

`get.dom.coef.from.3.genoprobs` should not typically be called by a user. It is used to reliably set up model coefficients from genotype probabilities.

Usage

```
get.dom.coef.from.3.genoprobs(paa, pab, pbb)
```

Arguments

paa	A vector of length n giving the probability of AA-type for each individual, or a matrix or data.frame of dimension n-by-3 giving probability of AA-type and the probability of AB-type and the probability of BB-type for each individual.
pab	A vector of length n giving the probability of AB-type for each individual, or missing if pa is a data.frame or matrix.
pbb	A vector of length n giving the probability of BB-type for each individual, or missing if pa is a data.frame or matrix.

Details

none

Value

Dominance Deviation coefficient vector.

Author(s)

Robert Corty <rcorty@gmail.com>

---

get.genoprobs.by.marker.name

Get Genotype Probabilities From a Cross Object By Marker (or Psuedomarker) Name

---

Description

get.genoprobs.by.marker.name is an accessor function that returns the genotype probabilities of each individual in the cross at the given marker or pseudomarker.

Usage

```
get.genoprobs.by.marker.name(cross, marker.name)
```

Arguments

cross	The cross from which the genetic information will be extracted.
marker.name	The name of the marker where we want to know each individuals most likely genotype.

Details

none

Value

Probability of each genotype at the given locus for each individual.

Author(s)

Robert Corty <rcorty@gmail.com>

---

get.genotypes.by.marker.name

Get Genotypes From a Cross Object By Marker Name

Description

`get.genotypes.by.marker.name` is an accessor function that returns the most likely genotype of each individual in the cross at the given marker.

Usage

```
get.genotypes.by.marker.name(cross, marker.name, use.genoprobs = TRUE,
  as.matrix = FALSE)
```

Arguments

`cross`	The cross from which the genetic information will be extracted.
`marker.name`	The name of the marker where we want to know each individuals most likely genotype.
`use.genoprobs`	Defaults to TRUE. Should we look at the genoprobs to figure out the most likely genotype? This ensures that there will be no NA. But in some cases there may be significant uncertainty, so this may oversimplify the true situation.
`as.matrix`	Defaults to FALSE. Should the resulting genotypes be returns as a vector of numeric values (default) or a matrix?

Details

none

Value

Most likely genotype (or NA) for all individuals in the cross at the specified locus.

Author(s)

Robert Corty <rcorty@gmail.com>

`get.peaks.from.scanonevar` *Get Local Maxima from Scanonevar*

Description

`get.peaks.from.scanonevar` scans the genome for loci such that the locus to the left has a lower value and the locus to the right has a lower value. This value can be either LOD score or -log10(p-value)

Usage

```
get.peaks.from.scanonevar(x, thresh)
```

Arguments

`x`	the `scanonevar` object from which the peaks will be identified
`thresh`	Optionally, the threshold over which a value has to be to be considered a peak. For example if one locus has a LOD score of 1 and the loci to its sides have LOD score of 0.9, that's not really an interesting or "peaky" locus. Defaults to 3 if x is in LOD units and 0.05 if x is in p-values

Details

none

Value

tbl_df of identified loci

Author(s)

Robert Corty `<rcorty@gmail.com>`

`is.scanonevar` *is.scanonevar*

Description

Tests whether an object is a valid scanonevar object

Usage

```
is.scanonevar(sov)
```

Arguments

sov	the object which is tested for being a scanonevar object

Details

none

Value

Returns TRUE if 'sov' is a valid scanonevar object, with attribute 'why.not' an empty list Returns FALSE if 'sov' is not a valid scanonevar object with attribute 'why.not' a list of error messages for failed tests

Author(s)

Robert Corty <rcorty@gmail.com>

See Also

scanonevar, scanonevar.to.p.values

---

margin.plot — *Plot Phenotype of interest Averaged (Marginalized) Across Specified Markers and Phenotypes*

Description

margin.plot should be used to visually investigate the relationship between the phenotype of interest and other phenotypes. margin.plot can also be used to visualize the relationship between the phenotype of interest and genetic loci of interest, but predictive.plot is usually preferrable.

Usage

```
margin.plot(cross, focal.phenotype.name, marginal.phen.names = NULL,
  marginal.marker.names = NULL, genotype.plotting.names = c("A", "H", "B"),
  subset = 1:nind(cross), col = rep(rgb(0.5, 0.5, 0.5, 0.5), nind(cross)),
  pch = 19, xlab.override = NA, ylab.override = NA, title.override = NA,
  title.cex = 1.5, circle.alpha = 0.2)
```

Arguments

cross	The cross object to be plotted
focal.phenotype.name	the phenotype to put on the y-axis
marginal.phen.names	a list of phenotypes to average over (put on the x-axis).

`marginal.marker.names`
: a list of marker names, whose values will be averaged over (put on the x-axis).

`genotype.plotting.names`
: Labels for the genotype groups. Defaults to `c('AA', 'AB', 'BB')`.

`subset`
: the subset of individuals to use

`col`
: optionally, color of dots, as in base R graphics. Defaults to gray.

`pch`
: optionally, plotting character, as in base R graphics. Defaults to 19 (disc).

`xlab.override`
: optionally, x axis label, as in base R graphics. Defaults to the name of the marginal marker.

`ylab.override`
: optionally, y axis label, as in base R graphics. Defaults to focal phenotype name.

`title.override`
: optionally, plot title, as in base R graphics. Defaults to 'focal phenotype name by marginal phenotype name'.

`title.cex`
: optionally, character expansion for title, as in base R graphics. Defaults to 1.5.

`circle.alpha`
: optionally, alpha (transparency) of discs. Defaults to 0.2.

Details

none

Value

None. Only makes plot.

Author(s)

Robert Corty <rcorty@gmail.com>

Examples

```
set.seed(27599)
   my.cross <- sim.cross(map = sim.map(), type = 'f2')
   my.cross$pheno$phenotype <- rnorm(n = 100,
                                     mean = my.cross$geno$`1`$data[,5],
                                     sd = my.cross$geno$`2`$data[,5])
   my.cross$pheno$sex <- rbinom(n = 100, size = 1, prob = 0.5)
   my.cross$pheno$cage <- sample(x = 1:5, size = 100, replace = TRUE)

   margin.plot(cross = my.cross,
               focal.phenotype.name = 'phenotype',
               marginal.phen.name = list('sex', 'cage'),
               marginal.marker.name = list('D1M5', 'D2M5'))
```

plot.scanonevar *plot.scanonevar*

Description

plot.scanonevar implements the plot generic for objects of class 'scanonevar'. Because scanonevar objects can be viewed in terms of LODs or empirical p-values, this plotting function checks the 'units' attribute to determine which to plot.

Usage

```
## S3 method for class 'scanonevar'
plot(x, y = NULL, chrs = unique(x$chr), units.to.plot,
  col = c("black", "blue", "red", "forestgreen"), bandcol = "lightgray",
  legends = if (is.null(y)) {     c("DGLM-joint", "DGLM-mean", "DGLM-var") }
  else {     c("DGLM-joint", "DGLM-mean", "DGLM-var", "LM") },
  legend.pos = "top", legend.ncol = 2, legend.cex = 1, gap = 25,
  incl.markers = TRUE, title = attr(x, "pheno"), title.cex = 1.5,
  ylim = c(0, 1.05 * max(coords.y.locus, na.rm = TRUE)),
  show.equations = (length(chrs) != 1), alpha.side = "left",
  line.width = 1, vertical.bar = NA, ...)
```

Arguments

x	the scanonevar object to be plotted
y	Optionally, a scanone object to be plotting for comparison to the scanonevar object.
chrs	Optionally, the subset of the chromosomes to plot
units.to.plot	Optionally, whether to plot 'lods' or 'emp.ps'.
col	Optionally, a vector specifying the colors of the scan lines. Defaults to c("black", "blue", "red", "da
bandcol	Optionally, a background color for the even-index chromosomes in this scan.
legends	Optionally, the name to put for each scan in the legend. Defaults to c('mean or var', 'mean', 'var',
legend.pos	Optionally, the corner/edge where the legend should be drawn. Defaults to 'top'.
legend.ncol	Optionally, the number of columns in the legend. Defaults to 2.
legend.cex	Optionally, character expansion for the legend. Defaults to 1.
gap	Optionally, The space between chromosomes in Mb. Defaults to 25.
incl.markers	Optionally, whether to draw a rug plot along the bottom indicating where the markers are. Defaults to TRUE.
title	Optionally, title for plot. Defaults to phenotype from scanonevar object.
title.cex	Optionally, title character expansion. Defaults to 1.5
ylim	Optionally, the y limits for the plot. Defaults to c(0, 1.05 * highest.point).

show.equations	Optionally, whether to write the modeling equations under the title. Defaults to TRUE.
alpha.side	Optionally, side to print 'alpha=0.05' and 'alpha=0.01' on the thresholds. Defaults to 'left'. Other option is 'right'
line.width	Optionally, width of plotted lines. Defaults to 1.
vertical.bar	Optionally, location to plot vertical line to draw attention to one peak. Defaults to NA.
...	optional graphical parameters

Details

If such a strong signal was observed that the empirical p-value underflows R's float type, this function produces an error. The author is open to suggestions on how to deal with this situation better.

These plots look a lot better when both x (the scanone.var object) and y (optional scanone for comparison) are in units of empirical p values than when they are in LOD units.

Value

Returns nothing. Only makes the plot.

Author(s)

Robert Corty <rcorty@gmail.com>

See Also

scanonevar, scanonevar.to.p.values

Examples

```
set.seed(27599)
   my.cross <- sim.cross(map = sim.map(), type = 'f2')
   my.cross$pheno$phenotype <- rnorm(n = 100,
                                     mean = my.cross$geno$`1`$data[,5],
                                     sd = my.cross$geno$`2`$data[,5])
   my.cross$pheno$sex <- rbinom(n = 100, size = 1, prob = 0.5)
   my.cross <- calc.genoprob(my.cross)

   my.scanonevar <- scanonevar(cross = my.cross,
                           mean.formula = 'phenotype ~ sex + mean.QTL.add + mean.QTL.dom',
                                var.formula = '~sex + var.QTL.add + var.QTL.dom',
                                chrs = 1:3)

   summary(my.scanonevar)

   plot(my.scanonevar)
```

predictive.plot *Plot Predictive Interval for Categorical Genotype/Phenotype Groups*

Description

predictive.plot should be used to visually investigate loci identified with plot.scanonevar or summary.scanonevar. The user can specify the same mean and variance formulae that were used in the scan, or specify new formulae to investigate interactions.

Usage

```
predictive.plot(cross, mean.formula, var.formula, marker.name, phen.name,
  title = paste("Predictive of", response.phen, "from", phen.name, "and",
  marker.name), title.cex = 1, genotype.plotting.names = c("AA", "AB",
  "BB"), ribbon.width = 10, xlim = NA, ylim = NA)
```

Arguments

cross	The cross object to be plotted
mean.formula	The formula that describes the response, and the covariates and genetic effects that influence it. The left hand side of the ~ must be a single phenotype that is in the cross. The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: mean.QTL.add (additive effect on the mean) and mean.QTL.dom (dominance deviation from additive on the mean).
var.formula	The formula that describes the covariates and the genetic effects that influence residual (environmental) variation. There should be nothing on the left of the ~ (Inferred to be residual variation). The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: var.QTL.add (additive effect on the variance) and var.QTL.dom (dominance deviation from additive on the variance).
marker.name	The name of the marker the effects of which we want to investigate and visualize.
phen.name	The categorical phenotype the effects of which we want to investigate and visualize.
title	Optionally, title for the plot. Defaults to 'Predictive of [response phenotype] from [predictive phenotype (e.g. sex)] and [marker name]
title.cex	Optionally, character expansion for title. Defaults to 1.
genotype.plotting.names	Labels for the genotype groups. Defaults to c('AA', 'AB', 'BB').
ribbon.width	Optionally, width of ribbon connecting same-phenotype (different genotype) groups. Defaults to 10.
xlim	Optionally specify x-axis limits. Defaults to data-dependent.
ylim	Optionally specify y-axis limits. Defaults to data.dependent.

Details

none

Value

None. Only makes plot.

Author(s)

Robert Corty <rcorty@gmail.com>

Examples

```
set.seed(27599)
   my.cross <- sim.cross(map = sim.map(), type = 'f2')
   my.cross <- calc.genoprob(my.cross)
   my.cross$pheno$phenotype <- rnorm(n = 100,
                                     mean = my.cross$geno$`1`$data[,5],
                                     sd = my.cross$geno$`2`$data[,5])
   my.cross$pheno$sex <- rbinom(n = 100, size = 1, prob = 0.5)
   my.cross$pheno$cage <- sample(x = 1:5, size = 100, replace = TRUE)

   predictive.plot(cross = my.cross,
                   mean.formula = 'phenotype ~ sex + mean.QTL.add + mean.QTL.dom',
                   var.formula = '~ sex + var.QTL.add + var.QTL.dom',
                   marker.name = 'D1M5',
                   phen.name = 'sex')

   predictive.plot(cross = my.cross,
                   mean.formula = 'phenotype ~ sex + mean.QTL.add + mean.QTL.dom',
                   var.formula = '~ sex + var.QTL.add + var.QTL.dom',
                   marker.name = 'D2M5',
                   phen.name = 'sex')
```

---

scan.via.dglm *Conduct a Scanonevar Using the DGLM Function*

Description

`scan.via.dglm` should not typically be called by a user. This function is used by both `scanonevar` and `scanonevar.perm`. This function is not typically necessary for a typical user.

Usage

```
scan.via.dglm(mean.alt.formula, var.alt.formula, genoprobs, mapping.df,
  chr.by.marker, pos.by.marker, marker.names, return.effects = FALSE,
  return.effect.ses = FALSE, return.effect.ps = FALSE,
  cor.threshold = 0.8, perm = 1:nrow(genoprobs), family = "gaussian")
```

Arguments

`mean.alt.formula`	The formula for the trait mean in the alternative model. `mean.null.formula` and `test.mean.effect` are inferred from it.
`var.alt.formula`	The formula for the trait variance in the alternative model. `var.null.formula` and `test.var.effect` are inferred from it.
`genoprobs`	The probability of each genotype for each individual.
`mapping.df`	The tbl_df with the response, all covariates, and space for the focal genotype.
`chr.by.marker`	a vector of the chromosome name of each marker
`pos.by.marker`	a vector of the position of each marker
`marker.names`	a vector of the name of each marker
`return.effects`	Logical indicating whether the estimated effects should be returned.
`return.effect.ses`	Logical indicating whether the standard errors of the estimated effects should be returned.
`return.effect.ps`	Logical indicating whether the p-value of the estimated effects should be returned.
`cor.threshold`	Numeric between 0 and 1 indicating how tightly a locus must be correlated with a covariate to be skipped. e.g. if cor.threshold is 0.8 (it's default) any locus with `cor(locus, covariate) > 0.8` will be skipped.
`perm`	The permutation to apply to the genotypes. Defaults to identity permutation.
`family`	Family of distribution to be modeled. Defaults to 'gaussian'. See dglm and glm documentation for other options. Most notable other options is 'poisson'

Details

none

Value

Returns a scanonevar object.

Author(s)

Robert Corty <rcorty@gmail.com>

See Also

`scanonevar`, `scanonevar.perm`

scanonevar *Conduct a scanonevar.*

Description

scanonevar conducts a scanonevar, a genome scan that takes into account genetic and non-genetic effects on residual (environmental) variation in the trait of interest.

Usage

```
scanonevar(cross, mean.formula, var.formula, return.effects = FALSE,
  return.effect.ses = FALSE, return.effect.ps = FALSE,
  chrs = unique(names(cross$geno)), exclusion.window = 0.8,
  family = stats::gaussian)
```

Arguments

cross	The cross on which the scanonevar will be conducted.
mean.formula	The formula that describes the response, and the covariates and genetic effects that influence it. The left hand side of the ~ must be a single phenotype that is in the cross. The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: mean.QTL.add (additive effect on the mean) and mean.QTL.dom (dominance deviation from additive on the mean).
var.formula	The formula that describes the covariates and the genetic effects that influence residual (environmental) variation. There should be nothing on the left of the ~ (Inferred to be residual variation). The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: var.QTL.add (additive effect on the variance) and var.QTL.dom (dominance deviation from additive on the variance).
return.effects	Logical indicating whether the estimated effects should be returned.
return.effect.ses	Logical indicating whether the standard errors of the estimated effects should be returned.
return.effect.ps	Logical indicating whether the p-value of the estimated effects should be returned.
chrs	The subset of chromosomes to scan (defaults to all chromosomes).
exclusion.window	Numeric between 0 and 1 indicating how tightly a locus must be correlated with a covariate to be skipped. e.g. if cor.threshold is 0.8 (it's default) any locus with `cor(locus, covariate) > 0.8` will be skipped.
family	Family of distribution to be modeled. Defaults to 'gaussian'. See dglm and glm documentation for other options. Most notable other options is 'poisson'

Details

none

Value

A scanonevar object.

Author(s)

Robert Corty <rcorty@gmail.com>

Examples

```
my.cross <- sim.cross(map = sim.map(), type = 'f2')
   my.cross$pheno$sex <- rbinom(n = 100, size = 1, prob = 0.5)
   my.cross <- calc.genoprob(my.cross)

   scanonevar(cross = my.cross,
              mean.formula = 'phenotype ~ sex + mean.QTL.add + mean.QTL.dom',
              var.formula = '~sex + var.QTL.add + var.QTL.dom',
              chrs = 1:3)
```

---

scanonevar.perm *Conduct Scanonevars on Permuted Genotype Data*

Description

scanonevar.perm conducts many scanonevars on permuted data and returns the maximum observed LOD score for each chromosome type in each scan. The results should be put into convert.scanonevar.to.empiric with a scan in LODs to convert that scan to empirical p-values. It's important that all the parameters used in scanonevar.perm are the same as the parameters that were used in the scanonevar that they will be used to convert to empirical ps.

Usage

```
scanonevar.perm(cross, mean.formula, var.formula, n.perms,
  chrs = unique(names(cross$geno)))
```

Arguments

cross	The cross on which the scanonevar will be conducted.
mean.formula	The formula that describes the response, and the covariates and genetic effects that influence it. The left hand side of the ~ must be a single phenotype that is in the cross. The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: mean.QTL.add (additive effect on the mean) and mean.QTL.dom (dominance deviation from additive on the mean).

var.formula	The formula that describes the covariates and the genetic effects that influence residual (environmental) variation. There should be nothing on the left of the ~ (Inferred to be residual variation). The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: var.QTL.add (additive effect on the variance) and var.QTL.dom (dominance deviation from additive on the variance).
n.perms	the number of permutations to conduct
chrs	The subset of chromosomes to scan (defaults to all chromosomes).

Details

It is recommended to use approximately 1000 permuted scans to produce highly-replicable, publication-quality empirial p-values. For this purpose, users are recommended to dispatch this function to many computers in parallel, carefully setting the seed on each computer to insure pseudo-randomness.

Value

Returns a tbl_df of maximum LOD score observed in each genome scan for each chromosome type.

Author(s)

Robert Corty <rcorty@gmail.com>

See Also

scanonevar, scanonevar.to.p.values

Examples

```
## Not run:
   my.perms <- scanonevar.perm(cross = my.cross,
                               n.perms = 10))


## End(Not run)
```

---

scanonevar.to.p.values

Convert Scanonevar from LODs to Empirical p-values

Description

scanonevar.to.p.values takes a scanonevar with LODs as units and maxes from permutation scans, estimates an extreme value distribution for the maxes, and returns the probability of observing the LOD scores in those EVDs.

Usage

```
scanonevar.to.p.values(scanonevar, perm.scan.maxes)
```

Arguments

scanonevar	the scanonevar in LODs to be converted to emprirical p values
perm.scan.maxes	the tbl_df object returned by scanonevar.perm, the maximum LOD score observed on a per-scan, per-chromosome-type basis in permutation scans.

Details

none

Value

Returns a scanonevar object in terms of p-values, with `attr(x, 'units') = 'emp.ps'`.

Author(s)

Robert Corty <rcorty@gmail.com>

See Also

scanonevar, scanonevar.perm

Examples

```
## Not run:
   my.scanonevar <- scanonevar.perm(cross = my.cross,

   my.perms <- scanonevar.perm(cross = my.cross,
                               n.perms = 10))

   scanonevar.to.p.values(scanonevar = my.scanonevar,
                          perm.scan.maxes = my.perms)

## End(Not run)
```

---

summary.scanonevar *Summary of Peaks in Scanonevar*

Description

summary.scanonevar prints out the loci in a scanonevar object that exceed thresh. It is an S3 generic for summary(). It handles scanonevar objects in both LOD units and empirical p value units.

Usage

```
## S3 method for class 'scanonevar'
summary(object, thresh, ...)
```

Arguments

object	the scanonevar object to be summarized
thresh	the threshold over which (for LODs) or under which (for emprirical p values) a locus will be printed.
...	additional arguments controlling the summary

Details

none

Value

None. Only prints results to screen.

Author(s)

Robert Corty <rcorty@gmail.com>

Examples

```
set.seed(27599)
   my.cross <- sim.cross(map = sim.map(), type = 'f2')
   my.cross$pheno$phenotype <- rnorm(n = 100,
                                     mean = my.cross$geno$`1`$data[,5],
                                     sd = my.cross$geno$`2`$data[,5])
   my.cross$pheno$sex <- rbinom(n = 100, size = 1, prob = 0.5)
   my.cross <- calc.genoprob(my.cross)

   my.scanonevar <- scanonevar(cross = my.cross,
                           mean.formula = 'phenotype ~ sex + mean.QTL.add + mean.QTL.dom',
                                var.formula = '~sex + var.QTL.add + var.QTL.dom',
                                chrs = 1:3)

   summary(my.scanonevar)

   plot(my.scanonevar)
```

units *Get units of a scanonevar object*

Description

Utility function to get the units of a scanonevar, which is either 'LODs' (logarithm of the odds between null and alternative model) or 'emp.ps' (empirically determined p.value of that LOD)

Usage

```
units(object)
```

Arguments

object	the scanonevar object whose units are interrogated

Details

none

Value

units of scanonevar object

Author(s)

Robert Corty <rcorty@gmail.com>

validate.convert.scanonevar.to.p.values *Check the Compatibility of the Scanonevar to be Converted with the Permutations to be Used in the Conversion*

Description

validate.convert.scanonevar.to.p.values should not typically be called by a user. This function is used by scanonevar.to.emp.ps

Usage

```
validate.convert.scanonevar.to.p.values(scan, null.scan.maxes)
```

Arguments

scan	the scanonevar to be converted
null.scan.maxes	the maximum LODs observed in permutation (null) scans to be used in the conversion.

Details

none

Value

Returns TRUE if the two arguments are compatible and FALSE otherwise.

Author(s)

Robert Corty <rcorty@gmail.com>

See Also

scanonevar.to.p.values, scanonevar, scanonevar.perm

---

validate.input.scanonevar

Validate the Input before Conducting a Scanonevar

Description

validate.input.scanonevar should not typically be called by a user. It validates the input to the scanonevar function.

Usage

```
validate.input.scanonevar(cross, mean.formula, var.formula,
  chrs = names(cross$geno))
```

Arguments

cross	The cross on which scanonevar will be performed, to be validated
mean.formula	The formula that describes the response, and the covariates and genetic effects that influence it. The left hand side of the ~ must be a single phenotype that is in the cross. The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: mean.QTL.add (additive effect on the mean) and mean.QTL.dom (dominance deviation from additive on the mean).
var.formula	The formula that describes the covariates and the genetic effects that influence residual (environmental) variation. There should be nothing on the left of the ~ (Inferred to be residual variation). The right hand side must use only phenotypes thata are in the cross, markers that are in the cross, and the special terms: var.QTL.add (additive effect on the variance) and var.QTL.dom (dominance deviation from additive on the variance).
chrs	The subset of chromosomes to scan (defaults to all chromosomes).

Details

none

Value

Validated and organized inputs for scanonevar

Author(s)

Robert Corty <rcorty@gmail.com>

Index