Package 'multic'

July 2, 2014

Type Package

Title Quantitative linkage analysis tools using the variance components approach

Version 0.3.8.1

Depends R (>= 2.2.0)

Suggests kinship2 (>= 1.3)

Date 2013-01-09

Author Eric Lunde, Mariza de Andrade, Beth Atkinson, Pat Votruba

Maintainer Pat Votruba <votruba.patrick@mayo.edu>

Description Calculate the polygenic and major gene models for quantitative trait linkage analysis using variance components approach. The 0.2.2 release includes bug fixes that allow multic to run properly on 64-bit systems. The 0.3.0 release includes a fully implemented sw2mloci function. As of 0.3.8,Splus version is no longer supported.

License GPL (>= 2) | file LICENSE

OS_type unix

Repository CRAN

Date/Publication 2014-01-07 09:45:38

NeedsCompilation yes

R topics documented:

---

addGE *Assess combinations of univariate multic objects*

Description

Determine whether there is any evidence that running a multivariate multic model will significantly increase the evidence of a genetic effect.

Usage

```
addGE(multic.objs, combine=2, plotit=FALSE, ibd.dist, statistic=c("lrt",
"wald"), legend=TRUE, ylim=NULL, ...)
```

Arguments

`multic.objs`	A list of 1-trait multic objects.
`combine`	Indicate how many traits should be examined together. The program will then look at all N traits choose 'combine'.
`plotit`	Logical, default=FALSE. If TRUE, a LOD plot is generated with a separate line for each combination of traits.
`ibd.dist`	The default is to use the distances from the first multic object. This options allows the user to provide a different set of distances.
`statistic`	Character, default="lrt". This determines whether the Wald statistic (MG/SE) or the LRT is used when combining the traits.
`legend`	Logical, default=TRUE. If TRUE and if plotit=TRUE then a legend is automatically provided.
`ylim`	The extremes of the range of the y-axis to pass to the plot function.
`...`	Allows for graphical parameters to be passed to the plot function (only applicable when plotit=TRUE).

Value

A data frame is returned if the function is assigned to new object. Included are the various combinations (listed in order 1-N), the Chi-square statistic, the p-value, the distance, and the LOD score.

Side Effects

If plotit=T, a plot is generated on the current graphical device.

References

M. de Andrade, C. Olswold, J.P. Slusser, L.A.Tordsen, E.J. Atkinson, K.G. Rabe, and S.L.Slager. Identification of genes involved in alcohol consumption and cigarette smoking. BMC Genetics, 6:S112, 2005.

See Also

multic, gene.eff

Examples

```
## Not run:
add2 <- addGE(list(bmi10, dia10, sys10), combine = 2, plotit = T, ylim=c(0,8), legend=F)
add3 <- addGE(list(bmi10, dia10, sys10), combine = 3, plotit = F)
lines(add3$cM, add3$lod, col=4, lwd=2, lty=4)
key(corner=c(0,1), lines=list(lwd=2, col=1:4, lty=1:4),
    text=list(c('BMI-Dia','BMI-Sys','Dia-Sys','BMI-Dia-Sys'), col=1:4))

## End(Not run)
```

---

expand.data *Create a "bootstrapped" version of a dataset to be used in multic.*

Description

When using multic to bootstrap over families, an appropriate data set is needed. By providing a random set of famids, expand.data creates such a dataset.

Usage

```
expand.data(famids, d.frame)
```

Arguments

famids	famids is a character or integer vector that specifies the family order in a "bootstrapped" fashon. Each index of famids is the famid (family identifier) from the original dataset not the index of the family. An example famids argument would be `famids <- sample(famid, length(unique(famid)), replace = TRUE)`. IMPORTANT NOTE: This sequence of famids must be the same as that passed to expand.multic. If they are not, the dataset and the external data will not match.

d.frame	the data.frame that holds the family structure and phenotype data. This should be the dataset that was used to sample famid.

Value

a data.frame that contains the bootstrapped version of the input dataset

See Also

expand.multic

Examples

```
## Not run:
famids <- sample(famid, length(unique(famid)), replace = TRUE)
expaned.ped.phen <- expand.data(famids, ped.phen)

## End(Not run)
```

---

expand.multic *Create "bootstrapped" versions of mloci.out and share.out for multic*

Description

expand.multic is a utility function to create "bootstrap"ed versions of mloci.out and share.out

Usage

```
expand.multic(famids, mloci.out=NULL, share.out=NULL)
```

Arguments

famids	famids is a character or integer vector that specifies the family order in a "bootstrapped" fashon. Each index of famids is the famid (family identifier) from the original dataset not the index of the family. An example famids argument would be `famids <- sample(famid, length(unique(famid)), replace = TRUE)`. IMPORTANT NOTE: This sequence of famids must be the same as that passed to expand.data. If they are not, the dataset and the external data will not match.
mloci.out	a character value specifying the name of an mloci.out file. This file needs to have the famid portion (i.e., the characters before the hyphen [-]) of the unique id for each entry.
share.out	a character value specifying the name of an share.out file. This file needs to have the famid portion (i.e., the characters before the hyphen [-]) of the unique id for each entry.

Value

a list of two elements. The first is the name of the new mloci.out file. The second element is the name of the new share.out. Either element may be NULL if the respective input was NULL.

Side Effects

the output files are created in the current directory. If either of the input files (mloci.out or share.out) were gzip'ed, expand.multic will gunzip them. Currently, this is done in their own directory. However, in the future, this can be done in a temporary. Also, a directory named "loci" is temporarily craeted to hold split mloci.out file.

See Also

expand.data

Examples

```
## Not run:
famids <- sample(famid, length(unique(famid)), replace = TRUE)
new.files <- expand.multic(famids, "input/mloci.out", "input/share.out")
mult.obj <- multic( -- your formula, data, famid, etc. here --
                   mloci.out = new.files$new.mloci.out,
                   share.out = new.files$new.share.out)

## End(Not run)
```

---

multic *Create a multic object*

Description

Calculate the polygenic and major gene models for quantitative trait linkage analysis using variance components approach.

Usage

```
multic(formula, data = sys.parent(), famid, id, dadid, momid, sex,
       mloci.out = NULL, share.out = "kinship", longitudinal = FALSE,
       subset = NULL, ascertainment = NULL,
       control = multic.control(...), ...)
```

Arguments

formula	a formula object, with the traits on the left of a ~ (tilde) operator and the covariates, separated by + operators, on the right. The traits may be a single numeric vector or a matrix. Commonly, traits are aggragated together using the cbind command. See the Examples section for examples.
famid	integer, numeric, or character vector specifying each individual's family identifier. Members of the same family must have the same famid and each family must have a unique famid. Any missing data will result in an error message and the termination of multic.

id	integer, numeric, or character vector specifying each individual's identifier. Members of the same family must have a unique id within the family. id does not have to be universally unique among all individuals. Any missing data will result in an error message and the termination of multic.
dadid	integer, numeric, or character vector specifying each individual's father identifier. This father identifier must have the same famid as the individual. Any missing data will result in an error message and the termination of multic.
momid	integer, numeric, or character vector specifying each individual's mother identifier. This mother identifier must have the same famid as the individual. Any missing data will result in an error message and the termination of multic.
sex	integer, numeric, or character vector specifying each individual's sex. Acceptable forms of sex-coding are "M", "m", or 1 for male and "F", "f", or 2 for female. Any missing data will result in an error message and the termination of multic.
data	data.frame in which to interpret the variables named in formula, famid, id, dadid, momid, sex, subset, and ascertainment arguments. If data is missing, the variables in formula should be in the search path.
mloci.out	a character value specifying a path to an mloci.out (or similarly formatted) file. Specifying a non-empty mloci.out file will allow multic to calculate sporadic models using the ibd (identity by decent) information in the mloci.out file. Due to the general size of mloci.out, it is often stored in .gz format. multic will manage this for the user. Whether the user specifies an mloci.out file with a .gz suffix or not will not effect how multic operates on the file. See solar2mloci for more details.
share.out	a character value specifying a path to a share.out (or similarly formatted) file. This file contains the amount of genetic material shared between each family member pair based on family structure only. It also contains boolean values to indicate whether two familiy members have a sibling-sibling, parent-parent, or parent-offspring relationships. Due to the general size of share.out, it is often stored in .gz format. multic will manage this for the user. Whether the user specifies the file wiht a .gz suffix format or not will not effect how multic operates on the file. See phi2share for more details.
longitudinal	logical flag: if TRUE, then fomula will be interpreted as a longitudinal model. In this case, the fomula argument requires special formatting as described. The number of traits on the left side of the ~ (tilde) is the number of time-points for multic to analyze. The number of covariates on the right side of the ~ (tilde) must be a multiple of the number of traits on the left side. That multiple is the amount of covariates to analyze at each time-point. All covariates for the first time point must be specified before any of the second, all second before any third, etc. See the Examples section for examples.
subset	a logical vector specifying which subset of the rows in data to use in the fit.
ascertainment	vector specifying each individual's ascertainment (effected) status. Acceptable forms of ascertainment are T, TRUE, or 1 for a proband (effected) and F, FALSE, or 0 for a non-proband (non-effected person).
control	list of iteration and algorithmic constants. See multic.control for their names and default values. These can also be given directly as arguments to

multic itself, instead of through a multic.control object. If control is specified, the arguments specified in ... will not be used.

... further arguments passed to multic.control to alter multic's default behavior.

Details

See the technical report.

Value

an object of class "multic". See multic.object for more details.

Side Effects

Many temporary files are created during multic's execution. These files are deleted afterwards (by default). If they are not deleted (due to a crash or some other unexpected action), use the included function clean() to delete them. Also, multic copies, gunzip's, and removes the copies of share.out and mloci.out (if specified).

References

Amos, C. I. (1994). "Robust variance-components approach for assessing genetic linkage in pedigrees." American Journal of Human Genetics 54(3): 535-543.

Almasy, L. and J. Blangero (1998). "Multipoint quantitative-trait linkage analysis in general pedigrees." American Journal of Human Genetics 62(5): 1198-1211.

See Also

multic.object, multic.control, phi2share, solar2mloci, solar2multic, sw2mloci

Examples

```
## Not run:
# Call multic with a univariate formula with two covariates and no
# markers (no mloci.out argument).
fit.ibd.uni <- multic(k.trig ~ sex.x + agexam,
                      data = ped.phen.data,
                      famid, id, fa, mo, sex.x,
                      share.out = 'multicInput/share.out')

# Call multic with a bivariate formula with three covariates, no
# markers (no mloci.out argument), and calculate the family log
# likelihoods.
fit.ibd.bi <- multic(cbind(k.trig, k.chol) ~ sex.x + agexam + agexam2,
                     data = ped.phen.data,
                     famid, id, fa, mo, sex.x,
                     share.out = 'multicInput/share.out',
                     calc.fam.log.liks = TRUE)

# Call multic with a longitudinal formula with six covariates letting
# the kinship library calculate the share.out argument.
```

```
long.fit <- multic(cbind(sbpA, sbpB, sbpC) ~
                     sexA + ageA + bmiA + generA + ageAg + smkA +
                     sexB + ageB + bmiB + generB + ageBg + smkB +
                     sexC + ageC + bmiC + generC + ageCg + smkC,
                   data = long.data,
                   famid, id, dadid, momid, sex,
                   longitudinal = TRUE)

## End(Not run)
```

`multic.control` *Set control parameters for multic*

Description

Allows users to alter the default behavior of multic

Usage

```
multic.control(epsilon = 1e-5,
               max.iterations = 50,
               boundary.fix = TRUE,
               constraints = c("E", "E", "E", "E", "F", "F", "F"),
               initial.values = NULL,
               save.output.files = FALSE,
               method = c("multic", "leastsq", "maxfun", "emvc"),
               calc.fam.log.liks = FALSE,
               calc.residuals = FALSE,
               keep.input = calc.residuals)
```

Arguments

`epsilon`	a numeric value specifying the convergence threshold. When the difference of an iteration's loglikelihood and the previous iteration's loglikelihood are less than `epsilon`, the value has "converged".
`max.iterations`	an integer value specifying the maximum number of iterations `multic` will take to converge during the polygenic and sporadic model calculations.
`boundary.fix`	logical flag: if `TRUE`, then the variances generated will be fixed to `0` and no longer estimated when they become less than `0.00001` (`1e-5`).
`constraints`	a character vector of length seven (7) specifying the constraints on the random effects variance components. Each value of the vector needs to be either `"E"` - 'E'stimate the variance and covariance, `"C"` - estimate the variance and 'C'onstrain the covariance, or `"F"` - 'F'ix the variance and covariance to `0`. Each index of constraints corresponds to (in this exact order) mu, polygene, major gene, environment, sibling-sibling, parent-parent, and parent-offspring.

`initial.values`
: numeric vector: use the specified initial values instead of calculating them automatically. This vector has a very specific length and order. If n is the number of traits and m is (n + (n-1) + (n-2) + ... + 1), then the length must be n + 6 * m. So for one trait (univariate), the length must be 7, for two traits (bivariate), 20, and so on. The position of the values in the vector is important as well. The first n terms are the mu starting values. The next starting values come in chunks of m. The next m values are the polygenic starting values, followed by major gene, environmental, sibling-sibling, parent-parent, and parent-offspring starting values. The metadata\$null.initial.values contains the placement of the starting values. You can use this to verify your order is correct.

`save.output.files`
: logical flag: if `TRUE`, then the multiple temporary output files `multic` generates are not removed. This is mostly for debugging purposes and is very likely to be not useful to the user community.

`method`
: a character value specifying the method to use in fitting the model. Possible values include `"multic"` (default), `"leastsq"`, `"maxfun"`, and `"emvc"` (all case insensitive).

`calc.fam.log.liks`
: logical flag: if `TRUE`, then the family log likelihoods will be returned in the `multic` object. WARNING: This significantly increases the size of the returned `multic` object.

`calc.residuals`
: logical flag: if `TRUE`, then the residuals will be calculated and Y beta differences and V matrix data will be returned in the `multic` object. WARNING: This dramatically increases the size of the returned `multic` object.

`keep.input`
: logical flag: if `TRUE`, then the traits and covariates will be saved in the `metdata` list of the `multic` object. Since the input is needed during special residual calculations, its default value is that of `calc.residuals`.

Value

a list that is designed to be supplied as a control argument to `multic`. The values for `multic.control` can be supplied directly in a call to `multic` (via the `...` parameter). These values are then filtered through `multic.control` inside `multic`.

See Also

`multic`, `multic.object`

Examples

```
## Not run:
## The following calls to multic are equivalent
multic(formula, data, control = multic.control(calc.fam.log.liks = TRUE,
                                               calc.residuals = TRUE))
multic(formula, data, calc.fam.log.liks = TRUE, calc.residuals = TRUE)

## End(Not run)
```

multic.object *a multic object*

Description

Object of class "multic" returned from the function multic.

Arguments

fam.log.liks	the log likelihoods and lod scores for each family at each marker (including the null hypothesis). fam.log.liks is a 3-dimensional matrix. The first dimension is indexed by the family identifiers. The second dimension is indexed by the words "log.lik" and "lod.score". The third dimension is indexed by the word "null" and the names of the marker file names. To calculate the family log likelihoods, calc.fam.log.liks = TRUE must be passed to multic via the ... parameter or a multic.control object. If fam.log.liks are not calculated, then fam.log.liks is a character vector providing instructions how to calculate the values.
fixed.effects	the estimate, standard error, t value, and p value of the fixed effects for the traits and covariates for the null hypothesis and each marker. fixed.effects is a 3-dimensional matrix. The first dimension is indexed by the trait and covariate names. The second dimension is indexed by the words "Estmate", "Std.err", "t.value", and "p.value". The third dimension is indexed by the word "null" and the marker file names.
polygenic	the estimate, standard error, Wald score, Wald score P-value, heritabilty estimate, standard error of the heritabilty estimate, and heritably estimate P-value for the variance and covariance of the polygenic effect of the formula for the null hypothesis and each marker. polygenic is a 3-dimensional matrix. The first dimension is indexed by the letter "s" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", "W.p.value", "h^2", "se.h^2", and "h.p.value". The third dimension is indexed by the word "null" and the marker file names.
major.gene1	the estimate, standard error, Wald score, Wald score P-value, heritabilty estimate, standard error of the heritabilty estimate, and heritably estimate P-value for the variance and covariance of the major gene effect of formula for the null hypothesis and each marker. major.gene1 is a 3-dimensional matrix. The first dimension is indexed by the letters "mg" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", "W.p.value", "h^2", "se.h^2", and "h.p.value". The third dimension is indexed by the word "null" and the marker file names.
environmental	the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the environmental effect of formula for the null hypothesis and

each marker. environmental is a 3-dimensional matrix. The first dimension is indexed by the letter "e" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", and "W.p.value". The third dimension is indexed by the word "null" and the marker file names.

sibling.sibling

the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the sibling to sibling effect of formula for the null hypothesis and each marker. sibling.sibling is a 3-dimensional matrix. The first dimension is indexed by the letters "sib" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", and "W.p.value". The third dimension is indexed by the word "null" and the marker file names. To receive valuable data, the 5th member of constraints in the multic.control object must be set to not "F" (fixed).

parent.parent the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the parent to parent effect of formula for the null hypothesis and each marker. parent.parent is a 3-dimensional matrix. The first dimension is indexed by the letter "p" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", and "W.p.value". The third dimension is indexed by the word "null" and the marker file names. To receive valuable data, the 6th member of constraints in the multic.control object must be set to not "F" (fixed).

parent.offspring

the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the parent to offspring effect of formula for the null hypothesis and each marker. parent.offspring is a 3-dimensional matrix. The first dimension is indexed by the letter "q" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", and "W.p.value". The third dimension is indexed by the word "null" and the marker file names. To receive valuable data, the 7th member of constraints in the multic.control object must be set to not "F" (fixed).

log.liks the log likelihood, centimorgan distance, log likelihood status, and lod score and P-value for the null hypothesis and each marker. log.liks is a data.frame. The row names are "null" and the markder file names. The column names are "log.likelihood", "distance", "log.lik.status", "lod.score", and "p.value". The log likelihood status represents whether the log likelihood converged before the maximum interations allowed or not and have the values of

either "converg" or "non-converg".

var.fixed — the variance of the fixed effects of the traits and covariates for the null hypothesis and each marker. var.fixed is a 3-dimensional matrix. The first and second dimensions are indexed by the trait and covariate names. The third dimension is indexed by the word "null" and the marker file names.

var.random — the variance of the polygenic, major gene, and environmental effects for the null hypothesis and each marker. var.random is a 3-dimensional matrix. The first and second dimensions are indexed as described by the polygenic, major.gene1, and environmental components above. The third dimension is indexed by the word "null" and the marker file names.

var.sandwich — a more precise variance estimator after using a sandwich estimator approach. This is only calculated if the multic object represents a univariate model. var.sandwich is a 3-dimensional matrix. The first and second dimensions are indexed by "s1", "mg1", and "e1". The third dimension is indexed by the word "null" and the marker file names.

cors — the Pearson, Spearman, genetic, environmental, and phenotypic correlations. cors is a list made up of the components "pearson", "spearman", "genetic", "environment", and "phenotype". Both "pearson" and "spearman" are their respective correlations between the traits and covariates. They are 2-dimensional matrices indexed by the trait and covariate names. "genetic", "environment", and "phenotype" are the respective correlations between the polygenic and environmenal estimates. They are 2 dimensional matrices. The first dimension is indexed by the word "null" and the marker file names. The second dimension is indexed as described by the covariance portions of the polygenic and environmenal components above.

v.matrices — the variance-covariance matrix of the trait (y) that incorporates the polygenic, major gene, shared common environment, and error matrices. v.matrices is a 2-dimensional matrix. The first dimension is indexed by the family identifier (famid) values. The second dimension is indexed by the word "null" and the marker file names. Currently, there are no individual identifiers on each of the V matrices. If the V matrices are not calculated, then v.matrices is a character vector providing instructions how to calculate the values.

residuals — the observed values minus the fitted values of the trait (y) divided by the square root of the V matrix for each family. If the residuals are not calculated, then residuals is a character vector providing instructions how to calculate the values.

descriptives — the total individuals used, mean, standard deviation, minimum, maximum, kurtosis, and skewness for each trait and covariate.

counts — various counts of the total number of pedigrees, people, females, males, and so on. This is mostly for passing data for print and summary to display and is very likely to be not useful to the user community.

call — how multic was called. call is a call object.

R.sq — the proportion of variance due to the covariates.

metadata — a list of useful data like start.time, finish.time, call, epsilon, trait.count, iterations, null.initial.values, method, etc.

Generation

This class of objects is returned by the `multic` function to represent a fitted variance components model.

Methods

Objects of this class have methods for the functions `polygene`, `print`, `plot`, `fitted`, `residuals`, and `summary`

See Also

`multic`

---

non.user.level.objects

This function is not intended for user-level use.

Description

This function was designed and implemented for internal use. It is not expected to be used by the user and thus, no significant documentation exists.

---

phi2share

Convert a SOLAR-formatted phi2 file into a multic-formatted share.out file.

Description

`phi2share` reads in the specified files and generates a multic-formatted share.out file. share.out contains unique identifiers, expected shared genetic material, and sibling, spousal, and parent-offspring true/false values.

Usage

```
phi2share(phi2, pedigree.file, pedindex.out, pedindex.cde,
          output.directory=".")
```

Arguments

phi2	a character value to specify the location of a SOLAR-formatted phi2 (or similarly formatted) file. Due to the general size of a typical phi2 file, it is often stored in .gz format. phi2share will manage this for the user. Whether the user specifies the file with a .gz suffix or not will not effect how phi2share operates on the file.
pedigree.file	a character value to specify the location of a .ped (or similarly formatted) file. This file must have a header of famid, id, fa, mo, and sex (case insensitive). The file must also be comma separated.
pedindex.out	a character value to specify the location of a pedindex.out file. This file must be the same that was output from SOLAR. It provides a mapping between the sequential number system assigned by SOLAR and the original family and individual identifiers.
pedindex.cde	a character value to specify the location of a pedindex.cde file. This file must be the same that was output from SOLAR. This file describes how pedindex.out is organized. This is necessary to read pedindex.out correctly.
output.directory	a character value specifying which directory to place the output share.out. If output.directory (including supporting path) does not exist yet, it will be created. The default directory is the current directory.

Side Effects

phi2share creates a local copy of, gunzip's, and removes the copy of phi2. It also will overwrite share.out and share.out.gz if they exist in output.directory.

See Also

solar2mloci, solar2multic

Examples

```
## Not run:
phi2share(phi2 = "phi2",
          pedigree.file = "chrom18.ped",
          pedindex.out = "pedindex.out",
          pedindex.cde = "pedindex.cde",
          output.directory = "multicInput")
phi2share("solarOutput/phi2.gz",
          "solarOutput/chrom18.ped",
          "solarOutput/pedindex.out",
          "solarOutput/pedindex.cde")

## End(Not run)
```

plot.multic *Plot a multic object*

Description

This is a method for the function plot() for objects inheriting from class `multic`. See plot or plot.default for the general behavior of this function and for the interpretation of x.

Usage

```
## S3 method for class 'multic'
plot(x, ylim=NULL, xlim=NULL, lty=1, col=1, lwd=1, ...)
```

Arguments

`x`	a `multic` object
`xlim`	range for x-axis
`ylim`	range for y-axis
`lty`	line type. Default=1
`col`	line color. Default=1
`lwd`	line width. Default=1
`...`	additional arguments like xlab, ylab, etc.

plotFamilyLods *Plot family contribution to peak LOD*

Description

Plot the top N families that contribute to the peak LOD score as obtained from a multic object

Usage

```
plotFamilyLods(x, npeakfams=5, title=NULL, title.cex=0.75, legend=T,
               legend.loc=c("left", "middle", "right", "extra"),
                type=c("top", "total", "proportion", "all"), ...)
```

Arguments

x	a multic object
npeakfams	number of top families to plot
title	title for the plot
title.cex	character size for the title
legend	logical (default = TRUE) indicating whether a legend is displayed or not
legend.loc	character string indicating the general legend location, if legend =T. "left" indicates the top left, "right" indicates the top right, "middle" indicates the upper middle, and "extra" indicates the legend should go on a separate plot.
type	character string indicating which plot should be shown. "top" shows the top npeakfams families, "total" shows the top families plus the overall total, "proportion" shows the percentage for the top families of the total LOD score, and "all" shows all 3 plots.
...	additional parameters to alter the default behavior of the plot.

Details

Based on the peak total lod score for the multic object, determine which families contribute most to that peak. Then display their contribution across the entire area that was used to fit the multic object.

Value

the lod scores for the top families

Note

The multic object must have been fit with the option "calc.fam.log.liks=TRUE". Additionally, the function will not work on polygenic multic objects.

See Also

multic.object, multic

Examples

```
## Not run:
mult10 <- multic(sys.avg ~ sex + agexam + agexam^2, data=d10,
                 famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                 mloci='multicInput/mloci.out.gz', share='multicInput/share.out.gz',
                 calc.fam.log.liks=T)

plotFamilyLods(mult10, npeakfams=3, plot="total")

## End(Not run)
```

print.multic *Use print() on a multic object*

Description

This is a method for the function print() for objects inheriting from class `multic`. See print or print.default for the general behavior of this function and for the interpretation of x.

Usage

```
## S3 method for class 'multic'
print(x, ...)
```

Arguments

`x`	a multic object
`...`	additional parameters to alter the default behavior of print.multic. Currently ... only exists to pass 'R CMD check' tests.

solar2mloci *Convert a directory of SOLAR-formatted ibd and/or mibd files into a multic-formatted mloci.out*

Description

`solar2mloci` reads all of the ibd and mibd files in the given directory, and creates mloci.out.gz in the specified output directory.

Usage

```
solar2mloci(directory, phi2, pedindex.out, pedindex.cde,
          ibd.dist = NULL, output.directory = ".",
          delete.fixed.dir = TRUE)
```

Arguments

`directory`	charater value specifying a path to a directory of SOLAR-formatted ibd and/or mibd files. These files are often kept in .gz format. `solar2mloci` will manage this for the user.
`phi2`	charater value specifying a path to a SOLAR-formatted phi2 file. Due to the general size of a typical phi2 file, it is often stored in .gz format. `solar2mloci` will manage this for the user. Whether the user specifies the file with a .gz suffix or not will not effect how `solar2mloci` operates on the file.

`pedindex.out` character value specifying a path to a SOLAR-formatted pedindex.out file. This must be the same file that was output from SOLAR. It provides a mapping between the sequential number system assigned by SOLAR and the original family and individual identifiers.

`pedindex.cde` character value specifying a path to a SOLAR-formatted pedindex.cde file. This must be the same file that was output from SOLAR. This file describes the format of pedindex.out. This is necessary to read pedindex.out correctly.

`ibd.dist` charater value specifying a path to a SOLAR-formatted .dist file that maps the character marker names of ibd files to their corresponding numeric centimorgan values.

`output.directory` charater value specifying a path to a directory where the output file mloci.out.gz will be created. If the directory (including supporting path) does not exist yet, it will be created. Also, `solar2mloci` will overwrite mloci.out and mloci.out.gz if they exist in `output.directory`.

`delete.fixed.dir` logical flag: if `TRUE`, delete the temporary directory used to "fix" the SOLAR-formatted ibds and mibd files. This is mostly for debugging purposes and is very likely to be not useful to the user community.

Side Effects

Due to write permissions possibly not allowing the user to gunzip and create files in the specified `directory`, `solar2mloci` first copies `directory` and `phi2` to the current directory. `solar2mloci` then creates a temporary directory to hold the "fixed," intermediate files that will be deleted (by default). Also, `solar2mloci` will overwrite mloci.out and mloci.out.gz if they exist in `output.directory`.

See Also

`phi2share`, `solar2multic`

Examples

```
## Not run:
solar2mloci(directory = "ibddir", phi2 = "phi2",
          pedindex.out = "pedindex.out", pedindex.cde = "pedindex.cde",
          ibd.dist = "solar.dist", output.directory = "multicInput",
          delete.fixed.dir = FALSE)

solar2mloci("mibds/chrm10", "solarOutput/phi2.gz",
          "solarOutput/pedindex.out", "solarOutput/pedindex.cde")

## End(Not run)
```

`solar2multic` *Convert SOLAR-formatted output into multic-formatted mloci.out and share.out*

Description

`solar2multic` is a utility function to convert the ibd and mibd files (identity by descent) files created by SOLAR into the multic input file mloci.out and convert the phi2 created by SOLAR into the multic input file share.out.

Usage

```
solar2multic(phi2, pedigree.file, pedindex.out, pedindex.cde,
             ibd.directory, ibd.dist = NULL, output.directory = ".",
             delete.fixed.dir = TRUE)
```

Arguments

`phi2`	character value specifying a path to a SOLAR-formatted phi2 file. Due to the general size of a typical phi2 file, it is often stored in .gz format. `solar2multic` will manage this for the user. Whether the user specifies the file with a .gz suffix or not will not effect how `solar2multic` operates on the file.
`pedigree.file`	character value specifying a path to a SOLAR-formatted pedigree structure file (.ped). This file must have a header of `famid`, `id`, `fa`, `mo`, and `sex` (case insensitive). The file must also be comma separated.
`pedindex.out`	character value specifying a path to a SOLAR-formatted pedindex.out file. This file must be the same that was output from SOLAR. It provides a mapping between the sequential number system assigned by SOLAR and the original family and individual identifiers.
`pedindex.cde`	character value specifying a path to a SOLAR-formatted pedindex.cde file. This file must be the same that was output from SOLAR. This file describes how pedindex.out is organized. This is necessary to read pedindex.out correctly.
`ibd.directory`	character value specifying a path to a directory containing SOLAR-formatted ibd and/or mibd files.
`ibd.dist`	charater value specifying a path to a SOLAR-formatted .dist file that maps the character marker names to numeric centimorgan values.
`output.directory`	character value specifying a path to a directory that the output files (mloci.out and share.out) will be placed. If any of the specified directory path does not exist, `solar2multic` will create the necessary directories.
`delete.fixed.dir`	logical flag: if `TRUE` (default), then the temporary directory that is created to hold intermediate files is deleted.

Side Effects

Due to write permissions possibly not allowing the user to gunzip and create files in the specified `directory`, `solar2multic` first copies `directory` and `phi2` to the current directory. `solar2multic` then creates a temporary directory to hold the "fixed," intermediate files that will be deleted (by default). Also, `solar2multic` will overwrite mloci.out, mloci.out.gz, share.out and share.out.gz if they exist in `output.directory`.

See Also

`solar2mloci`, `phi2share`

Examples

```
## Not run:
solar2multic(phi2 = "phi2.gz",
             pedigree.file = "chrom18.ped",
             pedindex.out = "pedindex.out",
             pedindex.cde = "pedindex.cde",
             ibd.directory = "mibddir",
             output.directory = "multicInput",
             delete.fixed.dir = FALSE)

solar2multic("solarOutput/phi2",
             "solarOutput/chrom10.ped",
             "solarOutput/pedindex.out",
             "solarOutput/pedindex.cde",
             "mibds/chrm10"))

## End(Not run)
```

---

subsets *Choose Sets Of Size k From The n-Elements Of a Vector*

Description

Return a matrix where each row is a set of size k chosen from the n elements in vector v. Optional to allow repeated elements.

Usage

```
subsets(n, k, v=1:n, allow.repeat=F)
```

Arguments

	<!–move the above two lines to just above the first optional argument–>
	Length of element vector.
`k`	Number of elements chosen in the subsets.

v	Vector including elements from which to choose. If not specified, integers 1 to n are used.
allow.repeat	Logical, if (T)rue, include repeats of the same integer.

Details

Optional to allow repeats in the subsets. Sets all sorted lowest element to highest. This function does almost the same as combinations(), but this function uses recursion, which allows it to work very fast. In using recursion, it runs into problems in the amount of data frames needed for the recursive calls. This limit is reached with an n near 20, and worse with a larger k.

Value

Matrix with k columns and varying number of rows. Each row is the size=k subset of integers 1:n. If no repeats allowed, it will have nCk = n!/((n-k)!k!) rows. If repeats allowed, (n+k)!/((n-1)!k!) rows.

Side Effects

<!–describe any side effects if they exist–>

References

Venebles, W.N. and Ripley, B.D., "Statistics and Computing", New York: Springer-Verlag. 2000. 49+

See Also

combine

Examples

```
## Not run:
sets <- subsets(5, 3, v=6:10)
sets.w.rep_subsets(5,3,v=6:10, allow.repeat=T)
  ## sets has 10 rows and sets w/repeats has 35 rows.

## End(Not run)
```

---

summary.multic *Use print() on a multic object*

Description

Produces a summary of a fitted multic object.

Usage

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

Arguments

object	an object of class "multic", usually a result of a call to "multic".
...	further arguments passed to or from other methods. Currently ... only exists to pass 'R CMD check' tests.

Details

This is a method for the function summary() for objects inheriting from class multic. See "summary" for the general behavior of this function.

Value

a list is returned with the following components. <s-arg name="call"> as contained on object </s-arg> <s-arg name="max.lod.score"> maximum lod score from the multic object </s-arg> <s-arg name="max.lod.locus"> IBD file that corresponds to the maximum lod score </s-arg> <s-arg name="max.lod.centimorgan"> position where the maximum lod score was found </s-arg> <s-arg name="n"> default=5. Number of families to examine for the top families </s-arg> <s-arg name="top.n.families"> ids of the top n families as they contribute to the maximum lod score </s-arg> <s-arg name="centimorgan.close.to.peak"> The minimum and maximum positions (cM) that produced a lod score greater than the "maximum - 1" and are contiguous to max.lod.locus. </s-arg>

See Also

multic, multic.object

Examples

```
## Not run: summary(fit)
```

---

sw2mloci *Convert SimWalk IBD files into a single mloci.out.*

Description

sw2mloci converts all IBD files in a given directory into a sinlge mloci.out needed my multic, altering the centimorgan values if a map argument is provided.

Usage

```
sw2mloci(ibd.directory, map="", output.directory=".", famid = NULL, id = NULL,
         dadid = NULL, momid = NULL, directory = NULL)
```

Arguments

ibd.directory	a character object specifying the name of the directory that contains the SimWalk IBD files. This can be an absolute or relative path.
map	a character object specifying the name of a .map file to be used to modify the centimorgan values in the mloci.out file. This can be an absolute or relative path and does not have to be in the same directory as the parameter ibd.directory.
output.directory	a character object specifying the name of the directory to put the finished mloci.out.gz. This can be an absolute or relative path. If the directory does not exist, it will be created.
famid	Vector of family id values.
id	Vector of individual id values.
dadid	Vector of father id values.
momid	Vector of mother id values
directory	Deprecated – replaced by ibd.directory, but included for backward compatibility. If not NULL and ibd.directory is NULL, then its value is assigned to ibd.directory.

Value

a character object specifying the name of the file created. In general, this will be "mloci.out.gz".

Side Effects

If a file named "mloci.out" or "mloci.out.gz" already exist in the current directory, sw2mloci will move them to "mloci.out.before" or "mloci.out.before.gz" respectively before doing any calculations. sw2mloci also copies the IBD files and map file (if it is specified) to a temp space. This is done to bypass any write permission issues. This temp space is deleted when the function is finished. It also creates a temp space to hold the intermediate mibd files. These also will be deleted at the end of the function.

See Also

There are similar functions to deal with SOLAR mibds, see phi2share, solar2mloci, and solar2multic.

Examples

```
## Not run:
sw2mloci("../otherInput/sw18", "../otherInput/sw18/c18.map")

sw2mloci("sw18")

sw2mloci(".", "sw18/c18.map", output.directory = "multicInput")

## End(Not run)
```

tRank *Empirical Normal Quantile transformation*

Description

Transform a vector using the empirical normal quantile distribution (also called the van der Waerden normal scores).

Usage

```
tRank(x)
```

Arguments

x	a numerical vector

Value

A numerical vector that are approximately normally distributed.

References

Lehman E.L. 1975. Nonparametrics: Statistical Methods Based On Ranks. Holden-Day, Inc., San Francisco, CA (page 97)

Examples

```
## Not run:
stem(tRank(1:100))

## End(Not run)
```

Index