

Package ‘trio’

April 23, 2012

Type Package

Title Detection of disease-associated SNP interactions in case-parent trio data

Version 1.5.0

Date 2012-04-20

Author Holger Schwender, Qing Li, Ingo Ruczinski

Maintainer Holger Schwender <holger.schw@gmx.de>

Suggests survival, haplo.stats, mcbiopi, siggenes, splines

Description Testing SNPs and SNP interactions with a genotypic TDT.

This package furthermore contains functions for computing pairwise values of LD measures and for identifying LD blocks, as well as functions for setting up matched case pseudo-control genotype data for case-parent trios in order to run trio logic regression, to impute missing genotypes in trios, or to simulate case-parent trios with disease risk dependent on SNP interaction.

License LGPL-2

Repository CRAN

Date/Publication 2012-04-23 12:22:24

R topics documented:

allelicTDT	2
colGxE	3
colGxGPerms	6
colTDTmaxTest	8
colTDTsam	9
findLDblocks	12
getLD	14
getMatPseudo	15
lrControl	16
ped2geno	18

plot.getLD	19
plot.LDBlocks	20
plot.trioLR	21
poly4root	23
print.colGxE	24
print.trioFS	25
print.trioLR	26
read.pedfile	27
removeSNPs	28
scoreTDT	29
tdt	31
tdtGxG	33
trio.check	35
trio.data	38
trio.permTest	38
trio.prepare	40
trio.sim	42
trioFS	44
trioLR	46

Index	49
--------------	-----------

allelicTDT	<i>Allelic TDT</i>
------------	--------------------

Description

Performs the allelic Transmission/Disequilibrium Test for each SNP contained in a genotype matrix.

Usage

```
allelicTDT(mat.snp, size = 50, correct = FALSE)
```

```
## S3 method for class 'aTDT'
print(x, top = 5, digits = 4, ...)
```

Arguments

mat.snp	a numeric matrix in which each column represents a SNP. Each column must be a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a ped-file by, e.g., employing ped2geno .
size	the number of SNPs considered simultaneously when computing the parameter estimates.

correct	should the test statistic be continuity corrected? If FALSE, $(b-c)^2/(b+d)$ will be used as test statistic, where b and c are the off-diagonal elements of the 2x2-table summarizing the transmitted and not transmitted alleles from the heterozygous parents. If TRUE, $(b-c -1)^2/(b+d)$ will be used as test statistic.
x	an object of class aTDT, i.e. the output of allelicTDT.
digits	number of digits that should be printed.
top	number of interactions that should be printed. If top is less than or equal to zero, set to NA, or larger than the number of SNPs, then the statistics for all SNPs are printed in the order as they were in the genotype matrix used as input into colTDT. Otherwise, the top interactions with the smallest p-values are printed.
...	ignored.

Value

An object of class aTDT containing the following numeric vectors:

stat	values of the test statistic of the allelic TDT,
pval	the corresponding p-values.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Spielman, R.S., McGinnis, R.E., and Ewens, W.J. (1993). Transmission Test for Linkage Disequilibrium: The Insulin Gene Region and Insulin-Dependent Diabetes Mellitus (IDDM). *American Journal of Human Genetics*, 52, 506-516.

See Also

[colTDT](#)

colGxE

Genotypic TDT for Gene-Environment Interactions

Description

Performs a genotypic TDT for gene-environment interactions for each SNP represented by a column of a matrix in genotype format and a binary environmental factor. If alpha1 is set to a value smaller than 1, then the two-step procedure of Gauderman et al. (2010) will be used to first select all SNPs showing a p-value smaller than alpha1 in a logistic regression of the environmental factor against the sums of the codings for the parents' genotypes at the respective SNP. In the second step, the genotypic TDT is then applied to the selected SNPs.

While colGxE computes the p-values based on asymptotic ChiSquare-distributions, colGxEPerms can be used to determine permutation-based p-values. Currently, no two-step procedure is provided for colGxEPerms.

Usage

```
colGxE(mat.snp, env, model = c("additive", "dominant", "recessive"),
       alpha1 = 1, size = 50, addGandE = TRUE, whichLRT = c("both", "2df", "1df", "none"),
       add2df = TRUE, addCov = FALSE, famid=NULL)
```

```
colGxEPerms(mat.snp, env, model = c("additive", "dominant", "recessive"),
            B = 10000, size = 20, addPerms = TRUE, famid = NULL, rand = NA)
```

Arguments

mat.snp	a numeric matrix in which each column represents a SNP. Each column must be a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a ped-file by, e.g., employing ped2geno .
env	a vector of length t (see mat.snp) containing for each offspring the value of a binary environmental variable, which must take the values 0 and 1.
model	type of model that should be fitted. Abbreviations are allowed. Thus, e.g., model = "dom" will fit a dominant model, and model = "r" an recessive model.
alpha1	a numeric value between 0 and 1 (excluding 0). If alpha1 = 1, all SNPs will be tested with a genotypic TDT. Otherwise, the two-step procedure of Gauderman et al. (2010) will be used to select all SNPs showing a p-value smaller than or equal to alpha1 in a logistic regression in which the environmental factor is used as response and the sums over the codings for the genotypes of the parents are employed as predictor. The genotypic TDT will then be applied to the selected SNPs. Since a logistic regression is employed in the first step, which requires a numerical determination of the parameter estimates, the two-step procedure will not lead to a reduction in computing time, but will increase the computing time.
size	the number of SNPs considered simultaneously when computing the parameter estimates.
addGandE	should the ORs and their confidence intervals for the exposed cases be added to the output?
whichLRT	character string specifying which likelihood ratio test should be added to the output. If "2df", 2 degree of freedom likelihood ratio tests comparing the fitted models (containing one parameter for the SNP and one for the gene-environment interaction) with models containing no factor will be performed. If "1df", one degree of freedom likelihood ratio tests comparing the fitted model (containing two parameters, one for the SNP and the other for the interaction) with models only containing the respective SNP will be added to the output. If "both" (default), both tests will be performed, whereas none test will be done, if whichLRT = "none".
add2df	should the results of a 2 df Wald test for testing both the SNP and the interaction effect simultaneously be added to the model?

addCov	should the covariance between the parameter estimations for the SNP and the gene-environment interaction be added to the output? Default is addCov = FALSE, as this covariance is given by the negative variance of the parameter estimate for the SNP.
famid	a vector of the same length as env specifying the family IDs for the corresponding values of the environmental variable in env. Can be used to reorder the vector env when the order of the trios differs between env and mat.snp.
B	number of permutations.
addPerms	should the matrices containing the permuted values of the test statistics for the SNP and the gene-environment interaction be added to the output?
rand	integer for setting the random number generator into a reproducible state.

Details

A conditional logistic regression model including two parameters, one for G , and the other for GxE , is fitted, where G is specified according to model.

Value

For colGxE, an object of class colGxE consisting of the following numeric matrices with two columns (one for each parameter):

coef	the estimated parameter,
se	the estimated standard deviation of the parameter estimate,
stat	Wald statistic,
OR	the odds ratio, i.e. $\exp(\text{coef})$
,	
lowerOR	the lower bound of the 95% confidence interval for OR,
upperOR	the upper bound of the 95% confidence interval for OR,
usedTrios	the number of trios affecting the parameter estimation,
env	vector containing the values of the environmental factor,
type	model,
addGandE	the value of addGandE,
addOther	a logical vector specifying which of the likelihood ratio tests and if the 2 df Wald test was performed,

and depending on the specifications in colGxE

cov	numeric vector containing the covariances,
lrt2df	a numeric matrix with two columns, in which the first column contains the values of the 1 df likelihood ratio test statistic and the second the corresponding p-values,
wald2df	a numeric matrix with two columns, in which the first column contains the values of the 2 df Wald test statistics and the second the corresponding p-values,

`lrt1df` a numeric matrix with two columns, in which the first column contains the values of the 2 df likelihood ratio test statistic and the second the corresponding p-values.

For `colGxEPerms`,

`stat` a matrix with two columns containing the values of gTDT statistics for the main effects of the SNPs and the gene-environment interactions when considering the original, unpermuted case-pseudo-control status,

`pval` a matrix with two columns comprising the permutation-based p-values corresponding to the test statistics in `stat`,

and if `addPerms = TRUE`

`matPermG` a matrix with B columns containing the values of the gTDT statistic for the SNPs when considering the B permutations of the case-pseudo-control status,

`matPermGxE` a matrix with B columns containing the values of the gTDT statistic for the gene-environment interactions when considering the B permutations of the case-pseudo-control status.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Gauderman, W.J., Thomas, D.C., Murcray, C.E., Conti, D., Li, D., and Lewinger, J.P. (2010). Efficient Genome-Wide Association Testing of Gene-Environment Interaction in Case-Parent Trios. *American Journal of Epidemiology*, 172, 116-122.

Schwender, H., Taub, M.A., Beaty, T.H., Marazita, M.L., and Ruczinski, I. (2011). Rapid Testing of SNPs and Gene-Environment Interactions in Case-Parent Trio Data Based on Exact Analytic Parameter Estimation. *Biometrics*. DOI: 10.1111/j.1541-0420.2011.01713.x.

See Also

[colTDT](#), [ped2geno](#)

colGxGPerms

Permutation-Based gTDT for Two-Way Interactions

Description

Computes the original and permuted values of the test statistic of the gTDT test as proposed by Cordell (2002) for each interaction between the pairs of SNPs in `mat.snp`.

Usage

```
colGxGPerms(mat.snp, n.perm = 1000, genes = NULL, col.out = NULL,
  warnError = TRUE, verbose = TRUE, rand = NA)
```

Arguments

mat.snp	a numeric matrix in which each column represents a SNP. Each column must be a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a ped-file by, e.g., employing ped2geno .
n.perm	number of permutations of the response for which the permuted values of the test statistic should be computed.
genes	a character vector containing the names of the genes to which the SNPs belong. If specified, only the two-way interactions between SNPs from different genes are tested. If NULL, all two-way interactions between all possible pairs of SNPs are tested.
col.out	the output of colGxG with <code>epistatic = TRUE</code> (which is the default in colGxG). If NULL, <code>compPermTDT2way</code> computes the values of the test statistic for the original permutation of the response.
warnError	logical indicating whether the statistics for the gTDT should be returned as NA if the fitting of the conditional logistic regression model fails. This might in particular happen when the two considered SNPs are in (strong) LD.
verbose	logical indicating whether some information on what is currently computed should be printed.
rand	numeric value. If specified, the random number generator is set into a reproducible state.

Value

A list consisting of

stat	a numeric vector containing the original values of the test statistic,
permStat	a numeric matrix containing the permuted values of the test statistic,
y.perm	a matrix containing the permutations of the response.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Cordell, H. J. (2002). Epistasis: What it Means, what it Doesn't mean, and Statistical Methods to Detect it in Humans. *Human Molecular Genetics*, 11, 2463-2468.

See Also

[colGxG](#)

colTDTmaxTest	<i>Maximum Genotypic TDT</i>
---------------	------------------------------

Description

Computes the maximum over the gTDT statistics for an additive, dominant, and recessive model. colTDTmaxTest additionally computes permutation-based p-values.

Usage

```
colTDTmaxTest(geno, perm = 10000, size = 50, chunk = 10000,
              minimum = 0.001, verbose = FALSE)
colTDTmaxStat(geno, size = 50)
```

```
## S3 method for class 'maxTestTrio'
print(x, top = 5, digits = 4, ...)
## S3 method for class 'maxStatTrio'
print(x, top = 5, digits = 4, ...)
```

Arguments

geno	a numeric matrix in which each column represents a SNP. Each column must be a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a ped-file by, e.g., employing ped2geno .
perm	number of permutations of the response for which the permuted values of the test statistic should be computed.
size	number of SNPs that should be considered simultaneously when estimating the parameter.
chunk	number of permutations that should be considered simultaneously in the computation of the p-values
minimum	minimum value that a test statistic must show that for the corresponding SNP the p-value is computed.
verbose	logical indicating whether some information on what is currently computed should be printed.
x	an object of class maxTestTrio or maxTestStat, i.e. the output of colTDTmaxTest or of colTDTmaxStat.
digits	number of digits that should be printed.
top	number of interactions that should be printed. If the number of interactions is smaller than or equal to top, then the statistics for all interactions are printed in the order of their computation. Otherwise, they the top Top interactions are printed.
...	ignored.

Value

For colTDTmaxStat, an object of class maxStatTrio consisting of a vector stat containing the values of the Max statistic for the SNPs in geno, a matrix max.stat containing the values of the gTDT statistic for testing an additive, a dominant, and a recessive effect, and additional information required by colTDTmaxTest.

For colTDTmaxTest, an object of class maxTestTrio consisting of stat, max.stat, and the unadjusted p-values pval corresponding to stat.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Schwender, H., Taub, M.A., Beaty, T.H., Marazita, M.L., and Ruczinski, I. (2011). Rapid Testing of SNPs and Gene-Environment Interactions in Case-Parent Trio Data Based on Exact Analytic Parameter Estimation. *Biometrics*. DOI: 10.1111/j.1541-0420.2011.01713.x.

See Also

[tdt](#)

colTDTsam

SAM and EBAM for Trio Data

Description

Performs a Significance Analysis of Microarrays (SAM; Tusher et al., 2001) or an Empirical Bayes Analysis of Microarrays (EBAM; Efron et al., 2001), respectively, based on the genotypic transmission/disequilibrium test statistic.

Usage

```
colTDTsam(mat.snp, model = c("additive", "dominant", "recessive", "max"),
  approx = NULL, B = 1000, size = 10, chunk = 100, rand = NA)
```

```
colTDTebam(mat.snp, model = c("additive", "dominant", "recessive", "max"),
  approx = NULL, B = 1000, size = 10, chunk = 100,
  n.interval = NULL, df.ratio = 3, df.dens = 3, knots.mode = TRUE,
  type.nclass = c("wand", "FD", "scott"), fast = FALSE, rand = NA)
```

Arguments

<code>mat.snp</code>	a matrix in genotype format, i.e. a numeric matrix in which each column is a vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks of rows in <code>mat.snp</code> must consist of the genotypes of father, mother, and offspring (in this order), where the genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a data frame in ped format by, e.g., employing ped2geno .
<code>model</code>	type of genetic mode of inheritance that should be considered. Either "additive" (default), "dominant", "recessive", or "max". If <code>model = "max"</code> , the maximum over the gTDT statistics for testing an additive, dominant, and recessive model is used as gTDT statistic. Abbreviations are allowed. Thus, e.g., <code>model = "dom"</code> will fit a dominant model, and <code>model = "r"</code> an recessive model.
<code>approx</code>	logical specifying whether the null distribution should be approximated by a χ^2 -distribution with one degree of freedom. If <code>approx = FALSE</code> , the null distribution is estimated based on a permutation method. If not specified, i.e. NULL, <code>approx</code> is set to TRUE, when an additive, dominant, or recessive mode of inheritance is considered, and <code>approx = FALSE</code> , when <code>model = "max"</code> . If <code>model = "max"</code> , it is not allowed to set <code>approx = TRUE</code> .
<code>B</code>	number of permutations used in the estimation of the null distribution, and thus, the computation of the null statistics. Ignored if <code>approx = TRUE</code> .
<code>size</code>	number of SNPs considered simultaneously when computing the gTDT statistics.
<code>chunk</code>	number of permutations considered simultaneously in the permutation procedure.
<code>n.interval</code>	the number of intervals used in the logistic regression with repeated observations for estimating the ratio of the null density to the density of the observed gTDT values in an EBAM analysis (if <code>approx = FALSE</code>), or in the Poisson regression used to estimate the density of the observed gTDT values (if <code>approx = TRUE</code>). For details, see Efron et al., 2001, or Schwender and Ickstadt, 2008, respectively. If NULL, <code>n.interval</code> is determined by the maximum of 139 (see Efron et al., 2001) and the number of intervals estimated by the method specified by <code>type.nclass</code> .
<code>df.ratio</code>	integer specifying the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations for estimating the ratio of the null density to the density of the observed gTDT values in an EBAM analysis. Only used when <code>approx</code> is set to FALSE.
<code>df.dens</code>	integer specifying the degrees of freedom of the natural cubic spline used in the Poisson regression to estimate the density of the observed gTDT values in an EBAM analysis. Only used when <code>approx</code> is set to TRUE.
<code>knots.mode</code>	logical specifying whether the <code>df.dens - 1</code> knots of the natural cubic spline are centered around the mode and not the median of the density when fitting the Poisson regression model to estimate the density of the observed gTDT values in an EBAM analysis. Only used when <code>approx</code> is set to TRUE. For details on this density estimation, see denspr .

type.nclass	character string specifying the procedure used to estimate the number of intervals of the histogram used in the logistic regression with repeated observations or the Poisson regression, respectively (see <code>n.interval</code>). Can be either "wand" (default), "FD", or "scott". Ignored if <code>n.interval</code> is specified. For details, see denspr .
fast	logical specifying whether a crude estimate for the number of permuted test scores larger than the respective observed gTDT value should be used. If FALSE, the exact number of permuted test scores larger than the respective observed gTDT value is computed.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Value

The output of `colTDTsam` or `colTDTebam` is an object of class SAM or EBAM, respectively. All the features implemented in the R package `siggenes` for an SAM or EBAM analysis, respectively, can therefore be used in the SAM or EBAM analysis of case-parent trio data implemented in `colTDTsam` or `colTDTebam`, respectively. For details, see [sam](#) or [ebam](#), respectively.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

- Efron, B., Tibshirani, R., Storey, J.D., and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *Journal of the American Statistical Association*, 96, 1151-1160.
- Schwender, H. and Ickstadt, K. (2008). Empirical Bayes Analysis of Single Nucleotide Polymorphisms. *BMC Bioinformatics*, 9, 144.
- Schwender, H., Taub, M.A., Beaty, T.H., Marazita, M.L., and Ruczinski, I. (2011). Rapid Testing of SNPs and Gene-Environment Interactions in Case-Parent Trio Data Based on Exact Analytic Parameter Estimation. *Biometrics*. DOI: 10.1111/j.1541-0420.2011.01713.x.
- Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *Proceedings of the National Academy of Science of the United States of America*, 98, 5116-5121.

See Also

[colTDT](#), [colTDTmaxStat](#), [sam](#), [ebam](#), [SAM-class](#), [EBAM-class](#)

findLDblocks *Identifying LD blocks*

Description

Finds LD blocks using the procedure proposed by Gabriel et al. (2002).

Usage

```
findLDblocks(x, alpha = 0.1, ciLD = c(0.7, 0.98), cuRecomb = 0.9,
             ratio = 9, alsoOthers = FALSE, parentsOnly = FALSE, iter = 50,
             snp.in.col = TRUE)
```

```
splitBlocks(blocks)
```

Arguments

x	either the output of <code>getLD</code> or <code>getLDlarge</code> , respectively, or a numeric matrix consisting of the integers 0, 1, and 2, where these integers are assumed to be the number of minor alleles that the respective SNPs shows at the respective subject. Missing values are allowed. By default, each column of this matrix represents a SNP, and each row a subject (for details, see <code>snp.in.col</code>). The SNPs must be ordered by their position on the considered chromosome.
alpha	numeric value between 0 and 1. For each pair of SNPs, a two-sided $100 * (1 - \alpha)\%$ confidence interval of D' is computed, and used to specify pairs of SNPs that are either in strong LD, or show historical evidence of recombination (see <code>ciLD</code> and <code>cuRecomb</code>). All SNP pairs not falling into these two categories are specified as 'Others'.
ciLD	numeric vector consisting of two values between 0 and 1. If the lower bound of the confidence interval of D' for a SNP pair is larger than or equal to the first value in <code>ciLD</code> and the upper bound is larger than or equal to the second value, then this pair of SNP is considered to be in strong LD.
cuRecomb	numeric value between 0 and 1. If the upper bound of the confidence interval of D' for a SNP pair is smaller than <code>cuRecomb</code> , then this pair of SNP is considered to show evidence of recombination.
ratio	numeric value larger than 1. If in a block of SNPs, the ratio of the number of SNP pairs being in strong LD to the number of SNPs showing evidence of recombination is larger than or equal to <code>ratio</code> , then this block will be identified as an LD-block. (Note that Gabriel et al. (2002) use <code>ratio = 19</code> instead of <code>ratio = 9</code> .) Overlapping blocks are avoided by employing the approach described in Wall and Pritchard (2003).
alsoOthers	logical value. Following the description of Wall and Pritchard (2003) the endmarkers of a LD block must be in strong LD. By default (i.e. if <code>alsoOthers = FALSE</code>), this condition is used. If <code>alsoOthers = TRUE</code> , the endmarkers can also be categorized as 'Others'.

parentsOnly	logical indicating whether only the genotypes of the parents, i.e. rows 1, 2, 4, 5, ... of x , should be used in the computation of the LD measures when x is in genotype format and contains case-parent trio data (see ped2geno and read.pedfile). If FALSE (default), all rows are used in the determination of the pairwise LD measure. Ignored if x is the output of <code>getLD</code> or <code>getLDlarge</code> .
iter	integer specifying the number of iterations used in the computation of D (for details, see getLD). Ignored if x is the output of <code>getLD</code> .
snp.in.col	logical specifying whether each column of x represents a SNP (and each row a subject). If FALSE, each row represents a SNP (and each column a subject). Ignored if x is the output of <code>getLD</code> or <code>getLDlarge</code> .
blocks	output of <code>findLDblocks</code> . See Details.

Details

The LD-blocks are estimated using the method of Gabriel et al. (2002) as described in Wall and Pritchard (2003), where we use the approximate variance estimates of D' proposed by Zabaleta et al. (1997).

Since in [trio.prepare](#) the LD blocks are restricted to a maximum of 7 SNPs, `splitBlocks` can be used to split LD blocks composed of more than 7 SNPs into smaller blocks, if the output of `findLDblocks` should be used in [trio.prepare](#) to prepare a matrix for a [trioLR](#) or [trioFS](#) analysis.

Value

An object of class `LDblocks` consisting of

<code>ld</code>	the output of <code>getLD</code> ,
<code>blocks</code>	a vector specifying which SNP belongs to which LD-block,
<code>vec.blocks</code>	a list in which each entry contains the names of the SNPs belonging to a specific LD-block,
<code>param</code>	a list of the input parameters.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

- Gabriel, S.B. et al. (2002). The Structure of Haplotype Blocks in the Human Genome. *Science*, 296, 2225-2229.
- Wall, J.D. and Pritchard J.K. (2003). Assessing the Performance of the Haplotype Block Model of Linkage Disequilibrium. *American Journal of Human Genetics*, 73, 502-515.
- Zapata, C., Alvarez, G., and Carollo, C. (1997). Approximate Variance of the Standardized Measure of Gametic Disequilibrium D' . *American Journal of Human Genetics*, 61, 771-774.

See Also

[plot.LDblocks](#), [getLD](#)

getLD

*Computation of LD Measures***Description**

While getLD computes the value of D' and r^2 for each pair of SNPs in a matrix, getLDlarge determines D' and r^2 between each SNP and a user-specified number of SNPs closest to the SNP on the corresponding chromosome. Thus, getLDlarge can be applied to much more SNPs than getLD.

Usage

```
getLD(x, which = c("both", "rSquare", "Dprime"), parentsOnly = FALSE,
      iter = 50, snp.in.col = TRUE, asMatrix = FALSE, addVarN = FALSE)
```

```
getLDlarge(x, neighbors=25, which=c("both", "rSquare", "Dprime"),
           parentsOnly=FALSE, iter=50, snp.in.col=TRUE, addVarN=FALSE)
```

Arguments

x	a numeric matrix consisting of 0, 1, and 2, where it is assumed that the values represent the numbers of minor alleles that the SNPs show. Missing values are allowed. By default, each column represents a SNP and each row a subject. This can be changed by setting snp.in.col = FALSE. It is assumed that the SNPs are ordered by their position on the considered chromosome.
neighbors	positive integer specifying the number of neighbors of a SNP (in both directions) on a chromosome for which D' or r^2 should be computed. Thus, for each SNP (except for the SNPs in the first and last neighbors columns of x), $2 * neighbors$ r^2 or D' values are computed.
which	which LD measures should be computed? Either "rSquare", or "Dprime", or the values of "both" measures are computed. The latter is the default.
parentsOnly	logical indicating whether only the genotypes of the parents, i.e. rows 1, 2, 4, 5, ... of x, should be used in the computation of the LD measures when x is in genotype format and contains case-parent trio data (see ped2geno and read.pedfile). If FALSE (default), all rows are used in the determination of the pairwise LD measure.
iter	integer specifying how many iterations are used in the procedure of Hill (1974) which is used to estimate D' .
snp.in.col	logical indicating whether each column of x represents a SNP (and each row a subject). If FALSE, each row represents a SNP (and each column a subject).
asMatrix	logical indicating whether the LD values are returned as a $m \times m$ matrix, where m is the number of SNPs. If FALSE, the LD values are returned as a vector of length $m * (m - 1) / 2$.

addVarN logical indicating whether for each pair of SNPs the number of non-missing values and the variance estimates of D' proposed by Zabaleta et al. (1997) should be added to the output. The variance estimates are required for the identification of LD-blocks with [findLDblocks](#).

Value

An object of class `getLD` or `getLDlarge` consisting (depending of the specification of which) the D' (`Dprime`) or r^2 (`rSquare`) values for each SNP pair, and (depending of the specification of `addVarN`) the variance estimates for D' (`varDprime`) and the numbers of non-missing values (`n`). Furthermore, the names of the SNPs (`rn`) will be added (in `getLD`, if `asMatrix = FALSE`).

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Hill, W.O. (1974). Estimation of Linkage Disequilibrium in Randomly Mating Populations. *Heredity*, 33, 229-239.

Zapata, C., Alvarez, G., and Carollo, C. (1997). Approximate Variance of the Standardized Measure of Gametic Disequilibrium D' . *American Journal of Human Genetics*, 61, 771-774.

See Also

[plot.getLD](#), [findLDblocks](#)

getMatPseudo

Generates Case-Pseudo-Control Matrix

Description

Generates a matrix containing the genotypes of the cases and the corresponding three pseudo-controls (i.e. the genotypes of the children and the respective corresponding three genotypes not transmitted from the parents).

Usage

```
getMatPseudo(mat.snp)
```

Arguments

`mat.snp` a numeric matrix in which each column represents a SNP. Each column must be a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a ped-file by, e.g., employing [ped2geno](#).

Value

A matrix with $4 * t$ rows, in which each block of four consecutive rows consists of the genotypes of the SNPs in `mat.snp` for the case and the three matched pseudo-controls corresponding to the respective block in `mat.snp`.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

See Also

[colTDT](#), [colTDT2way](#), [colGxE](#)

 lrControl

Control Parameters for Trio Logic Regression

Description

Specifies the control parameters for the search algorithms (i.e. either simulated annealing or MCMC) and the logic tree considered when fitting a trio logic regression model.

Usage

```
lrControl(start = 0, end = 0, iter = 0, earlyout = 0, update = 0,
          treesize = 8,opers = 1, minmass = 0, nburn = 1000, hyperpars = 0,
          output = 4)
```

Arguments

<code>start</code>	a numeric value specifying the upper temperature (on log10 scale) used as start temperature in simulated annealing. Must be larger than <code>end</code> . If both <code>start = 0</code> and <code>end = 0</code> , these temperatures will be chosen automatically (which is not the optimal way to specify these parameters).
<code>end</code>	a numeric value specifying the lowest temperature (on log10 scale) used in simulated annealing. Must be smaller than <code>start</code> . If both <code>start = 0</code> and <code>end = 0</code> , these temperatures will be chosen automatically (which is not the optimal way to specify these parameters).
<code>iter</code>	the number of iterations used in the (stochastic) search for the best trio logic regression model, i.e. either in simulated annealing (if the argument <code>search</code> in <code>trioLR</code> or <code>trioFS</code> is set to "sa") or in MCMC (if <code>search = "mcmc"</code>). If <code>iter = 0</code> , <code>iter</code> will be chosen automatically (similar to <code>start</code> and <code>end</code>) when simulated annealing is used, and will be set to <code>iter = 50000</code> when MCMC is employed.

earlyout	a non-negative integer providing an option to end the search before all <code>iter</code> iterations in simulated annealing are considered. If during five consecutive blocks of <code>earlyout</code> iterations, 10 or fewer moves proposed in simulated annealing are accepted in each of the blocks, then the search will terminate. Can help to stop the search earlier, when there is no progress in the search anymore. By default, all <code>iter</code> iterations are considered.
update	the number of iterations in simulated annealing or MCMC after which statistics for the current trio logic regression model are displayed. This argument allows to evaluate the progress in the search for the best trio logic regression model. By default, no updates are shown.
treesize	a positive integer specifying the maximum number of leaves allowed in the logic tree of a trio logic regression model.
opers	either 1, 2, or 3 specifying if both the AND and the OR operator (<code>opers = 1</code>), or only the AND operator (<code>opers = 2</code>), or only the OR operator (<code>opers = 3</code>) is considered when building the logic tree.
minmass	a non-negative integer specifying the number of cases and pseudo-controls for which the logic expression (i.e. the logic tree) needs to be 1 or for which the logic expression needs to be 0 to be considered as a logic tree in the trio logic regression model. By default, <code>minmass</code> is either set to 20% of the trios or to 15, whatever is less.
nburn	number of initial iterations in MCMC considered as burn-in MC trio logic regression, and therefore, ignored when computing the summaries.
hyperpars	a numeric value specifying the hyperparameter for the prior on the model size when performing a MC trio logic regression. More exactly, <code>hyperpars</code> is assumed to be $\log(P(\text{size} = k)/P(\text{size} = k + 1))$, where P is the prior on the model size.
output	a value specifying which statistics are returned in an MCMC trio logic regression analysis. If <code>output > 0</code> , then all fitted models are saved in a text file called "trioIrlisting.tmp" in the current working directory. By setting <code>output < 0</code> , this can be avoided. If <code>abs(output) > 1</code> , bivariate statistics are gathered. If <code>abs(output) > 2</code> , trivariate statistics are gathered. Otherwise, only univariate statistics are determined.

Details

More details on the different control parameters and their specification can be found on the help pages of the functions `logreg.anneal.control`, `logreg.tree.control`, and `logreg.mc.control` for the different types of control parameters available in the R package `LogicReg` for a standard logic regressions.

Value

A list containing all required control parameters.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

ped2geno

*Transformation of Ped-File***Description**

Transforms a ped-file into a genotype file as required by, e.g., the functions for computing the genotypic TDT.

Usage

```
ped2geno(ped, snpnames = NULL, coded = c("12", "AB", "ATCG", "1234"),
         naVal = 0, cols4ID = FALSE)
```

Arguments

ped	a data frame in ped format, i.e. the first six columns must contain information on the families as typically presented in ped files, where the column names of these six columns must be "famid", "pid", "fatid", "motid", "sex", "affected". The last two of these six columns are ignored. The IDs of individuals in the second column must be unique (not only within the family, but among all individuals). The columns following the six columns are assumed to contain the alleles of the SNPs, where the alleles are coded using the letters/numbers in coded, and missing values are coded by naVal. Thus, the seventh and the eighth column contain the two alleles for the first SNP, the ninth and tenth the two alleles for the second SNP, and so on. Contrary to the names of the first six columns, the names of the columns representing the SNPs are ignored, and SNP names can be specified using snpnames.
snpnames	a character vector containing the names of the SNPs. If not specified, generic names are assigned (i.e. SNP1, SNP2, ...). Ignored if ped just contains one SNPs.
coded	the coding used for the alleles of the SNPs. coded = "12", e.g., means that one of the alleles is coded by 1, and the other by 0. coded = "ATCG" means that the alleles are coded by the actual base.
naVal	the value used for specifying missing values.
cols4ID	logical indicating whether columns should be added to output matrix containing the family ID and the individual ID. If FALSE, the individual IDs are used as the row names of the output matrix.

Value

A vector (if ped consists of alleles for one SNP) or matrix (otherwise) containing one column for each SNP representing the genotypes of the respective SNP, where the genotypes are coded by 0, 1, 2 (i.e. the number of minor alleles), and missing values are represented by NA. The vector or matrix contains $3 * t$ values for each SNP genotyped at the t trios, where each block of 3 values is composed of the genotypes of the father, the mother, and the offspring (in this order) of a specific trio. If data for a family with more than one children are available, each of the children is treated as a separate trio.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

See Also

[tdt](#), [tdt2way](#), [trio.check](#)

 plot.getLD

Plotting a getLD or getLDlarge Object

Description

Plots either the pairwise r^2 or D' values computed by either `getLD` or `getLDlarge`. Can also be used to plot the categorizations used in the procedure of Gabriel et al. (2002).

Usage

```
## S3 method for class 'getLD'
plot(x, y = "rSquare", start = 1, end = NA, squared = TRUE,
     col = NULL, xlab = "", ylab = "", cexAxis = 0.8, alpha = 0.1,
     ciLD = c(0.7, 0.98), cuRecomb = 0.9, ...)
```

```
## S3 method for class 'getLDlarge'
plot(x, y = "rSquare", start = NA, end = NA, squared = TRUE,
     col = NULL, xlab = "", ylab = "", cexAxis = 0.8, alpha = 0.1,
     ciLD = c(0.7,0.98), cuRecomb = 0.9, ...)
```

Arguments

<code>x</code>	the output of <code>getLD</code> or <code>getLDlarge</code> .
<code>y</code>	either "rSquare" (default), "Dprime", or "gabriel" specifying the LD values that should be plotted.
<code>start</code>	integer or character string specifying the index or the name of the first SNP, respectively, that should be plotted, where the index corresponds to the column (or row if <code>snp.in.col = FALSE</code>) in the matrix used as input in <code>getLD</code> or <code>getLDlarge</code> .
<code>end</code>	integer or character string specifying the index or the name of the last SNP, respectively, that should be plotted.
<code>squared</code>	should the r^2 values be plotted? If <code>FALSE</code> , the r values are plotted. Only considered if <code>y = "rSquare"</code> .
<code>col</code>	a vector specifying the colors used in plotting of the LD values. If <code>y = "rSquare"</code> or <code>y = "Dprime"</code> , different levels of gray will be used by default (the darker, the higher is the LD value). If <code>y = "gabriel"</code> , strong LD is by default marked by blue fields, evidence of recombination by white color, and others by yellow.
<code>xlab</code>	character string naming the label of the x-axis.

ylab	character string naming the label of the y-axis.
cexAxis	a numeric value specifying the relative size of the SNP names displayed at the axes of the plot.
alpha	numeric value between 0 and 1. Only considered if y = "gabriel". For each pair of SNPs, a two-sided $100 * (1 - \alpha)\%$ confidence interval of D' is computed, and used to specify pairs of SNPs that are either in strong LD, or show historical evidence of recombination (see ciLD and cuRecomb). All SNP pairs not falling into these two categories are specified as 'Others'.
ciLD	numeric vector consisting of two values between 0 and 1. Only considered if y = "gabriel". If the lower bound of the confidence interval of D' for a SNP pair is larger than or equal to the first value in ciLD and the upper bound is larger than or equal to the second value, then this pair of SNP is considered to be in strong LD.
cuRecomb	numeric value between 0 and 1. Only considered if y = "gabriel". If the upper bound of the confidence interval of D' for a SNP pair is smaller than cuRecomb, then this pair of SNP is considered to show evidence of recombination.
...	further arguments of image

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Gabriel, S.B. et al. (2002). The Structure of Haplotype Blocks in the Human Genome. *Science*, 296, 2225-2229.

See Also

[getLD](#), [plot.LDblocks](#)

plot.LDblocks

Plotting a LDblock Object

Description

Plots either the pairwise D' values or the pairwise LD categorization used in the procedure of Gabriel et al. (2002). Additionally, the LD blocks are marked in this plot.

Usage

```
## S3 method for class 'LDblocks'
plot(x, y = "gabriel", col = NULL, start = 1, end = NA, xlab = "",
     ylab = "", cexAxis = 0.8, block.col = 2, block.lwd = 3, ...)
```

Arguments

x	the output of findLDblocks.
y	either "Dprime" or "gabriel" (default) specifying the LD values that should be plotted.
col	a vector specifying the colors used in plotting of the LD values. If y = "Dprime", different levels of gray will be used by default (the darker, the higher is the LD value). If y = "gabriel", strong LD is by default marked by blue fields, evidence of recombination by white color, and others by yellow.
start	integer or character string specifying the index or name of the first SNP, respectively, that should be plotted, where the index corresponds to the column (or row if snp.in.col = FALSE) of the matrix used as input in getLD or findLDblocks.
end	integer or character string specifying the index or name of the last SNP, respectively, that should be plotted.
xlab	character string naming the label of the x-axis.
ylab	character string naming the label of the y-axis.
cexAxis	a numeric value specifying the relative size of the SNP names displayed at the axes of the plot.
block.col	the color of the lines used to show the borders of the LD blocks.
block.lwd	numeric value specifying the size of the lines used to show the borders of the LD blocks
...	further arguments of image.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Gabriel, S.B. et al.~(2002). The Structure of Haplotype Blocks in the Human Genome. *Science*, 296, 2225-2229.

See Also

[findLDblocks](#), [plot.getLD](#)

plot.trioLR

Plotting for trioLR Objects

Description

Plots the logic trees or information on the visited models generated in a the trio logic regression analysis with trioLR.

Usage

```
## S3 method for class 'trioLR'
plot(x, whichTree = NA, freqType = 1, useNames = FALSE,
      addStats = TRUE, digits = 3, main = NULL, cexOper=1.5,
      cexLeaf=1.5, sizeLeaf=7, cexPar=1.3, ...)
```

Arguments

x	an object of class <code>trioLR</code> , i.e. the output of <code>trioLR</code> .
whichTree	positive integer specifying the model for which the logic tree should be plotted when several trio logic regression models with different maximum numbers of leaves have been fitted. Ignored if just one model has been fitted using simulated annealing or MCMC has been employed to perform a Trio Logic Regression.
freqType	positive integer between 1 and 3 specifying which statistics from the MC Trio Logic Regression analysis should be plotted. If <code>freqType = 1</code> , then for each variable, the percentage of models visited (after the burn-in) in the MCMC chain that contain this variable will be plotted. If <code>freqType = 2</code> , then for each pair of variables, this percentage will be shown. If <code>freqType = 3</code> , then for each pair of variables, the observed-to-expected ratio for being jointly in the models will be plotted. Ignored if simulated annealing or the greedy algorithm was used in the application of <code>trioLR</code> .
useNames	should the names of the variables be used in the plots? If <code>FALSE</code> , the index of the column is shown.
addStats	should the coefficient in the trio logic regression model and the score for the fitted model be shown in the plot? Ignored if MCMC has been used in <code>trioLR</code> .
digits	number of digits used in the presentation of the coefficient and score (see <code>addStats</code>). Ignored if <code>addStats = FALSE</code> or MCMC has been used in <code>trioLR</code> .
main	character string specifying the title that should be added to the plot. If <code>NULL</code> , a standard title will be added to the plot.
cexOper	the relative size of the AND- and OR-operators in the plotting of the logic tree. Ignored if MCMC has been used in <code>trioLR</code> .
cexLeaf	the relative size of the variable names shown in the logic tree. Ignored if MCMC has been used in <code>trioLR</code> .
sizeLeaf	the relative size of the boxes representing the leaves in the logic trees. Ignored if MCMC has been used in <code>trioLR</code> .
cexPar	the relative size of the coefficient and the score (see <code>addStats</code>) when plotting the logic tree. Ignored if <code>addStats = FALSE</code> or if MCMC has been used in <code>trioLR</code> .
...	ignored.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>, based on the plot functions implemented by Ingo Ruczinski and Charles Kooperberg in the R package `LogicReg`.

References

- Kooperberg, C. and Ruczinski, I. (2005). Identifying Interacting SNPs Using Monte Carlo Logic Regression. *Genetic Epidemiology*, 28, 157-170.
- Li, Q., Fallin, M.D., Louis, T.A., Lasserter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.
- Ruczinski, I., Kooperberg, C., and LeBlanc, M.L. (2003). Logic Regression. *Journal of Computational and Graphical Statistics*, 12, 475-511.

See Also

[trioLR](#)

poly4root

Roots of a Fourth Degree Polynomial

Description

While `poly4root` computes the (real-valued) roots of a polynomial of fourth degree, `poly4rootMat` can be applied to several polynomials of fourth degree at once by assuming that each row the input matrix contains the coefficients for one of the polynomials.

Usage

```
poly4root(a)
```

```
poly4rootMat(amat)
```

Arguments

- | | |
|-------------------|--|
| <code>a</code> | a numeric vector of length five specifying the coefficients of the polynomial $a[1]*x^4 + a[2]*x^3 + a[3]*x^2 + a[4]*x + a[5]$. |
| <code>amat</code> | a numeric matrix with five columns in which each row contains the five coefficients of a polynomial of fourth degree. |

Value

For `poly4root`, a vector containing the real-valued roots of the polynomial. For `poly4rootMat`, a matrix with four columns in which each row contains the real-valued roots of the corresponding polynomial. If a polynomial has less than four real-valued roots, the remaining entries in the corresponding row are set to NA.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

print.colGxE *Printing and Storing of colGxE objects*

Description

Prints the statistics computed with `colGxE`. `getGxEstats` generates a data frame containing these statistics.

Usage

```
## S3 method for class 'colGxE'
print(x, top = 5, digits = 4, onlyGxE = FALSE, ...)

getGxEstats(x, top = NA, sortBy = c("none", "gxe", "lrt2df", "wald2df", "lrt1df", "g"))
```

Arguments

<code>x</code>	an object of class <code>colGxE</code> , i.e. the output of the function <code>colGxE</code> .
<code>top</code>	number of top interactions that should be printed or stored in a data frame. If <code>top</code> is set to <code>NA</code> , 0, or to a value that is negative or larger than the number of interactions, then the statistics for all interactions are printed or stored in the same order as they were in the genotype matrix <code>mat.snp</code> used in <code>colGxE</code> . Otherwise, the <code>top</code> interactions with the smallest p-values are printed or stored, where <code>print</code> uses the p-values of the GxE effect to order the interactions, while in <code>generateGxEstats</code> the p-values of test specified by <code>sortBy</code> are employed. Ignored if <code>sortBy = "none"</code> .
<code>onlyGxE</code>	logical indicating whether only the statistics for the parameter of the GxE interaction should be printed. If <code>FALSE</code> , the statistics for both parameters in the model as well as the odds ratios for the exposed trios and statistics for the 2 df likelihood ratio test and the 2 df Wald test (if these odds ratios and statistics were computed by <code>colGxE</code>) are shown.
<code>digits</code>	number of digits that should be printed.
<code>...</code>	ignored.
<code>sortBy</code>	character string specifying by the p-value of which test the SNPs should be sorted. If <code>"none"</code> (default), the SNPs are not sorted and the SNPs are in the same order as in the genotype matrix used to specify <code>mat.snp</code> in <code>colGxE</code> .

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Schwender, H., Taub, M.A., Beaty, T.H., Marazita, M.L., and Ruczinski, I. (2011). Rapid Testing of SNPs and Gene-Environment Interactions in Case-Parent Trio Data Based on Exact Analytic Parameter Estimation. *Biometrics*. DOI: 10.1111/j.1541-0420.2011.01713.x.

See Also[colGxE](#)

print.trioFS	<i>Printing and plotting of a trioFS object</i>
--------------	---

Description

Prints or plots the most important interactions found in a trioFS analysis.

Usage

```
## S3 method for class 'trioFS'
print(x, topX = 5, show.prop = TRUE, coded = FALSE, digits = 2, ...)

## S3 method for class 'trioFS'
plot(x, topX = 15, show.prop = FALSE, coded = TRUE, cex = 0.9,
     pch = 16, col = 1, force.topX = FALSE, include0 = TRUE, add.v0 = TRUE,
     v0.col = "grey50", main = NULL, ...)
```

Arguments

x	an object of class trioFS, i.e. the output of trioFS .
topX	integer specifying how many interactions should be shown. If topX is larger than the number of interactions contained in x, all the interactions are shown. Additionally to the topX most important interactions, any interaction having the same importance as the topX most important one are printed or (if force.topX = FALSE) plotted.
show.prop	should the proportions of models containing the respective interactions be added to the output (if print is used)? If the output of trioFS should be plotted, then the proportions of models can be plotted instead of the values of the importance measure by setting show.prop = TRUE.
coded	should the coded variable names be displayed? Might be useful if the actual variable names are pretty long. The coded variable name of the j-th variable is X _j .
digits	number of digits shown in the printed output.
cex	a numeric value specifying the relative size of the text and symbols.
pch	specifies the used symbol. See the help of par for details.
col	the color of the text and the symbols. See the help of par for how colors can be specified.
force.topX	if TRUE exactly topX interactions are plotted. If FALSE (default) all interactions up to the topXth most important one and all interactions having the same importance as the topXth most important one are plotted.

include0	should the x -axis include zero regardless whether the importances of the shown interactions are much higher than 0?
add.v0	should a vertical line be drawn at $x = 0$? Ignored if include0 = FALSE and all importances are larger than zero.
v0.col	the color of the vertical line at $x = 0$. See the help page of par for how colors can be specified.
main	character string naming the title of the plot. If NULL, a standard title is added to the plot.
...	Ignored.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

See Also

[trioFS](#)

print.trioLR

Printing of trioLR Objects

Description

Prints information on the trio logic regression model(s) fitted with [trioLR](#).

Usage

```
## S3 method for class 'trioLR'
print(x, asDNF=FALSE, posBeta=FALSE, digits = 3, ...)
```

Arguments

x	an object of class trioLR, i.e.\ the output of trioLR .
asDNF	should the disjunctive normal form of the logic expression represented by the logic tree be printed? If FALSE, the logic expression is printed as found by the search algorithm in trio logic regression. An advantage of the disjunctive normal form representation is that the interactions comprised by the logic expression are given by the AND-combinations in the disjunctive normal form. Note that not necessarily the minimum disjunctive normal form is printed so that all interactions comprised by the model are shown, even if some of the interactions are redundant for the evaluating the logic tree.
posBeta	should the disjunctive normal form be determined as if the sign of the coefficient in trio logic regression model is positive? If FALSE, the sign is ignored when transforming the logic tree into its disjunctive normal form. If TRUE and the coefficient is negative, the complement of the logic expression is transformed into its disjunctive normal form and the coefficient is multiplied by -1. Ignored if asDNF = FALSE or the fitted logic tree only contains one leaf.

digits number of digits used in the printing of the score and the parameter estimate of the fitted trio logic regression model(s).

... ignored.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>, based on the plot functions implemented by Ingo Ruczinski and Charles Kooperberg in the R package LogicReg.

References

Kooperberg, C. and Ruczinski, I. (2005). Identifying Interacting SNPs Using Monte Carlo Logic Regression. *Genetic Epidemiology*, 28, 157-170.

Li, Q., Fallin, M.D., Louis, T.A., Lasserter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.

Ruczinski, I., Kooperberg, C., and LeBlanc, M.L. (2003). Logic Regression. *Journal of Computational and Graphical Statistics*, 12, 475-511.

See Also

[trioLR](#)

read.pedfile	<i>Reading a Ped File</i>
--------------	---------------------------

Description

Reads a ped file into R and creates a data frame in ped format, or transform the ped file into a matrix in genotype format.

Usage

```
read.pedfile(file, first.row = NA, coded = NULL, naVal = 0, sep = " ",
             p2g = FALSE, non.rs.IDs = FALSE, cols4ID=FALSE)
```

Arguments

file the filename (if necessary with path) of a ped file that should be read into R.

first.row logical indicating whether the first row of file also contains data for a subject. If FALSE, the first row is assumed to contain the SNP names. By default, read.pedfile tries to figure out automatically if the first column contains the SNP names or data for a subject.

coded a character string stating how the alleles of the SNPs are coded. Possible values are "12", "AB", "1234", "ATCG". For details, see [ped2geno](#). By default, read.pedfile tries to figure out automatically how the alleles are coded.

naVal	value or character string specifying how missing values in the SNP data are coded.
sep	character string specifying how the SNP names in the first row of file are separated. Ignored if <code>first.row = TRUE</code> .
p2g	logical indicating whether the ped file should be transformed into a matrix in genotype format. If <code>FALSE</code> , a data frame in ped format is returned. Otherwise, ped2geno is called within <code>read.pedfile</code> to transform the data frame into a matrix in genotype format, and the matrix is returned.
non.rs.IDs	logical indicating whether (some of) the SNP names are specified by other names than rs-IDs.
cols4ID	logical indicating whether columns should be added to output matrix containing the family ID and the individual ID. If <code>FALSE</code> , the individual IDs are used as the row names of the output matrix.

Value

A data frame in ped format (if `p2g = FALSE`), or a matrix in genotype format (if `p2g = TRUE`).

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

See Also

[ped2geno](#)

removeSNPs

Remove SNPs or Trios

Description

Functions for removing SNPs with a low minor allele frequency or a high percentage of missing values, for removing trios in which at least one member shows a high percentage of missing values, for ordering the SNPs by their position in the genome, and for computing the minor allele frequencies of the SNPs based on only the genotypes of the parents, where each parent is only used once in this computation, even if this person is part of more than one of the trios.

Usage

```
removeSNPs(geno, maf = NA, perc.na = NA)
```

```
removeTrios(geno, perc.na = 1)
```

```
orderSNPs(geno, map, snp = "SNP", orderBy = c("Chr", "Position"))
```

```
colMAFtrio(geno)
```

Arguments

geno	a matrix in genotype format, i.e. the output of <code>ped2geno</code> or <code>read.pedfile</code> with <code>p2g</code> set to <code>TRUE</code> .
maf	a numeric value. If specified, i.e. not <code>NA</code> , all SNPs with a minor allele frequency less than <code>maf</code> are removed, where <code>maf</code> can range from 0 and 0.2. If, e.g., <code>maf = 0</code> , monomorphic SNPs are removed.
perc.na	a numeric value between 0 and 1 specifying a cutoff for the percentage of missing values that a SNP or a subject is allowed to have. If more than $100 * \text{perc.na}\%$ of the genotypes of a SNP or a subject is missing, then this SNP or the trio to which this subject belong, respectively, is removed <code>geno</code> .
map	a data frame containing the chromosome and the position for all the SNPs in <code>geno</code> .
snp	a character string giving the (case-sensitive) name of the column of <code>map</code> containing the SNP IDs used as column names in <code>geno</code> .
orderBy	character string of length 2 specifying the (case-sensitive) names of the columns of <code>map</code> containing the chromosomes and the positions of the SNPs in <code>geno</code> .

Value

For `removeSNPs`, `removeTrios`, and `orderSNPs`, a reduced or ordered version of `geno`. For `colMAFtrio`, a vector containing the minor allele frequencies of the SNPs in `geno`.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

scoreTDT

Score Tests for SNPs, GxE, and GxG Interactions

Description

Performs score tests for all individual SNPs (`scoreTDT`), all interactions of each SNP with an environmental variable (`scoreGxE`), or all interactions of two SNPs (`scoreGxG`) comprised by an input matrix based on the same log-likelihood considered in the corresponding genotypic TDT, where in `scoreGxG` the conditional logistic regression model including only one parameter (for the interaction effect) is used.

Additionally, the maximum over the score statistics for testing an additive, dominant, and recessive effect can be determined using `scoreMaxStat`.

Usage

```

scoreTDT(mat.snp, model = c("additive", "dominant", "recessive"), size = 20)

scoreGxE(mat.snp, env, model = c("additive", "dominant", "recessive"), size = 20,
  famid = NULL)

scoreGxG(mat.snp, model = c("additive", "dominant", "recessive"), genes = NULL,
  size = 20)

scoreMaxStat(mat.snp, size = 20)

## S3 method for class 'scoreTDT'
print(x, top = 5, digits = 4, ...)

## S3 method for class 'scoreGxE'
print(x, top = 5, digits = 4, onlyGxE = FALSE, ...)

## S3 method for class 'maxScoreTrio'
print(x, top = 5, digits = 4, ...)

```

Arguments

<code>mat.snp</code>	a numeric matrix in which each column represents a SNP. Each column must be a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. This matrix might be generated from a ped-file by, e.g., employing ped2geno .
<code>model</code>	type of model that should be fitted. Abbreviations are allowed. Thus, e.g., <code>model = "dom"</code> will fit a dominant model, and <code>model = "r"</code> an recessive model.
<code>size</code>	the number of models considered simultaneously when computing the parameter estimates.
<code>env</code>	a vector of length t (see <code>mat.snp</code>) containing for each offspring the value of a binary environmental variable, which must take the values 0 and 1.
<code>famid</code>	a vector of the same length as <code>env</code> specifying the family IDs for the corresponding values of the environmental variable in <code>env</code> . Can be used to reorder the vector <code>env</code> when the order of the trios differs between <code>env</code> and <code>mat.snp</code> .
<code>genes</code>	a character vector containing the names of the genes (or LD-blocks or other genetic sets of SNPs) to which the SNPs belong. If specified, only the two-way interactions between SNPs from different genes (or LD-blocks or other genetic sets of SNPs) are tested. If NULL, all two-way interactions between all possible pairs of SNPs are tested.
<code>x</code>	an object of class <code>scoreTDT</code> , <code>scoreGxE</code> , or <code>maxScoreTrio</code> , i.e. the output of the function <code>scoreTDT</code> / <code>scoreGxG</code> , <code>scoreGxE</code> , or <code>scoreMaxStat</code> , respectively.
<code>digits</code>	number of digits that should be printed.

top	number of interactions that should be printed. If the number of interactions is smaller than or equal to top, then the statistics for all interactions are printed in the order of their computation. Otherwise, the top interactions with the smallest p-values are printed.
onlyGxE	logical indicating whether only the statistics for the parameter of the GxE interaction should be printed. If FALSE, the statistics for both parameters in the model are shown.
...	ignored.

Value

For scoreTDT and scoreGxG, an object of class scoreTDT containing numeric vectors

score	the scores for all SNPs or SNP interactions,
info	the denominators of the corresponding score statistics
,	
stat	the values of the score statistics for all SNPs or SNP interactions
,	
pval	the corresponding p-values computed based on a ChiSquare-distribution with 1 degree of freedom.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

See Also

[colTDT](#), [colGxE](#), [colTDT2way](#)

 tdt | *Genotypic TDT* |**Description**

Computes the genotypic TDT for a SNP or for each column of a matrix representing a SNP.

Usage

```
tdt(snp, model = c("additive", "dominant", "recessive"))

colTDT(mat.snp, model = c("additive", "dominant", "recessive"),
       size = 50)

## S3 method for class 'tdt'
print(x, digits = 4, ...)

## S3 method for class 'colTDT'
print(x, top = 5, digits = 4, ...)
```

Arguments

snp	a numeric vector of length $3 * t$ representing a SNP genotyped at t trios. Each of the t blocks (i.e. <code>snp[1:3]</code> , <code>snp[4:6]</code> , ...) must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. The vector must thus have the same structure as the output of <code>trio.check</code> , or the genotype example data sets such as <code>trio.gen1</code> (see <code>data(trio.gen1)</code>), and can be generated from a ped-file by, e.g., employing <code>ped2geno</code> .
mat.snp	a numeric matrix in which each column represents a SNP. Each of the SNPs must have the same structure as <code>snp</code> , and can, e.g., be generated from a ped-file by employing <code>ped2geno</code> .
model	type of model that should be fitted. Abbreviations are allowed. Thus, e.g., <code>model = "dom"</code> will fit a dominant model, and <code>model = "r"</code> an recessive model.
size	the number of SNPs considered simultaneously when computing the parameter estimates. Ignored if <code>fast = FALSE</code> .
x	an object of class <code>tdt</code> or <code>colTDT</code> , i.e. the output of the function <code>tdt</code> (or <code>tdtGxG</code>) or the function <code>colTDT</code> .
digits	number of digits that should be printed.
top	number of interactions that should be printed. If <code>top</code> is less than or equal to zero, set to NA, or larger than the number of SNPs, then the statistics for all SNPs are printed in the order as they were in the genotype matrix used as input into <code>colTDT</code> . Otherwise, the top interactions with the smallest p-values are printed.
...	ignored.

Value

An object of class `tdt` or `colTDT` consisting of the following numeric values or vectors, respectively:

coef	the estimated parameter,
se	the estimated standard deviation of the parameter estimate,
stat	Wald statistic,
OR	the odds ratio, i.e. $\exp(\text{coef})$

,
 lowerOR the lower bound of the 95% confidence interval for OR,
 upperOR the upper bound of the 95% confidence interval for OR,
 usedTrios the number of trios affecting the parameter estimation (only for colTDT),
 ... further internal parameters

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Schwender, H., Taub, M.A., Beaty, T.H., Marazita, M.L., and Ruczinski, I. (2011). Rapid Testing of SNPs and Gene-Environment Interactions in Case-Parent Trio Data Based on Exact Analytic Parameter Estimation. *Biometrics*. DOI: 10.1111/j.1541-0420.2011.01713.x.

See Also

[tdt2way](#), [ped2geno](#)

 tdtGxG

Genotypic TDT for Two-Way Interactions

Description

tdtGxG and colGxG perform the genotypic TDT for the interaction of two SNPs or of each pair of columns of a genotype matrix, respectively.

fastGxG provides a fast implementation for the genotypic TDT for two-way interactions when considering the simplest conditional logistic regression model only containing one parameter for the interaction effect. It thus leads to the same results as colGxG with `test = "screen"`. In fastGxGrec, an analytic solution to the genotypic TDT based on the simplest model for testing a recessive x recessive model is implemented, which is even faster than fastGxG with `model = "recessive"`. In future versions of this package, fastGxG and fastGxGrec will be joint with colGxG.

The genotypic TDT for testing two-way interactions makes use of the 16 possible genotypes that can be obtained from combining the parents' genotypes of the two considered SNPs. Thus, for each family, genotypes for one case (i.e. the affected offspring) and 15 pseudo-controls are used.

Usage

```
tdtGxG(snp1, snp2, test = c("epistatic", "lrt", "full", "screen"),
      model = c("additive", "dominant", "recessive"))
```

```
colGxG(mat.snp, test = c("epistatic", "lrt", "full", "screen"), genes = NULL,
      maf = FALSE, model = c("additive", "dominant", "recessive"))
```

```
fastGxG(mat.snp, model = c("additive", "dominant", "recessive"),
        genes = NULL, interval = c(-10, 10), tol = 10^-8, maxiter = 1000,
        size = 20)
```

```
fastGxGrec(mat.snp, genes = NULL, size = 20)
```

Arguments

snp1, snp2	numeric vectors of length $3 * t$ representing two SNPs genotyped at t trios. Each of the t blocks (i.e. snp1[1:3], snp1[4:6], ..., and snp2[1:3], snp2[4:6], ...) must consist of the genotypes of father, mother, and offspring (in this order). The genotypes must be coded by 0, 1, and 2. Missing values are allowed and need to be coded by NA. The vectors must thus have the same structure as the output of trio.check , or the genotype example data sets such as <code>trio.gen1</code> (see <code>data(trio.gen1)</code>), and can be generated from a ped-file by, e.g., employing ped2geno .
mat.snp	a numeric matrix in which each column represents a SNP. Each of the SNPs must have the same structure as snp, and can, e.g., be generated from a ped-file by employing ped2geno .
test	character string naming the GxG test that should be performed. If test = "epistatic", then a conditional logistic regression version of the test proposed by Cordell (2002) is used to test for epistatistical interactions. If test = "full", a conditional logistic regression model containing one parameter for each SNP and one parameter for the interaction of these two SNPs will be fitted and a Wald test for the interaction term will be performed, where a genetic model specified by model is assumed for both SNPs. If test = "lrt", a likelihood ratio test is performed comparing the fit of this model with the fit of a conditional logistic regression model only containing the two parameters for the main effects of the SNPs. If test = "screen", a conditional logistic regression model only composed of one parameter for the interaction of the two SNPs will be fitted and a Wald test will be performed, where the genetic model specified by model is assumed for both SNPs.
genes	a character vector containing the names of the genes to which the SNPs belong. If specified, only the two-way interactions between SNPs from different genes are tested. If NULL, all two-way interactions between all possible pairs of SNPs are tested.
maf	logical indicating whether the minor allele frequency (computed by considering the genotypes of only the parents) should be added to the output.
model	type of model that should be considered. Abbreviations are allowed. Thus, e.g., model = "dom" will consider a dominant model for each of the respective two SNPs, and model = "r" an recessive model. Ignored if epistatic = TRUE.
interval	the end-points of the interval to be searched for the root. For details, see uniroot .
tol	the desired accuracy/convergence tolerance. For details, see uniroot .
maxiter	the maximum number of iterations. For details, see uniroot .
size	the number of interactions considered simultaneously when computing the parameter estimates.

Value

Depending on test, the output contains statistics and p-values either of a likelihood ratio test (test = "epistatic" or test = "lrt") or the Wald statistics and the corresponding p-values for the interaction term in the conditional logistic regression model (test = "full" or test = "screen"). If maf = TRUE, a vector maf containing the minor allele frequencies of each SNP and a matrix mat.maf with two columns containing the SNP-wise minor allele frequencies for each tested pair of SNPs are added to the output of colGXG.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Cordell, H. J. (2002). Epistasis: What it Means, what it Doesn't mean, and Statistical Methods to Detect it in Humans. *Human Molecular Genetics*, 11, 2463-2468.

Schwender, H., Taub, M.A., Beaty, T.H., Marazita, M.L., and Ruczinski, I. (2011). Rapid Testing of SNPs and Gene-Environment Interactions in Case-Parent Trio Data Based on Exact Analytic Parameter Estimation. *Biometrics*. DOI: 10.1111/j.1541-0420.2011.01713.x.

See Also

[tdt](#), [ped2geno](#)

trio.check

Check Case-Parent Trio Data for Mendelian Errors

Description

This function checks case-parent trio data in linkage or genotype format for Mendelian errors. If no errors are found, the function returns an object suitable for input to the [trio.prepare](#) function. Otherwise, an object identifying the Mendelian errors is returned.

Usage

```
trio.check(dat, is.linkage=TRUE, replace=FALSE)
```

Arguments

dat A matrix or data frame of pedigree data in linkage format, or in genotype format. If the data are in **linkage format**, the file has to have the standard linkage/pedigree format. Each row describes an individual, and the columns are <famid> <pid> <fatid> <motid> <sex> <affected> <genotype:1_1> <genotype:1_2> ... <genotype:n_1> <genotype:n_2>. Here, <famid> is a unique identifier for each family, <pid> is a unique identifier for an individual within each family, <fatid> and <motid> identify the father and mother of the individual, <sex> denotes the gender, using the convention 1=male, 2=female, <affected> denotes the disease

status (0=unknown, 1=unaffected, 2=affected). Only one phenotype column is allowed. Each genotype is encoded using two columns (`<genotype:k_1>` and `<genotype:k_2>`), identifying the alleles (1 for the major allele, 2 for the minor allele, 0 if missing). Other values for the alleles will result in an error. Please see the data frames `trio.ped1` and `trio.ped2` contained in this package as examples for trio data in linkage file format (complete and with missing records, respectively).

If the data are in **genotype format**, each row in the object describes an individual, and each block of three consecutive rows describes the two parents and the affected child in a trio. The columns in the object are `<famid>` `<pid>` `<genotype_1>` ... `<genotype_n>`. Here, `<famid>` is a unique identifier for each family, `<pid>` is a unique identifier for an individual within each family (with each block of three consecutive rows describing the two parents and the affected child in a trio). Each `<genotype>` is encoded as an integer indicating the number of variant alleles (e.g. 0=common homozygote, 1=heterozygote, and 2=rare homozygote, and NA=missing genotype). Please see the data frames `trio.gen1` and `trio.gen2` contained in this package as examples for trio data in linkage file format (complete and with missing records, respectively).

<code>is.linkage</code>	A logical value indicating if the case parent data are in linkage file format (TRUE) or in genotype format (FALSE).
<code>replace</code>	A logical value indicating whether existing Mendelian errors should be replaced by missing values. For each Mendelian error found (for a particular trio at a particular locus), all three genotypes are replaced by NA, and an object suitable for input to the <code>trio.prepare</code> function is returned.

Details

The first function used from this package should always be `trio.check`. Unless otherwise specified, this function assumes that the data are in linkage format, however, genotype data can also be accommodated. If no Mendelian inconsistencies in the data provided are identified, `trio.check` creates an object that can be processed in the subsequent analysis with the `trio.prepare` function. If the data were in linkage format, the genotype information for each SNP will be converted into a single variable, denoting the number of variant alleles.

To delineate the genotype information for the pseudo-controls in the subsequent analysis, the trio data must not contain any Mendelian errors. The function `trio.check` returns a warning, and an R object with relevant information when Mendelian errors are encountered in the supplied trio data. It is the users responsibility to find the cause for the Mendelian errors and correct those, if possible. However, Mendelian inconsistencies are often due to genotyping errors and thus, it might not be possible to correct those in a very straightforward manner. In this instance, the user might want to encode the genotypes that cause these Mendelian errors in some of the trios as missing data. The function `trio.check` allows for this possibility, using the argument `replace=T`.

Value

The function `trio.check` returns a list with the following elements:

<code>trio</code>	A data frame with the genotypes of the trios, suitable for input to the function <code>trio.prepare</code> . This element will be NULL if Mendelian errors are detected.
-------------------	--

errors	This element will be NULL if no Mendelian errors are detected. Otherwise, this element will be a data frame with five columns, indicating the Mendelian errors detected in the object dat. The five columns of the data frame refer to the trio (trio), the family id (famid), the genotype (snp), the row numbers (r), and the column numbers (c).
trio.err	This element will be NULL if no Mendelian errors are detected. Otherwise, this element will be a data frame with the trio genotype data. If the input was a linkage file, the data will be converted from alleles to genotypes. If the input was a genotype file, this element will be identical to the input.

Author(s)

Qing Li, mail2qing@yahoo.com

References

Li, Q., Fallin, M.D., Louis, T.A., Lasserter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.

See Also

[trio.prepare](#)

Examples

```
## Not run:
data(trio.data)
trio.tmp = trio.check(dat=trio.ped1)
str(trio.tmp, max=1)
trio.tmp$trio[1:6,]

trio.tmp = trio.check(dat=trio.ped.err)
str(trio.tmp, max=1)
trio.tmp$errors
trio.tmp$trio.err[1:3, c(1,2, 11:12)]
trio.ped.err[1:3,c(1:2, 23:26)]

trio.tmp = trio.check(dat=trio.gen.err, is.linkage=FALSE)
trio.tmp$errors
trio.tmp$trio.err[1:6, c(1,2,7), drop=FALSE]

trio.rep = trio.check(dat=trio.gen.err, is.linkage=FALSE, replace=TRUE)
trio.rep$trio[1:6,c(1,2,7)]

## End(Not run)
```

trio.data

*Case-Parent Trio Data***Description**

trio.data contains several simulated data sets used in the different examples for the analyses with the functions in the R package trio.

For the applications of genotypic TDTs for individual SNPs and two-way interactions with, for example, `tdt` and `tdt2way`, respectively, trio.data contains a 300 x 6 matrix called `mat.test` consisting of genotype data for 100 trios genotyped at 6 SNPs.

For the preparation of the trio data for an application of trio logic regression with `trio.check` and `trio.prepare`, trio.data contains different data set containing genotype data for 10 SNPs in 100 trios in different formats.

trio.gen1, trio.gen2, and trio.gen.err consist of 12 columns and 300 rows, where the first two columns contain family identifier and individual identifier. In the columns afterwards, each SNPs is encoded in one variable denoting the number of minor alleles.

trio.ped1, trio.ped2, and trio.ped.err consist of 26 columns and 300 rows, where the first six columns identify the family structure of the data, and the phenotype. Besides the variables providing information on the family structure and the phenotypes (columns 1 to 6), each SNPs is encoded in two variables denoting the alleles.

Contrary to the other data sets, trio.gen.err and trio.ped.err contain Mendelian errors.

For the application of the functions `getLD` and `findLDblocks` for computing the pairwise LD values and for detecting the LD blocks, respectively, trio.data contains a 500 x 50 matrix called `LDblock` that is composed of genotype data for 10 LD blocks each consisting of 5 SNPs in strong LD.

Finally, for the simulation of trio data with `trio.sim`, trio.data contains examples for haplotype frequencies used in these simulations. Both `freq.hap` and `simuBkMap` are data.frames containing haplotype information, including the haplotype block identifier, haplotype, and haplotype frequency. While `freq.hap` is a data frame consisting of 20 rows and 3 columns, `simuBkMap` consists of 66 rows and 3 columns. `step3way` is a list internally used for simulation, containing some indexes and sampling frequencies.

Author(s)

LDdata and `mat.test`: Holger Schwender, <holger.schwender@udo.edu>; all other data sets: Qing Li, <mail2qing@yahoo.com>

trio.permTest

*Permutation Tests for Trio Logic Regression***Description**

Performs either a null-model or a conditional permutation test for a trio logic regression analysis.

Usage

```
trio.permTest(object, conditional = FALSE, n.perm = 10, nleaves = NULL,
              control = NULL, rand = NA)
```

Arguments

object	an object of class <code>trioLR</code> , i.e. the output of the function <code>trioLR</code> . This object must be the result of a trio logic regression analysis in which a single model has been fitted (i.e. in <code>trioLR</code> , <code>search</code> must have been set to "sa" and <code>nleaves</code> must have been a single integer).
conditional	should the conditional permutation test be performed? If <code>FALSE</code> , a null-model permutation test is done analogously to the null-model permutation test for a standard logic regression for population-based data implemented in the function <code>logreg</code> of the R package <code>LogicReg</code> . If <code>TRUE</code> , a test analogous to the conditional permutation test for a standard logic regression is performed.
n.perm	integer specifying the number of permutations.
nleaves	integer specifying the maximum number of leaves that the logic tree in the trio logic regression model is allowed to have. If <code>NULL</code> , the maximum number of leaves saved in <code>object</code> is used.
control	a list containing the control parameters for the search algorithms and the logic tree considered in <code>trioLR</code> , where the parameters for an MCMC run and the logic tree are ignored. If <code>NULL</code> (i.e. by default), the same values for the parameters are used that have been employed in the original analysis with <code>trioLR</code> . If other values should be used, it is highly recommended to specify <code>control</code> by employing <code>lrControl</code> .
rand	an integer. If specified, the random number generator will be set into a reproducible state.

Value

A list consisting of

origScore	NA, if <code>conditional = FALSE</code> , and otherwise, the score, i.e. the value of the partial likelihood, of the original model saved in <code>object</code>
,	
permScore	a vector of length <code>n.perm</code> containing the scores for the trio logic regression models built in the iterations of the permutation test.

Author(s)

Qing Li, <mail2qing@yahoo.com>. Modified by Holger Schwender.

References

Li, Q., Fallin, M.D., Louis, T.A., Lasseter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.

See Also[trioLR](#)

`trio.prepare`*Generate Trio Data Format Suitable for Trio Logic Regression*

Description

This function transforms case-parent data into a format suitable as input for trio logic regression. The function can also be used for the imputation of missing genotypes in case-parent data, while taking the existing SNP block structure into account.

Usage

```
trio.prepare(trio.dat, freq=NULL, blocks=NULL, logic=TRUE, ...)
```

Arguments

<code>trio.dat</code>	An object returned from the function trio.check .
<code>freq</code>	An optional data frame specifying haplotype blocks and frequencies. For an example, see the data frame <code>simuBkMap</code> contained in this package. If provided, the following argument <code>blocks</code> will be ignored. The object must have three columns in the following order: block identifiers (<code>key</code>), haplotypes (<code>hap</code>), and haplotype frequencies (<code>freq</code>). The block identifiers must be unique for each block. For each block, the haplotypes must be encoded as a string of the integers 1 and 2, where 1 refers to the major allele and 2 refers to the minor allele. The respective haplotype frequencies will be normalized to sum one.
<code>blocks</code>	An optional vector of integers, specifying (in sequence) the lengths of the linkage disequilibrium blocks. The sum of these integers must be equal to the total numbers of SNPs in the data set used as input. Using the integer 1 for SNPs not contained in LD blocks is required if this argument is used. If both arguments <code>freq</code> and <code>blocks</code> are <code>NULL</code> , complete linkage equilibrium is assumed (i.e., no correlation between the genotypes).
<code>logic</code>	A logical value indicating whether the trio data are returned with genotypes in dominant and recessive coding, suitable as input for trio logic regression (<code>TRUE</code>), or if the imputed data should be returned in genotype format, using one variable per SNP (<code>FALSE</code>).
<code>...</code>	Optional arguments that can be passed to function haplo.em .

Details

To create the genotypes for the pseudo-controls it is necessary to take the LD structure of the SNPs into account. This requires information on the LD blocks. It is assumed that the user has already delineated the block structure according to his or her method of choice. The function `trio.prepare`, which operates on an output object of `trio.check`, accepts the block length information as an argument. If this argument is not specified, a uniform block length of 1 (i.e., no LD structure) is assumed. If the haplotype frequencies are not specified, they are estimated from the parents' genotypes using the function `haplo.em`. The function then returns a list that contains the genotype information in binary format, suitable as input for trio logic regression. Since trio logic regression requires complete data, the function `trio.prepare` also performs an imputation of the missing genotypes. The imputation is based on the estimated or supplied haplotype information.

Value

<code>bin</code>	A matrix suitable as input for trio logic regression. The first column specifies the cases and pseudo-controls as required by logic regression using conditional logistic regression (the integer 3 for the probands followed by three zeros indicating the pseudo-controls). The following columns specify the (possibly imputed) genotypes in dominant and recessive coding, with two binary variables for each SNP. This is returned only if <code>logic = TRUE</code> .
<code>trio</code>	A data frame with imputed SNPs in genotype format derived from the input. This is returned only if <code>logic = FALSE</code> .
<code>miss</code>	A data frame with five columns indicating the missing genotypes in the input object. The five columns of the data frame refer to the family id (<code>famid</code>), the individual id (<code>pid</code>), the genotype (<code>snp</code>), the row numbers (<code>r</code>), and the column numbers (<code>c</code>). This element will be <code>NULL</code> if there are no missing data.
<code>freq</code>	The estimated or supplied haplotype information, in the same format as described in the Arguments above.

Acknowledgments

Support was provided by NIH grants R01 DK061662 and HL090577.

Author(s)

Qing Li, mail2qing@yahoo.com

References

Li, Q., Fallin, M.D., Louis, T.A., Lasserter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.

See Also

[trio.check](#), [haplo.em](#)

Examples

```
## Not run:
data(trio.data)
trio.tmp <- trio.check(dat=trio.ped1)
trio.bin <- trio.prepare(trio.dat=trio.tmp, blocks=c(1,4,2,3))
trio.bin$bin[1:8,]

## End(Not run)
```

trio.sim

Simulate Case-Parent Trios

Description

This function generates case-parents trios when the disease risk of children is specified by (possibly higher-order) SNP-SNP interactions. The SNP minor allele frequencies and/or haplotypes are specified by the user, as are the parameters in the logistic model that describes the disease risk.

Usage

```
trio.sim(freq, interaction="1R and 2D", prev=1e-3, OR=1, n=100, rep=1,
         step.save=NULL, step.load=NULL, verbose=FALSE)
```

Arguments

- | | |
|-------------|---|
| freq | A data frame specifying haplotype blocks and frequencies. For an example, see the data frame <code>simuBkMap</code> contained in this package. If provided, the following argument <code>blocks</code> will be ignored.

The object must have three columns in the following order: block identifiers (key), haplotypes (hap), and haplotype frequencies (freq). The block identifiers must be unique for each block. For each block, the haplotypes must be encoded as a string of the integers 1 and 2, where 1 refers to the major allele and 2 refers to the minor allele. The respective haplotype frequencies will be normalized to sum one. |
| interaction | A string that specifies the risk altering genotype interaction as a Boolean term, such as <i>"7D or 19R"</i> , or <i>"(not 10D) or 45D"</i> . Each locus can appear at most once in the string, and the Boolean term <i>not</i> can appear at most once before each locus, and must be enclosed in parenthesis, e.g., <i>"(not 3D)"</i> . Therefore, strings such as <i>"not (not 3D)"</i> and <i>"not 3D or 5R"</i> are prohibited. Parenthesis are also used to unambiguously define the Boolean expression as a binary tree, i.e., every parent node has exact two children. For example Thus, a long string such as <i>"1R or 3D or 5R"</i> must be written as <i>"(1R or 3D) or 5R"</i> or as <i>"1R or (3D or 5R)"</i> , even though the parenthesis are technically redundant. There is also a limit on the size of the interactions, please see Details below. |
| prev | The prevalence of the disease in the simulated population among non-carriers (the "un-exposed" group). |

OR	The odds ratio of disease in the simulated population, comparing carriers to non-carriers.
n	The number of case-parent trios simulated. The default is 100.
rep	The number of data set replicates generated. The default is 1.
step.save	The name of the binary file (without ".RData" extension) in which the object specifying the simulation mating tables and probabilities will be saved. The default value is NULL. In that case, the object will not be saved for re-use in later run. See Details .
step.load	The name of an existing binary file (without ".RData" extension) in which the object specifying the simulation mating tables and probabilities have been saved (see above). The default value is NULL. In that case, a new object will be generated.
verbose	A logical value indicating whether or not to print information about memory and time usage.

Details

The function `trio.sim` simulates case-parent trio data when the disease risk of children is specified by (possibly higher-order) SNP-SNP interactions. The mating tables and the respective sampling probabilities depend on the haplotype frequencies (or SNP minor allele frequencies when the SNP does not belong to a block). This information is specified in the `freq` argument of the function. The probability of disease is assumed to be described by the logistic term $\text{logit}(p) = a + b I[\text{Interaction}]$, where $a = \text{logit}(\text{prev})$ and $b = \log(\text{OR})$, with `prev` and `OR` specified by the user. Note that at this point only data for two risk groups (carriers versus non-carriers) can be simulated. Since the computational demands for generating the mating is dependent on the number of loci involved in the interactions and the lengths of the LD blocks that contain these disease loci, the interaction term can only consist of up to six loci, not more than one of those loci per block, and haplotype (block) lengths of at most 5 loci.

Generating the mating tables and the respective sampling probabilities necessary to simulate case-parent trios can be very time consuming for interaction models involving three or more SNPs. In simulation studies, many replicates of similar data are usually required, and generating these sampling probabilities in each instance would be a large and avoidable computational burden (CPU and memory). The sampling probabilities depend foremost on the interaction term and the underlying haplotype frequencies, and as long as these remain constant in the simulation study, the mating table information and the sampling probabilities can be "recycled". This is done by storing the relevant information (denoted as "step-stone") as a binary R file in the working directory (using the argument `step.save`), and loading the binary file again in future simulations (using the argument `step.load`), speeding up the simulation process dramatically. It is even possible to change the parameters `prev` and `OR` (corresponding to a and b in the logistic model) in these additional simulations, as the sampling probabilities can be adjusted accordingly.

Value

A list of matrices, containing the simulated data sets, in genotype format (indicating the number of variant alleles), including family and subject identifiers.

Author(s)

Qing Li, mail2qing@yahoo.com

References

Li, Q., Fallin, M.D., Louis, T.A., Lasseter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.

See Also

[trio.prepare](#)

Examples

```
## Not run:
data(trio.data)
sim = trio.sim(freq=simuBkMap, interaction="1R and 5R", prev=.001, OR=2, n=20, rep=1)
sim[[1]][1:6, 1:12]

## End(Not run)
```

trioFS

Trio Feature Selection

Description

Performs a trioFS (trio Feature Selection) analysis as proposed by Schwender et al. (2011) based on bagging/subsampling with base learner trio logic regression (Li et al., 2011).

Usage

```
## Default S3 method:
trioFS(x, y, B = 20, nleaves = 5, replace = TRUE, sub.frac = 0.632,
       control = lrControl(), fast = FALSE, addMatImp = TRUE, addModels = TRUE,
       verbose = FALSE, rand = NA, ...)

## S3 method for class 'trioPrepare'
trioFS(x, ...)
```

Arguments

x either an object of class `trioPrepare`, i.e. the output of [trio.prepare](#), or a binary matrix consisting of zeros and ones. If the latter, then each column of x must correspond to a binary variable (e.g., coding for a dominant or a recessive effect of a SNP), and each row to a case or a pseudo-control, where each trio is represented by a block of four consecutive rows of x containing the data for the case and the three matched pseudo-controls (in this order) so that the first four

rows of x comprise the data for the first trio, rows 5-8 the data for the second trio, and so on. Missing values are not allowed. A convenient way to generate this matrix is to use the function `trio.prepare`. Afterwards, `trioLR` can be directly applied to the output of `trio.prepare`.

<code>y</code>	a numeric vector specifying the case-pseudo-control status for the observations in x (if x is a binary matrix). Since in trio logic regression, cases are coded by a 3 and pseudo-controls by a 0, y is given by <code>rep(c(3, 0, 0, 0), n.trios)</code> , where <code>n.trios</code> is the number of trios for which genotype data is stored in x . Thus, the length of y must be equal to the number of rows in x . No missing values are allowed in y . If not specified, y will be automatically generated.
<code>B</code>	number of bootstrap samples or subsamples used in <code>trioFS</code>
<code>n.leaves</code>	maximum number of leaves, i.e. variables, in the logic tree considered in each of the B trio logic regression models (please note in trio logic regression the model consists only of one logic tree).
<code>replace</code>	should sampling of the trios be done with replacement? If <code>TRUE</code> , a Bootstrap sample of size <code>n.trios</code> is drawn from the <code>n.trios</code> trios in each of the B iterations. If <code>FALSE</code> , <code>ceiling(sub.frac * n.trios)</code> of the trios are drawn without replacement in each iteration.
<code>sub.frac</code>	a proportion specifying the fraction of trios that are used in each iteration to fit a trio logic regression model if <code>replace = FALSE</code> . Ignored if <code>replace = TRUE</code> .
<code>control</code>	a list of control parameters for the search algorithms and the logic trees considered when fitting the trio logic regression model, where the parameters for an MC logic regression are ignored. For details and the parameters, see <code>lrControl</code> , which is the function that should be used to specify <code>control</code> .
<code>fast</code>	should a greedy search be used instead of simulated annealing, i.e. the standard search algorithm in (trio) logic regression?
<code>addMatImp</code>	should the matrix containing the improvements due to the interactions in each of the iterations be added to the output, where the importance of each interaction is computed by the average over the B improvements due to this interaction?
<code>addModels</code>	should the B trio logic regression models be added to the output
<code>verbose</code>	should some comments on the progress the <code>trioFS</code> analysis be printed?
<code>rand</code>	positive integer. If specified, the random number generator is set into a reproducible state.
<code>...</code>	for the <code>trioPrepare</code> method, optional parameters to be passed to the low level function <code>trioFS.default</code> , i.e. all arguments of <code>trioFS.default</code> except for x and y . Otherwise, ignored.

Value

An object of class `trioFS` consisting of

<code>vim</code>	a numeric vector containing the values of the importance measure for the found interactions,
<code>prop</code>	a numeric vector consisting of the percentage of models that contain the respective found interactions,

primes	a character vector naming the found interactions,
param	a list of parameters used in the trioFS analysis, i.e. B, nleaves, and the sampling method,
mat.imp	if addMatImp = TRUE, a matrix containing the B improvements for each found interaction,
logreg.model	if addModel = TRUE, the B trio logic regression models,
inbagg	if addModel = TRUE, a list of length B in which each object specifies the trios used to fit the corresponding trio logic regression model.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Li, Q., Fallin, M.D., Louis, T.A., Lasserter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.

Schwender, H., Bowers, K., Fallin, M.D., and Ruczinski, I. (2011). Importance Measures for Epistatic Interactions# in Case-Parent Trios. *Annals of Human Genetics*, 75, 122-132.

See Also

[trioLR](#), [print.trioFS](#), [trio.prepare](#)

trioLR	<i>Trio Logic Regression</i>
--------	------------------------------

Description

Performs a trio logic regression analysis as proposed by Li et al. (2011), where trio logic regression is an adaptation of logic regression (Ruczinski et al., 2003) for case-parent trio data.

Usage

```
## Default S3 method:
trioLR(x, y, search = c("sa", "greedy", "mcmc"), nleaves = 5,
       penalty = 0, weights = NULL, control=lrControl(), rand = NA, ...)

## S3 method for class 'trioPrepare'
trioLR(x, ...)
```

Arguments

x	either an object of class <code>trioPrepare</code> , i.e. the output of <code>trio.prepare</code> , or a binary matrix consisting of zeros and ones. If the latter, then each column of x must correspond to a binary variable (e.g., coding for a dominant or a recessive effect of a SNP), and each row to a case or a pseudo-control, where each trio is represented by a block of four consecutive rows of x containing the data for the case and the three matched pseudo-controls (in this order) so that the first four rows of x comprise the data for the first trio, rows 5-8 the data for the second trio, and so on. Missing values are not allowed. A convenient way to generate this matrix is to use the function <code>trio.prepare</code> . Afterwards, <code>trioLR</code> can be directly applied to the output of <code>trio.prepare</code> .
y	a numeric vector specifying the case-pseudo-control status for the observations in x (if x is the binary matrix). Since in trio logic regression, cases are coded by a 3 and pseudo-controls by a 0, y is given by <code>rep(c(3, 0, 0, 0), n.trios)</code> , where <code>n.trios</code> is the number of trios for which genotype data is stored in x. Thus, the length of y must be equal to the number of rows in x. No missing values are allowed in y. If not specified, y will be automatically generated.
search	character string naming the search algorithm that should be used in the search for the best trio logic regression model. By default, i.e. <code>search = "sa"</code> , simulated annealing, the standard search algorithm for a logic regression is used. In this case, depending on the length of <code>nleaves</code> , either one trio logic regression model is fitted or several trio logic regression models of different sizes are fitted. For details, see <code>nleaves</code> . Alternatively, a greedy search can be used by setting <code>search = "greedy"</code> , or a MC logic regression analysis (Koopberg and Ruczinski, 2005) for case-parent trio data can be performed by setting <code>search = "mcmc"</code> .
nleaves	integer or vector of two integers specifying the maximum number of leaves, i.e. variables, in the logic tree of the trio logic regression model (please note in trio logic regression the model consists only of one logic tree). Must be a single integer, if <code>search = "greedy"</code> or <code>search = "mcmc"</code> . If <code>search = "sa"</code> , it can also be a vector of two integers, where the second integer must be larger than the first one. In this case, several trio logic regression models are fitted in which the maximum numbers of leaves range from <code>nleaves[1]</code> to <code>nleaves[2]</code> .
penalty	a non-negative value for the penalty parameter used in logic regression. The penalty takes the form <code>penalty</code> times the number of leaves in the model. By default, larger models are not penalized. <code>penalty</code> is only relevant when one logic regression model is fitted.
weights	a numeric vector containing one weight for each trio considered in x. Thus, <code>weights</code> must contain <code>nrow(x) / 4</code> positive values. By default, all trios are equally weighted.
control	a list of control parameters for the search algorithms and the logic tree considered when fitting a (trio) logic regression model. For these parameters, see <code>lrControl</code> , which is the function that should be used to specify <code>control</code> .
rand	integer. If specified, the random number generator will be set into a reproducible state.

... for the `trioPrepare` method, optional parameters to be passed to the low level function `trioLR.default`, i.e. all arguments of `trioLR.default` except for `x` and `y`. Otherwise, ignored.

Details

Trio logic regression is an adaptation of logic regression to case-parent trio data. Virtually all features for a standard logic regression analysis with the function `logreg` available in the R package `LogicReg` are also available for a trio logic regression analysis, either directly via `trioLR` or via the function `trio.permTest` for performing permutation tests.

For a detailed, comprehensive description on how to perform a logic regression analysis, and thus, a trio logic regression analysis, see the `Details` section of the help page for the function `logreg` in the R package `LogicReg`. For a detailed explanation on how to specify the parameters for simulated annealing, see the man page of the function `logreg.anneal.control` in the R package `LogicReg`.

Finally, an example for a trio logic regression analysis is given in the vignette `trio` available in the R package `trio`.

Value

An object of class `trioLR` composed of the same objects as an object of class `logreg`. For details, see the `Value` section of the function `logreg` from the R package `LogicReg`.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>, based on R and Fortran code by Ingo Ruczinski, <ingo@jhu.edu>, and Charles Kooperberg for a general logic regression as well as Fortran code by Ingo Ruczinski and Qing Li for trio logic regression.

References

- Kooperberg, C. and Ruczinski, I. (2005). Identifying Interacting SNPs Using Monte Carlo Logic Regression. *Genetic Epidemiology*, 28, 157-170.
- Li, Q., Fallin, M.D., Louis, T.A., Lasserter, V.K., McGrath, J.A., Avramopoulos, D., Wolyniec, P.S., Valle, D., Liang, K.Y., Pulver, A.E., and Ruczinski, I. (2010). Detection of SNP-SNP Interactions in Trios of Parents with Schizophrenic Children. *Genetic Epidemiology*, 34, 396-406.
- Ruczinski, I., Kooperberg, C., and LeBlanc, M.L. (2003). Logic Regression. *Journal of Computational and Graphical Statistics*, 12, 475-511.

See Also

[trio.prepare](#), [trio.check](#), [trio.permTest](#)

Index

- *Topic **IO**
 - read.pedfile, 27
 - *Topic **array**
 - allelicTDT, 2
 - colGxE, 3
 - colGxGPerms, 6
 - colTDTmaxTest, 8
 - colTDTsam, 9
 - getLD, 14
 - poly4root, 23
 - scoreTDT, 29
 - tdt, 31
 - tdtGxG, 33
 - *Topic **datagen**
 - trio.sim, 42
 - *Topic **datasets**
 - trio.data, 38
 - *Topic **file**
 - read.pedfile, 27
 - *Topic **hplot**
 - plot.getLD, 19
 - plot.LDblocks, 20
 - plot.trioLR, 21
 - print.trioFS, 25
 - *Topic **htest**
 - allelicTDT, 2
 - colGxE, 3
 - colGxGPerms, 6
 - colTDTmaxTest, 8
 - colTDTsam, 9
 - scoreTDT, 29
 - tdt, 31
 - tdtGxG, 33
 - trio.permTest, 38
 - *Topic **manip**
 - getMatPseudo, 15
 - ped2geno, 18
 - removeSNPs, 28
 - trio.check, 35
 - trio.prepare, 40
 - *Topic **math**
 - poly4root, 23
 - *Topic **models**
 - allelicTDT, 2
 - colGxE, 3
 - colGxGPerms, 6
 - colTDTmaxTest, 8
 - colTDTsam, 9
 - tdt, 31
 - tdtGxG, 33
 - trio.permTest, 38
 - *Topic **multivariate**
 - findLDblocks, 12
 - trioFS, 44
 - trioLR, 46
 - *Topic **print**
 - print.colGxE, 24
 - print.trioFS, 25
 - print.trioLR, 26
 - *Topic **regression**
 - trioFS, 44
 - trioLR, 46
 - *Topic **tree**
 - trioFS, 44
 - trioLR, 46
 - *Topic **univar**
 - getLD, 14
 - *Topic **utilities**
 - lrControl, 16
 - print.colGxE, 24
- allelicTDT, 2
- colGxE, 3, 16, 24, 25, 31
- colGxEPerms (colGxE), 3
- colGxG, 7
- colGxG (tdtGxG), 33
- colGxGPerms, 6
- colMAFtrio (removeSNPs), 28

- colTDT, 3, 6, 11, 16, 31
- colTDT (tdt), 31
- colTDT2way, 16, 31
- colTDT2way (tdtGxG), 33
- colTDTebam (colTDTsam), 9
- colTDTinter2way (tdtGxG), 33
- colTDTmaxStat, 11
- colTDTmaxStat (colTDTmaxTest), 8
- colTDTmaxTest, 8
- colTDTsam, 9
- compPermTDT2way (colGxGPerms), 6

- denspr, 10, 11

- ebam, 11

- fastGxG (tdtGxG), 33
- fastGxGrec (tdtGxG), 33
- findLDblocks, 12, 15, 21, 38
- freq.hap (trio.data), 38

- getGxEstats (print.colGxE), 24
- getLD, 12, 13, 14, 20, 38
- getLDlarge, 12
- getLDlarge (getLD), 14
- getMatPseudo, 15
- gtdt.ebam (colTDTsam), 9
- gtdt.stat (colTDTsam), 9

- haplo.em, 40, 41

- image, 20

- LDdata (trio.data), 38
- lrControl, 16, 39, 45, 47

- mat.test (trio.data), 38

- orderSNPs (removeSNPs), 28

- ped2geno, 2, 4, 6–8, 10, 13–15, 18, 27–30, 32–35
- plot.getLD, 15, 19, 21
- plot.getLDlarge (plot.getLD), 19
- plot.LDblocks, 13, 20, 20
- plot.trioFS (print.trioFS), 25
- plot.trioLR, 21
- poly4root, 23
- poly4rootMat (poly4root), 23
- print.aTDT (allelicTDT), 2
- print.colGxE, 24
- print.colTDT (tdt), 31
- print.colTDTepi (tdtGxG), 33
- print.LDblock (findLDblocks), 12
- print.maxScoreTrio (scoreTDT), 29
- print.maxStatTrio (colTDTmaxTest), 8
- print.maxTestTrio (colTDTmaxTest), 8
- print.scoreGxE (scoreTDT), 29
- print.scoreTDT (scoreTDT), 29
- print.tdt (tdt), 31
- print.tdtEpi (tdtGxG), 33
- print.trioFS, 25, 46
- print.trioLR, 26

- read.pedfile, 13, 14, 27, 29
- removeSNPs, 28
- removeTrios (removeSNPs), 28

- sam, 11
- scoreGxE (scoreTDT), 29
- scoreGxG (scoreTDT), 29
- scoreMaxStat (scoreTDT), 29
- scoreTDT, 29
- simuBkMap (trio.data), 38
- splitBlocks (findLDblocks), 12
- step3way (trio.data), 38

- tdt, 9, 19, 31, 35, 38
- tdt2way, 19, 33, 38
- tdt2way (tdtGxG), 33
- tdtGxG, 33
- trio.check, 19, 32, 34, 35, 38, 40, 41, 48
- trio.data, 38
- trio.gen.err (trio.data), 38
- trio.gen1 (trio.data), 38
- trio.gen2 (trio.data), 38
- trio.ped.err (trio.data), 38
- trio.ped1 (trio.data), 38
- trio.ped2 (trio.data), 38
- trio.permTest, 38, 48
- trio.prepare, 13, 35–38, 40, 44–48
- trio.sim, 38, 42
- trioFS, 13, 16, 25, 26, 44
- trioLR, 13, 16, 22, 23, 26, 27, 39, 40, 46, 46

- uniroot, 34