Package 'GWASpoly'

April 11, 2021

Title Genome-wide Association Studies for Autopolyploids

Version 2.05

Author Jeffrey B. Endelman, Umesh R. Rosyara

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Description Designed for genome-wide association studies in autopolyploids.

License GPL-3

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

Imports rrBLUP, methods, ggplot2, tidyr, stats

Suggests parallel, knitr

VignetteBuilder knitr

Collate 'GWASpoly.data.R'
'GWASpoly.K.R'
'GWASpoly.R'
'GWASpoly.fitted.R'
'GWASpoly.thresh.R'
'Keff.R'
'design.score.R'
'fit.QTL.R'
'get.QTL.R'
'get_x.R'
'make.full.R'
'manhattan.plot.R'
'qq.plot.R'
'qvalue.R'
'read.GWASpoly.R'
'score.calc.R'
'set.K.R'
'set.params.R'
'set.threshold.R'
'write.GWASpoly.R'

R topics documented:

fit.QTL 2

get.QTL 3
GWASpoly 3
GWASpoly.data-class 4
GWASpoly.fitted-class 4
GWASpoly.K-class 5
GWASpoly.thresh-class 5
manhattan.plot 6
qq.plot 6
read.GWASpoly 7
set.K 8
set.params 8
set.threshold 9
write.GWASpoly 10

Index **11**

`fit.QTL` *Test markers as QTL under backward elimination*

Description

Test markers as QTL under backward elimination

Usage

```
fit.QTL(data, trait, qtl, fixed = NULL)
```

Arguments

`data`	variable inheriting from class `GWASpoly.K`
`trait`	name of trait
`qtl`	list of markers to fit in the multi-QTL model (see Details)
`fixed`	list to specify fixed effects (see Details)

Details

Each element of `qtl` is a character vector of length two with format c("marker","model"). Each element of `fixed` is a character vector of length two: the first element is the name of the effect (must match column in phenotype input file) and the second element is either "factor" or "numeric". The p-value and R2 for each maker are based on the likelihood ratio test under backward elimination, comparing the deviance to the chi-squared distribution.

Value

data frame with partial r2 and p-values

get.QTL *Extract significant QTL*

Description

Output a table with all significant markers

Usage

```
get.QTL(data, traits = NULL, models = NULL)
```

Arguments

`data`	Output from `set.threshold`
`traits`	Vector of trait names (by default, all traits)
`models`	Vector of model names (by default, all models)

Details

Score = -log10(p). Effect = marker effect (not available for the general and diplo-general models).

Value

Data frame with results for significant markers

GWASpoly *Compute marker significance scores*

Description

Compute marker significance scores

Usage

```
GWASpoly(data, models, traits = NULL, params = NULL, n.core = 1, quiet = F)
```

Arguments

`data`	Output from `set.K`
`models`	Vector of model names
`traits`	Vector trait names (by default, all traits)
`params`	Optional list of params created by `set.params`
`n.core`	Number of cores for parallel computing
`quiet`	TRUE/FALSE whether to suppress output charting progress

Details

The following marker-effect models are available:

- "additive": Indicates the marker effect is proportional to the dosage of the alternate allele
- "X-dom": where X can be any integer between 1 and ploidy/2 and refers to the allele dosage needed for complete dominance (e.g., "1-dom" = simplex dominance, "2-dom" = duplex dominance). The software tries both dominance patterns for a given dosage model, e.g., whether the reference or alternate allele is dominant
- "diplo-general": All heterozygotes have the same effect
- "diplo-additive": All heterozygotes have the same effect, constrained to be halfway between the homozygous effects
- "general": There are no constraints on the effects of the different dosage levels

To specify additional model parameters, such as the inclusion of fixed effects (Q matrix) and the minimum minor allele frequency, use `set.params`

Value

Variable of class `GWASpoly.fitted`

`GWASpoly.data-class` *S4 class with genotype and phenotype data*

Description

S4 class with genotype and phenotype data

Slots

`map` data frame with marker,chrom,and position (either bp or cM)

`pheno` data frame of phenotypes

`geno` matrix (individuals x markers) of allele dosages (0,1,2,...ploidy)

`fixed` data frame of fixed effects

`ploidy` ploidy

`GWASpoly.fitted-class` *S4 class with results from genome-wide scan*

Description

S4 class with results from genome-wide scan

Slots

`map` data frame with marker,chrom,and position (either bp or cM)

`pheno` data frame of phenotypes

`geno` matrix with allele dosages

`fixed` data frame of fixed effects

`ploidy` ploidy

`K` covariance matrix for polygenic effect

`scores` -log10(p) results

`effects` estimated marker effects

`params` parameters used for the analysis

`GWASpoly.K-class` *S4 class with genotypes, phenotypes, and polygenic covariance*

Description

S4 class with genotypes, phenotypes, and polygenic covariance

Slots

`map` data frame with marker,chrom,and position (either bp or cM)

`pheno` data frame of phenotypes

`geno` matrix with allele dosages

`fixed` data frame of fixed effects

`ploidy` ploidy

`K` covariance matrix for polygenic effect

`GWASpoly.thresh-class` *S4 class with results from genome-wide scan and detection threshold*

Description

S4 class with results from genome-wide scan and detection threshold

Slots

`map` data frame with marker,chrom,and position (either bp or cM)

`pheno` data frame of phenotypes

`geno` matrix with allele dosages

`fixed` data frame of fixed effects

`ploidy` ploidy

`K` covariance matrix for polygenic effect

`scores` -log10(p) results

`effects` estimated marker effects

`params` parameters used for the analysis

`threshold` thresholds for significance

manhattan.plot *Create Manhattan plot*

Description

Create Manhattan plot

Usage

```
manhattan.plot(data, traits = NULL, models = NULL)
```

Arguments

`data`	Variable of class `GWASpoly.fitted`
`traits`	Vector of trait names (by default, all traits plotted)
`models`	Vector of model names (by default, all models plotted)

Details

Results for the ref and alt versions of the dominance model are combined. If `data` is the output from `set.threshold`, then the threshold is displayed as a horizontal dashed line when `models` contains a single model. Because the threshold varies between models, it is not drawn when multiple models are included. Although the ref and alt versions of each dominance model are slightly different (as seen with `qq.plot`), they are treated as a single model for the Manhattan plot, and the average threshold is shown.

Value

ggplot2 object

qq.plot *Quantile-Quantile (QQ) Plot*

Description

Inspect p-value inflation using a QQ plot

Usage

```
qq.plot(data, traits = NULL, models = NULL)
```

Arguments

`data`	Variable of class `GWASpoly.fitted`
`traits`	Vector of trait names (by default, all traits plotted)
`models`	Vector of model names (by default, all models plotted)

Value

ggplot2 object

read.GWASpoly *Read in marker and phenotype data*

Description

Read in marker and phenotype data

Usage

```
read.GWASpoly(ploidy, pheno.file, geno.file, format, n.traits, delim = ",")
```

Arguments

`ploidy`	Ploidy (e.g., 2 for diploid, 4 for tetraploid)
`pheno.file`	Name of the phenotype file
`geno.file`	Name of the genotype file
`format`	Format for the marker data. See details.
`n.traits`	Number of traits
`delim`	Character to indicate the delimiter in the data files (e.g., "," for csv, "\t" for tab-delimited)

Details

The first column of the phenotype file contains the genotype identifier, columns 2 through (n.traits + 1) contain trait values, and subsequent columns contain the levels (for factors) or numeric values (for covariates) of any fixed effects. The first three columns of the genotype file are (1) marker name, (2) chromosome, and (3) position. Subsequent columns contain the marker data for each individual in the population. Marker data can be coded in one of three formats:

- "numeric": markers are coded based on the dosage of the alternate allele, taking on values between 0 and ploidy
- "AB": e.g., AAAB, ABBB for tetraploids
- "ACGT": e.g., AAAT, GGCC for tetraploids

Only bi-allelic markers are allowed. As of version 2.02 of the package, fractional values of dosage are allowed for the "numeric" format, with missing values imputed by the population mean for each marker. The fractional values are only used for the additive genetic model; for the other models, dosages are rounded to the nearest whole number. If the input allele dosages are whole numbers, then missing values are imputed with the population mode (most frequent value) for each marker.

Value

Variable of class `GWASpoly.data`

set.K *Set covariance matrix for polygenic effect*

Description

Set covariance matrix for polygenic effect

Usage

```
set.K(data, K = NULL)
```

Arguments

data	Output from `read.GWASpoly`
K	Optional: user-supplied matrix

Details

By default, K is computed as $K=MM'$, where M is the centered genotype matrix (lines x markers). For GWAS, the overall scaling of K is irrelevant. At present, K is scaled such that the mean of its diagonal elements is 1. Alternatively, the user can supply any positive semidefinite K (with row.names that match the genotype identifiers).

Value

Variable of class `GWASpoly.K`

set.params *Set parameters*

Description

Set parameters

Usage

```
set.params(
  fixed = NULL,
  fixed.type = NULL,
  n.PC = 0,
  MAF = 0.05,
  geno.freq = 0.95,
  P3D = T
)
```

Arguments

`fixed`	Vector of names of fixed effects
`fixed.type`	Vector of effect types ("numeric" or "factor"), corresponding to the effects listed in "fixed"
`n.PC`	Number of principal components to include as covariates
`MAF`	Minimum minor allele frequency
`geno.freq`	Maximum genotype frequency (after applying dominance relations)
`P3D`	TRUE/FALSE whether to use the P3D approximation (variance components not re-estimated for every marker)

Details

The list returned by the function should be passed to `GWASpoly` function.

Value

A list with the following components

`fixed`	Names of fixed effects
`fixed.type`	Types of fixed effects
`n.PC`	Number of principal components to include as covariates
`min.MAF`	Minimum minor allele frequency
`max.geno.freq`	Maximum genotype frequency (after applying dominance relations)
`P3D`	TRUE/FALSE whether to use the P3D approximation

`set.threshold` *Set the significance threshold*

Description

Set the significance threshold

Usage

```
set.threshold(
  data,
  method = "M.eff",
  level = 0.05,
  n.permute = 1000,
  n.core = 1
)
```

Arguments

`data`	Variable of class `GWASpoly.fitted`
`method`	One of the following: "M.eff","Bonferroni","FDR","permute"
`level`	Genome-wide false positive or false discovery rate (depending on `method`).
`n.permute`	Number of permutations for method "permute"
`n.core`	Number of cores to use for multicore processing

Details

The default method, "M.eff", is a Bonferroni-type correction but using an effective number of markers that accounts for LD between markers (Moskvina and Schmidt, 2008). The FDR method is based on version 1.30.0 of the qvalue package.

Value

Variable of class `GWASpoly.thresh`

References

Moskvina V, Schmidt KM (2008) On multiple-testing correction in genome-wide association studies. Genetic Epidemiology 32:567-573. doi:10.1002/gepi.20331

`write.GWASpoly` *Write results to file*

Description

Write results to file

Usage

```
write.GWASpoly(data, trait, filename, what = "scores", delim = ",")
```

Arguments

`data`	Variable of class `GWASpoly.fitted`
`trait`	Trait name
`filename`	Filename
`what`	Either "scores" or "effects"
`delim`	Delimiter to use in the output file (default is comma)

Details

Score = -log10(p). Effect = marker effect (not available for the general and diplo-general models).

Index

fit.QTL, 2

get.QTL, 3
GWASpoly, 3
GWASpoly.data *(GWASpoly.data-class)*, 4
GWASpoly.data-class, 4
GWASpoly.fitted
    (GWASpoly.fitted-class), 4
GWASpoly.fitted-class, 4
GWASpoly.K, *2*
GWASpoly.K *(GWASpoly.K-class)*, 5
GWASpoly.K-class, 5
GWASpoly.thresh
    (GWASpoly.thresh-class), 5
GWASpoly.thresh-class, 5

manhattan.plot, 6

qq.plot, *6*, 6

read.GWASpoly, 7

set.K, 8
set.params, 8
set.threshold, *6*, 9

write.GWASpoly, 10