Package 'polyBreedR'

March 14, 2021

Title Genomics-assisted breeding for polyploids (and diploids)

Version 0.22

Author Jeffrey B. Endelman

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Description Genomics-assisted breeding for polyploids (and diploids)

Depends R (>= 4.0)

License GPL-3

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

Imports AGHmatrix, ggplot2, ggrepel, pedigree, grDevices, utils, tidyr, Matrix, methods

Suggests knitr, rmarkdown, asreml

VignetteBuilder knitr

R topics documented:

A_mat *Additive relationship matrix from pedigree*

Description

Additive relationship matrix from pedigree

Usage

```
A_mat(ped, ploidy, order.ped = TRUE)
```

Arguments

ped	Pedigree in three column format: id, mother, father
ploidy	2 or 4
order.ped	TRUE/FALSE does the pedigree need to be ordered so that progeny follow parents

Details

This is a wrapper that prepares the pedigree in the format required for R package AGHmatrix by Amadeu et al. (2016) (cite them if you use this function). A random bivalents model for tetraploid meiosis is assumed.

Value

Additive relationship matrix (dim: indiv x indiv)

References

Amadeu et al. (2016) Plant Genome 9, doi:10.3835/plantgenome2016.01.0009

check_ploidy *Check ploidy*

Description

Fraction of simplex or triplex markers

Usage

```
check_ploidy(geno, map)
```

Arguments

geno	Genotype matrix (markers x indiv)
map	Data frame with marker map (Marker, Chrom, Position)

Details

For every indiv in the genotype matrix, the fraction of markers per chromosome called as simplex or triplex is calculated, which should be low for diploids. A small amount of missing genotype data can be tolerated.

Value

List containing

mat Matrix (indiv x chrom) of results

plot ggplot2 barplot

`check_trio` *Check markers for parent-offspring trio*

Description

Check markers for parent-offspring trio

Usage

```
check_trio(parentage, geno, ploidy)
```

Arguments

`parentage`	Data frame with three columns: id, mother, father
`geno`	Matrix of allele dosages: markers x indiv
`ploidy`	2 or 4

Details

Computes the percentage of markers at which the two parents and offspring have incompatible allele dosages (for tetraploids, the random bivalents model is used). For dihaploid offspring of a single tetraploid parent, use `ploidy = 4` and "haploid" for the father in `parentage`, as well as a diploid (0,1,2) genotype for the offspring. A small amount of missing genotype data can be tolerated.

Value

Data frame with the percentage of incompatible markers for each trio

dart_tag *Extract Ref/Alt counts from DArTag data file*

Description

Extract Ref/Alt counts from DArTag data file

Usage

```
dart_tag(filename)
```

Arguments

filename	input filename

Details

Designed for standard two-row format from DArT. First 11 rows contain sample information. Column 1 contains the AlleleID in format MarkerName|Haplotype. Haplotypes are named Ref,RefMatch,Alt,AltMatch,Othe
Counts are combined for Ref + RefMatch, as well as Alt + AltMatch. Other haplotypes are discarded.

Value

3D array of allele counts with dimensions: markers, samples, alleles (ref/alt)

D_mat *Dominance genomic relationships*

Description

Coefficients and relationship matrix for digenic dominance effects with bi-allelic markers

Usage

```
D_mat(geno, ploidy)
```

Arguments

geno	Matrix of allele dosages: markers x indiv
ploidy	2 or 4

Details

Digenic dominance effects are based on the traditional orthogonal decomposition of genetic variance in panmictic populations (Fisher 1918; Kempthorne 1957; Endelman et al. 2018). The D matrix is computed from the coefficients and scaling factor according to D = tcrossprod(coeff/scale). Missing genotype data is replaced with the population mean.

Value

List containing

coeff Coefficients of the marker effects (dim: indiv x marker)

scale Scaling factor between markers and indiv

mat D matrix

References

Fisher (1918) Trans. Roy. Soc. Edin. 52:399-433.

Kempthorne (1957) An Introduction to Genetic Statistics.

Endelman et al. (2018) Genetics 209:77-87.

`geno_call` *Genotype calls*

Description

Genotype calls based on a normal mixture model

Usage

```
geno_call(
  data,
  filename,
  model.ploidy = 4,
  sample.ploidy = 4,
  min.posterior = 0,
  transform = TRUE
)
```

Arguments

`data`	matrix (markers x id) of input values for the normal mixture model
`filename`	CSV filename with the model parameters
`model.ploidy`	2 or 4 (default)
`sample.ploidy`	2 or 4 (default)
`min.posterior`	minimum posterior probability (default 0) for genotype call
`transform`	TRUE (default) or FALSE whether to apply arcsin square root transformation

Details

The first column of the CSV input file should be the SNP ID, followed by columns for the normal distribution means, standard deviations, and mixture probabilities. Genotype calls are based on the maximum a posteriori (MAP) method. If the posterior probability of the MAP genotype is less than `min.posterior`, then `NA` is returned for that sample. By default, an arcsin square root transformation is applied to the input values to match the approach used by R package fitPoly. To use a tetraploid mixture model for diploid samples, set `sample.ploidy = 2` and `model.ploidy = 4`.

Value

matrix of allele dosages (0,1,2,..ploidy) with dimensions markers x individuals

---

`get_pedigree` *Generate pedigree*

Description

Generate pedigree for a set of individuals

Usage

```
get_pedigree(id, pedfile, delim = ",", na.string = "NA", trim = TRUE)
```

Arguments

`id`	Vector of names of individuals
`pedfile`	Name of pedigree file
`delim`	Delimiter for the pedigree file (default is "," for CSV)
`na.string`	String used for NA in the pedigree file (default is "NA")
`trim`	TRUE/FALSE whether to trim pedigree (see Details)

Details

Finds ancestors of individuals in a three-column pedigree file (id,mother,father). The id column can be the identifier for an individual or cross. String matches must be exact or based on the naming convention crossID-progenyID. The returned pedigree is ordered using R package `pedigree` so that offspring follow parents. When `trim` is TRUE (default), the pedigree is trimmed to remove ancestors with only one offspring (which are not needed to compute the pedigree relationship matrix).

Value

Data frame with columns id, mother, father

`GvsA` *Plot G vs. A*

Description

Plot marker-based vs. pedigree-based additive relationship coefficients

Usage

```
GvsA(
  parentage,
  G,
  A,
  filename = NULL,
  thresh.G = Inf,
  thresh.A = 0.5,
  Gmax = NULL,
  Amax = NULL
)
```

Arguments

parentage	Data frame of individuals to plot, with 3 columns: id,mother,father
G	Genomic relationship matrix
A	Pedigree relationship matrix
filename	Name of PDF file to save the results (optional for one individual)
thresh.G	Threshold above which names are displayed (default Inf)
thresh.A	Threshold above which names are displayed (default 0.5)
Gmax	Upper limit for y-axis for plotting. If NULL, maximum value in G is used.
Amax	Upper limit for x-axis for plotting. If NULL, maximum value in A is used.

Details

Useful for finding and correcting pedigree errors. If the G or A coefficient for an individual exceeds the threshold, its name is displayed in the figure. If parentage contains one individual, by default a ggplot2 variable will be returned, but the result can also be written to file. If multiple individuals are present, a filename is required.

G_mat *Additive genomic relationships*

Description

Coefficients and relationship matrix for additive effects with bi-allelic markers

Usage

```
G_mat(geno, ploidy)
```

Arguments

geno	Matrix of allele dosages (markers x indiv)
ploidy	2 or 4

Details

Additive effects are based on the traditional orthogonal decomposition of genetic variance in panmictic populations (Fisher 1918; Kempthorne 1957; Endelman et al. 2018). The G matrix is computed from the coefficients and scaling factor according to G = tcrossprod(coeff/scale). Missing genotype data is replaced with the population mean.

Value

List containing

coeff Coefficients of the marker effects (dim: indiv x marker)

scale Scaling factor between markers and indiv

mat G matrix

References

Fisher (1918) Trans. Roy. Soc. Edin. 52:399-433.

Kempthorne (1957) An Introduction to Genetic Statistics.

Endelman et al. (2018) Genetics 209:77-87.

---

impute *Impute missing marker data*

Description

Impute marker data based on the population mean or mode

Usage

```
impute(geno, method)
```

Arguments

geno	Matrix of allele dosages with dimensions markers x indiv
method	Either "mean" or "mode"

Details

Missing values are imputed with either the population mean or mode (most frequent value) for each marker

Value

Imputed genotype matrix (markers x indiv)

merge_impute *Merge two genotype matrices and impute missing data*

Description

Merge two genotype matrices and impute missing data by BLUP

Usage

```
merge_impute(geno1, geno2, ploidy)
```

Arguments

geno1	Genotype matrix (coded 0...ploidy) with dimensions markers x indiv
geno2	Genotype matrix (coded 0...ploidy) with dimensions markers x indiv
ploidy	Either 2 or 4

Details

Designed to impute from low to high density markers. The BLUP method is equivalent to Eq. 4 of Poland et al. (2012), but this function is not iterative. Additional shrinkage toward the mean is applied if needed to keep the imputed values within the range [0,ploidy]. Missing data in the input matrices are imputed with the population mean for each marker. If an individual appears in both input matrices, it is renamed with suffixes ".1" and ".2" and treated as two different individuals. Monomorphic markers are removed.

Value

Imputed genotype matrix (markers x indiv)

References

Poland et al. (2012) Plant Genome 5:103-113.

MME-class *S4 class for solving the mixed model equations*

Description

S4 class for solving the mixed model equations

Slots

`data` data frame with id, env, blue, trait (optional)

`kernels` list of variance-covariance matrices for the genetic effects

`Rmat` residual variance-covariance matrix

predict_MME *Compute BLUPs by solving the Mixed Model Equations*

Description

Compute BLUPs by solving the Mixed Model Equations

Usage

```
predict_MME(data, weights = NULL, mask = NULL)
```

Arguments

data	variable of class MME
weights	named vector of weights for the genetic effects in BLUP. Default is 1 for all effects.
mask	(optional) data frame with column "id" and optional columns "env", "trait"

Details

Use the function Stage2 to create the object of class MME. BLUPs are computed at the average value of the fixed effects. If weights is used, the names must exactly match the names of the kernels in data. Using the argument mask, the phenotypes for a subset of the population can be masked, to enable cross-validation.

Value

For single trait analysis, function returns a data frame with columns: id,blup,r2. For multi-trait analysis, a list is returned containing

blup data frame of blups

r2 data frame of reliabilities

readXY *Read SNP array intensity data*

Description

Read SNP array intensity data

Usage

```
readXY(filename, skip, output = "ratio")
```

Arguments

filename	filename
skip	number of lines to skip before the header line with the column names
output	Either "ratio" or "theta"

Details

The first two columns of the tab-delimited input file should be the SNP and Sample ID. Columns labeled "X" and "Y" contain the signal intensities for the two alleles. Use `output` to specify whether to return the ratio = Y/(X+Y) or theta = atan(Y/X)*2/pi.

Value

matrix with dimensions markers x individuals

`Stage1` *Stage 1 analysis of multi-environment trials*

Description

Stage 1 analysis of multi-environment trials

Usage

```
Stage1(
  data,
  traits,
  effects = NULL,
  silent = TRUE,
  workspace = "500mb",
  pworkspace = "500mb"
)
```

Arguments

`data`	data frame with phenotype data
`traits`	vector of column names from `data`
`effects`	list of other effects in the model
`silent`	TRUE/FALSE, whether to suppress ASReml-R output
`workspace`	memory limit for ASRreml-R variance estimation
`pworkspace`	memory limit for ASRreml-R BLUE computation

Details

Stage 1 of the two-stage approach described by Damesa et al. 2017, using ASReml-R for variance component estimation (license is required). The variable `data` must have one column labeled "id" for the individuals, one labeled "env" for the environments, plus columns for each of the traits to be analyzed. The data for each environment x trait combination are analyzed independently with a linear mixed model. Argument `effects` is a named list of character vectors to specify other effects in the model. Each vector has two elements: the first is "fixed" or "random", and the second is "factor" or "numeric". For example, to include a random block effect, use `effects=list(block=c("random","factor"))`. To include stand.count as a numeric covariate, use `effects=list(stand.count=c("fixed","numeric"))`. By default, the workspace and pworkspace limits for ASReml-R are set at 500mb. If you get an error about insufficient memory, try increasing the appropriate value (workspace for variance estimation and pworkspace for BLUE computation).

Value

List containing

H2 matrix of broad-sense heritability for each env x trait combination

blue data frame of BLUEs for id x traits

blue.vcov list of BLUE + variance-covariance matrices (one matrix per trait)

References

Damesa et al. 2017. Agronomy Journal 109: 845-857. doi:10.2134/agronj2016.07.0395

`Stage2` *Stage 2 analysis of multi-environment trials*

Description

Stage 2 analysis of multi-environment trials

Usage

```
Stage2(data, kernels = NULL, silent = TRUE, workspace = "500mb")
```

Arguments

`data`	data frame of BLUEs from Stage 1 (see Details)
`kernels`	vector of variable names for variance-covariance matrices of the genetic effects (see Details)
`silent`	TRUE/FALSE, whether to suppress ASReml-R output
`workspace`	Memory limit for ASRreml-R variance estimation

Details

Stage 2 of the two-stage approach described by Damesa et al. 2017, using ASReml-R for variance component estimation (license is required). The variable `data` must contain at least three columns: id, env, blue. id is the individual identifier, and env represents the environment at which Stage 1 analysis was performed. blue is the BLUE from Stage 1 (NAs are not allowed). Two other column names are reserved: trait and loc. The former triggers a multivariate, multi-trait analysis. The latter triggers the inclusion of a random genotype x location effect (to be completed). To model the uncertainty in the BLUEs from Stage 1 in Stage 2, an additional random effect is included with a variance-covariance matrix named Omega (following notation from Damesa et al. 2017). This variable must be defined globally instead of passing it to the function. The function `Stage2_prep` can be used to prepare both `data` and Omega. By default, the model includes independent random effects for genotype (id). Additional genetic effects with specific covariance structure (such as the G matrix for genomic breeding values) can be included using the argument `kernels`, which is a vector of variable names (for example, "G") defined in the global environment. (Do not use the name "I" for a kernel; it is reserved for the independent genetic effect.) All individuals in `data` must be present in the kernel matrices, but the kernels can contain individuals not in `data` to make predictions for unphenotyped individuals using `predict_MME`. All kernel matrices must have the same rownames attribute. For numerical stability when inverting the kernel matrices, a small positive number (1e-5) is added to the diagonal elements. By default, the workspace memory

for ASReml-R is set at 500mb. If you get an error about insufficient memory, try increasing it. ASReml-R version 4.1.0.148 or later is required. For kernel matrix K, the variance reported in `vars` equal the variance component times the mean of the diagonal elements of ZKZ', which allows for easy computation of the proportion of variance.

Value

List containing

aic AIC

vars variances

trait.cov genetic variance-covariance matrix for the traits (for multi-trait analysis)

MME variable of class `MME` for use with `predict_MME`

References

Damesa et al. 2017. Agronomy Journal 109: 845-857. doi:10.2134/agronj2016.07.0395

`Stage2_prep` *Prepare data for Stage 2 analysis of multi-environment trials*

Description

Prepare data for Stage 2 analysis of multi-environment trials

Usage

```
Stage2_prep(blue.vcov, exclude.id = character(0), exclude.env = character(0))
```

Arguments

`blue.vcov`	named list of blue.vcov matrices from Stage 1
`exclude.id`	vector of individuals to exclude
`exclude.env`	vector of envs to exclude

Details

Designed to prepare data files for `Stage2` based on output from `Stage1`. Each element of `blue.vcov` is a matrix for one trait, with the first column containing BLUEs and the remainder is their variance-covariance. The rownames are the id and env concatenated with ":". The output `blue` is a data frame with id, env, and trait (when multiple traits are provided). This allows for multi-trait analysis in Stage 2.

Value

a list containing

blue data frame of BLUEs

Omega variance-covariance matrix of BLUEs

update_alias *Update names based on alias*

Description

Update names based on data frame with alias and preferred name

Usage

```
update_alias(x, alias, remove.space = TRUE)
```

Arguments

x	Vector of names to update
alias	Data frame with two columns: first is the preferred name and second is the alias
remove.space	TRUE/FALSE

Details

Parameter remove.space indicates whether blank spaces should be removed before string matching

Value

Vector with updated names

Index