Package ‘polyBreedR’

December 30, 2020

Title Using genome-wide markers for polyploid breeding

Version 0.14

Author Jeffrey B. Endelman

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Description Using genome-wide markers for polyploid breeding

Depends R (>= 3.5.0)

License GPL-3

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

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

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

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.

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,
  fixed = NULL,
  random = NULL,
  silent = FALSE,
  workspace = "500mb",
  pworkspace = "500mb"
)
```

Arguments

`data`	Data frame with phenotype data
`traits`	Vector of column names from `data`
`fixed`	Vector of column names from `data`
`random`	Vector of column names from `data`
`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 a column labeled "id" with the names of the different genotypes (i.e., clones or individuals). To include other variables (besides "id") in the model, include them in `fixed` or `random` as appropriate, and make sure they have the correct type in the data frame: factor vs. numeric. If multiple traits are included, a multivariate analysis is performed, and only plots with data for all traits are included. The `h2` matrix returned by the function contains the estimated genetic correlations above the diagonal, residual correlations below the diagonal, and plot-based heritability on the diagonal. For multivariate analysis, the data frame `blue` returned by the function is in long format, with a column named "trait". 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

aic AIC from ASReml-R

blue data frame of BLUEs

vcov variance-covariance matrix of the BLUEs

h2 matrix with heritability, genetic, and residual correlations (see Details)

References

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

`Stage2` *Stage 2 analysis of multi-environment trials (still under development)*

Description

Stage 2 analysis of multi-environment trials

Usage

```
Stage2(
  data,
  fixed = NULL,
  silent = FALSE,
  workspace = "500mb",
  pworkspace = "500mb"
)
```

Arguments

`data`	Data frame with BLUEs from Stage 1 (see Details)
`fixed`	Additional fixed effects, as a character vector
`silent`	TRUE/FALSE, whether to suppress ASReml-R output
`workspace`	Memory limit for ASRreml-R variance estimation
`pworkspace`	Memory limit for ASRreml-R BLUP computation

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: env, id, blue. The first column (env) is the environment identifier, which in plant breeding typically represents a location x year combination. The second column (id) is the genotype identifier, and the third column (blue) is the BLUE from Stage 1 (NAs are not allowed). There are two other reserved column names, which are optional: expt, loc. By default, a fixed effect for each environment is included, but there are situations where BLUEs from multiple experiments (expt) in one environment are included, in which case "expt" overrides "env" to specify the fixed effect portion of the model. When the population of environments includes multiple locations with more than one environment per location, "loc" leads to the inclusion of random effects for genotype x location. For more than 3 locations, a first-order factor-analytic model is used to reduce model complexity. Additional fixed effects can be specified using ASReml-R syntax with the argument `fixed` (make sure they have the correct type in `data`: numeric vs. factor). To model the uncertainty in the BLUEs from Stage 1 in Stage 2, an additional random effect is included with a constrained variance-covariance matrix named `Omega` (following the notation of Damesa et al. 2017). Due to limitations with ASReml-R, 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`. The main effect for genotype can also be partitioned into additive and non-additive effects by defining a global variable named `G` for the G matrix. If there are individuals in `data` but not in `G`, an error is returned. To make predictions for unphenotyped individuals, include them in `G`. 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 BLUP computation).

Value

List containing

aic AIC

fixed Fixed effect estimates and SE

vc Variance component estimates and SE

blup BLUPs and reliability (r2) for breeding values if G present or genotypic values if G not present

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(data, id = NULL)
```

Arguments

`data`	Named list containing output from Stage 1 (see Details)
`id`	Vector of genotype identifiers to include (default is all)

Details

Designed to prepare data files for `Stage2` based on output from `Stage1`. Each element of `data` is a list that contains at least two variables: "blue" and "vcov". The "blue" variable is a data frame with columns named "id" and "blue", and if multiple traits have been analyzed in Stage 1, there can be a third column named "trait". The "vcov" variable is the variance-covariance matrix of the BLUEs. By default, the function treats each element of `data` as a different environment, which in plant breeding typically represents a location x year combination. If data from multiple experiments per environment are included, each element of `data` should also contain the variable "env" to specify the environment name. Furthermore, when the dataset includes multiple locations with more than one environment per location, include "loc" for each element of `data` to model genotype x location effects.

Value

A list containing

blue data frame of BLUEs

Omega variance-covariance matrix of BLUEs

For multiple traits, the Omega variable is a list of matrices, one for each trait.

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