Package 'wgaim'

October 12, 2022

Type Package

Title Whole Genome Average Interval Mapping for QTL Detection and
Estimation using ASReml-R

Version 2.0-1

Date 2019-10-03

Author
Julian Taylor <julian.taylor@adelaide.edu.au> and Ari Verbyla <ari.verbyla@csiro.au>

Maintainer Julian Taylor <julian.taylor@adelaide.edu.au>

Depends R (>= 3.5.0), qtl

Imports ggplot2

Suggests knitr, rmarkdown, prettydoc

VignetteBuilder knitr

SystemRequirements asreml-R 4.x

Description A computationally efficient whole genome approach to detecting and estimating significant QTL in linkage maps using the flexible linear mixed modelling functionality of ASReml-R.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-03 05:10:02 UTC

R topics documented:

wgaim-package	*Whole Genome Average Interval Mapping (wgaim) for QTL detection and estimation*

Description

This package provides an efficient computational implementation of the QTL whole genome QTL analysis algorithm (wgaim) discussed in Verbyla et. al (2007, 2012) using extensions of the functionality provided in the linear mixed modelling R package **ASReml-R** V4.

Details

Package:	wgaim
Type:	Package
Version:	2.0-0
Date:	2019-08-12
License:	GPL 2

Welcome to version 2.0 of **wgaim**! The documentation given in this help file is only brief and users should consult the vignette available with the package by typing vignette("wgaim") at the prompt. Alternatively, users can also consult the individual help files for the main functions of the package.

The package provides a user friendly function [cross2int](#) for the conversion of "cross" objects created using read.cross in Bromans **qtl** package into an "interval" object ready for use in **wgaim**. Specifically, cross2int() performs additional calculations to impute missing marker values on each of the chromosomes across the full linkage map and also provides users with genetic distances and recombination fractions for the intervals. The returned object retains the class structure of an object created with read.cross and therefore allows further use with the **qtl** package if desired.

The package also provides a function for the graphical display of the chromosomes of a "cross" object. The method function [linkMap](#) displays the full or subsetted linkage map according to chromosome or distance as well as displays non-overlapping marker names on the right hand side.

QTL analysis is conducted using the function [wgaim](#) which, as its first argument, requires an asreml base model. High dimensional genetic components are allowed (See wgaim.asreml for more details). For convenience the default tracing of results from the asreml models can be outputted to a

file for further inspection. Outlier statistics and marker effects from each iteration can be viewed using `outStat`. Diagnostics of the likelihood ratio test performed for each forward step can be displayed using `tr.wgaim`. The function also displays an incremental probability value matrix of the QTL ascertained at each forward step of the algorithm.

Summary and print methods are available for the returned "wgaim" object and provide users with a detailed report on the QTL, their size, their flanking markers, significance (including LOD score if desired) and approximate contribution to the genetic variance. The returned "wgaim" object may also be plotted using the method function `linkMap`. This function plots the full linkage map subsetted for chromosome and distance as well as provides shaded QTL regions and highlighted flanking markers. Plotting of QTL for multiple traits is also possible (see `linkMap.default`).

Author(s)

Julian Taylor and Ari Verbyla Maintainer: Julian Taylor <julian.taylor@adelaide.edu.au>

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical And Applied Genetics*, **116**, 95-111.

See Also

`qtl-package`

---

cross2int	*Convert a cross genetic object to an interval object*

Description

Converts an object of class "cross" to an object with class "interval". The function also imputes missing markers.

Usage

```
cross2int(object, impute = "MartinezCurnow", consensus.mark = TRUE,
    id = "id", subset = NULL)
```

Arguments

object	an object of class `"cross"` that inherits one of the class structures `"bc"`, `"dh"`, `"f2"`, `"riself"`.
impute	a character string determining how missing values in the linkage map should be imputed. If `"Broman"`, then missing values are imputed according to Bromans rules. If `"MartinezCurnow"` then missing values are imputed according to the rules of Martinez & Curnow (1994) (see reference list). The default is `"MartinezCurnow"` (see Details).
consensus.mark	logical value. If `TRUE` co-locating marker sets are condensed to form consensus markers (see Details). Defaults to `TRUE`.
id	a character string or name of the unique identifier for each row of genotype data (see Details). Defaults to `"id"`
subset	a possible character vector naming the subset of chromosomes to be returned. Defaults to `NULL` implying return all chromosomes.

Details

This function provides the conversion of genetic data objects that have already been generated using `read.cross()` from Bromans **qtl** package, to `"interval"` objects ready for use with wgaim. Users should be aware that this function is restricted to certain populations. `object` must inherit one of the class structures `"bc"`, `"dh"`, `"f2"`, `"riself"`.

During the conversion process three important linkage map attributes are assessed.

1. The map may be subsetted using the subset argument

2. If `consensus.mark = TRUE` then co-located marker sets are reduced to form single consensus markers before missing values are imputed. The marker similarity is determined by the genetic distances that are given in the map component for each linkage group. If a set of markers co-locate the name of the first marker is chosen and a single consensus marker is determined by coalescing the genetic information from all markers in the set. A "(C)" is placed after the marker name for easy identification. The markers removed from each set are returned with the object and placed under `"colocated.markers"` for inspection if required.

3. Missing values are imputed according to the argument given by impute. This imputation results in a complete version of the marker data for each chromosome which is then used to create the interval data component `"interval.data"`. The complete marker data for each chromosome can be obtained from the `"imputed.data"` element of the returned list. It is therefore also possible to perform whole genome *marker* analysis using wgaim. See wgaim.asreml for more details.

Value

a list of class `"cross"` that also inherits the class `"interval"`. The list contains the following components

geno	A list with elements named by the corresponding names of the chromosomes. Each chromosome is itself a list with six elements: `"data"` is the actual estimated map matrix with rows as individuals named by `"id"` and markers as

columns; ″map″ is a vector of marker positions on the corresponding chromosome; ″imputed.data″ is identical to ″data″ matrix but with all NAs replaced by imputed values according to the rules of ″impute″; ″dist″ contains the genetic distance between adjacent markers or the genetic distances of the intervals; ″theta″ contains the recombination fractions for each interval; ″interval.data″ contains the recalculated intervals based on the recombination fractions and the missing marker information.

colocated.markers

If consensus.mark = TRUE, a four column data frame containing stacked binned sets of co-located markers. In each binned set the first marker is the unique consensus marker name used in the linkage map and the others are the co-located marker names that were omitted. Additionally for each binned set, the data frame also contains linkage group identification and marker position information.

pheno
A data.frame of phenotypic information with rows as individuals read in from read.cross. A copy of the column named by the ″id″ argument can be found here (see the help for the read.cross() function).

Author(s)

Julian Taylor and Ari Verblya

References

Martinez, O., Curnow. R. N. (1994) Missing markers when estimating quantitative trait loci using regression mapping. *Heredity*, **73**, 198-206.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical And Applied Genetics*, **116**, 95-111.

See Also

read.cross

Examples

```
## Not run:
# read in linkage map from a rotated .CSV file with "id" as the
# identifier for each unique row

wgpath <- system.file("extdata", package = "wgaim")
genoSxT <- read.cross("csvr", file="genoSxT.csv", genotypes=c("AA","BB"),
          na.strings = c("-", "NA"), dir = wgpath)
genoSxT <- cross2int(genoSxT, impute="MartinezCurnow", id = "id")

# plot linkage map
```

```
linkMap(genoSxT, cex = 0.5)


## End(Not run)
```

---

genoCxR   *Genotypic marker data for Cascades x RAC875-2 doubled haploid*
                                  population in R/qtl format

Description

Linkage map marker data for the Cascades x RAC875-2 doubled haploid population in the form of
an R/qtl cross object.

Usage

```
data(genoCxR)
```

Format

This data relates to a linkage map of 663 markers genotyped on 93 individuals. The linkage
map consists of 42 linkage groups spanning the whole genome. Coincident markers have been
removed reducing the linkage map to 458 markers. Map distances have been estimated using
`read.cross()` with the haldane mapping function. The data object is therefore an R/qtl cross
object. See `read.cross()` documentation for more details on the format of this object.

Examples

```
data(genoCxR, package = "wgaim")
linkMap(genoCxR, cex = 0.5)
```

---

genoRxK   *Genotypic marker data for RAC875 x Kukri doubled haploid popula-*
                                  tion in R/qtl format

Description

Linkage map marker data for the RAC875 x Kukri doubled haploid population in the form of an
R/qtl cross object.

Usage

```
data(phenoSxT)
```

Format

This data relates to a linkage map of 500 genetic markers genotyped on 368 individuals from the RAC875 x Kukri population. The linkage map consists of 21 linkage groups with varying numbers of markers. Map distances have been estimated using `read.cross()` with the kosambi mapping function. The data is therefore an R/qtl cross object. See `read.cross()` documentation for more details on the format of this object.

Examples

```
data(genoRxK, package = "wgaim")
linkMap(genoRxK, cex = 0.5)
```

---

genoSxT   *Genotypic marker data for Sunco x Tasman doubled haploid population in R/qtl format*

Description

Linkage map marker data for the Sunco x Tasman doubled haploid population in the form of an R/qtl cross object.

Usage

```
data(phenoSxT)
```

Format

This data relates to a linkage map of 287 genetic markers genotyped on 190 individuals from the Sunco x Tasman population. This set is reduced from the original 345 markers (a mixture of AFLP, RFLP and microsatellite markers and protein analysis). The reduction was created by discarding 58 markers which were co-located with one or more other markers. The linkage map consists of 21 linkage groups with varying numbers of markers. Map distances have been estimated using `read.cross()` with the kosambi mapping function. The data is therefore an R/qtl cross object. See `read.cross()` documentation for more details on the format of this object.

Examples

```
data(genoSxT, package = "wgaim")
linkMap(genoSxT, cex = 0.5)
```

```
linkMap.cross                      Plot a genetic linkage map
```

Description

Neatly plots the genetic linkage map with marker locations and marker names.

Usage

```
## S3 method for class 'cross'
linkMap(object, chr, chr.dist, marker.names = "markers",
    tick = FALSE, squash = TRUE, m.cex = 0.6, ...)
```

Arguments

object	object of class `"cross"`
chr	character string naming the subset of chromosomes to plot
chr.dist	a list containing named elements `"start"` and `"end"` containing the start and end distances in cM the genetic map should be subsetted by. Each of these may also be a vector of distances equal to the length of the number of linkage groups to be plotted.
marker.names	a character string naming the type of marker information to plot. If `"dist"` then distances are plotted alongside each chromosome on the left. If `"markers"` then marker names are plotted instead. Defaults to `"markers"`
tick	logical value. If TRUE then an axis with tick marks are generated for the chromosome names. Defaults to FALSE
squash	logical value. if TRUE then creates extra room on the left side of the chromosomes. This is useful for plotting trait names for QTL using `linkMap.wgaim()` and `linkMap.default()`
m.cex	the expansion factor to use for the marker names
...	arguments passed to `plot()` function to set up the plot region. Arguments may also be passed to `text()` function for the manipulation of the marker names

Details

This plotting procedure provides a visual display of the chromosomes without marker names overlapping vertically. The plotting region will adjust itself to ensure that all marker names are in the region. For this reason the value for `"m.cex"` is passed to the `text()` function and should be manipulated until an aesthetic genetic map is reached.

For large maps with many chromosomes, marker names and adjacent chromosomes will overlap horizontally. For the interest of readability this has not been corrected. For this particular situation it is suggested that the user horizontally maximise the plotting window until no overlapping occurs or subset the genetic map to achieve the desired result.

Value

This invisibly returns the following list for manipulation with linkMap.wgaim()

mt	A list named by the chromosomes with each element containing the locations of the marker names after correcting for overlapping
map	A list named by the chromosomes with each element containing the locations of markers on the chromosomes
chrpos	The numerical position of the chromosomes on the plotting region

Author(s)

Julian Taylor

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical and Applied Genetics*, **116**, 95-111.

See Also

linkMap.wgaim

Examples

```
data(genoSxT, package = "wgaim")

## plot linkage map with marker names

linkMap(genoSxT, cex = 0.5)

## plot linkage map with distances

linkMap(genoSxT, cex = 0.5, marker.names = "dist")
```

---

linkMap.default   *Plot a genetic linkage map with QTL for multiple traits*

Description

Neatly plots the genetic linkage map with marker locations, marker names and highlights QTL with their associated flanking markers for multiple traits obtained from a list of `wgaim` models.

Usage

```
## Default S3 method:
linkMap(object, intervalObj, chr, chr.dist, marker.names
    = "markers", flanking = TRUE, list.col = list(q.col = rainbow(length(object)),
    m.col = "red", t.col = rainbow(length(object))), list.cex =
    list(m.cex = 0.6, t.cex = 0.6), trait.labels = NULL, tick = FALSE, ...)
```

Arguments

object	a list object with elements inheriting the class `"wgaim"`
intervalObj	object of class `"cross"` or `"interval"`
chr	character string naming the subset of chromosomes to plot
chr.dist	a list containing named elements `"start"` and `"end"` containing the start and end distances in cM the genetic map should be subsetted by. Each of these may also be a vector of distances equal to the length of the number of linkage groups to be plotted.
marker.names	a character string naming the type of marker information to plot. If `"dist"` then distances are plotted alongside each chromosome on the left. If `"markers"` then marker names are plotted instead. Defaults to `"markers"`.
flanking	logical value. If `TRUE` then only plot marker names or distances for flanking markers of the QTL. Defaults to `TRUE`
list.col	named list of colors used to highlight the QTL regions and their flanking markers. `q.col` is the colors of the QTL regions (defaults to `rainbow(n)` where `n` is the length of `object`). `m.col` is the color the flanking markers. `t.col` is the color of the trait names used in each model (defaults to the same color as the QTL regions). See `par` for color options
list.cex	a named list object containing the character expansion factors for the marker names `m.cex` and the trait labels `t.cex`
trait.labels	character string naming the trait used in the model object, defaults to the response names extracted from the fixed compoenent of the base model.
tick	logical value. If `TRUE` then an axis with tick marks are generated for the chromosome names
...	arguments passed to the `plot()` function to set up the plot region. Arguments may also be passed to the `text()` function for the manipulation of the marker names

Details

This plotting procedure is a wrapper for `linkMap.wgaim` and displays QTL for multiple traits obtained from a list of models given by `object`. Alternative labels for the traits can be given, in model order, using `trait.labels`.

Color specific highligting of the QTL is also available using `clist`. This differs slightly from `linkMap.wgaim`. Here the `q.col` and `t.col` should be given a set of colors equal to the length of `object`. Let n be the length of `object`. Then if `q.col` is NULL or length of `q.col` is not equal to n then it defaults to `rainbow(n)`. If *t.col* is NULL or length of `t.col` is not equal to n or 1 then it defaults to the colors of `q.col`. Examples of different color combinations are given below in the examples.

The `list.cex` argument can be used to manipulate the character expansion of the marker names using `m.cex` or the character expansion of the `trait.labels` using `t.cex`. If a set of `"marker"` analyses has been performed then `pch` is used to plot a symbol at the location of the QTL. This character can be changed using the usual arguments such as `pch` or `cex` that are passed through the ... argument.

Value

For a set of `"interval"` analyses, the genetic linkage map is plotted with shaded QTL regions and highlighted flanking markers. For a set of `"marker"` analyses, symbols are placed at the QTL locations and the markers are highlighted.

Author(s)

Julian Taylor

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical and Applied Genetics*, **116**, 95-111.

See Also

linkMap.cross, linkMap.wgaim

Examples

```
## Not run:
## fit wgaim models

rktgw.qtl <- wgaim(rktgw.asf, intervalObj = genoRxK, merge.by = "Genotype",
```

```
                   trace = "trace.txt", na.action = na.method(x = "include"))

rkyld.qtl <- wgaim(rkyld.asf, intervalObj = genoRxK, merge.by = "Genotype",
                   trace = "trace.txt", na.action = na.method(x = "include"))

## plot QTL intervals

# matching rainbow QTL color and trait names, red flanking markers
# (default) and gray background markers.

linkMap(list(rktgw.qtl,rkyld.qtl), genoRxK, col = "gray")

# rainbow QTL color and black trait names, red flanking markers
# (default) and gray background markers.

linkMap(list(rktgw.qtl,rkyld.qtl), genoRxK, list.col = list(t.col =
"black", m.col = "red"), col = "gray")

# monochromatic plot: gray QTLs, black trait names, black flanking
# markers and gray background markers

linkMap(list(rktgw.qtl,rkyld.qtl), genoRxK, list.col = list(q.col =
rep(gray(0.8), 2), t.col = "black", mcol = "black"), col = "gray")


## End(Not run)
```

---

linkMap.wgaim   *Plot a genetic linkage map with QTL*

Description

Neatly plots the genetic linkage map with marker locations, marker names and highlights QTL with their associated flanking markers obtained from a `wgaim` model.

Usage

```
## S3 method for class 'wgaim'
linkMap(object, intervalObj, chr, chr.dist,
   marker.names = "markers", flanking = TRUE, list.col = list(q.col = "light blue",
     m.col = "red", t.col = "light blue"), list.cex = list(t.cex = 0.6,
     m.cex = 0.6), trait.labels = NULL, tick = FALSE, ...)
```

Arguments

object	object of class `"wgaim"`
intervalObj	object of class `"cross"` or `"interval"`
chr	character string naming the subset of chromosomes to plot

chr.dist	a list containing named elements "start" and "end" containing the start and end distances in cM the genetic map should be subsetted by. Each of these may also be a vector of distances equal to the length of the number of linkage groups to be plotted.
marker.names	a character string naming the type of marker information to plot. If "dist" then distances names plotted alongside each chromosome on the left. If "markers" then marker names are plotted instead. Defaults to "markers".
flanking	logical value. If TRUE then only plot marker names or distances for flanking markers of the QTL. Defualts to TRUE
list.col	named list of colours used to highlight the QTL regions and their flanking markers. q.col is the color of the QTL regions. m.col is the color the flanking markers. t.col is the color of the trait name used in the model object (see par for colour options)
list.cex	a named list object containing the character expansion factors for the marker names m.cex and the trait labels t.cex
trait.labels	character string naming the trait used in the model object
tick	logical value. If TRUE then an axis with tick marks are generated for the chromosome names
...	arguments passed to the plot() function to set up the plot region and plot any symbols if required

Details

This plotting procedure builds on linkMap.cross() by adding the QTL regions to the map and highlighting the appropriate markers obtained from a fit to wgaim. If the linkage map is subsetted and QTL regions fall outside the remaining map a warning will be given that the QTL have been omitted from the display.

The list.col arguments q.col, m.col and t.col have been added for personal colour highlighting of the QTL regions, flanking markers and trait names. For greater flexibility the procedure may also be given the usual col argument that will be passed to the other markers.

The list.cex argument can be used to manipulate the character expansion of the marker names using m.cex or the character expansion of the trait.labels using t.cex. If a "marker" analysis has been performed then pch is used to plot a symbol at the location of the QTL. This character can be changed using the usual arguments such as pch or cex that are passed through the ... argument.

Value

For an "interval" analysis, the genetic linkage map is plotted with shaded QTL regions and highlighted flanking markers. For a "marker" analysis, a symbol is placed at the QTL locations and the markers are highlighted.

Author(s)

Julian Taylor

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical and Applied Genetics*, **116**, 95-111.

See Also

[linkMap.cross](), [wgaim]()

Examples

```
## Not run:
# fit wgaim model

yield.qtl <- wgaim(yield.fm, intervalObj = genoRxK, merge.by = "Genotype",
                trace = "trace.txt", na.action = na.method(x = "include"))

# plot QTL

linkMap(yield.qtl, genoRxK, list.col = list(m.col = "red"), col = "gray")


## End(Not run)
```

---

outStat   *A faceted ggplot of the chromosome outlier statistics or the interval*
                                 blups/outlier statistics obtained from specified iteratons of wgaim.

Description

A faceted `ggplot()` of the chromosome outlier statistics or the interval blups/outlier statistics from specified iteratons of wgaim. The interval blups/outlier statistics appear as a trace across the genome separated by chromosomes and appropriately spaced by their cM distances.

Usage

```
outStat(object, intervalObj, iter = NULL, chr = NULL, statistic =
        "outlier", plot.chr = FALSE, chr.lines = FALSE)
```

Arguments

`object`	object of class `"wgaim"`.
`intervalObj`	object of class `"interval"`.
`iter`	range of integers determining which iterations will be plotted.
`chr`	character vector naming the subset of chromosomes to plot.
`statistic`	character string naming the type of diagnostic statistic to be plotted. Default is `"outlier"` (outlier statistics). Other option is `"blups"` for the scaled empirical blups calculated during each iteration.
`plot.chr`	logical value, if TRUE then plot chromosome outlier statistics. If FALSE then plot interval outlier statistics (see Details). Defaults to FALSE.
`chr.lines`	logical value, if TRUE then plot vertical lines to show separation of linkage groups. This is only useful if `plot.chr = FALSE`. Defaults to FALSE.

Details

If `plot.chr = TRUE` then outlier statistics for each chromosome are plotted in separate faceted panels for specified values of `chr` and `iter`. This option requies `selection="chromosome"` to be set in the `wgaim.asreml()` call. If `plot.chr = FALSE` then interval blups or outlier statistics are plotted in separate faceted panels for specified values of `chr` and `iter`.

Additionally, the set of significant QTL (chromosome and interval position) are extracted from the model `object` and annotated on the plot in their appropriate positions in each facet panel. Graphical aesthetics, such as themes, text, font etc. can be further manipulated through the inclusion of additional overlays to the returned `ggplot()` object.

Value

The blups or outlier statistics are plotted in a faceted `ggplot()` with information of significant QTL overlayed.

Author(s)

Julian Taylor

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical and Applied Genetics*, **116**, 95-111.

See Also

[tr.wgaim](), [wgaim]()

Examples

```
## Not run:
# fit wgaim model

rkyld.qtl <- wgaim(rkyld.asf, intervalObj = genoRxK, merge.by = "Genotype",
                   trace = "trace.txt", na.action = na.method(x = "include"))

# plot QTL interval outlier statistics

outStat(rkyld.qtl, genoRxK, iter = 1:5)


## End(Not run)
```

---

phenoCxR   *Phenotypic Cascades x RAC875-2 zinc experiment data frame*

Description

Zinc concentration data of a Doubled Haploid wheat population.

Usage

```
data(phenoCxR)
```

Format

This data relates to a glasshouse experiment involving a set of 90 Doubled Haploid (DH) lines from a crossing of Cascades x Rac875-2. The DH lines were allocated randomly to pots in the glasshouse using a randomised complete block design. There were also additional pots that contained 5 of each of the parents (Cascades and Rac-875-2). Two measurments were made, namely zinc concentration and shoot length. The data frame consists of 200 rows and 5 columns decribed below

id: A factor of 92 levels conataining the unique identification of the DH lines and parents.

Block: A factor of two levels indexing the blocks in the experiment

Type: A factor of 3 levels indexing the wheat variety (Doubled Haploid, Cascades, Rac875-2)

shoot: A numerical variable of shoot lengths for each plant

znconc: A numerical variable of zinc concentration levels for each plant

Examples

```
data(phenoCxR, package = "wgaim")
plot(phenoCxR$shoot, phenoCxR$znconc)
```

| phenoRxK | *Phenotypic RAC875 x Kukri trial data frame* |

Description

Phenotype data arising from a field trial of a Doubled Haploid population involving a crossing of the wheat varities RAC875 and Kukri

Usage

```
data(phenoRxK)
```

Format

This data relates to a field trial conducted in 2007 at the Roseworthy Cmapus of the University of Adelaide. The trial consisted of 2 replicates of 254 Doubled Haploid lines from a cross between wheat varieties RAC875 and Kukri. The DH lines, the parents (RAC875, Kukri) and control varieties (ATIL, SOKOLL, WEEBIL) were randomly allocated to 520 plots using a randomized complete bloack design. The trial was laid out in a 20 by 26 rectangular array. The data frame consists of 520 rows with 9 columns described by:

Genotype: A 254 level factor containing a unique identification for the wheat varieties involved in the experiment.

Type: A factor of four levels indexing the wheat varieties (Doubled Haploid, RAC875, Kukri, ATIL, SOKOLL, WEEBIL).

Range: A factor of 20 numeric levels indexing the field Range.

Row: A factor of 26 numeric levels indexing the field Rows.

Block: A factor of 2 levels indexing the Blocks of the experiment.

yield: A numeric vector of yield observations in kg/ha.

tgw: A numeric vector of thousand grain weight observations.

lrange: A centred numerical vector of the field Ranges.

lrow: A centred numerical vector of the field Rows.

Examples

```
data(phenoRxK, package = "wgaim")
```

---

phenoSxT   *Phenotypic Sunco x Tasman trial data frame*

Description

Phenotype data arising from a two-phase experiment involving a Doubled Haploid population from
a crossing of the wheat varities Sunco and Tasman

Usage

```
data(phenoSxT)
```

Format

This data relates to a two-phase epxeriment involving a set of 175 Doubled Haploid lines. In the
first phase DH lines were randomly allocated to plots using a complete block design with additional
plots containing the parents (Sunco, Tasman) as well as commercial lines (Frame, Janz, Krichauff,
Machete, RAC820, Trident). The trial was laid out in a rectangular array of 31 rows and 12 columns.
In the second phase 23% of the field samples were replicated in the milling process producing a total
of 456 milling samples. These partially replicated field samples were then randomly allocated to
38 mill days with 12 samples per mill day. The data frame consists of 456 rows with 11 columns.
These columns are

Expt: A one level of factor containing a unique identification for the experiment.

Type: A factor of nine levels indexing the wheat variety (Doubled Haploid, Sunco, Tasman, (Frame,
Janz, Krichauff, Machete, RAC820, Trident))

id: A factor of 183 levels uniquely identifying the wheat varieties involved in the experiment.

Range: A factor of 12 numeric levels indexing the field Range.

Row: A factor of 31 numeric levels indexing the field Rows.

Rep: A factor of 2 levels indexing the Block of the experiment

Millday: A factor of 38 numeric levels indexing the milling day

Millord: A factor of 12 levels indexing the milling order

myield: A numeric vector of milling yield observations from the second phase of the experiment.

lord: A centered numerical vector of milling orders, Millord

lrow: A centered numerical vector of Rows

Examples

```
data(phenoSxT, package = "wgaim")
```

---

qtlTable   *Stack QTL summary information into a super table*

Description

Stack QTL summary information into a super table ready for simple exporting.

Usage

```
qtlTable(..., intervalObj = NULL, labels = NULL, columns = "all")
```

Arguments

`...`	list of objects of class `"wgaim"`. All models must have been analysed with the same `gen.type` (see help for `wgaim.asreml()`).
`intervalObj`	a genetic object of class `"interval"` reequire in a wgaim analysis (see help for `wgaim.asreml()`). This is required to be non NULL
`labels`	a vector of character strings describing the trait names of each model QTL table.
`columns`	this can be either a numeric vector determining which columns of the QTL summaries should be outputted or `"all"` for all columns. The default is `"all"`.

Details

The super table is created by obtaining the QTL summaries for each model in `...` using `summary.wgaim()` and then row binding them together. An extra column is created on the left hand side of the super table for the trait names given in the `labels` argument. If `labels = NULL` then trait names are extracted from the left hand-side of the fixed component of the associated `wgaim` model. The returned super table allows simple exporting to spreadsheet software packages or with the R/LaTeX package **xtable**.

Value

A `data.frame` object with stacked QTL summaries

Author(s)

Julian Taylor

References

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

See Also

wgaim

Examples

```
## Not run:

## fit wgaim models

rktgw.qtl <- wgaim(rktgw.asf, intervalObj = genoRxK, merge.by = "Genotype",
                   trace = "trace.txt", na.action = na.method(x = "include"))

rkyld.qtl <- wgaim(rkyld.asf, intervalObj = genoRxK, merge.by = "Genotype",
                   trace = "trace.txt", na.action = na.method(x = "include"))

## create super table and export

qtlt <- qtlTable(rktgw.qtl, rkyld.qtl, labels = c("Conc.", "Shoot"))
print(xtable(qtlt), file = "superQTL.tex", include.rownames = FALSE)

## End(Not run)
```

---

summary.wgaim	*Summary and print methods for the class* "wgaim"

Description

Prints a QTL summary form the "wgaim" object in a presentable format

Usage

```
## S3 method for class 'wgaim'
summary(object, intervalObj, LOD = TRUE, ...)
## S3 method for class 'wgaim'
print(x, intervalObj, ...)
```

Arguments

object	an object of class "wgaim" (see Details)
x	an object of class "wgaim"
intervalObj	a data structure of class "cross" or "interval" containing the genotypic data
LOD	logical value. If TRUE LOD scores for QTL are calculated, defaults to TRUE
...	further arguments passed to or from other methods

Details

It is important that the intervalObj is not missing in summary.wgaim() or print.wgaim() as it contains vital summary information about each of the QTL detected.

The summary of the QTL differs depending on the method chosen in the wgaim.asreml call. If method = "random" then the significance of the QTL are summarized using a probablistic argument

based on the conditional distribution of the QTL sizes given the data (see Verbyla et. al, 2012 in References) Thus, for each QTL, a value is calculated that represents the probability that the QTL size is greater than zero (or less than zero if the effect is negative). If method = "fixed" then the significance of the QTL is summarized using a one degree of freedom Wald statistic.

Value

A summary of the QTL component of the "wgaim" object is printed to the screen. For each QTL detected, if an "interval" analysis was performed then summary.wgaim() prints which chromosome, name and distance of each flanking marker, size, probability/p-value, contribution of genetic variance and LOD score if desired. If a "marker" analysis was performed then the chromosome, name and distance of the associated marker, size, probability/p-value, contribution of genetic variance and LOD score are printed. print.wgaim() provides a narrative brief of the QTL detected.

Author(s)

Julian Taylor and Ari Verbyla

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical and Applied Genetics*, **116**, 195-211.

See Also

wgaim.asreml

Examples

```
## Not run:
# read in data

data(phenoRxK, package = "wgaim")
data(genoRxK, package = "wgaim")

# subset linkage map and convert to "interval" object

genoRxK <- subset(genoRxK, chr = c("1A", "2D1", "2D2", "3B"))
genoRxK <- cross2int(genoRxK, impute = "Martinez", id = "Genotype")

# base model

rkyld.asf <- asreml(yld ~ lrow, random = ~ Genotype + Range,
```

```
                          residual = ~ ar1(Range):ar1(Row), data = phenoRxK)

    # find QTL

    rkyld.qtl <- wgaim(rkyld.asf, intervalObj = genoRxK, merge.by = "Genotype",
                       trace = "trace.txt", na.action = na.method(x = "include"))

    # summarise

    print(rkyld.qtl, genoRxK)
    summary(rkyld.qtl, genoRxK)


    ## End(Not run)
```

---

tr.wgaim   *Display diagnostic information about the QTL detected.*

Description

Displays diagnostic infomation about QTL detection and significance for the sequence of models generated in a wgaim analysis.

Usage

```
## S3 method for class 'wgaim'
tr(object, iter = 1:length(object$QTL$effects),
      lik.out = TRUE, ...)
```

Arguments

object	an object of class "wgaim"
iter	a vector of integers specifying what rows of the p-value matrix to display
lik.out	logical value. If TRUE then diagnostic information about the testing of the genetic variance is given for all iterations.
...	arguments passed to print.default for displaying of information

Details

By default the printing of the objects occur with arguments quote = FALSE and right = TRUE. Users should avoid using these arguments.

Value

For the selected QTL, a probability value matrix is displayed with rows specified by iter. If lik.out = TRUE then a matrix with rows consisting of the likelihood with additive genetic variance, the likelihood without additive genetic variance (NULL model), the test statistic and the p-value for the statistic.

Author(s)

Julian Taylor

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

See Also

wgaim

Examples

```
## Not run:
# read in data

data(phenoRxK, package = "wgaim")
data(genoRxK, package = "wgaim")

# subset linkage map and convert to "interval" object

genoRxK <- subset(genoRxK, chr = c("1A", "2D1", "2D2", "3B"))
genoRxK <- cross2int(genoRxK, impute = "Martinez", id = "Genotype")

# base model

rkyld.asf <- asreml(yld ~ lrow, random = ~ Genotype + Range,
                    residual = ~ ar1(Range):ar1(Row), data = phenoRxK)

# find QTL

rkyld.qtl <- wgaim(rkyld.asf, intervalObj = genoRxK, merge.by = "Genotype",
                   trace = "trace.txt", na.action = na.method(x = "include"))

# diagnostic check

tr(rkyld.qtl, digits = 4)

## End(Not run)
```

---

wgaim.asreml   *wgaim method for class "asreml"*

Description

Implements the iterative Whole Genome Average Interval Mapping (wgaim) algorithm using the functionality of the flexible linear mixed modelling R package **ASReml-R** V4.

Usage

```
## S3 method for class 'asreml'
wgaim(baseModel, intervalObj, merge.by = NULL,
        fix.lines = TRUE, gen.type = "interval", method = "fixed",
        selection = "interval", exclusion.window = 20, breakout = -1,
        TypeI = 0.05, trace = TRUE, verboseLev = 0, ...)
```

Arguments

baseModel	a linear mixed model object of class "asreml" usually representing a base asreml() model to be extended.
intervalObj	a list object containing the genotypic data, usually an "interval" object obtained from using cross2int (see Details).
merge.by	a character string or name of the column(s) in phenoData and intervalObj to merge the phenotypic and genotypic data sets.
fix.lines	A logical value. If TRUE then lines existing in the phenotype data that do not exist in intervalObj are fixed and placed in the fixed component of the asreml() models (see Details). It is recommended to set this to TRUE. Defaults to TRUE.
gen.type	a character string determining the type of genetic data to be used in the analysis. Possibilities are "interval" and "markers". The default is "interval". (see Details).
method	a character string determining the type of algorithm to be used in the analysis. Possibilities are "random" and "fixed". The default is "random". (see Details).
selection	a character string determining the type of selection method that is used to select QTL in the analysis. Possibilities are "interval" and "chromosome". The default is "interval". (see Details).
exclusion.window	
	For each QTL, the distance in centimorgans on the left and right side of each QTL that is excluded from further analysis.
breakout	A numerical integer equivalent to the iteration where the algorithm breaks out. The default is -1 which ensures the algorithm finds all QTL before halting. (see Details)
TypeI	a numerical value determining the familywise alpha level of significance for detecting a QTL. The default is 0.05.

trace   An automatic tracing facility. If `trace = TRUE` then all `asreml` output is piped to the screen during the analysis. If `trace = "file.txt"`, then output from all asreml models is piped to `"file.txt"`. Both trace machanisms will display a message if a QTL is detected.

verboseLev   numerical value, either 0 or 1, determining the level of tracing outputted during execution of the algorithm A 0 value will produce the standard model fitting output from the fitted ASReml models involved in the forward selection. A value of 1 will add a table of chromosome and interval outlier statistics for each iteration.

...   Any extra arguments to be passed to the `asreml` calls. (see `?asreml` and `?asreml.options` for more information).

Details

In the initial call to `wgaim.asreml()`, the marker or interval information is collated from `intervalObj`. If `gen.type = "interval"` then midpoints of intervals are collated from the `"interval.data"` components of the chromosomes in `intervalObj`. If `gen.type = "markers"` then markers are collated from the `"imputed.data"` components of the chromosomes in `intervalObj`.

It is recommended to set `fix.lines = TRUE` to ensure additive and non-additive genetic variances are estimated from lines in the `merge.by` component of the phenotypic data that have genetic marker data in `intervalObj`. Lines in the phenotype `merge.by` factor not existing in `intervalObj` will be placed as a fixed factor (called `Gomit`) in the `asreml` model. Note, if there are others factors in the model that have some potential confounding with `Gomit` then `asreml` will indicate this with a simple message `'Terms with zero df listed in attribute 'zerodf' of the wald table'` at the end of its iterative maximisation. This confounding will have no effect on the outcome and can be safely ignored. If `fix.lines = FALSE` is set then all available lines in the `merge.by` component of the phenotypic data will be used to estimate the non-additive genetic variance. In this instance, users also need to be aware that `asreml` will output a large number of warnings due to an inherent mismatch in the levels of the lines contained in the phenotype data compared to the lines in `intervalObj`.

The `method` argument in `wgaim.asreml()` allows the user access to two algorithms. If `method = "fixed"` the algorithm places selected QTL as an additive set of fixed effects in the model as the forward selection algorithm proceeds. If `method = "random"` places selected QTL in the random part of the model as an additive set of random effects. This new formulation is outlined in Verbyla et. al (2012).

The `selection` argument determines the type of selection algorithm for the analysis. If `selection = "chromosome"` then outlier statistics for each chromosome are calculated and the largest chromosome or linkage group is chosen. The largest marker/interval outlier statistic in this linkage group is then selected as the putative QTL. If `selection = "interval"`, only marker/interval statistics are calculated and the largest marker/interval is chosen as the putative QTL.

Note: If a genetic map has a small number of markers on a linkage group then using `selection = "chromosome"` as the selection algorithm is known to be flawed (see Verbyla et. al, 2012). For this reason it is suggested that this option only be used when there are a moderate number of markers on each linkage group.

Users can break out of the algorithm using the `breakout` argument. If a numerical value greater than zero is given, then the forward selection algorithm breaks at the iteration equal to that value and returns the collected information to this point. This includes fixed/random QTL effects, diaganostic components such as interval/marker BLUPs and outlier statistics as well as the trace components of

the algorithm. It should be noted that the algorithm breaks out before a QTL has been moved to the fixed/random effects and estimated. Therefore a positive integer, say n will not return an estimate of the nth QTL but it will return the outlier statistics or BLUPs for the nth iteration.

It is recommended that `trace = "file.txt"` be used to pipe the sometimes invasive tracing of `asreml` licensing and fitting numerics for each model to a file. Errors, warnings and messages will still appear on screen during this process. Note some warnings that appear may be passed through from an asreml call and are outputted upon exit. These may be ignored as they are handled during the execution of the function.

Value

An object of class `"wgaim"` which also inherits the class `"asreml"` by default. The object returned is actually an `asreml` object (see `asreml.object`) with the addition of components from the QTL detection listed below.

QTL   A list of components from the significant QTL detected including a character vector of the significant QTL along with a vector of the QTL effect sizes. There are also a number of diagnostic meausres that can be found in `diag` that are used in conjunction with `tr.wgaim` and `outStat`.

Author(s)

Julian Taylor and Ari Verbyla

References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL http://www.jstatsoft.org/v40/i07/.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical and Applied Genetics*, **116**, 95-111.

See Also

print.wgaim, summary.wgaim

Examples

```
## Not run:
# read in data

data(phenoRxK, package = "wgaim")
data(genoRxK, package = "wgaim")

# subset linkage map and convert to "interval" object
```

```
genoRxK <- subset(genoRxK, chr = c("1A", "2D1", "2D2", "3B"))
genoRxK <- cross2int(genoRxK, impute = "Martinez", id = "Genotype")

# base model

rkyld.asf <- asreml(yld ~ Type + lrow, random = ~ Genotype + Range,
                    residual = ~ ar1(Range):ar1(Row), data = phenoRxK)

# detect and estimate QTL

rkyld.qtl <- wgaim(rkyld.asf, intervalObj = genoRxK, merge.by = "Genotype",
                    trace = "trace.txt", na.action = na.method(x = "include"))


## End(Not run)
```

Index