

Package ‘wgaim’

February 15, 2012

Type Package

Title Whole Genome Average Interval Mapping for QTL detection using mixed models

Version 1.2

Date 2011-09-21

Author Julian Taylor <julian.taylor@csiro.au>, Simon Diffey
<simon.diffey@dpi.nsw.gov.au>, Ari Verbyla <ari.verbyla@csiro.au> and Brian Cullis
<brian.cullis@dpi.nsw.gov.au>.

Maintainer Julian Taylor <julian.taylor@csiro.au>

Depends R (>= 2.0.0), qtl, lattice

SystemRequirements asreml-R

Description This package integrates sophisticated mixed modelling
methods with a whole genome approach to detecting significant QTL’s in linkage maps.

License GPL (>= 2)

Repository CRAN

Date/Publication 2011-11-27 09:14:22

R topics documented:

wgaim-package	2
cross2int	3
link.map.cross	6
link.map.default	7
link.map.wgaim	10
out.stat	12
qtlTable	13
raccas	15
summary.wgaim	15
tr.wgaim	17
wgaim.asreml	19
zinc	21

wgaim-package

Whole Genome Average Interval Mapping (wgaim) for QTL detection

Description

This package uses sophisticated mixed modelling methods with the addition of allowing a whole genome approach to detecting significant QTL in linkage maps.

Details

Package: wgaim
Type: Package
Version: 1.1
Date: 2011-10-24
License: GPL 2

Welcome to version 1.x of wgaim! The documentation given in this help file is only brief and users should eventually consult the soon to be released vignette that will be a companion to version 1.0 and above.

Version 1.x package highlights:

1. The inclusion of high dimensional genetic data
2. Users have a choice of whole genome marker or interval analysis.
3. Selection and estimation of QTL effects occurs using the implementation of Verbyla & Taylor (2011).

Deprecated functions:

1. `read.interval`: `cross2int` is the conversion function that should be used.
2. `wmerge`: This function is not required as the genetic data is merged within the wgaim call.

This package builds on the **qtl** package of Broman by including additional functions for whole genome QTL analysis of a full linkage map using linear mixed models.

The package provides a user friendly function `cross2int` for the conversion of "cross" objects created using `read.cross` in Broman's **qtl** package into an "interval" object or use in **wgaim**. Specifically, `cross2int` performs additional calculations for imputing missing 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 thus allows further use with the **qtl** package if desired.

The package also provides a very neat graphical display of the chromosomes of a "cross" object. The method function `link.map` 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.

Modelling of the QTL's is achieved using the functions `wgaim` which, as its first argument, requires an `asreml` base model. Version 1.0 of `wgaim` allows users to include high dimensional genetic components in a `wgaim` analysis (See `wgaim.asreml` for more details or the soon to be released vignette companion to this version). For convenience the default tracing of results from the `asreml` models is outputted to a file for further inspection. For diagnostic purposes, the outlier statistics from each iteration can be viewed using `out.stat`. 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 and significance (including LOD score). The returned "wgaim" object may also be plotted using the method function `link.map`. 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 `link.map.default`)

Author(s)

Julian Taylor, Simon Diffey, Ari Verbyla and Brian Cullis Maintainer: Julian Taylor <julian.taylor@csiro.au>

References

Verbyla, A. P & Taylor, J. D (2011). High dimensional whole genome average interval mapping and a random effects formulation. *Theoretical and Applied Genetics*. Submitted.

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(fullgeno, missgeno = "MartinezCurnow", rem.mark = TRUE,
          id = "id", subset = NULL)
```

Arguments

fullgeno	an object of class "cross" with restrictions (see Details)
missgeno	a character string determining how missing values in the linkage map should be imputed. If "Broman", then missing values are imputed according to Broman's 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).
rem.mark	logical value. If TRUE redundant markers are deleted and placed in the component of the object (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 created using `read.cross` from Broman's **qtl** package to "interval" objects ready for use with `wgaim`.

User should be aware that this function is restricted to populations with only two genotypes. `fullgeno` is checked for the the class structure `c("bc", "cross")`. If this is not present an error is returned.

During the conversion process, missing values are imputed according to the argument given by `missgeno`. This imputation results in a complete version of the marker data for each chromosome which is then used to create the interval data "intval". 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.

If `rem.mark = TRUE` then markers that are identical are removed before missing values are imputed. The marker similarity is found by estimating recombination fractions using `est.rf` from the **qtl** package. The removed list is returned with the object and placed under "cor.markers" for inspection if required.

Value

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

geno	This is 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 NA's replaced by imputed values according to the rules of "missgeno"; "dist" contains the genetic distance between adjacent markers or the genetic distances of the intervals; "theta" contains the recombination fractions for each interval; "intval" contains the recalculated intervals based on the recombination fractions and the missing marker information.
------	--

cor.markers	If rem.mark = TRUE, a three column matrix with each row describing which pairwise markers are correlated and what chromosome they are from.
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 read.cross).

Author(s)

Julian Taylor, Simon Diffey, Ari Verblyla and Brian Cullis

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/>.

Verblyla, 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")
racca <- read.cross("csvr", file="raccas.csv", genotypes=c("AA","AB"),
  na.strings = c("-", "NA"), dir = wgpath)
raccas <- cross2int(racca, missgeno="MartinezCurnow", id = "id")

# plot linkage map

link.map(raccas, cex = 0.5)

## End(Not run)
```

link.map.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'
link.map(object, chr, max.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
max.dist	a numerical value in cM determining the distance the genetic map should be subsetted by
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"
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 QTLs using link.map.wgaim and link.map.default
m.cex	the expansion factor to use for the marker names
...	arguments passed to "plot" to set up the plot region. Arguments may also be passed to "text" for the manipulation of the marker names

Details

This plotting procedure provides a neater 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 "text" 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 `link.map.wgaim`

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

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

[link.map.wgaim](#)

Examples

```
data(raccas, package = "wgaim")
link.map(raccas, cex = 0.5)
```

`link.map.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's with their associated flanking markers for multiple traits obtained from a list of `wgaim` fits.

Usage

```
## Default S3 method:
link.map(object, intervalObj, chr, max.dist, marker.names
  = "markers", 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

<code>object</code>	a list object with elements inheriting the class "wgaim"
<code>intervalObj</code>	object of class "cross" or "interval"
<code>chr</code>	character string naming the subset of chromosomes to plot
<code>max.dist</code>	a numerical value in cM determining the distance the genetic map should be subsetted by
<code>marker.names</code>	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".
<code>list.col</code>	named list of colors used to highlight the QTL regions and their flanking markers. <code>q.col</code> is the colors of the QTL regions (defaults to <code>rainbow(n)</code> where <code>n</code> is the length of <code>object</code>). <code>m.col</code> is the color the flanking markers. <code>t.col</code> is the color of the trait names used in each model (defaults to the same color as the QTL regions). See <code>par</code> for color options
<code>list.cex</code>	a named list object containing the character expansion factors for the marker names <code>m.cex</code> and the trait labels <code>t.cex</code>
<code>trait.labels</code>	character string naming the trait used in the model object, defaults to the names of the traits used in each model.
<code>tick</code>	logical value. If TRUE then an axis with tick marks are generated for the chromosome names
<code>...</code>	arguments passed to "plot" to set up the plot region. Arguments may also be passed to "text" for the manipulation of the marker names

Details

This plotting procedure is a wrapper for `link.map.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 highlighting of the QTL is also available using `clist`. This differs slightly from `link.map.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.

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 usual `...` 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

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[link.map.cross](#), [link.map.wgaim](#)**Examples**

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

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccasS,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

zn.qtlS <- wgaim(zn.fmS, phenoData = zinc, intervalObj = raccasS,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

## plot QTL intervals

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

link.map(list(zn.qtl,zn.qtlS), raccas, col = "gray")

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

link.map(list(zn.qtl,zn.qtlS), raccas, 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

link.map(list(zn.qtl,zn.qtlS), raccas, list.col = list(q.col =
rep(gray(0.8), 2), t.col = "black", mcol = "black"), col = "gray")

## End(Not run)
```

link.map.wgaim

*Plot a genetic linkage map with QTL's***Description**

Neatly plots the genetic linkage map with marker locations, marker names and highlights QTL's with their associated flanking markers obtained from a fit to wgaim.

Usage

```
## S3 method for class 'wgaim'
link.map(object, intervalObj, chr, max.dist,
  marker.names = "markers", 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
max.dist	a numerical value in cM determining the distance the genetic map should be subsetted by
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".
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 "plot" to set up the plot region and plot any sybols if required

Details

This plotting procedure builds on link.map.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

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

[link.map.cross](#), [wgaim](#)

Examples

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

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

# plot QTL

link.map(zn.qtl, raccas, list.col = list(m.col = "red"), col = "gray")

## End(Not run)
```

out.stat *Plot the interval outlier statistics from specified iterations of wgain*

Description

Plots the interval outlier statistics for specified iterations of wgain. The interval statistics appear as a trace across the genome separated by chromosome and appropriately spaced by their distances.

Usage

```
out.stat(object, intervalObj, int = TRUE, iter = NULL, chr = NULL, ...)
```

Arguments

object	object of class "wgain"
intervalObj	object of class "cross" or "interval"
int	logical value, if TRUE then plot interval outlier statistics. If FALSE then plot chromosome outlier statistics. This can only be done if the analysis method for the object was "fixed"
iter	numeric value determining which iterations will be plotted
chr	character string naming the subset of chromosomes to plot. This can only be used when int is TRUE
...	arguments passed to "xyplot" or "barchart" (with some restrictions, see Details)

Details

By default the interval outlier statistics are plotted in separate panels for each iteration in a set layout of 5 rows and one column. This cannot be adjusted and users should not attempt to use the layout argument. Viewing multiple pages can be done by specifying the appropriate iterations using the iter argument.

The set of QTL are obtained from the model and printed on the plot in their appropriate positions in each panel.

Value

The outlier statistics are plotted in a trellis panel plot.

Author(s)

Julian Taylor

References

Julian Taylor, Arunas Vebyla (2011). R Package wgain: 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

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

Examples

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

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

# plot QTL interval outlier statistics

out.stat(zn.qtl, raccas, iter = 1:5, cex = 0.4)

## End(Not run)
```

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

ldots	list of objects of class "wgaim". All "wgaim" models must have been analysed with the same genetic type (see <code>wgaim.asreml</code>)
intervalObj	a genetic object of class "interval" usually used in a <code>wgaim</code> fit
labels	a vector of character strings determining the trait names of each QTL table. if this is NULL then the trait names are found from the response of the <code>wgaim</code> model
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".
...	a numeric vector determining which columns for the summary should be returned (see Details)

Details

The super table is created by stacking the QTL summaries on top of each other using the models in ... from left to right. An extra column is created on the left hand side of the stacked table for the trait names given in the labels argument. The names are only placed in the first element of each table with NAs for the rest of the elements. This then allows simple exportation to spreadsheet packages or with the 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

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

zn.qtlS <- wgaim(zn.fmS, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

## create super table and export

qtlTable <- qtlTable(zn.fm, zn.fmS, labels = c("Conc.", "Shoot"))
print(xtable(qtlTable), file = "superQTL.tex", include.rownames = FALSE)

## End(Not run)
```

raccas	<i>Genotypic marker data</i>
--------	------------------------------

Description

Genotypic marker data representing a full linkage map from a Double Haploid population

Usage

```
data(raccas)
```

Format

A data object of class "cross" inheriting the class "interval".

Details

This is data on 93 individuals from a Double Haploid population typed at 663 markers over 40 chromosomes. Coincident markers have been omitted and the data set has been reduced to 468 markers. This genotypic data has actually been read in from a call to `read.interval` and is therefore a list with components of the a returned object from `read.interval`. See the aforementioned function for more details.

Examples

```
data(raccas, package = "wgaim")
link.map(raccas, cex = 0.5)
```

summary.wgaim	<i>Summary and print methods for the class "wgaim"</i>
---------------	--

Description

Prints a summary of 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"
x	an object of class "wgaim" (see Details)
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.

As the WGAIM algorithm now places the selected QTL effects in a separate random component term of the model, they are summarised appropriately using a probabilistic argument based on the conditional distribution of the QTL sizes given the data (see Verbyla & Taylor, 2011 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).

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 and LOD score (based on the random coefficient) if desired. If a "marker" analysis was performed then the chromosome, name and distance of the associated marker, size, probability and LOD score are printed. `print.wgaim` provides a narrative brief of the QTL's detected.

Author(s)

Julian Taylor, Simon Diffey, Ari Verbyla and Brian Cullis

References

Verbyla, A. P & Taylor, J. D (2011). High dimensional whole genome average interval mapping and a random effects formulation. *Theoretical and Applied Genetics*. Submitted.

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(zinc, package = "wgaim")
data(raccas, package = "wgaim")

# subset linkage map and convert to "interval" object

raccas <- subset(raccas, chr = c("1A1", "2D1", "4D2", "6A1"))
raccas <- cross2int(raccas, missgeno = "Martinez")

## base model

zn.fm <- asreml(znconc ~ Type, random = ~ Block + id, data = zinc)

# find QTL

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

# summarise

print(zn.qtl, raccasM)
summary(zn.qtl, raccasM)

## End(Not run)
```

tr.wgaim

Display diagnostic information about wgaim QTL model

Description

Displays diagnostic information about QTL detection and significance for the sequence of models used in a wgaim fit.

Usage

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

Arguments

object	an object of class "wgaim"
iter	a vector of integers describing the rows of the p-value matrix to display

diag.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 including these arguments.

Value

A probability value matrix for the successive QTL selected is displayed with rows according to the iterations specified. If `diag.out = TRUE` then a matrix with rows consisting of the likelihood with genetic variance, the likelihood without genetic variance (NULL model), the statistic and the p-value for the statistic.

Author(s)

Julian Taylor

References

Verbyla, A. P & Taylor, J. D (2011). High dimensional whole genome average interval mapping and a random effects formulation. *Theoretical and Applied Genetics*. Submitted.

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(zinc, package = "wgaim")
data(raccas, package = "wgaim")

# subset linkage map and convert to "interval" object

raccas <- subset(raccas, chr = c("1A1", "2D1", "4D2", "6A1"))
raccas <- cross2int(raccas, missgeno = "Martinez")

# base model

zn.fm <- asreml(znconc ~ Type, random = ~ Block + id, data = zinc)

# find QTL's
```

```

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

# diagnostic check

tr(zn.qtl, digits = 4)

## End(Not run)

```

wgaim.asreml	<i>wgaim method for class "asreml"</i>
--------------	--

Description

Fits an iterative Whole Genome Average Interval Mapping (wgaim) model for QTL detection

Usage

```

## S3 method for class 'asreml'
wgaim(baseModel, phenoData, intervalObj, merge.by = NULL,
      gen.type = "interval", method = "random", TypeI = 0.05, attempts = 5,
      trace = TRUE, verboseLev = 0, ...)

```

Arguments

baseModel	a model object of class "asreml" usually representing a base model with which to build the qtl model.
phenoData	a data frame containing the phenotypic elements used to fit baseModel. This data is checked against the base models data
intervalObj	a list object containing the genotypic data, usually an "interval" object obtained from using cross2int. This object may contain many more markers than observations (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.
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).
TypeI	a numerical value determining the level of significance for detecting a QTL. The default is 0.05.
attempts	An integer representing the number of attempts at convergence for the fixed or random qtl model. The default is 5.

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 mechanisms 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 other extra arguments to be passed to each of the asreml calls. These may also include asreml.control arguments.

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 "intval" component of `intervalObj`. If `gen.type = "markers"` then markers are collated from the "imputed.data" component of `intervalObj` (It should be noted that a "marker" analysis is less efficient than an "interval" analysis as it does not take into account the correlation of the marker effects in the specificity of the model; see Verbyla et. al, 2007).

The method argument in `wgaim.asreml` allows the user access to two algorithms. If `method = "fixed"` then the forward selection algorithm uses the legacy code packaged with pre-1.0 versions of **wgaim**. This code uses the chromosome statistics as a guide to its selection of QTL. This selection process is now known to be flawed when small linkage groups are present. For this reason it is suggested that this option only be used when there are a moderate number of markers on each linkage group.

This version of **wgaim** allows high dimensional marker information to be analysed. A simple transformation of the collated high dimensional marker set shows that it may be reduced to the number of genetic lines used in the analysis. This transformation is internal to the `wgaim.asreml` call and users can now expect a considerably large acceleration in the performance of **wgaim**.

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 measures that can be found in <code>diag</code> that are used in conjunction with <code>tr.wgaim</code> and <code>out.stat</code> .
-----	---

Author(s)

Julian Taylor, Simon Diffey, Ari Verbyla and Brian Cullis

References

Verbyla, A. P & Taylor, J. D (2011). High dimensional whole genome average interval mapping and a random effects formulation. *Theoretical and Applied Genetics*. Submitted.

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(zinc, package = "wgaim")
data(raccas, package = "wgaim")

# subset linkage map and convert to "interval" object

raccas <- subset(raccas, chr = c("1A1", "2D1", "4D2", "6A1"))
raccas <- cross2int(raccas, missgeno = "Martinez")

# base model

zn.fm <- asreml(znconc ~ Type, random = ~ Block + id, data = zinc)

# find QTL's

zn.qtl <- wgaim(zn.fm, phenoData = zinc, intervalObj = raccas,
merge.by = "id", trace = "trace.txt", na.method.X = "include")

## End(Not run)
```

zinc

Zinc data

Description

Zinc concentration data of a Double Haploid wheat population.

Usage

```
data(zinc)
```

Format

A data frame with 200 observations on the following 5 variables

id The identification for the Double Haploid or Control line

Block A blocking variable for the experiment

Type The type of wheat variety (Double Haploid, Cascades, Rac875-2)

shoot The shoot length of each plant

znconc Zinc concentration level of each plant

Examples

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

Index

*Topic **datasets**

raccas, [15](#)
zinc, [21](#)

*Topic **hplot**

link.map.cross, [6](#)
link.map.default, [7](#)
link.map.wgaim, [10](#)
out.stat, [12](#)

*Topic **package**

wgaim-package, [2](#)

*Topic **regression**

cross2int, [3](#)
qtlTable, [13](#)
summary.wgaim, [15](#)
tr.wgaim, [17](#)
wgaim.asreml, [19](#)

cross2int, [2](#), [3](#)

link.map, [2](#), [3](#)
link.map(link.map.cross), [6](#)
link.map.cross, [6](#), [9](#), [11](#)
link.map.default, [3](#), [7](#)
link.map.wgaim, [7](#), [9](#), [10](#)

out.stat, [3](#), [12](#)

print.summary.wgaim(summary.wgaim), [15](#)
print.wgaim, [21](#)
print.wgaim(summary.wgaim), [15](#)

qtl-package, [3](#)
qtlTable, [13](#)

raccas, [15](#)
read.cross, [5](#)

summary.wgaim, [15](#), [21](#)

tr.wgaim, [3](#), [13](#), [17](#)

wgaim, [3](#), [11](#), [13](#), [14](#), [18](#)

wgaim(wgaim.asreml), [19](#)
wgaim-package, [2](#)
wgaim.asreml, [16](#), [19](#)

zinc, [21](#)