Package 'qtl.outbred'

March 21, 2011

Type Package

Title Interface for Genotype Probabilities from Outbred Intercross

Version 2011.3.17

Date 2011-03-17

Author Ronnie Nelson and Xia Shen

Maintainer Xia Shen <xia.shen@lcb.uu.se>

Description The package calculates and imports genotype probabilities from outbred intercross designs into an R object. This allows fast scans for main effect QTL and two way interactions. Note to Linux users: On installation add the argument to the install.packages function: lib = paste(R.home(), '/library/',sep = '')

License GPL

LazyLoad yes

Depends qtl

R topics documented:

`qtl.outbred-package` *Interface for Genotype Probabilities of Outbred Intercross*

Description

The package calculates and imports genotype probabilities from outbred intercross designs into an R object. This allows fast scans for main effect QTL and two way interactions. Using this interface, most functions in package `qtl` can be used for outbred designs.

Details

Package:	qtl.outbred
Type:	Package
Version:	2011.3.17
Date:	2011-03-17
License:	GPL
LazyLoad:	yes
Depends:	qtl

Author(s)

Ronnie Nelson and Xia Shen

Maintainer: Xia Shen <xia.shen@lcb.uu.se>

See Also

impo.prob, calc.prob

Examples

```
## Not run:
file <- paste(R.home(), '/library/qtl.outbred/example_data.csv', sep = '')
data <- read.cross('csv', file = file, genotypes = c('AA', 'AB', 'BB'))
path <- paste(R.home(), '/library/qtl.outbred/', sep = '')
outbred.data <- impo.prob(cross.data = data, path = path, stepsize = 5)

# ----- QTL scan using Haley-Knott regression and plotting
hk.model <- scanone(outbred.data, method = 'hk', pheno.col = 2)
plot(hk.model)

## End(Not run)
```

---

calc.prob *Calculate genotype probabilities for outbred intercrosses using*

Description

The function calculates genotype probabilities from outbred intercross data using published software 'cnF2freq'. The software is compiled for Windows and Unix operating systems. However the source code is included in the package installation folder or can be downloaded from: http://user.it.uu.se/~carln/cnF2freq.html

Usage

```
calc.prob(os = 'unix', stepsize = 1,
marker.info.file = 'mi.txt', pedigree.file = 'mp.txt',
genotype.file = 'mg.txt', output.file = 'cnout')
```

Arguments

os	A string specifying the operating system, typically 'unix' or 'windows'.
stepsize	An integer giving the distance that the two consecutive test loci are spaced in centi-morgan.
marker.info.file	A marker information file required by cnF2freq, see Details.
pedigree.file	A pedigree information file required by cnF2freq, see Details.
genotype.file	A genotype data file required by cnF2freq, see Details.
output.file	An output file name required by cnF2freq, see Details.

Details

Three files are required for the calculating of genotype probabilities. All the files are tab-delimited. The marker information files provides information on the number of chromosomes, the total number of markers, followed by the number of per chromosomes and the name of each. Finally the space between each marker for each chromosome is provided. Note that the first column of each row should be filled with a '1' followed by the position data (each chromosome in a single row). The genotype file provide the genotypic information for each individual. The individuals IDs are indicated in the first column of each row. The following columns are filled with integer values indicating the genotype at each locus in sequence. Every 2 columns corresponds to one marker (one allele in each column) and the columns are arranged sequentially in the order described in the marker file. The marker pedigree file are arranged in fullsib families. For each family the the number of F2 individuals within the family are provided. This is followed wit the individual ID's in the first column (starting with the F0 generation, then the F1s and then the F2s). For each individual his/her parent's ID's are provided in the next two columns followed by it's sex. Note for the F0 generation their parents are indicated with a '0' while an additional column with their line origins are provided.

Missing values should be indicated as 0.

Three example files with the required format described above are provided in the package installation folder. 'mp.txt', 'mi.txt', and 'mg.txt' are the pedigree information file, marker information file, and the marker genotype file, respectively.

Note that Perl is required for some of the functions in the package qtl.outbred. and can be downloaded from: http://www.perl.org/

Value

The function generates a file named by output.file for each chromosome containing genotype probabilities that can be thereafter imported using impo.prob.

Author(s)

Ronnie Nelson and Xia Shen

See Also

impo.prob, qtl.outbred-package

Examples

```
## Not run:
# ----- calc.prob example
# (calculating the genotype probabilities from the files
# 'mi.txt', 'mp.txt' and 'mg.txt' at 5 cM intervals and produce the files
# 'p_chrom_A.txt', where A indicates the chromosome number for each chromosome).
setwd(paste(R.home(), '/library/qtl.outbred/', sep = ''))
calc.prob(stepsize = 5)

## End(Not run)
```

---

impo.prob *Import genotype probabilities into an existing cross object*

Description

The function imports genotype probabilities into an existing qtl object.

Usage

```
impo.prob(cross.data, path = NULL, stepsize = 1, Grid = FALSE)
```

Arguments

cross.data	A cross object typically read in via read.cross in package qtl.
path	The directory path where the genotype probabilities input files are located.
stepsize	An integer giving the distance that the two consecutive test loci are spaced in centi-morgan.
Grid	If the genotype probabilities are saved from a GridQTL coefficient file it can be imported directly if TRUE.

Details

The data in the object of class cross is used to create a new object of the same class, while importing the genotype probabilities from a set of files. It is necessary to import create the original object prior to using the impo.prob function (see read.cross for more information).

The genotype probabilities are imported from a file for each chromosome separately. In this format first column indicates the individual id. The next three columns are the genotype probabilities of being homozygous, heterozygous and homozygous for the other alleles respectively (i.e. AA, AB and BB). A separate row for each centi-morgan interval per individual is provided. The rows are arranged to have the probabilities per individual together (i.e. from position 1 to end of chromosome for individual 1 followed by position 1 to end of chromosome for individual 2 etc.). It is important to note that the filenames should be for each chromosome as follows (where A indicates the chromosome number): 'p_output_chrom_A.txt'. This is also the file output created by calc.prob.

A set of example files with the required format described above are provided in the package installation folder. 'p_output_chrom_1.txt' and 'p_output_chrom_2.txt' contains the genotype probabilities for the two chromosomes to be imported. 'example_data.csv' is an F2 intercross file that can be read by read.cross function in package qtl.

Alternatively, the genotype probabilities can be imported directly from the coefficient file generated by GridQTL. This file will contain the genotype probabilities for all the chromosomes specified when the job is submitted to the GridQTL server (see installation folder for an example file). It is important to note that the filename for this file should be 'Gout.coe', and should be located in the working directory.

Note that Perl is required for some of the functions in the package `qtl.outbred` and can be downloaded from: http://www.perl.org/

Value

The new cross object `cross.data` is returned, with the genotype probabilities imported.

Author(s)

Ronnie Nelson and Xia Shen

See Also

`calc.prob`, `qtl.outbred-package`, `read.cross`

Examples

```
## Not run:
# ----- impo.prob example
file <- paste(R.home(), '/library/qtl.outbred/example_data.csv', sep = '')
data <- read.cross('csv', file = file, genotypes = c('AA', 'AB', 'BB'))
outbred.data <- impo.prob(cross.data = data, stepsize = 5, path =
                          paste(R.home(), '/library/qtl.outbred/', sep = ''))

# ----- impo.prob example for GridQTL output
file <- paste(R.home(), '/library/qtl.outbred/example_data.csv', sep = '')
data <- read.cross('csv', file = file, genotypes = c('AA', 'AB', 'BB'))
setwd(paste(R.home(), '/library/qtl.outbred/', sep = ''))
outbred.data <- impo.prob(cross.data = data, stepsize = 5, path =
                          paste(R.home(), '/library/qtl.outbred/', sep = ''),
                          Grid = TRUE)

# ----- QTL scan using Haley-Knott regression and plotting
hk.model <- scanone(outbred.data, method = 'hk', pheno.col = 2)
plot(hk.model)

## End(Not run)
```

`plot.outbred` *Plot LOD curves for outbred data*

Description

Plot the LOD curve including the markers for a genome scan with the single-QTL model.

Usage

```
## S3 method for class 'outbred'
plot(x, old.data = data, chr = -1, gap = 25, ...)
```

Arguments

x	An object of class outbred, as output by outbred.
old.data	An object of class cross. See read.cross for details.
chr	Optional vector indicating the chromosomes to plot.
gap	Gap separating chromosomes (in cM).
...	Other arguments. See package qtl for details.

Details

This function allow the user to plot the results produced by scanone. The LOD scores for single or multiple chromosomes are plotted with the markers indicated.

See also plot.scanone for plotting the scan results but with the probabilities on the marker positions.

Value

None

Author(s)

Ronnie Nelson and Xia Shen

See Also

impo.prob, qtl.outbred-package, plot.scanone

Examples

```
## Not run:
# ----- impo.prob example
file <- paste(R.home(), '/library/qtl.outbred/example_data.csv', sep = '')
data <- read.cross('csv', file = file, genotypes = c('AA', 'AB', 'BB'))
path <- paste(R.home(), '/library/qtl.outbred/', sep = '')
outbred.data <- impo.prob(cross.data = data, path = path, stepsize = 5)

# ----- QTL scan using Haley-Knott regression and plotting
hk.model <- scanone(outbred.data, method = 'hk', pheno.col = 2)
plot(hk.model)

## End(Not run)
```

Index