Package 'statgenMPP'

October 29, 2024

Title QTL Mapping for Multi Parent Populations

Version 1.0.3

Date 2024-10-28

Description For Multi Parent Populations (MPP) Identity By Descend (IBD)
probabilities are computed using Hidden Markov Models. These probabilities
are then used in a mixed model approach for QTL Mapping as described in
Li et al. (<doi:10.1007/s00122-021-03919-7>).

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.6), statgenGWAS (>= 1.0.9)

Imports data.table, foreach, ggplot2, gridExtra, LMMsolver (>= 1.0.5),
Matrix, rlang, scales, spam, statgenIBD (>= 1.0.5), stringi

Suggests rmarkdown, knitr, tinytest

VignetteBuilder knitr

NeedsCompilation no

Author Martin Boer [aut] (<https://orcid.org/0000-0002-1879-4588>),
Wenhao Li [aut] (<https://orcid.org/0000-0001-5719-5775>),
Bart-Jan van Rossum [aut, cre]
(<https://orcid.org/0000-0002-8673-2514>),
Fred van Eeuwijk [ctb] (<https://orcid.org/0000-0003-3672-2921>)

Maintainer Bart-Jan van Rossum <bart-jan.vanrossum@wur.nl>

Repository CRAN

Date/Publication 2024-10-29 07:40:02 UTC

Contents

barleyMQM   *Pre-computed MQM output barley*

Description

Pre-computed MQM output for barley data used in vignette.

Usage

```
barleyMQM
```

Format

An object of class QTLMPP (inherits from GWAS) of length 5.

barleyPheno   *Phenotypic data for awn length in barley*

Description

Phenotypic data for awn length in barley

Usage

```
barleyPheno
```

Format

A data.frame with 916 rows and 3 columns.

genotype Genotype

Awn_length Awn_length in mm.

cross The cross the genotype belongs to.

References

Fine mapping of a major QTL for awn length in barley using a multiparent mapping population. Liller CB, Walla A, Boer MP, Hedley P, Macaulay M, Effgen S, von Korff M, van Esse GW, Koorn-neef M. Theor Appl Genet. 2017 Feb;130(2):269-281. doi:10.1007/s001220162807y.

---

calcIBDMPP	*IBD calculation for multi parental populations*

Description

IBD calculation for multi parental populations. Per cross IBD probabilities are calculated using `calcIBD` in the statgenIBD package. These probabilities are combined with optional phenotypic data and stored in a single object of class `gDataMPP`.

Usage

```
calcIBDMPP(
  crossNames,
  markerFiles,
  pheno,
  popType,
  mapFile,
  evalDist,
  grid = TRUE,
  verbose = FALSE
)
```

Arguments

crossNames	A character vector, the names of the crosses.
markerFiles	A character vector indicating the locations of the files with genotypic information for the populations. The files should be in tab-delimited format with a header containing marker names.
pheno	A data.frame or a list of data.frames with phenotypic data, with genotypes in the first column genotype and traits in the following columns. The trait columns should be numerical columns only. A list of data.frames can be used for replications, i.e. different trials.
popType	A character string indicating the type of population. One of DH, Fx, FxDH, BCx, BCxDH, BC1Sx, BC1SxDH, C3, C3DH, C3Sx, C3SxDH, C4, C4DH, C4Sx, C4SxDH (see Details).
mapFile	A character string indicating the location of the map file for the population. The file should be in tab-delimited format. It should consist of exactly three columns, marker, chromosome and position. There should be no header. The positions in the file should be in centimorgan.
evalDist	A numeric value, the maximum distance in cM between evaluation points.

grid	Should the extra markers that are added to assure the a maximum distince of `evalDist` be on a grid (`TRUE`) or in between marker existing marker positions (`FALSE`).
verbose	Should progress be printed?

Details

IBD probabilities can be calculated for many different types of populations. In the following table all supported populations are listed. Note that the value of x in the population types is variable, with its maximum value depicted in the last column.

Population type	Cross	Description	max. x
DH	biparental	doubled haploid population	
Fx	biparental	Fx population (F1, followed by x-1 generations of selfing)	8
FxDH	biparental	Fx, followed by DH generation	8
BCx	biparental	backcross, second parent is recurrent parent	9
BCxDH	biparental	BCx, followed by DH generation	9
BC1Sx	biparental	BC1, followed by x generations of selfing	7
BC1SxDH	biparental	BC1, followed by x generations of selfing and DH	6
C3	three-way	three way cross: (AxB) x C	
C3DH	three-way	C3, followed by DH generation	
C3Sx	three-way	C3, followed by x generations of selfing	7
C3SxDH	three-way	C3, followed by x generations of selfing and DH generation	6
C4	four-way	four-way cross: (AxB) x (CxD)	
C4DH	four-way	C4, followed by DH generation	
C4Sx	four-way	C4, followed by x generations of selfing	6
C4SxDH	four-way	C4, followed by x generations of selfing and DH generation	6

Value

An object of class `gDataMPP` with the following components:

map	a data.frame containing map data. Map is sorted by chromosome and position.
markers	a 3D matrix containing IBD probabilities.
pheno	data.frame or list of data.frames containing phenotypic data.
kinship	a kinship matrix.
covar	a data.frame with extra covariates (including the name of the cross).

Examples

```
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"



## Compute IBD probabilities for simulated population - AxB, AxC.
```

```
ABC <- calcIBDMPP(crossNames = c("AxB", "AxC"),
                  markerFiles = c(system.file("extdata/multipop", "AxB.txt",
                                              package = "statgenMPP"),
                                  system.file("extdata/multipop", "AxC.txt",
                                              package = "statgenMPP")),
                  pheno = pheno,
                  popType = "F4DH",
                  mapFile = system.file("extdata/multipop", "mapfile.txt",
                                        package = "statgenMPP"),
                  evalDist = 5)

summary(ABC)
```

---

createGDataMPP   *Create an object of class gDataMPP*

Description

Function for creating an object of class gDataMPP from an object of class IBDprob (computed or imported using statgenIBD) and phenotypic data.

Usage

```
createGDataMPP(IBDprob, pheno)
```

Arguments

IBDprob   An object of class IBDprob.

pheno   A data frame with at least columns "genotype" for the "genotype" and one or more numerical columns containing phenotypic information. A column "cross" can be used for indicating the cross the genotype comes from. This column is ignored if the IBDprob has an attribute genoCross containing this information (the default behaviour).

Value

An object of class gDataMPP

Examples

```
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"

## Compute IBD probabilities for simulated population using statgenIBD - AxB
AB <- statgenIBD::calcIBD(popType = "F4DH",
```

```
                                 markerFile = system.file("extdata/multipop", "AxB.txt",
                                                          package = "statgenMPP"),
                                 mapFile = system.file("extdata/multipop", "mapfile.txt",
                                                       package = "statgenMPP"),
                                 evalDist = 5)
## Create an object of class gDataMPP from IBD probabilities and
## phenotypic data.
ABMPP <- createGDataMPP(IBDprob = AB, pheno = pheno)
```

---

kinshipIBD   *Compute kinship matrix for IBD probabilities*

Description

Compute a kinship matrix or a list of chromosome specific kinship matrices for a 3D array of IBD probabilities. The kinship matrix is computed by averaging ZZ^t over all markers, where Z is the genotype x parents matrix for the marker. If chrSpecific = TRUE chromosome specific kinship matrices are computed for each chromosome based only on the markers on all other chromosomes.

Usage

```
kinshipIBD(markers, map = NULL, chrSpecific = TRUE)
```

Arguments

markers   An n x m x p array with IBD probabilities with genotypes in the rows (n), markers in the columns (m), and parents in the 3rd dimension (p).

map   A data.frame with columns chr for chromosome and pos for position. Positions should be in centimorgan (cM). They should not be cumulative over the chromosomes. Other columns are ignored. Marker names should be in the row names. These should match the marker names in the input file. Only required if chrSpecific = TRUE.

chrSpecific   Should chromosome specific kinship matrices be computed?

Value

A kinship matrix or a list of chromosome specific kinship matrices.

Examples

```
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"
```

```
## Compute IBD probabilities for simulated population - AxB, AxC.
ABC <- calcIBDMPP(crossNames = c("AxB", "AxC"),
                  markerFiles = c(system.file("extdata/multipop", "AxB.txt",
                                              package = "statgenMPP"),
                                  system.file("extdata/multipop", "AxC.txt",
                                              package = "statgenMPP")),
                  pheno = pheno,
                  popType = "F4DH",
                  mapFile = system.file("extdata/multipop", "mapfile.txt",
                                        package = "statgenMPP"),
                  evalDist = 5)

## Compute chromosome specific kinship matrices.
KChrSpec <- kinshipIBD(markers = ABC$markers, map = ABC$map)
```

---

maizeMQM	*Pre-computed MQM output maize*

Description

Pre-computed MQM output for maize data used in vignette.

Usage

```
maizeMQM
```

Format

An object of class QTLMPP (inherits from GWAS) of length 5.

---

maizeSQM	*Pre-computed SQM output maize*

Description

Pre-computed SQM output for maize data used in vignette.

Usage

```
maizeSQM
```

Format

An object of class QTLMPP (inherits from GWAS) of length 5.

plot.gDataMPP   *Plot function for the class* gDataMPP

Description

Creates a plot of an object of S3 class gDataMPP. The following types of plot can be made:

genMap A plot of the genetic map.

allGeno A plot showing for all genotypes the IBD probabilities of the parent with the highest probability per marker.

singleGeno A plot for a single genotype showing the IBD probabilities for all parents across the genome.

pedigree A plot showing the structure of the pedigree of the population.

See the respective sections for more details on the plots.

Usage

```
## S3 method for class 'gDataMPP'
plot(
  x,
  ...,
  plotType = c("genMap", "allGeno", "singleGeno", "pedigree"),
  genotype = NULL,
  title = NULL,
  output = TRUE
)
```

Arguments

x	An object of class gDataMPP.
...	Further arguments to be passed on to the actual plotting functions.
plotType	A character string indicating the type of plot to be made. One of "genMap", "singleGeno", "allGeno" or "pedigree".
genotype	A character string indicating the genotype for which the plot should be made. Only for plotType = "singleGeno".
title	A character string, the title of the plot.
output	Should the plot be output to the current device? If FALSE, only a ggplot object is invisibly returned.

Value

A ggplot object is invisibly returned.

genMap

> A plot is made showing the lengths of the chromosomes and the position of the markers that are present in the map. It is possible to highlight one or more markers using the extra parameter `highlight`.

allGeno

> A plot is made showing all genotypes and markers. Each combination of genotype and marker is colored according to the parent with the highest probability. A darker color indicates a higher probability.

singleGeno

> A plot is made for a single genotype, specified by `genotpye = "name_of_genotype"` showing the IBD probabilities for the selected genotype for all parents across the genome.

pedigree

> A plot is made showing the structure of the pedigree for the population in the `gDataMPP` object.

Examples

```
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"



## Compute IBD probabilities for simulated population - AxB, AxC.
ABC <- calcIBDMPP(crossNames = c("AxB", "AxC"),
                  markerFiles = c(system.file("extdata/multipop", "AxB.txt",
                                              package = "statgenMPP"),
                                  system.file("extdata/multipop", "AxC.txt",
                                              package = "statgenMPP")),
                  pheno = pheno,
                  popType = "F4DH",
                  mapFile = system.file("extdata/multipop", "mapfile.txt",
                                        package = "statgenMPP"),
                  evalDist = 5)

## Plot the genetic map.
plot(ABC, plotType = "genMap")

## Plot the genetic map and highlight marker EXT_3_30.
plot(ABC, plotType = "genMap", highlight = "EXT_3_30")

## Plot the IBD probabilities across the genome for all genotypes.
plot(ABC, plotType = "allGeno")

## Plot the IBD probabilities for genotype AxB0001.
```

```
plot(ABC, plotType = "singleGeno", genotype = "AxB0001")

## Plot the pedigree.
plot(ABC, plotType = "pedigree")
```

---

plot.QTLMPP   *Plot function for the class* QTLMPP

Description

Creates a plot of an object of S3 class QTLMPP. The following types of plot can be made:

QTLProfile A QTL profile plot, i.e. a plot of $-log10(p)$ values per marker.

parEffs A plot of effect sizes and directions per parent.

QTLRegion A plot highlighting the QTLs found on the genetic map.

QTLProfileExt A combination of the QTL profile and parental effects plotted above each other.

See details for a detailed description of the plots and the plot options specific to the different plots.

Usage

```
## S3 method for class 'QTLMPP'
plot(
  x,
  ...,
  plotType = c("QTLProfile", "parEffs", "QTLRegion", "QTLProfileExt"),
  title = NULL,
  output = TRUE
)
```

Arguments

x   An object of class QTLMPP.

...   further arguments to be passed on to the actual plotting functions.

plotType   A character string indicating the type of plot to be made. One of "QTLProfile",
                    "parEffs", and "QTLRegion".

title   A character string, the title of the plot.

output   Should the plot be output to the current device? If FALSE, only a ggplot object
                    is invisibly returned.

QTL Profile plot

A profile of all marker positions and corresponding $-log10(p)$ values is plotted. QTLs found are highlighted with red dots. The threshold is plotted as a horizontal line. If there are previously known marker effects, false positives and true negatives can also be marked.
Extra parameter options:

xLab A character string, the x-axis label. Default = `"Chromosomes"`

yLab A character string, the y-axis label. Default = `-log10(p)`

effects A character vector, indicating which markers correspond to a real (known) effect. Used for determining true/false positives and false negatives. True positives are colored green, false positives orange and false negatives yellow.

colPalette A color palette used for plotting. Default coloring is done by chromosome, using black and grey.

signLwd A numerical value giving the thickness of the points that are false/true positives/negatives. Default = 0.6

chr A vector of chromosomes to be plotted. By default, all chromosomes are plotted. Using this option allows restricting the plot to a subset of chromosomes.

Parental effects Plot

A plot of effect sizes for each of the parents for the QTLs found is created.
Extra parameter options:

xLab A character string, the x-axis label. Default = `"Chromosomes"`

yLab A character string, the y-axis label. Default = `"Parents"` #'

chr A vector of chromosomes to be plotted. By default, all chromosomes are plotted. Using this option allows restricting the plot to a subset of chromosomes.

QTL Region Plot

A plot of effect is created on which the positions of the QTLs found in the QTL detection are highlighted in red.
No extra parameter options.

Extended QTL Profile Plot

An extended version of the QTL Profile Plot, in which the QLT profile plot is combined with the parental effect plot to make it easier to assess the effects for each specific QTL found.
Extra parameter options:

chr A vector of chromosomes to be plotted. By default, all chromosomes are plotted. Using this option allows restricting the plot to a subset of chromosomes.

Examples

```
## Not run:
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"


## Compute IBD probabilities for simulated population - AxB, AxC.
ABC <- calcIBDMPP(crossNames = c("AxB", "AxC"),
                  markerFiles = c(system.file("extdata/multipop", "AxB.txt",
                                              package = "statgenMPP"),
                                  system.file("extdata/multipop", "AxC.txt",
                                              package = "statgenMPP")),
                  pheno = pheno,
                  popType = "F4DH",
                  mapFile = system.file("extdata/multipop", "mapfile.txt",
                                        package = "statgenMPP"),
                  evalDist = 5)

## Multi-QTL Mapping.
ABC_MQM <- selQTLMPP(ABC, trait = "yield")

## QTL Profile plot.
plot(ABC_MQM, plotType = "QTLProfile")

## Plot of parental effects for QTLs found.
plot(ABC_MQM, plotType = "parEffs")

## Plot of genetic map highlighting positions of QTLs found.
plot(ABC_MQM, plotType = "QTLRegion")

## Extended QTL Profile plot.
plot(ABC_MQM, plotType = "QTLProfileExt")

## End(Not run)
```

---

print.summary.QTLMPP   *Print summary of object of class* summary.QTLMPP

Description

Print summary of object of class summary.QTLMPP

Usage

```
## S3 method for class 'summary.QTLMPP'
print(x, ...)
```

Arguments

x	An object of class `summary.QTLMPP`.
...	Not used.

---

readRABBITMPP	*Read IBD probabilities*

Description

Read a file with IBD probabilities computed by the RABBIT software package. It is possible to additionally read the pedigree file that is also used by RABBIT. Reading this file allows for plotting the pedigree. Phenotypic data can be added from a data.frame.

Usage

```
readRABBITMPP(infile, pedFile = NULL, pheno = NULL)
```

Arguments

infile	A character string, a link to a .csv file with IBD probabilities.
pedFile	A character string, a link to a .csv file with pedigree information as used by RABBIT as input.
pheno	A data frame with at least columns "genotype" for the "genotype" and one or more numerical columns containing phenotypic information. A column "cross" can be used for indicating the cross the genotype comes from.

Value

An object of class `gDataMPP` with map and markers corresponding to the imported information in the imported .csv file.

References

Zheng, Chaozhi, Martin P Boer, and Fred A Van Eeuwijk. "Recursive Algorithms for Modeling Genomic Ancestral Origins in a Fixed Pedigree." G3 Genes|Genomes|Genetics 8 (10): 3231–45. https://doi.org/10.1534/G3.118.200340.

Examples

```
## Read RABBIT data for barley.
genoFile <- system.file("extdata/barley", "barley_magicReconstruct.zip",
                        package = "statgenMPP")
barleyMPMPP <- readRABBITMPP(unzip(genoFile, exdir = tempdir()))
```

---

selQTLMPP   *Multi round genome scans for QTL detection*

Description

Multi round genome scans for QTL detection.

Several rounds of QTL detection are performed. First a model is fitted without cofactors. If for at least one marker the $-log10(p)$ value is above the threshold the marker with the lowest p-Value is added as cofactor in the next round of QTL detection. This process continues until there are no new markers with a $-log10(p)$ value above the threshold or until the maximum number of cofactors is reached.

Usage

```
selQTLMPP(
  MPPobj,
  trait = NULL,
  QTLwindow = 10,
  threshold = 3,
  maxCofactors = NULL,
  K = NULL,
  computeKin = FALSE,
  parallel = FALSE,
  verbose = FALSE
)
```

Arguments

MPPobj	An object of class gDataMPP, typically the output of either [calcIBDMPP](#) or [readRABBITMPP](#).
trait	A character string indicating the trait QTL mapping is done for.
QTLwindow	A numerical value indicating the window around a QTL that is considered as part of that QTL.
threshold	A numerical value indicating the threshold for the $-log10p$ value of a marker to be considered a QTL.
maxCofactors	A numerical value, the maximum number of cofactors to include in the model. If NULL cofactors are added until no new cofactors are found.
K	A list of chromosome specific kinship matrices. If NULL and computeKin = FALSE no kinship matrix is included in the models.
computeKin	Should chromosome specific kinship matrices be computed?
parallel	Should the computation of variance components be done in parallel? This requires a parallel back-end to be registered. See examples.
verbose	Should progress and intermediate plots be output?

Details

By default only family specific effects and residual variances and no kinship relations are included in the model. It is possible to include kinship relations by either specifying computeKin = TRUE. When doing so the kinship matrix is computed by averaging ZZ^t over all markers, where Z is the genotype x parents matrix for the marker. It is also possible to specify a list of precomputed chromosome specific kinship matrices in K. Note that adding a kinship matrix to the model increases the computation time a lot, especially for large populations.

Value

An object of class QTLMPP

See Also

kinshipIBD

Examples

```
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"


## Compute IBD probabilities for simulated population - AxB, AxC.
ABC <- calcIBDMPP(crossNames = c("AxB", "AxC"),
                  markerFiles = c(system.file("extdata/multipop", "AxB.txt",
                                              package = "statgenMPP"),
                                  system.file("extdata/multipop", "AxC.txt",
                                              package = "statgenMPP")),
                  pheno = pheno,
                  popType = "F4DH",
                  mapFile = system.file("extdata/multipop", "mapfile.txt",
                                        package = "statgenMPP"),
                  evalDist = 5)

## Single-QTL Mapping.
ABC_SQM <- selQTLMPP(ABC, trait = "yield", maxCofactors = 0)

## Multi-QTL Mapping.
## Not run:
## Register parallel back-end with 2 cores.
doParallel::registerDoParallel(cores = 2)

## Run multi-QTL mapping.
ABC_MQM <- selQTLMPP(ABC, trait = "yield", parallel = TRUE)

## Run multi-QTL mapping - include kinship matrix.
ABC_MQM_kin <- selQTLMPP(ABC, trait = "yield", parallel = TRUE,
                         computeKin = TRUE)
```

```
## End(Not run)
```

---

summary.gDataMPP   *Summary function for the class* gDataMPP

Description

Gives a summary for an object of S3 class gDataMPP.

Usage

```
## S3 method for class 'gDataMPP'
summary(object, ..., trials = NULL)
```

Arguments

object	An object of class gDataMPP.
...	Not used.
trials	A vector of trials to include in the summary. These can be either numeric indices or character names of list items in pheno. If NULL, all trials are included.

Value

A list with a most four components:

mapSum A list with number of markers and number of chromosomes in the map.

markerSum A list with number of markers, number of genotypes and the distribution of the values within the markers.

phenoSum A list of data.frames, one per trial with a summary of all traits within the trial.

covarSum A list of data.frames, one per trial with a summary of all covariates within the trial.

All components are only present in the output if the corresponding content is present in the gDataMPP object.

summary.QTLMPP *Summary function for the class* QTLMPP

Description

Gives a summary for an object of S3 class QTLMPP.

Usage

```
## S3 method for class 'QTLMPP'
summary(object, ...)
```

Arguments

object   An object of class QTLMPP.

...   Not used.

Examples

```
## Not run:
## Read phenotypic data.
pheno <- read.delim(system.file("extdata/multipop", "AxBxCpheno.txt",
                                package = "statgenMPP"))
## Rename first column to genotype.
colnames(pheno)[1] <- "genotype"

## Compute IBD probabilities for simulated population - AxB, AxC.
ABC <- calcIBDMPP(crossNames = c("AxB", "AxC"),
                  markerFiles = c(system.file("extdata/multipop", "AxB.txt",
                                              package = "statgenMPP"),
                                  system.file("extdata/multipop", "AxC.txt",
                                              package = "statgenMPP")),
                  pheno = pheno,
                  popType = "F4DH",
                  mapFile = system.file("extdata/multipop", "mapfile.txt",
                                        package = "statgenMPP"),
                  evalDist = 5)

## Multi-QTL Mapping.
ABC_MQM <- selQTLMPP(ABC, trait = "yield")

## Print summary.
summary(ABC_MQM)

## End(Not run)
```

Index