Package 'QTLEMM'

September 8, 2022

Type Package

Title QTL Mapping and Hotspots Detection

Version 1.1.2

Description For QTL mapping, it consists of several functions to perform various tasks, including simulating or analyzing data, computing the significance thresholds and visualizing the QTL mapping results. The single-QTL or multiple-QTL method that allows a host of statistical models to be fitted and compared is applied to analyze the data for the estimation of QTL parameters. The models include the linear regression, permutation test, normal mixture model and truncated normal mixture model. The Gaussian stochastic process is implemented to compute the significance thresholds for QTL detection onto a genetic linkage map in the experimental populations. Two types of data, the complete genotyping or selective genotyping data, from various experimental populations, including backcross, F2, recombinant inbred (RI) populations, advanced intercrossed (AI) populations, are considered in the QTL mapping analysis. For QTL hotpot detection, the statistical methods can be developed based on either using the individual-level data or using the summarized data. We have proposed a statistical framework that can handle both the individual-level data and summarized QTL data for QTL hotspots detection. Our statistical framework can overcome the underestimation of threshold arising from ignoring the correlation structure among traits, and also identify the different types of hotspots with very low computational cost during the detection process. Here, we attempt to provide the R codes of our QTL mapping and hotspot detection methods for general use in genes, genomics and genetics studies. The QTL mapping methods for the complete and selective genotyping designs are based on the multiple interval mapping (MIM) model proposed by Kao, C.-H. , Z.-B. Zeng and R. D. Teasdale (1999) <doi:10.1534/genetics.103.021642> and H.-I Lee, H.-A. Ho and C.-H. Kao (2014) <doi:10.1534/genetics.114.168385>, respectively. The QTL hotspot detection analysis is based on the method by Wu, P.-Y., M.-.H. Yang, and C.-H. Kao (2021) <doi:10.1093/g3journal/jkab056>.

Imports mvtnorm, utils, stats, graphics, grDevices

URL https://github.com/py-chung/QTLEMM

BugReports https://github.com/py-chung/QTLEMM/issues

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Ping-Yuan Chung [cre],
Chen-Hung Kao [aut]

Maintainer Ping-Yuan Chung <pychung@webmail.stat.sinica.edu.tw>

Repository CRAN

Date/Publication 2022-09-08 09:42:55 UTC

R topics documented:

`D.make` *Generate D Matrix*

Description

Generate the genetic design matrix of specified QTL number and effects.

Usage

```
D.make(nQTL, type = "RI", a = TRUE, d = TRUE, aa = 0, dd = 0, ad = 0)
```

Arguments

`nQTL`	integer. The number of QTLs.
`type`	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").

a	integer or vector. A integer or vecter to decide the additive effects of which QTL will be considered in this design matrix. If a=TRUE, the additive effect of all QTLs will be considered. If a=0, no additive effect will be considered.
d	integer or vector. A integer or vecter to decide the dominant effects of which QTL will be considered in this design matrix. If d=TRUE, the dominant effect of all QTLs will be considered.If d=0, no dominant effect will be considered.
aa	vector or matrix. The additive-by-additive interaction. Tow format can be used in this parameter. One format is vector, in which every two elements indicate a combination of additive-by-additive interaction. The other format is a 2*i matrix, where i is the number of combination of interaction, and each column indicates the two interacting QTL. Besides, if aa=TRUE, all combinations of additive-by-additive interaction will be considered. If aa=0, no additive-by-additive interaction will be considered.
dd	vector or matrix. The dominant-by-dominant interaction. The format is the same as that in aa.
ad	vector or matrix. The additive-by-dominant interaction. The format is the same as that in aa. Note that, in each pair of QTLs, the first element indicates the additive effect, and the second element indicates the dominant effect.

Value

The genetic design matrix, whose elements are the coded variables of the QTL effects. it is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects in the MIM model.

Note

For parameter type, if type="BC", the design matrix contain only additive effect and additive by additive interaction. If type="AI" or type="RI", that will contain additive and dominance effects and all interaction.

For example of parameter aa, when aa=c(1,3,2,4,5,6), indicates that the interaction between QTL1 and QTL3, the interaction between QTL2 and QTL4, and that between QTL5 and QTL6 will be considered in the design matrix. Beside, the matrix format can expressed as aa=matrix(c(1,3,2,4,5,6),2,3). The parameters DD and AD are also expressed in the same way.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

Examples

```
D.make(4, d = c(1,3,4), aa = c(1,2,2,3), dd = c(1,3,1,4), ad = c(1,2,2,1,2,3,3,4))
D.make(5, type = "BC", a = c(1,3,4,5), aa = c(1,2,3,4,4,5))
```

EM.MIM *EM Algorithm for QTL MIM*

Description

Expectation-maximization algorithm for QTL multiple interval mapping.

Usage

```
EM.MIM(
  D.matrix,
  cp.matrix,
  y,
  E.vector0 = NULL,
  X = NULL,
  beta0 = NULL,
  variance0 = NULL,
  conv = 10^-5,
  console = TRUE
)
```

Arguments

D.matrix	matrix. The design matrix of QTL effects which is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects considered in the MIM model. The design matrix can be easily generated by the function D.make().
cp.matrix	matrix. The conditional probability matrix which is an n*g matrix, where n is the number of individuals, and g is the number of possible genotypes of QTLs. The conditional probability matrix can be easily generated by the function Q.make().
y	vector. An vector with n elements that contains the phenotype values of individuals.
E.vector0	vevtor. The initial value for QTL effects. The number of elements corresponds to the column dimension of the design matrix. If E.vector0=NULL, the initial value will be 0 for all effects.
X	matrix. The design matrix of the fixed factors except QTL effects. It is an n*k matrix, where n is the number of individuals, and k is the number of fixed factors. If X=NULL, the matrix will be an n*1 matrix that all elements are 1.
beta0	vector. The initial value for effects of the fixed factors. The number of elements corresponds the column dimension of the fixed factor design matrix. If beta0=NULL, the initial value will be the average of y.
variance0	numeric. The initial value for variance. If variance0=NULL, the initial value will be the variance of phenotype values.
conv	numeric. The convergence criterion of EM algorithm. The E and M steps will be iterated until a convergence criterion is satisfied.

console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

E.vector	The QTL effects calculated by EM algorithm.
beta	The effects of the fixed factors calculated by EM algorithm.
variance	The error variance calculated by EM algorithm.
PI.matrix	The posterior probabilities matrix after the process of EM algorithm.
log.likelihood	The log likelihood value of this model.
LRT	The LRT statistic of this model.
R2	The coefficient of determination of this model. This can be used as an estimate of heritability.
y.hat	The fitted values of trait values calculated by the estimated values from the EM algorithm.
iteration.time	The iteration time of EM algorithm.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

See Also

D.make Q.make EM.MIM2

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
D.matrix <- D.make(3, type = "RI", aa = c(1, 3, 2, 3), dd = c(1, 2, 1, 3), ad = c(1, 2, 2, 3))
cp.matrix <- Q.make(QTL, marker, geno, type = "RI", ng = 2)$cp.matrix
result <- EM.MIM(D.matrix, cp.matrix, y)
result$E.vector
```

EM.MIM2 *EM Algorithm for QTL MIM with Selective Genotyping*

Description

Expectation-maximization algorithm for QTL multiple interval mapping. This function can handle the genotype data witch is selective genotyping.

Usage

```
EM.MIM2(
  QTL,
  marker,
  geno,
  D.matrix,
  cp.matrix = NULL,
  y,
  yu = NULL,
  sele.g = "n",
  tL = NULL,
  tR = NULL,
  type = "RI",
  ng = 2,
  cM = TRUE,
  E.vector0 = NULL,
  X = NULL,
  beta0 = NULL,
  variance0 = NULL,
  conv = 10^-5,
  console = TRUE
)
```

Arguments

QTL	matrix. A q*2 matrix contains the QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.

`D.matrix` matrix. The design matrix of QTL effects which is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects considered in the MIM model. The design matrix can be easily generated by the function D.make().

`cp.matrix` matrix. The conditional probability matrix which is an n*g matrix, where n is the number of individuals which are genotyped, and g is the number of possible genotypes of QTLs. If cp.matrix=NULL, the function will calculate the conditional probability matrix for selective genotyping.

`y` vector. A vector that contains the phenotype values of individuals with genotyped.

`yu` vector. A vector that contains the phenotype value of the individuals without genotyped.

`sele.g` character. If sele.g="n", it will consider that the data is not a selective genotyping data but the complete genotyping data. If sele.g="p", it will consider that the data is a selective genotyping data, and use the proposed model (Lee 2014) to analyze. If sele.g="t", it will consider that the data is a selective genotyping data, and use the truncated model (Lee 2014) to analyze. If sele.g="f, it will consider that the data is a selective genotyping data, and use the frequency-based model (Lee 2014) to analyze. Note that the yu must be input when sele.g="p" of "f".

`tL` numeric. The lower truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tL=NULL, the yu must be input and the function will consider the minimum of yu as the lower truncation point.

`tR` numeric. The upper truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tR=NULL, the yu must be input and the function will consider the maximum of yu as the upper truncation point.

`type` character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").

`ng` integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.

`cM` logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.

`E.vector0` vevtor. The initial value for QTL effects. The number of elements corresponds to the column dimension of the design matrix. If E.vector0=NULL, the initial value will be 0 for all effects.

`X` matrix. The design matrix of the fixed factors except QTL effects. It is an n*k matrix, where n is the number of individuals, and k is the number of fixed factors. If X=NULL, the matrix will be an n*1 matrix that all elements are 1.

`beta0` vector. The initial value for effects of the fixed factors. The number of elements corresponds the column dimension of the fixed factor design matrix. If beta0=NULL, the initial value will be the average of y.

`variance0` numeric. The initial value for variance. If variance0=NULL, the initial value will be the variance of phenotype values.

conv	numeric. The convergence criterion of EM algorithm. The E and M steps will be iterated until a convergence criterion is satisfied.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

QTL	The QTL imformation of this analysis.
E.vector	The QTL effects calculated by EM algorithm.
beta	The effects of the fixed factors calculated by EM algorithm.
variance	The variance calculated by EM algorithm.
PI.matrix	The posterior probabilities matrix after the process of EM algorithm.
log.likelihood	The log likelihood value of this model.
LRT	The LRT statistic of this model.
R2	The coefficient of determination of this model. This can be used as an estimate of heritability.
y.hat	The fitted values of trait values with genotyping calculated by the estimated values from the EM algorithm.
yu.hat	The fitted values of trait values without genotyping calculated by the estimated values from the EM algorithm.
iteration.time	The iteration time of EM algorithm.
model	The model of this analysis, which contains complete genotyping model, proposed model, truncated model, and frequency-based model.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

H.-I LEE, H.-A. HO and C.-H. KAO 2014 A new simple method for improving QTL mapping under selective genotyping. Genetics 198: 1685-1698.

See Also

D.make Q.make EM.MIM

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# make the seletive genotyping data
ys <- y[y > quantile(y)[4] | y < quantile(y)[2]]
```

```
yu <- y[y >= quantile(y)[2] & y <= quantile(y)[4]]
geno.s <- geno[y > quantile(y)[4] | y < quantile(y)[2],]

# run and result
D.matrix <- D.make(3, type = "RI", aa = c(1, 3, 2, 3), dd = c(1, 2, 1, 3), ad = c(1, 2, 2, 3))
result <- EM.MIM2(QTL, marker, geno.s, D.matrix, y = ys, yu = yu, sele.g = "p")
result$E.vector
```

`EQF.permu` *EQF Permutation*

Description

The EQF matrix cluster permutation process for QTL hotspot detection.

Usage

```
EQF.permu(
  LOD.QTLdetect.result,
  ptime = 1000,
  alpha = 0.05,
  Q = TRUE,
  console = TRUE
)
```

Arguments

`LOD.QTLdetect.result`	list. The data list of the output from LOD.QTLdetect.
`ptime`	integer. The permutation time.
`alpha`	numeric. The type 1 error rate of detecting the hotspot.
`Q`	logical. If being TURE, the function will further be carrying out the population of Q method as the control group and shown as B in the output.
`console`	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

`EQF.matrix`	The matrix denote the EQF value of each bin.
`bin`	The bin infromation matrix used in this analyze.
`LOD.threshole`	The LOD threshold used in this analyze.
`cluster.number`	The number of QTLs in each cluster group.
`cluster.id`	The serial number of traits in each cluster group.
`cluster.matrix`	The new EQF matrix from the clustering process.
`permu.matrix.cluster`	the permutation result of the clustering method which has been sorted by order.
`permu.matrix.Q`	The permutation result of the Q method which has been sorted by order.
`EQF.threshold`	The EQF threshold calculated from the permutation process.

References

Wu, P.-Y., M.-.H. Yang, and C.-H. KAO 2021 A Statistical Framework for QTL Hotspot Detection. G3: Genes, Genomes, Genetics: jkab056.

See Also

LOD.QTLdetect EQF.plot

Examples

```
# load the example data
load(system.file("extdata", "LODexample.RDATA", package = "QTLEMM"))

# run and result
result <- EQF.permu(LOD.QTLdetect.result, ptime = 50)
result$cluster.number
```

EQF.plot *EQF plot*

Description

Depict the EQF plot by the result of permutation process to detect the QTL hotspot.

Usage

```
EQF.plot(result, plot.all = TRUE, plot.cr = TRUE)
```

Arguments

result	list. The data list of the output from LOD.QTLdetect() or EQF.permu().
plot.all	logical. If being TURE, output one figure of the EQF values over the bins.
plot.cr	logical. If being TURE, output the figures of the EQF values over the bins of each chromosome.

Value

One or several EQF plots.

References

Wu, P.-Y., M.-.H. Yang, and C.-H. KAO 2021 A Statistical Framework for QTL Hotspot Detection. G3: Genes, Genomes, Genetics: jkab056.

See Also

LOD.QTLdetect EQF.permu

Examples

```
# load the example data
load(system.file("extdata", "LODexample.RDATA", package = "QTLEMM"))

# run and result
EQF.plot(LOD.QTLdetect.result)
EQF.plot(EQF.permu.result)
```

`IM.search` *QTL search by IM*

Description

Expectation-maximization algorithm for QTL interval mapping to search the possible position of QTL in all chromosome.

Usage

```
IM.search(
  marker,
  geno,
  y,
  method = "EM",
  type = "RI",
  D.matrix = NULL,
  ng = 2,
  cM = TRUE,
  speed = 1,
  conv = 10^-5,
  d.eff = FALSE,
  LRT.thre = TRUE,
  simu = 1000,
  alpha = 0.05,
  detect = TRUE,
  QTLdist = 15,
  plot.all = TRUE,
  plot.cr = TRUE,
  console = TRUE
)
```

Arguments

`marker`	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.

geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
y	vector. A vector with n elements that contains the phenotype values of individuals.
method	character. method="EM" means the interval mapping method by Lander and Botstein (1989) is used in the analysis, while method="REG" means the approximate regression interval mapping method by Haley and Knott (1992) is considered in the analysis.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
D.matrix	matrix. The design matrix of the IM model. If D.matrix=NULL, the design matrix will be the constructed using the Cockerham's model. In BC population, it is a 2*1 matrix which is 0.5, -0.5 for additive effect. In RI or AI population, it is a 3*2 matrix whose first column is 1, 0, -1 for additive effect and second column is 0.5, -0.5, 0.5 for dominant effect.
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
speed	numeric. The walking speed of the QTL search (in cM).
conv	numeric. The convergent criterion of EM algorithm. The E and M steps will be iterated until a convergent criterion is satisfied.
d.eff	logical. Specify whether the dominant effect will be considered in the parameter estimation or not for AI or RI population.
LRT.thre	logical or numeric. If being TRUE, the LRT threshold will be computed based on the Gaussian stochastic process (Kao and Ho 2012). Or users can input a numerical value as the LRT threshold to assessing the significance of QTL detection.
simu	integer. To decide how many simulation samples will be used to compute the LRT threshold using the Gaussian process.
alpha	numeric. The type I error rate for the LRT threshold.
detect	logical. Whether the significant QTL whose LRT statistic is larger than the LRT threshold will be shown in the output dataset or not.
QTLdist	numeric. The minimum distance (cM) among different linked significant QTL.
plot.all	logical. If being TRUE, output the profile of LRT statistics for the genome in one figure.
plot.cr	logical. If being TRUE, output the profile of LRT statistics for the chromosomes.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

`effect`	The estimated effects and LRT statistics of all positions.
`LRT.threshold`	The LRT threshold value computed for the data using the Gaussian stochastic process (Kuo 2011; Kao and Ho 2012).
`detect.QTL`	The positions, effects and LRT statistics of the detected QTL significant using the obtained LRT threshold value.

Graphical outputs including LOD value and effect of each position.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

KAO, C.-H. and H.-A. Ho 2012 A score-statistic approach for determining threshold values in QTL mapping. Frontiers in Bioscience. E4, 2670-2682.

See Also

`EM.MIM` `IM.search2` `LRTthre`

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
result <- IM.search(marker, geno, y, type = "RI", ng = 2, speed = 7.5, conv = 10^-3, LRT.thre = 10)
result$detect.QTL
```

`IM.search2` *QTL search by IM with Selective Genotyping*

Description

Expectation-maximization algorithm for QTL interval mapping to search the possible position of QTL in all chromosome. This function can handle the genotype data witch is selective genotyping.

Usage

```
IM.search2(
  marker,
  geno,
  y,
  yu = NULL,
```

```
  sele.g = "n",
  tL = NULL,
  tR = NULL,
  method = "EM",
  type = "RI",
  D.matrix = NULL,
  ng = 2,
  cM = TRUE,
  speed = 1,
  conv = 10^-5,
  d.eff = FALSE,
  LRT.thre = TRUE,
  simu = 1000,
  alpha = 0.05,
  detect = TRUE,
  QTLdist = 15,
  plot.all = TRUE,
  plot.cr = TRUE,
  console = TRUE
)
```

Arguments

marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
y	vector. A vector that contains the phenotype values of individuals with genotyped.
yu	vector. A vector that contains the phenotype value of the individuals without genotyped.
sele.g	character. If sele.g="n", it will consider that the data is not a selective genotyping data but the complete genotyping data. If sele.g="p", it will consider that the data is a selective genotyping data, and use the proposed model (Lee 2014) to analyze. If sele.g="t", it will consider that the data is a selective genotyping data, and use the truncated model (Lee 2014) to analyze. If sele.g="f, it will consider that the data is a selective genotyping data, and use the frequency-based model (Lee 2014) to analyze. Note that the yu must be input when sele.g="p" of "f".
tL	numeric. The lower truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tL=NULL, the yu must be input and the function will consider the minimum of yu as the lower truncation point.
tR	numeric. The upper truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tR=NULL, the yu must be input and the function will consider the maximum of yu as the upper truncation point.

method	character. method="EM" means the interval mapping method by Lander and Botstein (1989) is used in the analysis, while method="REG" means the approximate regression interval mapping method by Haley and Knott (1992) is considered in the analysis.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
D.matrix	matrix. The design matrix of the IM model. If D.matrix=NULL, the design matrix will be the constructed using the Cockerham's model. In BC population, it is a 2*1 matrix which contains 0.5, -0.5 for additive effect. In RI or AI population, it is a 3*2 matrix which contains 1, 0, -1 for additive effect and 0.5, -0.5, 0.5 for dominant effect.
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
speed	numeric. The walking speed of the QTL search (in cM).
conv	numeric. The convergent criterion of EM algorithm. The E and M steps will be iterated until a convergent criterion is satisfied.
d.eff	logical. Specify whether the dominant effect will be considered in the parameter estimation or not for AI or RI population.
LRT.thre	logical or numeric. If being TRUE, the LRT threshold will be computed based on the Gaussian stochastic process (Kao and Ho 2012). Or users can input a numerical value as the LRT threshold to assessing the significance of QTL detection.
simu	integer. To decide how many simulation samples will be used to compute the LRT threshold using the Gaussian process.
alpha	numeric. The type I error rate for the LRT threshold.
detect	logical. Whether the significant QTL whose LRT statistic is larger than the LRT threshold will be shown in the output dataset or not.
QTLdist	numeric. The minimum distance (cM) among different linked significant QTL.
plot.all	logical. If being TRUE, output the profile of LRT statistics for the genome in one figure.
plot.cr	logical. If being TRUE, output the profile of LRT statistics for the chromosomes.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

effect	The estimated effects and LRT statistics of all positions.
LRT.threshold	The LRT threshold value computed for the data using the Gaussian stochastic process (Kuo 2011; Kao and Ho 2012).

detect.QTL	The positions, effects and LRT statistics of the detected QTL significant using the obtained LRT threshold value.
model	The model of selective genotyping data in this analyze.

Graphical outputs including LOD value and effect of each position.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

H.-I LEE, H.-A. HO and C.-H. KAO 2014 A new simple method for improving QTL mapping under selective genotyping. Genetics 198: 1685-1698.

KAO, C.-H. and H.-A. Ho 2012 A score-statistic approach for determining threshold values in QTL mapping. Frontiers in Bioscience. E4, 2670-2682.

See Also

EM.MIM2 IM.search LRTthre

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# make the seletive genotyping data
ys <- y[y > quantile(y)[4] | y < quantile(y)[2]]
yu <- y[y >= quantile(y)[2] & y <= quantile(y)[4]]
geno.s <- geno[y > quantile(y)[4] | y < quantile(y)[2],]

# run and result
result <- IM.search2(marker, geno.s, ys, yu, sele.g = "p", type = "RI", ng = 2,
speed = 7.5, conv = 10^-3, LRT.thre = 10)
result$detect.QTL
```

LOD.QTLdetect *QTL Detect by LOD*

Description

Detect QTL by likelihood of odds(LOD) matrix.

Usage

```
LOD.QTLdetect(LOD, bin, thre = 3, QTLdist = 20, console = TRUE)
```

Arguments

LOD	matrix. The LOD matrix which is a t*p matrix, where t is the number of traits and p is the number of bins on the chromosomes. The missing value should be NA.
bin	matrix. A n*2 matrix represent how many bins on each chromosome, where n is the number of chromosomes. The first column denotes the chromosome number, and the second column denote how many bins on that chromosome. Note that chromosome and must be divided in order.
thre	numeric. The LOD threshold. The LOD score under this threshold will be calculated as 0.
QTLdist	numeric. The minimum distance (bin) among different linked significant QTL.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

detect.QTL.number	The number of detected QTL in each trait.
QTL.matrix	The QTL position matrix. Where the elements 1 donate the position of QTL; elements 0 donate the bins whose LOD score is under the LOD threshold; other positions is shown as NA.
EQF.matrix	The matrix denote the EQF value of each bin.
linkage.QTL.number	The linkage QTL number of all detected QTL. In other words, it is the table denote how many QTL on one chromosome.
LOD.threshole	The LOD threshold used in this analyze.
bin	The bin infromation matrix used in this analyze.

References

Wu, P.-Y., M.-.H. Yang, and C.-H. KAO 2021 A Statistical Framework for QTL Hotspot Detection. G3: Genes, Genomes, Genetics: jkab056.

See Also

EQF.permu EQF.plot

Examples

```
# load the example data
load(system.file("extdata", "LODexample.RDATA", package = "QTLEMM"))
dim(LODexample) # 100 traits, 633 bins on chromosome

# run and result
result <- LOD.QTLdetect(LODexample, bin, thre = 3, QTLdist = 10)
result$detect.QTL.number
```

LRTthre *LRT Threshold*

Description

The LRT threshold for QTL interval mapping based on the Gaussian stochastic process (Kao and Ho 2012).

Usage

```
LRTthre(
  marker,
  type = "RI",
  ng = 2,
  cM = TRUE,
  ns = 200,
  gv = 25,
  speed = 1,
  simu = 1000,
  d.eff = FALSE,
  alpha = 0.05,
  console = TRUE
)
```

Arguments

marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
ns	integer. The number of individuals for generating the individual trait values. The changes in this values do not affect the outcome of the LRT threshold value significantly.
gv	numeric. The genetic variance for generating the individual trait values. The changes in this values do not affect the outcome of the LRT threshold value significantly.

speed	numeric. The walking speed of the QTL analysis (in cM).
simu	integer. To decide how many simulation samples will be used to compute the LRT threshold using the Gaussian process.
d.eff	logical. Specify whether the dominant effect will be considered in the parameter estimation or not for AI or RI population.
alpha	numeric. The type I error rate for the LRT threshold.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

The LRT threshold for QTL interval mapping.

References

KAO, C.-H. and H.-A. Ho 2012 A score-statistic approach for determining threshold values in QTL mapping. Frontiers in Bioscience. E4, 2670-2682.

See Also

rmvnorm

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
LRTthre(marker, type = "RI", ng = 2, speed = 2, simu = 60)
```

MIM.points *QTL Short Distance Correction by MIM*

Description

Expectation-maximization algorithm for QTL multiple interval mapping. Find the best QTL position near the designated QTL position.

Usage

```
MIM.points(
  QTL,
  marker,
  geno,
  y,
  method = "EM",
  type = "RI",
```

```
  D.matrix = NULL,
  ng = 2,
  cM = TRUE,
  scope = 5,
  speed = 1,
  conv = 10^-3,
  console = TRUE
)
```

Arguments

QTL	matrix. A q*2 matrix contains the QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
y	vector. An vector with n elements that contains the phenotype values of individuals.
method	character. method="EM" means the interval mapping method by Lander and Botstein (1989) is used in the analysis, while method="REG" means the approximate regression interval mapping method by Haley and Knott (1992) is considered in the analysis.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
D.matrix	matrix. The design matrix of QTL effects which is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects considered in the MIM model. The design matrix can be easily generated by the function D.make(). If being NULL, it Will automatically generate a design matrix with all additive and dominant effect and without any epistasis effect.
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
scope	numeric vector. The search scope of each QTL. In the MIM process, it will search forward and backward for the corresponding cM. User can assign a numeric number for every QTL or a numeric vector for each QTL. Note that 0

	denote that the corresponding QTL position is fixed, and the position of its surrounding positions will not be searched.
speed	numeric. The walking speed of the QTL search (in cM).
conv	numeric. The convergence criterion of EM algorithm. The E and M steps will be iterated until a convergence criterion is satisfied.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

effect	The estimated effects, log likelihood value, and LRT statistics of all searched positions.
QTL.best	The positions of the best QTL combination.
effect.best	The estimated effects and LRT statistics of the best QTL combination.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

See Also

EM.MIM MIM.points2

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
result <- MIM.points(QTL, marker, geno, y, type = "RI", ng = 2, scope = c(0,3,0), speed = 2)
result$QTL.best
result$effect.best
```

MIM.points2 *QTL Short Distance Correction by MIM with Selective Genotyping*

Description

Expectation-maximization algorithm for QTL multiple interval mapping. Find the best QTL position near the designated QTL position. This function can handle the genotype data witch is selective genotyping.

Usage

```
MIM.points2(
  QTL,
  marker,
  geno,
  y,
  yu = NULL,
  sele.g = "n",
  tL = NULL,
  tR = NULL,
  method = "EM",
  type = "RI",
  D.matrix = NULL,
  ng = 2,
  cM = TRUE,
  scope = 5,
  speed = 1,
  conv = 10^-3,
  console = TRUE
)
```

Arguments

QTL	matrix. A q*2 matrix contains the QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
y	vector. An vector that contains the phenotype values of individuals with genotyped.
yu	vector. An vector that contains the phenotype value of the individuals without genotyped.
sele.g	character. If sele.g="n", it will consider that the data is not a selective genotyping data but the complete genotyping data. If sele.g="p", it will consider that the data is a selective genotyping data, and use the proposed model (Lee 2014) to analyze. If sele.g="t", it will consider that the data is a selective genotyping data, and use the truncated model (Lee 2014) to analyze. If sele.g="f, it will consider that the data is a selective genotyping data, and use the frequency-based model (Lee 2014) to analyze. Note that the yu must be input when sele.g="p" of "f".

tL	numeric. The lower truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tL=NULL, the yu must be input and the function will consider the minimum of yu as the lower truncation point.
tR	numeric. The upper truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tR=NULL, the yu must be input and the function will consider the maximum of yu as the upper truncation point.
method	character. method="EM" means the interval mapping method by Lander and Botstein (1989) is used in the analysis, while method="REG" means the approximate regression interval mapping method by Haley and Knott (1992) is considered in the analysis.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
D.matrix	matrix. The design matrix of QTL effects which is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects considered in the MIM model. The design matrix can be easily generated by the function D.make(). If being NULL, it Will automatically generate a design matrix with all additive and dominant effect and without any epistasis effect.
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
scope	numeric vector. The search scope of each QTL. In the MIM process, it will search forward and backward for the corresponding cM. User can assign a numeric number for every QTL or a numeric vector for each QTL. Note that 0 denote that the corresponding QTL position is fixed, and the position of its surrounding positions will not be searched.
speed	numeric. The walking speed of the QTL search (in cM).
conv	numeric. The convergence criterion of EM algorithm. The E and M steps will be iterated until a convergence criterion is satisfied.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

effect	The estimated effects, log likelihood value, and LRT statistics of all searched positions.
QTL.best	The positions of the best QTL combination.
effect.best	The estimated effects and LRT statistics of the best QTL combination.
model	The model of selective genotyping data in this analyze.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

H.-I LEE, H.-A. HO and C.-H. KAO 2014 A new simple method for improving QTL mapping under selective genotyping. Genetics 198: 1685-1698.

See Also

EM.MIM2 MIM.points

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# make the seletive genotyping data
ys <- y[y > quantile(y)[4] | y < quantile(y)[2]]
yu <- y[y >= quantile(y)[2] & y <= quantile(y)[4]]
geno.s <- geno[y > quantile(y)[4] | y < quantile(y)[2],]

# run and result
result <- MIM.points2(QTL, marker, geno.s, ys, yu, sele.g = "p",
 type = "RI", ng = 2, scope = c(0,3,0), speed = 2)
result$QTL.best
result$effect.best
```

MIM.search *QTL search by MIM*

Description

Expectation-maximization algorithm for QTL multiple interval mapping. Find one more QTL in the presence of some known QTLs.

Usage

```
MIM.search(
  QTL,
  marker,
  geno,
  y,
  method = "EM",
  type = "RI",
  D.matrix = NULL,
```

```
  ng = 2,
  cM = TRUE,
  speed = 1,
  QTLdist = 15,
  conv = 10^-3,
  console = TRUE
)
```

Arguments

QTL	matrix. A q*2 matrix contains the known QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
y	vector. An vector with n elements that contains the phenotype values of individuals.
method	character. method="EM" means the interval mapping method by Lander and Botstein (1989) is used in the analysis, while method="REG" means the approximate regression interval mapping method by Haley and Knott (1992) is considered in the analysis.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
D.matrix	matrix. The design matrix of QTL effects which is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects considered in the MIM model. Note that the QTL number of the design matrix must be the original QTL number plus one. The design matrix can be easily generated by the function D.make(). If being NULL, it will automatically generate a design matrix with all additive and dominant effect and without any epistasis effect.
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
speed	numeric. The walking speed of the QTL search (in cM).
QTLdist	numeric. The minimum distance (cM) among different linked significant QTL. The position near the position of the known QTLs under this distance will not be consider as the candidate position in the search process.

conv	numeric. The convergence criterion of EM algorithm. The E and M steps will be iterated until a convergence criterion is satisfied.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

effect	The estimated effects, log likelihood value, and LRT statistics of all searched positions.
QTL.best	The positions of the best QTL combination.
effect.best	The estimated effects and LRT statistics of the best QTL combination.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

See Also

EM.MIM MIM.search2

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
QTL <- c(1, 23)
result <- MIM.search(QTL, marker, geno, y, type = "RI", ng = 2, speed = 15, QTLdist = 50)
result$QTL.best
result$effect.best
```

MIM.search2 *QTL search by MIM with Seletive Genotyping*

Description

Expectation-maximization algorithm for QTL multiple interval mapping. Find one more QTL in the presence of some known QTLs. This funtion can handle the genotype witch is seletive genotyping.

Usage

```
MIM.search2(
  QTL,
  marker,
  geno,
  y,
  yu = NULL,
  sele.g = "n",
  tL = NULL,
  tR = NULL,
  method = "EM",
  type = "RI",
  D.matrix = NULL,
  ng = 2,
  cM = TRUE,
  speed = 1,
  QTLdist = 15,
  conv = 10^-3,
  console = TRUE
)
```

Arguments

QTL	matrix. A q*2 matrix contains the QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
geno	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
y	vector. An vector that contains the phenotype values of individuals with genotyped.
yu	vector. An vector that contains the phenotype value of the individuals without genotyped.
sele.g	character. If sele.g="n", it will consider that the data is not a selective genotyping data but the complete genotyping data. If sele.g="p", it will consider that the data is a selective genotyping data, and use the proposed model (Lee 2014) to analyze. If sele.g="t", it will consider that the data is a selective genotyping data, and use the truncated model (Lee 2014) to analyze. If sele.g="f, it will consider that the data is a selective genotyping data, and use the frequency-based model (Lee 2014) to analyze. Note that the yu must be input when sele.g="p" of "f".

tL	numeric. The lower truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tL=NULL, the yu must be input and the function will consider the minimum of yu as the lower truncation point.
tR	numeric. The upper truncation point of phenotype value when sele.g="t". Note that when sele.g="t" and tR=NULL, the yu must be input and the function will consider the maximum of yu as the upper truncation point.
method	character. method="EM" means the interval mapping method by Lander and Botstein (1989) is used in the analysis, while method="REG" means the approximate regression interval mapping method by Haley and Knott (1992) is considered in the analysis.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
D.matrix	matrix. The design matrix of QTL effects which is a g*p matrix, where g is the number of possible QTL genotypes, and p is the number of effects considered in the MIM model. Note that the QTL number of the design matrix must be the original QTL number plus one. The design matrix can be easily generated by the function D.make(). If being NULL, it will automatically generate a design matrix with all additive and dominant effect and without any epistasis effect.
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
speed	numeric. The walking speed of the QTL search (in cM).
QTLdist	numeric. The minimum distance (cM) among different linked significant QTL. The position near the position of the known QTLs under this distance will not be consider as the candidate position in the search process.
conv	numeric. The convergence criterion of EM algorithm. The E and M steps will be iterated until a convergence criterion is satisfied.
console	logical. To decide whether the process of algorithm will be shown in the R console or not.

Value

effect	The estimated effects, log likelihood value, and LRT statistics of all searched positions.
QTL.best	The positions of the best QTL combination.
effect.best	The estimated effects and LRT statistics of the best QTL combination.
model	The model of selective genotyping data in this analyze.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

H.-I LEE, H.-A. HO and C.-H. KAO 2014 A new simple method for improving QTL mapping under selective genotyping. Genetics 198: 1685-1698.

See Also

EM.MIM2 MIM.search

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# make the seletive genotyping data
ys <- y[y > quantile(y)[4] | y < quantile(y)[2]]
yu <- y[y >= quantile(y)[2] & y <= quantile(y)[4]]
geno.s <- geno[y > quantile(y)[4] | y < quantile(y)[2],]

# run and result
QTL <- c(1, 23)
result <- MIM.search2(QTL, marker, geno.s, ys, yu, sele.g = "p",
 type = "RI", ng = 2, speed = 15, QTLdist = 50)
result$QTL.best
result$effect.best
```

---

progeny *Progeny Simulation*

Description

Generate the simulated phenotype and genotype data for a specified generation from various breeding schemes.

Usage

```
progeny(
  QTL,
  marker,
  type = "RI",
  ng = 2,
  cM = TRUE,
  E.vector = NULL,
  h2 = 0.5,
  size = 200
)
```

Arguments

QTL	matrix. A q*2 matrix contains the QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
marker	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.
E.vector	vector. Set the effect of QTLs. It should be a named vector, and the name of elements should be the effects of QTLs and their interaction. For example, the additive effect of QTL1 is coded to "a1"; the dominant effect of QTL2 is coded to "d2"; and the interaction of the additive effect of QTL2 and the dominant effect of QTL1 is coded to "a2:d1". So that, if the additive effect of QTL1 is 2, the dominant effect of QTL2 is 5, and the interaction of the additive effect of QTL2 and the dominant effect of QTL1 is 3, the user should input E.vector=c("a1"=2, "d2"=5, "a2:d1"=3). If E.vector=NULL, the phenotypic value will not be simulated.
h2	numeric. Set the heritability for simulated phenotypes. It should be a number between 0 and 1.
size	numeric. The population size of simulated progeny.

Value

phe	The phenotypic value of simulated progeny.
E.vector	The effect vector used in this simulation.
marker.prog	The marker genotype of simulated progeny.
QTL.prog	The QTL genotype of simulated progeny.

References

Haldane J.B.S. 1919. The combination of linkage values and the calculation of distance between the loci for linked factors. Genetics 8: 299–309.

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
result <- progeny(QTL, marker, type = "RI", ng = 5, E.vector = c("a1" = 2, "d2" = 5, "a2:d1" = 3),
h2 = 0.5, size = 200)
result$phe
```

`Q.make` *Generate Q Matrix*

Description

Generate the conditional probability matrix by the information of QTL and marker and the genotype data.

Usage

```
Q.make(
  QTL,
  marker,
  geno = NULL,
  interval = FALSE,
  type = "RI",
  ng = 2,
  cM = TRUE
)
```

Arguments

`QTL`	matrix. A q*2 matrix contains the QTL information, where the row dimension q is the number of QTLs in the chromosomes. The first column labels the chromosomes where the QTLs are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosome and position must be divided in order.
`marker`	matrix. A k*2 matrix contains the marker information, where the row dimension k is the number of markers in the chromosomes. The first column labels the chromosomes where the markers are located, and the second column labels the positions of QTLs (in morgan (M) or centimorgan (cM)). Note that chromosomes and positions must be divided in order.
`geno`	matrix. A n*k matrix contains the k markers of the n individuals. The marker genotypes of P1 homozygote (MM), heterozygote (Mm) and P2 homozygote (mm) are coded as 2, 1 and 0, respectively, and NA for missing value.
`interval`	logical. When the QTL with the same position of a marker, whether the marker will be skipped and not be regarded as a flanking marker or not. interval=TRUE presents the marker will be skipped.

type	character. The population type of the dataset. Include backcross (type="BC"), advanced intercross population (type="AI"), and recombinant inbred population (type="RI").
ng	integer. The generation number of the population type. For example, the BC1 population is type="BC" with ng=1; the AI F3 population is type="AI" with ng=3.
cM	logical. Specify the unit of marker position. cM=TRUE for centi-Morgan. Or cM=FALSE for Morgan.

Value

The output contains k conditional probability matrices for the k flanking marker pairs (the k Q-matrices) and a conditional probability matrix of each QTL for all individuals (the cp-matrix, if the genotype data of testing population is input).

Note

If geno=NULL, the function can be run too and the output will contain k Q-matrices but no cp-matrix.

References

KAO, C.-H. and Z.-B. ZENG 1997 General formulas for obtaining the maximum likelihood estimates and the asymptotic variance-covariance matrix in QTL mapping when using the EM algorithm. Biometrics 53, 653-665.

KAO, C.-H., Z.-B. ZENG and R. D. TEASDALE 1999 Multiple interval mapping for Quantitative Trait Loci. Genetics 152: 1203-1216.

Examples

```
# load the example data
load(system.file("extdata", "exampledata.RDATA", package = "QTLEMM"))

# run and result
result <- Q.make(QTL, marker, geno)
head(result$cp.matrix)
```

Qhot *QTL Hotspot*

Description

This function produces both the numerical and graphical summaries of the QTL hotspot detection in the genomes that are available on the worldwide web including the flanking markers of QTLs.

Usage

```
Qhot(DataQTL, DataCrop, ScanStep = 1, NH = 100, NP = 1000, save.pdf = TRUE)
```

Arguments

DataQTL	data.frame. A data-frame with 5 columns for QTL information. The first three columns denote the serial number of QTLs, trait names, and the chromosome numbers. The 4th and 5th denote the flanking marker positions(cM) of QTLs.
DataCrop	data.frame. A data-frame with 3 columns for chromosome information consisting of the names, center positions(cM) and lengths of chromosomes.
ScanStep	numeric. A value for the length(cM) of every bin.
NH	integer. A value for the number of spurious hotspots in the proposed method.
NP	integer. A value for permutation times to calculate the threshold.
save.pdf	logical. A logical variable, if save.pdf is set to be TRUE, the pdf file of plots will be saved in the working directory instead of being shown in the console.

Value

EQF	The expected QTL frequency(EQF) in every bin per chromosome.
P.threshold	The EQF thresholds for proposed method.
Q.threshold	The EQF thresholds for Q method.
nHot	The numbers of detected hotspots per chromosome for proposed method and Q method.

Graphical outputs for visualizing the summarized results including the expected QTL frequency of scan steps, the composition of QTLs for different traits in the detected hotspots.

Note

This program may generate a large amount of graphic output. It is recommended to save the result in PDF file by the argument "save.pdf".

References

Wu, P.-Y., M.-.H. Yang, and C.-H. KAO 2021 A Statistical Framework for QTL Hotspot Detection. G3: Genes, Genomes, Genetics: jkab056.

Examples

```
# load the example data
load(system.file("extdata", "QHOTexample.RDATA", package = "QTLEMM"))

# run and result
result <- Qhot(QTL.example, crop.example, 5, 20, 100, save.pdf = FALSE)
```

Index