Package 'mppR'

February 11, 2020

Type Package

Version 1.2.1

Date 2020-02-10

Title Multi-Parent Population QTL Analysis

Description Analysis of experimental multi-parent populations to detect regions of the genome (called quantitative trait loci, QTLs) influencing phenotypic traits. The population must be composed of crosses between a set of at least three parents (e.g. factorial design, 'diallel', or nested association mapping). The functions cover data processing, QTL detection, and results visualization. The implemented methodology is described by Garin, Wimmer, Mezmouk, Malosetti and van Eeuwijk (2017) <doi:10.1007/s00122-017-2923-3>.

License GPL-3

Imports ggplot2, graphics, grDevices, methods, parallel, qtl, stats, utils, igraph

Depends R(>= 3.1.0)

Suggests testthat

RoxygenNote 6.1.1

Encoding UTF-8

URL https://github.com/vincentgarin/mppR

BugReports https://github.com/vincentgarin/mppR/issues

NeedsCompilation no

Author Vincent Garin [aut, cre],
Valentin Wimmer [aut],
Dietrich Borchardt [ctb, dtc],
Fred van Eeuwijk [ctb, ths],
Marcos Malosetti [ctb, ths]

Maintainer Vincent Garin <vincent.garin6@gmail.com>

Repository CRAN

Date/Publication 2020-02-11 21:10:02 UTC

R topics documented:

create.mppData . . . 2
CV_partition . . . 4
design_connectivity . . . 5
IBD.mppData . . . 6
IBS.mppData . . . 8
inc_mat_QTL . . . 9
mppData . . . 11
mppData_init . . . 12
mpp_back_elim . . . 13
mpp_CIM . . . 14
mpp_CV . . . 16
mpp_perm . . . 19
mpp_proc . . . 20
mpp_SIM . . . 23
MQE_proc . . . 26
parent_cluster.mppData . . . 28
par_clu . . . 30
plot.QTLprof . . . 30
print.summary.mppData . . . 32
print.summary.QeffRes . . . 33
print.summary.QR2Res . . . 33
QC.mppData . . . 34
QTL_gen_effects . . . 36
QTL_pred_R2 . . . 38
QTL_R2 . . . 40
QTL_select . . . 42
subset.mppData . . . 43
summary.mppData . . . 44
summary.QeffRes . . . 45
summary.QR2Res . . . 46
USNAM_geno . . . 46
USNAM_map . . . 47
USNAM_pheno . . . 48

Index **50**

`create.mppData` *Create a multi-parent population data object*

Description

This function combines all raw data sources in a single data object of class `mppData`.

Usage

```
create.mppData(geno.off = NULL, geno.par = NULL, map = NULL,
  pheno = NULL, cross.ind = NULL, par.per.cross = NULL)
```

Arguments

geno.off	Character marker score matrix of the offspring with genotypes as row and markers as column. **Rows names must be the offspring genotypes identifiers similar to the one used in** pheno**. The columns names must be the marker names similar to the one used in** map**. Marker scores must be coded using one letter per allele. For example, AA, CC, GG, etc. Missing values must be coded** NA**.**
geno.par	Character marker score matrix of the parents with genotypes as row and markers as column. **Rows names must be the parents genotypes identifiers similar to the one used in** par.per.cross**. The columns names must be the marker names similar to the one used in** map**. Marker scores must be coded using one letter per allele. For example, AA, CC, GG, etc. Missing values must be coded** NA**.**
map	Three columns data.frame with: 1) character marker identifiers; 2) numeric chromosome; 3) numeric positions in centi-Morgan.**The marker identifiers must be identical to the column names of the maker matrices (**geno.off **and** geno.par**). The chromosome identifiers must start by 1 and increase by 1 unit, e.g. 1, 2, 3, ...**
pheno	Numeric matrix with one column per trait and rownames as genotypes identifiers. **The genotypes identifiers must be identical to the rownames of the offspring marker matrix (**geno.off**).**
cross.ind	Character vector indicating to which cross each genotype belongs.
par.per.cross	Three columns Character matrix specifying : 1) the cross indicators; 2) the parents 1 identifiers of the crosses; 3) the parents 2 identifiers of the crosses. **The list of crosses must contain the same cross indicators as in** cross.ind **and they must appear in the same order. The list of parent identifiers must be the same to the rownames of the argument** geno.par.

Value

a list of class mppData which contains the following elements

geno.off	Matrix with offspring marker scores.
geno.par	Matrix with parents marker scores.
pheno	Matrix with phenotypic trait values.
map	Data.frame with genetic marker information.
cross.ind	Cross indicator.
par.per.cross	Character matrix information about cross and the parents of the crosses.

The list also contain other arguments that will be filled later in the data processing.

Author(s)

Vincent Garin

Examples

```
data(USNAM_geno)
data(USNAM_map)
data(USNAM_pheno)

geno.off <- USNAM_geno[7:506, ]
geno.par <- USNAM_geno[1:6, ]
map <- USNAM_map
pheno <-  USNAM_pheno
cross.ind <- substr(rownames(pheno), 1, 4)
par.per.cross <- cbind(unique(cross.ind), rep("B73", 5),
                       rownames(geno.par)[2:6])

mppData <- create.mppData(geno.off = geno.off, geno.par = geno.par,
                          map = map, pheno = pheno, cross.ind = cross.ind,
                          par.per.cross = par.per.cross)
```

`CV_partition` *Cross validation partition*

Description

Partition the genotype indices into training and validation sets for cross-validation (CV).

Usage

```
CV_partition(cross.ind, k = 5)
```

Arguments

`cross.ind`	`Character` vector with the same length as the number of genotypes which specifies to which cross each genotype belongs.
`k`	`Numeric` value representing the number of subsets (fold) into which data are spread within cross. Default = 5.

Details

The genotype indices are randomly assigned within cross to k subsets (folds). Then each subset is used once as validation set, the remaining data go in the training set.

Value

Return:

`fold`	`List` of k lists (one for each fold). Each fold list contains two vectors with genotypes indices of the training (`$train.set`) and the validation set (`$val.set`).

Author(s)

Vincent Garin

See Also

mpp_CV

Examples

```
data(mppData)

part.cv <- CV_partition(cross.ind = mppData$cross.ind, k = 5)

part.cv[[1]]$train.set
part.cv[[1]]$val.set
```

---

design_connectivity *Connected parts of a MPP design*

Description

Determine the connected parts of a MPP design using the method of Weeks and Williams (1964) and the package igraph.

Usage

```
design_connectivity(par_per_cross, plot_des = TRUE, output_loc = NULL)
```

Arguments

par_per_cross	Three columns character matrix specifying : 1) the cross indicators ; 2) the parents 1 identifiers of the crosses; 3) the parents 2 identifiers of the crosses.
plot_des	Logical value indicating if connected part of the design should be plotted. Default = TRUE.
output_loc	Path where the plot of the design will be saved if the argument is given. Default = NULL.

Value

Return a list with each element representing one connected part of the design and the list of parents contained in this part.

If plot_des = TRUE and output_loc has been specified. A plot of the graph (con_plot.pdf) will be saved at the specified location.

Author(s)

Vincent Garin

References

Weeks, D. L., & Williams, D. R. (1964). A note on the determination of connectedness in an N-way cross classification. Technometrics, 6(3), 319-324.

Examples

```
data(mppData)

par_per_cross <- mppData$par.per.cross

con.part <- design_connectivity(par_per_cross)
```

IBD.mppData *IBD coding for* mppData *objects*

Description

The function first converts genotype data into ABH format. Then it calculates within cross identical by descent (IBD) probabilities.

Usage

```
IBD.mppData(mppData, het.miss.par = TRUE, subcross.ind = NULL,
  par.per.subcross = NULL, type, F.gen = NULL, BC.gen = NULL,
  type.mating = NULL, error.prob = 1e-04, map.function = "haldane")
```

Arguments

mppData	An object of class mppData. the mppData must have been processed using: create.mppData, QC.mppData, and IBS.mppData.
het.miss.par	Logical value. if het.miss.par = TRUE, the function will use the offspring segregation to try to infer the allele that was transmitted by the heterozygous or missing parent at a particular locus to make the ABH conversion. Default = TRUE.
subcross.ind	Optional character vector specifying to which sub-cross each genotype belong. Default = NULL.
par.per.subcross	Optional three columns Character matrix specifying : 1) the sub-cross indicators; 2) the parents 1 identifiers of the sub-crosses; 3) the parents 2 identifiers of the sub-crosses. Default = NULL.

type	Character indicator for the type of population analyzed: type = "F" for Fn (F cross n generations); type = "BC" for BCn (backcross n generations); type = "BCsFt" for backcross followed by selfing; type = "DH" for double haploids; and type = "RIL" for recombinant inbred lines. For RIL type specify if the population was obtain using selfing or sibling mating using type.mating. If type = "RIL" or "DH", the function does not assume heterozygous marker scores for these populations and convert them into missing (NA).
F.gen	Numeric integer representing the number of F generations. For example F.gen = 2 for F2. Default = NULL.
BC.gen	Numeric integer representing the number of backcross generations. For example BC.gen = 1 for single backcross. Default = NULL.
type.mating	Character specifying for a RIL population if it was obtained by selfing ("selfing") or by sibling mating ("sib.mat"). Default = NULL.
error.prob	Numeric value for assumed genotyping error rate used in the calculation of the penetrance Pr(observed genotype | true genotype). Default = 0.0001.
map.function	Character expression specifying the type of map function used to infer the IBD probabilities. possibility to choose between "haldane", "kosambi","c-f","morgan". Default = "haldane".

Details

The function first transforms genotype data into within cross ABH format. The function takes the parents of the different cross as reference and assigns the following scores: "A" if the offspring score is equivalent to parent 1; "B" if it is equivalent to parent 2; "H" if it is heterozygous. The function attributes NA for missing when: 1) the offspring score is missing; 2) the two parents have the same score; or 3) when at least one parental score is missing.

If a parent score is heterozygous or missing (het.miss.par = TRUE), the assignment rules are the following. If the two parents are heterozygous or one parent is heterozygous and the other missing, the offspring get NA since the parental origin can not be inferred with certainty. If one parent is heterozygous or missing and the second parent is homozygous, the function looks at offspring segregating pattern to infer which allele was transmitted by the heterozygous parent. If this is possible we consider the heteroxzygous parent as homozygous for the transmitted allele and use this allele score for ABH assignment.

The ABH assignment can be performed using sub-cross structure providing information about sub-cross in arguments subcross.ind and par.per.subcross.

Then the function calculates the IBD probabilities using read.cross() and calc.genoprob() functions from the R/qtl package (Broman et al. 2009).

The type of population must be specified in argument *type*. Different population types are possible: F-type ('F'), back-cross ('BC'), backcross followed by selfing ('BCsFt'), double haploid ('DH'), and recombinant imbred lines ('RIL'). The number of F and BC generations can be specified using F.gen and BC.gen. The argument type.mating specifies if F and RIL populations were obtained by selfing or by sibling mating.

DH and RIL populations are read as back-cross by R/qtl. For these two population types, heterozygous scores will be treated as missing values.

Value

an increased mppData object containing the the same elements as the mppData object provided as argument and the following new elements:

geno.IBD	A R/qtl cross.object containing the IBD probabilities.
n.zigo	Numeric value Indicating the number of different genotypes: 2 (AA/BB) or 3 (AA/AB/BB)
type	Character expression indicating the type of population.

Author(s)

Vincent Garin

References

Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. Bioinformatics 19:889-890.

Broman, K. W., & Sen, S. (2009). A Guide to QTL Mapping with R/qtl (Vol. 46). New York: Springer.

See Also

create.mppData, QC.mppData, IBS.mppData

Examples

```
data(mppData_init)

mppData <- QC.mppData(mppData_init)
mppData <- IBS.mppData(mppData = mppData)

mppData <- IBD.mppData(mppData = mppData, het.miss.par = TRUE, type = 'RIL',
                       type.mating = 'selfing')
```

IBS.mppData *IBS coding for* mppData *objects*

Description

Transform the genotype marker matrix of a mppData object into Identical by state (IBS) 0, 1, 2 format. The IBS score represent the number of copies of the minor allele.

Usage

```
IBS.mppData(mppData)
```

Arguments

mppData	An object of class mppData. The mppData must have been processed using: create.mppData and QC.mppData.

Value

an increased mppData object containing the the same elements as the mppData object provided as argument and the following new elements:

geno.IBS	Marker matrix with marker scores coded as 0, 1, 2 corresponding to the number of copies of the least frequent SNP allele.
allele.ref	matrix with reference allele scores. The first row represents the minor allele (lowest frequency), the second the one represent the major allele (largest frequency) and the two others the heterozygous scores.

Author(s)

Vincent Garin

See Also

create.mppData, QC.mppData

Examples

```
data(mppData_init)

mppData <- QC.mppData(mppData_init)

mppData <- IBS.mppData(mppData = mppData)
```

---

inc_mat_QTL *QTL incidence matrix*

Description

Build a single position QTL incidences matrix.

Usage

```
inc_mat_QTL(x, mppData, Q.eff, order.MAF = FALSE)
```

Arguments

x	Integer value indicating the genetic position on the map (mppData$map) of the QTL incidence matrix.
mppData	An object of class mppData.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
order.MAF	Logical value specifying if the QTL incidence matrix should be ordered by allele frequency for a parental and ancestral QTL incidence matrix. The column will be ordered from the least to the most frequent allele. Default = FALSE.

Value

Return:

QTL.mat	QTL incidence matrix. For the cross-specific model, it represents the difference between the number of allele from parent 2 or B and parent 1 or A divided by two. For parental (ancestral) model it represents the expected number of parental (ancestral) allele copies. For the bi-allelic model, it represents the number of copies of the least frequent allele.

Author(s)

Vincent Garin

See Also

mpp_SIM,

Examples

```
data(mppData)

QTLmatCr <- inc_mat_QTL(x = 2, mppData = mppData, Q.eff = "cr")

QTLmatPar <- inc_mat_QTL(x = 2, mppData = mppData, Q.eff = "par")

QTLmatAnc <- inc_mat_QTL(x = 2, mppData = mppData, Q.eff = "anc")

QTLmatBi <- inc_mat_QTL(x = 2, mppData = mppData, Q.eff = "biall")
```

mppData *Complete* mppData *object*

Description

Complete mppData object made from a sample data of the maize US nested association mapping (NAM) population. This mppData object went through all the steps of the data processing: create.mppData, QC.mppData, IBS.mppData, IBD.mppData, parent_cluster.mppData. The mppData contain all the data necessary for the QTL analysis procedures.

Usage

```
data(mppData)
```

Format

mppData

Details

The complete mppData object is a list containing the following elements:

1. geno.IBS: IBS genotype marker matrix.
2. geno.IBD: R/qtl cross.object containing the genotype within cross IBD probabilities.
3. geno.id: List of genotypes.
4. allele.ref: Matrix containing for each marker the most and least frequent marker scores and the two heterozygous scores.
5. geno.par: Parents marker matrix.
6. geno.par.clu: Parent marker data used to cluster the parents.
7. par.clu: Parent clustering results.
8. mono.anc: Positions for which the ancestral clustering was monomorphic.
9. pheno: Phenotypic trait matrix.
10. map: Genetic map corresponding to the geno (IBS, IBD, par) arguments.
11. haplo.map: Genetic map corresponding to geno.par.clu.
12. cross.ind: Vector indicating to which cross each genotype belongs.
13. par.per.cross: Matrix with for each cross the parent 1 and 2.
14. parents: Vector of parents.
15. type: Type of population.
16. n.cr: Number of crosses.
17. n.par: Number of parents.
18. n.anc: Average number of ancestral group along the genome.
19. n.zigo: Number of possible allelic computations 2 (AA/BB) or 3 (AA/AB/BB).
20. rem.mk: Removed markers in the data processing.
21. rem.gen: Removed genotypes in the data processing.
22. status: Indicates the level of progression in the data processing.

See Also

create.mppData, QC.mppData, IBS.mppData, IBD.mppData, parent_cluster.mppData

Examples

```
data(mppData)
```

---

mppData_init mppData *object with raw data*

Description

mppData object with raw genotypic and phenotypic data of a sample of the maize US nested association mapping (NAM) population. Different operations of quality control and data processing still need to be performed before the QTL detection analysis.

Usage

```
data(mppData_init)
```

Format

mppData

Details

see examples of the create.mppData.

See Also

create.mppData

Examples

```
data(mppData_init)
```

`mpp_back_elim` *Backward elimination on QTL candidates*

Description

Performs a backward elimination using a list of given QTLs positions. The positions with a p-value above the significance level `alpha`, are successively removed.

Usage

```
mpp_back_elim(mppData, trait = 1, QTL = NULL, Q.eff = "cr",
  alpha = 0.05)
```

Arguments

`mppData`	An object of class `mppData`.
`trait`	`Numerical` or `character` indicator to specify which trait of the `mppData` object should be used. Default = 1.
`QTL`	Object of class `QTLlist` representing a list of selected position obtained with the function `QTL_select` or vector of `character` marker positions names. Default = NULL.
`Q.eff`	`Character` expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see `mpp_SIM`. Default = "cr".
`alpha`	`Numeric` value indicating the level of significance for the backward elimination. Default = 0.05.

Details

The function starts with all QTL positions in the model and test the inclusion of each position as the last in the model. If all position p-values are below `alpha` the procedure stop. If not the position with the highest p-value is remove and the procedure continue until there is no more unsignificant position.

Value

Return:

`QTL`	`Data.frame` of class `QTLlist` with five columns : 1) QTL marker names; 2) chromosomes; 3) interger position indicators on the chromosome; 4) positions in centi-Morgan; and 5) -log10(p-values).

Author(s)

Vincent Garin

See Also

mpp_SIM

Examples

```
data(mppData)

SIM <- mpp_SIM(mppData)

QTL <- QTL_select(SIM)

QTL.sel <- mpp_back_elim(mppData = mppData, QTL = QTL)
```

---

mpp_CIM *MPP Composite Interval Mapping*

Description

Compute QTL models along the genome using cofactors representing other genetic positions for control.

Usage

```
mpp_CIM(mppData, trait = 1, Q.eff = "cr", cofactors = NULL,
  window = 20, plot.gen.eff = FALSE, n.cores = 1)
```

Arguments

mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
cofactors	Object of class QTLlist representing a list of selected position obtained with the function QTL_select or vector of character marker positions names. Default = NULL.
window	Numeric distance (cM) on the left and the right of a cofactor position where it is not included in the model. Default = 20.
plot.gen.eff	Logical value. If plot.gen.eff = TRUE, the function will save the decomposed genetic effects per cross/parent. These results can be plotted with the function plot.QTLprof to visualize a genome-wide decomposition of the genetic effects. **This functionality is only available for the cross-specific, parental and ancestral models.** Default value = FALSE.
n.cores	Numeric. Specify here the number of cores you like to use. Default = 1.

Details

For more details about the different models, see documentation of the function mpp_SIM. The function returns a -log10(p-value) QTL profile.

Value

Return:

CIM	Data.frame of class QTLprof. with five columns : 1) QTL marker names; 2) chromosomes; 3) interger position indicators on the chromosome; 4) positions in centi-Morgan; and 5) -log10(p-val). And if plot.gen.eff = TRUE, p-values of the cross or parental QTL effects.

Author(s)

Vincent Garin

See Also

mpp_SIM, QTL_select

Examples

```
# Cross-specific effect model
#############################

data(mppData)

SIM <- mpp_SIM(mppData = mppData, Q.eff = "cr")

cofactors <- QTL_select(Qprof = SIM, threshold = 3, window = 20)

CIM <- mpp_CIM(mppData = mppData, Q.eff = "cr", cofactors = cofactors,
               window = 20, plot.gen.eff = TRUE)

plot(x = CIM)
plot(x = CIM, gen.eff = TRUE, mppData = mppData, Q.eff = "cr")

# Bi-allelic model
##################

cofactors <- mppData$map[c(15, 63), 1]

CIM <- mpp_CIM(mppData = mppData, Q.eff = "biall", cofactors = cofactors,
               window = 20)

plot(x = CIM, type = "h")
```

mpp_CV *MPP cross-validation*

Description

Evaluation of MPP QTL detection procedure by cross-validation (CV).

Usage

```
mpp_CV(pop.name = "MPP_CV", trait.name = "trait1", mppData,
  trait = 1, her = 1, Rep = 10, k = 5, Q.eff = "cr",
  thre.cof = 3, win.cof = 50, N.cim = 1, window = 20,
  thre.QTL = 3, win.QTL = 20, backward = TRUE, alpha.bk = 0.05,
  n.cores = 1, verbose = TRUE, output.loc)
```

Arguments

pop.name	Character name of the studied population. Default = "MPP_CV".
trait.name	Character name of the studied trait. Default = "trait1".
mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
her	Numeric value between 0 and 1 representing the heritability of the trait. her can be a single value or a vector specifying each within cross heritability. Default = 1.
Rep	Numeric value representing the number of repetition of the k-fold procedure. Default = 10.
k	Numeric value representing the number of folds for the within cross partition of the population. Default = 5.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
thre.cof	Numeric value representing the -log10(p-value) threshold above which a position can be peaked as a cofactor. Default = 3.
win.cof	Numeric value in centi-Morgan representing the minimum distance between two selected cofactors. Default = 50.
N.cim	Numeric value specifying the number of time the CIM analysis is repeated. Default = 1.
window	Numeric distance (cM) on the left and the right of a cofactor position where it is not included in the model. Default = 20.
thre.QTL	Numeric value representing the -log10(p-value) threshold above which a position can be selected as QTL. Default = 3.

win.QTL	Numeric value in centi-Morgan representing the minimum distance between two selected QTLs. Default = 20.
backward	Logical value. If backward = TRUE, the function performs a backward elimination on the list of selected QTLs. Default = TRUE.
alpha.bk	Numeric value indicating the significance level for the backward elimination. Terms with p-values above this value will iteratively be removed. Default = 0.05.
n.cores	Numeric. Specify here the number of cores you like to use. Default = 1.
verbose	Logical value indicating if the progresses of the CV should be printed. Default = TRUE.
output.loc	Path where a folder will be created to save the results.

Details

For details on the MPP QTL detection models see mpp_SIM documentation. The CV scheme is adapted from Utz et al. (2000) to the MPP context. A single CV run works like that:

1. Generation of a k-fold partition of the data. The partition is done within crosses. Each cross is divided into k subsets. Then for the kth repetition, the kth subset is used as validation set, the rest goes into the training set.
2. For the kth repetition, utilization of the training set for cofactor selection and multi-QTL model determination (mpp_SIM and mpp_CIM). If backward = TRUE, the final list of QTLs is tested simultaneously using a backward elimination (mpp_back_elim).
3. Use the list of detected QTLs in the training set to calculate the proportion of genetic variance explained by all detected QTLs in the training set (p.ts = R2.ts/h2). Where R2.ts is the adjusted R squared and h2 is the average within cross heritability (her). By default, her = 1, which mean that

   For each single QTL effect, difference partial R squared are also calculated. Difference R squared are computed by doing the difference between a model with all QTLs and a model without the ith position. For details about R squared computation and adjustment look at QTL_R2.
4. Use the estimates of the QTL effects in the training set (B.ts) to predict the phenotypic values of the validation set. y.pred.vs = X.vs*B.ts. Computes the predicted R squared in the validation set using the squared Pearson correlation coefficient between the real values (y.vs) and the predicted values (y.pred.vs). R2.vs = cor(y.ts,y.pred.ts)^2. Then the predicted genetic variance in the validation set (p.vs) is equal to p.vs = R2.vs/h2. For heritability correction, the user can provide a single value for the average within cross heritability or a vector specifying each within cross heritability. By default, her = 1, which means that the results represent the proportion of phenotypic variance explained (predicted) in the training (validation) sets.

   The predicted R squared is computed per cross and then averaged at the population level (p.ts). Both results are returned. Partial QTL predicted R squared are also calculated using the difference between the predicted R squared using all QTL and the predicted R squared without QTL i. The bias between p.ts and p.vs is calculated as bias = 1 - (p.vs/p.ts).

Value

List containing the following results items:

CV_res	Data.frame containing for each CV run: 1) the number of detected QTL; 2) the proportion of explained genetic variance in the TS (p.ts); 3) the proportion of predicted genetic variance in the VS (p.vs) at the population level (average of within cross prediction); the bias between p.ts and p.vs (bias = 1-(p.vs/p.ts)).
p.vs.cr	Matrix containing the within cross p.vs for each CV run.
QTL	Data.frame containing: 1) the list of QTL position detected at least one time during the entire CV process; 2) the number of times the position has been detected; 3) the average partial p.ts of the QTL position; 4) the average partial p.vs of the QTL position; 5) the average partial bias of the QTL position.
QTL.profiles	Data.frame -log10(p-value) QTL profiles of the different CV runs.

The results elements return as R object are also saved as text files at the specified output location (output.loc). A transparency plot of the CV results (plot.pdf) is also saved.

Author(s)

Vincent Garin

References

Utz, H. F., Melchinger, A. E., & Schon, C. C. (2000). Bias and sampling error of the estimated proportion of genotypic variance explained by quantitative trait loci determined from experimental data in maize using cross validation and validation with independent samples. Genetics, 154(4), 1839-1849.

See Also

mpp_back_elim, mpp_CIM, mpp_perm, mpp_SIM, QTL_R2

Examples

```
data(mppData)

# Specify a location where your results will be saved
my.loc <- tempdir()

CV <- mpp_CV(pop.name = "USNAM", trait.name = "ULA", mppData = mppData,
her = .4, Rep = 1, k = 3, verbose = FALSE, output.loc = my.loc)
```

mpp_perm *QTL significance threshold by permutation*

Description

Determination of an empirical null distribution of the QTL significance threshold for a MPP QTL analysis using permutation test (Churchill and Doerge, 1994).

Usage

```
mpp_perm(mppData, trait = 1, Q.eff = "cr", N = 1000, q.val = 0.95,
  verbose = TRUE, n.cores = 1)
```

Arguments

mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
N	Number of permutations. Default = 1000.
q.val	Single numeric value or vector of desired quantiles from the null distribution. Default = 0.95.
verbose	Logical value indicating if progression of the function should be printed. It will not affect the printing of the other functions called by mpp_perm(), especially the printing of asreml(). Default = TRUE.
n.cores	Numeric. Specify here the number of cores you like to use. Default = 1.

Details

Performs N permutations of the trait data and computes each time a genome-wide QTL profile. For every run, it stores the highest -log10(p-val). These values can be used to build a null distribution for the QTL significance threshold. Quantile values can be determined from the previous distribution. For more details about the different possible models and their assumptions see mpp_SIM documentation.

Value

Return:

List with the following object:

max.pval	Vector of the highest genome-wide -log10(p-values).
q.val	Quantile values from the QTL significance threshold null distribution.
seed	Numeric vector of random generated seed values for each permutation.

Author(s)

Vincent Garin

References

Churchill, G. A., & Doerge, R. W. (1994). Empirical threshold values for quantitative trait mapping. Genetics, 138(3), 963-971.

See Also

mpp_SIM

Examples

```
data(mppData)

Perm <- mpp_perm(mppData = mppData, Q.eff = "cr", N = 5)
```

---

mpp_proc *MPP QTL analysis*

Description

Multi-parent population QTL analysis.

Usage

```
mpp_proc(pop.name = "MPP", trait.name = "trait1", mppData, trait = 1,
  Q.eff = "cr", plot.gen.eff = FALSE, thre.cof = 3, win.cof = 50,
  N.cim = 1, window = 20, thre.QTL = 3, win.QTL = 20,
  backward = TRUE, alpha.bk = 0.05, ref.par = NULL,
  sum_zero = FALSE, CI = FALSE, drop = 1.5, text.size = 18,
  n.cores = 1, verbose = TRUE, output.loc)
```

Arguments

pop.name	Character name of the studied population. Default = "MPP".
trait.name	Character name of the studied trait. Default = "trait1".
mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.

Q.eff	Character expression indicating the assumption concerning the QTL effect: 1) "cr" for cross-specific effects; 2) "par" parental effects; 3) "anc" for an ancestral effects; 4) "biall" for a bi-allelic effects. For more details see mpp_SIM. Default = "cr".
plot.gen.eff	Logical value. If plot.gen.eff = TRUE, the function will save the decomposed genetic effects per cross/parent. These results can be plotted with the function plot.QTLprof to visualize a genome-wide decomposition of the genetic effects. **This functionality is ony available for the cross-specific, parental and ancestral models.** Default value = FALSE.
thre.cof	Numeric value representing the -log10(p-value) threshold above which a position can be peaked as a cofactor. Default = 3.
win.cof	Numeric value in centi-Morgan representing the minimum distance between two selected cofactors. Default = 50.
N.cim	Numeric value specifying the number of time the CIM analysis is repeated. Default = 1.
window	Numeric distance (cM) on the left and the right of a cofactor position where it is not included in the model. Default = 20.
thre.QTL	Numeric value representing the -log10(p-value) threshold above which a position can be selected as QTL. Default = 3.
win.QTL	Numeric value in centi-Morgan representing the minimum distance between two selected QTLs. Default = 20.
backward	Logical value. If backward = TRUE, the function performs a backward elimination on the list of selected QTLs. Default = TRUE.
alpha.bk	Numeric value indicating the significance level for the backward elimination. Terms with p-values above this value will iteratively be removed. Default = 0.05.
ref.par	Optional Character expression defining the parental allele that will be used as reference to compute QTL effects for the parental model. For the ancestral model, the ancestral class containing the reference parent will be set as reference. **This option can only be used if the MPP design is composed of a unique connected part**. Default = NULL.
sum_zero	Optional Logical value specifying if the QTL effect of a parental or an ancestral model should be calculated using the sum to zero constraint. Default = FALSE.
CI	Logical value. If CI = TRUE, the function will compute a -log10(pval) drop confidence interval for each QTL after calculating a CIM- profile (without cofactors on the scanned chromosome). Default = FALSE.
drop	Numeric -log10(p-value) drop value at the limits of the interval. Default = 1.5.
text.size	Numeric value specifying the size of graph axis text elements. Default = 18.
n.cores	Numeric. Specify here the number of cores you like to use. Default = 1.
verbose	Logical value indicating if the steps of mpp_proc should be printed. Default = TRUE.
output.loc	Path where a folder will be created to save the results.

Details

The function run a full MPP QTL detection using models with different possible assumptions concerning the number of alleles at the QTL position. For more details about the different models, see documentation of the function `mpp_SIM`. The procedure is the following:

1. Simple interval mapping (SIM) to select cofactor (`mpp_SIM`).
2. Composite interval mapping (CIM) with selected cofactors (`mpp_CIM`).
3. Optional backward elimination on the list of QTL candidates (`backward = TRUE`) (`mpp_back_elim`).
4. Computation of the QTL genetic effects (`QTL_gen_effects`) and proportion of the phenotypic variation explained by the QTLs (R squared) (`QTL_R2`).
5. Optional QTL confidence interval computation from a CIM- profile (excluding cofactors on the scanned chromosome) (`argument CI=TRUE`).

Value

Return:

List containing the following items:

`n.QTL`	Number of detected QTLs.
`cofactors`	`Data.frame` with cofactors positions.
`QTL`	`Data.frame` with QTL positions.
`R2`	`List` containing R squared statistics of the QTL effects. For details see `QTL_R2` output section.
`QTL.effects`	`List` of QTLs genetic effects. For details see `QTL_gen_effects` output section.
`QTL.CI`	If `CI = TRUE`, confidence interval information of the QTLs.

Some output files are also saved at the specified location (`output.loc`):

1. A QTL report (QTL_REPORT.txt) with: 1) the number of detected QTLs; 2) the global R squared statistics; 3) for each QTL, position information (plus confidence interval if `CI = TRUE`) and estimated QTL genetic effects per cross or parents (for details see `QTL_gen_effects`).
2. The SIM and CIM results in a text file (SIM.txt, CIM.txt).
3. The list of cofactors (cofactors.txt).
4. The list of QTL (QTL.txt).
5. The QTL R squared statistics (QTL_R2.txt) (for details see `QTL_R2`).
6. If `CI = TRUE`, the QTL confidence intervals (QTL_CI.txt).
7. General results of the QTL detection process: number of QTLs and global adjusted and non-adjusted R squared statistics (QTL_genResults.txt).
8. The plot of the CIM profile (QTL_profile.pdf) with dotted vertical lines representing the cofactors positions. If `plot.gen.eff = TRUE`, plot of the genetic effects per cross or parents (gen_eff.pdf) with dashed lines representing the QTL positions. For more details see `plot.QTLprof`

Author(s)

Vincent Garin

See Also

mpp_back_elim, mpp_CIM, mpp_perm, mpp_SIM, plot.QTLprof, QTL_gen_effects, QTL_R2

Examples

```
data(mppData)

# Specify a location where your results will be saved
my.loc <- tempdir()

# Cross-specific model

USNAM_cr <- mpp_proc(pop.name = "USNAM", trait.name = "ULA",
                     mppData = mppData, plot.gen.eff = TRUE, CI = TRUE,
                     verbose = FALSE, output.loc = my.loc)
```

---

mpp_SIM *MPP Simple Interval Mapping*

Description

Computes single QTL models along the genome using different models.

Usage

```
mpp_SIM(mppData, trait = 1, Q.eff = "cr", plot.gen.eff = FALSE,
  n.cores = 1)
```

Arguments

mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".

`plot.gen.eff` `Logical` value. If `plot.gen.eff = TRUE`, the function will save the decomposed genetic effects per cross/parent. These results can be plotted with the function `plot.QTLprof` to visualize a genome-wide decomposition of the genetic effects. **This functionality is ony available for the cross-specific, parental and ancestral models.** Default value = FALSE.

`n.cores` `Numeric`. Specify here the number of cores you like to use. Default = 1.

Details

The implemented models vary according to the number of alleles assumed at the QTL position and their origin. Four assumptions for the QTL effect are possible.

Concerning the type of QTL effect, the first option is a cross-specific QTL effects model (`Q.eff = "cr"`). In this model, the QTL effects are assumed to be nested within cross which leads to the estimation of one parameter per cross. The cross-specific model corresponds to the disconnected model described in Blanc et al. 2006.

A second possibility is the parental model (`Q.eff = "par"`). The parental model assumes one QTL effect (allele) per parent that are independent from the genetic background. This means that QTL coming form parent i has the same effect in all crosses where this parent is used. This model is supposed to produce better estimates of the QTL due to larger sample size when parents are shared between crosses.

In a connected MPP (`design_connectivity`), if np - 1 < nc, where np is the number of parents and nc the number of crosses, the parental model should be more powerful than the cross-specific model because it estimate a reduced number of QTL parameters. This gain in power will be only true if the assumption of constant parental effect through crosses holds. Calculated with HRT assumption, the parental model corresponds to the connected model presented in Blanc et al. (2006).

The third type of model is the ancestral model (`Q.eff = "anc"`). This model tries to use genetic relatedness that could exist between parents. Indeed, the parental model assumes that parent are independent which is not the case. Using genetic relatedness between the parents, it is possible group these parents into a reduced number of ancestral cluster. Parents belonging to the same ancestral group are assumed to transmit the same allele (Jansen et al. 2003; Leroux et al. 2014). The ancestral model estimate therefore one QTL effect per ancestral class. Once again, the theoretical expectation is a gain of QTL detection power by the reduction of the number of parameters to estimate. The HRT ancestral model correspond to the linkage desequilibrium linkage analysis (LDLA) models used by Bardol et al. (2013) or Giraud et al. (2014).

The final possibility is the bi-allelic model (`Q.eff = "biall"`). Bi-allelic genetic predictor are a single vector with value 0, 1 or 2 corresponding to the number of allele copy of the least frequent SNP allele. Relatedness between lines is therefore defined via identical by state (IBS) measurement. This model corresponds to models used for association mapping. For example, it is similar to model B in Wurschum et al. (2012) or association mapping model in Liu et al. (2012).

Value

Return:

`SIM` `Data.frame` of class `QTLprof`. with five columns : 1) QTL marker names; 2) chromosomes; 3) interger position indicators on the chromosome; 4) positions in centi-Morgan; and 5) -log10(p-val). And if `plot.gen.eff = TRUE`, p-values of the cross or parental QTL effects.

Author(s)

Vincent Garin

References

Bardol, N., Ventelon, M., Mangin, B., Jasson, S., Loywick, V., Couton, F., ... & Moreau, L. (2013). Combined linkage and linkage disequilibrium QTL mapping in multiple families of maize (Zea mays L.) line crosses highlights complementarities between models based on parental haplotype and single locus polymorphism. Theoretical and applied genetics, 126(11), 2717-2736.

Blanc, G., Charcosset, A., Mangin, B., Gallais, A., & Moreau, L. (2006). Connected populations for detecting quantitative trait loci and testing for epistasis: an application in maize. Theoretical and Applied Genetics, 113(2), 206-224.

Giraud, H., Lehermeier, C., Bauer, E., Falque, M., Segura, V., Bauland, C., ... & Moreau, L. (2014). Linkage Disequilibrium with Linkage Analysis of Multiline Crosses Reveals Different Multiallelic QTL for Hybrid Performance in the Flint and Dent Heterotic Groups of Maize. Genetics, 198(4), 1717-1734.

Jansen, R. C., Jannink, J. L., & Beavis, W. D. (2003). Mapping quantitative trait loci in plant breeding populations. Crop Science, 43(3), 829-834.

Leroux, D., Rahmani, A., Jasson, S., Ventelon, M., Louis, F., Moreau, L., & Mangin, B. (2014). Clusthaplo: a plug-in for MCQTL to enhance QTL detection using ancestral alleles in multi-cross design. Theoretical and Applied Genetics, 127(4), 921-933.

Liu, W., Reif, J. C., Ranc, N., Della Porta, G., & Wurschum, T. (2012). Comparison of biometrical approaches for QTL detection in multiple segregating families. Theoretical and Applied Genetics, 125(5), 987-998.

Meuwissen T and Luo, Z. (1992). Computing inbreeding coefficients in large populations. Genetics Selection Evolution, 24(4), 305-313.

Wurschum, T., Liu, W., Gowda, M., Maurer, H. P., Fischer, S., Schechert, A., & Reif, J. C. (2012). Comparison of biometrical models for joint linkage association mapping. Heredity, 108(3), 332-340.

See Also

plot.QTLprof

Examples

```
# Cross-specific model
######################

data(mppData)

SIM <- mpp_SIM(mppData = mppData, Q.eff = "cr", plot.gen.eff = TRUE)

plot(x = SIM)
plot(x = SIM, gen.eff = TRUE, mppData = mppData, Q.eff = "cr")
```

```
# Bi-allelic model
##################

SIM <- mpp_SIM(mppData = mppData, Q.eff = "biall")

plot(x = SIM, type = "h")
```

`MQE_proc` *Multi-QTL effect MPP analysis*

Description

Build multi-QTL effects (MQE) models in which different QTL effects (cross-specific, parental, ancestral or bi-allelic) can be assumed at different loci.

Usage

```
MQE_proc(pop.name = "MPP_MQE", trait.name = "trait1", mppData = NULL,
  trait = 1, Q.eff, threshold = 4, window = 30, backward = TRUE,
  alpha.bk = 0.05, plot.MQE = FALSE, n.cores = 1, verbose = TRUE,
  output.loc)
```

Arguments

`pop.name`	`Character` name of the studied population. Default = "MPP_MQE".
`trait.name`	`Character` name of the studied trait. Default = "trait1".
`mppData`	An object of class `mppData`.
`trait`	`Numerical` or `character` indicator to specify which trait of the `mppData` object should be used. Default = 1.
`Q.eff`	`Character` vector of possible QTL effects the user want to test. Elements of `Q.eff` can be "cr", "par", "anc" or "biall". For details look at `mpp_SIM`.
`threshold`	`Numeric` value representing the -log10(p-value) threshold above which a position can be considered as significant. Default = 4.
`window`	`Numeric` distance (cM) on the left and the right of a cofactor position where it is not included in the model. Default = 30.
`backward`	`Logical` value. If `backward = TRUE`, the function performs a backward elimination on the list of selected QTLs. Default = TRUE.
`alpha.bk`	`Numeric` value indicating the significance level for the backward elimination. Default = 0.05.
`plot.MQE`	`Logical` value. If `plot.MQE = TRUE`, the function will plot the last run of the MQE model determination. Default = FALSE.
`n.cores`	`Numeric`. Specify here the number of cores you like to use. Default = 1.

verbose	Logical value indicating if the steps of MQE_proc should be printed. Default = TRUE.
output.loc	Path where a folder will be created to save the results.

Details

The possible QTL effect that the user wants to allow must be specified in Q.eff. The procedure is the following:

1. Forward regression to determine a MQE model with different possible assumptions for the QTL effect at different loci. The function use.
2. Optional backward elimination (backward = TRUE) on the final list of detected QTLs.
3. Estimation of the QTL genetic effects and R squared statistics.
4. Optional plot (plot.MQE = TRUE) of the last CIM run of the forward regression using the function.

Value

Return:

List containing the following items:

n.QTL	Number of detected QTLs.
QTL	Data.frame with QTL positions.
R2	list containing R squared statistics of the QTL effects. for details see QTL_R2.
QTL.effects	List of genetic effects per QTL.

Some output files are also saved at the location specified (output.loc):

1. A QTL report (QTL_REPORT.txt) with: 1) the number of detected QTLs; 2) the global R squared statistics; 3) for each QTL, position information and estimated QTL genetic effect per cross or parents.
2. The list of QTLs (QTL.txt).
3. The QTL R squared statistics (QTL_R2.txt) (for details see QTL_R2).
4. General results of the QTL detection process: Number of QTL and global adjusted and non-adjusted R squared statistics. (QTL_genResults.txt).
5. if plot.MQE = TRUE, a plot of the last QTL detection run profile (plot_MQE.pdf).

Author(s)

Vincent Garin

See Also

mpp_perm, mpp_SIM, QTL_R2

Examples

```
data(mppData)

# Specify a location where your results will be saved
my.loc <- tempdir()

MQE <- MQE_proc(pop.name = "USNAM", trait.name = "ULA", mppData = mppData,
                Q.eff = c("par", "biall"), verbose = FALSE,
                output.loc = my.loc)
```

`parent_cluster.mppData`

Parent clustering for `mppData` *objects*

Description

Integrate the parent clustering information to the mppData object. The parent clustering is necessary to compute the ancestral model. If the parent clustering step is skipped, the ancestral model can not be used but the other models (cross-specific, parental, and bi-allelic) can still be computed.

Usage

```
parent_cluster.mppData(mppData, par.clu = NULL)
```

Arguments

`mppData`	An object of class `mppData`. the `mppData` must have been processed using: `create.mppData`, `QC.mppData`, `IBS.mppData`, and `IBD.mppData`.
`par.clu`	`Interger matrix` representing the results of a parents genotypes clustering. The columns represent the parental lines and the rows the markers. The columns names must be the same as the parents list of the mppData object. The rownames must be the same as the map marker list of the mppData object. At a particular position, parents with the same value are assumed to inherit from the same ancestor. for more details, see `par_clu`. Default = NULL.

Details

At a single marker position, two parents can be grouped into a similar ancestral classes if we assume that they receive there allele from a common ancetor. The parent clustering information (`par.clu`) describe parental relatedness and which parent belong to which ancestral group. For example, at marker i, we could have five parents (pA, pB, pC, pD, pE) and the following clustering information (1, 2, 1, 2, 3). This means that pA and pC received their allele from the same ancetor (A1). pB and pD also have a shared ancestor (A2) who is different from (A1). And pE was not included in any group and can be seen as an indepedent ancestral group (A3).

The parent clustering information is provided via `par.clu`. It is an `interger matrix` with markers in row and parents in columns. At a particular marker position, parents with the same value are assumed to inherit from the same ancestor. for more details, see `par_clu`.

The parent clustering can be performed using the R package 'clusthaplo' that can be found there: https://cran.r-project.org/src/contrib/Archive/clusthaplo/. The 'clusthaplo' option is not integrated in this version of mppR. However, a version of mppR with function calling clusthaplo can be found on github https://github.com/vincentgarin/mppR (branch mppR_clusthaplo).

Value

An increased `mppData` object containing the the same elements as the `mppData` object provided as argument and the following new elements:

`par.clu`	`Integer matrix` with rows repersenting markers and columns corresponding to the parents. At a single marker position, parents with the same value were clustered in the same ancestral group.
`n.anc`	Average number of ancestral clusters along the genome.
`mono.anc`	Positions for which the ancestral clustering was monomorphic.

Author(s)

Vincent Garin

See Also

`create.mppData`, `QC.mppData`, `IBS.mppData`, `IBD.mppData`, `par_clu`

Examples

```
data(mppData_init)
data(par_clu)

mppData <- QC.mppData(mppData_init)
mppData <- IBS.mppData(mppData = mppData)

mppData <- IBD.mppData(mppData = mppData, type = 'RIL',
                       type.mating = 'selfing')

mppData <- parent_cluster.mppData(mppData = mppData, par.clu  = par_clu)
```

par_clu *Parental clustering*

Description

Example of parental clustering object.

Usage

```
data(par_clu)
```

Details

The parent clustering matrix specifies at each genome position the results of a parent clustering into ancestral groups. The matrix rows represent the position and the columns correspond to each parent. For example, if we have at the ith row (1, 2, 3, 2, 1), this means that parents 1 and 5 are in the same group, that 2 and 4 are in another one and that the third parent was assigned to any group.

See Also

parent_cluster.mppData

Examples

```
data(par_clu)
```

plot.QTLprof *plot QTL profile*

Description

Plots the -log10(p-val) profile of a QTL analysis or a genome-wide genetic effect plot using package ggplot2.

Usage

```
## S3 method for class 'QTLprof'
plot(x, gen.eff = FALSE, mppData, Q.eff, QTL = NULL,
  type = "l", main = "QTL profile", threshold = 3, text.size = 18,
  ...)
```

Arguments

x	Object of class QTLprof returned by the function mpp_SIM or mpp_CIM.
gen.eff	Logical. Specify the type of plot. If gen.eff = FALSE, standard QTL profile. If gen.eff = TRUE, genome-wide genetic effect plot. In that case, the QTLprof object in x must have been calculated with argument plot.gen.eff = TRUE. Default = FALSE.
mppData	An object of class mppData. Only required if gen.eff = TRUE.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental effects; 3) "anc" for ancestral effects. Only required if gen.eff = TRUE
QTL	Optional argument. List of QTL positions. Object of class QTLlist representing a list of selected position obtained with the function QTL_select or two columns numeric matrix with the chromosome and the position in cM. These positions will be drawn on the graph. Default = NULL.
type	Character expression indicating the type of plot should be drawn: "l" for lines , "h" for vertical bar. Default = "l".
main	Title of the graph. Default = "QTL profile".
threshold	Numeric QTL significance threshold value draw on the plot. Default = 3.
text.size	Numeric value specifying the size of graph axis text elements. Default = 18.
...	Ignored.

Details

The user can plot regular QTL profiles (gen.eff = FALSE) with -log10(p-val) plotted against genetic position or genome-wide genetic effects plots (gen.eff = TRUE). To plot the genome-wide genetic effects, the SIM and CIM QTL profile must have been computed with plot.gen.eff = TRUE.

The genome-wide genetic effects plots is a visualisation of the significance of the QTL effect per cross or per parents along the genome. For a cross-specific QTL profile (Q.eff = "cr"): Blue color means that the allele coming from parent A(1) increases the phenotypic value and parent B(2) decreases it and red that parent A(1) decreases the trait and parent B(2) increases it.

For a parental (Q.eff = "par") or an ancestral model (Q.eff = "anc"), the results are given per parents. The significance of the effect must be interpreted as a deviation with respect to the reference of each connected part. The reference allele is always defined as the most frequent one. Blue (Red) colour means a signicative negative (positive) effect with respect to the reference of the connected part.

The reference parental allele can change at each position according to the segregation rate. The parent are plotted from the top to the bottom according to the number of time their allele is set as reference. Therefore interpretation of the genetic effect plot should be done with caution. In that case, the plot should be taken as a rough indication of the signal distribution.

The colour intensity increase with the significance of the effect (p-val). The p-val are transformed into a color code (z). If p-val c [0.00001; 0.05]: z = -log10(p-val). If p-val < 0.00001: z=6. This scale allows to plot only the significant effects (p-val <= 0.05) and prevent the color scale to be determine by highly significant values (p-val < 0.00001). The colours red (positive) and blue (negative) correspond to the sign of the QTL effect.

For both type of plot, the user can pass a list of cofactors or QTL position to the argument QTL. These positions will be drawn on the graph using dotted lines.

Author(s)

Vincent Garin

See Also

mpp_SIM, mpp_CIM, QTL_select

Examples

```
data(mppData)

SIM <- mpp_SIM(mppData = mppData)
QTL <- QTL_select(SIM)
plot(x = SIM, QTL = QTL)

SIM <- mpp_SIM(mppData = mppData, Q.eff = "cr", plot.gen.eff = TRUE)
QTL <- QTL_select(SIM)
plot(x = SIM, gen.eff = TRUE, mppData = mppData, Q.eff = "cr", QTL = QTL)
```

---

print.summary.mppData *Print summary.mppData object*

Description

Print summary.mppData object

Usage

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

Arguments

x	object of class summary.mppData
...	Ignored.

Examples

```
data(mppData)
sum.mppData <- summary(mppData)
print(sum.mppData)
```

print.summary.QeffRes *Print summary.QeffRes object*

Description

Print summary.QeffRes object

Usage

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

Arguments

x	object of class summary.QeffRes
...	Ignored.

Examples

```
data(mppData)
SIM <- mpp_SIM(mppData)
QTL <- QTL_select(SIM)
QTL.effects <- QTL_gen_effects(mppData = mppData, QTL = QTL, Q.eff = "cr")
sum.QeffRes <- summary(QTL.effects)
print(sum.QeffRes)
```

print.summary.QR2Res *Print summary.QR2Res object*

Description

Print summary.QR2Res object

Usage

```
## S3 method for class 'summary.QR2Res'
print(x, ...)
```

Arguments

x	object of class summary.QR2Res
...	Ignored.

Examples

```
data(mppData)
SIM <- mpp_SIM(mppData)
QTL <- QTL_select(SIM)
Q_R2 <- QTL_R2(mppData = mppData, QTL = QTL, Q.eff = "cr")
sum.QR2Res <- summary(Q_R2)
print(sum.QR2Res)
```

`QC.mppData` *Quality control for* `mppData` *objects*

Description

Perform different operations of quality control (QC) on the marker data of an `mppData` object.

Usage

```
QC.mppData(mppData, mk.miss = 0.1, gen.miss = 0.25, n.lim = 15,
  MAF.pop.lim = 0.05, MAF.cr.lim = NULL, MAF.cr.miss = TRUE,
  MAF.cr.lim2 = NULL, verbose = TRUE, n.cores = 1)
```

Arguments

`mppData`	An object of class `mppData` formed with `create.mppData`.
`mk.miss`	`Numeric` maximum marker missing rate at the whole population level comprised between 0 and 1. Default = 0.1.
`gen.miss`	`Numeric` maximum genotype missing rate at the whole population level comprised between 0 and 1. Default = 0.25.
`n.lim`	`Numeric` value specifying the minimum cross size. Default = 15.
`MAF.pop.lim`	`Numeric` minimum marker minor allele frequency at the population level. Default = 0.05.
`MAF.cr.lim`	`Numeric vector` specifying the critical within cross MAF. Marker with a problematic segregation rate in at least one cross is either set as missing within the problematic cross (`MAF.cr.miss = TRUE`), or remove from the marker matrix (`MAF.cr.miss = FALSE`). For default value see details.
`MAF.cr.miss`	`Logical` value specifying if maker with a too low segregation rate within cross (`MAF.cr.lim`) should be put as missing (`MAF.cr.miss = TRUE`) or discarded (`MAF.cr.miss = FALSE`). Default = TRUE.
`MAF.cr.lim2`	`Numeric`. Alternative option for marker MAF filtering. Only markers segregating with a MAF larger than `MAF.cr.lim2` in at least one cross will be kept for the analysis. Default = NULL.
`verbose`	`Logical` value indicating if the steps of the QC should be printed. Default = TRUE.
`n.cores`	`Numeric`. Specify here the number of cores you like to use. Default = 1.

Details

The different operations of the quality control are the following:

1. Remove markers with more than two alleles.
2. Remove markers that are monomorphic or fully missing in the parents.
3. Remove markers with a missing rate higher than `mk.miss`.
4. Remove genotypes with more missing markers than `gen.miss`.
5. Remove crosses with less than `n.lim` genotypes.
6. Keep only the most polymorphic marker when multiple markers map at the same position.
7. Check marker minor allele frequency (MAF). Different strategy can be used to control marker MAF:

   A) A first possibility is to filter marker based on MAF at the whole population level using `MAF.pop.lim`, and/or on MAF within crosses using `MAF.cr.lim`.

   The user can give the its own vector of critical values for MAF within cross using `MAF.cr.lim`. By default, the within cross MAF values are defined by the following function of the cross-size n.ci: MAF(n.ci) = 0.5 if n.ci c [0, 10] and MAF(n.ci) = (4.5/n.ci) + 0.05 if n.ci > 10. This means that up to 10 genotypes, the critical within cross MAF is set to 50 decreases when the number of genotype increases until 5

   If the within cross MAF is below the limit in at least one cross, then marker scores of the problematic cross are either put as missing (`MAF.cr.miss = TRUE`) or the whole marker is discarded (`MAF.cr.miss = FALSE`). By default, `MAF.cr.miss = TRUE` which allows to include a larger number of markers and to cover a wider genetic diversity.

   B) An alternative is to select only markers that segregate in at least on cross at the `MAF.cr.lim2` rate.

Value

a filtered `mppData` object containing the the same elements as `create.mppData` after filtering. It contains also the following new elements:

`geno.id`	`Character` vector of genotpes identifiers.
`ped.mat`	Four columns `data.frame`: 1) the type of genotype: "offspring" for the last genration and "founder" for the genotypes above the offspring in the pedigree; 2) the genotype indicator; 3-4) the parent 1 (2) of each line.
`geno.par.clu`	Parent marker matrix without monomorphic or completely missing markers.
`haplo.map`	Genetic map corresponding to the list of marker of the `geno.par.clu` object.
`parents`	List of parents.
`n.cr`	Number of crosses.
`n.par`	Number of parents.
`rem.mk`	Vector of markers that have been removed.
`rem.geno`	Vector of genotypes that have been removed.

Author(s)

Vincent Garin

See Also

create.mppData

Examples

```
data(mppData_init)

mppData <- QC.mppData(mppData = mppData_init, n.lim = 15, MAF.pop.lim = 0.05,
                      MAF.cr.miss = TRUE, mk.miss = 0.1,
                      gen.miss = 0.25, verbose = TRUE)
```

QTL_gen_effects *QTL genetic effects*

Description

Computes a multi-QTL model with a list of QTL candidates (QTL) and return the decomposed QTL effects per cross or per parents.

Usage

```
QTL_gen_effects(mppData, trait = 1, QTL = NULL, Q.eff = "cr",
  ref.par = NULL, sum_zero = FALSE)
```

Arguments

mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
QTL	Object of class QTLlist representing a list of selected position obtained with the function QTL_select or vector of character marker positions names. Default = NULL.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
ref.par	Optional Character expression defining the parental allele that will be used as reference for the parental model. For the ancestral model, the ancestral class containing the reference parent will be set as reference. **This option can only be used if the MPP design is composed of a unique connected part**. Default = NULL.
sum_zero	Optional Logical value specifying if the QTL effect of a parental or an ancestral model should be calculated using the sum to zero constraint. Default = FALSE.

Details

This function computes for each QTL position the genetic effects of the cross, parental, ancestral or SNP allele components. For the cross-specific model (`Q.eff = "cr"`), the genetics effects represent the substitution effect of an single allele from the parent 2 (or B) with respect to an allele coming from the parent 1 or A. All effects are given in absolute value with the parent that carries the positive allele.

For the parental and the ancestral model (`Q.eff = "par"` or `"anc"`), it is possible to estimate maximum n-1 parental or ancestral alleles per interconnected part of the design. For these two models, one parental (ancestral) allele is set as reference per interconnected part of the design. Effects of the other alleles are estimated as deviation with respect to the reference. Connected parts of the design can be determined using Weeks and Williams (1964) method (`design_connectivity`). By default, the reference allele is the most frequent one. The user can also specify a parental allele that will be used as reference using the argument `ref.par`. This option is only available if the MPP design is composed of a unique connected part.

For the parental and ancestral model it is also possible to estimate the QTL effects using a sum to zero constraint `sum_zero = TRUE`. In that case, the effects of the different parental (ancestral) allele will represent the deviation with respect to the average trait value.

For the bi-allelic model (`Q.eff = "biall"`), the genetic effects represent the effects of a single allele copy of the least frequent allele.

Value

Return:

Object of class `QeffRes` containing the following elements:

`Qeff` `List` of `data.frame` (one per QTL) containing the following information:

1. QTL genetic effects per cross or parent.
2. Standard error of the QTL effects.
3. Test statistics of the effects (t-test or Wald statistic).
4. P-value of the test statistics.
5. Significance of the QTL effects.
6. For cross-specific model, parent with the positive additive effects.
7. For parental and ancestral model, indicator of connected part of the design and reference.
8. Allele scores of the parents if `geno.par` is non NULL in the `mppData` object.

`tab.Qeff` `data.frame` with one column per QTL giving the QTL genetic effects per cross or per parent with its significance. The first two rows indicate the chromosome and the position in cM of each QTL.

Author(s)

Vincent Garin

References

Weeks, D. L., & Williams, D. R. (1964). A note on the determination of connectedness in an N-way cross classification. Technometrics, 6(3), 319-324.

See Also

QTL_select, summary.QeffRes

Examples

```
data(mppData)

# QTL candidates

SIM <- mpp_SIM(mppData)
QTL <- QTL_select(SIM)

# Cross-specific model

QTL.effects <- QTL_gen_effects(mppData = mppData, QTL = QTL, Q.eff = "cr")
summary(QTL.effects)

# Parental model

QTL.effects <- QTL_gen_effects(mppData = mppData, QTL = QTL, Q.eff = "par")
summary(QTL.effects)

# Ancestral model

QTL.effects <- QTL_gen_effects(mppData = mppData, QTL = QTL, Q.eff = "anc")
summary(QTL.effects)

# Bi-allelic model

QTL.effects <- QTL_gen_effects(mppData = mppData, QTL = QTL, Q.eff = "biall")
summary(QTL.effects)
```

QTL_pred_R2 *Predicted QTL global and partial R squared*

Description

Compute predicted R squared in a validation set using QTLs detected in a training set. These values are corrected by the heritability her.

Usage

```
QTL_pred_R2(mppData.ts, mppData.vs, trait = 1, Q.eff = "cr",
  QTL = NULL, her = 1)
```

Arguments

mppData.ts	An object of class mppData for the training set.
mppData.vs	An object of class mppData for the validation set.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
QTL	Object of class QTLlist representing a list of selected position obtained with the function QTL_select or vector of character marker positions names. Default = NULL.
her	Numeric value between 0 and 1 representing the heritability of the trait. her can be a single value or a vector specifying each within cross heritability. Default = 1.

Details

Compute QTLs predicted R squared in a validation set (mppData.vs). These QTLs have been previously detected in a training set (mppData.ts). The global R squared (R2 = cor(y.ts,y.pred.ts)^2) is obtained using the Pearson squared correlation between the observed trait values in the validation set (y.vs) and predicted values using estimated QTL effects in the training set (y.pred.vs = X.vs * B.ts).

After that the values are corrected by the general or within cross heritability her. By default her = 1 which means that the R squared represent the proportion of explained phenotypic variance. The values are returned per cross (R2.cr) or averaged at the population level (glb.R2).

Partial R squared statistics are also calculated for each individual position. The partial R squared are computed by making the difference between the global R squared and the R squared computed without the ith position.

Value

Return:

List containing the following objects:

glb.R2	Global predicted R squared corrected for the heritability of all QTL terms. Doing the average of the within cross predicted R squared (R2.cr)
R2.cr	Within cross predicted R squared corrected for the heritability
part.R2.diff	Vector of predicted partial R squared corrected for the heritability doing the difference between the full model and a model minus the ith QTL.

Author(s)

Vincent Garin

See Also

QTL_R2, QTL_select

Examples

```
data(mppData)

folds <- CV_partition(cross.ind = mppData$cross.ind, k = 5)

mppData.ts <- subset(x = mppData, gen.list = folds[[1]]$train.set)

mppData.vs <- subset(x = mppData, gen.list = folds[[1]]$val.set)

SIM <- mpp_SIM(mppData = mppData)
QTL <- QTL_select(SIM)

QTL_pred_R2(mppData.ts = mppData.ts, mppData.vs = mppData.vs, QTL = QTL)
```

QTL_R2 *QTL global and partial R squared*

Description

Computes the global and partial (adjusted) R squared of a list of QTLs using a linear model.

Usage

```
QTL_R2(mppData, trait = 1, QTL = NULL, Q.eff = "cr",
  glb.only = FALSE)
```

Arguments

mppData	An object of class mppData.
trait	Numerical or character indicator to specify which trait of the mppData object should be used. Default = 1.
QTL	Object of class QTLlist representing a list of selected position obtained with the function QTL_select or vector of character marker positions names. Default = NULL.
Q.eff	Character expression indicating the assumption concerning the QTL effects: 1) "cr" for cross-specific; 2) "par" for parental; 3) "anc" for ancestral; 4) "biall" for a bi-allelic. For more details see mpp_SIM. Default = "cr".
glb.only	Logical value. If glb.only = TRUE, only the global and global adjusted R squared will be returned. Default = FALSE.

Details

The function computes R squared statistics using a linear model. The extra variance explained by a full model containing the QTL terms with respect to a reduced model containing only the cross intercept terms and uses the ratio between the residual sum of square of these two models: R2 = 1-(RSS(f))/(RSS(r)).

Partial R squared for each individual QTL position can also be calculated. Two types of partial R squared are returned. The first one uses the difference between the R squared obtained with all QTL positions and the R squared obtain with all position minus the ith one (difference R squared). The second method used only the ith QTL position in the model (single R squared).

For both global and partial R squared, it is possible to obtained adjusted measurements taking the number of degrees of freedom into consideration using an adaptation of the formula given by Utz et al. (2000): R.adj = R-(z/(N-z-n.cr))*(1-R) where z is the total number of estimated components of the genetic effect. N is the total number of phenotypic information, and n.cr is the number of intercept (cross) terms.

Value

Return:

object of class `QR2Res` containing the following objects:

`glb.R2`	Global R squared of all QTL terms.
`glb.adj.R2`	Global adjusted R squared of all QTL terms.
`part.R2.diff`	Vector of partial R squared doing the difference between the full model and a model minus the ith QTL.
`part.adj.R2.diff`	Vector of partial adjusted R squared doing the difference between the full model and a model minus the ith QTL.
`part.R2.sg`	Vector of partial R squared using only the ith QTL.
`part.adj.R2.sg`	Vector of partial adjusted R squared using only the ith QTL.

Author(s)

Vincent Garin

References

Utz, H. F., Melchinger, A. E., & Schon, C. C. (2000). Bias and sampling error of the estimated proportion of genotypic variance explained by quantitative trait loci determined from experimental data in maize using cross validation and validation with independent samples. Genetics, 154(4), 1839-1849.

See Also

`QTL_select`, `summary.QR2Res`

Examples

```
data(mppData)

SIM <- mpp_SIM(mppData)
QTL <- QTL_select(Qprof = SIM, threshold = 3, window = 20)
Q_R2 <- QTL_R2(mppData = mppData, QTL = QTL, Q.eff = "cr")
summary(Q_R2)
```

QTL_select *QTL candidates selection*

Description

Selection of QTL candidate positions.

Usage

```
QTL_select(Qprof, threshold = 3, window = 50, verbose = TRUE)
```

Arguments

Qprof	Object of class QTLprof returned by the function mpp_SIM or mpp_CIM.
threshold	Numeric value representing -log10(p-value) threshold above which a position can be considered as a QTL candidate. Default = 3.
window	Numeric value in centi-Morgan representing the minimum distance between two selected positions. Default = 50.
verbose	Logical value specifying if the detection of no QTL should be printed. Default = TRUE.

Details

The function select QTL positions that are above the given threshold per chromosome. Once a position has been selected, and exclusion window is set around that position. Positions falling into that region will not be candidate anymore. The search continue until there is no more candidate position above the threshold.

Value

Return:

QTL	Data.frame of class QTLlist with five columns : 1) QTL marker names; 2) chromosomes; 3) interger position indicators on the chromosome; 4) positions in centi-Morgan; and 5) -log10(p-values).

References

This function is a modification of the QTL.reduce function coming from the Biometris pipepline.

RAP (R Analytical Pipeline) (V0.9.1) May 2011

Authors: Paul Eilers (1), Gerrit Gort (1), Sabine Schnabel (1), Lucia Gutierrez(1, 2), Marcos Malosetti(1), Joost van Heerwaarden, and Fred van Eeuwijk(1)

(1) Wageningen University and Research Center, Netherlands (2) Facultad de Agronomia, UDELAR, Uruguay

See Also

mpp_SIM, mpp_CIM, mpp_perm

Examples

```
data(mppData)

SIM <- mpp_SIM(mppData)

QTL <- QTL_select(Qprof = SIM, threshold = 3)
```

`subset.mppData` *Subset* `mppData` *object*

Description

Pull out a specified set of markers and/or genotypes from a `mppData` object.

Usage

```
## S3 method for class 'mppData'
subset(x, mk.list = NULL, gen.list = NULL, ...)
```

Arguments

`x`	An object of class `mppData`.
`mk.list`	Optional `character` vector, `numeric` position vector or `logical` vector representing marker to keep. Default = NULL.
`gen.list`	Optional `character` vector, `numeric` position vector or `logical` vector representing genotypes to keep. Default = NULL.
`...`	Ignored.

Value

Return:

The mppData object but with only the specified subset of data.

Author(s)

Vincent Garin

Examples

```
### Marker subset

data(mppData)

# Random selection of markers
mk.list <-  sample(mppData$map[, 1], 50)
mppData_sub <- subset(x = mppData, mk.list = mk.list)

# Selection of chromosome 1 marker
mk.list <-  (mppData$map[, 2] == 1)
mppData_sub <- subset(x = mppData, mk.list = mk.list)

### Genotype subset

# Random selection of genotypes
gen.list <-  sample(mppData$geno.id, 200)
mppData_sub <- subset(x = mppData, gen.list = gen.list)

# Selection of genotype from cross 2 and 5
crosses <- unique(mppData$cross.ind)
gen.list <-  mppData$geno.id[mppData$cross.ind %in% crosses[c(2, 5)]]
mppData_sub <- subset(x = mppData, gen.list = gen.list)

### Marker and genotype subset

mk.list <-  sample(mppData$map[, 1], 50)
gen.list <-  sample(mppData$geno.id, 200)
mppData_sub <- subset(x = mppData, mk.list = mk.list,
gen.list = gen.list)
```

---

summary.mppData *Summary of* mppData *object*

Description

summary for object of class mppData.

Usage

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

Arguments

object	An object of class mppData.
...	Ignored.

Examples

```
data(mppData)
summary(mppData)
```

---

summary.QeffRes *Summary of* QeffRes *object*

Description

summary for object of class QeffRes.

Usage

```
## S3 method for class 'QeffRes'
summary(object, QTL = NULL, ...)
```

Arguments

object	An object of class QeffRes obtained with function QTL_gen_effects.
QTL	Numeric vector indicating the QTL positions for which the QTL effect must be printed. Default = NULL.
...	Ignored.

See Also

QTL_gen_effects

Examples

```
data(mppData)
SIM <- mpp_SIM(mppData)
QTL <- QTL_select(SIM)
QTL.effects <- QTL_gen_effects(mppData = mppData, QTL = QTL, Q.eff = "cr")
summary(QTL.effects)
```

summary.QR2Res *Summary of* QR2Res *object*

Description

summary for object of class QR2Res.

Usage

```
## S3 method for class 'QR2Res'
summary(object, ...)
```

Arguments

object	An object of class QR2Res obtained with function QTL_R2.
...	Ignored.

See Also

QTL_R2

Examples

```
data(mppData)
SIM <- mpp_SIM(mppData)
QTL <- QTL_select(SIM)
Q_R2 <- QTL_R2(mppData = mppData, QTL = QTL, Q.eff = "cr")
summary(Q_R2)
```

USNAM_geno *Reduced genotype data maize US-NAM population*

Description

Selection of markers and genotypes from the maize US nested association mapping (NAM) population (McMullen et al., 2009).

Usage

```
data(USNAM_geno)
```

Format

data.frame

Details

Sample of the marker matrix of the US-NAM population. The selection correspond to 102 markers coming from the two first chromosomes present in USNAM_map and the 506 genotypes. These genotypes correspond to the selected phenotypic values in USNAM_pheno. The selected genotypes come from the following crosses: (B73 x CML103), (B73 x CML322), (B73 x CML52), (B73 x Hp301), (B73 x M37W). The data of the 6 parental lines are also included. The data are available on www.panzea.org.

Source

http://www.panzea.org

References

McMullen, M. D., Kresovich, S., Villeda, H. S., Bradbury, P., Li, H., Sun, Q., ... & Buckler, E. S. (2009). Genetic properties of the maize nested association mapping population. Science, 325(5941), 737-740.

See Also

USNAM_pheno, USNAM_map

Examples

```
data(USNAM_geno)
```

USNAM_map *Reduced map maize US-NAM population*

Description

Reduced map of the maize US nested association mapping (NAM) population (McMullen et al., 2009).

Usage

```
data(USNAM_map)
```

Format

data.frame

Details

Selection of 102 markers from the two first chromosomes of the Maize US-NAM population (McCullen et al., 2009). The data are available on www.panzea.org.

Source

http://www.panzea.org

References

McMullen, M. D., Kresovich, S., Villeda, H. S., Bradbury, P., Li, H., Sun, Q., ... & Buckler, E. S. (2009). Genetic properties of the maize nested association mapping population. Science, 325(5941), 737-740.

See Also

USNAM_geno, USNAM_pheno

Examples

```
data(USNAM_map)
```

USNAM_pheno *Reduced phenotype data from Maize US-NAM population*

Description

Reduced phenotype data from the Maize US nested association mapping (NAM) population (McMullen et al., 2009).

Usage

```
data(USNAM_pheno)
```

Format

data.frame

Details

Upper leaf angle (ULA) trait values with genotypes identifiers as rownames. These genotypes correspond to the 500 offspring genotypes of the marker matrix USNAM_geno. The data are available on www.panzea.org.

Source

http://www.panzea.org

References

McMullen, M. D., Kresovich, S., Villeda, H. S., Bradbury, P., Li, H., Sun, Q., ... & Buckler, E. S. (2009). Genetic properties of the maize nested association mapping population. Science, 325(5941), 737-740.

See Also

USNAM_geno, USNAM_map

Examples

```
data(USNAM_pheno)
```

Index

∗Topic **datasets**
mppData, 11
mppData_init, 12
par_clu, 30
USNAM_geno, 46
USNAM_map, 47
USNAM_pheno, 48

create.mppData, 2, *6*, *8, 9*, *11, 12*, *28, 29*, *34–36*
CV_partition, 4

design_connectivity, 5, *24*, *37*

IBD.mppData, 6, *11, 12*, *28, 29*
IBS.mppData, *6*, *8*, 8, *11, 12*, *28, 29*
inc_mat_QTL, 9

mpp_back_elim, 13, *17, 18*, *22, 23*
mpp_CIM, 14, *17, 18*, *22, 23*, *31, 32*, *42, 43*
mpp_CV, *5*, 16
mpp_perm, *18*, 19, *23*, *27*, *43*
mpp_proc, 20
mpp_SIM, *10*, *13–23*, 23, *26, 27*, *31, 32*, *36*, *39, 40*, *42, 43*
mppData, 11
mppData_init, 12
MQE_proc, 26

par_clu, *28, 29*, 30
parent_cluster.mppData, *11, 12*, 28, *30*
plot.QTLprof, *14*, *21–25*, 30
print.summary.mppData, 32
print.summary.QeffRes, 33
print.summary.QR2Res, 33

QC.mppData, *6*, *8, 9*, *11, 12*, *28, 29*, 34
QTL_gen_effects, *22, 23*, 36, *45*
QTL_pred_R2, 38
QTL_R2, *17, 18*, *22, 23*, *27*, *40*, 40, *46*
QTL_select, *13–15*, *31, 32*, *36*, *38–41*, 42

subset.mppData, 43
summary.mppData, 44
summary.QeffRes, *38*, 45
summary.QR2Res, *41*, 46

USNAM_geno, 46, *48, 49*
USNAM_map, *47*, 47, *49*
USNAM_pheno, *47, 48*, 48