Package 'metaGE'

February 23, 2025

Title Meta-Analysis for Detecting Genotype x Environment Associations

Version 1.2.1

Maintainer Annaïg De Walsche <annaig.de-walsche@inrae.fr>

Description Provides functions to perform all steps of genome-wide association meta-analysis for studying Genotype x Environment interactions, from collecting the data to the manhattan plot. The procedure accounts for the potential correlation between studies. In addition to the Fixed and Random models, one can investigate the relationship between QTL effects and some qualitative or quantitative covariate via the test of contrast and the meta-regression, respectively. The methodology is available from: (De Walsche, A., et al. (2025) \doi{10.1371/journal.pgen.1011553}).

License GPL-3

Depends R (>= 3.0.2)

Imports corrplot, data.table, dplyr, emdbook, furrr, future, ggplot2, ggrepel, gplots, graphics, grDevices, ks, purrr, qqman, Rfast, stats, stringr, tibble, tidyr, utils, viridis, yarrr

Suggests DT, knitr, rmarkdown, tidyverse

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Annaïg De Walsche [aut, cre] (<https://orcid.org/0000-0003-0603-1716>), Tristan Mary-Huard [aut] (<https://orcid.org/0000-0002-3839-9067>)

Repository CRAN

Date/Publication 2025-02-22 23:10:02 UTC

Contents

`autocor` *Computation of the autocorrelation*

Description

The function autocor computes the autocorrelation.(function from localscore)

Usage

```
autocor(x)
```

Arguments

x	A numeric vector.

Value

the autocorrelation.

CheckContrast *Check and reformat the matrix of contrast*

Description

The function CheckContrast check and reformat the matrix of contrast.

Usage

```
CheckContrast(Contrast, ContrastName)
```

Arguments

Contrast	A matrix of contrast.
ContrastName	The name of the contrast.

Value

The matrix of contrast in the right format.

CheckIncidence *Check and reformat the matrix of incidence*

Description

The function CheckIncidence check and reformat the matrix of incidence.

Usage

```
CheckIncidence(Incidence, IncidenceName)
```

Arguments

Incidence	A matrix of incidence, as obtained from metaGE.incidence().
IncidenceName	The name of the incidence.

Value

The matrix of incidence in the right format.

ContrastStatTest *Compute the statistic of the contrast test.*

Description

The function ContrastStatTest compute the statistic of the contrast test.

Usage

```
ContrastStatTest(Incidence, Contrast = NULL, Zmat, MatCorr, IncidenceName)
```

Arguments

Incidence	A matrix of incidence, as obtained from metaGE.incidence().
Contrast	A matrix of contrast, if NULL the identity matrix is used. (NULL by default)
Zmat	A matrix containing the Zscores of all markers (in rows) in each environment (in columns).
MatCorr	The inter-environments correlation matrix. Can be computed using metaGE.cor().
IncidenceName	The name of the incidence.

Value

A dataset of two columns containing the pvalue of the test of contrast and the minimum number of environment per group of all markers.

ContrastStatTest.NA *Compute the statistic of the contrast test in presence of missing values*

Description

The function ContrastStatTest compute the statistic of the contrast test.

Usage

```
ContrastStatTest.NA(
  Incidence,
  Contrast = NULL,
  Zmat,
  MatCorr,
  Data,
  Configs.list,
  IncidenceName
)
```

Arguments

Incidence	A matrix of incidence, as obtained from metaGE.incidence.
Contrast	A matrix of contrast, if NULL the identity matrix is used. (NULL by default)
Zmat	A matrix containing the Zscores of all markers (in rows) in each environment (in columns).
MatCorr	The inter-environments correlation matrix. Can be computed using metaGE.cor().
Data	A dataset containing the effect, the pvalues and the na configuration for all marker
Configs.list	A vector containing the NA configurations present in the dataset
IncidenceName	The name of the incidence.

Value

A dataset of two columns containing the pvalue of the test of contrast and the minimum number of environment per group of all markers.

---

envDesc *Description of the environments.*

Description

A dataset containing variables describing the 22 environments.

Usage

```
envDesc
```

Format

A data frame with 22 rows and 3 variables:

- FileName: environment name
- Temp: temperature
- Water: water condition

FastKerFdr *FastKerFdr*

Description

Computes H1 posteriors of the Z-scores.

Usage

```
FastKerFdr(
  Z,
  p0,
  plotting = FALSE,
  NbKnot = 1e+05,
  tol = 1e-05,
  max_iter = 10000
)
```

Arguments

Z	A vector containing Zscores
p0	A double between 0 and 1. A priori proportion of H0 hypotheses
plotting	A boolean saying to plot or not (FALSE by default)
NbKnot	The (maximum) number of knot for the kde procedure.(1e5 by default)
tol	a tolerance value for convergence (1e-5 by default)
max_iter	the maximum number of iterations allowed for the algorithm to converge or complete its process.(Default is 1e4.)

Value

tau is the vector of H1 posteriors

GetH0Items *GetH0Items*

Description

This function give the index of the markers which seems not significant (under H0)

Usage

```
GetH0Items(Zmat, Threshold = 0.8, plotting = FALSE, Cores = NULL)
```

Arguments

Zmat	A matrix containing the Zscore (in rows) for each environment (in columns)
Threshold	Threshold on posteriors (to be H1) to filter markers for correlation computation (0.6 by default)
plotting	A boolean saying to plot or not (FALSE by default)
Cores	The number of cores to used, optional. By default, availableCores()-1 cores is used.

Value

A vector of index of markers which seems not significant (under H0)

---

lindley *Computation of the lindley process from scores.*

Description

The function lindley computes the lindley process from scores.(function from localscore)

Usage

```
lindley(scores)
```

Arguments

scores	A numeric vector.

Value

the lindley.

LLikelihoodT_vect *LLikelihoodT_vect*

Description

This function compute the values of loglikelihood for all markers.

Usage

```
LLikelihoodT_vect(Zmat, Delta, P, Mu, Tau)
```

Arguments

Zmat	A matrix containing the Zscores of all markers (in rows) in each environment (in columns)
Delta	A vector containing the diagonal coefficients of the diagonal matrix obtained by the diagonalization of the correlation matrix
P	Matrix such that MatCorr = P Delta t(P), with Delta diagonal
Mu	A vector containing the average effect of the markers
Tau	A vector containing the heterogeneity between environments of the markers

Value

A vector containing the value of the Log-Likelihood of all markers

---

MakeQQplot *Drawing a QQplot*

Description

The function MakeQQplot displays the QQplot of the -log10(pvalues).

Usage

```
MakeQQplot(Pvalues, Name = NULL, Xrange = NULL, Yrange = NULL)
```

Arguments

Pvalues	A vector containing pvalues.
Name	A name of the corresponding test. (optional)
Xrange	A range for the x axis. (optional)
Yrange	A range for the y axis. (optional)

---

metaData *Results of different GWAS.*

Description

A dataset containing the results of 10 different genetic association studies testing the association between a set of 25,436 markers and the grain yield. The data are extrated from: Drops Amaizing available on the https://doi.org/10.15454/6TL2N4 website. This dataset were obtained thanks to the *metaGE.collect* function.

Usage

```
metaData
```

Format

A data frame with 25,436 rows and 35 variables:

- CHR: chromosome of the marker
- POS: position of the marker
- MARKER: name of the marker
- FREQ.env: maf of the marker in the environment env
- EFFECT.env: regression coefficient of the marker in the environment env
- EFFECT_SE.env: standard error of the regression coefficient of the marker in the environment env
- PVAL.env: pvalue of the marker in the environment env
- WEIGHT.env: weight of the marker in the environment env
- ALLELE0: allele0
- ALLELE1: allele1

---

metaGE.collect *Collect the results of GWAS data from different files*

Description

This function merges files containing the summary statistics of GWAS in different environments (one file per environment).

Usage

```
metaGE.collect(
  FileNames,
  VariableNames,
  MinFreq = 0,
  DropDuplicates = TRUE,
  Verbose = FALSE,
  NA.rmv = TRUE
)
```

Arguments

FileNames	A list containing the file paths to merge (one trait only) or a list of such lists
VariableNames	A named list containing the column names in the original files corresponding to the variables : MARKER, CHR, POS, EFFECT, PVAL (optional: FREQ, ALLELE0, ALLELE1) ; or a list of such lists.
MinFreq	A numeric value allowing to filter markers based on the maf. (optional)
DropDuplicates	A boolean indicating whether duplicate markers should be removed or not. (TRUE by default)
Verbose	A boolean indicating whether progression messages should be printed or not. (FALSE by default)
NA.rmv	A boolean indicating if the NA should be removed or not (TRUE by default)

Details

Each file MUST contain the variables below:

- MARKER: the marker name
- CHR: the chromosome
- POS: the position of the marker
- EFFECT: the mean effect of the marker
- PVAL: the pvalue

Each file might contain the variables:

- FREQ: MAF
- ALLELE0: Allele coding for allele 0
- ALLELE1: Allele coding for allele 1

Value

A list with the following elements:

Data	A tibble containing all the columns of interest of all the files from FileNames.
RemovedMarkers	Same kind of tibble, but containing the markers that have been removed due to unclear allele coding, maf

Examples

```
require(dplyr)
require(tibble)
require(stringr)
RepData <- system.file("extdata", package = "metaGE")
# Get the complete list of association files
File.list <- list.files(RepData ,full.names = TRUE) %>%
            tibble(Names = .) %>%
            mutate(ShortNames = Names %>%
                  str_remove(pattern = paste0(RepData,"/")) %>%
                  str_remove(pattern = "_DF.txt"))  %>%
            select(ShortNames,Names) %>%
            deframe
###Build the dataset
## First provide the list of variable names
Names.list <- list(MARKER="Marker_Name",
                  CHR="Chromosome",
                  POS="Marker_Position",
                  FREQ="Maf",
                  EFFECT="SNP_Weight",
                  PVAL="Pvalue",
                  ALLELE0="Allele1",
                  ALLELE1="Allele2")

MinFreq <- 0.07

## Now collect
metaData <- metaGE.collect(File.list, Names.list,MinFreq = MinFreq)
```

metaGE.cor *Infer inter-environment correlation matrix*

Description

This function infer the inter-environment correlation matrix from the z-scores after filtering markers with high probability of being under H1.

Usage

```
metaGE.cor(Data, Threshold = 0.6, NA.omit = TRUE, Cores = NULL)
```

Arguments

Data	A dataset containing the effects and pvalues of each marker (in rows) in each environment (in columns) as obtained by metaGE.collect().
Threshold	Threshold on posteriors (to be H1) to filter markers before computing correlation (0.6 by default).
NA.omit	A boolean: should the NA be removed for the inter-environment correlation matrix computation (TRUE by default).
Cores	The number of cores to used, optional. By default, availableCores()-1 cores is used.

Value

The inter-environment correlation matrix

Examples

```
require(corrplot)
data("metaData")
Threshold <- 0.8
matCorr <- metaGE.cor(metaData, Threshold = Threshold)
corrplot(matCorr,order = "hclust")
```

metaGE.fit *Meta-analysis procedure: Fixed or Random effect.*

Description

Quantitative trait loci detection via Fixed or Random effect meta-analysis GWAS procedure.

Usage

```
metaGE.fit(Data, MatCorr, Method, NA.omit = TRUE, DropZScores = FALSE)
```

Arguments

Data	A dataset containing the estimated marker effect and its associated pvalue of each marker (in rows) in each environment (in columns), as obtained from metaGE.collect().
MatCorr	The inter-environments correlation matrix. Can be computed using metaGE.cor().
Method	A string specifying the method to be performed: either "Fe" or "Re".
NA.omit	A boolean specifying whether the markers with some NA values should be removed. (TRUE by default)
DropZScores	A boolean specifying whether the Zscores should be dropped from the dataset or not.(FALSE by default)

Details

Different tests may be performed:

- Fixed Effect (Fe), to identify markers with a stable effect across environments.
- Random Effect (Re), to identify markers whose effects may be unstable across environments.

Value

The dataset Data with supplementary columns:

- PVALUE: The pvalue of the MA test,
- Mu: Estimate of the mean marker effect,
- Tau: Estimate of the variance of the marker effect, for the Random model only,
- the Zscores for each environment if DropZScores = FALSE.

Examples

```
require(dplyr)
# Import the data
data("metaData")

# Compute the inter-environment correlation matrix
matCorr <- metaGE.cor(metaData, Threshold = 0.8)

# Fixed Effect
FeDF <- metaGE.fit(metaData, matCorr, Method = "Fe")
head(FeDF %>% select(CHR, POS, MARKER, Mu, Tau, PVALUE))

# Random Effect
ReDF <- metaGE.fit(metaData, matCorr, Method = "Re")
head(ReDF %>% select(CHR, POS, MARKER, Mu, Tau, PVALUE))
```

`metaGE.heatmap` *Draw the heatmap to see markers effects across environments.*

Description

The function metaGE.heatmap displays the heatmap of the zscores, the estimated marker effects or the pvalues of each markers (in rows) in each environments (in columns).

Usage

```
metaGE.heatmap(
  Data,
  Prefix = "Z.",
  EnvGroups = NULL,
  QTLsVarName = NULL,
  RowOrder = TRUE,
  ColOrder = TRUE,
  ShowDendrogram = FALSE,
  Colors = c("red", "black", "green"),
  Main = ""
)
```

Arguments

`Data`	A dataset containing the zscores, the effects or the pvalues of each marker (in rows) in each environment (in columns), as obtained from `metaGE.fit()`.
`Prefix`	The prefix of the score to display in the heatmap: `"Z."` for the zscores, `"EFFECT."` for the effects and `"PVAL."` for the pvalues.(`"Z."` by default)
`EnvGroups`	A dataset containing the names of the environments (in the first column) and the groups to which the environments belong (in the second column). (optional)
`QTLsVarName`	The name of the column indicating to which QTL the marker belongs. (optional)
`RowOrder`	A boolean specifying whether to reorder the markers or not. (`TRUE` by default)
`ColOrder`	A boolean specifying whether to reorder the environments or not. (`TRUE` by default)
`ShowDendrogram`	A boolean specifying whether to show the clustering of the rows and/or the columns. (`FALSE` by default)
`Colors`	A vector of three colors corresponding to the color scale of the Heatmap.(optional)
`Main`	The main to display.(optional)

Value

The heatmap

Examples

```
require(dplyr)
# Import the data
data("metaData")

# Compute the inter-environment correlation matrix
matCorr <- metaGE.cor(metaData, Threshold = 0.8)

# Fit the Fixed Effect model
FeDF <- metaGE.fit(metaData, matCorr, Method = "Fe")

# Control the FDR (here Benjamini-Hochberg)
Alpha <- 0.05
Signif <-  FeDF$PVALUE %>% p.adjust(method = "BH") %>% `<`(Alpha) %>% which

# Draw the z-scores heatmap of the significant markers
heatmap <- metaGE.heatmap(Data = FeDF[Signif,],
                          Prefix = "Z.")
```

`metaGE.incidence` *Create the matrix of incidence.*

Description

The function metaGE.incidence convert categorical variable describing the environments into a matrix of dummy variables with in rows the levels of the variables and in columns the environment.

Usage

```
metaGE.incidence(VarName, Covariate, EnvName, Data, AtLeast = 1)
```

Arguments

`VarName`	The name of the column containing the categorical variable in the Covariate dataset.
`Covariate`	A dataset containing categorical variables (in columns) describing the environments (in rows).
`EnvName`	The name of the column containing the names of the environment in the Covariate dataset.
`Data`	A dataset containing the effects and pvalues of each marker (in rows) in each environment (in columns), as obtained from `metaGE.collect()`.
`AtLeast`	A numeric value indicating the minimum number of environments must belong to each level (equals 1 by default).

Details

The names of the environment must be the same as used in the Data dataset.

Value

A binary matrix containing indicator variables with in rows the levels of the variables and in columns the environment.

Examples

```
# Import the data
data("metaData")
data("envDesc")

# Build the matrix of incidence
(Incidence.Temp <- metaGE.incidence(VarName = "Temp",Covariate = envDesc,
                                    EnvName = "ShortName", Data = metaData))
```

`metaGE.lscore` *Compute the local score from a set of pvalues.*

Description

The function metaGE.lscore computes the local score and the significant regions from a set of pvalues.

Usage

```
metaGE.lscore(Data, PvalName, xi)
```

Arguments

`Data`	A dataset containing the following columns: CHR, POS, MARKER and `PvalName`.
`PvalName`	The name of the column containing the p-value.
`xi`	The threshold of the score, `xi` = 1,2,3 or 4.

Details

This function is directly inherited from the scorelocalfunctions.R R code file of Fariello MI, Boitard S, Mercier S, et al., as available on the https://forge-dga.jouy.inra.fr/projects/local-score website. The technical details of the computation can be found in Fariello MI, Boitard S, Mercier S, et al. Accounting for linkage disequilibrium in genome scans for selection without individual genotypes: The local score approach. doi:10.1111/mec.14141. The function computes a local score for the detection of significant regions based on the hypothesis that the H0 distribution of the pvalues is uniform. Under this hypothesis the local score follows a Gumbel distribution (under H0) whose parameters depend on the threshold `xi` and on the autocorrelation between pvalues within each chromosome. The threshold has to be selected in 1,2,3,4 and the autocorrelation is computed internally.

Value

A list with the following elements:

Data	The dataset Data with the local score as supplementary column.
SigZones	A dataset containing information about the significant regions.
SigMarker	A dataset containing the significant markers.
ChrThreshold	A dataset containing the chromosome-wide significance thresholds.

Examples

```
require(dplyr)
# Import the data
data("metaData")

# Compute the inter-environment correlation matrix
matCorr <- metaGE.cor(metaData, Threshold = 0.8)

# Fit the Fixed Effect model
FeDF <- metaGE.fit(metaData, matCorr, Method = "Fe")

# Compute the score local
xi <- 2
FeScore <- metaGE.lscore(FeDF,"PVALUE", xi)
#FeScore$SigZones
```

---

metaGE.manhattan *Draw the Manhattan plot.*

Description

The function metaGE.manhattan displays the Manhattan plot of the -log10(p-value) or the local score of each marker along the genome.

Usage

```
metaGE.manhattan(
  Data,
  VarName,
  Threshold = NULL,
  SigZones = NULL,
  Score = FALSE,
  AnnotateMarkers = NULL,
  Main = "",
  col = c("grey", "black"),
  colSigZones = "blue",
  Ylim = NULL
)
```

Arguments

Data	A dataset containing the columns: CHR, POS, MARKER and the variable to plot for each marker, as obtained from metaGE.fit().
VarName	The name of the column containing the variable to plot, generally the p-value or a score.
Threshold	A threshold in order to draw a "genome-wide significant" line. (optional)
SigZones	A dataset containing the significant zones to plot, as obtained from metaGE.lscore(). Must have columns: CHR, Start, End. (optional)
Score	A boolean. If FALSE, the -log10 of the variable is plotted, useful for plotting p-values. If TRUE, the raw values of the variable is plotted, useful for plotting scores. (FALSE by default)
AnnotateMarkers	A list of markers name to annotate in the plot. (optional)
Main	The main to display. (optional)
col	A character vector indicating which colors to alternate for different chromosomes. (optional)
colSigZones	A character indicating which color to plot the significant zones.("blue" by default)
Ylim	Two numeric values, specifying the lower limit and the upper limit of the y-axe scale. (optional)

Value

The Manhattan plot.

Examples

```
require(dplyr)
# Import the data
data("metaData")

# Compute the inter-environment correlation matrix
matCorr <- metaGE.cor(metaData, Threshold = 0.8)

# Fit the Fixed Effect model
FeDF <- metaGE.fit(metaData, matCorr, Method = "Fe")

# Control the FDR (here Benjamini-Hochberg)
Alpha <- 0.05
Signif <-  FeDF$PVALUE %>% p.adjust(method = "BH") %>% `<`(Alpha) %>% which

# Draw the corresponding manhattan plot
PvalThresholdFe <- FeDF[Signif,]$PVALUE%>% max %>% max(.,0)
manhattan_pval <- metaGE.manhattan(Data = FeDF,VarName = 'PVALUE',
                              Threshold = PvalThresholdFe,
                               Main = '-log10(Pval) alongside the chromosome Fe method')
```

```
# Compute the score local
xi <- 2
FeScore <- metaGE.lscore(FeDF,"PVALUE", xi)

# Draw the corresponding manhattan plot
manhattan_lscore <- metaGE.manhattan(Data = FeScore$Data,VarName = 'SCORE',
                                     SigZones = FeScore$SigZones, Score = TRUE,
                                  Main = 'Local score alongside the chromosome Fe method')
```

`metaGE.pvalplot` *Display visual checks of pvalues.*

Description

The function metaGE.pvalplot displays the pvalue distribution and the QQplot of the -log10(pvalues).

Usage

```
metaGE.pvalplot(Pvalues, Main = "")
```

Arguments

`Pvalues`	A vector containing pvalues.
`Main`	The main to display.(optional)

Value

No return value, the plot is displayed in the active graphics window.

Examples

```
# Import the data
data("metaData")

# Compute the inter-environment correlation matrix
matCorr <- metaGE.cor(metaData, Threshold = 0.8)

# Fit the Fixed Effect model
FeDF <- metaGE.fit(metaData, matCorr, Method = "Fe")

# Check the pvalues
metaGE.pvalplot(Pvalues = FeDF$PVALUE, Main= "Pvalue Fe")
```

`metaGE.regplot` *Plot the z-score of a marker according to a covariate.*

Description

The function metaGE.regplot displays the graph of the z-scores of a marker according to a covariate.

Usage

```
metaGE.regplot(
  Data,
  Covariate,
  EnvName,
  MarkerName,
  VarName,
  Zscore = FALSE,
  aesCol = NULL,
  Main = ""
)
```

Arguments

`Data`	A dataset containing the columns: MARKER and the z-scores or the effects of each marker (in rows) in each environment (in columns), as obtained from `metaGE.collect()`.
`Covariate`	A dataset containing the values of one or more covariates (in columns) in each environment (in rows).
`EnvName`	The name of the column containing the names of the environment in the `Covariate` dataset.
`MarkerName`	The name of the marker.
`VarName`	The name of the column containing the covariable to plot.
`Zscore`	A boolean. If `FALSE`, the estimated marker effects is plotted. If `TRUE`, the z-scores of the marker is plotted. (`FALSE` by default)
`aesCol`	The name of the column in the `Covariate` dataset containing a qualitative covariable to specify the color of the points. (optional)
`Main`	The main to display.(optional)

Value

The plot

Examples

```
data("metaData")
data("envDesc")
metaGE.regplot(Data = metaData, Covariate = envDesc, EnvName = "ShortName",
           MarkerName = "AX-91369217", VarName = "Tnight.mean", aesCol = "Classification")
```

`metaGE.test` *Meta-analysis test for Genotype x Environment interactions: Contrast or Regression.*

Description

The function metaGE.test compute meta-analysis contrast or regression test.

Usage

```
metaGE.test(
  Data,
  MatCorr,
  Incidence = NULL,
  Contrast = NULL,
  Covariate = NULL,
  EnvName = NULL,
  NA.omit = TRUE,
  DropZScores = FALSE
)
```

Arguments

`Data`	A dataset containing the estimated marker effect and its associated pvalue of each marker (in rows) in each environment (in columns), as obtained from `metaGE.collect()`.
`MatCorr`	The inter-environment correlation matrix. It can be compute by the `metaGE.cor()` function.
`Incidence`	A matrix of incidence, as obtained from `metaGE.incidence()` or a list of such matrix.
`Contrast`	A matrix of contrast, or a list of such matrix.
`Covariate`	A dataset containing the values of one or more covariates (in columns) in each environment (in rows).
`EnvName`	The name of the column containing the names of the environment in the `Covariate` dataset.
`NA.omit`	A boolean specifying whether the markers with some `NA` values should be removed from the test procedure. (`TRUE` by default)
`DropZScores`	A boolean specifying whether the Zscores should be dropped from the dataset or not. (`FALSE` by default)

Details

If Incidence is provided, the function will perform all the corresponding tests of contrast. If Covariate is provided, the function will perform all the corresponding meta-regression tests. The Contrast can be NULL, in this case the identity matrix is used.

Value

The dataset Data with supplementary columns containing the PVALUE of each test performed.

Examples

```
require(dplyr)

 # Import the data
data("metaData")
data("envDesc")

# Compute the inter-environment correlation matrix
matCorr <- metaGE.cor(metaData, Threshold = 0.8)

#### Contrast test
# Build the matrix of incidence
Incidence.Water <- metaGE.incidence(VarName = "Water",Covariate = envDesc,
                                    EnvName = "ShortName", Data = metaData)

# Perform the contrast test
ContrastDF <- metaGE.test(metaData, matCorr,Incidence = Incidence.Water, Contrast = NULL)
head(ContrastDF %>% select(CHR, POS, MARKER, PVALUE.Contrast1))

#### Regression test
RegressionDF <- metaGE.test(metaData,matCorr, Covariate = envDesc[,c(1,5)],EnvName = "ShortName" )
head(RegressionDF %>% select(CHR, POS, MARKER, PVALUE.Tnight.mean))
```

ReadData *ReadData*

Description

This function read the one file, select interesting columns and rename them.

Usage

```
ReadData(ListN, FileN, VarN, MinFreq = 0)
```

Arguments

ListN	The name of the list of files where the file to read belongs or NULL if there is only one list of files
FileN	The name of the file to read

VarN	A named list containing the column names in the file corresponding to the variables below: MARKER, CHR, POS, EFFECT, PVAL. (optional: FREQ, ALLELE0, ALLELE1)
MinFreq	A numeric value allowing to filter to keep markers with MAF > MinFreq

Value

A tibble with the interesting columns selected and renamed.

RegressionStatTest *Compute the pvalue of the meta-regression test.*

Description

The function RegressionStatTest compute the statistic and the pvalue of the regression test.

Usage

```
RegressionStatTest(Covariate, CovName, Zmat, MatCorr)
```

Arguments

Covariate	A dataset containing the values of one Covariate (in columns) in each environment (in rows).
CovName	The name the Covariate.
Zmat	A matrix containing the Zscores of all markers (in rows) in each environment (in columns).
MatCorr	The inter-environments correlation matrix. Can be computed using metaGE.cor().

Value

A dataset of two columns containing the pvalue of the meta-regression test and the number of environment used to perform the test of all markers.

RegressionStatTestNA *Compute the pvalue of the regression test in presence of missing values.*

Description

The function RegressionStatTest compute the statistic and the pvalue of the regression test.

Usage

```
RegressionStatTestNA(Covariate, CovName, Zmat, MatCorr, Data, Configs.list)
```

Arguments

Covariate	A dataset containing the values of one covariate (in columns) in each environment (in rows).
CovName	The name the covariate.
Zmat	A matrix containing the Zscores of all markers (in rows) in each environment (in columns).
MatCorr	The inter-environments correlation matrix. Can be computed using metaGE.cor().
Data	A dataset containing the effect, the pvalues and the NA configuration for all marker
Configs.list	A vector containing the NA configurations present in the dataset

Value

A dataset of two columns containing the pvalue of the meta-regression test and the number of environment used to perform the test of all markers.

sig_sl *Computation of the significative regions*

Description

The function sig_sl computes the significative regions from a lindley process given a significance threshold.(function from localscore)

Usage

```
sig_sl(lind, pos, th)
```

Arguments

lind	The lindley
pos	The position
th	The threshold

Value

the significance threshold.

---

thresUnif *Computation of the significance threshold*

Description

The function thresUnif computes the significance threshold.(function from localscore)

Usage

```
thresUnif(L, cor, xi, alpha = 0.05)
```

Arguments

L	The length of the chromosome
cor	The autocorrelation of the chromosome
xi	The threshold of the score, xi = 1,2,3 or 4.
alpha	The nominal threshold.

Details

The distribution of the p-values is uniform, the local score follows a Gumbel distribution under the null.

Value

the significance threshold.

Index