

# Package: qch (via r-universe)

September 12, 2024

**Title** Query Composite Hypotheses

**Version** 2.0.0

**Maintainer** Tristan Mary-Huard <tristan.mary-huard@agroparistech.fr>

**Description** Provides functions for the joint analysis of  $Q$  sets of p-values obtained for the same list of items. This joint analysis is performed by querying a composite hypothesis, i.e. an arbitrary complex combination of simple hypotheses, as described in Mary-Huard et al. (2021) <[doi:10.1093/bioinformatics/btab592](https://doi.org/10.1093/bioinformatics/btab592)> and De Walsche et al.(2023) <[doi:10.1101/2024.03.17.585412](https://doi.org/10.1101/2024.03.17.585412)>. In this approach, the  $Q$ -uplet of p-values associated with each item is distributed as a multivariate mixture, where each of the  $2^Q$  components corresponds to a specific combination of simple hypotheses. The dependence between the p-value series is considered using a Gaussian copula function. A p-value for the composite hypothesis test is derived from the posterior probabilities.

**License** GPL-3

**Depends** R (>= 2.10)

**Imports** copula, dplyr, graphics, ks, purrr, qvalue, Rcpp, stats, stringr, utils

**LinkingTo** Rcpp, RcppArmadillo

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**NeedsCompilation** yes

**Author** Tristan Mary-Huard [aut, cre] (<<https://orcid.org/0000-0002-3839-9067>>), Annaig De Walsche [aut] (<<https://orcid.org/0000-0003-0603-1716>>), Franck Gauthier [ctb] (<<https://orcid.org/0000-0003-0574-065X>>)

**Date/Publication** 2024-03-29 22:50:02 UTC

**Repository** <https://tmaryhuard.r-universe.dev>

**RemoteUrl** <https://github.com/cran/qch>

**RemoteRef** HEAD

**RemoteSha** cbf2b368ab31a41e547094a19018cbaf7516da19

## Contents

Copula.Hconfig_gaussian_density . . . . .	2
EM_calibration_gaussian . . . . .	3
EM_calibration_gaussian_memory . . . . .	4
EM_calibration_indep . . . . .	5
EM_calibration_indep_memory . . . . .	5
f1_separation_signed . . . . .	6
FastKerFdr_signed . . . . .	7
FastKerFdr_unsigned . . . . .	7
fHconfig_sum_update_gaussian_copula_ptr_parallel . . . . .	8
fHconfig_sum_update_ptr_parallel . . . . .	9
gaussian_copula_density . . . . .	10
GetH1AtLeast . . . . .	10
GetH1Equal . . . . .	11
GetHconfig . . . . .	12
prior_update_arma_ptr_parallel . . . . .	12
prior_update_gaussian_copula_ptr_parallel . . . . .	13
PvalSets . . . . .	14
PvalSets_cor . . . . .	14
qch.fit . . . . .	15
qch.test . . . . .	16
R.MLE . . . . .	18
R.MLE.check . . . . .	18
R.MLE.memory . . . . .	19
R_MLE_update_gaussian_copula_ptr_parallel . . . . .	20
<b>Index</b>	<b>21</b>

---

Copula.Hconfig\_gaussian\_density

*Gaussian copula density for each Hconfiguration.*

---

### Description

Gaussian copula density for each Hconfiguration.

### Usage

Copula.Hconfig\_gaussian\_density(Hconfig, F0Mat, F1Mat, R)

**Arguments**

Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <a href="#">GetHconfig()</a> function.
F0Mat	a matrix containing the evaluation of the marginal cdf under H0 at each items, each column corresponding to a p-value serie.
F1Mat	a matrix containing the evaluation of the marginal cdf under H1 at each items, each column corresponding to a p-value serie.
R	the correlation matrix.

**Value**

A matrix containing the evaluation of the Gaussian density function for each Hconfiguration in columns.

---

EM\_calibration\_gaussian

*EM calibration in the case of the gaussian copula (unsigned)*

---

**Description**

EM calibration in the case of the gaussian copula (unsigned)

**Usage**

```
EM_calibration_gaussian(
  Hconfig,
  F0Mat,
  F1Mat,
  fHconfig,
  R.init,
  Prior.init,
  Precision = 1e-06
)
```

**Arguments**

Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <a href="#">GetHconfig()</a> function.
F0Mat	a matrix containing the evaluation of the marginal cdf under H0 at each items, each column corresponding to a p-value serie.
F1Mat	a matrix containing the evaluation of the marginal cdf under H1 at each items, each column corresponding to a p-value serie.
fHconfig	a matrix containing config densities evaluated at each items, each column corresponding to a configurations.
R.init	the initialization of the correlation matrix of the gaussian copula parameter.
Prior.init	the initialization of prior probabilities for each of the H-configurations.
Precision	Precision for the stop criterion. (Default is 1e-6)

**Value**

A list of 2 objects 'priorHconfig' and 'Rcopula'. Object 'priorHconfig' is a vector of estimated prior probabilities for each of the H-configurations. Object 'Rcopula' is the estimated correlation matrix of the gaussian copula.

---

EM\_calibration\_gaussian\_memory

*EM calibration in the case of the gaussian copula (unsigned) with memory management*

---

**Description**

EM calibration in the case of the gaussian copula (unsigned) with memory management

**Usage**

```
EM_calibration_gaussian_memory(
  Logf0Mat,
  Logf1Mat,
  F0Mat,
  F1Mat,
  Prior.init,
  R.init,
  Hconfig,
  Precision = 1e-06,
  threads_nb
)
```

**Arguments**

Logf0Mat	a matrix containing the $\log(f_0(x_{i,q}))$
Logf1Mat	a matrix containing the $\log(f_1(x_{i,q}))$
F0Mat	a matrix containing the evaluation of the marginal cdf under H0 at each items, each column corresponding to a p-value serie.
F1Mat	a matrix containing the evaluation of the marginal cdf under H1 at each items, each column corresponding to a p-value serie.
Prior.init	the initialization of prior probabilities for each of the H-configurations.
R.init	the initialization of the correlation matrix of the gaussian copula parameter.
Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <a href="#">GetHconfig()</a> function.
Precision	Precision for the stop criterion. (Default is 1e-6)
threads_nb	The number of threads to use.

**Value**

A list of 2 objects 'priorHconfig' and 'Rcopula'. Object 'priorHconfig' is a vector of estimated prior probabilities for each of the H-configurations. Object 'Rcopula' is the estimated correlation matrix of the gaussian copula.

---

EM\_calibration\_indep    *EM calibration in the case of conditional independence*

---

**Description**

EM calibration in the case of conditional independence

**Usage**

```
EM_calibration_indep(fHconfig, Prior.init, Precision = 1e-06)
```

**Arguments**

fHconfig	a matrix containing config densities evaluated at each items, each column corresponding to a configurations.
Prior.init	the initialization of prior probabilities for each of the H-configurations.
Precision	Precision for the stop criterion. (Default is 1e-6)

**Value**

a vector of estimated prior probabilities for each of the H-configurations.

---

EM\_calibration\_indep\_memory  
*EM calibration in the case of conditional independence with memory management (unsigned)*

---

**Description**

EM calibration in the case of conditional independence with memory management (unsigned)

**Usage**

```
EM_calibration_indep_memory(  
  Logf0Mat,  
  Logf1Mat,  
  Prior.init,  
  Hconfig,  
  Precision = 1e-06,  
  threads_nb  
)
```

**Arguments**

Logf0Mat	a matrix containing the $\log(f_0(x_{i_q}))$
Logf1Mat	a matrix containing the $\log(f_1(x_{i_q}))$
Prior.init	the initialization of prior probabilities for each of the H-configurations.
Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <code>GetHconfig()</code> function.
Precision	Precision for the stop criterion. (Default is 1e-6)
threads_nb	The number of threads to use.

**Value**

a vector of estimated prior probabilities for each of the H-configurations.

---

f1\_separation\_signed    *Signed case function: Separate f1 into f+ and f-*

---

**Description**

Signed case function: Separate f1 into f+ and f-

**Usage**

```
f1_separation_signed(XMat, f0Mat, f1Mat, p0, plotting = FALSE)
```

**Arguments**

XMat	a matrix of probit-transformed p-values, each column corresponding to a p-value serie.
f0Mat	a matrix containing the evaluation of the marginal density functions under H0 at each items, each column corresponding to a p-value serie.
f1Mat	a matrix containing the evaluation of the marginal density functions under H1 at each items, each column corresponding to a p-value serie.
p0	the proportions of H0 items for each series.
plotting	boolean, should some diagnostic graphs be plotted. Default is FALSE.

**Value**

A list of 4 objects 'f1plusMat', 'f1minusMat', 'p1plus', 'p1minus'. Object 'f1plusMat' is a matrix containing the evaluation of the marginal density functions under H1plus at each items, each column corresponding to a p-value serie. Object 'f1minusMat' is a matrix containing the evaluation of the marginal density functions under H1minus at each items, each column corresponding to a p-value serie. Object 'p1plus' is an estimate of the proportions of H1plus items for each series. Object 'p1minus' is an estimate of the proportions of H1minus items for each series.

---

FastKerFdr\_signed      *FastKerFdr signed*

---

### Description

FastKerFdr signed

### Usage

```
FastKerFdr_signed(X, p0 = NULL, plotting = FALSE, NbKnot = 1e+05, tol = 1e-05)
```

### Arguments

X	a vector of probit-transformed p-values (corresponding to a p-value serie)
p0	a priori proportion of H0 hypotheses
plotting	boolean, should some diagnostic graphs be plotted. Default is FALSE.
NbKnot	The (maximum) number of knot for the kde procedure. Default is 1e5
tol	a tolerance value for convergence. Default is 1e-5

### Value

A list of 3 objects. Object 'p0' is an estimate of the proportion of H0 hypotheses, Object 'tau' is the vector of H1 posteriors, Object 'f1' is a numeric vector, each coordinate i corresponding to the evaluation of the H1 density at point xi, where xi is the ith item in X. Object 'F1' is a numeric vector, each coordinate i corresponding to the evaluation of the H1 ;cdf at point xi, where xi is the ith item in X.

---

FastKerFdr\_unsigned      *FastKerFdr unsigned*

---

### Description

FastKerFdr unsigned

### Usage

```
FastKerFdr_unsigned(  
  X,  
  p0 = NULL,  
  plotting = FALSE,  
  NbKnot = 1e+05,  
  tol = 1e-05  
)
```

**Arguments**

X	a vector of probit-transformed p-values (corresponding to a p-value serie)
p0	a priori proportion of H0 hypotheses
plotting	boolean, should some diagnostic graphs be plotted. Default is FALSE.
NbKnot	The (maximum) number of knot for the kde procedure. Default is 1e5
tol	a tolerance value for convergence. Default is 1e-5

**Value**

A list of 3 objects. Object 'p0' is an estimate of the proportion of H0 hypotheses, Object 'tau' is the vector of H1 posteriors, Object 'f1' is a numeric vector, each coordinate  $i$  corresponding to the evaluation of the H1 density at point  $x_i$ , where  $x_i$  is the  $i$ th item in X. Object 'F1' is a numeric vector, each coordinate  $i$  corresponding to the evaluation of the H1 cdf at point  $x_i$ , where  $x_i$  is the  $i$ th item in X.

---

fHconfig\_sum\_update\_gaussian\_copula\_ptr\_parallel

*Computation of the sum  $\sum_c(w_c \cdot \psi_c)$  using Gaussian copula parallelized version*

---

**Description**

Computation of the sum  $\sum_c(w_c \cdot \psi_c)$  using Gaussian copula parallelized version

**Usage**

```
fHconfig_sum_update_gaussian_copula_ptr_parallel(
  Hconfig,
  NewPrior,
  Logf0Mat,
  Logf1Mat,
  zeta0,
  zeta1,
  R,
  Rinv,
  threads_nb = 0L
)
```

**Arguments**

Hconfig	list of vector of 0 and 1, corresponding to the configurations
NewPrior	a double vector containing the prior $w_c$
Logf0Mat	a double matrix containing the $\log(f_0(x_{i_q}))$
Logf1Mat	a double matrix containing the $\log(f_1(x_{i_q}))$



zeta0	a double matrix containing the qnorm(F0(x_iq))
zeta1	a double matrix containing the qnorm(F1(x_iq))
R	a double matrix corresponding to the copula parameter
Rinv	a double matrix corresponding to the inverse copula parameter
threads_nb	an int the number of threads

**Value**

a double vector containing  $\text{sum\_c}(w\_c*\psi\_c)$

---

fHconfig\_sum\_update\_ptr\_parallel  
*Computation of the sum  $\text{sum\_c}(w\_c*\psi\_c)$  parallelized version*

---

**Description**

Computation of the sum  $\text{sum\_c}(w\_c*\psi\_c)$  parallelized version

**Usage**

```
fHconfig_sum_update_ptr_parallel(
  Hconfig,
  NewPrior,
  Logf0Mat,
  Logf1Mat,
  threads_nb = 0L
)
```

**Arguments**

Hconfig	list of vector of 0 and 1, corresponding to the configurations
NewPrior	a double vector containing the prior $w\_c$
Logf0Mat	a double matrix containing the $\log(f0(x_iq))$
Logf1Mat	a double matrix containing the $\log(f1(x_iq))$
threads_nb	an int the number of threads

**Value**

a double vector containing  $\text{sum\_c}(w\_c*\psi\_c)$

---

gaussian\_copula\_density  
*Gaussian copula density*

---

**Description**

Gaussian copula density

**Usage**

```
gaussian_copula_density(zeta, R, Rinv)
```

**Arguments**

zeta	the matrix of probit-transformed observations.
R	the correlation matrix.
Rinv	the inverse correlation matrix.

**Value**

A numeric vector, each coordinate  $i$  corresponding to the evaluation of the Gaussian copula density function at observation  $zeta_i$ .

---

GetH1AtLeast	<i>Specify the configurations corresponding to the composite <math>H_1</math> test "AtLeast".</i>
--------------	---

---

**Description**

Specify which configurations among Hconfig correspond to the composite alternative hypothesis : {at least "AtLeast"  $H_1$  hypotheses are of interest }

**Usage**

```
GetH1AtLeast(Hconfig, AtLeast, Consecutive = FALSE, SameSign = FALSE)
```

**Arguments**

Hconfig	A list of all possible combination of $H_0$ and $H_1$ hypotheses generated by the <a href="#">GetHconfig()</a> function.
AtLeast	How many $H_1$ hypotheses at least for the item to be of interest ? (an integer or a vector).
Consecutive	Should the significant test series be consecutive ? (optional, default is FALSE).
SameSign	Should the significant test series have the same sign ? (optional, default is FALSE).

**Value**

A vector 'Hconfig.H1' of components of Hconfig that correspond to the 'AtLeast' specification.

**See Also**

[GetH1Equal\(\)](#)

**Examples**

```
GetH1AtLeast(GetHconfig(4),2)
```

---

GetH1Equal	<i>Specify the configurations corresponding to the composite <math>H_1</math> test "Equal".</i>
------------	---

---

**Description**

Specify which configurations among Hconfig correspond to the composite alternative hypothesis :{Exactly "Equal"  $H_1$  hypotheses are of interest }

**Usage**

```
GetH1Equal(Hconfig, Equal, Consecutive = FALSE, SameSign = FALSE)
```

**Arguments**

Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <a href="#">GetHconfig()</a> function.
Equal	What is the exact number of $H_1$ hypotheses for the item to be of interest? (an integer or a vector).
Consecutive	Should the significant test series be consecutive ? (optional, default is FALSE).
SameSign	Should the significant test series have the same sign ? (optional, default is FALSE).

**Value**

A vector 'Hconfig.H1' of components of Hconfig that correspond to the 'Equal' specification.

**See Also**

[GetH1AtLeast\(\)](#)

**Examples**

```
GetH1Equal(GetHconfig(4),2)
```

---

GetHconfig                      *Generate the  $H_0/H_1$  configurations.*

---

**Description**

Generate all possible combination of simple hypotheses  $H_0/H_1$ .

**Usage**

```
GetHconfig(Q, Signed = FALSE)
```

**Arguments**

Q                                  The number of test series to be combined.  
Signed                              Should the sign of the effect be taken into account? (optional, default is FALSE).

**Value**

A list 'Hconfig' of all possible combination of  $H_0$  and  $H_1$  hypotheses among  $Q$  hypotheses tested.

**Examples**

```
GetHconfig(4)
```

---

prior\_update\_arma\_ptr\_parallel  
*Update of the prior estimate in EM algo parallelized version*

---

**Description**

Update of the prior estimate in EM algo parallelized version

**Usage**

```
prior_update_arma_ptr_parallel(  

  Hconfig,  

  fHconfig_sum,  

  OldPrior,  

  Logf0Mat,  

  Logf1Mat,  

  threads_nb = 0L  

)
```

**Arguments**

Hconfig	list of vector of 0 and 1, corresponding to the configurations
fHconfig_sum	a double vector containing $\text{sum}_c(w_c * \psi_c)$ , obtained by <code>fHconfig_sum_update_ptr_parallel()</code>
OldPrior	a double vector containing the prior $w_c$
Logf0Mat	a double matrix containing the $\log(f_0(x_{i_q}))$
Logf1Mat	a double matrix containing the $\log(f_1(x_{i_q}))$
threads_nb	an int the number of threads

**Value**

a double vector containing the new estimate of prior  $w_c$

---

prior\_update\_gaussian\_copula\_ptr\_parallel

*Update of the prior estimate in EM algo using Gaussian copula, parallelized version*

---

**Description**

Update of the prior estimate in EM algo using Gaussian copula, parallelized version

**Usage**

```
prior_update_gaussian_copula_ptr_parallel(
  Hconfig,
  fHconfig_sum,
  OldPrior,
  Logf0Mat,
  Logf1Mat,
  zeta0,
  zeta1,
  R,
  Rinv,
  threads_nb = 0L
)
```

**Arguments**

Hconfig	list of vector of 0 and 1, corresponding to the configurations
fHconfig_sum	a double vector containing $\text{sum}_c(w_c * \psi_c)$ , obtained by <code>fHconfig_sum_update_ptr_parallel()</code>
OldPrior	a double vector containing the prior $w_c$
Logf0Mat	a double matrix containing the $\log(f_0(x_{i_q}))$
Logf1Mat	a double matrix containing the $\log(f_1(x_{i_q}))$
zeta0	a double matrix containing the $qnorm(F_0(x_{i_q}))$

zeta1	a double matrix containing the $qnorm(F1(x_{iq}))$
R	a double matrix corresponding to the copula parameter
Rinv	a double matrix corresponding to the inverse copula parameter
threads_nb	an int the number of threads

**Value**

a double vector containing the new estimate of prior  $w_c$

---

PvalSets	<i>Synthetic example to illustrate the main qch functions</i>
----------	---

---

**Description**

PvalSets is a data.frame with 10,000 rows and 3 columns. Each row corresponds to an item, columns 'Pval1' and 'Pval2' each correspond to a test serie over the items, and column 'Class' provides the truth, i.e. if item  $i$  belongs to class 1 then the H0 hypothesis is true for the 2 tests, if item  $i$  belongs to class 2 (resp. 3) then the H0 hypothesis is true for the first (resp. second) test only, and if item  $i$  belongs to class 4 then both H0 hypotheses are false (for the first and the second test).

**Usage**

PvalSets

**Format**

A data.frame

---

PvalSets_cor	<i>Synthetic example to illustrate the main qch functions using gaussian copula</i>
--------------	---

---

**Description**

PvalSets\_cor is a data.frame with 10,000 rows and 3 columns. Each row corresponds to an item, columns 'Pval1' and 'Pval2' each correspond to a test serie over the items, and column 'Class' provides the truth, i.e. if item  $i$  belongs to class 1 then the H0 hypothesis is true for the 2 tests, if item  $i$  belongs to class 2 (resp. 3) then the H0 hypothesis is true for the first (resp. second) test only, and if item  $i$  belongs to class 4 then both H0 hypotheses are false (for the first and the second test). The correlation between the two pvalues series within each class is 0.3.

**Usage**

PvalSets\_cor

**Format**

A data.frame

---

qch.fit	<i>Infer posterior probabilities of <math>H_0/H_1</math> configurations.</i>
---------	--

---

### Description

For each item, estimate the posterior probability for each configuration. This function use either the model accounting for the dependence structure through a Gaussian copula function (`copula=="gaussian"`) or assuming the conditional independence (`copula=="indep"`). Utilizes parallel computing, when available. For package documentation, see [qch-package](#).

### Usage

```
qch.fit(
  pValMat,
  EffectMat = NULL,
  Hconfig,
  copula = "indep",
  threads_nb = 0,
  plotting = FALSE,
  Precision = 1e-06
)
```

### Arguments

pValMat	A matrix of p-values, each column corresponding to a p-value serie.
EffectMat	A matrix of estimated effects corresponding to the p-values contained in pValMat. If specified, the procedure will account for the direction of the effect. (optional, default is NULL)
Hconfig	A list of all possible combination of $H_0$ and $H_1$ hypotheses generated by the <a href="#">GetHconfig()</a> function.
copula	A string specifying the form of copula to use. Possible values are "indep" and "gaussian". Default is "indep" corresponding to the independent case.
threads_nb	The number of threads to use. The number of thread will set to the number of core available by default.
plotting	A boolean. Should some diagnostic graphs be plotted ? Default is FALSE.
Precision	The precision for EM algorithm to infer the parameters. Default is 1e-6.

### Value

A list with the following elements:

prior	vector of estimated prior probabilities for each of the H-configurations.
Rcopula	the estimated correlation matrix of the Gaussian copula. (if applicable)
Hconfig	the list of all configurations.

- If the storage permits, the list will additionally contain:

posterior    matrix providing for each item (in row) its posterior probability to belong to each of the H-configurations (in column)  
 fhconfig    matrix containing  $\psi_c$  densities evaluated at each items, each column corresponding to a configuration.

- Else, the list will additionally contain:

f0Mat        matrix containing the evaluation of the marginal densities under  $H_0$  at each items, each column corresponding to a configuration  
 f1Mat        matrix containing the evaluation of the marginal densities under  $H_1$  at each items, each column corresponding to a configuration  
 F0Mat        matrix containing the evaluation of the marginal cdf under  $H_0$  at each items, each column corresponding to a configuration  
 F1Mat        matrix containing the evaluation of the marginal cdf under  $H_1$  at each items, each column corresponding to a configuration  
 fhconfig\_sum    vector containing  $(\sum_c w_c \psi_c(Z_i))$  for each items  $i$ .

The elements of interest are the posterior probabilities matrix, posterior, the estimated proportion of observations belonging to each configuration, prior, and the estimated correlation matrix of the Gaussian copula, Rcopula. The remaining elements are returned primarily for use by other functions.

## Examples

```
data(PvalSets_cor)
PvalMat <- as.matrix(PvalSets_cor[,-3])
## Build the Hconfig objects
Q <- 2
Hconfig <- GetHconfig(Q)

## Run the function
res.fit <- qch.fit(pValMat = PvalMat, Hconfig = Hconfig, copula="gaussian")

## Display the prior of each class of items
res.fit$prior

## Display the correlation estimate of the gaussian copula
res.fit$Rcopula

## Display the first posteriors
head(res.fit$posterior)
```

---

qch.test

*Perform composite hypothesis testing.*

---

## Description

Perform any composite hypothesis test by specifying the configurations 'Hconfig.H1' corresponding to the composite alternative hypothesis among all configurations 'Hconfig'.

By default, the function performs the composite hypothesis test of being associated with "at least  $q$ " simple tests, for  $q = 1, \dots, Q$ .



**Usage**

```
qch.test(res.qch.fit, Hconfig, Hconfig.H1 = NULL, Alpha = 0.05, threads_nb = 0)
```

**Arguments**

res.qch.fit	The result provided by the <a href="#">qch.fit()</a> function.
Hconfig	A list of all possible combination of $H_0$ and $H_1$ hypotheses generated by the <a href="#">GetHconfig()</a> function.
Hconfig.H1	An integer vector (or a list of such vector) of the Hconfig index corresponding to the composite alternative hypothesis configuration(s). Can be generated by the <a href="#">GetH1AtLeast()</a> or <a href="#">GetH1Equal()</a> functions. If NULL, the composite hypothesis tests of being associated with "at least $q$ " simple tests, for $q=1,..Q$ are performed.
Alpha	the nominal Type I error rate for FDR control. Default is 0.05.
threads_nb	The number of threads to use. The number of thread will set to the number of core available by default.

**Value**

A list with the following elements:

Rejection	a matrix providing for each item the result of the composite hypothesis test, after adaptive Benjamin-Höchberg m
lFDR	a matrix providing for each item its local FDR estimate.
Pvalues	a matrix providing for each item its p-value of the composite hypothesis test.

**See Also**

[qch.fit\(\)](#), [GetH1AtLeast\(\)](#), [GetH1Equal\(\)](#)

**Examples**

```
data(PvalSets_cor)
PvalMat <- as.matrix(PvalSets_cor[,-3])
Truth <- PvalSets[,3]

## Build the Hconfig objects
Q <- 2
Hconfig <- GetHconfig(Q)

## Infer the posteriors
res.fit <- qch.fit(pValMat = PvalMat, Hconfig = Hconfig, copula="gaussian")

## Run the test procedure with FDR control
H1config <- GetH1AtLeast(Hconfig,2)
res.test <- qch.test(res.qch.fit = res.fit,Hconfig = Hconfig, Hconfig.H1 = H1config)
table(res.test$Rejection$AtLeast_2,Truth==4)
```

---

R.MLE	<i>Gaussian copula correlation matrix Maximum Likelihood estimator.</i>
-------	---

---

**Description**

Gaussian copula correlation matrix Maximum Likelihood estimator.

**Usage**

```
R.MLE(Hconfig, zeta0, zeta1, Tau)
```

**Arguments**

Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <a href="#">GetHconfig()</a> function.
zeta0	a matrix containing the $\Phi(F_0(Z_{iq}))$ , each column corresponding to a p-value serie.
zeta1	a matrix containing the $\Phi(F_1(Z_{iq}))$ , each column corresponding to a p-value serie.
Tau	a matrix providing for each item (in row) its posterior probability to belong to each of the H-configurations (in columns).

**Value**

Estimate of the correlation matrix.

---

R.MLE.check	<i>Check the Gaussian copula correlation matrix Maximum Likelihood estimator</i>
-------------	--

---

**Description**

Check the Gaussian copula correlation matrix Maximum Likelihood estimator

**Usage**

```
R.MLE.check(R)
```

**Arguments**

R	Estimate of the correlation matrix.
---	-------------------------------------

**Value**

Estimate of the correlation matrix.

---

R.MLE.memory	<i>Gaussian copula correlation matrix Maximum Likelihood estimator (memory handling)</i>
--------------	--

---

### Description

Gaussian copula correlation matrix Maximum Likelihood estimator (memory handling)

### Usage

```
R.MLE.memory(
  Hconfig,
  fHconfig_sum,
  OldPrior,
  Logf0Mat,
  Logf1Mat,
  zeta0,
  zeta1,
  OldR,
  OldRinv
)
```

### Arguments

Hconfig	A list of all possible combination of H0 and H1 hypotheses generated by the <a href="#">GetHconfig()</a> function.
fHconfig_sum	a vector containing $\text{sum}_c(w_c * \psi_c)$ for each items.
OldPrior	a vector containing the prior probabilities for each of the H-configurations.
Logf0Mat	a matrix containing $\log(f0Mat)$ , each column corresponding to a p-value serie.
Logf1Mat	a matrix containing $\log(f1Mat)$ , each column corresponding to a p-value serie.
zeta0	a matrix containing $\text{qnorm}(F0Mat)$ , each column corresponding to a p-value serie.
zeta1	a matrix containing $\text{qnorm}(F1Mat)$ , each column corresponding to a p-value serie.
OldR	the copula correlation matrix.
OldRinv	the inverse of copula correlation matrix.

### Value

Estimate of the correlation matrix.

---

R\_MLE\_update\_gaussian\_copula\_ptr\_parallel

*Update the estimate of R correlation matrix of the gaussian copula, parallelized version*

---

### Description

Update the estimate of R correlation matrix of the gaussian copula, parallelized version

### Usage

```
R_MLE_update_gaussian_copula_ptr_parallel(  
  Hconfig,  
  fHconfig_sum,  
  OldPrior,  
  Logf0Mat,  
  Logf1Mat,  
  zeta0,  
  zeta1,  
  OldR,  
  OldRinv,  
  RhoIndex,  
  threads_nb = 0L  
)
```

### Arguments

Hconfig	list of vector of 0 and 1, corresponding to the configurations
fHconfig_sum	a double vector containing $\text{sum}_c(w_c * \psi_c)$ , obtained by <code>fHconfig_sum_update_ptr_parallel()</code>
OldPrior	a double vector containing the prior $w_c$
Logf0Mat	a double matrix containing the $\log(f_0(x_{i_q}))$
Logf1Mat	a double matrix containing the $\log(f_1(x_{i_q}))$
zeta0	a double matrix containing the $qnorm(F_0(x_{i_q}))$
zeta1	a double matrix containing the $qnorm(F_1(x_{i_q}))$
OldR	a double matrix corresponding to the copula parameter
OldRinv	a double matrix corresponding to the inverse copula parameter
RhoIndex	a int matrix containing the index of lower triangular part of a matrix
threads_nb	an int the number of threads

### Value

a double vector containing the lower triangular part of the MLE of R

# Index

## \* datasets

PvalSets, [14](#)

PvalSets\_cor, [14](#)

Copula.Hconfig\_gaussian\_density, [2](#)

EM\_calibration\_gaussian, [3](#)

EM\_calibration\_gaussian\_memory, [4](#)

EM\_calibration\_indep, [5](#)

EM\_calibration\_indep\_memory, [5](#)

f1\_separation\_signed, [6](#)

FastKerFdr\_signed, [7](#)

FastKerFdr\_unsigned, [7](#)

fHconfig\_sum\_update\_gaussian\_copula\_ptr\_parallel,  
[8](#)

fHconfig\_sum\_update\_ptr\_parallel, [9](#)

gaussian\_copula\_density, [10](#)

GetH1AtLeast, [10](#)

GetH1AtLeast(), [11](#), [17](#)

GetH1Equal, [11](#)

GetH1Equal(), [11](#), [17](#)

GetHconfig, [12](#)

GetHconfig(), [3](#), [4](#), [6](#), [10](#), [11](#), [15](#), [17–19](#)

prior\_update\_arma\_ptr\_parallel, [12](#)

prior\_update\_gaussian\_copula\_ptr\_parallel,  
[13](#)

PvalSets, [14](#)

PvalSets\_cor, [14](#)

qch.fit, [15](#)

qch.fit(), [17](#)

qch.test, [16](#)

R.MLE, [18](#)

R.MLE.check, [18](#)

R.MLE.memory, [19](#)

R\_MLE\_update\_gaussian\_copula\_ptr\_parallel,  
[20](#)