

# Package ‘calm’

January 2, 2025

**Type** Package

**Title** Covariate Assisted Large-scale Multiple testing

**Version** 1.21.0

**Description** Statistical methods for multiple testing with covariate information. Traditional multiple testing methods only consider a list of test statistics, such as p-values. Our methods incorporate the auxiliary information, such as the lengths of gene coding regions or the minor allele frequencies of SNPs, to improve power.

**License** GPL (>=2)

**Encoding** UTF-8

**LazyData** false

**Imports** mgcv, stats, graphics

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**biocViews** Bayesian, DifferentialExpression, GeneExpression, Regression, Microarray, Sequencing, RNASeq, MultipleComparison, Genetics, ImmunoOncology, Metabolomics, Proteomics, Transcriptomics

**RoxygenNote** 6.1.1

**BugReports** <https://github.com/k22liang/calm/issues>

**git\_url** <https://git.bioconductor.org/packages/calm>

**git\_branch** devel

**git\_last\_commit** 2e49d35

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2025-01-02

**Author** Kun Liang [aut, cre]

**Maintainer** Kun Liang <kun.liang@uwaterloo.ca>

## Contents

calm . . . . .	2
CLfdr . . . . .	3
EstFDR . . . . .	5
EstNullProp_RB . . . . .	5
pso . . . . .	6
<b>Index</b>	<b>7</b>

---

calm	<i>Covariate Assisted Large-scale Multiple testing</i>
------	--

---

## Description

Statistical methods for multiple testing with covariate information.

## Details

Package:	calm
Type:	Package
Version:	0.9.0
Date:	2019-06-22
License:	GPL (>= 2)
LazyLoad:	yes

## Author(s)

Kun Liang <kun.liang@uwaterloo.ca>

Maintainer: Kun Liang <kun.liang@uwaterloo.ca>

## References

Liang, K (2019) *Empirical Bayes analysis of RNA sequencing experiments with auxiliary information*.

## See Also

[CLfdr](#)

---

CLfdr *Conditional local FDR (CLfdr)*

---

**Description**

CLfdr returns the local false discovery rate (FDR) conditional on auxiliary covariate information

**Usage**

```
CLfdr(x, y, pval = NULL, pi0.method = "RB", bw.init = NULL,
      bw = NULL, reltol = 1e-04, n.subsample = NULL, check.gam = FALSE,
      k.gam = NULL, info = TRUE)
```

**Arguments**

x	covariates, could be a vector of length $m$ or a matrix with $m$ rows.
y	a vector of $z$ -values of length $m$ .
pval	a vector of $p$ -values of length $m$ . The $p$ -values are only used to computed the overall true null proportion when <code>pi0.method="RB"</code> .
pi0.method	method to estimate the overall true null proportion ( $\pi_0$ ). "RB" for the right-boundary procedure (Liang and Nettleton, 2012, JRSSB) or "JC" (Jin and Cai, 2007, JASA).
bw.init	initial values for bandwidth, optional. If not specified, normal-reference rule will be used.
bw	bandwidth values.
reltol	relative tolerance in optim function.
n.subsample	size of the subsample when estimating bandwidth.
check.gam	indicator to perform <code>gam.check</code> function on the nonparametric fit.
k.gam	tuning parameter for <code>mgcv::gam</code> .
info	indicator to print out fitting information.

**Details**

In many multiple testing applications, the auxiliary information is widely available and can be useful. Such information can be summary statistics from a similar experiment or disease, the lengths of gene coding regions, and minor allele frequencies of SNPs.

$y$  is a vector of  $m$   $z$ -values, one of each hypothesis under test. The  $z$ -values follow  $N(0,1)$  if their corresponding null hypotheses are true. Other types of test statistics, such as  $t$ -statistics and  $p$ -values can be transformed to  $z$ -values. In practice, if the distribution of  $z$ -values is far from  $N(0,1)$ , recentering and rescaling of the  $z$ -values may be necessary.

$x$  contains auxiliary covariate information. For a single covariate,  $x$  should be a vector of length  $m$ . For multiple covariates,  $x$  should be a matrix with  $m$  rows. The covariates can be either continuous or ordered.

`pi0.method` specifies the method used to estimate the overall true null proportion. If the  $z$ -values are generated from the normal means model, the "JC" method from Jin and Cai (2007) JASA can be a good candidate. Otherwise, the right-boundary procedure ("RB", Liang and Nettleton, 2012, JRSSB) is used.

`bw` are bandwidth values for estimating local alternative density. Suppose there are  $p$  covariates, then `bw` should be a vector of  $p+1$  positive numerical values. By default, these bandwidth values are chosen by cross-validation to minimize a certain error measure. However, finding the optimal bandwidth values by cross-validation can be computationally intensive, especially when  $p$  is not small. If good estimates of bandwidth values are available, for example, from the analysis of a similar dataset, the bandwidth values can be specified explicitly to save time.

`reltol` specifies the relative convergence tolerance when choosing the bandwidth values (`bw`). It will be passed on to `stats::optim()`. For most analyses, the default value of  $1e-4$  provides reasonably good results. A smaller value such as  $1e-5$  or  $1e-6$  could be used for further improvement at the cost of more computation time.

### Value

<code>fdr</code>	a vector of local FDR estimates. <code>fdr[i]</code> is the posterior probability of the $i$ th null hypothesis is true given all the data. $1-fdr[i]$ is the posterior probability of being a signal (the corresponding null hypothesis is false).
<code>FDR</code>	a vector of FDR values (q-values), which can be used to control FDR at a certain level by thresholding the FDR values.
<code>pi0</code>	a vector of true null probability estimates. This contains the prior probabilities of being null.
<code>bw</code>	a vector of bandwidths for conditional alternative density estimation
<code>fit.gam</code>	an object of <code>mgcv::gam</code>

### Author(s)

Kun Liang, <kun.liang@uwaterloo.ca>

### References

Liang (2019), Empirical Bayes analysis of RNA sequencing experiments with auxiliary information, to appear in Annals of Applied Statistics

### Examples

```
data(pso)
ind.nm <- is.na(pso$tval_mic)
x <- pso$len_gene[ind.nm]
# normalize covariate
x <- rank(x)/length(x)
y <- pso$zval[ind.nm]
# assign names to the z-values helps to give names to the output variables
names(y) <- row.names(pso)[ind.nm]

fit.nm <- CLfdr(x=x, y=y)
fit.nm$fdr[1:5]
```

---

EstFDR	<i>FDR estimation</i>
--------	-----------------------

---

**Description**

False discovery rate (FDR) estimation from local FDR

**Usage**

```
EstFDR(fdr)
```

**Arguments**

fdr                    vector of local FDR

**Value**

the estimate of the FDR

**Examples**

```
lfdr <- c(runif(900), rbeta(100, 1, 10))
FDR <- EstFDR(lfdr)
sum(FDR < 0.05)
```

---

EstNullProp_RB	<i>Right-boundary procedure</i>
----------------	---------------------------------

---

**Description**

True null proportion ( $\pi_0$ ) estimator of Liang and Nettleton (2012), JRSSB

**Usage**

```
EstNullProp_RB(pval, lambda.vec = 0.05 * seq_len(19))
```

**Arguments**

pval                    vector of p-values  
lambda.vec              vector of lambda candidates (excluding 0 and 1)

**Value**

the estimate of the overall true null proportion

**Examples**

```
pval <- c(runif(900), rbeta(100, 1, 10))
EstNullProp_RB(pval)
```

---

pso

*Psoriasis RNA-seq dataset*

---

### Description

A dataset containing the test statistics to analyze an RNA-seq study of psoriasis.

### Usage

pso

### Format

A dataset with the following vectors:

**zval** 16490 z-values of genes with matching microarray data

**len\_gene** 16490 gene coding region length for zval

**tval\_mic** 16490 matching microarray t-statistics

### Source

Liang (2019), Empirical Bayes analysis of RNA sequencing experiments with auxiliary information, to appear in *Annals of Applied Statistics*;

### Examples

```
data(pso)
dim(pso)
# total number of genes without matching microarray data
sum(is.na(pso$tval_mic))
```

# Index

\* **datasets**

    pso, [6](#)

\* **package**

    calm, [2](#)

calm, [2](#)

CLfdr, [2](#), [3](#)

EstFDR, [5](#)

EstNullProp\_RB, [5](#)

pso, [6](#)

stats::optim(), [4](#)