

# Package ‘bacon’

December 23, 2024

**Type** Package

**Title** Controlling bias and inflation in association studies using the empirical null distribution

**Version** 1.34.0

**Author** Maarten van Iterson [aut, cre], Erik van Zwet [ctb]

**Maintainer** Maarten van Iterson <[mviterson@gmail.com](mailto:mviterson@gmail.com)>

**Description** Bacon can be used to remove inflation and bias often observed in epigenome- and transcriptome-wide association studies. To this end bacon constructs an empirical null distribution using a Gibbs Sampling algorithm by fitting a three-component normal mixture on z-scores.

**License** GPL (>= 2)

**Encoding** UTF-8

**Depends** R (>= 3.3), methods, stats, ggplot2, graphics, BiocParallel, ellipse

**Suggests** BiocStyle, knitr, rmarkdown, testthat, roxygen2

**biocViews** ImmunoOncology, StatisticalMethod, Bayesian, Regression, GenomeWideAssociation, Transcriptomics, RNASeq, MethylationArray, BatchEffect, MultipleComparison

**RoxygenNote** 7.3.1

**Collate** 'BaconClass.R' 'BaconMethods.R' 'bacon.R' 'normmixture.R'

**VignetteBuilder** knitr

**NeedsCompilation** yes

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** a065860

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

**Repository** Bioconductor 3.20

**Date/Publication** 2024-12-23

## Contents

|                   |           |
|-------------------|-----------|
| bacon             | 2         |
| Bacon-class       | 4         |
| bias              | 4         |
| dnormmix          | 5         |
| es                | 6         |
| estimates         | 6         |
| fit               | 7         |
| inflation         | 8         |
| meta              | 8         |
| plot,Bacon-method | 9         |
| plotnormmix       | 10        |
| posteriors        | 10        |
| pval              | 12        |
| rnormmix          | 12        |
| se                | 13        |
| topTable          | 14        |
| traces            | 15        |
| tstat             | 15        |
| <b>Index</b>      | <b>17</b> |

---

|       |                      |
|-------|----------------------|
| bacon | <i>Gibbs sampler</i> |
|-------|----------------------|

---

## Description

Gibbs Sampler Algorithm to fit a three component normal mixture to z-scores

## Usage

```
bacon(
  teststatistics = NULL,
  effectsizes = NULL,
  standarderrors = NULL,
  niter = 5000L,
  nburnin = 2000L,
  nbins = 1000,
  trim = 0.999,
  level = 0.05,
  na.exclude = FALSE,
  verbose = FALSE,
  priors = list(sigma = list(alpha = 1.28, beta = 0.36), mu = list(lambda = c(0, 3, -3),
    tau = c(1000, 100, 100)), epsilon = list(gamma = c(90, 5, 5))),
  globalSeed = 42,
  parallelSeed = 42
)
```

**Arguments**

|                |   |
|----------------|---|
| teststatistics | numeric vector or matrix of test-statistics   |
| effectsizes    | numeric vector or matrix of effect-sizes  |
| standarderrors | numeric vector or matrix of standard errors   |
| niter          | number of iterations  |
| nburnin        | length of the burnin period   |
| nbins          | default 1000 else bin test-statistics   |
| trim           | default 0.999 trimming test-statistics  |
| level          | significance level used to determine prop. null for starting values   |
| na.exclude     | see ?na.exclude   |
| verbose        | default FALSE   |
| priors         | list of parameters for the prior distributions  |
| globalSeed     | default 42 global seed. If set to NULL, randomization will occur for sequential and parallel bacon calls  |
| parallelSeed   | default 42 BiocParallel RNGseed. If input statistics are a matrix and globalSeed=NULL, setting parallelSeed=NULL will allow randomization across parallel processes within a bacon call and across separate calls to bacon. |

**Value**

object of class-Bacon

**Author(s)**

mvaniterson

**References**

Implementation is based on a version from Zhihui Liu <https://macsphere.mcmaster.ca/handle/11375/9368>

**Examples**

```
##simulate some test-statistic from a normal mixture
##and run bacon
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))
bc <- bacon(y)
##extract all estimated mixture parameters
estimates(bc)
##extract inflation
inflation(bc)
##extract bias
bias(bc)

##extract bias and inflation corrected test-statistics
head(tstat(bc))

##inspect the Gibbs Sampling output
traces(bc)
posteriors(bc)
fit(bc)
```

```

##simulate multiple sets of test-statistic from a normal mixture
##and run bacon
y <- matrix(rnormmix(10*2000, c(0.9, 0, 1, 0, 4, 1)), ncol=10)
bc <- bacon(y)
##extract all estimated mixture parameters
estimates(bc)
##extract only the inflation
inflation(bc)
##extract only the bias
bias(bc)
##extract bias and inflation corrected P-values
head(pval(bc))
##extract bias and inflation corrected test-statistics
head(tstat(bc))

```

---

Bacon-class

*An S4 class container for storing Gibbs Sampler input and output*

---

### Description

An S4 class container for storing Gibbs Sampler input and output

### Slots

teststatistics numeric vector or matrix of test-statistics  
 effectsizes numeric vector or matrix of effect-sizes  
 standarderrors numeric vector or matrix of standard errors  
 traces array of Gibbs Sampler traces  
 estimates vector or matrix of parameter estimates  
 priors list of parameters of for the prior distributions  
 niter number of iterations  
 nburnin length of the burnin period

---

bias

*Method to extract the estimated bias from the 'bacon'-object*

---

### Description

Method to extract the estimated bias from the 'bacon'-object

### Usage

```

bias(object)

## S4 method for signature 'Bacon'
bias(object)

```

**Arguments**

object            'bacon'-object

**Value**

vector or matrix of inflation

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))
##nbins = 100 to speed up the calculations
bc <- bacon(y, nbins=100)
bias(bc)
```

---

dnormmix

*density of a k-component normal mixture*

---

**Description**

density of a k-component normal mixture

**Usage**

```
dnormmix(x, theta)
```

**Arguments**

x                    x like dnorm(x, ...  
theta                parameters of the mixture proportion, mean and sd

**Details**

details follow

**Value**

density of a k-component normal mixture

**Author(s)**

mvaniterson

**Examples**

```
n <- 2000
theta <- c(0.8, 0, 1, 0, 4, 1)
x <- rnormmix(n, theta)
hist(x, freq=FALSE, n=100)
curve(dnormmix(x, theta), add=TRUE, lwd=2)
```

---

es *Method to extract inflation- and bias-corrected effect-sizes*

---

### Description

Method to extract inflation- and bias-corrected effect-sizes

### Usage

```
es(object, corrected = TRUE)

## S4 method for signature 'Bacon'
es(object, corrected = TRUE)
```

### Arguments

object            'bacon'-object  
corrected        optional return uncorrected

### Value

vector or matrix of effect-sizes

### See Also

[bacon](#)

### Examples

```
es <- replicate(6, rnormmix(2000, c(0.9, 0, 1, 0, 4, 1)))
se <- replicate(6, 0.8*sqrt(4/rchisq(2000,df=4)))
bc <- bacon(NULL, es, se)
head(es(bc))
```

---

estimates *Method to extract the estimated parameters from the 'bacon'-object*

---

### Description

Method to extract the estimated parameters from the 'bacon'-object

### Usage

```
estimates(object)

## S4 method for signature 'Bacon'
estimates(object)
```

### Arguments

object            'bacon'-object

**Value**

vector or matrix of estimates

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))
##nbins = 100 to speed up the calculations
bc <- bacon(y, nbins=100)
estimates(bc)
```

---

fit

*Method to plot mixture fit*

---

**Description**

Method to plot mixture fit

**Usage**

```
fit(object, index = 1, ...)

## S4 method for signature 'Bacon'
fit(object, index, col = "grey75", border = "grey75", ...)
```

**Arguments**

|        |  |
|--------|--|
| object | 'bacon'-object                                     |
| index  | if multiple sets of test-statistics where provided |
| ...    | additional plotting parameters                     |
| col    | line color default 'grey75'                        |
| border | border color 'grey75'                              |

**Value**

plot of the Gibbs Sampler mixture fit

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))
##nbins = 100 to speed up the calculations
bc <- bacon(y, nbins=100)
fit(bc)
```

inflation

*Method to extract the estimated inflation from the 'bacon'-object*

---

**Description**

Method to extract the estimated inflation from the 'bacon'-object

**Usage**

```
inflation(object)
```

```
## S4 method for signature 'Bacon'  
inflation(object)
```

**Arguments**

object           'bacon'-object

**Value**

vector or matrix of inflation

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))  
##nbins = 100 to speed up the calculations  
bc <- bacon(y, nbins=100)  
inflation(bc)
```

---

meta*fixed meta-analysis*

---

**Description**

Perform fixed meta-analysis using inflation and bias corrected effect-sizes and standard errors

**Usage**

```
meta(object, corrected = TRUE, ...)
```

```
## S4 method for signature 'Bacon'  
meta(object, corrected = TRUE, ...)
```



**Arguments**

|           |                             |
|-----------|-----------------------------|
| object    | 'bacon'-object              |
| corrected | optional return uncorrected |
| ...       | additional arguments        |

**Details**

TODO maybe add idea's from [http://www.netstorm.be/home/meta\\_analysis#metaAnalysisU](http://www.netstorm.be/home/meta_analysis#metaAnalysisU)

**Value**

object of class 'bacon' with added fixed-effect meta-analysis test-statistics, effect-sizes and standard-errors

**See Also**

[bacon](#)

**Examples**

```
es <- replicate(6, rnormmix(2000, c(0.9, 0, 1, 0, 4, 1)))
se <- replicate(6, 0.8*sqrt(4/rchisq(2000,df=4)))
bc <- bacon(NULL, es, se)
mbc <- meta(bc)
```

---

plot,Bacon-method      *plot hist or qq*

---

**Description**

simple ggplot2 plotting function for 'bacon'-object

**Usage**

```
## S4 method for signature 'Bacon'
plot(x, y, type = c("hist", "qq"))
```

**Arguments**

|      |                |
|------|----------------|
| x    | 'bacon'-object |
| y    | NULL           |
| type | hist or qq     |

**Value**

either qq-plot of P-values or histogram of Test-statistics

---

plotnormmix                    *plot normal mixtures*

---

**Description**

plot normal mixtures

**Usage**

```
plotnormmix(x, theta, ...)
```

**Arguments**

|       |  |
|-------|--|
| x     | vector of test statistics                    |
| theta | parameters describing the mixture components |
| ...   | arguments passed to hist                     |

**Details**

details follow

**Value**

return plot with histogram of the data and mixture and individual components

**Author(s)**

mvaniterson

**Examples**

```
n <- 2000
theta <- c(0.8, 0, 1, 0, 4, 1)
x <- rnormmix(n, theta)
plotnormmix(x, theta)
```

---

posteriors                    *Method to plot posterior distribution*

---

**Description**

Method to plot posterior distribution

**Usage**

```
posterior(
  object,
  thetas = c("sigma.0", "p.0"),
  index = 1,
  alphas = c(0.95, 0.9, 0.75),
  xlab = "",
  ylab = "",
  ...
)

## S4 method for signature 'Bacon'
posterior(
  object,
  thetas = c("sigma.0", "p.0"),
  index = 1,
  alphas = c(0.95, 0.9, 0.75),
  xlab = "",
  ylab = "",
  ...
)
```

**Arguments**

|        |  |
|--------|--|
| object | 'bacon'-object                                     |
| thetas | which thetas to plot                               |
| index  | if multiple sets of test-statistics where provided |
| alphas | significance level confidence ellipses             |
| xlab   | optional xlab                                      |
| ylab   | optional ylab                                      |
| ...    | additional plotting parameters                     |

**Value**

plot of the Gibbs Sampler posterior probabilities

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))
##nbins = 100 to speed up the calculations
bc <- bacon(y, nbins=100)
posterior(bc)
```

---

pval *Method to extract inflation- and bias-corrected P-values*

---

### Description

Method to extract inflation- and bias-corrected P-values

### Usage

```
pval(object, corrected = TRUE)
```

```
## S4 method for signature 'Bacon'
pval(object, corrected = TRUE)
```

### Arguments

object            'bacon'-object  
corrected        optional return uncorrected

### Value

vector or matrix of P-values

### See Also

[bacon](#)

### Examples

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))
bc <- bacon(y, nbins=100) #nbins = 100 to speed up the calculations
head(pval(bc))
```

---

rnormmix *sample from a normal mixture*

---

### Description

sample from a normal mixture

### Usage

```
rnormmix(n, theta, shuffle = TRUE)
```

### Arguments

n                    size  
theta                parameters  
shuffle              shuffle return vectors or keep nulls and alternative ordered (null, alts)

**Details**

details follow

**Value**

n samples from a normal mixture with parameters theta

**Author(s)**

mvaniterson

**Examples**

```
n <- 2000
theta <- c(0.8, 0, 1, 0, 4, 1)
x <- rnormmix(n, theta)
```

---

se

*Method to extract inflation- and bias-corrected standard errors*

---

**Description**

Method to extract inflation- and bias-corrected standard errors

**Usage**

```
se(object, corrected = TRUE)

## S4 method for signature 'Bacon'
se(object, corrected = TRUE)
```

**Arguments**

|           |                             |
|-----------|-----------------------------|
| object    | 'bacon'-object              |
| corrected | optional return uncorrected |

**Value**

vector or matrix of standard-errors

**See Also**

[bacon](#)

**Examples**

```
es <- replicate(6, rnormmix(2000, c(0.9, 0, 1, 0, 4, 1)))
se <- replicate(6, 0.8*sqrt(4/rchisq(2000,df=4)))
bc <- bacon(NULL, es, se)
head(se(bc))
```

---

`topTable`*topTable*

---

**Description**

Extract top features after meta analysis

**Usage**

```
topTable(  
  object,  
  number = 10,  
  adjust.method = "bonf",  
  sort.by = c("pval", "eff.size")  
)  
  
## S4 method for signature 'Bacon'  
topTable(  
  object,  
  number = 10,  
  adjust.method = "bonf",  
  sort.by = c("pval", "eff.size")  
)
```

**Arguments**

|                            |   |
|----------------------------|---|
| <code>object</code>        | 'bacon'-object  |
| <code>number</code>        | return specified number of top features, n=-1 return all features |
| <code>adjust.method</code> | P-value multiple testing adjustment method default bonferroni     |
| <code>sort.by</code>       | order results by pval or eff.size                                 |

**Value**

table with top features

**See Also**

[bacon](#)

**Examples**

```
es <- replicate(6, rnormmix(2000, c(0.9, 0, 1, 0, 4, 1)))  
se <- replicate(6, 0.8*sqrt(4/rchisq(2000,df=4)))  
bc <- bacon(NULL, es, se)  
mbc <- meta(bc)  
topTable(mbc)
```

---

|        |   |
|--------|---|
| traces | <i>Method to plot Gibbs sampling traces</i> |
|--------|---|

---

**Description**

Method to plot Gibbs sampling traces

**Usage**

```
traces(object, burnin = TRUE, index = 1)
```

```
## S4 method for signature 'Bacon'  
traces(object, burnin = TRUE, index = 1)
```

**Arguments**

|        |  |
|--------|--|
| object | 'bacon'-object                                     |
| burnin | include burnin period default true                 |
| index  | if multiple sets of test-statistics where provided |

**Value**

plot of the Gibbs Sampler traces

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))  
##nbins = 100 to speed up the calculations  
bc <- bacon(y, nbins=100)  
traces(bc)
```

---

|       |  |
|-------|--|
| tstat | <i>Method to extract inflation- and bias-corrected test-statistics</i> |
|-------|--|

---

**Description**

Method to extract inflation- and bias-corrected test-statistics

**Usage**

```
tstat(object, corrected = TRUE)
```

```
## S4 method for signature 'Bacon'  
tstat(object, corrected = TRUE)
```

**Arguments**

object            'bacon'-object  
corrected        optional return uncorrected

**Value**

vector or matrix of test-statistics

**See Also**

[bacon](#)

**Examples**

```
y <- rnormmix(2000, c(0.9, 0, 1, 0, 4, 1))  
##nbins = 100 to speed up the calculations  
bc <- bacon(y, nbins=100)  
head(tstat(bc))
```



# Index

bacon, [2](#), [5–9](#), [11–16](#)  
Bacon-class, [4](#)  
bias, [4](#)  
bias, Bacon-method (bias), [4](#)  
  
dnormmix, [5](#)  
  
es, [6](#)  
es, Bacon-method (es), [6](#)  
estimates, [6](#)  
estimates, Bacon-method (estimates), [6](#)  
  
fit, [7](#)  
fit, Bacon-method (fit), [7](#)  
  
inflation, [8](#)  
inflation, Bacon-method (inflation), [8](#)  
  
meta, [8](#)  
meta, Bacon-method (meta), [8](#)  
  
plot, Bacon-method, [9](#)  
plotnormmix, [10](#)  
posteriors, [10](#)  
posteriors, Bacon-method (posteriors), [10](#)  
pval, [12](#)  
pval, Bacon-method (pval), [12](#)  
  
rnormmix, [12](#)  
  
se, [13](#)  
se, Bacon-method (se), [13](#)  
  
topTable, [14](#)  
topTable, Bacon-method (topTable), [14](#)  
traces, [15](#)  
traces, Bacon-method (traces), [15](#)  
tstat, [15](#)  
tstat, Bacon-method (tstat), [15](#)