

# Package ‘MBAmethyl’

April 8, 2026

**Type** Package

**Title** Model-based analysis of DNA methylation data

**Version** 1.45.0

**Date** 2014-10-03

**Author** Tao Wang, Mengjie Chen

**Maintainer** Tao Wang <tao.wang.tw376@yale.edu>

**Description** This package provides a function for reconstructing DNA methylation values from raw measurements. It iteratively implements the group fused lars to smooth related-by-location methylation values and the constrained least squares to remove probe affinity effect across multiple sequences.

**Depends** R (>= 2.15)

**License** Artistic-2.0

**biocViews** DNAMethylation, MethylationArray

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

**git\_branch** devel

**git\_last\_commit** a70414d

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2026-04-07

## Contents

MBAmethyl-package . . . . .	2
MBAmethyl . . . . .	3
<b>Index</b>	<b>5</b>

---

MBAmethyl-package

*Model-based analysis of DNA methylation data*

---

## Description

This package provides functions for reconstructing DNA methylation values from raw measurements. It utilizes both the information from biological replicates and neighboring probes by explicitly modeling the probe-specific effect and encouraging the neighboring similarity by a group fused lasso penalty.

## Details

Package: MBAmethyl  
Type: Package  
Version: 0.99.0  
Date: 2014-08-24  
License: Artistic-2.0

## Author(s)

Tao Wang, Mengjie Chen

Maintainer: Tao Wang <tao.wang.tw376@yale.edu>

## References

~~ Literature or other references for background information ~~

## Examples

```
p <- 80
n <- 40
K <- 2
k <- K - 1
cp <- numeric()
L <- c(0, floor(p / K) * (1 : k), p)
cp <- floor(p / K) * (1 : k) + 1

## phi0: probe effects; theta0: true methylation values; part: partition of probe indices
phi0 <- runif(p, 0.5, 2.0)
theta0 <- matrix(0, p, n)
part <- list()

for (s in 1 : K) {
  part[[s]] <- (L[s] + 1) : L[s + 1]
```

```

    phi0[part[[s]]] <- phi0[part[[s]]] / sqrt(mean(phi0[part[[s]]]^2))
  }

  theta0[part[[1]], ] <- rep(1, length(part[[1]]))
  theta0[part[[2]], ] <- rep(1, length(part[[2]]))

  error <- matrix(runif(p * n, 0, 0.1), p, n)
  Y <- theta0 * phi0 + error
  fit <- MBAmethyl(Y, steps = 10)

```

MBAmethyl

*Model-based analysis of DNA methylation data***Description**

This function reconstructs DNA methylation values from raw measurements. It iteratively implements the group fused lars to smooth related-by-location methylation values and the constrained least squares to remove probe affinity effect across multiple sequences. It also contains a criterion-based method (AIC or BIC) for selecting the tuning parameter.

**Usage**

```
MBAmethyl(Y, wts = .defaultWeights(nrow(Y)), steps = min(dim(Y)) - 1)
```

**Arguments**

Y	An observed matrix (p x n) of methylation values (beta values); p is the number of probes and n is the number of samples;
wts	A pre-specified vector of weights. By default, we use the probe index-dependent weight scheme, $wts_i = \sqrt{p / i / (p - i)}$ for $i = 1, \dots, p$ ;
steps	Limit the number of steps taken. One can use this option to perform early stopping.

**Value**

ans.aic	A list corresponds to the AIC, containing estimated beta values, estimated probed effects, estimated change-point locations, residual sum of squares, and degree of freedom.
ans.bic	A list corresponds to the BIC, containing estimated beta values, estimated probed effects, estimated change-point locations, residual sum of squares, and degree of freedom.

**Author(s)**

Tao Wang, Mengjie Chen

## References

paper under review

## Examples

```
p <- 80
n <- 40
K <- 2
k <- K - 1
cp <- numeric()
L <- c(0, floor(p / K) * (1 : k), p)
cp <- floor(p / K) * (1 : k) + 1

## phi0: probe effects; theta0: true methylation values; part: partition of probe indices
phi0 <- runif(p, 0.5, 2.0)
theta0 <- matrix(0, p, n)
part <- list()

for (s in 1 : K) {
  part[[s]] <- (L[s] + 1) : L[s + 1]
  phi0[part[[s]]] <- phi0[part[[s]]] / sqrt(mean(phi0[part[[s]]]^2))
}

theta0[part[[1]], ] <- rep(1, length(part[[1]]))
theta0[part[[2]], ] <- rep(1, length(part[[2]]))

error <- matrix(runif(p * n, 0, 0.1), p, n)
Y <- theta0 * phi0 + error
fit <- MBAmethyl(Y, steps = 10)
```

# Index

\* **methylation**

MBAmethyl, [3](#)

\* **package**

MBAmethyl-package, [2](#)

MBAmethyl, [3](#)

MBAmethyl-package, [2](#)