Package: methFuse (via r-universe)

October 25, 2026

Title Functional Segmentation of the Methylome

Version 1.1.0

Description Implements FUSE (Functional Segmentation of DNA methylation data), a data-driven method for identifying spatially coherent DNA methylation segments from whole-genome bisulfite sequencing (WGBS) count data. The method performs hierarchical clustering of CpG sites based on methylated and unmethylated read counts across multiple samples and determines the optimal number of segments using an information criterion (AIC or BIC). Resulting segments represent regions with homogeneous methylation profiles across the input cohort while allowing sample-specific methylation levels. The package provides functions for clustering, model selection, tree cutting, segment-level summarization, and visualization. Input can be supplied as count matrices or extracted directly from 'BSseq' and 'methrix' objects.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, methods

Suggests bsseq, methrix, beachmat, GenomicRanges, SummarizedExperiment, DelayedArray, testthat (>= 3.0.0), knitr, rmarkdown, graphics

VignetteBuilder knitr

Config/testthat/edition 3

URL https://holmsusa.github.io/methFuse/

NeedsCompilation yes

Author Susanna Holmström [aut, cre] (ORCID: <https://orcid.org/0009-0005-7542-5913>), Antti Hakkinen [ctb, cph]

Maintainer Susanna Holmström <susanna.k.holmstrom@helsinki.fi>

Repository https://cran.r-universe.dev

Date/Publication 2026-03-04 10:10:02 UTC
RemoteUrl https://github.com/cran/methFuse
RemoteRef HEAD
RemoteSha 36ab1c58aea1c071427501357886e99ecd2fac23

Contents

fuse.cluster 2
fuse.cut.tree 3
fuse.segment 4
fuse.summary 5
number.of.clusters 7
plot.fuse_summary 8

Index **9**

`fuse.cluster` *Perform Hierarchical Clustering on Methylation Data*

Description

Produces a hierarchical clustering tree based on the input matrices of counts.

Usage

```
fuse.cluster(x, ...)

## Default S3 method:
fuse.cluster(x, K1, chr = NULL, pos = NULL, ...)

## S3 method for class 'BSseq'
fuse.cluster(x, ...)

## S3 method for class 'methrix'
fuse.cluster(x, ...)
```

Arguments

`x`	Input object. One of: **matrix** Unmethylated count matrix (K0). **BSseq** A `BSseq` object. **methrix** A `methrix` object.
`...`	Additional arguments if K0 is a matrix.
`K1`	Methylated count matrix (if x is matrix).
`chr`	Chromosome labels (if x is matrix).
`pos`	Genomic positions (if x is matrix.

Value

A clustering tree of class `hclust`.

Examples

```
# Example: Clustering generated data
set.seed(1234)
K0 <- matrix(
  rep(c(sample(0:20, 200, replace = TRUE), sample(20:40, 200, replace = TRUE)), 2),
  nrow = 100, byrow = TRUE
)
K1 <- matrix(
  rep(c(sample(20:40, 200, replace = TRUE), sample(0:20, 200, replace = TRUE)), 2),
  nrow = 100, byrow = TRUE
)
tree <- fuse.cluster(K0, K1)
tree
```

`fuse.cut.tree` *Cut Hierarchical Clustering Tree into Clusters*

Description

Divides the clustering tree into a specified number of clusters.

Usage

```
fuse.cut.tree(tree, k)
```

Arguments

`tree`	Clustering tree of class `hclust`.
`k`	Number of clusters

Value

A vector indicating which cluster each element in the original data frame belonged to

Examples

```
# Example: Cutting small tree in 2 segments
set.seed(1234)
K0 <- matrix(
 rep(c(sample(0:20, 200, replace = TRUE), sample(20:40, 200, replace = TRUE)), 2),
 nrow = 100, byrow = TRUE
)
K1 <- matrix(
 rep(c(sample(20:40, 200, replace = TRUE), sample(0:20, 200, replace = TRUE)), 2),
```

```
 nrow = 100, byrow = TRUE
)
tree <- fuse.cluster(K0, K1)

segments <- fuse.cut.tree(tree, 2)
segments
```

`fuse.segment` *Full FUSE segmentation pipeline*

Description

Performs the full FUSE segmentation workflow: hierarchical clustering, model selection, tree cutting, and genomic segment summarization.

Usage

```
fuse.segment(x, ...)

## Default S3 method:
fuse.segment(x, K1, chr, pos, method = c("BIC", "AIC"), ...)

## S3 method for class 'BSseq'
fuse.segment(x, method = c("BIC", "AIC"), ...)

## S3 method for class 'methrix'
fuse.segment(x, method = c("BIC", "AIC"), ...)
```

Arguments

`x`	Input object. One of: **matrix** Unmethylated count matrix (K0). **BSseq** A `BSseq` object. **methrix** A `methrix` object.
`...`	Additional arguments (matrix input only)
`K1`	Methylated count matrix (if x is matrix).
`chr`	Chromosome labels (if x is matrix).
`pos`	Genomic positions (if x is matrix.
`method`	Information criterion for model selection: `"BIC"` (default) or `"AIC"`. For internal use, `x` corresponds to the unmethylated count matrix (`K0`).

Details

`fuse.segment()` is an S3 generic with methods for:

`matrix` Raw count matrices (K0, K1) with genomic annotation.

`BSseq` Bioconductor `BSseq` objects.

`methrix` Bioconductor `methrix` objects (supports DelayedMatrix).

Value

An object of class `fuse_summary`, containing:

summary Data frame with one row per genomic segment.

betas_per_segment Matrix of per-sample methylation estimates.

raw_beta Per-CpG methylation estimates.

raw_pos Genomic positions of CpGs.

Automatic data extraction

For `BSseq` objects:

- Methylated counts are obtained via `getCoverage(x, "M")`
- Unmethylated counts via `getCoverage(x, "Cov") - M`
- Chromosome and position from `rowRanges(x)`

For `methrix` objects:

- Methylated counts via `get_matrix(x, "M")`
- Total coverage via `get_matrix(x, "C")`
- Unmethylated counts computed as `C - M`
- Genomic coordinates extracted from locus metadata

`fuse.summary` *Summarize FUSE Segmentation Results*

Description

Summarizes FUSE segmentation results into one row per segment, including genomic coordinates, CpG count, segment length, average methylation (beta), and stability flag based on likelihood testing. Also returns per-sample methylation estimates for each segment. Result can be visualized using plot(result).

Usage

```
fuse.summary(K0, K1, chr, pos, segments)
```

Arguments

K0	Integer or numeric matrix of unmethylated counts.
K1	Integer or numeric matrix of methylated counts.
chr	Character vector giving the chromosome for each site.
pos	Numeric vector giving genomic coordinates for each site.
segments	Integer vector giving segment IDs for each site in K0 and K1.

Value

A list with four elements:

summary A data frame with one row per segment and the following columns:

- Segment: Segment ID
- Chr: Chromosome
- Start: Start genomic coordinate
- End: End genomic coordinate
- CpGs: Number of CpGs in the segment
- Length: Genomic length (End - Start + 1)
- Beta: Average methylation across samples and CpGs
- Coherent: Logical indicator (TRUE if segment is coherently methylated, else FALSE)

betas_per_segment Matrix of per-sample methylation estimates for each segment (rows = segments, columns = samples).

raw_beta Average beta per CpG site, used for plotting.

raw_pos Genomic position for every given CpG site, used for plotting.

Examples

```
set.seed(1234)
K0 <- matrix(
  rep(c(sample(0:20, 200, replace = TRUE), sample(20:40, 200, replace = TRUE)), 2),
  nrow = 100, byrow = TRUE
)
K1 <- matrix(
  rep(c(sample(20:40, 200, replace = TRUE), sample(0:20, 200, replace = TRUE)), 2),
  nrow = 100, byrow = TRUE
)
tree <- fuse.cluster(K0, K1)
segments <- fuse.cut.tree(tree, 4)
res <- fuse.summary(K0, K1, rep("chr1", nrow(K0)), 1:nrow(K0), segments)
head(res$summary)
head(res$betas_per_segment)
```

`number.of.clusters` *Find Optimal Number of Clusters*

Description

Determines the optimal number of clusters to cut a hierarchical clustering tree, based on the selected information criterion (e.g., BIC or AIC).

Usage

```
number.of.clusters(tree, n, method = c("BIC", "AIC"))
```

Arguments

`tree`	Clustering tree of class `hclust`.
`n`	Number of samples in the original data.
`method`	Information criterion method. One of `"BIC"` or `"AIC"`.

Value

An integer representing the optimal number of clusters.

Examples

```
# Example: Determine number of clusters in dummy data set
set.seed(1234)
K0 <- matrix(
  rep(c(sample(0:20, 200, replace = TRUE), sample(20:40, 200, replace = TRUE)), 2),
  nrow = 100, byrow = TRUE
)
K1 <- matrix(
  rep(c(sample(20:40, 200, replace = TRUE), sample(0:20, 200, replace = TRUE)), 2),
  nrow = 100, byrow = TRUE
)
tree <- fuse.cluster(K0, K1)
k <- number.of.clusters(tree, ncol(K0), 'BIC')
k
```

`plot.fuse_summary` *Plot method for FUSE segmentation results*

Description

Plotting method for fuse_summary.

Usage

```
## S3 method for class 'fuse_summary'
plot(x, ..., segments_to_plot = 1:50)
```

Arguments

`x`	A fuse_summary object
`...`	Additional arguments
`segments_to_plot`	Integer vector of segment indices

Details

Raw CpG-level methylation values are shown as grey points. Segment-level methylation is shown as horizontal bars (red = hypermethylated, blue = hypomethylated).

Value

No return value, called for side effects.

Index

fuse.cluster, 2
fuse.cut.tree, 3
fuse.segment, 4
fuse.summary, 5

number.of.clusters, 7

plot.fuse_summary, 8