Package ‘mmcmcBayes’

December 21, 2025

Type Package

Title Multistage MCMC Method for Detecting DMRs

Version 0.2.0

Description Implements differential methylation region (DMR) detection using a multistage Markov chain Monte Carlo (MCMC) algorithm based on the alpha-skew generalized normal (ASGN) distribution. Version 0.2.0 removes the Anderson-Darling test stage, improves computational efficiency of the core ASGN and multistage MCMC routines, and adds convenience functions for summarizing and visualizing detected DMRs. The methodology is based on Yang (2025) <https://www.proquest.com/docview/3218878972>.

License GPL-3

URL https://github.com/zyang1919/mmcmcBayes

BugReports https://github.com/zyang1919/mmcmcBayes/issues

Depends R (>= 4.0.0)

Imports stats, graphics, MCMCpack (>= 1.4-0)

Suggests kSamples, testthat (>= 3.0.0), knitr, rmarkdown

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.3

NeedsCompilation no

Author Zhexuan Yang [aut, cre],
Duchwan Ryu [aut],
Feng Luan [aut]

Maintainer Zhexuan Yang <zky5198@psu.edu>

Repository CRAN

Date/Publication 2025-12-20 23:50:02 UTC

Contents

asgn_func 2
cancer_demo 4
compare_dmrs 4
mmcmcBayes 6
normal_demo 9
plot_dmr_region 9
summarize_dmrs 11

Index **13**

`asgn_func` *Fit an Alpha-Skewed Generalized Normal (ASGN) Distribution by MCMC*

Description

`asgn_func()` fits an alpha-skewed generalized normal (ASGN) distribution to univariate numeric data using an MCMC algorithm. The ASGN family provides a flexible parametric model that can accommodate skewness and, for certain parameter values, bimodal density shapes.

Usage

```
asgn_func(
  data,
  priors = NULL,
  mcmc = list(nburn = 5000, niter = 10000, thin = 1),
  seed = NULL
)
```

Arguments

`data` A numeric vector or a one-column matrix. In typical use this is a vector of sample-wise regional mean M-values (one per sample).

`priors` Optional list of prior hyperparameters. If `NULL`, weakly informative priors are constructed from the data. If provided, expected components are `alpha`, `mu`, and `sigma2`.

`mcmc` A list of MCMC parameters:

- `nburn`: Number of burn-in iterations (default: 5000)
- `niter`: Number of sampling iterations (default: 10000)
- `thin`: Thinning interval (default: 1)

`seed` Optional integer random seed for reproducibility. If `NULL`, no seed is set.

Details

The input `data` may be any univariate numeric sample (vector or a one-column matrix). In the `mmcmcBayes` workflow, `data` is typically a vector of sample-wise regional mean M-values (one value per sample). However, `asgn_func()` is not specific to DNA methylation and can be used more generally for fitting skewed or potentially bimodal continuous data.

Value

A list with the following elements:

- `posteriors`: Vector of posterior means for `alpha`, `mu`, and `sigma2`.
- `summary`: Data frame containing parameter estimates and 95% credible intervals with columns `parameter`, `estimate`, `lower_ci`, and `upper_ci`.

If there are fewer than two non-missing observations, a default value `posteriors = c(1, 0, 1)` is returned, and `summary` is omitted.

Author(s)

Zhexuan Yang, Duchwan Ryu, and Feng Luan

References

Mahmoudi, E., Jafari, H., & Meshkat, R. (2019). Alpha-skew generalized normal distribution and its applications. *Applications and Applied Mathematics: An International Journal (AAM)*, **14**, 784-804.

See Also

`mmcmcBayes` for the main DMR detection function,

Examples

```
# Generate sample data
set.seed(2021)
dt <- rgamma(1000, shape = 2, rate = 1)
dt <- as.matrix(dt, ncol=1)

result <- asgn_func(dt)
print(result$summary)
```

cancer_demo — *Cancer Methylation Demo Data*

Description

A demonstration dataset containing methylation M-values for cancer samples. Used for testing and examples in the mmcmcBayes package.

Usage

```
cancer_demo
```

Format

A data frame with CpG sites and methylation values.

Source

The first 5000 CpG sites of Chromosome 6 of 450K dataset.

Examples

```
data(cancer_demo)
head(cancer_demo)
```

compare_dmrs — *Compare Differentially Methylated Regions (DMRs) from Two Methods*

Description

`compare_dmrs()` identifies overlapping regions between two sets of differentially methylated regions (DMRs), typically obtained from two different detection methods. It reports pairwise overlaps and a simple overlap percentage that can be used to assess consistency between methods.

Usage

```
compare_dmrs(rst1, rst2)
```

Arguments

`rst1`	A data frame containing the first set of DMR results. Must contain at least the columns `Chromosome`, `Start_CpG`, and `End_CpG`.
`rst2`	A data frame containing the second set of DMR results, in the same format as `rst1`. The `Chromosome` values must be comparable to those in `rst1` (e.g., both using `"chr6"` or both using `"6"`).

Details

This function compares genomic regions between two DMR result objects. For each region in `rst1`, it searches for regions in `rst2` on the same chromosome that have any overlap in CpG index range (partial or complete).

CpG identifiers in `Start_CpG` and `End_CpG` are assumed to contain an embedded numeric component that reflects their ordering along the genome (e.g., `"cg00017002"`). Internally, these IDs are converted to numeric values by stripping non-digit characters; rows for which this conversion fails are removed before comparison.

For each overlapping pair of regions, the function computes

- the size of the overlapping segment (in CpG index units), and
- an overlap percentage, defined as

$$\text{overlap_size}/\max(\text{length_method1}, \text{length_method2}) \times 100,$$

  where `length_method1` and `length_method2` are the lengths of the two regions in CpG index units.

The overlap percentage is therefore symmetric in the two methods and can be interpreted as "how much of the larger region is covered by the overlap."

Value

A data frame with one row per overlapping pair of regions and the columns:

- `Chromosome`: chromosome name of the overlapping regions.
- `Start_CpG_Method1`: start CpG ID from `rst1`.
- `End_CpG_Method1`: end CpG ID from `rst1`.
- `Start_CpG_Method2`: start CpG ID from `rst2`.
- `End_CpG_Method2`: end CpG ID from `rst2`.
- `Overlap_Percentage`: overlap percentage relative to the larger of the two regions (in CpG index units), rounded to two decimals.

Returns `NULL` if no overlaps are found or if, after cleaning, one of the inputs has no usable rows.

Author(s)

Zhexuan Yang, Duchwan Ryu, and Feng Luan

See Also

Related functions in this package: `mmcmcBayes` for DMR detection using multi-stage MCMC, `asgn_func` for parameter estimation with ASGN distribution

Examples

```
# Create sample DMR results
dmr_method1 <- data.frame(
  Chromosome = c("chr1", "chr1", "chr2"),
  Start_CpG = c("cg0001", "cg0050", "cg0100"),
  End_CpG = c("cg0020", "cg0070", "cg0150")
)

dmr_method2 <- data.frame(
  Chromosome = c("chr1", "chr2", "chr2"),
  Start_CpG = c("cg0005", "cg0120", "cg0090"),
  End_CpG = c("cg0025", "cg0160", "cg0110")
)

# Compare overlapping regions
overlaps <- compare_dmrs(dmr_method1, dmr_method2)
```

`mmcmcBayes` *Multi-stage MCMC Bayesian Method for DMR Detection*

Description

This function implements a multistage MCMC Bayesian method for detecting differentially methylated regions (DMRs) between two groups (typically cancer and normal). The method operates on methylation measurements on the M-values.

For each candidate region and for each group, the function summarizes the region at the sample level by averaging M-values across CpG sites within the region. These sample-wise means are using an alpha-skewed generalized normal (ASGN) distribution. A Bayes factor (BF) comparing the two groups is then used within a multistage region-splitting scheme to identify the DMRs.

Usage

```
mmcmcBayes(
  cancer_data,
  normal_data,
  stage = 1,
  max_stages = 3,
  num_splits = 50,
  mcmc = NULL,
  priors_cancer = NULL,
  priors_normal = NULL,
  bf_thresholds = c(0.5, 0.8, 1.05)
)
```

Arguments

- `cancer_data` — A data frame of methylation data for the cancer group. Rows correspond to CpG sites and columns to variables. The first two columns must be `CpG_ID` and `Chromosome`, and the remaining columns must be numeric M-values for cancer samples.
- `normal_data` — A data frame of methylation data for the normal group in the same format and CpG ordering as `cancer_data`.
- `stage` — Integer indicating the starting stage for the multistage analysis. Usually left at the default `stage = 1`.
- `max_stages` — Integer giving the maximum number of stages in the splitting procedure (default `3`). Larger values allow deeper splitting of regions at the cost of additional computation.
- `num_splits` — Integer giving the number of subregions created when a region is split at each stage (default `50`). Increasing `num_splits` typically improves sensitivity but increases computation time.
- `mcmc` — A list of MCMC control parameters passed to `asgn_func`. Expected components are `nburn` (burn-in iterations), `niter` (total iterations), and `thin` (thinning interval). If `NULL`, default values `list(nburn = 5000, niter = 10000, thin = 1)` are used.
- `priors_cancer` — Optional list of prior hyperparameters for the ASGN model in the cancer group, passed to `asgn_func`. If `NULL`, default priors from `asgn_func` are used.
- `priors_normal` — Optional list of prior hyperparameters for the ASGN model in the normal group, passed to `asgn_func`. If `NULL`, default priors from `asgn_func` are used.
- `bf_thresholds` — Numeric vector of Bayes factor thresholds, one for each stage (e.g., `c(0.5, 0.8, 1.05)`). If the length of `bf_thresholds` is shorter than `max_stages`, the last value is recycled so that each stage has an associated threshold. If `NULL`, default thresholds `c(0.5, 0.8, 1.05)` are used.

Details

The inputs `cancer_data` and `normal_data` must have the same set of CpG sites in the same order. Each row corresponds to a CpG site, and the first two columns are required to be:

- `CpG_ID`: character CpG identifier.
- `Chromosome`: chromosome label (integer or character).

All remaining columns are assumed to be numeric M-values for individual samples in the respective group (e.g., `"M_sample1"`, `"M_sample2"`, ...).

For each group, a sample wise mean M-values are computed and passed to `asgn_func` to obtain posterior mean of the ASGN parameters. A Bayes factor (BF) comparing the two groups is then computed for the current region. If the BF exceeds a stage-specific threshold, the region is either accepted as a DMR (at the final stage) or split into subregions and analyzed at the next stage. This continues until either `max_stages` is reached or no subregion passes the BF thresholds.

The values used in the examples are intentionally small to ensure fast execution and are not intended as recommended settings for real analyses.

Value

A data frame with one row per detected DMR and the following columns:

- `Chromosome`: chromosome label.
- `Start_CpG`: CpG ID where the region starts.
- `End_CpG`: CpG ID where the region ends.
- `CpG_Count`: number of CpG sites in the region.
- `Decision_Value`: final Bayes factor for the region.
- `Stage`: stage at which the region was detected.

If no regions pass the BF thresholds, `NULL` is returned.

Author(s)

Zhexuan Yang, Duchwan Ryu, and Feng Luan

See Also

`asgn_func` for ASGN parameter estimation, `plot_dmr_region` for visualizing individual DMR profiles, `summarize_dmrs` for summarizing detected regions, `compare_dmrs` for comparing DMR sets.

Examples

```
# Load the datasets
data(cancer_demo)
data(normal_demo)

mcmc <- list(nburn = 1000, niter = 2000, thin = 1)

set.seed(2021)

rst <- mmcmcBayes(cancer_demo, normal_demo,
                  stage = 1,
                  max_stages = 2,
                  num_splits = 5,
                  mcmc = mcmc,
                  priors_cancer = NULL,
                  priors_normal = NULL,
                  bf_thresholds = c(0.5, 0.8, 1.05))

print(rst)
```

normal_demo — *Normal Methylation Demo Data*

Description

A demonstration dataset containing methylation M-values for normal samples. Used for testing and examples in the mmcmcBayes package.

Usage

```
normal_demo
```

Format

A data frame with CpG sites and methylation values.

Source

Derived from the first 5000 CpG sites of Chromosome 6 of the 450K dataset (processed and reduced for example use only).

Examples

```
data(normal_demo)
head(normal_demo)
```

plot_dmr_region — *Plot mean methylation profiles for a selected DMR*

Description

Visualize a single DMR detected by `mmcmcBayes()` by plotting the mean M-values across CpG sites in the region for the cancer and normal groups.

Usage

```
plot_dmr_region(
  dmr_table,
  cancer_data,
  normal_data,
  dmr_index = 1,
  main = NULL,
  ...
)
```

Arguments

`dmr_table`	A data frame of DMRs, typically the result of `mmcmcBayes()` with `return_mcmc = FALSE`.
`cancer_data`	A data frame of cancer group methylation data in the format used by `mmcmcBayes()` (rows = CpGs, columns = metadata + M-values for cancer samples).
`normal_data`	A data frame of normal-group methylation data in the same format and CpG ordering as `cancer_data`.
`dmr_index`	Integer index of the DMR to plot (row index in `dmr_table`). Defaults to 1.
`main`	Optional main title for the plot. If `NULL`, a title is constructed from the region information.
`...`	Additional arguments passed to `plot()`.

Details

This function takes the DMR table returned by `mmcmcBayes()` together with the original `cancer_data` and `normal_data` matrices used in the analysis. It selects one DMR by index and computes, for each CpG site in the region, the mean M-value across cancer samples and across normal samples. These two mean profiles are then plotted against the CpG index within the region.

The input data frames `cancer_data` and `normal_data` are expected to have the same CpG sites in the same order, with at least the columns `CpG_ID` and `Chromosome`, followed by one column per sample containing M-values. Sample columns are identified automatically as those whose names start with `"M_sample"`.

CpG sites whose mean M-value is `NA` or non finite in either group are removed prior to plotting. The x axis index therefore refers to the filtered CpG positions within the selected region.

Value

The function is called for its side effect of creating a plot. It returns (invisibly) a list containing the CpG IDs in the region and the mean profiles for cancer and normal groups.

Examples

```
data(cancer_demo)
data(normal_demo)

set.seed(2021)
mcmc <- list(nburn = 1000, niter = 2000, thin = 1)

dmr_res <- mmcmcBayes(
  cancer_demo, normal_demo,
  stage         = 1,
  max_stages    = 2,
  num_splits    = 5,
  mcmc          = mcmc,
  priors_cancer = NULL,
  priors_normal = NULL,
  bf_thresholds = c(0.5, 0.8, 1.05)
)
```

```
plot_dmr_region(dmr_res, cancer_demo, normal_demo, dmr_index = 1)
```

---

summarize_dmrs *Summarize DMR results from* mmcmcBayes

Description

Convenience function to summarize the DMR table returned by mmcmcBayes(). It reports the number of detected regions, summaries of region sizes and decision values, and counts by chromosome and (optionally) by stage.

Usage

```
summarize_dmrs(dmr_table)
```

Arguments

dmr_table	A data frame of DMRs, typically the return value of mmcmcBayes().

Details

The function is designed to work with the data frame returned by mmcmcBayes(). At minimum, the input dmr_table is expected to contain the columns Chromosome, CpG_Count, and Decision_Value. If a Stage column is present, counts by stage are also reported.

If dmr_table is NULL or has zero rows, a summary object with n_dmrs = 0 is returned and all other components are NULL.

Value

An object of class "mmcmcBayes_dmr_summary", which is a list with elements:

- n_dmrs: Total number of detected DMRs.
- size_summary: Named numeric vector of summary statistics for CpG_Count (min, 1st quartile, median, mean, 3rd quartile, max), or NULL if CpG_Count is not available.
- decision_summary: Named numeric vector of summary statistics for Decision_Value (min, 1st quartile, median, mean, 3rd quartile, max), or NULL if Decision_Value is not available.
- by_chromosome: A data frame with counts of DMRs per chromosome, or NULL if Chromosome is not available.
- by_stage: A data frame with counts of DMRs per stage, if a Stage column is present; otherwise NULL.

Examples

```
data(cancer_demo)
data(normal_demo)

mcmc <- list(nburn = 1000, niter = 2000, thin = 1)

set.seed(2021)
dmr_res <- mmcmcBayes(
  cancer_demo, normal_demo,
  stage         = 1,
  max_stages    = 2,
  num_splits    = 5,
  mcmc          = mcmc,
  priors_cancer = NULL,
  priors_normal = NULL,
  bf_thresholds = c(0.5, 0.8, 1.05)
)

summary_res <- summarize_dmrs(dmr_res)
summary_res
```

Index

∗ **datasets**
    cancer_demo, 4
    normal_demo, 9

asgn_func, 2, *5*, *7*, *8*

cancer_demo, 4
compare_dmrs, 4, *8*

mmcmcBayes, *3*, *5*, 6

normal_demo, 9

plot_dmr_region, *8*, 9

summarize_dmrs, *8*, 11