Package 'vmrseq'

July 3, 2025

Type Package

Title Probabilistic Modeling of Single-cell Methylation Heterogeneity

Version 1.0.1

Description High-throughput single-cell measurements of DNA methylation allows studying intercellular epigenetic heterogeneity, but this task faces the challenges of sparsity and noise. We present vmrseq, a statistical method that overcomes these challenges and identifies variably methylated regions accurately and robustly.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

biocViews Software, ImmunoOncology, DNAMethylation, Epigenetics, SingleCell, Sequencing, WholeGenome

Depends R (>= 4.5.0)

Imports bumphunter, dplyr, BiocParallel, DelayedArray, GenomicRanges, ggplot2, methods, tidyr, locfit, gamlss.dist, recommenderlab, HDF5Array, data.table, SummarizedExperiment, IRanges, S4Vectors

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/nshen7/vmrseq

BugReports https://github.com/nshen7/vmrseq/issues

git_url https://git.bioconductor.org/packages/vmrseq

git_branch RELEASE_3_21

git_last_commit 2e21cc1

git_last_commit_date 2025-04-30

Repository Bioconductor 3.21

Date/Publication 2025-07-02

Author Ning Shen [aut, cre]

Maintainer Ning Shen <ning.shen.wk@gmail.com>

Contents

`cell_1` *cell_1*

Description

This dataset is an example of a single-cell file that can be input to `vmrseq::poolData` function. It contains the first 10000 CpG sites in a random cell from mouse frontal cortext dataset published by Luo et al. (2017).

Usage

```
data(cell_1)
```

Format

A data frame with 10000 rows and 5 variables (no column names):

chr Chromosome

mc_count Genomic coordinate

pos Strand information

strand Number of methylated reads

total Number of reads in total

References

Luo, Chongyuan et al. *Single-cell methylomes identify neuronal subtypes and regulatory elements in mammalian cortex.*. Science (New York, N.Y.) vol. 357,6351 (2017): 600-604.

Examples

```
data(cell_1)
head(cell_1)
```

`cell_2` *cell_2*

Description

This dataset is an example of a single-cell file that can be input to `vmrseq::poolData` function. It contains the first 10000 CpG sites in a random cell from mouse frontal cortext dataset published by Luo et al. (2017).

Usage

```
data(cell_2)
```

Format

A data frame with 10000 rows and 5 variables (no column names):

chr Chromosome

mc_count Genomic coordinate

pos Strand information

strand Number of methylated reads

total Number of reads in total

References

Luo, Chongyuan et al. *Single-cell methylomes identify neuronal subtypes and regulatory elements in mammalian cortex.*. Science (New York, N.Y.) vol. 357,6351 (2017): 600-604.

Examples

```
data(cell_2)
head(cell_2)
```

`cell_3` *cell_3*

Description

This dataset is an example of a single-cell file that can be input to `vmrseq::poolData` function. It contains the first 10000 CpG sites in a random cell from mouse frontal cortext dataset published by Luo et al. (2017).

Usage

```
data(cell_3)
```

Format

A data frame with 10000 rows and 5 variables (no column names):

chr Chromosome

mc_count Genomic coordinate

pos Strand information

strand Number of methylated reads

total Number of reads in total

References

Luo, Chongyuan et al. *Single-cell methylomes identify neuronal subtypes and regulatory elements in mammalian cortex.*. Science (New York, N.Y.) vol. 357,6351 (2017): 600-604.

Examples

```
data(cell_3)
head(cell_3)
```

`HDF5NAdrop2matrix` *Coerce a NA-dropped HDF5 matrix back to regular matrix form where NAs are not dropped to 0.*

Description

Coerce a NA-dropped HDF5 matrix back to regular matrix form where NAs are not dropped to 0.

Usage

```
HDF5NAdrop2matrix(hdf5_assay)
```

Arguments

hdf5_assay	an DelayedMatrix object with dropped NA values (e.g. an assay from HDF5SummarizedExpriment object saved by vmrseq::data.pool)

Value

Returns a matrix.

Examples

```
# load example data
toy.se <- HDF5Array::loadHDF5SummarizedExperiment(system.file("extdata", "toy", package = "vmrseq"))

# run the function
HDF5NAdrop2matrix(SummarizedExperiment::assays(toy.se)$M_mat)
```

---

poolData *Pool single-cell file together into an HDF5-based SummarizedExperiment object with sparse matrix representation.*

Description

This function pools individual-cell CpG read files into a SummarizedExperiment object so that it can be input to the `vmrseq.smooth` function. Note that in each cell, sites with hemimethylation or intermediate methylation levels (i.e., 0 < meth_read/total_read < 1) will be removed.

Usage

```
poolData(
  cellFiles,
  sep,
  writeDir,
  chrNames,
  colData = NULL,
  sparseNAdrop = TRUE
)
```

Arguments

cellFiles	Vector of character strings indicating single-cell file paths you wish to pool into SE object(s). Cell files should be in BED-like format, where the first 5 columns in each file must be: <chr>, <pos>, <strand>, <meth_read>, <total_read>, in strict order. Each row contains info of one CpG. Strand is assumed to be properly filpped. Cell files can be zipped ones in .gz format.
sep	the field separator character. Values on each line of cell file are separated by this character.

writeDir	A single character string indicating a folder directory where you wish to store the processed SE object(s). The SE will be stored in HDF5 format.
chrNames	Single or a vector of character strings representing chromosome names in cell files. Only chromosomes listed in selectChrs will be processed.
colData	A DataFrame or data.frame object containing colData for the SE object (applied to all chromosomes).
sparseNAdrop	Logical value indicating whether or not to use NA-dropped sparseMatrix representation. 'NA-dropped' means replacing NAs as zeros and then represent 0 as a very small positive value close to 0 so that the data matrix is converted to a traditional sparseMatrix form with 0 as default entry value. (see sparseNAMatrix-class from package recommenderlab). Using NA-dropped sparseMatrix representation helps to save RAM during data processing, as well as storage space saving to disk. Currently only supports TRUE!!

Value

Directly write out to the 'writeDir' and does not return anything.

Examples

```
poolData(cellFiles = your_cell_file_list, sep = ",", chrNames = "chr1", writeDir = "your/write/path")
```

---

regionSummary *Compute regional methylation information for individual cells.*

Description

This function summarize the methylated CpG count and total CpG count per region per cell.

Usage

```
regionSummary(
  SE,
  region_ranges,
  sparseNAdrop = is_sparse(assays(SE)[[1]]),
  verbose = TRUE,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

SE	SummarizedExperiment object with one (and only one) assay that contains *binary* methylation status of CpG sites in individual cells. In usual analysis workflow (of vmrseq), SE should be the output of vmrseq::data.pool.
region_ranges	GRanges object that contains genomic coordinates of regions of interest.

sparseNAdrop	logical value that represents whether the NA values are droppped in the input SE object. SE objects output by vmrseq::data.pool are NA dropped. See ?vmrseq::data.pool for details about NA-dropped representation.
verbose	logical value that indicates whether progress messages should be printed to std-out. Defaults value is TRUE.
BPPARAM	a BiocParallelParam object to specify the parallel backend. The default option is BiocParallel::bpparam() which will automatically creates a cluster appropriate for the operating system.

Value

Returns a SummarizedExperiment object of dimension (# regions x # cells). In total the object contains thress assays. Specifically, 'M' and 'Cov' represent the number of methylated CpGs and the number of covered CpGs in each region per cell; and 'MF' (stands for methylation fraction) represents the regional average methylation level computed by 'M/Cov'.

Examples

```
# load example data
toy.se <- HDF5Array::loadHDF5SummarizedExperiment(system.file("extdata", "toy", package = "vmrseq"))
data(toy.results)

# run vmrseq.fit
regions.se <- regionSummary(SE = toy.se, region_ranges = toy.results$vmr.ranges[1:3])
regions.se
```

---

toy.gr *toy.gr*

Description

This GRanges object. It was generated by running the example from vmrseq::vmrseqSmooth function.

Usage

```
data(toy.gr)
```

Format

A GRanges object.

Examples

```
data(toy.gr)
toy.gr
```

toy.results *toy.results*

Description

This `list` object. It was generated by running the example from `vmrseq::vmrseqFit` function.

Usage

```
data(toy.results)
```

Format

A `GRanges` object.

Examples

```
data(toy.results)
toy.results
```

tpEstimate *Estimate transition probilities conditioning on CpG-CpG distance*

Description

This function first computes transition probilities conditioning on CpG-CpG distance in each cell. Then the probs from individual cells are smoothed over CpG-CpG distance using ‘loess‘ with inverse-variance fitting.

Usage

```
tpEstimate(
  list,
  max_dist_bp = 2000,
  buffer_bp = 3000,
  lags = 1:10,
  BPPARAM = bpparam(),
  degree = 2,
  span = 0.02,
  ...
)
```

Arguments

list	a list of data.frame objects. Each data.frame contains information of 1 unit of training data (can be a cell or a subtype) and should have 3 columns in strict rder of: (chr), (pos), (binary methyl value). Column names are not necessary.
max_dist_bp	positive integer value indicating the maximum CpG-CpG distance in base pairs before the transition probabilities reach constant value. Default value is 2000.
buffer_bp	length of buffer in base pairs used to fit smoothing curve near maximum distance and plot diagnostics. Default is 3000.
lags	a vector indicating possible number of lagged CpGs for estimation of transition probabilities.
BPPARAM	a BiocParallelParam object to specify the parallel backend. The default option is BiocParallel::bpparam() which will automatically creates a cluster appropriate for the operating system.
degree	'degree' argument for 'loess' function
span	'span' argument for 'loess' function
...	additional arguments passed into the 'loess' function.

Value

a 'transitProbs' object. Postfixes rule in the output variables: P(0|0) => '00'; P(0|1) => '01'; P(1|0) => '10'; P(1|1) => '11'.

Examples

```
# load example data
data(cell_1, cell_2, cell_3)

# process the data to align with input format
library(dplyr)
df_1 <- cell_1 %>% mutate(meth = round(mc_count / total)) %>% select(chr, pos, meth)
df_2 <- cell_2 %>% mutate(meth = round(mc_count / total)) %>% select(chr, pos, meth)
df_3 <- cell_3 %>% mutate(meth = round(mc_count / total)) %>% select(chr, pos, meth)

# run tpEstimate
list <- list(df_1, df_2, df_3)
tpEstimate(list)
```

---

tpPlot *Plot transition probability distribution*

Description

Plot transition probability distribution

Usage

```
tpPlot(tp, line_size = 0.2, plot_train = TRUE, point_size = 0.2)
```

Arguments

tp	'transitProbs' object storing information about trained transition probabilities. Can be obtained from function 'estimTransitProbs'
line_size	size of fitted loess smooth line. Default value is 0.2.
plot_train	logical value indiating whether to plot training data. Default is TRUE.
point_size	size of training data points. Only applicable when 'plot_train = T'. Default value is 0.2.

Value

A plot of the transition probability distribution

Examples

```
tpPlot(vmrseq:::tp0)
```

---

vmrseqFit *Construct candidate regions and detect variably methylated regions.*

Description

Construct candidate regions (CRs) by taking groups of consecutive loci that exceed threshold on the variance of smoothed relative methylation levels and detect variably methylated regions (VMRs) by optimizing a hidden Markov model (HMM).

Usage

```
vmrseqFit(
  gr,
  alpha = 0.05,
  maxGap = 2000,
  stage1only = FALSE,
  minNumCR = 5,
  minNumVMR = 5,
  gradient = TRUE,
  tp = NULL,
  control = vmrseqOptimControl(),
  verbose = TRUE,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

gr	GRanges object output by vmrseq::vmrseqSmooth, containing genomic coordinates (chr, start, end) and summarized information (meth, total, var) of CpG sites in the input dataset.
alpha	positive scalar value between 0 and 1 that represents the designated significance level for determining variance threshold of candidate regions construction. The variance threshold is determined by taking 1-alpha quantile value of an approximate null distribution of variance simulated from the beta priors of emission probability in the hidden Markov model. Default value of alpha is 0.05.
maxGap	integer value representing maximum number of base pairs in between neighboring CpGs to be included in the same VMR. Default value is 2000 bp.
stage1only	boolean value indicating whether the algorithm should run stage 1 of vmrseq (the construction of candidate regions) only. If set to TRUE, the function will output only the candidate regions. Default is FALSE.
minNumCR	positive integer value representing the minimum number of CpG sites within a candidate region. Default value is 5.
minNumVMR	positive integer value representing the minimum number of CpG sites within a variably methylated region. Default value is 5.
gradient	logical value indicating whether exponentiated gradient descent shall be applied to update prevalence parameter. Default is TRUE. If set as FALSE, initial values (i.e., value of inits arguments in vmrsqe::vmrseqOptimControl, can be set up in the control argument of this function) are used as prevalence parameter for decoding hidden states.
tp	a 'transitProbs-class' object that contains the transition probability distribution used for HMM optimization. Default value is transition probability vmrseq:::tp0 built in the package that was previously trained on mouse brain cells. See manuscript for training procedure and data source.
control	list of miscellaneous parameters used to control optimization of the HMM model. Default is output of vmrseq::vmrseqOptimControl(). Can be changed by tweaking arguments in function vmrseq::vmrseqOptimControl().
verbose	logical value that indicates whether progress messages should be printed to stdout. Defaults value is TRUE.
BPPARAM	a BiocParallelParam object to specify the parallel backend. The default option is BiocParallel::bpparam() which will automatically creates a cluster appropriate for the operating system.

Value

The results object is a list of 6 elements that contains the following information: 1. 'gr': The 'Granges' object that has been input to 'vmrseqFit' with two added metadata columns: + 'cr_index' = Index in reference to rows of 'cr.ranges', denoting row number of the candidate region to which the CpG site belongs. + 'vmr_index' = Index in reference to rows of 'vmr.ranges', denoting row number of the variably methylated region to which the CpG site belongs. 2. 'vmr.ranges': A 'Granges' object with the coordinates of each detected variably methylated region (each row is a VMR), with metadata columns: + 'num_cpg' = Number of observed CpG sites in the VMR. +

‘start_ind‘ = Index of the starting CpG sites in reference to rows of ‘gr‘. + ‘end_ind‘ = Index of the ending CpG sites in reference to rows of ‘gr‘. + ‘pi‘ = Prevalence of the methylated grouping (see manuscript for details) + ‘loglik_diff‘ = Difference in log-likelihood of two-grouping and one-grouping HMM fitted to the VMR; can be used to rank the VMRs. 3. ‘cr.ranges‘: A ‘Granges‘ object with the coordinates of each candidate region (each row is a candidate region), with metadata column: + ‘num_cpg‘ = Number of observed CpG sites in the candidate region. 4. ‘alpha‘: Designated significance level (default 0.05, can be changed by user with function argument). It is used for determining the threshold on variance used for constructing candidate. The threshold is computed by taking the (1-alpha) quantile of an approximate null distribution of variance (see manuscript for details). 5. ‘var_cutoff‘: Variance cutoff computed from ‘alpha‘. 6. ‘bb_params‘: Beta-binomial parameter used in emission probability of the HMM model; they are determined by the magnitude of the input dataset (see manuscript for details).

Examples

```
# load example data
data(toy.gr)
# run vmrseqFit
toy.results <- vmrseqFit(toy.gr)
toy.results
```

---

vmrseqOptimControl *Auxiliary function as user interface for vmrseq optimization.*

Description

Typically only used when calling vmrseq function with the option `control`.

Usage

```
vmrseqOptimControl(
  inits = c(0.2, 0.5, 0.8),
  epsilon = 0.001,
  backtrack = TRUE,
  eta = ifelse(backtrack, 0.05, 0.005),
  maxIter = 100
)
```

Arguments

`inits`	vector of numeric values between 0 and 1 representing initial values of pi_1 shall be taken in optimization algorithm.
`epsilon`	numeric value representing the convergence upper bound for the algorithm.
`backtrack`	logical value indicating whether to use backtracking line search to automatically adjust learning rate. Default is TRUE.

eta	a numeric value representing the learning rate in optimization. Default is ifelse(backtrack, 0.05, 0.005).
maxIter	positive integer value representing the maximum number of iterations in optimization algorithm.

Value

the list of arguments for optimization control

Examples

```
vmrseqOptimControl()
```

---

vmrseqSmooth *Smoothing on single-cell bisulfite sequencing data for the purpose of constructing candidate regions.*

Description

vmrseqSmooth takes a SummarizedExperiment object with information of methylation level of individual cells as input, and perform a kernel smoother to 'relative' methylation levels of individual cells prior to constructing candidate regions. Purpose of the smoothing is to adjust for uneven coverage biases and borrow information from nearby sites. See manuscript for detailed description.

Usage

```
vmrseqSmooth(
  SE,
  bpWindow = 2000,
  sparseNAdrop = is_sparse(assays(SE)[[1]]),
  verbose = TRUE,
  BPPARAM = bpparam()
)
```

Arguments

SE	SummarizedExperiment object with one (and only one) assay that contains *binary* methylation status of CpG sites in individual cells. We recommend using output by poolData (i.e., an NA-dropped HDF5-based SummarizedExperiment object) to prevent running out of memory.
bpWindow	positive integer that represents the width (in bp) of smoothing window. Default value is 2000.
sparseNAdrop	logical value that represents whether the NA values are droppped in the input SE object. SE objects output by poolData are NA dropped. See ?vmrseq::poolData for details about NA-dropped representation.

verbose — logical value that indicates whether progress messages should be printed to stdout. Defaults value is TRUE.

BPPARAM — a BiocParallelParam object to specify the parallel backend. The default option is BiocParallel::bpparam() which will automatically creates a cluster appropriate for the operating system.

Value

a GRanges object that contains the result of smoothing. The object retains genomic coordinates (chr, start, end) of input CpG sites, in the same order as in the input SE object. Three column are added (on top of original metadata columns for the CpG sites in SE, if any): 1. meth: methylated cell count of the CpG 2. total: total (non-missing) cell count of the CpG 3. var: variance computed based on individual-cell smoothed relative methylation levels.

See Also

poolData, vmrseqFit

Examples

```
# load example data
toy.se <- HDF5Array::loadHDF5SummarizedExperiment(system.file("extdata", "toy", package = "vmrseq"))

# preprocessing
total <- DelayedArray::rowSums(SummarizedExperiment::assays(toy.se)$M_mat > 0)
toy.se <- subset(toy.se, total >= 3)

# run vmrseqSmooth
toy.gr <- vmrseqSmooth(toy.se)
toy.gr
```

Index