Package ‘MethScope’

December 9, 2025

Title Ultra-Fast Analysis of Sparse DNA Methylome via Recurrent Pattern Encoding

Version 1.0.0

Description Methods for analyzing DNA methylation data via Most Recurrent Methylation Patterns (MRMPs). Supports cell-type annotation, spatial deconvolution, unsupervised clustering, and cancer cell-of-origin inference. Includes C-backed summaries for YAME “.cg/.cm” files (overlap counts, log2 odds ratios, beta/depth aggregation), an XGBoost classifier, NNLS deconvolution, and plotting utilities. Scales to large spatial and single-cell methylomes and is robust to extreme sparsity.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports xgboost, dplyr, utils, tidyr, stringr, caret, doParallel, parallel, ggplot2, uwot, magrittr, FNN, data.table, nnls

Suggests knitr, rmarkdown, spelling

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation yes

Maintainer Hongxiang Fu <fhx@seas.upenn.edu>

Language en-US

Author Hongxiang Fu [aut, cre] (ORCID:
<https://orcid.org/0000-0002-9873-8606>),
Wanding Zhou [cph, fnd],
The SAMtools/HTSlib authors [ctb, cph] (BGZF components; see inst/COPYRIGHTS),
Attractive Chaos [ctb, cph] (Author and copyright holder of khash.h (klib, MIT license))

Repository CRAN

Date/Publication 2025-12-09 16:00:07 UTC

Contents

`confidence_score` *Produce confidence score for XGBoost prediction*

Description

Produce confidence score for XGBoost prediction

Usage

```
confidence_score(vec)
```

Arguments

`vec`	A vector of predicted probability for each cell type

Value

A numeric confidence score from 0 to 1.

`confidence_score_top95` *Produce confidence score based on top 95 percent for XGBoost prediction*

Description

Produce confidence score based on top 95 percent for XGBoost prediction

Usage

```
confidence_score_top95(vec)
```

Arguments

`vec`	A vector of predicted probability for each cell type

Value

A numeric confidence score from 0 to 1.

`filter_cell` *Filter final prediction to reduce noise*

Description

Filter final prediction to reduce noise

Usage

```
filter_cell(pred_result, knn_res, KNeighbor = 5)
```

Arguments

`pred_result`	The prediction result from XGBoost
`knn_res`	knn graph from smooth_matrix
`KNeighbor`	Number of knn neighbors to use for smoothing (Default: 5)

Value

The final prediction result after dropping few cell types

GenerateInput *Generate pattern level data for cell type annotation*

Description

Generate pattern level data for cell type annotation

Usage

```
GenerateInput(query_fn, knowledge_fn)
```

Arguments

query_fn	File path to query .cg
knowledge_fn	File path to pattern file .cm

Value

A cell by pattern matrix.

Examples

```
qry <- system.file("extdata", "toy.cg", package = "MethScope")
msk <- system.file("extdata", "toy.cm", package = "MethScope")
res <- GenerateInput(qry, msk)
```

GenerateReference *Generate reference pattern labels (no default writing)*

Description

Generate reference pattern labels (no default writing)

Usage

```
GenerateReference(binary_file, min_CG = 50, output_path = NULL)
```

Arguments

binary_file	Path to the pattern strings file (one string per line).
min_CG	Minimum CpG count a pattern must have to keep its own ID (default: 50). Patterns with frequency <= min_CG are grouped as "Pna".
output_path	Optional file path to write the resulting labels. If NULL (default), nothing is written and the labels are only returned.

Value

A character vector of pattern labels (same length/order as the input file).

Examples

```
## Not run:
# DO write only to a temp location in examples/vignettes/tests:
tmp_out <- file.path(tempdir(), "patterns.txt")
labs <- GenerateReference("path/to/pattern_strings.txt", min_CG = 50, output_path = tmp_out)
# Or skip writing and just get the vector:
labs <- GenerateReference("path/to/pattern_strings.txt", min_CG = 50)

## End(Not run)
```

imputeRowMean *Impute missing value for 100K window matrix*

Description

Impute missing value for 100K window matrix

Usage

```
imputeRowMean(mtx, na_percent = 30)
```

Arguments

mtx	A cell by 100K window data frame with missing values
na_percent	A na percent threshold to be filterd (Default: 30)

Value

A cell by 100K window data frame with imputed values

Input_training *Train XGBoost model to predict cell type*

Description

Train XGBoost model to predict cell type

Usage

```
Input_training(
  summary_results,
  cell_type_label,
  number_patterns = 1000,
  cross_validation = FALSE,
  xgb_parameters = list()
)
```

Arguments

`summary_results`
a wide cell by pattern matrix generated from GenerateInput function

`cell_type_label`
a vector of the corresponding cell type label for each row of the summary results

`number_patterns`
a numeric value to indicate number of patterns to be used (Default: 1000)

`cross_validation`
a boolean varaible whether to perform cross_validation to obtain the best hyper parameters for the model

`xgb_parameters` an optional list for xgb model parameters provided by the user

Value

the xgb model trained

`nnls_deconv` *Estimate cell type relative proportion*

Description

Estimate cell type relative proportion

Usage

```
nnls_deconv(ref, mixture_matrix, number_patterns = 1000, var_threshold = 0.01)
```

Arguments

`ref` An imputed wide cell by pattern matrix generated from GenerateInput function using reference Pseudobulk

`mixture_matrix` An imputed wide cell by pattern matrix generated from GenerateInput function

`number_patterns`
a numeric value to indicate number of patterns to be used (Default: 1000)

`var_threshold` a numeric value to indicate variance that should filter the patterns (Default: 0.1)

Value

A cell type by cell matrix showing the relative cell type proportion estimate for each cells

---

PlotConfusion *Generate confusion table for the final prediction*

Description

Generate confusion table for the final prediction

Usage

```
PlotConfusion(prediction_result, actual_label, log2 = FALSE)
```

Arguments

`prediction_result`	Prediction result from PredictCellType
`actual_label`	Ground truth cell label
`log2`	Log scale count (Default: False)

Value

A ggplot2 confusion table object.

PlotF1 *Generate F1 score barplot for each class*

Description

Generate F1 score barplot for each class

Usage

```
PlotF1(prediction_result, actual_label)
```

Arguments

`prediction_result`	Prediction result from PredictCellType
`actual_label`	Ground truth cell label

Value

A ggplot2 object.

PlotUMAP — *Generate UMAP for the final prediction based on cell patterns*

Description

Generate UMAP for the final prediction based on cell patterns

Usage

```
PlotUMAP(predictMatrix, prediction_result, n_component = 30, seed = 123, ...)
```

Arguments

`predictMatrix`	a wide cell by pattern matrix generated from GenerateInput function
`prediction_result`	Prediction result from PredictCellType
`n_component`	Number of PCA components to use (Default: 30)
`seed`	A number for random seed (Default: 123)
`...`	Additional arguments passed to `uwot::umap` (e.g., `n_neighbors`, `metric`).

Value

A list of two ggplot2 UMAP object.

PlotUMAP_fixedwindow — *Generate UMAP for the final prediction based on fixed window eg.100kb bin widows*

Description

Generate UMAP for the final prediction based on fixed window eg.100kb bin widows

Usage

```
PlotUMAP_fixedwindow(
  query_fn,
  knowledge_fn,
  prediction_result,
  n_component = 30,
  seed = 123,
  ...
)
```

Arguments

`query_fn` File path to query .cg

`knowledge_fn` File path to 100bk bins window or reference pattern

`prediction_result` Prediction result from PredictCellType

`n_component` Number of PCA components to use (Default: 30)

`seed` A number for random seed (Default: 123)

`...` Additional arguments passed to `uwot::umap` (e.g., `n_neighbors`, `metric`).

Value

A list of two ggplot2 UMAP object.

`PredictCellType` *Predict cell type annotation from the trained model*

Description

Predict cell type annotation from the trained model

Usage

```
PredictCellType(bst_model, predictMatrix, smooth = FALSE, KNeighbor = 5)
```

Arguments

`bst_model` The boosting model trained from ModelTrain

`predictMatrix` A wide cell by pattern matrix generated from GenerateInput function

`smooth` A Boolean variable to indicate whether smooth the matrix (Default: FALSE)

`KNeighbor` number of knn neighbors to use for smoothing (Default: 5)

Value

A cell by cell type matrix with confidence score and labeled cell type.

Examples

```
qry <- system.file("extdata", "toy.cg", package = "MethScope")
msk <- system.file("extdata", "toy.cm", package = "MethScope")
res <- GenerateInput(qry, msk)
## Not run:
prediction <- PredictCellType(Liu2021_MouseBrain_P1000,res)

## End(Not run)
```

smooth_matrix *Smooth cell by pattern matrix to reduce noise*

Description

Smooth cell by pattern matrix to reduce noise

Usage

```
smooth_matrix(predictMatrix, KNeighbor = 5)
```

Arguments

`predictMatrix`	A wide cell by pattern matrix generated from GenerateInput function
`KNeighbor`	Number of knn neighbors to use for smoothing (Default: 5)

Value

A wide cell by pattern matrix after smoothing and knn graph

Index