

# Package ‘Melissa’

December 16, 2025

**Type** Package

**Title** Bayesian clustering and imputation of single cell methylomes

**Version** 1.26.0

**Description** Melissa is a Bayesian probabilistic model for jointly clustering and imputing single cell methylomes. This is done by taking into account local correlations via a Generalised Linear Model approach and global similarities using a mixture modelling approach.

**Depends** R (>= 3.5.0), BPRMeth, GenomicRanges

**License** GPL-3 | file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Imports** data.table, parallel, ROCR, matrixcalc, mclust, ggplot2, doParallel, foreach, MCMCpack, cowplot, magrittr, mvtnorm, truncnorm, assertthat, BiocStyle, stats, utils

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**biocViews** ImmunoOncology, DNAMethylation, GeneExpression, GeneRegulation, Epigenetics, Genetics, Clustering, FeatureExtraction, Regression, RNASeq, Bayesian, KEGG, Sequencing, Coverage, SingleCell

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

**git\_branch** RELEASE\_3\_22

**git\_last\_commit** edad671

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

**Repository** Bioconductor 3.22

**Date/Publication** 2025-12-15

**Author** C. A. Kapourani [aut, cre]

**Maintainer** C. A. Kapourani <kapouranis.andreas@gmail.com>

## Contents

|                                       |           |
|---------------------------------------|-----------|
| binarise_files . . . . .              | 2         |
| cluster_ari . . . . .                 | 3         |
| cluster_error . . . . .               | 4         |
| create_melissa_data_obj . . . . .     | 4         |
| eval_cluster_performance . . . . .    | 6         |
| eval_imputation_performance . . . . . | 7         |
| extract_y . . . . .                   | 8         |
| filter_regions . . . . .              | 9         |
| impute_met_files . . . . .            | 10        |
| impute_test_met . . . . .             | 11        |
| init_design_matrix . . . . .          | 12        |
| log_sum_exp . . . . .                 | 13        |
| Melissa . . . . .                     | 14        |
| melissa . . . . .                     | 14        |
| melissa_encode_dt . . . . .           | 16        |
| melissa_gibbs . . . . .               | 17        |
| melissa_synth_dt . . . . .            | 19        |
| partition_dataset . . . . .           | 19        |
| plot_melissa_profiles . . . . .       | 20        |
| <b>Index</b>                          | <b>22</b> |

---

|                |                           |
|----------------|---------------------------|
| binarise_files | <i>Binarise CpG sites</i> |
|----------------|---------------------------|

---

## Description

Script for binarising CpG sites and formatting the coverage file so it can be directly used from the BPRMeth package. The format of each file is the following: <chr> <start> <met\_level>, where met\_level can be either 0 or 1. To read compressed files, e.g ending in .gz or .bz2, the R.utils package needs to be installed.

## Usage

```
binarise_files(indir,outdir = NULL, format = 1, no_cores = NULL)
```

## Arguments

|          |                                                                                                                                                                                                                                                                               |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| indir    | Directory containing the coverage files, output from Bismark.                                                                                                                                                                                                                 |
| outdir   | Directory to store the output files for each cell with exactly the same name. If NULL, then a directory called 'binarised' inside 'indir' will be create by default.                                                                                                          |
| format   | Integer, denoting the format of coverage file. When set to '1', the coverage file format is assumed to be: "<chr> <start> <end> <met_prg> <met_reads> <unmet_reads>". When set to '2', then the format is assumed to be: "<chr> <start> <met_prg> <met_reads> <unmet_reads>". |
| no_cores | Number of cores to use for parallel processing. If NULL, no parallel processing is used.                                                                                                                                                                                      |

**Value**

No value is returned, the binarised data are stored in the outdir.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [filter\\_regions](#)

**Examples**

```
## Not run:  
# Met directory  
met_dir <- "name_of_met_dir"  
  
binarise_files(met_dir)  
  
## End(Not run)
```

---

cluster\_ari

*Compute clustering ARI*

---

**Description**

cluster\_ari computes the clustering performance in terms of the Adjusted Rand Index (ARI) metric.

**Usage**

```
cluster_ari(C_true, C_post)
```

**Arguments**

|        |                                                              |
|--------|--------------------------------------------------------------|
| C_true | True cluster assignemnts.                                    |
| C_post | Posterior responsibilities of predicted cluster assignemnts. |

**Value**

The clustering ARI.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

---

|               |                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cluster_error | <i>Compute clustering assignment error cluster_error computes the clustering assignment error; i.e. the average number of incorrect cluster assignments:</i> |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

$$OE = \sum_{n=1}^N I(LT_n \neq LP_n) / N$$


---

### Description

Compute clustering assignment error

cluster\_error computes the clustering assignment error, i.e. the average number of incorrect cluster assignments:

$$OE = \sum_{n=1}^N (I(LT_n \neq LP_n)) / N$$

### Usage

```
cluster_error(C_true, C_post)
```

### Arguments

|        |                                                  |
|--------|--------------------------------------------------|
| C_true | True cluster assignemnts.                        |
| C_post | Posterior mean of predicted cluster assignemnts. |

### Value

The clustering assignment error

### Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

---

|                         |                                                 |
|-------------------------|-------------------------------------------------|
| create_melissa_data_obj | <i>Create methylation regions for all cells</i> |
|-------------------------|-------------------------------------------------|

---

### Description

Wrapper function for creating methylation regions for all cells, which is the input object for Melissa prior to filtering.

**Usage**

```
create_melissa_data_obj(
  met_dir,
  anno_file,
  chrom_size_file = NULL,
  chr_discarded = NULL,
  is_centre = FALSE,
  is_window = TRUE,
  upstream = -5000,
  downstream = 5000,
  cov = 5,
  sd_thresh = -1,
  no_cores = NULL
)
```

**Arguments**

|                 |                                                                                                                                                                                                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| met_dir         | Directory of (binarised) methylation files, each file corresponds to a single cell.                                                                                                                                                                                                                |
| anno_file       | The annotation file with 'tab' delimited format: "chromosome", "start", "end", "strand", "id", "name" (optional). Read the 'BPRMeth' documentation for more details.                                                                                                                               |
| chrom_size_file | Optional file name to read genome chromosome sizes.                                                                                                                                                                                                                                                |
| chr_discarded   | Optional vector with chromosomes to be discarded.                                                                                                                                                                                                                                                  |
| is_centre       | Logical, whether 'start' and 'end' locations are pre-centred. If TRUE, the mean of the locations will be chosen as centre. If FALSE, the 'start' will be chosen as the center; e.g. for genes the 'start' denotes the TSS and we use this as centre to obtain K-bp upstream and downstream of TSS. |
| is_window       | Whether to consider a predefined window region around centre. If TRUE, then 'upstream' and 'downstream' parameters are used, otherwise we consider the whole region from start to end location.                                                                                                    |
| upstream        | Integer defining the length of bp upstream of 'centre' for creating the genomic region. If is_window = FALSE, this parameter is ignored.                                                                                                                                                           |
| downstream      | Integer defining the length of bp downstream of 'centre' for creating the genomic region. If is_window = FALSE, this parameter is ignored.                                                                                                                                                         |
| cov             | Integer defining the minimum coverage of CpGs that each region must contain.                                                                                                                                                                                                                       |
| sd_thresh       | Optional numeric defining the minimum standard deviation of the methylation change in a region. This is used to filter regions with no methylation variability.                                                                                                                                    |
| no_cores        | Number of cores to be used for parallel processing of data.                                                                                                                                                                                                                                        |

**Value**

A melissa\_data\_obj object, with the following elements:

- met: A list of elements of length N, where N are the total number of cells. Each element in the list contains another list of length M, where M is the total number of genomic regions, e.g. promoters. Each element in the inner list is an I x 2 matrix, where I are the total number of observations. The first column contains the input observations x (i.e. CpG locations) and the 2nd column contains the corresponding methylation level.
- anno\_region: The annotation object.
- opts: A list with the parameters that were used for creating the object.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[binarise\\_files](#), [melissa](#), [filter\\_regions](#)

**Examples**

```
## Not run:
# Met directory
met_dir <- "name_of_met_dir"
# Annotation file name
anno_file <- "name_of_anno_file"

obj <- create_melissa_data_obj(met_dir, anno_file)

# Extract annotation regions
met <- obj$met

# Extract annotation regions
anno <- obj$anno_region

## End(Not run)
```

---

eval\_cluster\_performance

*Evaluate clustering performance*

---

**Description**

eval\_cluster\_performance is a wrapper function for computing clustering performance in terms of ARI and clustering assignment error.

**Usage**

```
eval_cluster_performance(obj, C_true)
```

**Arguments**

|        |                                     |
|--------|-------------------------------------|
| obj    | Output of Melissa inference object. |
| C_true | True cluster assignemnts.           |

**Value**

The ‘melissa’ object, with an additional slot named ‘clustering’, containing the ARI and clustering assignment error performance.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [filter\\_regions](#), [eval\\_imputation\\_performance](#), [eval\\_cluster\\_performance](#)

**Examples**

```
## Extract synthetic data
dt <- melissa_synth_dt

# Partition to train and test set
dt <- partition_dataset(dt)

# Create basis object from BPRMeth package
basis_obj <- BPRMeth::create_rbf_object(M = 3)

# Run Melissa
melissa_obj <- melissa(X = dt$met, K = 2, basis = basis_obj, vb_max_iter = 10,
  vb_init_nstart = 1, is_parallel = FALSE, is_verbose = FALSE)

# Compute cluster performance
melissa_obj <- eval_cluster_performance(melissa_obj, dt$opts$C_true)

cat("ARI: ", melissa_obj$clustering$ari)
```

---

eval\_imputation\_performance

*Evaluate imputation performance*

---

**Description**

eval\_imputation\_performance is a wrapper function for computing imputation/clustering performance in terms of different metrics, such as AUC and precision recall curves.

**Usage**

```
eval_imputation_performance(obj, imputation_obj)
```

**Arguments**

|                |                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------|
| obj            | Output of Melissa inference object.                                                                                             |
| imputation_obj | List containing two vectors of equal length, corresponding to true methylation states and predicted/imputed methylation states. |

**Value**

The ‘melissa’ object, with an additional slot named ‘imputation’, containing the AUC, F-measure, True Positive Rate (TPR) and False Positive Rate (FPR), and Precision Recall (PR) curves.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [impute\\_test\\_met](#), [filter\\_regions](#), [eval\\_imputation\\_performance](#), [eval\\_cluster\\_performance](#)

**Examples**

```
# First take a subset of cells to efficiency
# Extract synthetic data
dt <- melissa_synth_dt

# Partition to train and test set
dt <- partition_dataset(dt)

# Create basis object from BPRMeth package
basis_obj <- BPRMeth::create_rbf_object(M = 3)

# Run Melissa
melissa_obj <- melissa(X = dt$met, K = 2, basis = basis_obj, vb_max_iter = 10,
  vb_init_nstart = 1, is_parallel = FALSE, is_verbose = FALSE)

imputation_obj <- impute_test_met(obj = melissa_obj, test = dt$met_test)

melissa_obj <- eval_imputation_performance(obj = melissa_obj,
  imputation_obj = imputation_obj)

cat("AUC: ", melissa_obj$imputation$auc)
```

---

extract\_y

*Extract responses y*

---

**Description**

Given a list of observations, extract responses y

**Usage**

```
extract_y(X, coverage_ind)
```

**Arguments**

|              |                                  |
|--------------|----------------------------------|
| X            | Observations                     |
| coverage_ind | Which observations have coverage |

**Value**

The design matrix H



```
# Run on synthetic data from Melissa package
filt_obj <- filter_by_variability(melissa_encode_dt, min_var = 0.1)
```

---

|                  |                                         |
|------------------|-----------------------------------------|
| impute_met_files | <i>Impute/predict methylation files</i> |
|------------------|-----------------------------------------|

---

## Description

Make predictions of missing methylation states, i.e. perform imputation using Melissa. Each file in the directory will be used as input and a new file will be created in `outdir` with an additional column containing the predicted met state (value between 0 and 1). Note that predictions will be made only on annotation regions that were used for training Melissa. Check [impute\\_test\\_met](#), if you want to make predictions only on test data.

## Usage

```
impute_met_files(
  met_dir,
  outdir = NULL,
  obj,
  anno_region,
  basis = NULL,
  is_predictive = TRUE,
  no_cores = NULL
)
```

## Arguments

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>met_dir</code>       | Directory of methylation files, each file corresponds to a single cell. It should contain three columns <code>&lt;chr&gt;</code> <code>&lt;pos&gt;</code> <code>&lt;met_state&gt;</code> (similar to the input required by <a href="#">create_melissa_data_obj</a> ), where <code>met_state</code> can be any value that denotes missing CpG information, e.g. -1. Note that files can contain also CpGs for which we have coverage information, and we can check the predictions made by Melissa, hence the value can also be 0 (unmet) or 1 (met). Predictions made by Melissa, will not change the <code>&lt;met_state&gt;</code> column. Melissa will just add an additional column named <code>&lt;predicted&gt;</code> . |
| <code>outdir</code>        | Directory to store the output files for each cell with exactly the same name. If NULL, then a directory called 'imputed' inside 'met_dir' will be created by default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>obj</code>           | Output of Melissa inference object.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>anno_region</code>   | Annotation region object. This will be the output of <a href="#">create_melissa_data_obj</a> function, e.g. <code>melissa_data\$anno_region</code> . This is required to select those regions that were used to train Melissa.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>basis</code>         | Basis object, if NULL we perform imputation using Melissa, otherwise using BPRMeth (then <code>obj</code> should be BPRMeth output).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>is_predictive</code> | Logical, use predictive distribution for imputation, or choose the cluster label with the highest responsibility.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>no_cores</code>      | Number of cores to be used for parallel processing of data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Value**

A new directory `outdir` containing files (cells) with predicted / imputed methylation states per CpG location.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [filter\\_regions](#)

**Examples**

```
## Not run:
# Met directory
met_dir <- "name_of_met_dir"
# Annotation file name
anno_file <- "name_of_anno_file"
# Create data object
melissa_data <- create_melissa_data_obj(met_dir, anno_file)
# Run Melissa
melissa_obj <- melissa(X = melissa_data$met, K = 2)
# Annotation object
anno_region <- melissa_data$anno_region

# Perform imputation
impute_met_dir <- "name_of_met_dir_for_imputing_cells"
out <- impute_met_files(met_dir = impute_met_dir, obj = melissa_obj,
                       anno_region = anno_region)

## End(Not run)
```

---

impute\_test\_met

---

*Impute/predict test methylation states*


---

**Description**

Make predictions of missing methylation states, i.e. perform imputation using Melissa. This requires keeping a subset of data as a held out test set during Melissa inference. If you want to impute a whole directory containing cells (files) with missing methylation levels, see [impute\\_met\\_files](#).

**Usage**

```
impute_test_met(
  obj,
  test,
  basis = NULL,
  is_predictive = TRUE,
  return_test = FALSE
)
```

**Arguments**

|               |                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| obj           | Output of Melissa inference object.                                                                               |
| test          | Test data to evaluate performance.                                                                                |
| basis         | Basis object, if NULL we perform imputation using Melissa, otherwise using BPRMeth.                               |
| is_predictive | Logical, use predictive distribution for imputation, or choose the cluster label with the highest responsibility. |
| return_test   | Whether or not to return a list with the predictions.                                                             |

**Value**

A list containing two vectors, the true methylation state and the predicted/imputed methylation states.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [filter\\_regions](#), [eval\\_imputation\\_performance](#), [eval\\_cluster\\_performance](#)

**Examples**

```
# Extract synthetic data
dt <- melissa_synth_dt

# Partition to train and test set
dt <- partition_dataset(dt)

# Create basis object from BPRMeth package
basis_obj <- BPRMeth::create_rbf_object(M = 3)

# Run Melissa
melissa_obj <- melissa(X = dt$met, K = 2, basis = basis_obj, vb_max_iter=10,
  vb_init_nstart = 1, is_parallel = FALSE, is_verbose = FALSE)

imputation_obj <- impute_test_met(obj = melissa_obj,
  test = dt$met_test)
```

---

|                    |                                   |
|--------------------|-----------------------------------|
| init_design_matrix | <i>Initialise design matrices</i> |
|--------------------|-----------------------------------|

---

**Description**

Given a list of observations, initialise design matrices H for computational efficiency.

**Usage**

```
init_design_matrix(basis, X, coverage_ind)
```

**Arguments**

|              |                                  |
|--------------|----------------------------------|
| basis        | Basis object.                    |
| X            | Observations                     |
| coverage_ind | Which observations have coverage |

**Value**

The design matrix H

---

|             |                                   |
|-------------|-----------------------------------|
| log_sum_exp | <i>Compute stable log-sum-exp</i> |
|-------------|-----------------------------------|

---

**Description**

log\_sum\_exp computes the log sum exp trick for avoiding numeric underflow and have numeric stability in computations of small numbers.

**Usage**

```
log_sum_exp(x)
```

**Arguments**

|   |                          |
|---|--------------------------|
| x | A vector of observations |
|---|--------------------------|

**Value**

The logs-sum-exp value

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**References**

<https://hips.seas.harvard.edu/blog/2013/01/09/computing-log-sum-exp/>

---

|         |                                                                              |
|---------|------------------------------------------------------------------------------|
| Melissa | Melissa: <i>Bayesian clustering and imputation of single cell methylomes</i> |
|---------|------------------------------------------------------------------------------|

---

**Description**

Bayesian clustering and imputation of single cell methylomes

**Usage**

```
.datatable.aware
```

**Format**

An object of class `logical` of length 1.

**Value**

Melissa main package documentation.

**Author(s)**

C.A.Kapourani <kapouranis.andreas@gmail.com>

**See Also**

[melissa](#), [create\\_melissa\\_data\\_obj](#), [partition\\_dataset](#), [plot\\_melissa\\_profiles](#), [filter\\_regions](#)

---

|         |                                                           |
|---------|-----------------------------------------------------------|
| melissa | <i>Cluster and impute single cell methylomes using VB</i> |
|---------|-----------------------------------------------------------|

---

**Description**

`melissa` clusters and imputes single cells based on their methylome landscape on specific genomic regions, e.g. promoters, using the Variational Bayes (VB) EM-like algorithm.

**Usage**

```
melissa(
  X,
  K = 3,
  basis = NULL,
  delta_0 = NULL,
  w = NULL,
  alpha_0 = 0.5,
  beta_0 = NULL,
  vb_max_iter = 300,
  epsilon_conv = 1e-05,
  is_kmeans = TRUE,
  vb_init_nstart = 10,
```

```

    vb_init_max_iter = 20,
    is_parallel = FALSE,
    no_cores = 3,
    is_verbose = TRUE
)

```

## Arguments

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>                | The input data, which has to be a <a href="#">list</a> of elements of length $N$ , where $N$ are the total number of cells. Each element in the list contains another list of length $M$ , where $M$ is the total number of genomic regions, e.g. promoters. Each element in the inner list is an $I \times 2$ matrix, where $I$ are the total number of observations. The first column contains the input observations $x$ (i.e. CpG locations) and the 2nd columns contains the corresponding methylation level. |
| <code>K</code>                | Integer denoting the total number of clusters $K$ .                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>basis</code>            | A 'basis' object. E.g. see <code>create_basis</code> function from BPRMeth package. If NULL, will an RBF object with 3 basis functions will be created.                                                                                                                                                                                                                                                                                                                                                            |
| <code>delta_0</code>          | Parameter vector of the Dirichlet prior on the mixing proportions $\pi$ .                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>w</code>                | Optional, an $M \times D \times K$ array of the initial parameters, where first dimension are the genomic regions $M$ , 2nd the number of covariates $D$ (i.e. basis functions), and 3rd are the clusters $K$ . If NULL, will be assigned with default values.                                                                                                                                                                                                                                                     |
| <code>alpha_0</code>          | Hyperparameter: shape parameter for Gamma distribution. A Gamma distribution is used as prior for the precision parameter $\tau$ .                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>beta_0</code>           | Hyperparameter: rate parameter for Gamma distribution. A Gamma distribution is used as prior for the precision parameter $\tau$ .                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>vb_max_iter</code>      | Integer denoting the maximum number of VB iterations.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>epsilon_conv</code>     | Numeric denoting the convergence threshold for VB.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>is_kmeans</code>        | Logical, use Kmeans for initialization of model parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>vb_init_nstart</code>   | Number of VB random starts for finding better initialization.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>vb_init_max_iter</code> | Maximum number of mini-VB iterations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>is_parallel</code>      | Logical, indicating if code should be run in parallel.                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>no_cores</code>         | Number of cores to be used, default is <code>max_no_cores - 1</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>is_verbose</code>       | Logical, print results during VB iterations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## Value

An object of class `melissa` with the following elements:

- `W`: An  $(M+1) \times K$  matrix with the optimized parameter values for each cluster,  $M$  are the number of basis functions. Each column of the matrix corresponds a different cluster  $k$ .
- `W_Sigma`: A list with the covariance matrices of the posterior parameter  $W$  for each cluster  $k$ .
- `r_nk`: An  $(N \times K)$  responsibility matrix of each observations being explained by a specific cluster.
- `delta`: Optimized Dirichlet parameter for the mixing proportions.
- `alpha`: Optimized shape parameter of Gamma distribution.
- `beta`: Optimized rate parameter of the Gamma distribution

- basis: The basis object.
- lb: The lower bound vector.
- labels: Cluster assignment labels.
- pi\_k: Expected value of mixing proportions.

### Details

The modelling and mathematical details for clustering profiles using mean-field variational inference are explained here: <http://rpubs.com/cakapourani/>. More specifically:

- For Binomial/Bernoulli observation model check: <http://rpubs.com/cakapourani/vb-mixture-bpr>
- For Gaussian observation model check: <http://rpubs.com/cakapourani/vb-mixture-lr>

### Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

### See Also

[create\\_melissa\\_data\\_obj](#), [partition\\_dataset](#), [plot\\_melissa\\_profiles](#), [impute\\_test\\_met](#), [impute\\_met\\_files](#), [filter\\_regions](#)

### Examples

```
# Example of running Melissa on synthetic data

# Create RBF basis object with 4 RBFs
basis_obj <- BPRMeth::create_rbf_object(M = 4)

set.seed(15)
# Run Melissa
melissa_obj <- melissa(X = melissa_synth_dt$met, K = 2, basis = basis_obj,
  vb_max_iter = 10, vb_init_nstart = 1, vb_init_max_iter = 5,
  is_parallel = FALSE, is_verbose = FALSE)

# Extract mixing proportions
print(melissa_obj$pi_k)
```

---

melissa\_encode\_dt

*Synthetic ENCODE single cell methylation data*

---

### Description

Small synthetic ENCODE data generated by inferring methylation profiles from bulk ENCODE data, and subsequently generating single cells. It consists of  $N = 200$  cells and  $M = 100$  genomic regions. The data are in the required format for directly running Melissa and are used as a case study for the vignette.

### Usage

melissa\_encode\_dt

**Format**

A list object containing methylation regions, annotation data and the options used for creating the data. This in general would be the output of the [create\\_melissa\\_data\\_obj](#) function. It has the following three objects:

- met: A list containing the methylation data, each element of the list is a different cell.
- anno\_region: Corresponding annotation data for each genomic region.
- opts: Parameters/options used to generate the data.

**Value**

Synthetic ENCODE methylation data

**See Also**

[create\\_melissa\\_data\\_obj](#)

---

melissa\_gibbs

*Gibbs sampling algorithm for Melissa model*

---

**Description**

melissa\_gibbs implements the Gibbs sampling algorithm for performing clustering of single cells based on their DNA methylation profiles, where the observation model is the Bernoulli distributed Probit Regression likelihood. NOTE: that Gibbs sampling is really slow and we recommend using the VB implementation: [melissa](#).

**Usage**

```
melissa_gibbs(  
  X,  
  K = 2,  
  pi_k = rep(1/K, K),  
  w = NULL,  
  basis = NULL,  
  w_0_mean = NULL,  
  w_0_cov = NULL,  
  dir_a = rep(1, K),  
  lambda = 1/2,  
  gibbs_nsim = 1000,  
  gibbs_burn_in = 200,  
  inner_gibbs = FALSE,  
  gibbs_inner_nsim = 50,  
  is_parallel = TRUE,  
  no_cores = NULL,  
  is_verbose = FALSE  
)
```

**Arguments**

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>X</code>                | A list of length <code>I</code> , where <code>I</code> are the total number of cells. Each element of the list contains another list of length <code>N</code> , where <code>N</code> is the total number of genomic regions. Each element of the inner list is an <code>L x 2</code> matrix of observations, where 1st column contains the locations and the 2nd column contains the methylation level of the corresponding CpGs. |
| <code>K</code>                | Integer denoting the number of clusters <code>K</code> .                                                                                                                                                                                                                                                                                                                                                                          |
| <code>pi_k</code>             | Vector of length <code>K</code> , denoting the mixing proportions.                                                                                                                                                                                                                                                                                                                                                                |
| <code>w</code>                | A <code>N x M x K</code> array, where each column contains the basis function coefficients for the corresponding cluster.                                                                                                                                                                                                                                                                                                         |
| <code>basis</code>            | A 'basis' object. E.g. see <code>create_rbf_object</code> from <code>BPRMeth</code> package                                                                                                                                                                                                                                                                                                                                       |
| <code>w_0_mean</code>         | The prior mean hyperparameter for <code>w</code>                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>w_0_cov</code>          | The prior covariance hyperparameter for <code>w</code>                                                                                                                                                                                                                                                                                                                                                                            |
| <code>dir_a</code>            | The Dirichlet concentration parameter, prior over <code>pi_k</code>                                                                                                                                                                                                                                                                                                                                                               |
| <code>lambda</code>           | The complexity penalty coefficient for penalized regression.                                                                                                                                                                                                                                                                                                                                                                      |
| <code>gibbs_nsim</code>       | Argument giving the number of simulations of the Gibbs sampler.                                                                                                                                                                                                                                                                                                                                                                   |
| <code>gibbs_burn_in</code>    | Argument giving the burn in period of the Gibbs sampler.                                                                                                                                                                                                                                                                                                                                                                          |
| <code>inner_gibbs</code>      | Logical, indicating if we should perform Gibbs sampling to sample from the augmented BPR model.                                                                                                                                                                                                                                                                                                                                   |
| <code>gibbs_inner_nsim</code> | Number of inner Gibbs simulations.                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>is_parallel</code>      | Logical, indicating if code should be run in parallel.                                                                                                                                                                                                                                                                                                                                                                            |
| <code>no_cores</code>         | Number of cores to be used, default is <code>max_no_cores - 1</code> .                                                                                                                                                                                                                                                                                                                                                            |
| <code>is_verbose</code>       | Logical, print results during EM iterations                                                                                                                                                                                                                                                                                                                                                                                       |

**Value**

An object of class `melissa_gibbs`.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[melissa](#), [create\\_melissa\\_data\\_obj](#), [partition\\_dataset](#), [filter\\_regions](#)

**Examples**

```
# Example of running Melissa Gibbs on synthetic data

# Create RBF basis object with 4 RBFs
basis_obj <- BPRMeth::create_rbf_object(M = 4)

set.seed(15)
# Run Melissa Gibbs
melissa_obj <- melissa_gibbs(X = melissa_synth_dt$met, K = 2, basis = basis_obj,
  gibbs_nsim = 10, gibbs_burn_in = 5, is_parallel = FALSE, is_verbose = FALSE)
```

```
# Extract mixing proportions
print(melissa_obj$pi_k)
```

---

|                  |                                               |
|------------------|-----------------------------------------------|
| melissa_synth_dt | <i>Synthetic single cell methylation data</i> |
|------------------|-----------------------------------------------|

---

## Description

Small synthetic data for quick analysis. It consists of  $N = 50$  cells and  $M = 50$  genomic regions.

## Usage

```
melissa_synth_dt
```

## Format

A list object containing methylation regions, annotation data and the options used for creating the data. This in general would be the output of the [create\\_melissa\\_data\\_obj](#) function. It has the following three objects:

- met: A list containing the methylation data, each element of the list is a different cell.
- anno\_region: Corresponding annotation data for each genomic region.
- opts: Parameters/options used to generate the data.

## Value

Synthetic methylation data

## See Also

[create\\_melissa\\_data\\_obj](#)

---

|                   |                                                             |
|-------------------|-------------------------------------------------------------|
| partition_dataset | <i>Partition synthetic dataset to training and test set</i> |
|-------------------|-------------------------------------------------------------|

---

## Description

Partition synthetic dataset to training and test set

## Usage

```
partition_dataset(
  dt_obj,
  data_train_prcg = 0.5,
  region_train_prcg = 0.95,
  cpg_train_prcg = 0.5,
  is_synth = FALSE
)
```

**Arguments**

dt\_obj                Melissa data object

data\_train\_prcg      Percentage of genomic regions that will be fully used for training, i.e. across the whole region we will have no CpGs missing.

region\_train\_prcg    Fraction of genomic regions to keep for training set, i.e. some genomic regions will have no coverage at all during training.

cpg\_train\_prcg      Fraction of CpGs in each genomic region to keep for training set.

is\_synth             Logical, whether we have synthetic data or not.

**Value**

The Melissa object with the following changes. The 'met' element will now contain only the 'training' data. An additional element called 'met\_test' will store the data that will be used during testing to evaluate the imputation performance. These data will not be seen from Melissa during inference.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [filter\\_regions](#)

**Examples**

```
# Partition the synthetic data from Melissa package
dt <- partition_dataset(melissa_encode_dt)
```

---

plot\_melissa\_profiles    *Plot predictive methylation profiles*

---

**Description**

This function plots the predictive distribution of the methylation profiles inferred using the Melissa model. Each colour corresponds to a different cluster.

**Usage**

```
plot_melissa_profiles(
  melissa_obj,
  region = 1,
  title = "Melissa profiles",
  x_axis = "genomic region",
  y_axis = "met level",
  x_labels = c("Upstream", "", "Centre", "", "Downstream"),
  ...
)
```

**Arguments**

|             |                                                            |
|-------------|------------------------------------------------------------|
| melissa_obj | Clustered cell subtypes using Melissa inference functions. |
| region      | Genomic region number.                                     |
| title       | Plot title                                                 |
| x_axis      | x axis label                                               |
| y_axis      | x axis label                                               |
| x_labels    | x axis ticks labels                                        |
| ...         | Additional parameters                                      |

**Value**

A ggplot2 object.

**Author(s)**

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

**See Also**

[create\\_melissa\\_data\\_obj](#), [melissa](#), [filter\\_regions](#), [eval\\_imputation\\_performance](#), [eval\\_cluster\\_performance](#)

**Examples**

```
# Extract synthetic data
dt <- melissa_synth_dt

# Create basis object from BPRMeth package
basis_obj <- BPRMeth::create_rbf_object(M = 3)

# Run Melissa
melissa_obj <- melissa(X = dt$met, K = 2, basis = basis_obj, vb_max_iter = 10,
  vb_init_nstart = 1, is_parallel = FALSE, is_verbose = FALSE)

gg <- plot_melissa_profiles(melissa_obj, region = 10)
```

# Index

## \* datasets

- Melissa, [14](#)
  - melissa\_encode\_dt, [16](#)
  - melissa\_synth\_dt, [19](#)
- .datatable.aware (Melissa), [14](#)
- binarise\_files, [2, 6](#)
- cluster\_ari, [3](#)
- cluster\_error, [4](#)
- create\_melissa\_data\_obj, [3, 4, 7–12, 14, 16–21](#)
- eval\_cluster\_performance, [6, 7, 8, 12, 21](#)
- eval\_imputation\_performance, [7, 7, 8, 12, 21](#)
- extract\_y, [8](#)
- filter\_by\_coverage\_across\_cells  
(filter\_regions), [9](#)
- filter\_by\_cpg\_coverage  
(filter\_regions), [9](#)
- filter\_by\_variability (filter\_regions),  
[9](#)
- filter\_cpgs, (filter\_regions), [9](#)
- filter\_regions, [3, 6–8, 9, 11, 12, 14, 16, 18, 20, 21](#)
- impute\_met\_files, [10, 11, 16](#)
- impute\_test\_met, [8, 10, 11, 16](#)
- init\_design\_matrix, [12](#)
- list, [15](#)
- log\_sum\_exp, [13](#)
- Melissa, [14](#)
- melissa, [3, 6–9, 11, 12, 14, 14, 17, 18, 20, 21](#)
- melissa\_cluster, (melissa), [14](#)
- melissa\_encode\_dt, [16](#)
- melissa\_filter (filter\_regions), [9](#)
- melissa\_gibbs, [17](#)
- melissa\_impute, (melissa), [14](#)
- melissa\_synth\_dt, [19](#)
- melissa\_vb (melissa), [14](#)
- partition\_dataset, [14, 16, 18, 19](#)
- plot\_melissa\_profiles, [14, 16, 20](#)