

Package: mdclust (via r-universe)

July 21, 2026

Type Package

Title Maximum Diversity Clustering

Version 0.99.2

Description Standalone implementation of the maximum diversity clustering algorithm (Algorithm 2) from Korsunsky et al. (2019), Nature Methods. Soft spherical k-means with a batch-diversity penalty.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports Rcpp

LinkingTo Rcpp, RcppArmadillo

Config/roxygen2/version 8.0.0

Suggests testthat (>= 3.0.0), knitr, quarto, ggplot2, harmony (>= 2.0.0), patchwork

Config/testthat/edition 3

VignetteBuilder quarto

SystemRequirements quarto

biocViews Software, Clustering, DimensionReduction

URL <https://github.com/Herbermann/mdclust>

BugReports <https://github.com/Herbermann/mdclust/issues>

Repository <https://biocstaging.r-universe.dev>

Date/Publication 2026-07-21 08:45:54 UTC

RemoteUrl <https://github.com/BiocStaging/mdclust>

RemoteRef HEAD

RemoteSha 01fe57f9df769b110891e60e92825735ccee3dc1

Contents

mdclust	2
plot.mdclust	4
print.mdclust	5
summary.mdclust	5
Index	6

mdclust	<i>Maximum Diversity Clustering</i>
---------	-------------------------------------

Description

Soft spherical k-means with a batch-diversity penalty, as described in Korsunsky et al. 2019 (Harmony, Nature Methods), Algorithm 2. Extracted here as a standalone algorithm with no dependency on the harmony package. For more information on hyperparameters, also check the harmony documentation. Default values are chosen in accordance with harmony default options.

Usage

```
mdclust(  
  Z,  
  batch,  
  K,  
  Y_init = "kmeans++",  
  theta = 2,  
  sigma = 0.1,  
  tau = 0,  
  alpha = 1,  
  block_size = 0.05,  
  max_iter = 200L,  
  epsilon = 1e-05,  
  max_iter_kmeans = 10L,  
  epsilon_kmeans = 1e-04,  
  weight_power = 1,  
  verbose = FALSE  
)
```

Arguments

Z	Numeric matrix, d x N. Observations are columns, embedding dimensions are rows (e.g. PCs). Does not need to be pre-normalized; Columns are L2-normalized internally.
batch	Factor, character, or integer vector of length N giving the batch label for each element.
K	Integer. Number of clusters.

<code>Y_init</code>	Initial centroid positions. Either 'kmeans++', 'kmeans' or provide an explicit matrix that is used as initial centroids for maximum diversity clustering. Default 'kmeans++'
<code>theta</code>	Numeric. Diversity penalty strength. 0 = plain spherical k-means (no batch mixing enforced). Default 2.
<code>sigma</code>	Numeric. Soft-cluster width. Larger values spread each cell across more clusters. Default 0.1.
<code>tau</code>	Numeric. Small-batch protection: batches with fewer than $\sim K \cdot \tau$ cells get a reduced theta. 0 disables. Default 0.
<code>alpha</code>	Numeric. Smoothing for the O/E ratio to prevent division-by-zero. Default 1.
<code>block_size</code>	Numeric in (0, 1]. Fraction of cells updated per block. Smaller = more accurate O/E bookkeeping, slower. Default 0.05.
<code>max_iter</code>	Integer. Maximum clustering iterations. Default 200.
<code>epsilon</code>	Numeric. Convergence tolerance on the objective. Default 1e-5.
<code>max_iter_kmeans</code>	Integer. Maximum iterations for initial kmeans refinement, if applicable.
<code>epsilon_kmeans</code>	Numeric. Convergence tolerance for initial kmeans refinement, if applicable.
<code>weight_power</code>	Numeric. Power law rescaling of kmeans++ weight. Default 1.0
<code>verbose</code>	Logical. Print iteration info. Default FALSE.

Details

Maximum Diversity Clustering implements the clustering objective used in Harmony (Korsunsky et al., 2019) independently of the Harmony integration workflow. The algorithm performs soft spherical k-means while encouraging clusters to contain a balanced representation of user-specified batches through a diversity penalty.

Cluster assignments are probabilistic and are returned as the matrix `R`. Hard assignments are obtained by assigning each observation to the cluster with the highest probability.

Value

A list containing

R Soft cluster assignments ($K \times N$).

Y Cluster centroids ($d \times K$).

cluster Hard assignments.

objective List containing optimisation traces: total, distance, entropy, diversity.

O Observed cluster-by-batch contingency table.

E Expected cluster-by-batch contingency table.

logOE Weighted log observed/expected ratio used in the diversity penalty.

batch_levels Batch labels

converged Whether convergence criterion was met.

References

Korsunsky I, Millard N, Fan J, *et al.* (2019). Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods* **16**, 1289–1296. doi:10.1038/s41592-019-0619-0

Examples

```
set.seed(1)

Z <- matrix(rnorm(10 * 100), nrow = 10)
batch <- rep(c("A", "B"), each = 50)

fit <- mdclust(
  Z,
  batch,
  Y_init = "kmeans++",
  K = 4
)

fit
summary(fit)

head(fit$cluster)

plot(fit)
```

plot.mdclust	<i>Plot a mdclust object</i>
--------------	------------------------------

Description

Minimal plot of the optimization trace, showing total objective as well as partial objectives for quick visual diagnostics.

Usage

```
## S3 method for class 'mdclust'
plot(x, ...)
```

Arguments

x	An object returned by <code>mdclust</code> .
...	Unused.

Value

The input object, invisibly.

print.mdclust	<i>Print a mdclust object</i>
---------------	-------------------------------

Description

Displays a concise human-readable overview of a fitted clustering model. Shows minimal information on algorithm convergence status, object size, and how to access detailed elements.

Usage

```
## S3 method for class 'mdclust'  
print(x, ...)
```

Arguments

x	An object returned by mdclust .
...	Unused.

Value

The input object, invisibly.

summary.mdclust	<i>Summarize a mdclust object</i>
-----------------	-----------------------------------

Description

Similar to [print.mdclust](#). Contains more details on state of the algorithm and objective and convergence.

Usage

```
## S3 method for class 'mdclust'  
summary(object, ...)
```

Arguments

object	An object returned by mdclust .
...	Unused.

Value

The input object, invisibly.

Index

`mdclust`, [2](#), [4](#), [5](#)

`plot.mdclust`, [4](#)

`print.mdclust`, [5](#), [5](#)

`summary.mdclust`, [5](#)