

# Package: mbkmeans (via r-universe)

June 30, 2024

**Type** Package

**Title** Mini-batch K-means Clustering for Single-Cell RNA-seq

**Version** 1.21.0

**Description** Implements the mini-batch k-means algorithm for large datasets, including support for on-disk data representation.

**Depends** R (>= 3.6)

**Imports** methods, DelayedArray, Rcpp, S4Vectors, SingleCellExperiment, SummarizedExperiment, ClusterR, benchmarkme, Matrix, BiocParallel

**Suggests** beachmat, HDF5Array, Rhdf5lib, BiocStyle, TENxPBMCDData, scater, DelayedMatrixStats, bluster, knitr, testthat, rmarkdown

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**LinkingTo** Rcpp, RcppArmadillo (>= 0.7.2), Rhdf5lib, beachmat, ClusterR

**SystemRequirements** C++11

**VignetteBuilder** knitr

**biocViews** Clustering, GeneExpression, RNASeq, Software, Transcriptomics, Sequencing, SingleCell

**BugReports** <https://github.com/drisso/mbkmeans/issues>

**Repository** <https://bioc.r-universe.dev>

**RemoteUrl** <https://github.com/bioc/mbkmeans>

**RemoteRef** HEAD

**RemoteSha** 8980ffa3a5a9186b0f7b0bd033bdab62936757b0

## Contents

blocksize . . . . .	2
clusterRows . . . . .	2
compute_wcss . . . . .	3
mbkmeans . . . . .	3
MbkmeansParam . . . . .	6
mini_batch . . . . .	6
predict_mini_batch . . . . .	8
predict_mini_batch_r . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

blocksize	<i>blocksize</i>
-----------	------------------

---

### Description

Return the maximum number of rows to use based on the amount of ram memory.

### Usage

```
blocksize(data, ram = get_ram())
```

### Arguments

data	matrix-like object.
ram	the max amount of ram (in bytes) to use.

### Value

Numeric value of the maximum number of rows.

### Examples

```
data <- matrix(NA, nrow = 100, ncol=1000)
blocksize(data, ram=1e6)
```

---

clusterRows	<i>Cluster rows of a matrix</i>
-------------	---------------------------------

---

### Description

Cluster rows of a matrix-like object with a variety of algorithms.

### Details

This function is deprecated. Please use the clusterRows function in the bluster Bioconductor package.

---

compute_wcss	<i>Compute Within-Cluster Sum of Squares</i>
--------------	----------------------------------------------

---

**Description**

Given a vector of cluster labels, a matrix of centroids, and a dataset, it computes the WCSS.

**Usage**

```
compute_wcss(clusters, cent, data)
```

**Arguments**

clusters	numeric vector with the cluster assignments.
cent	numeric matrix with the centroids (clusters in rows, variables in columns).
data	matrix-like object containing the data (numeric or integer).

**Value**

A numeric vector with the value of WCSS per cluster.

**Examples**

```
data = matrix(1:30,nrow = 10)
cl <- mini_batch(data, 2, 10, 10)
compute_wcss(cl$Clusters, cl$centroids, data)
```

---

mbkmeans	<i>Mini-Batch k-means for large single cell sequencing data</i>
----------	-----------------------------------------------------------------

---

**Description**

This is an implementation of the mini-batch k-means algorithm of Sculley (2010) for large single cell sequencing data with the dimensionality reduction results as input in the reducedDim() slot.

**Usage**

```
mbkmeans(x, ...)
```

```
## S4 method for signature 'SummarizedExperiment'
```

```
mbkmeans(x, whichAssay = 1, ...)
```

```
## S4 method for signature 'SingleCellExperiment'
```

```
mbkmeans(x, reduceMethod = "PCA", whichAssay = 1, ...)
```

```

## S4 method for signature 'LinearEmbeddingMatrix'
mbkmeans(x, ...)

## S4 method for signature 'ANY'
mbkmeans(
  x,
  clusters,
  batch_size = min(500, NCOL(x)),
  max_iters = 100,
  num_init = 1,
  init_fraction = batch_size/NCOL(x),
  initializer = "kmeans++",
  compute_labels = TRUE,
  calc_wcss = FALSE,
  early_stop_iter = 10,
  verbose = FALSE,
  CENTROIDS = NULL,
  tol = 1e-04,
  BPPARAM = BiocParallel::SerialParam(),
  ...
)

```

### Arguments

<code>x</code>	The object on which to run mini-batch k-means. It can be a matrix-like object (e.g., <code>matrix</code> , <code>Matrix</code> , <code>DelayedMatrix</code> , <code>HDF5Matrix</code> ) with genes in the rows and samples in the columns. Specialized methods are defined for <code>SummarizedExperiment</code> and <code>SingleCellExperiment</code> .
<code>...</code>	passed to <code>'blockApply'</code> .
<code>whichAssay</code>	The assay to use as input to mini-batch k-means. If <code>x</code> is a <code>SingleCellExperiment</code> , this is ignored unless <code>reduceMethod = NA</code> .
<code>reduceMethod</code>	Name of dimensionality reduction results to use as input to mini-batch k-means. Set to <code>NA</code> to use the full matrix.
<code>clusters</code>	the number of clusters
<code>batch_size</code>	the size of the mini batches. By default, it equals the minimum between the number of observations and 500.
<code>max_iters</code>	the maximum number of clustering iterations
<code>num_init</code>	number of times the algorithm will be run with different centroid seeds
<code>init_fraction</code>	proportion of data to use for the initialization centroids (applies if <code>initializer</code> is <code>kmeans++</code> ). Should be a float number between 0.0 and 1.0. By default, it uses the relative batch size.
<code>initializer</code>	the method of initialization. One of <code>kmeans++</code> and <code>random</code> . See details for more information
<code>compute_labels</code>	logical indicating whether to compute the final cluster labels.

<code>calc_wcss</code>	logical indicating whether the per-cluster WCSS is computed. Ignored if <code>'compute_labels = FALSE'</code> .
<code>early_stop_iter</code>	continue that many iterations after calculation of the best within-cluster-sum-of-squared-error
<code>verbose</code>	either TRUE or FALSE, indicating whether progress is printed during clustering
<code>CENTROIDS</code>	a matrix of initial cluster centroids. The rows of the <code>CENTROIDS</code> matrix should be equal to the number of clusters and the columns should be equal to the columns of the data
<code>tol</code>	a float number. If, in case of an iteration ( <code>iteration &gt; 1</code> and <code>iteration &lt; max_iters</code> ) <code>'tol'</code> is greater than the squared norm of the centroids, then <code>kmeans</code> has converged
<code>BPPARAM</code>	See the <code>'BiocParallel'</code> package. Only the label assignment is done in parallel.

### Details

The implementation is largely based on the [MiniBatchKmeans](#) function of the `ClusterR` package. The contribution of this package is to provide support for on-disk data representations such as HDF5, through the use of `DelayedMatrix` and `HDF5Matrix` objects, as well as for sparse data representation through the classes of the `Matrix` package. We also provide high-level methods for objects of class `SummarizedExperiment`, `SingleCellExperiment`, and `LinearEmbeddingMatrix`.

This function performs k-means clustering using mini batches.

**kmeans++:** `kmeans++` initialization. Reference : <http://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf> AND <http://stackoverflow.com/questions/5466323/how-exactly-does-k-means-work>

**random:** random selection of data rows as initial centroids

### Value

A list with the following attributes: `centroids`, `WCSS_per_cluster`, `best_initialization`, `iters_per_initialization`.  
a list with the following attributes: `centroids`, `WCSS_per_cluster`, `best_initialization`, `iters_per_initialization`

### Author(s)

Lampros Mouselimis and Yuwei Ni

### References

Sculley. Web-Scale K-Means Clustering. WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. ACM 978-1-60558-799-8/10/04.

<https://github.com/mlampros/ClusterR>

### Examples

```
library(SummarizedExperiment)
se <- SummarizedExperiment(matrix(rnorm(100), ncol=10))
mbkmeans(se, clusters = 2)
library(SingleCellExperiment)
```

```
sce <- SingleCellExperiment(matrix(rnorm(100), ncol=10))
mbkmeans(sce, clusters = 2, reduceMethod = NA)
x<-matrix(rnorm(100), ncol=10)
mbkmeans(x,clusters = 3)
```

---

MbkmeansParam	<i>Mini-batch k-means clustering</i>
---------------	--------------------------------------

---

### Description

Run the mini-batch k-means [mbkmeans](#) function with the specified number of centers within `clusterRows` from the `bluster` Bioconductor package.

### Usage

```
MbkmeansParam(centers, ...)
```

### Arguments

<code>centers</code>	An integer scalar specifying the number of centers. Alternatively, a function that takes the number of observations and returns the number of centers. Note, the <a href="#">mbkmeans</a> function uses the argument <code>clusters</code> argument to represent this argument. However, we use <code>centers</code> to match
<code>...</code>	Further arguments to pass to <a href="#">mbkmeans</a> .

### Details

This function is deprecated. Please use the `MbkmeansParam` function in the `bluster` Bioconductor package.

---

mini_batch	<i>Mini_batch</i>
------------	-------------------

---

### Description

Mini-batch-k-means for matrix-like objects

**Usage**

```

mini_batch(
  data,
  clusters,
  batch_size,
  max_iters,
  num_init = 1L,
  init_fraction = 1,
  initializer = "kmeans++",
  compute_labels = TRUE,
  calc_wcss = FALSE,
  early_stop_iter = 10L,
  verbose = FALSE,
  CENTROIDS = NULL,
  tol = 1e-04
)

```

**Arguments**

<code>data</code>	numeric or integer matrix-like object.
<code>clusters</code>	the number of clusters.
<code>batch_size</code>	the size of the mini batches.
<code>max_iters</code>	the maximum number of clustering iterations.
<code>num_init</code>	number of times the algorithm will be run with different centroid seeds.
<code>init_fraction</code>	percentage of data to use for the initialization centroids (applies if initializer is <i>kmeans++</i> ). Should be a float number between 0.0 and 1.0.
<code>initializer</code>	the method of initialization. One of <i>kmeans++</i> and <i>random</i> . See details for more information.
<code>compute_labels</code>	logical indicating whether to compute the final cluster labels.
<code>calc_wcss</code>	logical indicating whether the within-cluster sum of squares should be computed and returned (ignored if <code>'compute_labels = FALSE'</code> ).
<code>early_stop_iter</code>	continue that many iterations after calculation of the best within-cluster-sum-of-squared-error.
<code>verbose</code>	logical indicating whether progress is printed on screen.
<code>CENTROIDS</code>	an optional matrix of initial cluster centroids. The rows of the <code>CENTROIDS</code> matrix should be equal to the number of clusters and the columns should be equal to the columns of the data.
<code>tol</code>	convergence tolerance.

**Details**

This function performs k-means clustering using mini batches. It was inspired by the implementation in <https://github.com/mlampros/ClusterR>.

The input matrix can be in any format supported by the ‘DelayedArray’ / ‘beachmat’ framework, including the matrix classes defined in the ‘Matrix’ package and the ‘HDFMatrix’ class.

There are two possible initializations.

**kmeans++:** kmeans++ initialization.

**random:** random selection of data rows as initial centroids.

### Value

a list with the following attributes:

centroids: the final centroids;

WCSS\_per\_cluster (optional): the final per-cluster WCSS.

best\_initialization: which initialization value led to the best WCSS solution;

iters\_per\_initialization: number of iterations per each initialization;

Clusters (optional): the final cluster labels.

### References

Sculley, D., 2010, April. Web-scale k-means clustering. In Proceedings of the 19th international conference on World wide web (pp. 1177-1178). ACM.

Arthur, D. and Vassilvitskii, S., 2007, January. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035). Society for Industrial and Applied Mathematics.

### Examples

```
data = matrix(1:30,nrow = 10)
mini_batch(data, 2, 10, 10)
```

---

predict\_mini\_batch      *Predict\_mini\_batch*

---

### Description

Prediction function for mini-batch k-means applied to matrix-like objects.

### Usage

```
predict_mini_batch(data, CENTROIDS)
```

### Arguments

**data**                      matrix-like object containing numeric or integer data (observations in rows, variables in columns).

**CENTROIDS**                a matrix of initial cluster centroids. The rows of the CENTROIDS matrix should be equal to the number of clusters and the columns should equal the columns of the data.



## Details

This function takes the data and the output centroids and returns the clusters.

This implementation relies very heavily on the [MiniBatchKmeans](#) implementation. We provide the ability to work with other matrix-like objects other than base matrices (e.g, DelayedMatrix and HDF5Matrix) through the beachmat library.

## Value

it returns a vector with the clusters.

## Author(s)

Yuwei Ni

## Examples

```
data(iris)
km = mini_batch(as.matrix(iris[,1:4]), clusters = 3,
               batch_size = 10, max_iters = 10)
clusters = predict_mini_batch(as.matrix(iris[,1:4]),
                             CENTROIDS = km$centroids)
```

---

predict\_mini\_batch\_r *Compute labels for mini-batch k-means*

---

## Description

Given a data matrix and a centroid matrix, it assigns each data point to the closest centroid, using block processing.

## Usage

```
predict_mini_batch_r(
  data,
  centroids,
  BPPARAM = BiocParallel::SerialParam(),
  ...
)
```

## Arguments

data	a matrix-like object with features in row and samples in columns.
centroids	a matrix with the coordinates of the centroids.
BPPARAM	for parallel computations. See the ‘BiocParallel’ package.
...	passed to ‘blockApply’.

**Value**

a vector of cluster labels for each observation.

**Examples**

```
data(iris)
km <- mini_batch(as.matrix(iris[,1:4]), clusters = 3,
                 batch_size = 10, max_iters = 100)
predict_mini_batch_r(t(as.matrix(iris[,1:4])), km$centroids)
```

# Index

blocksize, [2](#)

clusterRows, [2](#)

compute\_wcss, [3](#)

mbkmeans, [3](#), [6](#)

mbkmeans, ANY-method (mbkmeans), [3](#)

mbkmeans, LinearEmbeddingMatrix-method  
(mbkmeans), [3](#)

mbkmeans, SingleCellExperiment-method  
(mbkmeans), [3](#)

mbkmeans, SummarizedExperiment-method  
(mbkmeans), [3](#)

MbkmeansParam, [6](#)

mini\_batch, [6](#)

MiniBatchKmeans, [5](#), [9](#)

predict\_mini\_batch, [8](#)

predict\_mini\_batch\_r, [9](#)

show, MbkmmeansParam-method  
(MbkmeansParam), [6](#)