

# Package ‘akmbiclust’

May 7, 2026

**Title** Alternating K-Means Biclustering

**Version** 0.1.0

**Description** Implements the alternating k-means biclustering algorithm in Fraiman and Li (2020) <[doi:10.48550/arXiv.2009.04550](https://doi.org/10.48550/arXiv.2009.04550)>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Author** Zichao Li [aut, cre],  
Nicolas Fraiman [aut]

**Maintainer** Zichao Li <[lizichao@live.unc.edu](mailto:lizichao@live.unc.edu)>

**Repository** CRAN

**Date/Publication** 2020-09-24 09:10:15 UTC

## Contents

akmbiclust . . . . .	1
<b>Index</b>	<b>3</b>

---

akmbiclust	<i>Alternating k-means biclustering</i>
------------	---

---

## Description

This function uses the alternating k-means biclustering algorithm to extract the k biclusters in the matrix X. See the paper "Biclustering with Alternating K-Means" for more details.

**Usage**

```
akmbiclust(X, k, lambda = 0, nstart = 1)
```

**Arguments**

X	Data matrix.
k	The number of biclusters.
lambda	Regularization parameter. Default is 0.
nstart	The number of random initializations. Default is 1.

**Value**

A list containing three objects:

row_labels	The bicluster labels of every row.
col_labels	The bicluster labels of every column.
loss	The loss of the produced biclusters.

**Author(s)**

Nicolas Fraiman and Zichao Li

**References**

N. Fraiman and Z. Li (2020). Biclustering with Alternating K-Means. arXiv preprint arXiv:2009.04550.

**Examples**

```
# we create a 100 by 100 matrix X which has an underlying 2 by 2 block structure.
# The entries in the two 50 by 50 blocks on the top left and bottom right follow
# i.i.d. normal with mean 0 and variance 4. The entries in the two 50 by 50 blocks
# on the top right and bottom left follow i.i.d. normal with mean 0 and variance 1.

X <- matrix(rnorm(10000, 0, 1), 100, 100)
X[1:50, 1:50] <- matrix(rnorm(2500, 0, 2), 50, 50)
X[51:100, 51:100] <- matrix(rnorm(2500, 0, 2), 50, 50)

# Alternating k-means biclustering
# Result: perfect
result <- akmbiclust(X, 2, lambda = 0, nstart = 100)
result$row_labels
result$col_labels

# Separate k-means clustering on the rows and columns
# Result: random
kmeans(X, 2)$cluster
kmeans(t(X), 2)$cluster
```

# Index

akmbiclust, 1