

# Package: fusionclust (via r-universe)

September 10, 2024

**Title** Clustering and Feature Screening using L1 Fusion Penalty

**Version** 1.0.0

**Description** Provides the Big Merge Tracker and COSCI algorithms for convex clustering and feature screening using L1 fusion penalty.

**Depends** R (>= 3.4.1)

**License** GPL(>=2)

**Encoding** UTF-8

**LazyData** true

**Imports** bbmle, stats, graphics

**RoxygenNote** 6.0.1

**URL** <https://github.com/trambakbanerjee/fusionclust>

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

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

**RemoteUrl** <https://github.com/trambakbanerjee/fusionclust>

**RemoteRef** HEAD

**RemoteSha** 8b6d438ece9f8bb2864ba039e7f2a283100aba02

## Contents

bmt	2
cosci_is	3
cosci_is_select	4
nclust	5

<b>Index</b>	<b>7</b>
--------------	----------

---

**bmt***Big Merge Tracker*

---

**Description**

Solves an L1 relaxed univariate clustering criterion and returns a sequence of  $\lambda$  values where the clusters merge

**Usage**

```
bmt(x, alpha = 0.1, small.perturbation = 10(-6))
```

**Arguments**

<code>x</code>	observation vector
<code>alpha</code>	merging threshold. Default is 0.1
<code>small.perturbation</code>	a small positive number to remove ties. Default is $10^{(-6)}$

**Details**

solves a convex relaxation of the univariate clustering criterion given by equation (2) in the referenced paper and generates a sequence of cluster merges and corresponding  $\lambda$  values. See algorithm 1 in the referenced paper for more details.

**Value**

1. `path` - number of clusters on the big merge path
2. `lambda.path` - sequence of lambda where clusters merge
3. `index` - cluster index at the point where clusters merge
4. `merge` - merge points
5. `split` - split points
6. `prob` - merging proportion
7. `boundaries` - cluster boundaries

**References**

1. P. Radchenko, G. Mukherjee, Convex clustering via l1 fusion penalization, J. Roy. Statist. Soc. Ser. B (Statistical Methodology) (2017) doi:10.1111/rssb.12226.

**See Also**

[nclust](#)

**Examples**

```
library(fusionclust)
set.seed(42)
x<- c(rnorm(1000,-2,1), rnorm(1000,2,1))
out<- bmt(x)
```

---

`cosci_is`*Rank the p features in an n by p design matrix*

---

**Description**

Ranks the  $p$  features in an  $n$  by  $p$  design matrix where  $n$  represents the sample size and  $p$  is the number of features.

**Usage**

```
cosci_is(dat, min.alpha, small.perturbation = 10(-6))
```

**Arguments**

<code>dat</code>	<code>n</code> by <code>p</code> data matrix
<code>min.alpha</code>	the smallest threshold (typically set to 0)
<code>small.perturbation</code>	a small positive number to remove ties. Default value is $10^{(-6)}$

**Details**

Uses the univariate merging algorithm [bmt](#) and produces a score for each feature that reflects its relative importance for clustering.

**Value**

a  $p$  vector of scores

**References**

1. Banerjee, T., Mukherjee, G. and Radchenko P., Feature Screening in Large Scale Cluster Analysis, Journal of Multivariate Analysis, Volume 161, 2017, Pages 191-212
2. P. Radchenko, G. Mukherjee, Convex clustering via l1 fusion penalization, J. Roy. Statist. Soc. Ser. B (Statistical Methodology) (2017) doi:10.1111/rssb.12226.

**See Also**

[bmt](#), [cosci\\_is\\_select](#)

## Examples

```
library(fusionclust)
set.seed(42)
noise<-matrix(rnorm(49000),nrow=1000,ncol=49)
set.seed(42)
signal<-c(rnorm(500,-1.5,1),rnorm(500,1.5,1))
x<-cbind(signal,noise)
scores<- cosci_is(x,0)
```

---

cosci_is_select	<i>Use a data driven approach to select the features</i>
-----------------	--

---

## Description

Once you have the feature scores from `cosci_is`, you can select the features

1. based on a pre-defined threshold,
2. using table A.10 in the paper[1] to determine an appropriate threshold or,
3. using a data driven approach described in the references to select the features and obtain an implicit threshold value.

`cosci_is_select` implements option 3.

## Usage

```
cosci_is_select(score, gamma)
```

## Arguments

score	a p vector of scores
gamma	what proportion of the p features is noise? If your sample size n is smaller than 100, setting gamma = 0.85 is recommended. Otherwise set gamma = 0.9

## Details

Converts the problem of screening out features with lower scores into a problem in large scale multiple testing and uses the procedure described in reference [2] to determine the signal features.

## Value

a vector of selected features

## References

1. Banerjee, T., Mukherjee, G. and Radchenko P., Feature Screening in Large Scale Cluster Analysis, Journal of Multivariate Analysis, Volume 161, 2017, Pages 191-212
2. T. Cai, W. Sun, W., Optimal screening and discovery of sparse signals with applications to multistage high throughput studies, J. Roy. Statist. Soc. Ser. B (Statistical Methodology) 79, no. 1 (2017) 197-223

**See Also**[cosci\\_is](#)**Examples**

```
library(fusionclust)
set.seed(42)
noise<-matrix(rnorm(49000),nrow=1000,ncol=49)
set.seed(42)
signal<-c(rnorm(500,-1.5,1),rnorm(500,1.5,1))
x<-cbind(signal,noise)
scores<- cosci_is(x,0)
features<-cosci_is_select(scores,0.9)
```

---

nclust	<i>No.of clusters</i>
--------	-----------------------

---

**Description**

Estimates the number of clusters from the [bmt](#) run

**Usage**

```
nclust(bmt_output, prob_threshold = 0.5)
```

**Arguments**

`bmt_output` output from the [bmt](#) run  
`prob_threshold` probability threshold. Default is 0.5. Do not change it unless you know what you are doing. See the referenced paper

**Details**

Estimates the number of clusters as the number of big merges + 1. The probability threshold is an adjustment that renders this estimation process more robust to sampling fluctuations. If the sum of the sample frequencies for the two merging clusters in the last big merge is less than 50 percent, we do not report any merges and thus are left with just 1 cluster. See the referenced paper for more details.

**Value**

The number of clusters

**References**

1. P. Radchenko, G. Mukherjee, Convex clustering via l1 fusion penalization, J. Roy. Statist. Soc. Ser. B (Statistical Methodology) (2017) doi:10.1111/rssb.12226.

**See Also**[bmt](#)**Examples**

```
library(fusionclust)
set.seed(42)
x<- c(rnorm(1000,-2,1), rnorm(1000,2,1))
out<- bmt(x)
k<- nclust(out)
```

# Index

bmt, [2](#), [3](#), [5](#), [6](#)

cosci\_is, [3](#), [4](#), [5](#)

cosci\_is\_select, [3](#), [4](#)

nclust, [2](#), [5](#)