Package 'SCBiclust'

October 12, 2022

Title Identifies Mean, Variance, and Hierarchically Clustered Biclusters

Version 1.0.1

Date 2022-06-09

Author Erika S. Helgeson, Qian Liu, Guanhua Chen, Michael R. Kosorok , and Eric Bair

Maintainer Erika S. Helgeson <helge@umn.edu>

Description Identifies a bicluster, a submatrix of the data such that the features and observations within the submatrix differ from those not contained in submatrix, using a two-step method. In the first step, observations in the bicluster are identified to maximize the sum of weighted between cluster feature differences. The method is described in Helgeson et al. (2020) <doi:10.1111/biom.13136>. 'SCBiclust' can be used to identify biclusters which differ based on feature means, feature variances, or more general differences.

Depends R (>= 3.4.0)

Imports sparcl, sigclust

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-09 22:40:22 UTC

R topics documented:

`PermBiclust.beta.ks` *'SCBiclust' method for identifying means-based biclusters with Kolmogorov-Smirnov test of feature weights*

Description

'SCBiclust' method for identifying means-based biclusters with Kolmogorov-Smirnov test of feature weights

Usage

```
PermBiclust.beta.ks(
  x,
  nperms = 1000,
  silent = TRUE,
  maxnum.bicluster = 5,
  ks.alpha = 0.05
)
```

Arguments

`x`	a dataset with n rows and p columns, with observations in rows.
`nperms`	number of $Beta(\frac{1}{2}, (p-1)/2)$ distributed variables generated for each feature (default=1000)
`silent`	should progress be printed? (default=TRUE)
`maxnum.bicluster`	The maximum number of biclusters returned
`ks.alpha`	significance level for Kolmogorov-Smirnov test.

Details

Observations in the bicluster are identified such that they maximize the feature-weighted square-root of the between cluster sum of squares. Features in the bicluster are identified based on their contribution to the clustering of the observations. Feature weights are generated in a similar fashion as `KMeansSparseCluster` except with a modified objective function and no sparsity constraint.

This algoritm uses a numerical approximation to $E(\sqrt{B})$ where $B \sim Beta(\frac{1}{2}, (p-1)/2)$ as the expected null distribution for feature weights. The Kolmogorov-Smirnov test is used to assess if feature weights deviate from the expected null distribution.

Value

The function returns a S3-object with the following attributes:

- `num.bicluster`: The number of biclusters estimated by the procedure.
- `x.residual`: The data matrix `x` after removing the signals

- `which.x`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data observation is in the given bicluster.
- `which.y`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data feature is in the given bicluster.

Author(s)

Erika S. Helgeson, Qian Liu, Guanhua Chen, Michael R. Kosorok , and Eric Bair

Examples

```
test <- matrix(rnorm(100*200), nrow=100, ncol=200)
test[1:20,1:20] <- test[1:20,1:20]+rnorm(20*20, 2)
test[16:30,51:80] <- test[16:30,51:80]+rnorm(15*30, 3)
PermBiclust.beta.ks(test, silent=TRUE)
```

PermBiclust.sigclust *'SCBiclust' method for identifying means-based biclusters*

Description

'SCBiclust' method for identifying means-based biclusters

Usage

```
PermBiclust.sigclust(
  x,
  nperms = 1000,
  silent = TRUE,
  maxnum.bicluster = 5,
  alpha = 0.05,
  icovest = 1
)
```

Arguments

`x` a dataset with n rows and p columns, with observations in rows.

`nperms` number of $Beta(\frac{1}{2},(p-1)/2)$ distributed variables generated for each feature (default=1000)

`silent` should progress be printed? (default=TRUE)

`maxnum.bicluster` The maximum number of biclusters returned

`alpha` significance level for `sigclust` test.

`icovest` Coviariance estimation type for `sigclust` test

Details

Observations in the bicluster are identified such that they maximize the feature-weighted between cluster sum of squares. Features in the bicluster are identified based on their contribution to the clustering of the observations. Feature weights are generated in a similar fashion as KMeansSparseCluster except with a modified objective function and no sparsity constraint. This algoritm uses a numerical approximation to $E(\sqrt{B})$ where $B \sim Beta(\frac{1}{2}, (p-1)/2)$ as the expected null distribution for feature weights. The sigclust algorithm is used to test the strength of the identified clusters.

Value

The function returns a S3-object with the following attributes:

- `num.bicluster`: The number of biclusters estimated by the procedure.
- `x.residual`: The data matrix `x` after removing the signals
- `which.x`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data observation is in the given bicluster.
- `which.y`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data feature is in the given bicluster.

Author(s)

Erika S. Helgeson, Qian Liu, Guanhua Chen, Michael R. Kosorok , and Eric Bair

Examples

```
 test <- matrix(rnorm(60*180), nrow=60, ncol=180)
test[1:15,1:15] <- test[1:15,1:15]+rnorm(15*15, 2)
test[16:30,51:80] <- test[16:30,51:80]+rnorm(15*30, 3)
PermBiclust.sigclust(test, silent=TRUE)
```

PermBiclust.sigclust_stop

'SCBiclust' method for identifying means-based biclusters with optional cluster significance testing

---

Description

'SCBiclust' method for identifying means-based biclusters with optional cluster significance testing

Usage

```
PermBiclust.sigclust_stop(
  x,
  nperms = 1000,
  silent = TRUE,
  maxnum.bicluster = 5,
  alpha = 0.05,
  icovest = 1,
  sc = TRUE
)
```

Arguments

x	a dataset with n rows and p columns, with observations in rows.
nperms	number of $Beta(\frac{1}{2}, (p-1)/2)$ distributed variables generated for each feature (default=1000)
silent	should progress be printed? (default=TRUE)
maxnum.bicluster	The maximum number of biclusters returned
alpha	significance level for sigclust test.
icovest	Coviariance estimation type for sigclust test
sc	should the sigclust test be used? (default=TRUE)

Details

Observations in the bicluster are identified such that they maximize the feature-weighted between cluster sum of squares. Features in the bicluster are identified based on their contribution to the clustering of the observations. Feature weights are generated in a similar fashion as KMeansSparseCluster except with a modified objective function and no sparsity constraint. This algoritm uses a numerical approximation to $E(\sqrt{B})$ where $B \sim Beta(\frac{1}{2}, (p-1)/2)$ as the expected null distribution for feature weights. Use of the sigclust algorithm to test the strength of the identified clusters is optional in this implementation of the algorithm.

Value

The function returns a S3-object with the following attributes:

- num.bicluster: The number of biclusters estimated by the procedure.
- x.residual: The data matrix x after removing the signals
- which.x: A list of length num.bicluster with each list entry containing a logical vector denoting if the data observation is in the given bicluster.
- which.y: A list of length num.bicluster with each list entry containing a logical vector denoting if the data feature is in the given bicluster.

Author(s)

Erika S. Helgeson, Qian Liu, Guanhua Chen, Michael R. Kosorok , and Eric Bair

Examples

```
test <- matrix(rnorm(60*180), nrow=60, ncol=180)
test[1:15,1:15] <- test[1:15,1:15]+rnorm(15*15, 2)
test[16:30,51:80] <- test[16:30,51:80]+rnorm(15*30, 3)
PermBiclust.sigclust_stop(test, silent=TRUE)
```

PermHclust.sigclust *'SCBiclust' method for identifying hierarchically clustered biclusters*

Description

'SCBiclust' method for identifying hierarchically clustered biclusters

Usage

```
PermHclust.sigclust(
  x = NULL,
  method = c("average", "complete", "single", "centroid"),
  wbound = sqrt(ncol(x)),
  alpha = 0.05,
  dat.perms = 1000,
  dissimilarity = c("squared.distance", "absolute.value"),
  silent = TRUE,
  sigstep = FALSE
)
```

Arguments

`x` a dataset with n rows and p columns, with observations in rows.

`method` method for agglomeration. See documentation in `hclust`. (default="average")

`wbound` the tuning parameter for sparse hierarchical clustering. See documentation in `HierarchicalSparseCluster`. (default=`sqrt(ncol(x))`)

`alpha` significance level for `sigclust` test.

`dat.perms` number of $Beta(\frac{1}{2},(p-1)/2)$ distributed variables generated for each feature (default=1000)

`dissimilarity` How should dissimilarity be calculated? (default is "squared.distance").

`silent` should progress be printed? (default=TRUE)

`sigstep` Should `sigclust` be used to assess the strength of identified clusters? (default=FALSE)

Details

Observations in the bicluster are identified such that they maximize the feature-weighted version of the dissimilarity matrix as implemented in `HierarchicalSparseCluster`. Features in the bicluster are identified based on their contribution to the clustering of the observations. #' This algoritm uses a numerical approximation to $E(\sqrt{B})$ where $B \sim Beta(\frac{1}{2},(p-1)/2)$ as the expected null distribution for feature weights.

Value

The function returns a S3-object with the following attributes:

- `which.x`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data observation is in the given bicluster.
- `which.y`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data feature is in the given bicluster.

Author(s)

Erika S. Helgeson, Qian Liu, Guanhua Chen, Michael R. Kosorok , and Eric Bair

Examples

```
test <- matrix(nrow=500, ncol=50)
theta <- rep(NA, 500)
theta[1:300] <- runif(300, 0, pi)
theta[301:500] <- runif(200, pi, 2*pi)
test[1:300,seq(from=2,to=40,by=2)] <- -2+5*sin(theta[1:300])
test[301:500,seq(from=2,to=40,by=2)] <- 5*sin(theta[301:500])
test[1:300,seq(from=1,to=39,by=2)] <- 5+5*cos(theta[1:300])
test[301:500,seq(from=1,to=39,by=2)] <- 5*cos(theta[301:500])
test[,1:40] <- test[,1:40] + rnorm(40*500, 0, 0.2)
test[,41:50] <- rnorm(10*500, 0, 1)
test.PermBiclust <- PermHclust.sigclust(x=test, method='single', dissimilarity='squared.distance')
```

VarPermBiclust.chisqdiff *'SCBiclust' method for identifying variance-based biclusters*

Description

'SCBiclust' method for identifying variance-based biclusters

Usage

```
VarPermBiclust.chisqdiff(
  x,
  min.size = max(5, round(nrow(x)/20)),
  nperms = 1000,
  silent = TRUE
)
```

Arguments

`x` a dataset with n rows and p columns, with observations in rows.

`min.size` Minimum size of observations included in a valid bicluster (default=`max(5,round(nrow(x)/20))`)

`nperms` number of $\chi^2_{n_1}$ and $\chi^2_{n_2}$ variables generated for each feature where n_1 and n_2 are the number of observations in cluster 1 and cluster 2, respectively. (default=100)

`silent` should progress be printed? (default=TRUE)

Details

Observations in the bicluster are identified such that they maximize the feature-weighted sum of between cluster difference in feature variances. Features in the bicluster are identified based on their contribution to the clustering of the observations. This algoritm uses a numerical approximation $log(abs(\chi^2_{n_1} - chi^2_{n_2}) + 1)$ as the expected null distribution for feature weights.

`VarPermBiclust.chisqdiff` will identify at most one variance bicluster. To identify additional biclusters first the feature signal of the identified bicluster should be removed by scaling the variance of elements in the previously identified bicluster, Then `VarPermBiclust.chisqdiff` can be used on the residual data matrix. (see example)

Value

The function returns a S3-object with the following attributes:

- `which.x`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data observation is in the given bicluster.
- `which.y`: A list of length `num.bicluster` with each list entry containing a logical vector denoting if the data feature is in the given bicluster.

Author(s)

Erika S. Helgeson, Qian Liu, Guanhua Chen, Michael R. Kosorok , and Eric Bair

Examples

```
test <- matrix(rnorm(100*50, mean=1, sd=2), nrow=100)
test[1:30, 1:20] <- matrix(rnorm(30*20, mean=1, sd=15), nrow=30)
test.VarPermBiclust <- VarPermBiclust.chisqdiff(test)
x=test.VarPermBiclust$which.x
y=test.VarPermBiclust$which.y
# Code for identifying additional biclusters after removing bicluster signal

temp <- scale(test)
temp[x,y] <-t(t(temp[x,y])*(apply(temp[!x,y],2,sd)/
                              apply(temp[x,y],2,sd)))
test.VarPermBiclust.2 <- VarPermBiclust.chisqdiff(temp)
```

Index