Package: biclust (via r-universe)

November 13, 2026

Version 2.0.3.1.9001

Date 2021-05-22

Author Sebastian Kaiser, Rodrigo Santamaria, Tatsiana Khamiakova, Martin Sill, Roberto Theron, Luis Quintales, Friedrich Leisch, Ewoud De Troyer and Sami Leon.

Maintainer Sebastian Kaiser <dr.sebastian.kaiser@gmail.com>

Title BiCluster Algorithms

Depends R (>= 2.10), MASS, grid, colorspace, lattice

Imports methods, flexclust, additivityTests, tidyr, ggplot2

Suggests isa2

Description The main function biclust() provides several algorithms to find biclusters in two-dimensional data: Cheng and Church (2000, ISBN:1-57735-115-0), spectral (2003) <doi:10.1101/gr.648603>, plaid model (2005) <doi:10.1016/j.csda.2004.02.003>, xmotifs (2003) <doi:10.1142/9789812776303_0008> and bimax (2006) <doi:10.1093/bioinformatics/btl060>. In addition, the package provides methods for data preprocessing (normalization and discretisation), visualisation, and validation of bicluster solutions. Compatibility updates by Renova.

License GPL-3

LazyLoad yes

NeedsCompilation yes

Config/pak/sysreqs libicu-dev

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-13 17:52:07 UTC

RemoteUrl https://github.com/renova-r/biclust

RemoteRef renova-2.0.3.1.9001

RemoteSha 9b88b81886b954d9b673b05ee90446eec3b22cca

Contents

BCBimax . . . 3
BCCC . . . 4
BCPlaid . . . 5
BCQuest . . . 6
BCSpectral . . . 8
BCXmotifs . . . 9
BicatYeast . . . 11
biclust . . . 11
Biclust-class . . . 12
biclustbarchart . . . 13
bicluster . . . 14
biclustmember . . . 15
BiclustMethod-class . . . 16
bimax.grid . . . 17
binarize . . . 17
bubbleplot . . . 18
ChiaKaruturi . . . 20
coherence . . . 21
computeObservedFstat . . . 23
diagnoseColRow . . . 24
diagnosticPlot . . . 26
diagnosticPlot2 . . . 27
diagnosticTest . . . 28
discretize . . . 31
drawHeatmap . . . 32
EisenYeast . . . 33
ensemble . . . 33
heatmapBC . . . 35
isoverlapp . . . 36
jaccardind . . . 36
parallelCoordinates . . . 37
plaid.grid . . . 39
plotclust . . . 40
predictBimax . . . 41
SyntrenEcoli . . . 42
writeBiclusterResults . . . 42
writeclust . . . 43

Index **45**

BCBimax *The Bimax Bicluster algorithm*

Description

Performs Bimax Biclustering based on the framework by Prelic et. al.(2006). It searches for submatrices of ones in a logical matrix. Uses the original C code of the authors.

Usage

```
## S4 method for signature 'matrix,BCBimax'
biclust(x, method=BCBimax(), minr=2, minc=2, number=100)
## S4 method for signature 'matrix,BCrepBimax'
biclust(x, method=BCrepBimax(), minr=2, minc=2, number=100, maxc=12)
```

Arguments

x	A logical matrix which represents the data.
method	Here BCBimax, to perform Bimax algorithm
minr	Minimum row size of resulting bicluster.
minc	Minimum column size of resulting bicluster.
number	Number of Bicluster to be found.
maxc	Maximum column size of resulting bicluster.

Value

Returns an object of class Biclust.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

References

Prelic, A.; Bleuler, S.; Zimmermann, P.; Wil, A.; Buhlmann, P.; Gruissem, W.; Hennig, L.; Thiele, L. & Zitzler, E. A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data Bioinformatics, Oxford Univ Press, 2006, 22, 1122-1129

See Also

biclust, Biclust

Examples

```
test <- matrix(rnorm(5000), 100, 50)
test[11:20,11:20] <- rnorm(100, 3, 0.1)
loma <- binarize(test,2)
res <- biclust(x=loma, method=BCBimax(), minr=4, minc=4, number=10)
res
```

BCCC *The CC Bicluster algorithm*

Description

Performs CC Biclustering based on the framework by Cheng and Church (2000). Searches for submatrices with a score lower than a specific treshold in a standardized data matrix.

Usage

```
## S4 method for signature 'matrix,BCCC'
biclust(x, method=BCCC(), delta = 1.0, alpha=1.5, number=100)
```

Arguments

x	Data matrix.
method	Here BCCC, to perform CC algorithm
delta	Maximum of accepted score.
alpha	Scaling factor.
number	Number of bicluster to be found.

Value

Returns an object of class Biclust.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

References

Cheng, Y. & Church, G.M. Biclustering of Expression Data Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, 2000, 1, 93-103

See Also

biclust, Biclust

Examples

```
test <- matrix(rbinom(400, 50, 0.4), 20, 20)
res <- biclust(test, method=BCCC(), delta=1.5,  alpha=1, number=10)
res
```

BCPlaid *The Plaid Model Bicluster algorithm*

Description

Performs Plaid Model Biclustering as described in Turner et al., 2003. This is an improvement of original 'Plaid Models for Gene Expression Data' (Lazzeroni and Owen, 2002). This algorithm models data matrices to a sum of layers, the model is fitted to data through minimization of error.

Usage

```
## S4 method for signature 'matrix,BCPlaid'
biclust(x, method=BCPlaid(), cluster="b", fit.model = y ~ m + a + b,
  background = TRUE, background.layer = NA, background.df = 1, row.release = 0.7,
  col.release = 0.7, shuffle = 3, back.fit = 0, max.layers = 20, iter.startup = 5,
   iter.layer = 10, verbose = TRUE)
```

Arguments

x	The data matrix where biclusters have to be found
method	Here BCPlaid, to perform Plaid algorithm
cluster	'r', 'c' or 'b', to cluster rows, columns or both (default 'b')
fit.model	Model (formula) to fit each layer. Usually, a linear model is used, that estimates three parameters: m (constant for all elements in the bicluster), a(contant for all rows in the bicluster) and b (constant for all columns). Thus, default is: y ~ m + a + b.
background	If 'TRUE' the method will consider that a background layer (constant for all rows and columns) is present in the data matrix.
background.layer	If background='TRUE' a own background layer (Matrix with dimension of x) can be specified.
background.df	Degrees of Freedom of backround layer if background.layer is specified.
shuffle	Before a layer is added, it's statistical significance is compared against a number of layers obtained by random defined by this parameter. Default is 3, higher numbers could affect time performance.
iter.startup	Number of iterations to find starting values
iter.layer	Number of iterations to find each layer
back.fit	After a layer is added, additional iterations can be done to refine the fitting of the layer (default set to 0)

row.release	Scalar in [0,1](with interval recommended [0.5-0.7]) used as threshold to prune rows in the layers depending on row homogeneity
col.release	As above, with columns
max.layers	Maximum number of layer to include in the model
verbose	If 'TRUE' prints extra information on progress.

Value

Returns an Biclust object.

Author(s)

Adaptation of original code from Heather Turner from Rodrigo Santamaria <rodri@usal.es>. <rodri@usal.es>

References

Heather Turner et al, "Improved biclustering of microarray data demonstrated through systematic performance tests",Computational Statistics and Data Analysis, 2003, vol. 48, pages 235-254.

Lazzeroni and Owen, "Plaid Models for Gene Expression Data", Standford University, 2002.

Examples

```
#Random matrix with embedded bicluster
test <- matrix(rnorm(5000),100,50)
test[11:20,11:20] <- rnorm(100,3,0.3)
res<-biclust(test, method=BCPlaid())
res

#microarray matrix
data(BicatYeast)
res<-biclust(BicatYeast, method=BCPlaid(), verbose=FALSE)
res
```

BCQuest *The Questmotif Bicluster algorithm*

Description

Performs Questmotif Biclustering a Bicluster algorithm for questionairs based on the framework by Murali and Kasif (2003). Searches subgroups of questionairs with same or similar answer to some questions.

Usage

```
## S4 method for signature 'matrix,BCQuest'
biclust(x, method=BCQuest(), ns=10, nd=10, sd=5, alpha=0.05, number=100)
## S4 method for signature 'matrix,BCQuestord'
biclust(x, method=BCQuestord(), d=1, ns=10, nd=10, sd=5, alpha=0.05, number=100)
## S4 method for signature 'matrix,BCQuestmet'
biclust(x, method=BCQuestmet(), quant=0.25, vari=1, ns=10, nd=10, sd=5,
  alpha=0.05, number=100)
```

Arguments

x	Data Matrix.
method	Here BCQuest, to perform Questmotif algorithm
ns	Number of questions choosen.
nd	Number of repetitions.
sd	Sample size in repetitions.
alpha	Scaling factor for column result.
number	Number of bicluster to be found.
d	Half margin of intervall question values should be in (Intervall is mean-d,mean+d).
quant	Which quantile to use on metric data
vari	Which varianz to use for metric data

Value

Returns an object of class Biclust.

Extends

Class "BiclustMethod", directly.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

References

Murali, T. & Kasif, S. Extracting Conserved Gene Expression Motifs from Gene Expression Data Pacific Symposium on Biocomputing, sullivan.bu.edu, 2003, 8, 77-88

See Also

biclust, Biclust

BCSpectral *The Spectral Bicluster algorithm*

Description

Performs Spectral Biclustering as described in Kluger et al., 2003. Spectral biclustering supposes that normalized microarray data matrices have a checkerboard structure that can be discovered by the use of svd decomposition in eigenvectors, applied to genes (rows) and conditions (columns).

Usage

```
## S4 method for signature 'matrix,BCSpectral'
biclust(x, method=BCSpectral(), normalization="log", numberOfEigenvalues=6,
minr=2, minc=2, withinVar=1, n_clusters = NULL, n_best = 3)
```

Arguments

x	The data matrix where biclusters are to be found
method	Here BCSpectral, to perform Spectral algorithm
normalization	Normalization method to apply to mat. Three methods are allowed as described by Kluger et al.: "log" (Logarithmic normalization), "irrc" (Independent Rescaling of Rows and Columns) and "bistochastization". If "log" normalization is used, be sure you can apply logarithm to elements in data matrix, if there are values under 1, it automatically will sum to each element in mat (1+abs(min(mat))) Default is "log", as recommended by Kluger et al.
numberOfEigenvalues	the number of eigenValues considered to find biclusters. Each row (gene) eigenVector will be combined with all column (condition) eigenVectors for the first numberOfEigenValues eigenvalues. Note that a high number could increase dramatically time performance. Usually, only the first eigenvectors are used. With "irrc" and "bistochastization" methods, first eigenvalue contains background (irrelevant) information, so it is ignored.
minr	minimum number of rows that biclusters must have. The algorithm will not consider smaller biclusters.
minc	minimum number of columns that biclusters must have. The algorithm will not consider smaller biclusters.
withinVar	maximum within variation allowed. Since spectral biclustering outputs a checkerboard structure despite of relevance of individual cells, a filtering of only relevant cells is necessary by means of this within variation threshold.
n_clusters	vector with first element the number of row clusters and second element the number of column clusters. If n_clusters = NULL, the number of clusters will be estimated.
n_best	number of eigenvectors to which the data is projected for the final clustering step, recommended values are 2 or 3.

Value

Returns an object of class `Biclust`.

Author(s)

Sami Leon <Sami_Leon@URMC.Rochester.edu>

Rodrigo Santamaria <rodri@usal.es>

References

Kluger et al., "Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions", Genome Research, 2003, vol. 13, pages 703-716

Examples

```
# Random matrix with embedded bicluster
test <- matrix(rnorm(5000),100,50)
test[11:20,11:20] <- rnorm(100,10,0.1)
image(test)

shuffled_test <- test[sample(nrow(test)), sample(ncol(test))]
image(shuffled_test)

# Without specifying the  number of row and column clusters
res1 <- spectral(shuffled_test,normalization="log", numberOfEigenvalues=6,
                 minr=2, minc=2, withinVar=1, n_clusters = NULL, n_best = 3)
res1
image(shuffled_test[order(res1@info$row_labels), order(res1@info$column_labels)])


# Specifying the  number of row and column clusters
res2 <- spectral(shuffled_test,normalization="log", numberOfEigenvalues=6,
                 minr=2, minc=2, withinVar=1, n_clusters = 2, n_best = 3)
res2
image(shuffled_test[order(res2@info$row_labels), order(res2@info$column_labels)])
```

BCXmotifs *The Xmotifs Bicluster algorithm*

Description

Performs XMotifs Biclustering based on the framework by Murali and Kasif (2003). Searches for a submatrix where each row as a similar motif through all columns. The Algorihm needs a discret matrix to perform.

Usage

```
## S4 method for signature 'matrix,BCXmotifs'
biclust(x, method=BCXmotifs(), ns=10, nd=10, sd=5, alpha=0.05, number=100)
```

Arguments

x	Data Matrix.
method	Here BCXmotifs, to perform Xmotifs algorithm
ns	Number of columns choosen.
nd	Number of repetitions.
sd	Sample size in repetitions.
alpha	Scaling factor for column result.
number	Number of bicluster to be found.

Value

Returns an object of class Biclust.

Extends

Class "BiclustMethod", directly.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

References

Murali, T. & Kasif, S. Extracting Conserved Gene Expression Motifs from Gene Expression Data Pacific Symposium on Biocomputing, sullivan.bu.edu, 2003, 8, 77-88

See Also

biclust, Biclust

Examples

```
data(BicatYeast)
x<-discretize(BicatYeast)
res <- biclust(x, method=BCXmotifs(), ns=20, nd=20, sd=5, alpha=0.01, number=10)
res
```

BicatYeast *BicAT Yeast*

Description

Microarray data matrix for 80 experiments with Saccharomyces Cerevisiae organism extracted from BicAT example data set.

Usage

```
data(BicatYeast)
```

Format

Data structure with information about the expression levels of 419 probesets over 70 conditions Row names follow Affymetrix probeset notation

Source

BicAT datasets at http://www.tik.ee.ethz.ch/sop/bicat/

biclust *The biclust Method*

Description

The function `biclust` is the main function of the package. It calculates the bicluster in a data matrix using the algorithm specified in the method-argument. Currently the package contains 5 different methods for the use in `biclust`. For each algorithm see the class help files for further details. For some algorithms preproccessing is necessary, e.g. `BCBimax` only runs with a logical matrix.

Usage

```
## S4 method for signature 'matrix,BiclustMethod'
biclust(x,method,...)
## S4 method for signature 'matrix,character'
biclust(x,method,...)
```

Arguments

`x`	Data matrix.
`method`	An object of class `"BiclustMethod"` or a character string with the name of a `"BiclustMethod"`-class.
`...`	Additional Parameters of the `"BiclustMethod"`

Value

Returns an object of class `Biclust`.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

`Biclust-class`, `BCCC`, `BCXmotifs`, `BCPlaid`, `BCSpectral`, `BCBimax`, `BCQuest`, `BiclustMethod-class`

Examples

```
test <- matrix(rbinom(400, 50, 0.4), 20, 20)
res1 <- biclust(test, method=BCCC(), delta=1.5,  alpha=1, number=10)
```

Biclust-class *The Biclust Class*

Description

Biclust is the class structure for results of a bicluster algorithm. It contains all information needed for further processing. The `show` Method gives the Name of the Algorithm used and the first Bicluster found. The `summary` Method gives sizes of all bicluster found.

Objects from the Class

Objects can be created by performing a bicluster algorithm via the `biclust()` function.

Slots

Objects of class `Biclust` have the following slots:

`Parameters`: Saves input Parameters in a list

`RowxNumber`: Logical Matrix which contains 1 in [i,j] if Row i is in Bicluster j

`NumberxCol`: Logical Matrix which contains 1 in [i,j] if Col j is in Bicluster i

`Number`: Number of Bicluster

`info`: Additional Outputs from the different bicluster algorithms

Details

`RowxNumber` and `NumberxCol` are named after the arrangement of the data they contain. The column results are transposed in order to ensure a easy processing.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

biclust, BiclustMethod-class

---

biclustbarchart *Bicluster Barchart*

Description

Draws a barchart for a Bicluster result representing the columns

Usage

```
biclustbarchart(x, Bicres, which=NULL, ...)
```

Arguments

x	The data matrix
Bicres	BiclustResult object with a bicluster result set. If this value is set to NULL, the data matrix is drawn as a heatmap, without any reordering. Default NULL.
which	If specified gives the ploting order of the columns from bottom to top
...	Additional plot options passed to barchart

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

bubbleplot for simultaneous representation of biclusters, parallelCoordinatesfor single representation of biclusters as lines of gene or condition profiles, drawHeatmapfor Heatmap representation of biclusters and biclustmember for a membership graph.

Examples

```
  set.seed(1)
  x=matrix(rnorm(900),30,30)
  x[1:5,1:5]=rnorm(25,3,0.3)
  x[11:15,11:15]=rnorm(25,-3,0.3)
  x[21:25,21:25]=rnorm(25,6,0.3)
  colnames(x)<-paste("Var.",1:30)
  bics <- biclust(x,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m
+ a + b, iter.startup = 5, iter.layer = 30,  verbose = TRUE)
  biclustbarchart(x,bics, col="#A3E0D8")
  ord<-bicorder(bics, cols=TRUE, rev=TRUE)
  biclustbarchart(x,bics,which=ord)
```

bicluster *Extract Bilcuster*

Description

Function to extract the bicluster or the row and column numbers from a given bicluster result

Usage

```
bicluster(x, BicRes, number= 1:BicRes@Number)
biclusternumber(BicRes, number= 1:BicRes@Number)
```

Arguments

x	The data matrix
BicRes	BiclustResult object
number	Which bicluster to be extracted

Value

Returns a list containing all extracted bicluster

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

writeclust,writeBiclusterResults

Examples

```
s2=matrix(rnorm(400),20,20)
s2[12:16,12:16]=rnorm(25,3,0.3)
set.seed(1)
bics <- biclust(s2,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
iter.startup = 5, iter.layer = 30,  verbose = TRUE)
bicluster(s2, bics)
biclusternumber(bics)
```

biclustmember *Bicluster Membership Graph*

Description

Draws a membership graph cluster x columns

Usage

```
biclustmember(bicResult, x, mid = T, cl_label = "", which=NA,
  main = "BiCluster Membership Graph", xlab="Cluster",
  color=diverge_hcl(101, h = c(0, 130)), ...)

clustmember(res, x, mid = T, cl_label = "", which=NA,
  main = "Cluster Membership Graph", xlab="Cluster",
  color=diverge_hcl(101, h = c(0, 130)), ...)

bicorder(bicResult, cols=TRUE, rev=FALSE)
```

Arguments

x	The data matrix
bicResult	BiclustResult object with a bicluster result set.
res	Cluster Result (is converted into a kcca object)
mid	If TRUE, shows the value of the remaining objects inside the cluster value, else shows both aside each other.
cl_label	Ticks of x-axis
which	If specified gives the ploting order of the columns from bottom to top
main	Gives the title of the plot
xlab	Label of x-axis
color	Range of colors for the plot
...	Additional plot options or if neccessary option for as.kcca
cols	If TRUE orders the column by appearance in the bicluster, else orders the rows.
rev	If TRUE reverses the order

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

bubbleplot for simultaneous representation of biclusters, parallelCoordinatesfor single representation of biclusters as lines of gene or condition profiles, drawHeatmapfor Heatmap representation of biclusters and biclustbarchart for a barchart.

Examples

```
set.seed(1)
x=matrix(rnorm(900),30,30)
x[1:5,1:5]=rnorm(25,3,0.3)
x[11:15,11:15]=rnorm(25,-3,0.3)
x[21:25,21:25]=rnorm(25,6,0.3)
colnames(x)<-paste("Var.",1:30)
bics <- biclust(x,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
iter.startup = 5, iter.layer = 30,  verbose = TRUE)

biclustmember(bics,x)

ord<-bicorder(bics, cols=TRUE, rev=TRUE)

biclustmember(bics,x,which=ord)
```

BiclustMethod-class *The BiclustMethod Virtual Class*

Description

BiclustMethod is the virtual class structure for algorithms provided in the package. In order to use the `biclust()` function a algorithm has to have a class inherit from here.

Algorithms

Currently 6 classes inherit from BiclustMethod: BCCC, BCXmotifs, BCPlaid, BCSpectral, BCBimax, BCQuest

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

biclust, Biclust-class, BCCC, BCXmotifs, BCPlaid, BCSpectral, BCBimax, BCQuest, BiclustMethod-class

bimax.grid *Parameter Grid for BCBimax Biclustering*

Description

Generates a list containing parameter settings for the ensemble algorithm.

Usage

```
bimax.grid(method = "BCBimax", minr = c(10, 11), minc = c(10, 11), number = 10)
```

Arguments

method	Here BCBimax, to perform Bimax algorithm
minr	Minimum row size of resulting bicluster.
minc	Minimum column size of resulting bicluster.
number	Number of Bicluster to be found.

Value

A list containing parameter settings

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

ensemble, BCBimax

Examples

```
bimax.grid()
```

binarize *Binarize*

Description

Methods to convert a real matrix to a binary matrix.

Usage

```
binarize(x, threshold=NA)
binarizeByPercentage(x,percentage, error=0.2, gap=0.1)
densityOnes(x)
```

Arguments

x	The data matrix to be binarized.
threshold	Threshold used to binarize. Values over threshold will be set to 1, the rest to 0. If threshold is NA, median is used as threshold. Default NA.
percentage	Percentage of ones against zeros desired in the binary matrix.
error	Percentage of ones against zeros in the final matrix will be in [percentage-error, percentage+error]. Default 0.2
gap	Value used for incremental search of threshold. Default 0.1

Details

The binarize function returns a matrix binarized by input threshold, or by the median if no threshold is given.

The binarizeByPercentage function returns a matrix binarize by input percentage, given as desired density of ones against zeros.

The densityOnes function returns the percentage of ones against zeros in a logical matrix

Author(s)

Rodrigo Santamaria <rodri@usal.es>

Examples

```
data(BicatYeast)
m1=binarize(BicatYeast)
m2=binarize(BicatYeast, 0.2)
m3=binarizeByPercentage(BicatYeast, 5)
densityOnes(m3)
densityOnes(m2)
densityOnes(m1)
drawHeatmap(BicatYeast)
drawHeatmap(m1)
drawHeatmap(m2)
drawHeatmap(m3)
```

---

bubbleplot *Bubbleplot*

Description

Draws a bubble plot where each bicluster is represented as a circle (bubble). Color represents the bicluster set to which bicluster pertains (up to three bicluster sets can be represented simultaneously). Brightness represents the bicluster homogeneity (darker, less homogeneous). Size represents the size of the bicluster, as (number of genes)x(number of conditions). Location is a 2D-projection of gene and condition profiles.

Usage

```
bubbleplot(x, bicResult1, bicResult2=NULL, bicResult3=NULL, projection="mean",
  showLabels=FALSE)
```

Arguments

x	The data matrix from which biclusters were identified.
bicResult1	BiclustResult object with a bicluster result set whose biclusters will be drawn in green.
bicResult2	BiclustResult object with an optional second bicluster result set. Will be drawn in red (default NULL)
bicResult3	BiclustResult object with an optional third bicluster result set. Will be drawn in blue (default NULL)
projection	Projection algorithm used to position bubbles. Allowed projections are 'mean', 'isomds' and 'cmdscale' (default 'mean'). See details section for a broader explanation.
showLabels	If 'TRUE', puts a label over each bubble that tells the number within the corresponding bicluster result (default 'FALSE').

Details

Position of circles depend on a 2D projection of the multidimensional point formed by rows and columns present in the bicluster. For example, if we have a 3x3 matrix to analyze and we find a bicluster with rows 1 and 3 and columns 2 and 3, the corresponding multidimensional point will be p=(1,0,1,0,1,1). For this example, 'mean' projection will map the bicluster with the point x=(1+3)/2=2 and y=(2+3)/2=2,5. Other projections will take the point p and project it following the corresponding algorithms (see the corresponding help pages for details)

Note

Bubbleplot 2D-projection, as any multidimensional scaling, loses information, trying to take the main relationships and trends of n-dimensional data. Thus, locations and intersections between bubbles-biclusters are only an estimate of its similarity. This visualization should be used just as a help to understand overall behavior of biclustering methods, detect trends and outliers, etc.

Author(s)

Rodrigo Santamaria <rodri@usal.es>

See Also

drawHeatmap for single representation of biclusters inside data matrix, parallelCoordinates for single representation of biclusters as lines of gene or condition profiles, cmdscale, isomds for multidimensional scaling and plot for other point representations.

Examples

```
#Simplified yeast microarray data
## Not run:
data(BicatYeast)
set.seed(1)
bics1 <- biclust(BicatYeast,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
row.release = 0.7, col.release = 0.7,
verbose = FALSE, max.layers = 10, iter.startup = 5,
iter.layer = 30)
bubbleplot(BicatYeast,bics1, showLabels=TRUE)

loma=binarize(BicatYeast,2)
bics2=biclust(loma,BCBimax(), minr=4, minc=4, number=10)
bubbleplot(BicatYeast,bics1,bics2)

## End(Not run)
```

ChiaKaruturi *Chia and Karuturi Function*

Description

Function computing scores as described in the paper of Chia and Karuturi (2010)

Usage

```
ChiaKaruturi(x, bicResult, number)
```

Arguments

x	Data Matrix
bicResult	Biclust object from biclust package
number	Number of bicluster in the output for computing the scores

Details

The function computes row (T) and column (B) effects for a chosen bicluster. The scores for columns within bicluster have index 1, the scores for columns outside the bicluster have index 2. Ranking score is SB, stratification score is TS.

Value

Data.Frame with 6 slots: T, B scores for within and outside bicluster, SB and TS scores

Author(s)

Tatsiana KHAMIAKOVA <tatsiana.khamiakova@uhasselt.be>

References

Chia, B. K. H. and Karuturi, R. K. M. (2010) Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms. Algorithms for Molecular Biology, 5, 23.

See Also

diagnosticPlot, computeObservedFstat, diagnoseColRow

Examples

```
#---simulate dataset with 1 bicluster ---#
xmat<-matrix(rnorm(50*50,0,0.25),50,50) # background noise only
rowSize <- 20 #number of rows in a bicluster
colSize <- 10 #number of columns in a bicluster
a1<-rnorm(rowSize,1,0.1) #sample row effect from N(0,0.1) #adding a coherent values bicluster:
b1<-rnorm((colSize),2,0.25)  #sample column effect from N(0,0.05)
mu<-0.01 #constant value signal
 for ( i in 1 : rowSize){
  for(j in 1: (colSize)){
   xmat[i,j] <- xmat[i,j] + mu + a1[i] + b1[j]
  }
 }
 #--obtain a bicluster by running an algorithm---#
plaidmab <- biclust(x=xmat, method=BCPlaid(), cluster="b", fit.model = y ~ m + a+ b,
background = TRUE, row.release = 0.6, col.release = 0.7, shuffle = 50, back.fit = 5,
max.layers = 1, iter.startup = 100, iter.layer = 100, verbose = TRUE)

#Get Chia and Karuturi scores:
ChiaKaruturi(x=xmat, bicResult = plaidmab, number = 1)
```

---

coherence *Coherence measures*

Description

Different preliminary measures of how much constant or (additive, multiplicative, sign) coherent a bicluster is, following Madeira and Oliveira classification of biclusters.

Usage

```
constantVariance(x, resultSet, number, dimension="both")
additiveVariance(x, resultSet, number, dimension="both")
multiplicativeVariance(x, resultSet, number, dimension="both")
signVariance(x, resultSet, number, dimension="both")
```

Arguments

x	The data matrix from which biclusters were identified
resultSet	BiclustResult object with a bicluster result set where is the bicluster to measure
number	Number of the bicluster withing the result set
dimension	"both" for determining overall variance, "row" for gene variance and "col" for column variance. Default "both"

Details

Returns the corresponding variance of genes or conditions as the average of the sum of euclidean distances between all rows and/or columns of the bicluster. For additive, multiplicative and sign variance first a transformation of the bicluster is done, so variance is computed on a matrix that reflects difference, rest or change of sign between rows, columns or both.

The lower the value returned, the more constant or coherent the bicluster is. If the value returned is 0, the bicluster is ideally constant or coherent. Usually, a value above 1-1.5 is enough to determine the bicluster is not constant or coherent.

Note

There are preliminary measures for coherence. Since transformations are different, measures are not normalized and comparison between, for example, additive and multiplicative variance is not meaningful. Only comparisons between different measures of the same kind of variance are reliable by now.

Author(s)

Rodrigo Santamaria <rodri@usal.es>

Examples

```
#Simplified yeast microarray data
data(BicatYeast)
set.seed(1)
bics1 <- biclust(BicatYeast,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
row.release = 0.7, col.release = 0.7,
verbose = FALSE, max.layers = 10, iter.startup = 5,
iter.layer = 30)

constantVariance(BicatYeast, bics1,1,"row")
constantVariance(BicatYeast, bics1,1,"col")
constantVariance(BicatYeast, bics1,1,"both")
additiveVariance(BicatYeast, bics1,1,"both")
multiplicativeVariance(BicatYeast, bics1,1,"both")
signVariance(BicatYeast, bics1,1,"both")
```

computeObservedFstat *Diagnostic F Statistic Calculation*

Description

Functions for obtaining F statistics within bicluster and the significance levels. The main effects considered are row, column and interaction effect.

Usage

```
computeObservedFstat(x, bicResult, number)
```

Arguments

x	Data Matrix
bicResult	Biclust object from biclust package
number	Number of bicluster in the output for computing observed statistics

Details

F-statistics are calculated from the two-way ANOVA mode with row anc column effect. The full model with interaction is unidentifiable, thus, Tukey's test for non-additivity is used to detect an interaction within a bicluster. p-values are obtained from assymptotic F distributions.

Value

Data frame with three rows ("Row Effect", "Column Effect", "Tukey test") and 2 columns for corresponding statistics (Fstat) and their p-values (PValue). 2

Author(s)

Tatsiana KHAMIAKOVA <tatsiana.khamiakova@uhasselt.be>

See Also

diagnosticTest, diagnosticPlot2, ChiaKaruturi, diagnoseColRow

Examples

```
#---simulate dataset with 1 bicluster ---#
xmat<-matrix(rnorm(50*50,0,0.25),50,50) # background noise only
rowSize <- 20 #number of rows in a bicluster
colSize <- 10 #number of columns in a bicluster
a1<-rnorm(rowSize,1,0.1) #sample row effect from N(0,0.1) #adding a coherent values bicluster:
b1<-rnorm((colSize),2,0.25)  #sample column effect from N(0,0.05)
mu<-0.01 #constant value signal
 for ( i in 1 : rowSize){
  for(j in 1: (colSize)){
```

```
   xmat[i,j] <- xmat[i,j] + mu + a1[i] + b1[j]
  }
 }
 #--obtain a bicluster by running an algorithm---#
plaidmab <- biclust(x=xmat, method=BCPlaid(), cluster="b", fit.model = y ~ m + a+ b,
background = TRUE, row.release = 0.6, col.release = 0.7, shuffle = 50, back.fit = 5,
max.layers = 1, iter.startup = 100, iter.layer = 100, verbose = TRUE)

#Calculate statistics and their p-values to infer about the structure within bicluster:
Structure <- computeObservedFstat(x=xmat, bicResult = plaidmab, number = 1)
```

`diagnoseColRow` *Bootstrap Procedure for Bicluster Diagnostics*

Description

Calculate the significance of the discovered patter in the data based on the bootstrapping procedure.

Usage

```
diagnoseColRow(x, bicResult, number, nResamplings, replace = TRUE)
```

Arguments

`x`	data matrix, which `biclust` function was applied to
`bicResult`	object of class `biclust`, containing result of a biclustering algorithm
`number`	number of bicluster from the output for the diagnostics
`nResamplings`	number of bootstrap replicates
`replace`	logical flag for bootstrap (TRUE), or sampling without replacement (FALSE)

Details

The function computes observed F statistics for row and column effect based on two-way ANOVA model. Bootstrap procedure is used to evaluate the significance of discovered bicluster. Based on `nResamplings` replicates, the disribution of F statistics for row and column effects are obtained. The p-value is computed as

$$P(A) = \frac{\#\left\{F^*(A)_b > F(A)^{obs}\right\}}{nResamplings + 1}$$

Low p-values denote non-random selection of columns for a given bicluster. Large p-values show that in other columns for a given set of genes in the bicluster structure is similar. Hence, bicluster columns were just randomly picked by an algorithm for a set of co-regulated genes.

Value

`bootstrapFstats`
: matrix with two columns, containing values of bootstrap F-statistics. The first column corresponds to row, the second column corresponds to column.

`observedFstatRow`
: observed F-statistics for the row effect

`observedFstatCol`
: observed F-statistics for the column effect

`bootstrapPvalueRow`
: bootstrap p value for row effect

`bootstrapPvalueCol`
: bootstrap p value for column effect

Author(s)

Tatsiana KHAMIAKOVA <tatsiana.khamiakova@uhasselt.be>

See Also

diagnosticTest, diagnosticPlot2, diagnosticPlot, computeObservedFstat, ChiaKaruturi

Examples

```
#---simulate dataset with 1 bicluster ---#
xmat<-matrix(rnorm(50*50,0,0.25),50,50) # background noise only
rowSize <- 20 #number of rows in a bicluster
colSize <- 10 #number of columns in a bicluster
a1<-rnorm(rowSize,1,0.1) #sample row effect from N(0,0.1) #adding a coherent values bicluster:
b1<-rnorm((colSize),2,0.25)  #sample column effect from N(0,0.05)
mu<-0.01 #constant value signal
 for ( i in 1 : rowSize){
  for(j in 1: (colSize)){
   xmat[i,j] <- xmat[i,j] + mu + a1[i] + b1[j]
  }
 }
 #--obtain a bicluster by running an algorithm---#
plaidmab <- biclust(x=xmat, method=BCPlaid(), cluster="b", fit.model = y ~ m + a+ b,
background = TRUE, row.release = 0.6, col.release = 0.7, shuffle = 50, back.fit = 5,
max.layers = 1, iter.startup = 100, iter.layer = 100, verbose = TRUE)

#Run boosotrap procedure:
Bootstrap <- diagnoseColRow(x=xmat, bicResult = plaidmab, number = 1, nResamplings = 999,
  replace = TRUE)
diagnosticPlot(bootstrapOutput = Bootstrap)  # plotting distribution of bootstrap replicates
```

diagnosticPlot *Diagnostic F Statistics Visualization*

Description

Plots distributions of bootstrap replicates of F-statistics for row and column effect and highlights the observed statistics

Usage

```
diagnosticPlot(bootstrapOutput)
```

Arguments

bootstrapOutput
: output of diagnoseColRow function, containing bootstrap replicates and observed F-statistics

Value

No value is returned. The plot is constructed in a current device.

Author(s)

Tatsiana KHAMIAKOVA <tatsiana.khamiakova@uhasselt.be>

See Also

diagnoseColRow, computeObservedFstat

Examples

```
#---simulate dataset with 1 bicluster ---#
xmat<-matrix(rnorm(50*50,0,0.25),50,50) # background noise only
rowSize <- 20 #number of rows in a bicluster
colSize <- 10 #number of columns in a bicluster
a1<-rnorm(rowSize,1,0.1) #sample row effect from N(0,0.1) #adding a coherent values bicluster:
b1<-rnorm((colSize),2,0.25)  #sample column effect from N(0,0.05)
mu<-0.01 #constant value signal
 for ( i in 1 : rowSize){
  for(j in 1: (colSize)){
   xmat[i,j] <- xmat[i,j] + mu + a1[i] + b1[j]
  }
 }
 #--obtain a bicluster by running an algorithm---#
plaidmab <- biclust(x=xmat, method=BCPlaid(), cluster="b", fit.model = y ~ m + a+ b,
background = TRUE, row.release = 0.6, col.release = 0.7, shuffle = 50, back.fit = 5,
max.layers = 1, iter.startup = 100, iter.layer = 100, verbose = TRUE)
```

```
#Run bootsrap procedure:
Bootstrap <- diagnoseColRow(x=xmat, bicResult = plaidmab, number = 1,
  nResamplings = 999, replace = TRUE)

# plotting distribution of bootstrap replicates
diagnosticPlot(bootstrapOutput = Bootstrap)
```

`diagnosticPlot2` *Diagnostics F Statistiics Visualization*

Description

Plots distributions of bootstrap replicates of F-statistics for row, column and multiplicative effects obtained from `diagnosticTest` (when `save_F=TRUE`). Contains an option to highlight the observed statistics.

Usage

```
diagnosticPlot2(diagnosticTest, number = 1, StatVal = TRUE,
  binwidth = NULL)
```

Arguments

`diagnosticTest`	output of `diagnosticTest` with `save_F=TRUE` which contains the F-statistics and sampling replicates.
`number`	Number of which BC to plot. This needs to be one of the Biclusters requested in in `diagnosticTest`.
`StatVal`	Boolean value to draw the observed statistic on the distribution plots.
`binwidth`	The width of the bins.

Value

Returns a `ggplot` object.

Author(s)

Ewoud De Troyer

Examples

```
## Not run:
#Random matrix with embedded bicluster (with multiplicative effect)
test <- matrix(rnorm(5000),100,50)
roweff <- sample(1:5,10,replace=TRUE)
coleff <- sample(1:5,10,replace=TRUE)
```

```
test[11:20,11:20] <- test[11:20,11:20] +
  matrix(coleff,nrow=10,ncol=10,byrow=TRUE) +
  matrix(roweff,nrow=10,ncol=10) +
  roweff %*% t(coleff)


#Apply Plaid Biclustering
res <- biclust(test, method=BCPlaid())

#Apply default diagnosticTest
out <- diagnosticTest(BCresult=res, data=test, save_F=TRUE, number=1,
                   statistics=c("F","Tukey","ModTukey","Tusell","Mandel","LBI","JandG"),
                      samplingtypes=c("Permutation","SemiparPerm","SemiparBoot",
                      "PermutationCor","SamplingCor","NormSim"))

#Plot Distributions
diagnosticPlot2(out,number=1)

## End(Not run)
```

`diagnosticTest` *Testing Procedure for Bicluster Diagnostics*

Description

Calculate the statistical value of the row, column and multiplicative effect based on discovered biclusters in the data. Additionally multiple sampling methods are available to compute the statistical significance through p-values.

Usage

```
diagnosticTest(BCresult, data, number = 1:BCresult@Number, verbose = TRUE,
  statistics = c("F", "Tukey"), sampling = TRUE, samplingtypes = NULL,
  nSim = 1000, alpha = 0.05, save_F = FALSE)
```

Arguments

`BCresult`	An object of class `biclust` containing the result of a biclustering algorithm
`data`	data matrix, which `biclust` function was applied to
`number`	Vector of bicluster numbers of which the diagnostics should be calculated. (default = all available biclusters)
`verbose`	Boolean value to print progression of computed statistics.
`statistics`	Vector select which statistics to compute. (default = `c("F","Tukey")`) • `"F"` (Row and column F statistics of two-way ANOVA with one replicate for cell) • `"Tukey"` (Tukey's test for non-additivity)

- "ModTukey" (mtukey.test)
- "Tusell" (tusell.test)
- "Mandel" (mandel.test)
- "LBI" (lbi.test)
- "JandG" (johnson.graybill.test)

sampling — Boolean value to apply sampling methods to compute statistical significance (default=TRUE). If FALSE only the "Theoretical" p-values are computed. If TRUE, both the "Theoretical" and samplingtypes p-values are computed.

samplingtypes — Vector of sampling methods for sampling=TRUE. (default=NULL=c("Permutation","SemiparPerm"))

- "Permutation"
- "SemiparPerm"
- "SemiparBoot"
- "PermutationCor"
- "SamplingCor"
- "NormSim"

See Details for more info.

nSim — Number of permutations/bootstraps.

alpha — Significance level (default=0.05)

save_F — Option to save the permuted/bootstraped statistics. This is necessary for diagnosticPlot2

Details

Due to the uncertainty of discovering the true bicluster(s) in the data, it's often advisable to not rely on the theoretical p-values but instead retrieve the p-values through a sampling procedure.

Available p-values/sampling types for each statistical method:

- "F": "Theoretical" and "Permutation" for both row and column effect.
- "Tukey": "Theoretical", "SemiparPerm" and "SemiparBoot".
- "ModTukey": "Theoretical", "SemiparPerm", "SemiparBoot", "PermutationCor" and "SamplingCor".
- "Tusell": "SemiparPerm", "SemiparBoot" and "NormSim".
- "Mandel": "Theoretical", "SemiparPerm" and "SemiparBoot".
- "LBI": "SemiparPerm", "SemiparBoot" and "NormSim".
- "JandG": "SemiparPerm", "SemiparBoot" and "NormSim".

More info on the sampling types can be found in the secion below. If available, the "Theoretical" will always be computed. By default when sampling=TRUE, a sampling method without replacement is chosen, namely "Permutation" and "SemiparPerm".

When save_F=TRUE, the null distributions of the statistics can be visualised with diagnosticPlot2.

Disclaimer: While their functionality did not change, some functions of the **additivityTests** package were altered in order to be able to return the permuted/bootstrapped statistics and p-values.

Value

Returns a list with `length(number)` elements. Each element corresponds with the requested biclusters and is a list containing:

- `table`: a data frame where each row is `statistics` and `samplingtypes` (including Theoretical) combination. The data frame contains the `Method`, `Type` (p-value type), `StatVal` (statistical value), `CritVal` (critical value), `pVal` and `Sign` (0/1 significance indicator based on `alpha`).
- `save_F`: if `save_F=TRUE`, a (`nSim` x number of permuted/bootstrapped p-values) matrix contained the sampled statistics.

Sampling Types

For each sampling type a permuted/bootstrapped BC is created as following:

- `"Permutation"`: Sample a BC from the entire dataset with replacement.
- `"SemiparPerm"`: A semi-parametric permutation procedure. Two-way ANOVA is applied on the original BC and the residual matrix extracted. A new residual matrix is created by sampling *without replacement* from the original residual matrix. The sampled BC is then generated by adding this sampled residual matrix on top the mean, row and column effect of the ANOVA procedure of the original BC.
- `"SemiparBoot"`: A semi-parametric bootstrapping procedure. Two-way ANOVA is applied on the original BC and the residual matrix extracted. A new residual matrix is created by sampling *with replacement* from the original residual matrix. The sampled BC is then generated by adding this sampled residual matrix on top the mean, row and column effect of the ANOVA procedure of the original BC.
- `"PermutationCor"`: See `correction=1` parameter of `mtukey.test`. More info in Simecek and Simeckova (2012).
- `"SamplingCor"`: See `correction=2` parameter of `mtukey.test`. More info in Simecek and Simeckova (2012).
- `"NormSim"`: Sample a BC from a standard normal distribution. This sampling procedure is used for some methods in the **additivityTests** package.

Author(s)

Ewoud De Troyer

References

Tukey, J.W.: One Degree of Freedom for Non-additivity, *Biometrics* **5**, pp. 232-242, 1949.

Simecek, Petr, and Simeckova, Marie. "Modification of Tukey's additivity test." *Journal of Statistical Planning and Inference*, **2012**.

Examples

```
## Not run:
#Random matrix with embedded bicluster (with multiplicative effect)
test <- matrix(rnorm(5000),100,50)
roweff <- sample(1:5,10,replace=TRUE)
coleff <- sample(1:5,10,replace=TRUE)
test[11:20,11:20] <- test[11:20,11:20] +
  matrix(coleff,nrow=10,ncol=10,byrow=TRUE) +
  matrix(roweff,nrow=10,ncol=10) +
  roweff %*% t(coleff)


#Apply Plaid Biclustering
res <- biclust(test, method=BCPlaid())

#Apply default diagnosticTest
out <- diagnosticTest(BCresult=res, data=test, save_F=TRUE, number=1,
                   statistics=c("F","Tukey","ModTukey","Tusell","Mandel","LBI","JandG"),
                      samplingtypes=c("Permutation","SemiparPerm","SemiparBoot",
                      "PermutationCor","SamplingCor","NormSim"))

out[[1]]$table

## End(Not run)
```

`discretize` *Create a discret matrix*

Description

Some biclusteralgorithms need a discret matrix to perform well. This function delivers a discret matrix with either a given number of levels of equally spaced intervals from minimum to maximum, or levels of same size using the quantiles.

Usage

```
discretize(x,nof=10,quant=FALSE)
```

Arguments

`x`	The data matrix from which should be dicretized
`nof`	Number of levels
`quant`	If TRUE using the quantiles, else using equally spaced levels

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

Examples

```
#Discretize yeast microarray data
data(BicatYeast)
discretize(BicatYeast[1:10,1:10])
```

---

drawHeatmap *Draw Heatmap*

Description

Draws a microarray data matrix as a heatmap, with rows and colums reordered so the rows and columns of the input bicluster will be at top-left of the matrix.

Usage

```
drawHeatmap(x,bicResult=NULL,number=NA,local=TRUE, beamercolor=FALSE,paleta,...)
drawHeatmap2(x,bicResult=NULL,number=NA,plotAll=FALSE)
```

Arguments

x	The data matrix where the bicluster is to be drawn.
bicResult	BiclustResult object with a bicluster result set. If this value is set to NULL, the data matrix is drawn as a heatmap, without any reordering. Default NULL.
number	Bicluster to be drawn from the result set 'bicResult'. If bicResult is set to NULL, this value is ignored. Default NA
local	If TRUE, only rows and columns of the bicluster were drawn.
plotAll	If TRUE, all Bicluster of result set 'bicResult' were drawn.
beamercolor	If TRUE, palete colors are used.
paleta	Colors
...	Additional plot options

Details

'plotAll' only works if there is a exclusive rows and column Result!

Author(s)

Rodrigo Santamaria <rodri@usal.es>, Sebastian Kaiser

See Also

bubbleplot for simultaneous representation of biclusters.\ parallelCoordinatesfor single representation of biclusters as lines of gene or condition profiles.

Examples

```
#Random 100x50 matrix with a single, up-regulated 10x10 bicluster
s2=matrix(rnorm(5000),100,50)
s2[11:20,11:20]=rnorm(100,3,0.3)
set.seed(1)
bics <- biclust(s2,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
iter.startup = 5, iter.layer = 30,  verbose = TRUE)
drawHeatmap(s2,bics,1)
```

EisenYeast *Eisen Yeast*

Description

Microarray data matrix for 80 experiments with Saccharomyces Cerevisiae organism by Eisen Lab.

Usage

```
data(EisenYeast)
```

Format

Data frame with information about the expression levels of 6221 genes over 80 conditions. Missing values have been imputed using k-nearest neighbor averaging implemented in impute.knn() from library 'impute' (using default k=10). Gene names follow ORF (Open Reading Format) notation.

Source

Eisen Lab at http://rana.lbl.gov/EisenData.htm

---

ensemble *Ensemble Methods for Bicluster Algorithms*

Description

Calculates an ensemble of biclusters from different parameter setting of possible different bicluster algorithms.

Usage

```
ensemble(x, confs, rep = 1, maxNum = 5, similar = jaccard2, thr = 0.8, simthr =0.7,
  subs = c(1, 1), bootstrap = FALSE, support = 0, combine=firstcome, ...)
```

Arguments

x	Data Matrix
confs	Matrix containing parameter sets
rep	Number of repetitions for each parameter set
maxNum	Maximum number of biclusters taken from each run
similar	Function to produce a similarity matrix of bicluster
thr	Threshold for similarity
simthr	Proportion of row column combinations in bicluster
subs	Vector of proportion of rows and columns for subsampling. Default c(1,1) means no subsampling.
bootstrap	Should bootstrap sampling be used (logical: replace=bootstrap).
support	Which proportion of the runs must contain the bicluster to have enough support to report it (between 0 and 1).
combine	Function to combine the single bicluster only firstcome and hcl for hierarchical clustering are possible at the moment.
...	Arguments past to the combine function.

Details

Two different kinds (or both combined) of ensembling is possible. Ensemble of repeated runs or ensemble of runs on subsamples.

Value

Return an object of class Biclust

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

Biclust-class, plaid.grid, bimax.grid

Examples

```
## Not run:
data(BicatYeast)
ensemble.plaid <- ensemble(BicatYeast,plaid.grid()[1:5],rep=1,maxNum=2, thr=0.5, subs = c(1,1))
ensemble.plaid
x <- binarize(BicatYeast)
ensemble.bimax <- ensemble(x,bimax.grid(),rep=10,maxNum=2,thr=0.5, subs = c(0.8,0.8))
ensemble.bimax

## End(Not run)
```

heatmapBC *Overlapping Heatmap*

Description

Other than drawHeatmap this function plots all or a chosen number of bicluster in one plot even if they were overlapping.

Usage

```
heatmapBC(x, bicResult, number = 0, local = TRUE, order = FALSE,
          outside = FALSE, ...)
```

Arguments

x	The data matrix where the bicluster is to be drawn.
bicResult	BiclustResult object with a bicluster result set.
number	Number of bicluster to be drawn from the result set 'bicResult'. If the default 0 is chosen all bicluster of the bicResult are drawn.
local	If TRUE, only rows and columns of the bicluster are drawn. This argument is only used if number is not set to 0.
order	If TRUE, rows and columns are ordered by their values.
outside	If TRUE, Boxes are drawn for overlapping
...	Additional plot options

Details

Overlap plotting only works for two neighbor bicluster defined by the order in the number slot.

Author(s)

Sebastian Kaiser

See Also

drawHeatmap,parallelCoordinates

Examples

```
set.seed(1234)
data(BicatYeast)
resplaid <- biclust(BicatYeast, BCPlaid(), verbose = FALSE)
heatmapBC(x = BicatYeast, bicResult = resplaid)
```

isoverlapp *Is Bicresult overlapping?*

Description

Checks if Biclusterresult includes overlapping rows or columns

Usage

```
isoverlapp(bicResult)
```

Arguments

bicResult	Result of biclust function

Value

Overlapping	Is there overlapping
Max.bicluster.Rows	Maximal number of bicluster a single row is in
Max.bicluster.Cols	Maximal number of bicluster a single col is in

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

drawHeatmap

jaccardind *Jaccardind*

Description

An adaption of the Jaccard Index for clustering is calculated.

Usage

```
jaccardind(bicres1,bicres2)
jaccard2(Rows, Cols)
```

Arguments

bicres1	A object of class Biclust
bicres2	A object of class Biclust
Rows	Matrix containing rows of biclusters
Cols	Matrix containing cols of biclusters

Details

The function calculates the percentage of datapoints in the same bicluster structure from all datapoints at least included in one bicluster.

Value

jaccardind calculates the Jaccard index jaccard2 returns a similarity matrix containing the Jaccard index between all biclusters (upper triangle matrix)

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

Examples

```
## Not run:
data(BicatYeast)
res1<-biclust(BicatYeast, method=BCPlaid(), back.fit = 2, shuffle = 3,
  fit.model = ~m + a + b,iter.startup = 5, iter.layer = 30,  verbose = TRUE)
res2<-biclust(BicatYeast, method=BCCC())
jaccardind(res1,res2)


## End(Not run)
```

---

parallelCoordinates *Parallel Coordinates*

Description

Represents expression levels through gene and/or condition profiles in a bicluster as lines.

Usage

```
parallelCoordinates(x, bicResult, number, plotBoth = FALSE, plotcol = TRUE,
compare = TRUE, info = F, bothlab = c("Rows", "Columns"), order = FALSE,
order2 = 0,ylab = "Value" , col=1,...)
```

Arguments

x	The data matrix of the bicluster to be drawn
bicResult	BiclustResult object with a bicluster result set
number	Bicluster to be drawn from the result set 'bicResult'
plotBoth	If 'TRUE', Parallel Coordinates of rows (Genes) and columns (Conditions) were drawn one below the other.
plotcol	If 'TRUE', columns profiles are drawn, so each line represents one of the columns in the bicluster. Otherwise, row profiles are drawn. Default 'TRUE'
compare	If 'TRUE', values of the complete data matrix are considered and drawn as shaded lines. Default 'TRUE'
info	If 'TRUE', a prepared Title is drawn
bothlab	Names of the x Axis if PlotBoth
order	Rows and/or Columns are in increasing order.
order2	Which ordering.
ylab	ylab
col	col
...	Plot Parameters

Author(s)

Rodrigo Santamaria, Martin Sill and Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

drawHeatmap for alternative representation of biclusters and bubbleplot for simultaneous representation of biclusters.

Examples

```
  #Random 100x50 matrix with a single, up-regulated 10x10 bicluster
  s2=matrix(rnorm(5000),100,50)
  s2[11:20,11:20]=rnorm(100,3,0.3)
  set.seed(1)
  bics <- biclust(s2,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
  iter.startup = 5, iter.layer = 30,  verbose = TRUE)
  parallelCoordinates(x=s2,bicResult=bics,number=1, plotBoth=TRUE,
plotcol=TRUE, compare=TRUE, info=TRUE,bothlab=c("Genes Bicluster
1","Conditions Bicluster 1"), order =TRUE)
  parallelCoordinates(x=s2,bicResult=bics,number=1, plotBoth=FALSE, plotcol=TRUE,
    compare=FALSE, info=TRUE)
```

plaid.grid *Parameter Grid for BCPlaid Biclustering*

Description

Generates a list containing parameter settings for the ensemble algorithm.

Usage

```
plaid.grid(method = "BCPlaid", cluster = "b", fit.model = y ~ m + a + b,
  background = TRUE, background.layer = NA, background.df = 1,
  row.release = c(0.5, 0.6, 0.7), col.release = c(0.5, 0.6, 0.7),
  shuffle = 3, back.fit = 0, max.layers = 20, iter.startup = 5,
  iter.layer = 10, verbose = FALSE)
```

Arguments

method	Here BCPlaid, to perform Plaid algorithm
cluster	'r', 'c' or 'b', to cluster rows, columns or both (default 'b')
fit.model	Model (formula) to fit each layer. Usually, a linear model is used, that estimates three parameters: m (constant for all elements in the bicluster), a(contant for all rows in the bicluster) and b (constant for all columns). Thus, default is: y ~ m + a + b.
background	If 'TRUE' the method will consider that a background layer (constant for all rows and columns) is present in the data matrix.
background.layer	If background='TRUE' a own background layer (Matrix with dimension of x) can be specified.
background.df	Degrees of Freedom of backround layer if background.layer is specified.
shuffle	Before a layer is added, it's statistical significance is compared against a number of layers obtained by random defined by this parameter. Default is 3, higher numbers could affect time performance.
iter.startup	Number of iterations to find starting values
iter.layer	Number of iterations to find each layer
back.fit	After a layer is added, additional iterations can be done to refine the fitting of the layer (default set to 0)
row.release	Scalar in [0,1](with interval recommended [0.5-0.7]) used as threshold to prune rows in the layers depending on row homogeneity
col.release	As above, with columns
max.layers	Maximum number of layer to include in the model
verbose	If 'TRUE' prints extra information on progress.

Value

A list containing parameter settings

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

ensemble, BCPlaid

Examples

```
plaid.grid()
```

---

plotclust *Barplot of Bicluster*

Description

Draws a graph to compare the values inside the diffrent biclusters with the values outside the bicluster

Usage

```
plotclust(res,x,bicluster=TRUE,legende=FALSE,noC=5,wyld=3,Titel="Plotclust",...)
```

Arguments

x	The data matrix
res	BiclustResult object if bicluster=TRUE else a normal kcca object.
bicluster	If TRUE,res is treated as a BiclustResult object
legende	Draws a legend.
noC	Number of Clusters drawn
wyld	Gives the distance between plot and axis.
Titel	Gives the title of the plot.
...	Additional plot options

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

bubbleplot for simultaneous representation of biclusters. parallelCoordinatesfor single representation of biclusters as lines of gene or condition profiles. drawHeatmapfor Heatmap representation of biclusters.

Examples

```
s2=matrix(rnorm(400),20,20)
s2[12:16,12:16]=rnorm(25,3,0.3)
set.seed(1)
bics <- biclust(s2,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
iter.startup = 5, iter.layer = 30,  verbose = TRUE)
plotclust(bics,s2)
```

---

predictBimax *Predict from a BCrepBimax Result*

Description

Predicts cluster membership for new data rows given a BCrepBimax Result

Usage

```
predictBimax(BCrepBimax, x)
```

Arguments

BCrepBimax	Result of biclust function with method BCrepBimax
x	The data matrix which clustermembership should be predicted

Value

Returns a vector with clustermembership of data x of class.

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

See Also

BCrepBimax

SyntrenEcoli *SynTReN E. coli*

Description

Synthetic microarray data matrix generated by Syntren for 20 experiments using 200 genes from Transcription Regulatory Network of Shen-Orr et al. (2002).

Usage

```
data(SyntrenEcoli)
```

Format

Data structure with information about the expression levels of 200 genes over 20 conditions. Conditions are named as C1... C20

Source

SynTReN software can be downloaded at http://homes.esat.kuleuven.be/~kmarchal/SynTReN/index.html

References

Shen-Orr et al., "Network motifs in the transcriptional regulation network of Escherichia coli", Nature Genetics 2002, volume 31, pages 64-68.

Tim Van den Bulcke et al., "SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms", BMC Bioinformatics, 2006, volume 7, number 43.

writeBiclusterResults *writeBiclusterResults*

Description

Write bicluster results to a file

Usage

```
writeBiclusterResults(fileName, bicResult, bicName, geneNames, arrayNames,
  append=FALSE, delimiter=" ")
```

Arguments

fileName	Path to the file were biclusters are written.
bicResult	Biclusters results as a Biclust class.
bicName	Brief description for the biclustering algorithm used.
geneNames	Array of strings with gene (row) names in the analyzed data matrix
arrayNames	Array of strings with condition (column) names in the analyzed data matrix
append	If true, adds the bicluster results to previous information in the text file, if it exists. Default false.
delimiter	delimiter string between gene and condition names. Default " ".

Author(s)

Rodrigo Santamaria <rodri@usal.es>

Examples

```
## Not run:
  data(BicatYeast)
  res <- biclust(BicatYeast, method=BCCC(), delta=1.5,  alpha=1, number=10)
 writeBiclusterResults("results.txt", res,"CC with delta 1.5", dimnames(BicatYeast)[1][[1]],
    dimnames(BicatYeast)[2][[1]])

## End(Not run)
```

---

writeclust *Write a Bicluster as a Cluster Result*

Description

Draws a graph to compare the values inside the diffrent biclusters with the values outside the bicluster

Usage

```
writeclust(Biclusterresult,row=TRUE,noC=10)
```

Arguments

Biclusterresult	BiclustResult object
row	If TRUE, cluster of rows were written.
noC	Number of Clusters written

Author(s)

Sebastian Kaiser <sebastian.kaiser@stat.uni-muenchen.de>

Examples

```
s2=matrix(rnorm(400),20,20)
s2[12:16,12:16]=rnorm(25,3,0.3)
set.seed(1)
bics <- biclust(s2,BCPlaid(), back.fit = 2, shuffle = 3, fit.model = ~m + a + b,
iter.startup = 5, iter.layer = 30,  verbose = TRUE)
writeclust(bics)
```

Index

∗ **array**
  binarize, 17
  coherence, 21
  writeBiclusterResults, 42
∗ **bicluster**
  diagnoseColRow, 24
∗ **bootstrap**
  diagnoseColRow, 24
∗ **classes**
  Biclust-class, 12
  BiclustMethod-class, 16
∗ **classif**
  BCBimax, 3
  BCCC, 4
  BCPlaid, 5
  BCQuest, 6
  BCXmotifs, 9
  isoverlapp, 36
  predictBimax, 41
∗ **cluster**
  BCBimax, 3
  BCCC, 4
  BCPlaid, 5
  BCQuest, 6
  BCSpectral, 8
  BCXmotifs, 9
  biclust, 11
  biclustbarchart, 13
  bicluster, 14
  biclustmember, 15
  bimax.grid, 17
  bubbleplot, 18
  ChiaKaruturi, 20
  coherence, 21
  discretize, 31
  drawHeatmap, 32
  ensemble, 33
  heatmapBC, 35
  isoverlapp, 36
  jaccardind, 36
  parallelCoordinates, 37
  plaid.grid, 39
  plotclust, 40
  predictBimax, 41
  writeclust, 43
∗ **datasets**
  BicatYeast, 11
  EisenYeast, 33
  SyntrenEcoli, 42
∗ **hplot**
  biclustbarchart, 13
  biclustmember, 15
  bubbleplot, 18
  drawHeatmap, 32
  heatmapBC, 35
  parallelCoordinates, 37
  plotclust, 40
  writeclust, 43
∗ **manip**
  binarize, 17
  writeBiclusterResults, 42
∗ **models**
  BCPlaid, 5
∗ **multivariate**
  BCSpectral, 8

additiveVariance *(*coherence*)*, 21

BCBimax, 3, *12*, *16*, *17*
BCBimax-class *(*BCBimax*)*, 3
BCCC, 4, *12*, *16*
BCCC-class *(*BCCC*)*, 4
BCPlaid, 5, *12*, *16*, *40*
BCPlaid-class *(*BCPlaid*)*, 5
BCQuest, 6, *12*, *16*
BCQuest-class *(*BCQuest*)*, 6
BCQuestmet *(*BCQuest*)*, 6
BCQuestmet-class *(*BCQuest*)*, 6
BCQuestord *(*BCQuest*)*, 6

BCQuestord-class *(BCQuest)*, 6
BCrepBimax, *41*
BCrepBimax *(BCBimax)*, 3
BCrepBimax-class *(BCBimax)*, 3
BCSpectral, 8, *12*, *16*
BCSpectral-class *(BCSpectral)*, 8
BCXmotifs, 9, *12*, *16*
BCXmotifs-class *(BCXmotifs)*, 9
BicatYeast, 11
Biclust, *3, 4*, *7*, *10*
Biclust *(Biclust-class)*, 12
biclust, *3, 4*, *7*, *10*, 11, *13*, *16*
biclust,matrix,BCBimax-method *(BCBimax)*, 3
biclust,matrix,BCCC-method *(BCCC)*, 4
biclust,matrix,BCPlaid-method *(BCPlaid)*, 5
biclust,matrix,BCQuest-method *(BCQuest)*, 6
biclust,matrix,BCQuestmet-method *(BCQuest)*, 6
biclust,matrix,BCQuestord-method *(BCQuest)*, 6
biclust,matrix,BCrepBimax-method *(BCBimax)*, 3
biclust,matrix,BCSpectral-method *(BCSpectral)*, 8
biclust,matrix,BCXmotifs-method *(BCXmotifs)*, 9
biclust,matrix,BiclustMethod-method *(biclust)*, 11
biclust,matrix,character-method *(biclust)*, 11
biclust,matrix,function-method *(biclust)*, 11
Biclust-class, 12
biclust-method *(biclust)*, 11
biclustbarchart, 13, *15*
bicluster, 14
biclusternumber *(bicluster)*, 14
biclustmember, *13*, 15
BiclustMethod, *7*, *10*
BiclustMethod *(BiclustMethod-class)*, 16
BiclustMethod-class, 16
BiclustResult *(Biclust-class)*, 12
bicorder *(biclustmember)*, 15
Bimax *(BCBimax)*, 3
bimax.grid, 17, *34*
binarize, 17
binarizeByPercentage *(binarize)*, 17
bubble plot *(bubbleplot)*, 18
bubbleplot, *13*, *15*, 18, *32*, *38*, *40*

CC *(BCCC)*, 4
ChiaKaruturi, 20, *23*, *25*
clustmember *(biclustmember)*, 15
coherence, 21
computeObservedFstat, *21*, 23, *25, 26*
constantVariance *(coherence)*, 21

densityOnes *(binarize)*, 17
diagnoseColRow, *21*, *23*, 24, *26*
diagnosticPlot, *21*, *25*, 26
diagnosticPlot2, *23*, *25*, 27, *29*
diagnosticTest, *23*, *25*, *27*, 28
discretize, 31
drawHeatmap, *13*, *15*, *19*, 32, *35, 36*, *38*, *40*
drawHeatmap2 *(drawHeatmap)*, 32

E. coli microarray synthetic data matrix *(SyntrenEcoli)*, 42
EisenYeast, 33
ensemble, *17*, 33, *40*
Escherichia Coly data matrix generated by SynTReN *(SyntrenEcoli)*, 42

heatmap *(drawHeatmap)*, 32
heatmapBC, 35

isoverlapp, 36

jaccard2 *(jaccardind)*, 36
jaccardind, 36
johnson.graybill.test, *29*

Kluger biclustering *(BCSpectral)*, 8

lbi.test, *29*

mandel.test, *29*
mtukey.test, *29, 30*
multiplicativeVariance *(coherence)*, 21

parallel coordinates *(parallelCoordinates)*, 37
parallelCoordinates, *13*, *15*, *19*, *32*, *35*, 37, *40*
plaid *(BCPlaid)*, 5

plaid model (BCPlaid), 5
plaid.grid, *34*, 39
plotclust, 40
predictBimax, 41

Quest (BCQuest), 6

repBimax (BCBimax), 3

Saccharomices Cerevisiae data matrix (EisenYeast), 33
Saccharomices Cerevisiae reduced data matrix (BicatYeast), 11
show,Biclust-method (Biclust-class), 12
signVariance (coherence), 21
spectral (BCSpectral), 8
spectral biclustering (BCSpectral), 8
star graph (bubbleplot), 18
summary,Biclust-method (Biclust-class), 12
SyntrenEcoli, 42

turner biclustering (BCPlaid), 5
tusell.test, *29*

writeBiclusterResults, *14*, 42
writeBiclusters (writeBiclusterResults), 42
writeclust, *14*, 43

Xmotif (BCXmotifs), 9

yeast microarray data matrix (EisenYeast), 33
yeast microarray reduced data matrix (BicatYeast), 11