

# Package ‘biclust’

December 9, 2009

**Version** 0.9.1

**Date** 2009-12-04

**Author** Sebastian Kaiser, Rodrigo Santamaria, Martin Sill, Roberto Theron, Luis Quintales and Friedrich Leisch.

**Maintainer** Sebastian Kaiser <Sebastian.Kaiser@stat.uni-muenchen.de>

**Title** BiCluster Algorithms

**Depends** MASS, grid, vcd, colorspace

**Imports** methods

**Suggests** flexclust, isa2

**Description** The main function `biclust` provides several algorithms to find biclusters in two-dimensional data: Cheng and Church, Spectral, Plaid Model, Xmotifs and Bimax. In addition, the package provides methods for data preprocessing (normalization and discretisation), visualisation, and validation of bicluster solutions.

**License** GPL-2

**LazyLoad** yes

**Repository** CRAN

**Repository/R-Forge/Project** biclust

**Repository/R-Forge/Revision** 158

**Date/Publication** 2009-12-09 08:53:20

**R topics documented:**

BCBimax	2
BCCC	3
BCPlaid	4
BCQuest	6
BCSpectral	7
BCXmotifs	8
BicatYeast	10
biclust	10
Biclust-class	11
biclustmember	12
BiclustMethod-class	13
binarize	14
bubbleplot	15
coherence	16
discretize	18
drawHeatmap	18
EisenYeast	19
jaccardind	20
parallelCoordinates	21
plotclust	22
predictBimax	23
SyntrenEcoli	24
writeBiclusterResults	24
writeclust	25
<b>Index</b>	<b>27</b>

BCBimax

*The Bimax Bicluster algorithm***Description**

Performs Bimax Biclustering based on the framework by Prelic et. al.(2006). It searches for sub-matrices 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**

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

**Usage**

```
## S4 method for signature 'matrix,BCCC':
biclust(x, method=BCCC(), delta, 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, 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**

<code>x</code>	The data matrix where biclusters have to be found
<code>method</code>	Here BCPlaid, to perform Plaid algorithm
<code>cluster</code>	'r', 'c' or 'b', to cluster rows, columns or both (default 'b')
<code>fit.model</code>	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 (constant for all rows in the bicluster) and b (constant for all columns). Thus, default is: $y \sim m + a + b$ .
<code>background</code>	If 'TRUE' the method will consider that a background layer (constant for all rows and columns) is present in the data matrix.
<code>shuffle</code>	Before a layer is added, its 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.
<code>iter.startup</code>	Number of iterations to find starting values
<code>iter.layer</code>	Number of iterations to find each layer
<code>back.fit</code>	After a layer is added, additional iterations can be done to refine the fitting of the layer (default set to 0)
<code>row.release</code>	Scalar in [0,1](with interval recommended [0.5-0.7]) used as threshold to prune rows in the layers depending on row homogeneity
<code>col.release</code>	As above, with columns
<code>max.layers</code>	Maximum number of layer to include in the model
<code>verbose</code>	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", Stanford 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,
```

### 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](mailto: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](http://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=3,  
        minr=2, minc=2, withinVar=1)
```

**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 very 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.

**Value**

Returns an object of class `Biclust`.

**Author(s)**

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)
res1 <- biclust(test, method=BCSpectral(), numberOfEigenvalues=1)
res1
```

**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 Algorithm 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 rows 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 preprocessing 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

<code>x</code>	Data matrix.
<code>method</code>	An object of class "BiclustMethod" or a character string with the name of a "BiclustMethod"-class.
<code>...</code>	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](#)

---

biclustmember      *Draw Heatmap*

---

**Description**

Draws a membership graph cluster x columns

**Usage**

```
biclustmember(bicResult, x, mid = T, cl_label = "", which=NA, main = "BiCluster Membership Graph")
clustmember(res, x, mid = T, cl_label = "", which=NA, main = "Cluster Membership Graph")
bicorder(bicResult, cols=TRUE, rev=FALSE)
```

**Arguments**

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

**Author(s)**

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

**See Also**

[bubbleplot](#) for simultaneous representation of biclusters. [parallelCoordinates](#) for single representation of biclusters as lines of gene or condition profiles. [drawHeatmap](#) for Heatmap representation of biclusters.

**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](#)

---

`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

<code>x</code>	The data matrix to be binarized.
<code>threshold</code>	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.
<code>percentage</code>	Percentage of ones against zeros desired in the binary matrix.
<code>error</code>	Percentage of ones against zeros in the final matrix will be in [percentage-error, percentage+error]. Default 0.2
<code>gap</code>	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) $\times$ (number of conditions). Location is a 2D-projection of gene and condition profiles.

**Usage**

```
bubbleplot(x, bicResult1, bicResult2=NULL, bicResult3=NULL, projection="mean", show
```

**Arguments**

<code>x</code>	The data matrix from which biclusters were identified.
<code>bicResult1</code>	BiclustResult object with a bicluster result set whose biclusters will be drawn in green.
<code>bicResult2</code>	BiclustResult object with an optional second bicluster result set. Will be drawn in red (default NULL)
<code>bicResult3</code>	BiclustResult object with an optional third bicluster result set. Will be drawn in blue (default NULL)
<code>projection</code>	Projection algorithm used to position bubbles. Allowed projections are 'mean', 'isomds' and 'cmdscales' (default 'mean'). See details section for a broader explanation.
<code>showLabels</code>	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, [cmdscales](#), [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)
```

---

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**

<code>x</code>	The data matrix from which biclusters were identified
<code>resultSet</code>	BiclustResult object with a bicluster result set where is the bicluster to measure
<code>number</code>	Number of the bicluster withing the result set
<code>dimension</code>	"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")
```

---

discretize	<i>Create a discret matrix</i>
------------	--------------------------------

---

### Description

Some bicluster algorithms 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 discretized
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	<i>Draw Heatmap</i>
-------------	---------------------

---

### Description

Draws a microarray data matrix as a heatmap, with rows and columns 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**

<code>x</code>	The data matrix where the bicluster is to be drawn.
<code>bicResult</code>	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.
<code>number</code>	Bicluster to be drawn from the result set 'bicResult'. If bicResult is set to NULL, this value is ignored. Default NA
<code>local</code>	If TRUE, only rows and columns of the bicluster were drawn.
<code>plotAll</code>	If TRUE, all Bicluster of result set 'bicResult' were drawn.
<code>beamercolor</code>	If TRUE, palette colors are used.
<code>palette</code>	Colors
<code>...</code>	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.\ [parallelCoordinates](#) for 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)
```

**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>

---

```
jaccardind
```

```
Jaccardind
```

---

**Description**

An adaption of the Jaccard Index for clustering is calculated.

**Usage**

```
jaccardind(bicres1, bicres2)
```

**Arguments**

```
bicres1      A object of class Biclust
bicres2      A object of class Biclust
```

**Details**

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

**Author(s)**

Sebastian Kaiser <[sebastian.kaiser@stat.uni-muenchen.de](mailto: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 +
res2<-biclust(BicatYeast, method=BCCC())
jaccardindlight(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)
```

**Arguments**

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

**Author(s)**

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

**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)
parallelCoordinates(x=s2,bicResult=bics,number=1, plotBoth=FALSE, plotcol=TRUE, compare=FALSE)
```

---

plotclust

*Barplot of Bicluster*


---

**Description**

Draws a graph to compare the values inside the different 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. [parallelCoordinates](#) for single representation of biclusters as lines of gene or condition profiles. [drawHeatmap](#) for 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=F
```

**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**

```
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]])
```

---

writeclust

*Write a Bicluster as a CLuster Result*

---

**Description**

Draws a graph to compare the values inside the different 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

- \*Topic **array**
    - binarize, 14
    - coherence, 16
    - writeBiclusterResults, 24
  - \*Topic **classes**
    - Biclust-class, 11
    - BiclustMethod-class, 13
  - \*Topic **classif**
    - BCBimax, 2
    - BCCC, 3
    - BCPlaid, 4
    - BCQuest, 5
    - BCXmotifs, 8
    - predictBimax, 23
  - \*Topic **cluster**
    - BCBimax, 2
    - BCCC, 3
    - BCPlaid, 4
    - BCQuest, 5
    - BCSpectral, 6
    - BCXmotifs, 8
    - biclust, 10
    - biclustmember, 12
    - bubbleplot, 15
    - coherence, 16
    - discretize, 18
    - drawHeatmap, 18
    - jaccardind, 20
    - parallelCoordinates, 21
    - plotclust, 22
    - predictBimax, 23
    - writeclust, 25
  - \*Topic **datasets**
    - BicatYeast, 9
    - EisenYeast, 19
    - SyntrenEcoli, 24
  - \*Topic **hplot**
    - biclustmember, 12
    - bubbleplot, 15
    - drawHeatmap, 18
    - parallelCoordinates, 21
    - plotclust, 22
    - writeclust, 25
  - \*Topic **manip**
    - binarize, 14
    - writeBiclusterResults, 24
  - \*Topic **models**
    - BCPlaid, 4
  - \*Topic **multivariate**
    - BCSpectral, 6
- additiveVariance (*coherence*), 16
- BCBimax, 2, 10, 13
- BCBimax-class (*BCBimax*), 2
- BCCC, 3, 10, 13
- BCCC-class (*BCCC*), 3
- BCPlaid, 4, 10, 13
- BCPlaid-class (*BCPlaid*), 4
- BCQuest, 5, 10, 13
- BCQuest-class (*BCQuest*), 5
- BCQuestmet (*BCQuest*), 5
- BCQuestmet-class (*BCQuest*), 5
- BCQuestord (*BCQuest*), 5
- BCQuestord-class (*BCQuest*), 5
- BCrepBimax, 23
- BCrepBimax (*BCBimax*), 2
- BCrepBimax-class (*BCBimax*), 2
- BCSpectral, 6, 10, 13
- BCSpectral-class (*BCSpectral*), 6
- BCXmotifs, 8, 10, 13
- BCXmotifs-class (*BCXmotifs*), 8
- BicatYeast, 9
- Biclust, 2, 3, 6, 9
- Biclust (*Biclust-class*), 11
- biclust, 2, 3, 6, 9, 10, 11, 13
- biclust, matrix, BCBimax-method (*BCBimax*), 2

- biclust, matrix, BCCC-method  
(*BCCC*), 3
- biclust, matrix, BCPlaid-method  
(*BCPlaid*), 4
- biclust, matrix, BCQuest-method  
(*BCQuest*), 5
- biclust, matrix, BCQuestmet-method  
(*BCQuest*), 5
- biclust, matrix, BCQuestord-method  
(*BCQuest*), 5
- biclust, matrix, BCrepBimax-method  
(*BCBimax*), 2
- biclust, matrix, BCSpectral-method  
(*BCSpectral*), 6
- biclust, matrix, BCXmotifs-method  
(*BCXmotifs*), 8
- biclust, matrix, BiclustMethod-method  
(*biclust*), 10
- biclust, matrix, character-method  
(*biclust*), 10
- biclust, matrix, function-method  
(*biclust*), 10
- Biclust-class, 10, 13
- Biclust-class, 11
- biclust-method (*biclust*), 10
- biclustmember, 12
- BiclustMethod, 6, 8
- BiclustMethod  
(*BiclustMethod-class*), 13
- BiclustMethod-class, 10, 11, 13
- BiclustMethod-class, 13
- BiclustResult (*Biclust-class*), 11
- bicorder (*biclustmember*), 12
- Bimax (*BCBimax*), 2
- binarize, 14
- binarizeByPercentage (*binarize*),  
14
- bubble plot (*bubbleplot*), 15
- bubbleplot, 12, 15, 19, 21, 22
- CC (*BCCC*), 3
- clustmember (*biclustmember*), 12
- coherence, 16
- constantVariance (*coherence*), 16
- densityOnes (*binarize*), 14
- discretize, 18
- drawHeatmap, 12, 16, 18, 21, 22
- drawHeatmap2 (*drawHeatmap*), 18
- E. coli microarray synthetic  
data matrix  
(*SyntrenEcoli*), 24
- EisenYeast, 19
- Escherichia Coly data matrix  
generated by SynTReN  
(*SyntrenEcoli*), 24
- heatmap (*drawHeatmap*), 18
- jaccardind, 20
- Kluger biclustering (*BCSpectral*),  
6
- multiplicativeVariance  
(*coherence*), 16
- parallel coordinates  
(*parallelCoordinates*), 21
- parallelCoordinates, 12, 16, 19, 21, 22
- plaid (*BCPlaid*), 4
- plaid model (*BCPlaid*), 4
- plotclust, 22
- predictBimax, 23
- Quest (*BCQuest*), 5
- repBimax (*BCBimax*), 2
- Saccharomices Cerevisiae data  
matrix (*EisenYeast*), 19
- Saccharomices Cerevisiae  
reduced data matrix  
(*BicatYeast*), 9
- show, Biclust-method  
(*Biclust-class*), 11
- signVariance (*coherence*), 16
- spectral (*BCSpectral*), 6
- spectral biclustering  
(*BCSpectral*), 6
- star graph (*bubbleplot*), 15
- summary, Biclust-method  
(*Biclust-class*), 11
- SyntrenEcoli, 24
- turner biclustering (*BCPlaid*), 4
- writeBiclusterResults, 24
- writeBiclusters  
(*writeBiclusterResults*), 24

writeclust, [25](#)

Xmotif (*BCXmotifs*), [8](#)

yeast microarray data matrix  
(*EisenYeast*), [19](#)

yeast microarray reduced data  
matrix (*BicatYeast*), [9](#)