

# Package ‘biosvd’

April 22, 2016

**Type** Package

**Title** Package for high-throughput data processing, outlier detection, noise removal and dynamic modeling

**Version** 2.6.0

**Date** 2014-10-08

**Author** Anneleen Daemen <daemena@gene.com>, Matthew Brauer  
<matthejb@gene.com>

**Maintainer** Anneleen Daemen <daemena@gene.com>, Matthew Brauer  
<matthejb@gene.com>

**Description** The biosvd package contains functions to reduce the input data set from the feature x assay space to the reduced diagonalized eigenfeature x eigenassay space, with the eigenfeatures and eigenassays unique orthonormal superpositions of the features and assays, respectively. Results of SVD applied to the data can subsequently be inspected based on generated graphs, such as a heatmap of the eigenfeature x assay matrix and a bar plot with the eigenexpression fractions of all eigenfeatures. These graphs aid in deciding which eigenfeatures and eigenassays to filter out (i.e., eigenfeatures representing steady state, noise, or experimental artifacts; or when applied to the variance in the data, eigenfeatures representing steady-scale variance). After possible removal of steady state expression, steady-scale variance, noise and experimental artifacts, and after re-applying SVD to the normalized data, a summary html report of the eigensystem is generated, containing among others polar plots of the assays and features, a table with the list of features sortable according to their coordinates, radius and phase in the polar plot, and a visualization of the data sorted according to the two selected eigenfeatures and eigenassays with colored feature/assay annotation information when provided. This gives a global picture of the dynamics of expression/intensity levels, in which individual features and assays are classified in groups of similar regulation and function or similar cellular state

and biological phenotype.

**License** Artistic-2.0

**Depends** R (>= 3.1.0)

**Imports** BiocGenerics, Biobase, methods, grid, graphics, NMF

**biocViews** TimeCourse, Visualization

**NeedsCompilation** no

## R topics documented:

assayColorMap<-	3
biosvd	3
compute,Eigensystem-method	4
contrast<-	5
Eigensystem-class	6
EigensystemPlotParam-class	8
exclude,Eigensystem-method	10
featureColorMap<-	11
figure<-	12
filenames<-	13
HeLaData_exp_DoubleThym_2	13
negativeValues<-	14
palette<-	15
path<-	15
plot,Eigensystem,EigensystemPlotParam-method	16
plots<-	17
prefix<-	18
project,Eigensystem-method	19
report,Eigensystem,EigensystemPlotParam-method	20
StarvationData	21
whichAssays<-	22
whichEigenassays<-	23
whichEigenfeatures<-	23
whichFeatures<-	24
whichPolarAxes<-	25
YeastData_alpha	25

**Index**

**27**

---

assayColorMap<-            *Set assayColorMap for object of EigensystemPlotParam*

---

**Description**

Set assayColorMap for object of EigensystemPlotParam

**Usage**

```
assayColorMap(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot assayColorMap

**Details**

Set assayColorMap for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

biosvd                    *Package for high-throughput data processing, outlier detection, noise removal and dynamic modeling*

---

**Description**

Biosvd package for high-throughput data processing, outlier detection, noise removal and dynamic modeling

**Details**

This package allows for high-throughput data processing, outlier detection, noise removal and dynamic modeling, based on the framework of Singular Value Decomposition. It provides the user with summary graphs and an interactive html report.

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**Examples**

```
example( report )
```

---

```
compute,Eigensystem-method
```

*Compute the eigensystem for a feature by assay matrix*

---

**Description**

Computes the eigensystem for a feature by assay matrix by applying Singular Value Decomposition.

**Usage**

```
## S4 method for signature 'Eigensystem'
compute(object, apply = c("data", "variance"))
```

**Arguments**

object	object of class matrix, data.frame, Expressionset or Eigensystem, containing the feature x assay expression or intensity data
apply	the actual expression/intensity data (data) or variance in the data (variance) to which to apply the function (default = data)

**Details**

Function compute decomposes the input data set from the feature x assay space to the reduced diagonalized "eigenfeatures x eigenassays" space, with the eigenfeatures and eigenassays unique orthonormal superpositions of the features and assays, respectively. This approach allows filtering out eigenfeatures and eigenassays that are inferred to represent noise or experimental artifacts, either at the expression/intensity level or the variance level. The function can be applied to an object of class matrix, data.frame, ExpressionSet, or eigensystem.

**Value**

Object of class Eigensystem

**Methods**

```
signature(x = "Eigensystem")
signature(x = "data.frame")
signature(x = "matrix")
signature(x = "ExpressionSet")
```

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**References**

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci U.S.A. 97(18), 10101-10106 (2000).

**See Also**

Eigensystem-class

Other X.eigensystem.: [exclude](#), [exclude](#), [Eigensystem-method](#); [plot](#), [plot](#), [Eigensystem](#), [EigensystemPlotParam-method](#), [report](#), [report](#), [Eigensystem](#), [EigensystemPlotParam-method](#)

**Examples**

```
## Metabolomics starvation data obtained from http://genomics-pubs.princeton.edu/StarvationMetabolomics/Download
data(StarvationData)
```

```
## Computes the eigensystem for the actual expression/intensity data
eigensystem <- compute(StarvationData)
```

```
## Computes the eigensystem for the variance in the data
eigensystem <- compute(StarvationData, apply="variance")
```

---

contrast<- *Set contrast for object of EigensystemPlotParam*

---

**Description**

Set contrast for object of EigensystemPlotParam

**Usage**

```
contrast(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot contrast

**Details**

Set contrast for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

Eigensystem-class      *The Eigensystem class*

---

**Description**

Eigensystem is a list-based class for storing the results of applying Singular Value Decomposition (SVD) to a feature by assay data set. Objects are normally created by [compute](#), [Eigensystem-method](#).

**Details**

The Eigensystem class stores the original data and all SVD-derived information obtained with `compute`. Data in the Eigensystem are organized into different slots, 'matrix', 'signMatrix', 'assayMatrix', 'featureMatrix', 'eigenassays', 'eigenexpressions', 'eigenfeatures', 'assaycorrelations', 'featurecorrelations', 'fractions', 'entropy', 'apply', 'excludeEigenfeatures', and 'colorIdFeatures'. Brief descriptions of these slots are provided below.

**Slots**

Eigensystem objects contain the following slots

**matrix:** matrix containing the feature by assay data without missing values

**signMatrix:** matrix containing the sign of each element in matrix

**assayMatrix:** matrix containing additional information about assays, with rows as assays and columns as additional variables

**featureMatrix:** matrix containing additional information about features, with rows as features and columns as additional variables

**eigenassays:** matrix containing the feature by eigenassay data, with each column in eigenassays corresponding to a left singular vector, representing genome-wide expression, proteome-wide abundance or metabolome-wide intensity in the corresponding eigenassay

**eigenexpressions:** numeric vector containing the eigenexpression fraction of each eigenfeature, eigenassay-pair, constituting the diagonal elements of the diagonal matrix connecting the left and right singular values; the diagonal matrix reflects the decoupling and decorrelation of the data, with expression of each eigenfeature restricted to the corresponding eigenassay

**eigenfeatures:** matrix containing the eigenfeatures by assay data, with each row corresponding to a right singular vector, representing the expression, abundance or intensity of the corresponding eigenfeature across all assays

**assaycorrelations:** matrix containing the correlation between the eigenassays as rows and the assays as columns

**featurecorrelations:** matrix containing the correlation between the eigenfeatures as rows and features as columns

**fractions:** numeric vector containing the eigenexpression fraction for each eigenfeature, eigenassay-pair, defined as the relative fraction of overall expression that each eigenfeature and eigenassay capture

**entropy:** numeric value between 0 and 1 giving the Shannon entropy as measure for data complexity, with an entropy of 0 corresponding to an ordered and redundant data set with all expression captured by a single eigenfeature, eigenassay-pair, and an entropy of 1 corresponding to a disordered and random data set with all eigenfeature, eigenassay-pairs equally expressed

**apply:** character containing whether the eigensystem should be computed for the actual data or the variance in the data

**excludeEigenfeatures:** numeric vector containing eigenfeature 1 and 2 in case they capture >85% of the data with eigenfeature 2 capturing at least 15%, otherwise numeric value containing eigenfeature 1

**colorIdFeatures:** numeric vector or factor containing annotation information on the features

### Accessors

```
matrix(x), matrix(x) <- value
signMatrix(x), signMatrix(x) <- value
assayMatrix(x), assayMatrix(x) <- value
featureMatrix(x), featureMatrix(x) <- value
eigenassays(x), eigenassays(x) <- value
eigenexpressions(x), eigenexpressions(x) <- value
eigenfeatures(x), eigenfeatures(x) <- value
assaycorrelations(x), assaycorrelations(x) <- value
featurecorrelations(x), featurecorrelations(x) <- value
fractions(x), fractions(x) <- value
entropy(x), entropy(x) <- value
apply(x), apply(x) <- value
excludeEigenfeatures(x), excludeEigenfeatures(x) <- value
colorIdFeatures(x), colorIdFeatures(x) <- value
```

### Author(s)

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

### References

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci U.S.A. 97(18), 10101-10106 (2000).

### See Also

[compute,Eigensystem-method](#)

**Examples**

```
## Metabolomics starvation data obtained from http://genomics-pubs.princeton.edu/StarvationMetabolomics/Download
data(StarvationData)

## An object from class Eigensystem is obtained with the compute method
eigensystem <- compute(StarvationData)

## Obtain entropy
entropy(eigensystem)
```

---

EigensystemPlotParam-class

*The EigensystemPlotParam class*


---

**Description**

EigensystemPlotParam is a list-based class for storing the parameters needed to specify plot features used by `link{plot,plot-method}` in plotting Eigensystem diagnostics, projections and transformations.

**Details**

The EigensystemPlotParam class stores the list of desired plots, color palettes and keys, figure file and directory names and other necessary parameters. Data in the EigensystemPlotParam may be organized into slots: 'plots', 'palette', 'whichAssays', 'whichFeatures', 'whichEigenassays', 'whichEigenfeatures', 'whichPolarAxes', 'assayColorMap', 'featureColorMap', 'contrast', 'negativeValues', 'path', 'prefix', 'filename', 'figure', brief descriptions of which follow.

**Slots**

EigensystemPlotParam objects contain the following slots

**plots:** character vector indicating one or more plot choices from: "eigenfeatureHeatmap", "eigenassayHeatmap", "sortedHeatmap", "fraction","scree","zoomedFraction", "lines", "allLines", "eigenfeaturePolar", "eigenassayPolar". Defaults to all.

**palette:** function defining the palette to be used for heatmaps. Default is a Blue-Yellow color ramp.

**whichAssays:** numeric vector listing which assays are to be included in the plot(s). Default is all.

**whichFeatures:** numeric vector which features to include in plots. Default is all.

**whichEigenassays:** numeric vector which eigenassays to include in plots.

**whichEigenfeatures:** numeric vector which eigenfeatures to include in plots. Defaults to first four.

**whichPolarAxes:** numeric vector which two eigenassays/eigenfeatures to include in polar plots. Default is first two.



**assayColorMap:** `assayColorMap` and `featureColorMap` are optional lists of colors corresponding to the levels of these annotations for assays and features. The elements of each list are named by the annotation column they correspond to. Each element is itself a named vector of colors, named by the level of the annotation it reflects (see example).

**featureColorMap:** `list`

**contrast:** numeric value specifying the contrast to use in heatmaps.

**negativeValues:** logical indicating whether scaling of values for heatmap should result in range that includes negative values. Default is TRUE.

**path:** character specifying the path of files for figure output. Default is current working directory.

**prefix:** character specifying an optional prefix to add to filenames. Default is biosvd.

**filenames:** character optional name for the files containing the plots. Names correspond to elements of 'plots' vector.

**figure:** logical specifying whether to output plots into files. Default is FALSE.

### Accessors

```
plots(x), plots(x) <- value
palette(x), palette(x) <- value
whichAssays(x), whichAssays(x) <- value
whichFeatures(x), whichFeatures(x) <- value
whichEigenassays(x), whichEigenassays(x) <- value
whichEigenfeatures(x), whichEigenfeatures(x) <- value
whichPolarAxes(x), whichPolarAxes(x) <- value
assayColorMap(x), assayColorMap(x) <- value
featureColorMap(x), featureColorMap(x) <- value
contrast(x), contrast(x) <- value
negativeValues(x), negativeValues(x) <- value
path(x), path(x) <- value
prefix(x), prefix(x) <- value
filenames(x), filenames(x) <- value
figure(x), figure(x) <- value
```

### Author(s)

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

### References

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U.S.A.* 97(18), 10101-10106 (2000).

## Examples

```
data(YeastData_alpha)

params <- new("EigensystemPlotParam")
cellcycle.col.map <- c("orange2", "darkgreen", "blue2", "magenta2", "red2")
names(cellcycle.col.map) <- c("S", "G2", "M", "M/G1", "G1")
assayColorMap(params) <- list(Cell.cycle.stage=cellcycle.col.map)
featureColorMap(params) <- list(Cell.cycle.stage=NA)
```

---

exclude,Eigensystem-method

*Excludes specified eigenfeatures/eigenassays from the data*

---

## Description

Excludes specified eigenfeatures/eigenassays from the original data

## Usage

```
## S4 method for signature 'Eigensystem'
exclude(x, excludeEigenfeatures = NULL)
```

## Arguments

`x` object of class eigensystem  
`excludeEigenfeatures` vector of positive integers representing eigenfeatures to be excluded from the data

## Details

The function excludes eigenfeatures/eigenassays from the data that correspond to steady-state intensity, steady-scale variance, experimental artifacts and/or noise as specified by the user. In case no eigenfeatures are specified, the eigenfeature(s) corresponding to steady-state/steady-scale is/are removed. Filtering out steady-state expression/intensity corresponds to centering the expression/intensity patterns at steady-state expression/intensity level (arithmetic mean of expression/intensity  $\sim 0$ ). Filtering out steady-scale variance corresponds to normalization by the steady scale of expression/intensity variance (geometric mean of variance  $\sim 1$ ).

## Value

Object of class eigensystem

## Methods

```
signature(x = "Eigensystem")
```

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**References**

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci U.S.A. 97(18), 10101-10106 (2000).

**See Also**

Other X.eigensystem.: [compute](#), [compute,Eigensystem-method](#), [compute,ExpressionSet-method](#), [compute,data.frame-method](#), [compute,matrix-method](#); [plot,plot,Eigensystem,EigensystemPlotParam-method](#); [report,report,Eigensystem,EigensystemPlotParam-method](#)

**Examples**

```
## Metabolomics starvation data obtained from http://genomics-pubs.princeton.edu/StarvationMetabolomics/Download
data(StarvationData)

## Computes the eigensystem for the actual data
eigensystem <- compute(StarvationData)
## Excludes the eigenfeature(s) representing steady-state expression/intensity as defined in compute
exclude(eigensystem)
## Excludes user-specified eigenfeatures 1, 4 and 5
exclude(eigensystem, excludeEigenfeatures=c(1,4,5))

## Computes the eigensystem for the variance in the data
eigensystem <- compute(StarvationData, apply="variance")
## Excludes the eigenfeature(s) representing steady-scale variance as defined in compute
eigensystem <- exclude(eigensystem)
## Excludes none of the eigenfeatures and recalculates the eigensystem for the actual data
eigensystem <- exclude(eigensystem, excludeEigenfeatures=0)
```

---

featureColorMap<-      *Set featureColorMap for object of EigensystemPlotParam*

---

**Description**

Set featureColorMap for object of EigensystemPlotParam

**Usage**

```
featureColorMap(x) <- value
```

**Arguments**

x                      object of class Eigensystem  
value                    new value for slot featureColorMap

**Details**

Set featureColorMap for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

figure<- *Set figure for object of EigensystemPlotParam*

---

**Description**

Set figure for object of EigensystemPlotParam

**Usage**

```
figure(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot figure

**Details**

Set figure for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

filenames<-                    *Set filenames for object of EigensystemPlotParam*

---

**Description**

Set filenames for object of EigensystemPlotParam

**Usage**

```
filenames(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot filenames

**Details**

Set filenames for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

HeLaData\_exp\_DoubleThym\_2

*Human HeLa Cell Cycle Expression*

---

**Description**

Cells were arrested at the beginning of S phase by using a double thymidine block. Upon release from the thymidine block, cells were sampled every 1-2 hours for 44 hours during which the cells completed three cell cycles.

**Usage**

```
data(HeLaData_exp_DoubleThym_2)  
HeLaData
```

**Format**

An ExpressionSet with gene x sample expression data, gene annotation and sample information. Expression data comprise the un-logtransformed ratio of gene expression to reference mRNA from an asynchronous HeLa culture. For the samples cell cycle phase is known. For >850 genes that were identified by Whitfield et al to be periodically expressed during the cell cycle, the phase was determined based on correlation with genes known to be expressed in each cell cycle phase (e.g. cyclin E1 at the G1/S boundary, RAD51 in S phase, and TOP2A in G2).

**Author(s)**

Anneleen Daemen, Matthew Brauer

**Source**

Whitfield ML, Sherlock G, Saldanha AJ, et al. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. Mol Biol Cell 13, 1977-2000 (2002).

---

*negativeValues*<-            *Set negativeValues for object of EigensystemPlotParam*

---

**Description**

Set *negativeValues* for object of *EigensystemPlotParam*

**Usage**

```
negativeValues(x) <- value
```

**Arguments**

<i>x</i>	object of class <i>Eigensystem</i>
<i>value</i>	new value for slot <i>negativeValues</i>

**Details**

Set *negativeValues* for object of *EigensystemPlotParam*

**Value**

object of class *Eigensystem*

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

*EigensystemPlotParam*-class

---

palette<-                    *Set palette for object of EigensystemPlotParam*

---

**Description**

Set palette for object of EigensystemPlotParam

**Usage**

```
palette(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot palette

**Details**

Set palette for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

path<-                    *Set path for object of EigensystemPlotParam*

---

**Description**

Set path for object of EigensystemPlotParam

**Usage**

```
path(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot path

**Details**

Set path for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

*plot,Eigensystem,EigensystemPlotParam-method*

*Generate visualizations of the eigensystem*

---

**Description**

Generates up to ten visualizations of the eigensystem to aid in deciding which eigenfeatures and eigenassays to filter out (representing noise, steady state, steady-scale, experimental artifacts), or to aid in exploring the dynamics of expression/intensity levels over time or between different groups of assays.

**Usage**

```
## S4 method for signature 'Eigensystem,EigensystemPlotParam'
plot(x, y, ...)
```

**Arguments**

x	object of class Eigensystem
y	object of class EigensystemPlotParam
...	Additional arguments that can be passed on

**Details**

The function generates a heatmap of the eigenfeatures by assays with use of the given contrast factor (eigenfeatureHeatmap), a heatmap of the features by eigenassays with use of the given contrast factor (eigenassayHeatmap), a heatmap of the features by assays, with features sorted according to two selected eigenfeatures (sortedHeatmap), a bar plot with the eigenexpression fractions of all eigenfeatures (fraction), a screeplot for the eigenexpression fractions (scree), a bar plot with the eigenexpression fractions of the eigenfeatures without the dominant eigenfeature(s) (zoomedFraction), the intensity levels of selected eigenfeatures across the assays (by default eigenfeatures 1-4) (lines), the intensity levels of all eigenfeatures across the assays (allLines), polar plot for the assays according to their correlation with two eigenassays (eigenassayPolar), and polar plot for the features according to their correlation with two eigenfeatures (eigenfeaturePolar).



**Methods**

```
signature(x = "Eigensystem", y = "EigensystemPlotParam")
```

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**References**

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci U.S.A. 97(18), 10101-10106 (2000).

**See Also**

Other X.eigensystem.: [compute](#), [compute](#), [Eigensystem-method](#), [compute](#), [ExpressionSet-method](#), [compute](#), [data.frame-method](#), [compute](#), [matrix-method](#); [exclude](#), [exclude](#), [Eigensystem-method](#); [report](#), [report](#), [Eigensystem](#), [EigensystemPlotParam-method](#)

**Examples**

```
## Metabolomics starvation data obtained from http://genomics-pubs.princeton.edu/StarvationMetabolomics/Download
data(StarvationData)

## Computes the eigensystem for the actual data
eigensystem <- compute(StarvationData)
## Generates all provided plots for the eigensystem
params <- new("EigensystemPlotParam")
if (.Platform$OS.type != "windows") plot(eigensystem, params)
## Generates all provided plots for the eigensystem, with use of contrast 2 for the heatmap
contrast(params) <- 2
if (.Platform$OS.type != "windows") plot(eigensystem, params)
## Generates the fraction and lines plot for the eigensystem
params <- new("EigensystemPlotParam")
plots(params) <- c("fraction", "lines")
if (.Platform$OS.type != "windows") plot(eigensystem, params)
```

---

plots<-

---

*Set plots for object of EigensystemPlotParam*


---

**Description**

Set plots for object of EigensystemPlotParam

**Usage**

```
plots(x) <- value
```

**Arguments**

x                    object of class Eigensystem  
value                new value for slot plots

**Details**

Set plots for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

prefix<-

*Set prefix for object of EigensystemPlotParam*

---

**Description**

Set prefix for object of EigensystemPlotParam

**Usage**

```
prefix(x) <- value
```

**Arguments**

x                    object of class Eigensystem  
value                new value for slot prefix

**Details**

Set prefix for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

`project,Eigensystem-method`*Projects the data onto one or two eigenfeatures and eigenassays*

---

**Description**

Returns the rectangular and polar coordinates of the eigensystem projection onto one or two eigenfeatures and eigenassays

**Usage**

```
## S4 method for signature 'Eigensystem'  
project(x, axes = c(2, 1), type = "features")
```

**Arguments**

x	object of class Eigensystem
axes	numerical vector specifying eigenfeatures to project onto (default c(2,1))
type	string specifying the dimensions to return the coordinates for, either features or assays (default features)

**Details**

The function allows the sorting of the data according to one or two specified eigenfeatures and eigenassays. This gives a global picture of the dynamics of expression/intensities, in which individual features and assays are classified in groups of similar regulation and function or similar cellular state and biological phenotype.

**Value**

data.frame

**Methods**`signature(x = "Eigensystem")`**Author(s)**

Anneleen Daemen &lt;daemen.anneleen@gene.com&gt;, Matthew Brauer &lt;brauer.matthew@gene.com&gt;

**References**

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci U.S.A. 97(18), 10101-10106 (2000).

## Examples

```
## Data obtained from http://genomics-pubs.princeton.edu/StarvationMetabolomics/Download.shtm
data(StarvationData)

## Computes the eigensystem for the actual data
eigensystem <- compute(StarvationData)
## Excludes the eigenfeatures representing steady-state intensity
eigensystem <- exclude(eigensystem)

## Find the projection of the data onto eigenfeature 1 and 2
projection <- project(eigensystem)
## Project the data onto eigenfeature 3 and 4
projection <- project(eigensystem, axes=c(4,3))
```

---

```
report,Eigensystem,EigensystemPlotParam-method
Creates a report of the eigensystem
```

---

## Description

Generates a summarizing report of the eigensystem

## Usage

```
## S4 method for signature 'Eigensystem,EigensystemPlotParam'
report(x, y, ...)
```

## Arguments

x	object of class Eigensystem
y	object of class EigensystemPlotParam
...	Additional arguments that can be passed on

## Details

The function generates a txt file of the eigensystem, containing the list of features with their coordinates, radius and phase in the polar plot according to two selected eigenfeatures.

## Methods

```
signature(x = "Eigensystem", y = "EigensystemPlotParam")
```

## Author(s)

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

## See Also

Other X.eigensystem.: [compute](#), [compute](#), [Eigensystem-method](#), [compute](#), [ExpressionSet-method](#), [compute](#), [data.frame-method](#), [compute](#), [matrix-method](#); [exclude](#), [exclude](#), [Eigensystem-method](#); [plot](#), [plot](#), [Eigensystem](#), [EigensystemPlotParam-method](#)

## Examples

```
## Metabolomics starvation data obtained from http://genomics-pubs.princeton.edu/StarvationMetabolomics/Download
data(StarvationData)

## Computes the eigensystem for the actual data
eigensystem <- compute(StarvationData)
## Exclude the eigenfeatures representing steady-state intensity
eigensystem <- exclude(eigensystem)
## Computes the eigensystem on the variance in the data after filtering out steady-state intensity
eigensystem <- compute(eigensystem, apply="variance")
## No exclusion of eigenfeatures representing steady-scale variance
eigensystem <- exclude(eigensystem, excludeEigenfeatures=0)

## Generate report for eigenfeatures 1 and 2
params <- new("EigensystemPlotParam")
if (.Platform$OS.type %in% "windows") path(params) <- getwd()
report(eigensystem, params)
## Generate report for eigenfeatures 2 and 3
whichPolarAxes(params) <- c(3,2)
report(eigensystem, params)
```

---

StarvationData

*Human HeLa Cell Cycle Expression*

---

## Description

Metabolic response to starvation was studied in two microbes, *Escherichia coli* and *Saccharomyces cerevisiae*, to determine whether metabolome response to nutrient deprivation is similar across both organisms. Sixty-eight cellular metabolites were analyzed by LC-MS/MS in both bacteria and yeast, after nutrient starvation with carbon and nitrogen. Cells were sampled for 8 hours.

## Usage

```
data(StarvationData)
StarvationData
```

## Format

An ExpressionSet with metabolite x sample intensity data and sample information. Intensity data comprise the logtransformed relative metabolite concentration changes with respect to experiment initiation at time point 0 hrs. For the samples, information is provided on species (Bacteria vs Yeast), starvation (Carbon vs Nitrogen) and time point in hrs.

**Author(s)**

Anneleen Daemen, Matthew Brauer

**Source**

Brauer MJ, Yuan J, Bennett BD, et al. Conservation of the metabolomic response to starvation across two divergent microbes. Proc Nat Acad Sci U.S.A. 103(51), 19302-19307 (2006).

---

whichAssays<-                    *Set whichAssays for object of EigensystemPlotParam*

---

**Description**

Set whichAssays for object of EigensystemPlotParam

**Usage**

```
whichAssays(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot whichAssays

**Details**

Set whichAssays for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

whichEigenassays<-     *Set whichEigenassays for object of EigensystemPlotParam*

---

**Description**

Set whichEigenassays for object of EigensystemPlotParam

**Usage**

```
whichEigenassays(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot whichEigenassays

**Details**

Set whichEigenassays for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

whichEigenfeatures<-     *Set whichEigenfeatures for object of EigensystemPlotParam*

---

**Description**

Set whichEigenfeatures for object of EigensystemPlotParam

**Usage**

```
whichEigenfeatures(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot whichEigenfeatures

**Details**

Set whichEigenfeatures for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

`whichFeatures<-`            *Set whichFeatures for object of EigensystemPlotParam*

---

**Description**

Set whichFeatures for object of EigensystemPlotParam

**Usage**

```
whichFeatures(x) <- value
```

**Arguments**

<code>x</code>	object of class Eigensystem
<code>value</code>	new value for slot whichFeatures

**Details**

Set whichFeatures for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class



---

whichPolarAxes<-            *Set whichPolarAxes for object of EigensystemPlotParam*

---

**Description**

Set whichPolarAxes for object of EigensystemPlotParam

**Usage**

```
whichPolarAxes(x) <- value
```

**Arguments**

x	object of class Eigensystem
value	new value for slot whichPolarAxes

**Details**

Set whichPolarAxes for object of EigensystemPlotParam

**Value**

object of class Eigensystem

**Author(s)**

Anneleen Daemen <daemen.anneleen@gene.com>, Matthew Brauer <brauer.matthew@gene.com>

**See Also**

EigensystemPlotParam-class

---

YeastData\_alpha            *Yeast Cell Cycle Expression*

---

**Description**

To catalog genes in *Saccharomyces cerevisiae* whose transcript levels vary periodically within the cell cycle, mRNA levels in samples from yeast cultures were synchronized in G1 phase with alpha factor arrest. After release of the alpha factor, cells were sampled every 7 minutes over a timespan of 140 minutes, during which the cells synchronously completed two cell cycles.

**Usage**

```
data(YeastData_alpha)
YeastData
```

**Format**

An ExpressionSet with gene x sample expression data, gene annotation and sample information. Expression data comprise the un-logtransformed ratio of gene expression to reference mRNA from an asynchronous yeast culture. For the samples cell cycle phase is known. For 800 cell cycle-regulated genes, the phase in which these genes reach their peak expression was determined based on published timing of the expression of known cell cycle-regulated genes.

**Author(s)**

Anneleen Daemen, Matthew Brauer

**Source**

Spellman PT, Sherlock G, Zhang MQ, et al. Comprehensive identification of cell cycle-regulated genes of the Yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell* 9, 3273-3297 (1998).

**References**

Alter O, Brown PO and Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U.S.A.* 97(18), 10101-10106 (2000).

# Index

## \*Topic **classes**

Eigensystem-class, 6  
EigensystemPlotParam-class, 8

## \*Topic **datasets**

HeLaData\_exp\_DoubleThym\_2, 13  
StarvationData, 21  
YeastData\_alpha, 25

## \*Topic **methods**

Eigensystem-class, 6  
EigensystemPlotParam-class, 8

## \*Topic **package**

biosvd, 3

apply (Eigensystem-class), 6  
assayColorMap  
    (EigensystemPlotParam-class), 8  
assayColorMap<-, 3  
assaycorrelations (Eigensystem-class), 6  
assayMatrix (Eigensystem-class), 6

biosvd, 3  
biosvd-package (biosvd), 3

colorIdFeatures (Eigensystem-class), 6  
compute, 11, 17, 21  
compute (compute, Eigensystem-method), 4  
compute, data.frame-method  
    (compute, Eigensystem-method), 4  
compute, Eigensystem-method, 4, 7  
compute, ExpressionSet-method  
    (compute, Eigensystem-method), 4  
compute, matrix-method  
    (compute, Eigensystem-method), 4  
contrast (EigensystemPlotParam-class), 8  
contrast<-, 5

eigenassays (Eigensystem-class), 6  
eigenexpressions (Eigensystem-class), 6  
eigenfeatures (Eigensystem-class), 6  
Eigensystem (Eigensystem-class), 6

Eigensystem-class, 6  
EigensystemPlotParam  
    (EigensystemPlotParam-class), 8  
EigensystemPlotParam-class, 8  
entropy (Eigensystem-class), 6  
exclude, 5, 17, 21  
exclude (exclude, Eigensystem-method), 10  
exclude, Eigensystem-method, 10  
excludeEigenfeatures  
    (Eigensystem-class), 6

featureColorMap  
    (EigensystemPlotParam-class), 8  
featureColorMap<-, 11  
featurecorrelations  
    (Eigensystem-class), 6  
featureMatrix (Eigensystem-class), 6  
figure (EigensystemPlotParam-class), 8  
figure<-, 12  
filenames (EigensystemPlotParam-class),  
    8  
filenames<-, 13  
fractions (Eigensystem-class), 6

HeLaData (HeLaData\_exp\_DoubleThym\_2), 13  
HeLaData\_exp\_DoubleThym\_2, 13

matrix (Eigensystem-class), 6

negativeValues  
    (EigensystemPlotParam-class), 8  
negativeValues<-, 14

palette (EigensystemPlotParam-class), 8  
palette<-, 15  
path (EigensystemPlotParam-class), 8  
path<-, 15  
plot, 5, 11, 21

plot  
    (plot, Eigensystem, EigensystemPlotParam-method),  
    16

plot, Eigensystem, EigensystemPlotParam-method,  
16

plots (EigensystemPlotParam-class), 8

plots<-, 17

prefix (EigensystemPlotParam-class), 8

prefix<-, 18

project (project, Eigensystem-method), 19

project, Eigensystem-method, 19

report, 5, 11, 17

report  
(report, Eigensystem, EigensystemPlotParam-method),  
20

report, Eigensystem, EigensystemPlotParam-method,  
20

show, Eigensystem-method  
(Eigensystem-class), 6

show, EigensystemPlotParam-method  
(EigensystemPlotParam-class), 8

signMatrix (Eigensystem-class), 6

StarvationData, 21

whichAssays  
(EigensystemPlotParam-class), 8

whichAssays<-, 22

whichEigenassays  
(EigensystemPlotParam-class), 8

whichEigenassays<-, 23

whichEigenfeatures  
(EigensystemPlotParam-class), 8

whichEigenfeatures<-, 23

whichFeatures  
(EigensystemPlotParam-class), 8

whichFeatures<-, 24

whichPolarAxes  
(EigensystemPlotParam-class), 8

whichPolarAxes<-, 25

YeastData (YeastData\_alpha), 25

YeastData\_alpha, 25