

Package ‘biosvd’

October 7, 2014

Type Package

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

Version 1.0.0

Date 2013-12-11

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 (>= 3.0.0)

Imports gplots, BiocGenerics, Biobase, methods, grid, hwriter, ReportingTools, graphics

biocViews TimeCourse, Visualization

R topics documented:

biosvd	2
compute,Eigensystem-method	3
Eigensystem-class	4
exclude,Eigensystem-method	6
HeLaData_exp_DoubleThym_2	8
plot,Eigensystem,ANY-method	8
report,Eigensystem-method	10
sort,Eigensystem-method	12
StarvationData	13
YeastData_alpha	14

Index **15**

biosvd	<i>Package for high-throughput data processing, outlier detection, noise removal and dynamic modeling</i>
--------	---

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

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

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](#), [exclude,Eigensystem-method](#); [plot,plot,Eigensystem,ANY-method](#), [plot,Eigensystem,ANY-method](#); [report,report,Eigensystem-method](#), [report,Eigensystem-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")
```

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)
```

`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 ~ 0). Filtering out steady-scale variance corresponds to normalization by the steady scale of expression/intensity variance (geometric mean of variance ~ 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,data.frame-method](#), [compute,Eigensystem-method](#), [compute,Eigensystem-method](#), [compute,ExpressionSet-method](#), [compute,matrix-method](#); [plot](#), [plot,Eigensystem,ANY-method](#), [plot,Eigensystem,ANY-method](#); [report](#), [report,Eigensystem-method](#), [report,Eigensystem-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)
```

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).

plot,Eigensystem,ANY-method*Generate visualizations of the eigensystem*

Description

Generates up to 5 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,ANY
plot(x, y, contrast = 3, plots = c("heatmap",
  "fraction", "zoomedFraction", "lines", "allLines"), prefix = "biosvd",
  dir = getwd(), figure = FALSE)
```

Arguments

x	object of class eigensystem
contrast	contrast used for the heatmap visualization (default 3)
plots	plots to be shown: heatmap, fraction, zoomedFraction, lines and/or allLines
prefix	string to start the plot names with (default 'biosvd')
dir	directory to save the plots to (default work directory)
figure	boolean indicating whether figures should be shown (TRUE) or saved as pdf (FALSE, default)
y	the y coordinates of points in the plot, <i>optional</i> if x is an appropriate structure.

Details

The function generates a heatmap of the eigenfeatures by assays with use of the given contrast factor (heatmap), a bar plot with the eigenexpression fractions of all eigenfeatures (fraction), a bar plot with the eigenexpression fractions of the eigenfeatures without the dominant eigenfeature(s) (zoomedFraction), the intensity levels of eigenfeatures 1 to 4 across the assays (lines), and the intensity levels of all eigenfeatures across the assays (allLines).

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](#), [data.frame-method](#), [compute,Eigensystem-method](#), [compute,Eigensystem-method](#), [compute,ExpressionSet-method](#), [compute,matrix-method](#); [exclude](#), [exclude,Eigensystem-method](#), [exclude,Eigensystem-method](#); [report](#), [report,Eigensystem-method](#), [report,Eigensystem-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
plot(eigensystem)
## Generates all provided plots for the eigensystem, with use of contrast 2 for the heatmap
plot(eigensystem, contrast=2)
## Generates the fraction and lines plot for the eigensystem
plot(eigensystem, plots=c("fraction","lines"))
```

```
report,Eigensystem-method
```

Generates a summarizing report of the eigensystem

Description

Generates a summarizing report of the eigensystem

Usage

```
## S4 method for signature Eigensystem
report(x, eigenfeature.xaxis = 2,
       eigenfeature.yaxis = 1, colorIdAssays = rep(1, ncol(matrix(eigensystem))),
       colorIdFeatures = rep(1, nrow(matrix(eigensystem))), contrast = 3,
       prefix = "biosvd", dir = getwd())
```

Arguments

x	output of compute, plot or exclude
eigenfeature.xaxis	positive integer giving the eigenfeature to be shown on the x-axis of the polar plots
eigenfeature.yaxis	positive integer giving the eigenfeature to be shown on the y-axis of the polar plots
colorIdAssays	vector of integers, factor or a string variable in the pheno data of the original data set in case from class ExpressionSet, with assay annotation used for coloring purposes on the polar plot
colorIdFeatures	vector of integers, factor or a string variable in the feature data of the original data set in case from class ExpressionSet, with feature annotation used for coloring purposes on the polar plot
contrast	positive integer giving the contrast used for the heatmap visualization
prefix	prefix to start the plot and html file with
dir	directory to which the plots and html table are saved

Details

The function generates an html report of the eigensystem, containing polar plots for the assays and features separately that show the assays/features according to their correlation with two eigenfeatures/eigenassays, and a table with the list of features, sortable according to their coordinates, radius and phase in the polar plot.

The function also generates a visualization of the sorted data according to the two given eigenfeatures and eigenassays, showing a heatmap of the features by assays with colored feature/assay annotation information when provided, a heatmap of the features by eigenassays, and the intensity levels of the two sorted eigenassays across all features.

Methods

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

Author(s)

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

See Also

Other X.eigensystem.: [compute](#), [compute,data.frame-method](#), [compute,Eigensystem-method](#), [compute,Eigensystem-method](#), [compute,ExpressionSet-method](#), [compute,matrix-method](#); [exclude](#), [exclude,Eigensystem-method](#), [exclude,Eigensystem-method](#); [plot](#), [plot,Eigensystem,ANY-method](#), [plot,Eigensystem,ANY-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 without coloring of the assays and features
report(eigensystem)
## Generate report for eigenfeatures 2 and 3
report(eigensystem, eigenfeature.xaxis=2, eigenfeature.yaxis=3)
## Generate report for eigenfeatures 1 and 2 with coloring of the assays as indicated by the variable "Starvation" in
report(eigensystem, colorIdAssays="Starvation")
## Generate report for eigenfeatures 1 and 2 with use of prefix "AwesomeStudy" for the names of the plots and html ta
report(eigensystem, prefix="AwesomeStudy")
```

sort,Eigensystem-method

Sorts the data according to two eigenfeatures and eigenassays

Description

Sorts the data according to two eigenfeatures and eigenassays

Usage

```
## S4 method for signature Eigensystem
sort(x, decreasing = FALSE, eigenfeature.xaxis = 2,
     eigenfeature.yaxis = 1, colorIdFeatures = rep(1,
     nrow(matrix(eigensystem))))
```

Arguments

x	object of class eigensystem
decreasing	FALSE
eigenfeature.xaxis	first eigenfeature to sort against (default 2)
eigenfeature.yaxis	second eigenfeature to sort against (default 1)
colorIdFeatures	vector of integers or a string variable in the feature data of the original data set in case from class ExpressionSet, with feature annotation used for coloring purposes on the polar plot

Details

The function sorts the data according to two certain 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

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).

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)

## Sort the data according to eigenfeature 1 and 2
eigensystem.sorted <- sort(eigensystem)
## Sort the data according to eigenfeature 3 and 4
eigensystem.sorted <- sort(eigensystem, eigenfeature.xaxis=4, eigenfeature.yaxis=3)
## Visualization of the data after sorting to eigenfeature 3 and 4
plot(eigensystem.sorted)
```

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).

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, 4
- *Topic **datasets**
 - HeLaData_exp_DoubleThym_2, 8
 - StarvationData, 13
 - YeastData_alpha, 14
- *Topic **methods**
 - Eigensystem-class, 4
- *Topic **package**
 - biosvd, 2

- apply (Eigensystem-class), 4
- assaycorrelations (Eigensystem-class), 4
- assayMatrix (Eigensystem-class), 4

- biosvd, 2
- biosvd-package (biosvd), 2

- colorIdFeatures (Eigensystem-class), 4
- compute, 7, 9, 11
- compute (compute, Eigensystem-method), 3
- compute, data.frame-method
 - (compute, Eigensystem-method), 3
- compute, Eigensystem-method, 3, 6
- compute, ExpressionSet-method
 - (compute, Eigensystem-method), 3
- compute, matrix-method
 - (compute, Eigensystem-method), 3

- eigenassays (Eigensystem-class), 4
- eigenexpressions (Eigensystem-class), 4
- eigenfeatures (Eigensystem-class), 4
- Eigensystem (Eigensystem-class), 4
- Eigensystem-class, 4
- entropy (Eigensystem-class), 4
- exclude, 4, 9, 11
- exclude (exclude, Eigensystem-method), 6
- exclude, Eigensystem-method, 6
- excludeEigenfeatures
 - (Eigensystem-class), 4

- featurecorrelations
 - (Eigensystem-class), 4
- featureMatrix (Eigensystem-class), 4
- fractions (Eigensystem-class), 4

- HeLaData (HeLaData_exp_DoubleThym_2), 8
- HeLaData_exp_DoubleThym_2, 8

- matrix (Eigensystem-class), 4

- plot, 4, 7, 11
- plot (plot, Eigensystem, ANY-method), 8
- plot, Eigensystem, ANY-method, 8

- report, 4, 7, 9
- report (report, Eigensystem-method), 10
- report, Eigensystem-method, 10

- show, Eigensystem-method
 - (Eigensystem-class), 4
- signMatrix (Eigensystem-class), 4
- sort (sort, Eigensystem-method), 12
- sort, Eigensystem-method, 12
- StarvationData, 13

- YeastData (YeastData_alpha), 14
- YeastData_alpha, 14