Package ‘moduleColor’

February 27, 2008

Version 1.03-04

Date 2008-02-27

Title Basic module functions

Author Peter Langfelder <Peter.Langfelder@gmail.com> and Steve Horvath <SHorvath@mednet.ucla.edu>

Maintainer Peter Langfelder <Peter.Langfelder@gmail.com>

Depends R, stats, impute, grDevices, dynamicTreeCut

ZipData no

License GPL version 2 or newer

Description Methods for color labeling, calculation of eigengenes, merging of closely related modules.

URL http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting/

R topics documented:

- checkSets
- collectGarbage
- consensusMEDissimilarity
- consensusOrderMEs
- fixDataStructure
- labels2colors
- mergeCloseModules
- moduleColor-package
- moduleColor.getMEprefix
- moduleColor.revisionDate
- moduleColor.setMEprefix
- moduleColor.version
- moduleEigengenes
- moduleNumber
- multiSetMEs
- normalizeLabels
- orderMEs
- plotHclustColors
- removeGreyME
- standardColors
checkSets

Description

Checks whether given sets have the correct format and retrieves dimensions.

Usage

checkSets(data, checkStructure = FALSE, useSets = NULL)

Arguments

data  A vector of lists; in each list there must be a component named data whose content is a matrix or dataframe or array of dimension 2.
checkStructure  If FALSE, incorrect structure of data will trigger an error. If TRUE, an appropriate flag (see output) will be set to indicate whether data has correct structure.
useSets  Optional specification of entries of the vector data that are to be checked. Defaults to all components. This may be useful when data only contains information for some of the sets.

Details

For multiset calculations, many quantities (such as expression data, traits, module eigengenes etc) are presented by a common structure, a vector of lists (one list for each set) where each list has a component data that contains the actual (expression, trait, eigengene) data for the corresponding set in the form of a dataframe. This function checks whether data conforms to this convention and retrieves some basic dimension information (see output).

Value

A list with components

nSets  Number of sets (length of the vector data).
nGenes  Number of columns in the data components in the lists. This number must be the same for all sets.
nSamples  A vector of length nSets giving the number of rows in the data components.
structureOK  Only set if the argument checkStructure equals TRUE. The value is TRUE if the parameter data passes a few tests of its structure, and FALSE otherwise. The tests are not exhaustive and are meant to catch obvious user errors rather than be bulletproof.

Author(s)

Peter Langfelder, (Peter.Langfelder@gmail.com)
**collectGarbage**

Iterative garbage collection.

**Description**

Performs garbage collection until free memory indicators show no change.

**Usage**

`collectGarbage()`

**Value**

None.

**Author(s)**

Steve Horvath

---

**consensusMEDissimilarity**

Consensus dissimilarity of module eigengenes.

**Description**

Calculates consensus dissimilarity \((1 - \text{cor})\) of given module eigengenes realized in several sets.

**Usage**

`consensusMEDissimilarity(MEs, useAbs = FALSE, useSets = NULL, method = "consensus")`

**Arguments**

- **MEs**
  - Module eigengenes of the same modules in several sets.
- **useAbs**
  - Controls whether absolute value of correlation should be used instead of correlation in the calculation of dissimilarity.
- **useSets**
  - If the consensus is to include only a selection of the given sets, this vector (or scalar in the case of a single set) can be used to specify the selection. If NULL, all sets will be used.
- **method**
  - A character string giving the method to use. Allowed values are (abbreviations of) "consensus" and "majority". The consensus dissimilarity is calculated as the minimum of given set dissimilarities for "consensus" and as the average for "majority".

**Details**

This function calculates the individual set dissimilarities of the given eigengenes in each set, then takes the (parallel) maximum or average over all sets. For details on the structure of input data, see `checkSets`.
consensusOrderMEs

Value
A dataframe containing the matrix of dissimilarities, with names and rownames set appropriately.

Author(s)
Peter Langfelder, (Peter.Langfelder@gmail.com)

See Also
checkSets

Description
Reorder given (eigen-)vectors such that similar ones (as measured by correlation) are next to each other. This is a multi-set version of orderMEs: the dissimilarity used can be of consensus type (for each pair of eigenvectors the consensus dissimilarity is the maximum of individual set dissimilarities over all sets) or of majority type (for each pair of eigenvectors the consensus dissimilarity is the average of individual set dissimilarities over all sets).

Usage
consensusOrderMEs(MEs, useAbs = FALSE, useSets = NULL, greyLast = TRUE, greyName = paste(moduleColor.getMEprefix(), "grey", sep = ""), method = "consensus")

Arguments

MEs  Module eigengenes of several sets in a multi-set format (see checkSets). A vector of lists, with each list corresponding to one dataset and the module eigengenes in the component data, that is MEs[[set]]$data[sample, module] is the expression of the eigengene of module module in sample sample in dataset set. The number of samples can be different between the sets, but the modules must be the same.

useAbs  Controls whether vector similarity should be given by absolute value of correlation or plain correlation.

useSets  Allows the user to specify for which sets the eigengene ordering is to be performed.

greyLast  Normally the color grey is reserved for unassigned genes; hence the grey module is not a proper module and it is conventional to put it last. If this is not desired, set the parameter to FALSE.

greyName  Name of the grey module eigengene.

method  A character string giving the method to be used calculating the consensus dissimilarity. Allowed values are (abbreviations of) "consensus" and "majority". The consensus dissimilarity is calculated as the maximum of given set dissimilarities for "consensus" and as the average for "majority".
**Details**

Ordering module eigengenes is useful for plotting purposes. This function calculates the consensus or majority dissimilarity of given eigengenes over the sets specified by `useSets` (defaults to all sets). A hierarchical dendrogram is calculated using the dissimilarity and the order given by the dendrogram is used for the eigengenes in all other sets.

**Value**

A vector of lists of the same type as `MEs` containing the re-ordered eigengenes.

**Author(s)**

Peter Langfelder, (Peter.Langfelder@gmail.com)

**See Also**

`moduleEigengenes`, `multiSetMEs`, `orderMEs`

---

**Description**

Encapsulates single-set data in a wrapper that makes the data suitable for functions working on multiset data collections.

**Usage**

```r
fixDataStructure(data, verbose = 0, indent = 0)
```

**Arguments**

- `data`:
  A dataframe, matrix or array with two dimensions to be encapsulated.

- `verbose`:
  Controls verbosity. 0 is silent.

- `indent`:
  Controls indentation of printed progress messages. 0 means no indentation, every unit adds two spaces.

**Details**

For multiset calculations, many quantities (such as expression data, traits, module eigengenes etc) are presented by a common structure, a vector of lists (one list for each set) where each list has a component `data` that contains the actual (expression, trait, eigengene) data for the corresponding set in the form of a dataframe. This function creates a vector of lists of length 1 and fills the component `data` with the content of parameter `data`.

**Value**

As described above, input data in a format suitable for functions operating on multiset data collections.
labels2colors

Author(s)
Peter Langfelder, Peter.Langfelder@gmail.com

See Also
checkSets

Examples

```r
singleSetData = matrix(rnorm(100), 10,10);
encapsData = fixDataStructure(singleSetData);
length(encapsData)
names(encapsData[[1]])
dim(encapsData[[1]]$data)
all.equal(encapsData[[1]]$data, singleSetData);
```

---

 converts numerical labels to colors.

Description

Converts a vector or array of numerical labels into a corresponding vector or array of colors corresponding to the labels.

Usage

```r
labels2colors(labels, zeroIsGrey = TRUE, colorSeq = NULL)
```

Arguments

- `labels` Vector of non-negative integer labels.
- `zeroIsGrey` If TRUE, labels 0 will be assigned color grey. Otherwise, labels below 1 will trigger an error.
- `colorSeq` Color sequence corresponding to labels. If not given, a standard sequence will be used.

Details

The standard sequence start with well-distinguishable colors, and after about 40 turns into a quasirandom sampling of all colors available in R with the exception of all shades of grey (and gray).

If the input `labels` have a dimension attribute, it is copied into the output, meaning the dimensions of the returned value are the same as those of the input `labels`.

Value

A vector or array of character strings of the same length or dimensions as `labels`.

Author(s)
Peter Langfelder, Peter.Langfelder@gmail.com
mergeCloseModules

Examples

labels = c(0:20);
labels2colors(labels);

mergeCloseModules    Merge close modules of gene expression data.

Description

Merges modules in gene expression networks that are too close as measured by the correlation of their eigengenes.

Usage

mergeCloseModules(exprData, colors, cutHeight = 0.2, MEs = NULL,
                   useAbs = FALSE, iterate = TRUE, relabel = FALSE,
                   colorSeq = NULL, getNewMEs = TRUE, useSets = NULL,
                   checkDataFormat = TRUE, unassdColor = "grey",
                   verbose = 1, indent = 0)

Arguments

eexprData          Expression data in a multi-set format (see checkSets).
colors             A vector giving module colors for genes. The method only makes sense when genes have the same color label in all sets, hence a single vector.
cutHeight          Maximum dissimilarity (i.e., 1-correlation) that qualifies modules for merging.
MEs                If module eigengenes have been calculated before, the user can save some computational time by inputting them. If they are not given, they will be calculated.
useAbs             Specifies whether absolute value of correlation or plain correlation (of module eigengenes) should be used in calculating module dissimilarity.
iterate            Controls whether the merging procedure should be repeated until there is no change. If FALSE, only one iteration will be executed.
relabel            Controls whether, after merging, color labels should be ordered by module size.
colorSeq           Color labels to be used for relabeling. Defaults to the standard color order used in this package.
getNewMEs          Controls whether module eigengenes of merged modules should be calculated and returned.
useSets            A vector of scalar allowing the user to specify which sets will be used to calculate the consensus dissimilarity of module eigengenes. Defaults to all sets.
checkDataFormat    If TRUE, the function will check exprData and MEs for correct multi-set structure. If single set data is given, it will be converted into a format useable for the function. If FALSE, incorrect structure of input data will trigger an error.
unassdColor        Specifies the string that labels unassigned genes. Module of this color will not enter the module eigengene clustering and will not be merged with other modules.
Details

This function returns the color labels for modules that are obtained from the input modules by merging ones that are closely related. The relationships are quantified by correlations of module eigengenes; a "consensus" measure is defined as the minimum over the corresponding relationship in each set. Once the (dis-)similarity is calculated, average linkage hierarchical clustering of the module eigengenes is performed, the dendrogram is cut at the height \texttt{cutHeight} and modules on each branch are merged. Optionally, the process is repeated until no more modules are merged.

Value

A list with components

\begin{itemize}
  \item \texttt{colors} Color labels for the genes corresponding to merged modules.
  \item \texttt{dendro} Hierarchical clustering dendrogram (average linkage) of the eigengenes of the most recently computed tree. If \texttt{iterate} was set TRUE, this will be the dendrogram of the merged modules, otherwise it will be the dendrogram of the original modules.
  \item \texttt{oldDendro} Hierarchical clustering dendrogram (average linkage) of the eigengenes of the original modules.
  \item \texttt{cutHeight} The input \texttt{cutHeight}.
  \item \texttt{oldMEs} Module eigengenes of the original modules in the sets given by \texttt{useSets}.
  \item \texttt{newMEs} Module eigengenes of the merged modules in the sets given by \texttt{useSets}.
\end{itemize}

Author(s)

Peter Langfelder, (Peter.Langfelder@gmail.com)

---

moduleColor-package

Basic module functions

Description

Methods for color labeling, calculation of eigengenes, merging of closely related modules.

Details

Package:       moduleColor
Version:       1.03-04
Date:          2008-02-27
Depends:       R, stats, impute, grDevices, dynamicTreeCut
ZipData:       no
License:       GPL version 2 or newer
URL:           http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting/
Packaged:      Wed Feb 27 13:02:08 2008; plangfelder
Built:         R 2.4.1; 2008-02-27 13:02:57; unix
moduleColor.getMEprefix

Get the prefix used to label module eigengenes.

Description

Returns the currently used prefix used to label module eigengenes. When returning module eigengenes in a dataframe, names of the corresponding columns will start with the given prefix.
Usage

moduleColor.getMEprefix()

Details

Returns the prefix used to label module eigengenes. When returning module eigengenes in a
dataframe, names of the corresponding columns will consist of the corresponding color label pre-ceded by the given prefix. For example, if the prefix is "PC" and the module is turquoise, the corresponding module eigengene will be labeled "PCturquoise". Most of old code assumes "PC", but "ME" is more instructive and used in some newer analyses.

Value

A character string.

Author(s)

Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

See Also

moduleColor.setMEprefix, moduleEigengenes

---

moduleColor.revisionDate

Get the last revision date of the package.

---

Description

Returns the last revision date of the package.

Usage

moduleColor.revisionDate()

Value

A character string.

Author(s)

Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩
moduleColor.setMEprefix

*Set the prefix used to label module eigengenes.*

**Description**
Sets the prefix used to label module eigengenes. When returning module eigengenes in a dataframe, names of the corresponding columns will start with the given prefix.

**Usage**
```r
moduleColor.setMEprefix(prefix)
```

**Arguments**
- `prefix` A character string of length 2. Recommended values are "PC" (the default start-up value) and "ME".

**Details**
Sets the prefix used to label module eigengenes. When returning module eigengenes in a dataframe, names of the corresponding columns will consist of the corresponding color label preceded by the given prefix. For example, if the prefix is "PC" and the module is turquoise, the corresponding module eigengene will be labeled "PCturquoise". Most of old code assumes "PC", but "ME" is more instructive and used in some newer analyses.

**Value**
None.

**Author(s)**
Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

**See Also**
- `moduleColor.getMEprefix`
- `moduleEigengenes`

---

moduleColor.version

*Get the version number of the package.*

**Description**
Returns the version number of the package.

**Usage**
```r
moduleColor.version()
```
moduleEigengenes

Value
A character string.

Author(s)
Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

moduleEigengenes Calculate module eigengenes.

Description
Calculates module eigengenes (1st principal component) of modules in a given single dataset.

Usage
moduleEigengenes(expr, colors, impute = TRUE, nPC = 1, align = "along average", verbose = 0, indent = 0)

Arguments
expr Expression data for a single set in the form of a dataframe where rows are samples and columns are genes (probes).

colors A vector of the same length as the number of probes in expr, giving module color for all probes (genes). Color "grey" is reserved for unassigned genes.

impute If TRUE, expression data will be checked for the presence of NA entries and if the latter are present, numerical data will be imputed, using function impute.knn and probes from the same module as the missing datum. The function impute.knn uses a fixed random seed giving repeatable results.

nPC Number of principal components to be calculated. If only eigengenes are needed, it is best to set it to 1 (default). If variance explained is needed as well, use value NULL. This will cause all principal components to be computed, which is slower.

align Controls whether eigengenes, whose orientation is undetermined, should be aligned with average expression (align = "along average", the default) or left as they are (align = ""). Any other value will trigger an error.

verbose Controls verbostity of printed progress messages. 0 means silent, up to (about) 5 the verbosity gradually increases.

indent A single non-negative integer controlling indentation of printed messages. 0 means no indentation, each unit above that adds two spaces.

Details
Module eigengene is defined as the first principal component of the expression matrix of the corresponding module.
**Value**

A list with the following components:

- **eigengenes**: Module eigengenes in a dataframe, with each column corresponding to one eigengene. The columns are named by the corresponding color with an "ME" prepended, e.g., MEturquoise etc.

- **averageExpr**: If align == "along average", a dataframe containing average normalized expression in each module. The columns are named by the corresponding color with an "AE" prepended, e.g., AEturquoise etc.

- **varExplained**: A dataframe in which each column corresponds to a module, with the component varExplained[PC, module] giving the variance of module module explained by the principal component no. PC. This is only accurate if all principal components have been computed (input nPC = NULL). At most 5 principal components are recorded in this dataframe.

- **nPC**: A copy of the input nPC.

**Author(s)**

Steve Horvath (SHorvath@mednet.ucla.edu), Peter Langfelder (Peter.Langfelder@gmail.com)

**References**


---

**moduleNumber**

*Fixed-height cut of a dendrogram.*

**Description**

Detects branches of on the input dendrogram by performing a fixed-height cut.

**Usage**

```r
moduleNumber(dendro, cutHeight = 0.9, minSize = 50)
```

**Arguments**

- **dendro**: a hierarchical clustering dendrogram such as one returned by hclust.

- **cutHeight**: Maximum joining heights that will be considered.

- **minSize**: Minimum cluster size.

**Details**

All contiguous branches below the height cutHeight that contain at least minSize objects are assigned unique positive numerical labels; all unassigned objects are assigned label 0.

**Value**

A vector of numerical labels giving the assignment of each object.
multiSetMEs

Note
The numerical labels may not be sequential. See normalizeLabels for a way to put the labels into a standard order.

Author(s)
Peter Langfelder, Peter.Langfelder@gmail.com

See Also
hclust, cutree, normalizeLabels

multiSetMEs  Calculate module eigengenes.

Description
Calculates module eigengenes for several sets.

Usage
multiSetMEs(exprData, colors, universalColors = NULL, useSets = NULL, impute = TRUE,
nPC = 1, align = "along average", verbose = 1, indent = 0)

Arguments
 exprData Expression data in a multi-set format (see checkSets). A vector of lists, with each list corresponding to one microarray dataset and expression data in the component `data`, that is `expr[[set]]$data[sample, probe]` is the expression of probe `probe` in sample `sample` in dataset `set`. The number of samples can be different between the sets, but the probes must be the same.
colors A matrix of dimensions (number of probes, number of sets) giving the module assignment of each gene in each set. The color “grey” is interpreted as unassigned.
universalColors Alternative specification of module assignment. A single vector of length (number of probes) giving the module assignment of each gene in all sets (that is the modules are common to all sets). If given, takes precedence over color.
useSets If calculations are requested in (a) selected set(s) only, the set(s) can be specified here. Defaults to all sets.
impute Logical. If TRUE, expression data will be checked for the presence of NA entries and if the latter are present, numerical data will be imputed, using function impute.knn and probes from the same module as the missing datum. The function impute.knn uses a fixed random seed giving repeatable results.
nPC Number of principal components to be calculated. If only eigengenes are needed, it is best to set it to 1 (default). If variance explained is needed as well, use value NULL. This will cause all principal components to be computed, which is slower.
align Controls whether eigengenes, whose orientation is undetermined, should be aligned with average expression (align = "along average", the default) or left as they are (align = "."). Any other value will trigger an error.
**normalizeLabels**

Controls verbosity of printed progress messages. 0 means silent, up to (about) 5 the verbosity gradually increases.

A single non-negative integer controlling indentation of printed messages. 0 means no indentation, each unit above that adds two spaces.

**Details**

See `moduleEigengenes` for details on calculation of module eigengenes in individual sets. This function simply calls `moduleEigengenes` for each set represented in `exprData`.

**Value**

A vector of lists similar in spirit to the input `exprData`. For each set there is a list with the following components:

- **data**
  Module eigengenes in a dataframe, with each column corresponding to one eigengene. The columns are named by the corresponding color with an "ME" prepended, e.g., `MEturquoise` etc.

- **averageExpr**
  If `align == "along average"`, a dataframe containing average normalized expression in each module. The columns are named by the corresponding color with an "AE" prepended, e.g., `AEturquoise` etc.

- **varExplained**
  A dataframe in which each column corresponds to a module, with the component `varExplained[PC, module]` giving the variance of module `module` explained by the principal component no. `PC`. This is only accurate if all principal components have been computed (input `nPC = NULL`). At most 5 principal components are recorded in this dataframe.

- **nPC**
  A copy of the input `nPC`.

**Author(s)**

Peter Langfelder, (Peter.Langfelder@gmail.com)

**See Also**

`moduleEigengenes`

---

**normalizeLabels**

**Transform numerical labels into normal order.**

**Description**

Transforms numerical labels into normal order, that is the largest group will be labeled 1, next largest 2 etc. Label 0 is optionally preserved.

**Usage**

```r
normalizeLabels(labels, keepZero = TRUE)
```

**Arguments**

- **labels**
  Numerical labels.

- **keepZero**
  If TRUE (the default), labels 0 are preserved.
Value

A vector of the same length as input, containing the normalized labels.

Author(s)

Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

---

**orderMEs**

*Put close eigenvectors next to each other*

---

**Description**

Reorder given (eigen-)vectors such that similar ones (as measured by correlation) are next to each other.

**Usage**

```r
orderMEs(MEs, greyLast = TRUE,
    greyName = paste(moduleColor.getMEprefix(), "grey", sep=""),
    orderBy = 1, order = NULL,
    useSets = NULL, verbose = 0, indent = 0)
```

**Arguments**

- **MEs**
  Module eigengenes in a multi-set format (see checkSets). A vector of lists, with each list corresponding to one dataset and the module eigengenes in the component data, that is MEs[[set]]$data[sample, module] is the expression of the eigengene of module module in sample sample in dataset set. The number of samples can be different between the sets, but the modules must be the same.

- **greyLast**
  Normally the color grey is reserved for unassigned genes; hence the grey module is not a proper module and it is conventional to put it last. If this is not desired, set the parameter to FALSE.

- **greyName**
  Name of the grey module eigengene.

- **orderBy**
  Specifies the set by which the eigengenes are to be ordered (in all other sets as well). Defaults to the first set in useSets (or the first set, if useSets is not given).

- **order**
  Allows the user to specify a custom ordering.

- **useSets**
  Allows the user to specify for which sets the eigengene ordering is to be performed.

- **verbose**
  Controls verbostity of printed progress messages. 0 means silent, nonzero verbose.

- **indent**
  A single non-negative integer controlling indentation of printed messages. 0 means no indentation, each unit above zero adds two spaces.

**Details**

Ordering module eigengenes is useful for plotting purposes. For this function the order can be specified explicitly, or a set can be given in which the correlations of the eigengenes will determine the order. For the latter, a hierarchical dendrogram is calculated and the order given by the dendrogram is used for the eigengenes in all other sets.
Value

A vector of lists of the same type as MEs containing the re-ordered eigengenes.

Author(s)

Peter Langfelder, (Peter.Langfelder@gmail.com)

See Also

moduleEigengenes, multiSetMEs, consensusOrderMEs

---

plotHclustColors  Plot color bars corresponding to modules

Description

Plot color bars corresponding to modules, usually beneath a dendrogram.

Usage

plotHclustColors(dendro, colors, rowLabels = NULL, cex.rowLabels = 0.9, ...)

Arguments

dendro A dendrogram such as returned by hclust.

colors Coloring of objects on the dendrogram. Either a vector (one color per object) or a matrix (can also be an array or a data frame) with each column giving one color per object. Each column will be plotted as a horizontal row of colors under the dendrogram.

rowLabels Labels for the colorings given in colors. The labels will be printed to the left of the color rows in the plot. If the argument is given, it must be a vector of length equal to the number of columns in colors. If not given, names(colors) will be used if available. If not, sequential numbers starting from 1 will be used.

cex.rowLabels Font size scale factor for the row labels. See par.

... Other parameters to be passed on to the plotting method (such as main for the main title etc).

Details

It is often useful to plot module assignment (by color) that was obtained by cutting a hierarchical dendrogram, to visually check whether the obtained modules are meaningful, or which one of several possible module assignments looks best. One way to do it is to section the screen into two parts, plot the dendrogram (via plot(hclust)) in the upper section and use this function to plot colors in the order corresponding to the dendrogram in the lower section.

Value

None.
**stdandardColors**

Colors this library uses for labeling modules.

**Description**

Returns the vector of color names in the order they are assigned by other functions in this library.

**Usage**

`standardColors(n = NULL)`

**Arguments**

- `n`  
  Number of colors requested. If `NULL`, all (approx. 450) colors will be returned. Any other invalid argument such as less than one or more than maximum (`length(standardColors())`) will trigger an error.
**standardColors**

**Value**

A vector of character color names of the requested length.

**Author(s)**

Peter Langfelder, (Peter.Langfelder@gmail.com)

**Examples**

```r
standardColors(10);
```
Index

*Topic cluster
  moduleNumber, 13
*Topic color
  labels2colors, 6
  standardColors, 18
*Topic hplot
  plotHclustColors, 17
*Topic misc
  checkSets, 1
  consensusMEDissimilarity, 3
  consensusOrderMEs, 4
  fixDataStructure, 5
  mergeCloseModules, 7
  moduleColor.getMEprefix, 9
  moduleColor.revisionDate, 10
  moduleColor.setMEprefix, 10
  moduleColor.version, 11
  moduleEigengenes, 12
  multiSetMEs, 14
  normalizeLabels, 15
  orderMEs, 16
  removeGreyME, 18
  standardColors, 18
*Topic package
  moduleColor-package, 8
*Topic utilities
  collectGarbage, 2
  checkSets, 1, 3–5, 7, 14, 16, 18
  collectGarbage, 2
  consensusMEDissimilarity, 3
  consensusOrderMEs, 4, 17
  cutree, 14
  fixDataStructure, 5
  hclust, 14, 17
  labels2colors, 6
  mergeCloseModules, 7
  moduleColor
    (moduleColor-package), 8
    moduleColor-package, 8
    moduleColor.getMEprefix, 9, 11
  moduleColor.revisionDate, 10
  moduleColor.setMEprefix, 10, 11
  moduleColor.version, 11
  moduleEigengenes, 5, 10, 11, 12, 15, 17
  moduleNumber, 13
  multiSetMEs, 5, 14, 17
  normalizeLabels, 14, 15
  orderMEs, 4, 5, 16
  par, 17
  plotHclustColors, 17
  removeGreyME, 18
  standardColors, 18