

# 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 . . . . .	2
collectGarbage . . . . .	3
consensusMEDissimilarity . . . . .	3
consensusOrderMEs . . . . .	4
fixDataStructure . . . . .	5
labels2colors . . . . .	6
mergeCloseModules . . . . .	7
moduleColor-package . . . . .	8
moduleColor.getMEprefix . . . . .	9
moduleColor.revisionDate . . . . .	10
moduleColor.setMEprefix . . . . .	11
moduleColor.version . . . . .	11
moduleEigengenes . . . . .	12
moduleNumber . . . . .	13
multiSetMEs . . . . .	14
normalizeLabels . . . . .	15
orderMEs . . . . .	16
plotHclustColors . . . . .	17
removeGreyME . . . . .	18
standardColors . . . . .	18

**Index****20**


---

checkSets	<i>Check structure and retrieve sizes of a group of datasets.</i>
-----------	---

---

**Description**

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

**Usage**

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

**Arguments**

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

<code>nSets</code>	Number of sets (length of the vector <code>data</code> ).
<code>nGenes</code>	Number of columns in the <code>data</code> components in the lists. This number must be the same for all sets.
<code>nSamples</code>	A vector of length <code>nSets</code> giving the number of rows in the <code>data</code> components.
<code>structureOK</code>	Only set if the argument <code>checkStructure</code> equals <code>TRUE</code> . The value is <code>TRUE</code> if the parameter <code>data</code> passes a few tests of its structure, and <code>FALSE</code> 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](#).

**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](#)

---

consensusOrderMEs *Put close eigenvectors next to each other in several sets.*

---

**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 <a href="#">checkSets</a> ). A vector of lists, with each list corresponding to one dataset and the module eigengenes in the component <code>data</code> , that is <code>MEs[[set]]\$data[sample, module]</code> is the expression of the eigengene of module <code>module</code> in sample <code>sample</code> in dataset <code>set</code> . 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 <code>FALSE</code> .
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) <code>"consensus"</code> and <code>"majority"</code> . The consensus dissimilarity is calculated as the maximum of given set dissimilarities for <code>"consensus"</code> and as the average for <code>"majority"</code> .

## 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](#)

---

`fixDataStructure`    *Put single-set data into a form useful for multiset calculations.*

---

## Description

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

## Usage

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

## Arguments

<code>data</code>	A dataframe, matrix or array with two dimensions to be encapsulated.
<code>verbose</code>	Controls verbosity. 0 is silent.
<code>indent</code>	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.

**Author(s)**

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

**See Also**

[checkSets](#)

**Examples**

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

---

labels2colors

*Convert 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**

```
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 quasi-random 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)

**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**

<code>exprData</code>	Expression data in a multi-set format (see <a href="#">checkSets</a> ).
<code>colors</code>	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.
<code>cutHeight</code>	Maximum dissimilarity (i.e., 1-correlation) that qualifies modules for merging.
<code>MEs</code>	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.
<code>useAbs</code>	Specifies whether absolute value of correlation or plain correlation (of module eigengenes) should be used in calculating module dissimilarity.
<code>iterate</code>	Controls whether the merging procedure should be repeated until there is no change. If FALSE, only one iteration will be executed.
<code>relabel</code>	Controls whether, after merging, color labels should be ordered by module size.
<code>colorSeq</code>	Color labels to be used for relabeling. Defaults to the standard color order used in this package.
<code>getNewMEs</code>	Controls whether module eigengenes of merged modules should be calculated and returned.
<code>useSets</code>	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.
<code>checkDataFormat</code>	If TRUE, the function will check <code>exprData</code> and <code>MEs</code> 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.
<code>unassdColor</code>	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.

verbose	Controls verbosity 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

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 `cutHeight` and modules on each branch are merged. Optionally, the process is repeated until no more modules are merged.

### Value

A list with components

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

### 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

**Index:**

checkSets	Retrieve basic sizes of a group of datasets.
collectGarbage	Iterative garbage collection.
consensusMEDissimilarity	
consensusOrderMEs	Consensus dissimilarity of module eigengenes. Put close eigenvectors next to each other in several sets.
fixDataStructure	Put single-set data into a form useful for multiset calculations.
labels2colors	Convert numerical labels to colors.
mergeCloseModules	Merge close modules of gene expression data.
moduleColor-package	Basic module functions.
moduleColor.getMEprefix	Get the prefix used to label module eigengenes.
moduleColor.setMEprefix	Set the prefix used to label module eigengenes.
moduleColor.version	Returns the version number of the package.
moduleColor.revisionDate	Returns the revision date of the package.
moduleEigengenes	Calculate module eigengenes.
moduleNumber	Fixed-height cut of a dendrogram.
multiSetMEs	Calculate module eigengenes.
normalizeLabels	Transform numerical labels into normal order.
orderMEs	Put close eigenvectors next to each other
plotHclustColors	Plot color bars corresponding to modules
removeGreyME	Remove the grey module eigengene from given eigengenes.
standardColors	Colors this library uses for labeling modules.

**Author(s)**

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

**References****See Also****Examples**


---

```
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 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**

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

```
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

```
moduleColor.version()
```

**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",
```

**Arguments**

<code>expr</code>	Expression data for a single set in the form of a dataframe where rows are samples and columns are genes (probes).
<code>colors</code>	A vector of the same length as the number of probes in <code>expr</code> , giving module color for all probes (genes). Color "grey" is reserved for unassigned genes.
<code>impute</code>	If <code>TRUE</code> , expression data will be checked for the presence of NA entries and if the latter are present, numerical data will be imputed, using function <code>impute.knn</code> and probes from the same module as the missing datum. The function <code>impute.knn</code> uses a fixed random seed giving repeatable results.
<code>nPC</code>	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 <code>NULL</code> . This will cause all principal components to be computed, which is slower.
<code>align</code>	Controls whether eigengenes, whose orientation is undetermined, should be aligned with average expression ( <code>align = "along average"</code> , the default) or left as they are ( <code>align = ""</code> ). Any other value will trigger an error.
<code>verbose</code>	Controls verbosity of printed progress messages. 0 means silent, up to (about) 5 the verbosity gradually increases.
<code>indent</code>	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**

Zhang, B. and Horvath, S. (2005), "A General Framework for Weighted Gene Co-Expression Network Analysis", Statistical Applications in Genetics and Molecular Biology: Vol. 4: No. 1, Article 17

---

moduleNumber	<i>Fixed-height cut of a dendrogram.</i>
--------------	--

---

**Description**

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

**Usage**

```
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.

**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	<i>Calculate module eigengenes.</i>
-------------	-------------------------------------

---

**Description**

Calculates module eigengenes for several sets.

**Usage**

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

**Arguments**

exprData	Expression data in a multi-set format (see <a href="#">checkSets</a> ). A vector of lists, with each list corresponding to one microarray dataset and expression data in the component data, that is <code>expr[[set]]\$data[sample, probe]</code> is the expression of probe <code>probe</code> in sample <code>sample</code> in dataset <code>set</code> . 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 <code>color</code> .
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 <code>impute.knn</code> and probes from the same module as the missing datum. The function <code>impute.knn</code> 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 ( <code>align = "along average"</code> , the default) or left as they are ( <code>align = ""</code> ). Any other value will trigger an error.

verbose	Controls verbosity 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

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 <code>align == "along average"</code> , 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 <code>varExplained[PC, module]</code> giving the variance of module <code>module</code> explained by the principal component no. <code>PC</code> . This is only accurate if all principal components have been computed (input <code>nPC = NULL</code> ). At most 5 principal components are recorded in this dataframe.
nPC	A copy of the input <code>nPC</code> .

### 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

```
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**

```
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 <a href="#">checkSets</a> ). A vector of lists, with each list corresponding to one dataset and the module eigengenes in the component data, that is <code>MEs[[set]]\$data[sample, module]</code> is the expression of the eigengene of module <code>module</code> in sample <code>sample</code> in dataset <code>set</code> . 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 <code>FALSE</code> .
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 <code>useSets</code> (or the first set, if <code>useSets</code> 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 verbosity 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**

<code>dendro</code>	A dendrogram such as returned by <a href="#">hclust</a> .
<code>colors</code>	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.
<code>rowLabels</code>	Labels for the colorings given in <code>colors</code> . 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 <code>colors</code> . If not given, <code>names(colors)</code> will be used if available. If not, sequential numbers starting from 1 will be used.
<code>cex.rowLabels</code>	Font size scale factor for the row labels. See <a href="#">par</a> .
<code>...</code>	Other parameters to be passed on to the plotting method (such as <code>main</code> 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 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.

**Author(s)**

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

`removeGreyME` *Removes the grey eigengene from a given collection of eigengenes.*

**Description**

Given module eigengenes either in a single data frame or in a multi-set format, removes the grey eigengenes from each set. If the grey eigengenes are not found, a warning is issued.

**Usage**

```
removeGreyME(MEs, greyMEName = paste(moduleColor.getMEprefix(), "grey", sep=""))
```

**Arguments**

<code>MEs</code>	Module eigengenes, either in a single data frame (typically for a single set), or in a multi-set format. See <a href="#">checkSets</a> for a description of the multi-set format.
<code>greyMEName</code>	Name of the module eigengene (in each corresponding data frame) that corresponds to the grey color. This will typically be "PCgrey" or "MEgrey". If the module eigengenes were calculated using standard functions in this library, the default should work.

**Value**

Module eigengenes in the same format as input (either a single data frame or a vector of lists) with the grey eigengene removed.

**Author(s)**

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

`standardColors` *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**

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

**Value**

A vector of character color names of the requested length.

**Author(s)**

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

**Examples**

```
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, 11
    - 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