

# Package ‘EnrichedHeatmap’

October 12, 2016

**Type** Package

**Title** Making Enriched Heatmaps

**Version** 1.2.2

**Date** 2016-4-27

**Author** Zuguang Gu

**Maintainer** Zuguang Gu <z.gu@dkfz.de>

**Depends** R (>= 3.1.2), grid, ComplexHeatmap (>= 1.9.7), GenomicRanges, IRanges, locfit

**Imports** methods, matrixStats, stats, GetoptLong

**Suggests** testthat (>= 0.3), knitr, markdown, circlize (>= 0.3.1)

**VignetteBuilder** knitr

**Description** Enriched heatmap is a special type of heatmap which visualizes the enrichment of genomic signals on specific target regions. Here we implement enriched heatmap by ComplexHeatmap package. Since this type of heatmap is just a normal heatmap but with some special settings, with the functionality of ComplexHeatmap, it would be much easier to customize the heatmap as well as concatenating to a list of heatmaps to show correspondance between different data sources.

**biocViews** Software, Visualization, Sequencing, GenomeAnnotation, Coverage

**URL** <https://github.com/jokergoo/EnrichedHeatmap>

**License** GPL (>= 2)

**Repository** Bioconductor

**Date/Publication** 2016-4-27 00:00:00

**NeedsCompilation** no

## R topics documented:

+AdditiveUnit . . . . .	2
anno_enriched . . . . .	3

copyAttr . . . . .	4
default_smooth_fun . . . . .	5
draw-dispatch . . . . .	5
draw-EnrichedHeatmap-method . . . . .	6
draw-EnrichedHeatmapList-method . . . . .	7
EnrichedHeatmap . . . . .	8
EnrichedHeatmap-class . . . . .	9
EnrichedHeatmapList . . . . .	10
EnrichedHeatmapList-class . . . . .	11
enriched_score . . . . .	11
getSignalsFromList . . . . .	12
makeWindows . . . . .	14
normalizeToMatrix . . . . .	15
print.normalizedMatrix . . . . .	17
show-dispatch . . . . .	18
show-EnrichedHeatmap-method . . . . .	19
show-EnrichedHeatmapList-method . . . . .	19
[.normalizedMatrix . . . . .	20

## Index 21

---

+.AdditiveUnit	<i>Add heatmaps or row annotations to a heatmap list</i>
----------------	--

---

### Description

Add heatmaps or row annotations to a heatmap list

### Usage

```
## S3 method for class 'AdditiveUnit'
x + y
```

### Arguments

x	an <a href="#">EnrichedHeatmap-class</a> object, a <a href="#">Heatmap-class</a> object, a <a href="#">HeatmapAnnotation-class</a> object or a <a href="#">HeatmapList-class</a> object.
y	an <a href="#">EnrichedHeatmap-class</a> object, a <a href="#">Heatmap-class</a> object, a <a href="#">HeatmapAnnotation-class</a> object or a <a href="#">HeatmapList-class</a> object.

### Details

It overwrites `+.AdditiveUnit` in the `ComplexHeatmap` package.

### Value

A [HeatmapList-class](#) object or an [EnrichedHeatmapList-class](#) object

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
# users should not use it directly
NULL
```

---

anno_enriched	<i>Annotation function to show the enrichment</i>
---------------	---

---

**Description**

Annotation function to show the enrichment

**Usage**

```
anno_enriched(gp = gpar(col = "red"), pos_line = TRUE, pos_line_gp = gpar(lty = 2),
  yaxis = TRUE, ylim = NULL, value = c("mean", "sum"), yaxis_side = "right",
  yaxis_gp = gpar(fontsize = 8), show_error = FALSE)
```

**Arguments**

gp	graphic parameters
pos_line	whether draw vertical lines which represent the position of target
pos_line_gp	graphic parameters
yaxis	whether show yaxis
ylim	ranges on y-axis
value	what type of value corresponds to the y-axis
yaxis_side	side of y-axis
yaxis_gp	graphic parameters for yaxis
show_error	whether show error regions which are $\pm 1$ sd to the mean value. Color of error area is same as the corresponding lines with 75 percent transparency.

**Details**

This annotation functions shows mean values of columns in the normalized matrix which represents the enrichment of the signals to the targets.

If rows are splitted, there will also be multiple lines in this annotation.

It should only be placed as column annotation of the Enriched Heatmap.

**Value**

A column annotation function which can be set to top\_annotation argument in [EnrichedHeatmap](#).

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
load(paste0(system.file("extdata", "chr21_test_data.RData", package = "EnrichedHeatmap")))
tss = promoters(genes, upstream = 0, downstream = 1)
mat1 = normalizeToMatrix(H3K4me3, tss, value_column = "coverage",
  extend = 5000, mean_mode = "w0", w = 50, trim = c(0, 0.01))
EnrichedHeatmap(mat1, col = c("white", "red"), name = "H3K4me3",
  top_annotation = HeatmapAnnotation(lines = anno_enriched(gp = gpar(col = 2:4))),
  top_annotation_height = unit(2, "cm"),
  km = 3, row_title_rot = 0)
```

---

copyAttr

*Copy attributes to another object*

---

**Description**

Copy attributes to another object

**Usage**

```
copyAttr(x, y)
```

**Arguments**

x	object 1
y	object 2

**Details**

The [normalizeToMatrix](#) object actually is a matrix but with more additional attributes attached. This function is used to copy these new attributes when dealing with the matrix.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
NULL
```

---

default\_smooth\_fun     *Default smooth function*

---

### Description

Default smooth function

### Usage

```
default_smooth_fun(x)
```

### Arguments

x                    input numeric vector

### Details

The smooth function is applied to every row in the normalized matrix. For this default smooth function, [locfit](#) is first tried on the vector. If there is error, [loess](#) smoothing is tried afterwards. If both smoothing are failed, there will be an error.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
# There is no example  
NULL
```

---

draw-dispatch             *Method dispatch page for draw*

---

### Description

Method dispatch page for draw.

### Dispatch

draw can be dispatched on following classes:

- [draw, EnrichedHeatmapList-method, EnrichedHeatmapList-class](#) class method
- [draw, EnrichedHeatmap-method, EnrichedHeatmap-class](#) class method

**Examples**

```
# no example
NULL
```

---

draw-EnrichedHeatmap-method  
*Draw a single heatmap*

---

**Description**

Draw a single heatmap

**Usage**

```
## S4 method for signature 'EnrichedHeatmap'
draw(object, internal = FALSE, ...)
```

**Arguments**

object	an <a href="#">EnrichedHeatmap-class</a> object.
internal	only used internally.
...	pass to <a href="#">draw,HeatmapList-method</a> .

**Details**

The function creates an [EnrichedHeatmapList-class](#) object which only contains a single heatmap and call [draw,EnrichedHeatmapList-method](#) to make the final heatmap.

**Value**

An [EnrichedHeatmapList-class](#) object.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
# see documentation of EnrichedHeatmap
NULL
```

---

draw-EnrichedHeatmapList-method

*Draw a list of heatmaps*

---

## Description

Draw a list of heatmaps

## Usage

```
## S4 method for signature 'EnrichedHeatmapList'  
draw(object, padding = unit(c(2, 2, 2, 2), "mm"),  
      newpage= TRUE, ...)
```

## Arguments

object	an <a href="#">EnrichedHeatmapList-class</a> object
padding	padding of the plot. Elements correspond to bottom, left, top, right paddings.
newpage	whether to create a new page
...	pass to <a href="#">make_layout,HeatmapList-method</a> or <a href="#">draw,HeatmapList-method</a>

## Details

It calls [draw,HeatmapList-method](#) to make the plot but with some adjustment specifically for enriched heatmaps.

## Value

An [EnrichedHeatmapList](#) object

## Author(s)

Zuguang Gu <z.gu@dkfz.de>

## Examples

```
# see documentation of EnrichedHeatmap  
NULL
```

---

EnrichedHeatmap      *Constructor method for EnrichedHeatmap class*

---

## Description

Constructor method for EnrichedHeatmap class

## Usage

```
EnrichedHeatmap(mat, score_fun = enriched_score, row_order = NULL, pos_line = TRUE,
  pos_line_gp = gpar(lty = 2), axis_name = NULL, axis_name_rot = NULL,
  axis_name_gp = gpar(fontsize = 10), border = TRUE, cluster_rows = FALSE,
  show_row_dend = FALSE, ...)
```

## Arguments

mat	a matrix which is returned by <a href="#">normalizeToMatrix</a>
score_fun	a function which calculates enriched scores for rows in mat. This function can be self-defined, take a look at <a href="#">enriched_score</a> to find out how to design it. Note if row clustering is turned on, this argument is ignored.
row_order	row order. If it is specified, score_fun is ignored.
pos_line	whether draw vertical lines which represent the position of target
pos_line_gp	graphic parameters for lines
axis_name	names for axis which is below the heatmap. If the targets are single points, axis_name is a vector of length three which corresponds to upstream, target itself and downstream. If the targets are regions with width larger than 1, axis_name should be a vector of length four which corresponds to upstream, start of targets, end of targets and downstream.
axis_name_rot	rotation for axis names
axis_name_gp	graphic parameters for axis names
border	whether show border of the heatmap
cluster_rows	clustering on rows are turned off by default
show_row_dend	whether show dendrograms on rows
...	pass to <a href="#">Heatmap</a>

## Details

[EnrichedHeatmap-class](#) is inherited from [Heatmap-class](#). Following parameters are set with pre-defined values:

**row\_order** the rows are sorted by the enriched score which is calculated by score\_fun. The sorting is applied decreasingly.

**cluster\_columns** enforced to be FALSE



**show\_row\_names** enforced to be FALSE  
**show\_column\_names** enforced to be FALSE  
**bottom\_annotation** enforced to be NULL  
**column\_title\_side** enforced to be top

With above pre-defined values, no graphics will be drawn below the heatmap, then the space below the heatmap can be used to add a new graph which contains the axis. A (or two) line which corresponds to the position of target will be added to the heatmap body as well.

Same as the [Heatmap-class](#), users can make more controls on the heatmap such as apply clustering on rows, or split rows by data frame or k-means clustering. Users can also add more than one heatmaps by + operator.

For a detailed demonstration, please go to the vignette.

### Value

An [EnrichedHeatmap-class](#) object which is inherited from [Heatmap-class](#).

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
load(paste0(system.file("extdata", "chr21_test_data.RData",
  package = "EnrichedHeatmap")))
mat3 = normalizeToMatrix(meth, cgi, value_column = "meth", mean_mode = "absolute",
  extend = 5000, w = 50, empty_value = 0.5)
EnrichedHeatmap(mat3, name = "methylation", column_title = "methylation near CGI")
EnrichedHeatmap(mat3, name = "meth1") + EnrichedHeatmap(mat3, name = "meth2")
# for more examples, please go to the vignette
```

---

EnrichedHeatmap-class *Class for a single heatmap*

---

### Description

Class for a single heatmap

### Details

The [EnrichedHeatmap-class](#) is inherited from [Heatmap-class](#).

### Methods

The [EnrichedHeatmap-class](#) provides following methods:

- [EnrichedHeatmap](#): constructor method.
- [draw,EnrichedHeatmap-method](#): draw a single heatmap.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
# There is no example  
NULL
```

---

EnrichedHeatmapList    *Constructor method for EnrichedHeatmapList class*

---

**Description**

Constructor method for EnrichedHeatmapList class

**Usage**

```
EnrichedHeatmapList(...)
```

**Arguments**

```
...                    arguments
```

**Details**

There is no public constructor method for the [EnrichedHeatmapList-class](#).

**Value**

No value is returned.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
# no example  
NULL
```

---

EnrichedHeatmapList-class  
*Class for a list of heatmaps*

---

### Description

Class for a list of heatmaps

### Details

The [EnrichedHeatmapList-class](#) is inherited from [HeatmapList-class](#).

### Methods

The [EnrichedHeatmapList-class](#) provides following methods:

- [draw, EnrichedHeatmapList-method](#): draw a list of heatmaps.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
# There is no example  
NULL
```

---

enriched\_score      *Enriched scores*

---

### Description

Enriched scores

### Usage

```
enriched_score(x1, x2, x3)
```

### Arguments

x1	a vector corresponding to values in upstream windows
x2	a vector corresponding to values in target windows
x3	a vector corresponding to values in downstream windows

### Details

The function calculates how the signal is enriched in the targets. The score is the sum of values weighted by the reciprocal of the distance to the targets.

Basically, to be a score function which calculates enriched score, it should accept three arguments which are explained in **Arguments** section and return a single value. Rows are sorted decreasingly by the enriched scores.

### Value

A numeric value.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
enriched_score(c(1, 2, 3), c(1, 2, 1), c(3, 2, 1))
enriched_score(c(3, 2, 1), c(2, 1, 2), c(1, 2, 3))
```

---

getSignalsFromList     *Get signals from a list*

---

### Description

Get signals from a list

### Usage

```
getSignalsFromList(lt, fun = function(x) mean(x, na.rm = TRUE))
```

### Arguments

lt	a list of objects which are returned by <a href="#">normalizeToMatrix</a> . Objects in the list should come from same settings.
fun	a self-defined function which gives mean signals across samples. If we assume the objects in the list correspond to different samples, then different regions in the targets are the first dimension, different positions upstream or downstream of the targets are the second dimension, and different samples are the third dimension. This self-defined function can have one argument which is the vector containing values in different samples in a specific position to a specific target region. Or it can have a second argument which is the index for the current target.

## Details

Let's assume you have a list of histone modification signals for different samples and you want to visualize the mean pattern across samples. You can first normalize histone mark signals for each sample and then calculate means values across all samples. In following example code, `hm_gr_list` is a list of `GRanges` objects which contain positions of histone modifications, `tss` is a `GRanges` object containing positions of gene TSS.

```
mat_list = NULL
for(i in seq_along(hm_gr_list)) {
  mat_list[[i]] = normalizeToMatrix(hm_gr_list[[i]], tss, value_column = "density")
}
```

Applying `getSignalsFromList()` to `mat_list`, it gives a new normalized matrix which contains mean signals and can be directly used in `EnrichedHeatmap()`.

```
mat = getSignalsFromList(mat_list)
EnrichedHeatmap(mat)
```

Next let's consider a second scenario: we want to see the correlation between histone modification and gene expression. In this case, `fun` can have a second argument so that users can correspond histone signals to the expression of the associated gene. In following code, `expr` is a matrix of expression, columns in `expr` correspond to elements in `hm_gr_list`, rows in `expr` are same as `tss`.

```
mat = getSignalsFromList(mat_list,
  fun = function(x, i) cor(x, expr[i, ], method = "spearman"))
```

Then `mat` here can be used to visualize how gene expression is correlated to histone modification around TSS.

```
EnrichedHeatmap(mat)
```

## Value

A [normalizeToMatrix](#) object which can be directly used for [EnrichedHeatmap](#).

## Author(s)

Zuguang Gu <z.gu@dkfz.de>

## Examples

```
NULL
```

---

 makeWindows

*Split regions into windows*


---

**Description**

Split regions into windows

**Usage**

```
makeWindows(query, w = NULL, k = NULL, direction = c("normal", "reverse"),
            short.keep = FALSE)
```

**Arguments**

query	a <a href="#">GRanges</a> object.
w	window size, a value larger than 1 means the number of base pairs and a value between 0 and 1 is the percent to the current region.
k	number of partitions for each region. If it is set, all other arguments are ignored.
direction	where to start the splitting. See 'Details' section.
short.keep	if the the region can not be splitted equally under the window size, whether to keep the windows that are smaller than the window size. See 'Details' section.

**Details**

Following illustrates the meaning of direction and short.keep:

```
----->----- one region, split by 3bp window (">" means the direction of the sequence)
aaabbbccc direction = "normal", short.keep = FALSE
aaabbbcccd direction = "normal", short.keep = TRUE
  aaabbbccc direction = "reverse", short.keep = FALSE
  abbbcccddd direction = "reverse", short.keep = TRUE
```

**Value**

A [GRanges](#) object with two additional columns attached:

- `.i_query` which contains the correspondance between small windows and original regions in query
- `.i_window` which contains the index of the small window on the current region.

**Author(s)**

Zuguang gu <z.gu@dkfz.de>

**Examples**

```

query = GRanges(seqnames = "chr1", ranges = IRanges(start = c(1, 11, 21), end = c(10, 20, 30)))
makeWindows(query, w = 2)
makeWindows(query, w = 0.2)
makeWindows(query, w = 3)
makeWindows(query, w = 3, direction = "reverse")
makeWindows(query, w = 3, short.keep = TRUE)
makeWindows(query, w = 3, direction = "reverse", short.keep = TRUE)
makeWindows(query, w = 12)
makeWindows(query, w = 12, short.keep = TRUE)
makeWindows(query, k = 2)
makeWindows(query, k = 3)
query = GRanges(seqnames = "chr1", ranges = IRanges(start = c(1, 11, 31), end = c(10, 30, 70)))
makeWindows(query, w = 2)
makeWindows(query, w = 0.2)

```

---

normalizeToMatrix	<i>Normalize associations between genomic signals and target regions into a matrix</i>
-------------------	--

---

**Description**

Normalize associations between genomic signals and target regions into a matrix

**Usage**

```

normalizeToMatrix(signal, target, extend = 5000, w = max(extend)/50,
  value_column = NULL, mapping_column = NULL, empty_value = ifelse(smooth, NA, 0),
  mean_mode = c("absolute", "weighted", "w0", "coverage"), include_target = any(width(target) > 1),
  target_ratio = ifelse(all(extend == 0), 1, 0.1), k = min(c(20, min(width(target))))),
  smooth = FALSE, smooth_fun = default_smooth_fun, trim = 0)

```

**Arguments**

signal	a <a href="#">GRanges</a> object.
target	a <a href="#">GRanges</a> object.
extend	extended base pairs to the upstream and downstream of target. It can be a vector of length one or two. If it is length one, it means extension to the upstream and downstream are the same.
w	window size for splitting upstream and downstream.
value_column	column index in signal that will be mapped to colors. If it is NULL, an internal column which all contains 1 will be used.
mapping_column	mapping column to restrict overlapping between signal and target. By default it tries to look for all regions in signal that overlap with every target.
empty_value	values for small windows that don't overlap with signal.

mean_mode	when a window is not perfectly overlapped to signal, how to summarize values to this window. See 'Details' section for a detailed explanation.
include_target	whether include target in the heatmap. If the width of all regions in target is 1, include_target is enforced to FALSE.
target_ratio	the ratio of target in the full heatmap. If the value is 1, extend will be reset to 0.
k	number of windows only when target_ratio = 1 or extend == 0, otherwise ignored.
smooth	whether apply smoothing on rows in the matrix.
smooth_fun	the smoothing function that is applied to each row in the matrix. This self-defined function accepts a numeric vector (may contains NA values) and returns a vector with same length. If the smoothing is failed, the function should call <code>stop</code> to throw errors so that <code>normalizeToMatrix</code> can catch how many rows are failed in smoothing. See the default <code>default_smooth_fun</code> for example.
trim	percent of extreme values to remove. IF it is a vector of length 2, it corresponds to the lower quantile and higher quantile. e.g. <code>c(0.01, 0.01)</code> means to trim outliers less than 1st quantile and larger than 99th quantile.

## Details

In order to visualize associations between signal and target, the data is transformed into a matrix and visualized as a heatmap by `EnrichedHeatmap` afterwards.

Upstream and downstream also with the target body are splitted into a list of small windows and overlap to signal. Since regions in signal and small windows do not always 100 percent overlap, there are four different average modes:

Following illustrates different settings for `mean_mode` (note there is one signal region overlapping with other signals):

```

  40      50      20      values in signal
+++++++  +++   +++++  signal
           30      values in signal
           ++++++  signal
=====
  4  6  3      3      window (17bp), there are 4bp not overlapping to any signal region.
           overlap

```

absolute:  $(40 + 30 + 50 + 20)/4$

weighted:  $(40*4 + 30*6 + 50*3 + 20*3)/(4 + 6 + 3 + 3)$

w0:  $(40*4 + 30*6 + 50*3 + 20*3)/(4 + 6 + 3 + 3 + 4)$

coverage:  $(40*4 + 30*6 + 50*3 + 20*3)/17$

To explain it more clearly, let's consider three scenarios:

First, we want to calculate mean methylation from 3 CpG sites in a 20bp window. Since methylation is only measured at CpG site level, the mean value should only be calculated from the 3 CpG sites while not the non-CpG sites. In this case, absolute mode should be used here.

Second, we want to calculate mean coverage in a 20bp window. Let's assume coverage is 5 in 1bp ~ 5bp, 10 in 11bp ~ 15bp and 20 in 16bp ~ 20bp. Since coverage is kind of attribute for



all bases, all 20 bp should be taken into account. Thus, here `w0` mode should be used which also takes account of the 0 coverage in 6bp ~ 10bp. The mean coverage will be calculated as  $(5*5 + 10*5 + 20*5)/(5+5+5+5)$ .

Third, genes have multiple transcripts and we want to calculate how many transcripts exist in a certain position in the gene body. In this case, values associated to each transcript are binary (either 1 or 0) and coverage mean mode should be used.

### Value

A matrix with following additional attributes:

**upstream\_index** column index corresponding to upstream of target

**target\_index** column index corresponding to target

**downstream\_index** column index corresponding to downstream of target

**extend** extension on upstream and downstream

**smooth** whether smoothing was applied on the matrix

**failed\_rows** index of rows which are failed for smoothing

The matrix is wrapped into a simple `normalizeToMatrix` class.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
signal = GRanges(seqnames = "chr1",
  ranges = IRanges(start = c(1, 4, 7, 11, 14, 17, 21, 24, 27),
    end = c(2, 5, 8, 12, 15, 18, 22, 25, 28)),
  score = c(1, 2, 3, 1, 2, 3, 1, 2, 3))
target = GRanges(seqnames = "chr1", ranges = IRanges(start = 10, end = 20))
normalizeToMatrix(signal, target, extend = 10, w = 2)
normalizeToMatrix(signal, target, extend = 10, w = 2, include_target = TRUE)
normalizeToMatrix(signal, target, extend = 10, w = 2, value_column = "score")
```

---

```
print.normalizedMatrix
```

```
Print normalized matrix
```

---

### Description

Print normalized matrix

### Usage

```
## S3 method for class 'normalizedMatrix'
print(x, ...)
```

**Arguments**

x                    the normalized matrix returned by [normalizeToMatrix](#)  
...                   other arguments

**Value**

No value is returned.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
# There is no example  
NULL
```

---

show-dispatch

*Method dispatch page for show*

---

**Description**

Method dispatch page for show.

**Dispatch**

show can be dispatched on following classes:

- [show, EnrichedHeatmapList-method, EnrichedHeatmapList-class](#) class method
- [show, EnrichedHeatmap-method, EnrichedHeatmap-class](#) class method

**Examples**

```
# no example  
NULL
```

---

show-EnrichedHeatmap-method

*Draw the single heatmap with default parameters*

---

### Description

Draw the single heatmap with default parameters

### Usage

```
## S4 method for signature 'EnrichedHeatmap'  
show(object)
```

### Arguments

object            an [EnrichedHeatmap-class](#) object.

### Details

Actually it calls [draw,EnrichedHeatmap-method](#), but only with default parameters. If users want to customize the heatmap, they can pass parameters directly to [draw,EnrichedHeatmap-method](#).

### Value

An [EnrichedHeatmapList-class](#) object.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
# see documentation of EnrichedHeatmap  
NULL
```

---

show-EnrichedHeatmapList-method

*Draw a list of heatmaps with default parameters*

---

### Description

Draw a list of heatmaps with default parameters

### Usage

```
## S4 method for signature 'EnrichedHeatmapList'  
show(object)
```

**Arguments**

object            an [EnrichedHeatmapList-class](#) object.

**Details**

Actually it calls [draw, EnrichedHeatmapList-method](#), but only with default parameters. If users want to customize the heatmap, they can pass parameters directly to [draw, EnrichedHeatmapList-method](#).

**Value**

An [EnrichedHeatmapList-class](#) object.

**Examples**

```
# see documentation of EnrichedHeatmap
NULL
```

---

[.normalizedMatrix     *Subset normalized matrix by rows*

---

**Description**

Subset normalized matrix by rows

**Usage**

```
## S3 method for class 'normalizedMatrix'
x[i, j, drop = FALSE]
```

**Arguments**

x                    the normalized matrix returned by [normalizeToMatrix](#)  
i                    row index  
j                    column index  
drop                whether drop the dimension

**Value**

A [normalizedMatrix](#) class object.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
# There is no example
NULL
```

# Index

`+.AdditiveUnit`, [2](#)  
`[.normalizedMatrix`, [20](#)

`anno_enriched`, [3](#)

`copyAttr`, [4](#)

`default_smooth_fun`, [5](#), [16](#)  
`draw (draw-dispatch)`, [5](#)  
`draw, EnrichedHeatmap-method`  
    (`draw-EnrichedHeatmap-method`),  
    [6](#)  
`draw, EnrichedHeatmapList-method`  
    (`draw-EnrichedHeatmapList-method`),  
    [7](#)  
`draw-dispatch`, [5](#)  
`draw-EnrichedHeatmap-method`, [6](#)  
`draw-EnrichedHeatmapList-method`, [7](#)

`enriched_score`, [8](#), [11](#)  
`EnrichedHeatmap`, [3](#), [8](#), [9](#), [13](#), [16](#)  
`EnrichedHeatmap-class`, [9](#)  
`EnrichedHeatmapList`, [7](#), [10](#)  
`EnrichedHeatmapList-class`, [11](#)

`getSignalsFromList`, [12](#)  
`GRanges`, [14](#), [15](#)

`Heatmap`, [8](#)

`locfit`, [5](#)  
`loess`, [5](#)

`makeWindows`, [14](#)

`normalizeToMatrix`, [4](#), [8](#), [12](#), [13](#), [15](#), [16](#), [18](#),  
    [20](#)

`print.normalizedMatrix`, [17](#)

`show (show-dispatch)`, [18](#)  
`show, EnrichedHeatmap-method`  
    (`show-EnrichedHeatmap-method`),  
    [19](#)  
`show, EnrichedHeatmapList-method`  
    (`show-EnrichedHeatmapList-method`),  
    [19](#)  
`show-dispatch`, [18](#)  
`show-EnrichedHeatmap-method`, [19](#)  
`show-EnrichedHeatmapList-method`, [19](#)  
`stop`, [16](#)