Package ‘EnrichedHeatmap’

Type Package
Title Making Enriched Heatmaps
Version 1.4.0
Date 2016-9-26
Author Zuguang Gu
Maintainer Zuguang Gu <z.gu@dkfz.de>
Depends R (>= 3.1.2), grid, ComplexHeatmap (>= 1.11.1), 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 correspondence between different data sources.

Bioconductor Software, Visualization, Sequencing, GenomeAnnotation, Coverage
URL https://github.com/jokergoo/EnrichedHeatmap
License GPL (>= 2)
Repository Bioconductor
Date/Publication 2016-9-26 00:00:00
NeedsCompilation no

R Topics Documented:

<table>
<thead>
<tr>
<th>R Topics</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>+.AdditiveUnit</td>
<td>2</td>
</tr>
<tr>
<td>anno_enriched</td>
<td>3</td>
</tr>
<tr>
<td>copyAttr</td>
<td>4</td>
</tr>
<tr>
<td>default_smooth_fun</td>
<td>4</td>
</tr>
<tr>
<td>draw-dispatch</td>
<td>5</td>
</tr>
<tr>
<td>draw-EnrichedHeatmap-method</td>
<td>5</td>
</tr>
</tbody>
</table>
Description

Add heatmaps or row annotations to a heatmap list

Usage

```r
## S3 method for class 'AdditiveUnit'

x + y
```

Arguments

- `x`: an `EnrichedHeatmap-class` object, a `Heatmap-class` object, a `HeatmapAnnotation-class` object or a `HeatmapList-class` object.
- `y`: an `EnrichedHeatmap-class` object, a `Heatmap-class` object, a `HeatmapAnnotation-class` object or a `HeatmapList-class` 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

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

**Description**

Annotation function to show the enrichment

**Usage**

```r
anno_enriched(gp = gpar(col = "red"), pos_line = TRUE, pos_line_gp = gpar(lty = 2),
yaxis = TRUE, ylim = NULL, value = c("mean", "sum", "abs_mean", "abs_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 ±1 se to the mean value. Color of error area is same as the corresponding lines with 75 percent transparency.

**Details**

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

If rows are split, 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**

```r
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.
draw-dispatch

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```r
# 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**

```r
# no example
NULL
```

draw-EnrichedHeatmap-method

Draw a single heatmap

**Description**

Draw a single heatmap

**Usage**

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

**Arguments**

- `object` an EnrichedHeatmap-class object.
- `internal` only used internally.
- `...` pass to draw,HeatmapList-method.
### draw-EnrichedHeatmapList-method

**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**

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

### draw-EnrichedHeatmapList-method

*Draw a list of heatmaps*

**Description**

Draw a list of heatmaps

**Usage**

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

**Arguments**

- `object`: an `EnrichedHeatmapList-class` object
- `padding`: padding of the plot. Elements correspond to bottom, left, top, right paddings.
- `newpage`: whether to create a new page
- `...`: pass to `make_layout,HeatmapList-method` or `draw,HeatmapList-method`

**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

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

---

**EnrichedHeatmap**  
Constructor method for EnrichedHeatmap class

**Description**

Constructor method for EnrichedHeatmap class

**Usage**

```r
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 `normalizeToMatrix`
- `score_fun`: a function which calculates enriched scores for rows in `mat`. This function can be self-defined, take a look at `enriched_score` 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 `Heatmap`

**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
**EnrichedHeatmapList**

**Examples**

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

---

**EnrichedHeatmapList**  
*Constructor method for EnrichedHeatmapList class*

**Description**

Constructor method for EnrichedHeatmapList class

**Usage**

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

**Arguments**

```r
... 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**

```r
# 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

----------

<table>
<thead>
<tr>
<th>enriched_score</th>
<th>Enriched scores</th>
</tr>
</thead>
</table>

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 normalizeToMatrix. 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 mean 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.

```r
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().

```r
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.

```r
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.

```r
EnrichedHeatmap(mat)
```

**Value**

A `normalizeToMatrix` object which can be directly used for `EnrichedHeatmap`.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

`NULL`

### makeWindows

**Description**

Split regions into windows

**Usage**

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

**Arguments**

- `query`: a `GRanges` 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 region cannot be split 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
- `aaabbbcccc` `direction = "normal", short.keep = FALSE`
- `aaabbbcccdd` `direction = "normal", short.keep = TRUE`
- `aaabbbcccc` `direction = "reverse", short.keep = FALSE`
- `abbbcccddeee` `direction = "reverse", short.keep = TRUE`
**normalizeToMatrix**

**Value**

A `GRanges` object with two additional columns attached:

- `.i_query` which contains the correspondence 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**

```r
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 = 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)
```

**Description**

Normalize associations between genomic signals and target regions into a matrix.

**Usage**

```r
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 `GRanges` object.
- `target` a `GRanges` 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.
normalizeToMatrix

w

value_column

mapping_column

empty_value

mean_mode

include_target

target_ratio

k

smooth

smooth_fun

trim

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

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

weighted: \((40\times4 + 30\times6 + 50\times3 + 20\times3)/(4 + 6 + 3 + 3)\)

w0: \((40\times4 + 30\times6 + 50\times3 + 20\times3)/(4 + 6 + 3 + 3 + 4)\)

coverage: \((40\times4 + 30\times6 + 50\times3 + 20\times3)/17\)

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 \( \text{w0} \) mode should be used which also takes account of the 0 coverage in 6bp ~ 10bp. The mean coverage will be calculated as \( (5 \times 5 + 10 \times 5 + 20 \times 5)/(5 + 5 + 5) \).

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**

```r
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**

```r
## 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 \texttt{draw,EnrichedHeatmapList-method}, but only with default parameters. If users want to customize the heatmap, they can pass parameters directly to \texttt{draw,EnrichedHeatmapList-method}.

Value

An \texttt{EnrichedHeatmapList-class} object.

Examples

# see documentation of EnrichedHeatmap
NULL

\[ .normalizedMatrix \]

\textit{Subset normalized matrix by rows}

Description

Subset normalized matrix by rows

Usage

\[
\texttt{\# S3 method for class 'normalizedMatrix'} \\
\texttt{x[i, j, drop = FALSE]}
\]

Arguments

\begin{itemize}
  \item \texttt{x} the normalized matrix returned by \texttt{normalizeToMatrix}
  \item \texttt{i} row index
  \item \texttt{j} column index
  \item \texttt{drop} whether drop the dimension
\end{itemize}

Value

A \texttt{normalizedMatrix} class object.

Author(s)

Zuguang Gu \texttt{<z.gu@dkfz.de>}

Examples

# There is no example
NULL
Index

+.AdditiveUnit, 2
[].normalizedMatrix, 18
anno_enriched, 3
copyAttr, 4
default_smooth_fun, 4, 14
draw (draw-dispatch), 5
draw,EnrichedHeatmap-method
  (draw-EnrichedHeatmap-method), 5
draw,EnrichedHeatmapList-method
  (draw-EnrichedHeatmapList-method), 6
draw-dispatch, 5
draw-EnrichedHeatmap-method, 5
draw-EnrichedHeatmapList-method, 6
enriched_score, 7, 10
EnrichedHeatmap, 3, 7, 8, 12, 14
EnrichedHeatmap-class, 8
EnrichedHeatmapList, 6, 9
EnrichedHeatmapList-class, 9
getSignalsFromList, 11
GRanges, 12, 13
Heatmap, 7
locfit, 4
loess, 4
makeWindows, 12
normalizeToMatrix, 4, 7, 11, 12, 13, 14, 16, 18
print.normalizedMatrix, 15
show (show-dispatch), 16
show, EnrichedHeatmap-method
  (show-EnrichedHeatmap-method), 17
show-dispatch, 16
show-EnrichedHeatmap-method, 17
show-EnrichedHeatmapList-method, 17
stop, 14

19