Package ‘GMD’

February 19, 2015

Type Package

Version 0.3.3

Date 2014-08-26 15:46:45 EDT

Title Generalized Minimum Distance of distributions

Description GMD is a package for non-parametric distance measurement between two discrete frequency distributions.

License GPL (>= 2)

LazyLoad TRUE

NeedsCompilation no

URL http://CRAN.R-project.org/package=GMD

Repository CRAN

Depends R (>= 3.1.0)

Imports stats, grDevices, gplots

Suggests datasets, MASS, cluster

Collate 'GMD-package.R' 'imports.R' 'util.R' 'GMD-internal.R'
 'GMD-data.R' 'ghist.R' 'gdist.R' 'css.R' 'heatmap3.R' 'gmdm.R'

Maintainer Xiaobei Zhao <xiaobei@binf.ku.dk>

Author Xiaobei Zhao [aut, cre, cph],
  Albin Sandelin [aut]

Date/Publication 2014-08-27 14:29:53

R topics documented:

  GMD-package .........................................................  2
  bedgraph.to.depth ...............................................  4
  cage ...............................................................  4
  chipseq ............................................................  5
  css .................................................................  6
  elbow ...............................................................  7
The Package for Generalized Minimum Distance (GMD) Computation

Description

Compute Generalized Minimum Distance (GMD) between discrete distributions and clustering tools.

Details

Package: GMD
Type: Package
License: GPL (>= 2)

This package contains:
1) modules and functions for GMD computation, with GMD algorithm implemented in C to interface with R.
2) related clustering and visualization tools for distributions.

An overview of functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ghist</td>
<td>Generalized Histogram Computation and Visualization</td>
</tr>
<tr>
<td>gdist</td>
<td>Generalized Distance Matrix Computation</td>
</tr>
<tr>
<td>css</td>
<td>Computing Clustering Sum-of-Squares and evaluating the clustering by the “elbow” method</td>
</tr>
<tr>
<td>heatmap.3</td>
<td>Enhanced Heatmap Representation with Dendrogram and Partition</td>
</tr>
<tr>
<td>gmdp</td>
<td>Computation of GMD on a pair of histograms</td>
</tr>
<tr>
<td>gmdm</td>
<td>Computation of GMD Matrix on a set of histograms</td>
</tr>
</tbody>
</table>
To install from online repositories (e.g. CRAN) install.packages(pkgs="GMD", repos="http://cran.r-project.org")

Sometimes the official repository might not be up to date, then you may install it from a downloaded source file; please replace 'current-version' with actual version numbers: Note that as new versions are release, the 'current-version' changes. install.packages(pkgs="GMD_<current-version>.tar.gz", repos=NULL)

Load the package and get a complete list of functions, use library(GMD) ls("package:GMD")

help documentation of the package help(GMD) # this page

Value

NULL

Author(s)

Xiaobei Zhao and Albin Sandelin
Maintainer: Xiaobei Zhao <xiaobei (at) biinf.ku.dk>

References


http://dx.plos.org/10.1371/journal.pone.0023409

See citation("GMD") for BibTeX entries for LaTeX users.

See Also

gmdp, gmdm, cage, chipseq, ghist, gdist, css, elbow, heatmap

Examples

## Not run:
require(GMD) # load GMD
help(GMD) # a help document of GMD
data(package="GMD") # a list of datasets available in GMD
ls("package:GMD") # a list of functions available in GMD
help(package="GMD") # help documentation on GMD
citation("GMD") # citation for publications
demo("GMD-demo") # run the demo

## view GMD Description
packageDescription("GMD")

## view GMD vignette
vignette("GMD-vignette",package="GMD")

## End(Not run)
bedgraph.to.depth  

*Convert a BedGraph file to a vector of depth-like signals*

**Description**

Convert a BedGraph file to a vector of depth-like signals

**Usage**

```
bedgraph.to.depth(infpath, chr, start, end, reverse = FALSE)
```

**Arguments**

- `infpath` character, the path of a bedgraph file
- `chr` character, the chromosome
- `start` numeric, the start position (0-base)
- `end` numeric, the end position
- `reverse` logical, whether a reverse strand

**Value**

numeric

**Author(s)**

Xiaobei Zhao

---

cage  

*CAGE Data: Transcription Start Site Distributions (TSSDs) by CAGE tags*

**Description**

Transcription Start Site Distributions (TSSDs) by CAGE tags.

**Usage**

```
cage
cagel
```

**Details**

cage is a list of 20 named TSSDs. cagel is a longer version of *cage*, with 50 named TSSDs.
References


http://dx.plos.org/10.1371/journal.pone.0023409

See Also

gmdp and gmdm, with examples using cage. chipseq for histone marks by ChIP-seq reads.

Examples

```r
cage

```
See Also

gmdp and gmdm, with examples using chipseq. cage for data of Transcription Start Sites (TSSs) by CAGE tags.

Examples

```r
require(GMD)
help(chipseq)
data(chipseq_mES)
class(chipseq_mES)
length(chipseq_mES)
names(chipseq_mES)
## Not run: chipseq_mES
data(chipseq_hCD4T)
names(chipseq_hCD4T)
```

---

**css**

*Clustering Sum-of-Squares for clustering evaluation*

**Description**

Evaluation on the variance of a clustering model using squared Euclidean distances, based on distance matrix and cluster membership.

**Usage**

```r
css(dist.obj,clusters)
```

***## Computing Sum-of-Squares given Hierarchical Clustering
## S3 method for class 'hclust'
css(dist.obj, hclust.obj=NULL,hclust.FUN=hclust,
hclust.FUN.MoreArgs=list(method="ward"),k=NULL)***

**Arguments**

- `dist.obj` a 'dist' object as produced by dist or gdist.
- `clusters` a vector with cluster memberships.
- `k` numeric, the upper bound of the number of clusters to compute. DEFAULT: 20 or the number of observations (if less than 20).
- `hclust.obj` a 'hclust' object, generated by hclust.
- `hclust.FUN` a function, to generate a hierarchical clustering. Ignored with hclust.obj specified. DEFAULT: hclust
- `hclust.FUN.MoreArgs` a list, containing arguments that are passed to hclust.FUN.
elbow

Details

Clustering Sum-of-Squares for clustering evaluation.

Value

css returns a ‘css’ object, which is a list containing the following components

\[
\begin{align*}
\text{k} & : \text{number of clusters} \\
\text{wss} & : k \text{ within-cluster sum-of-squares} \\
\text{totwss} & : \text{total within-cluster sum-of-square} \\
\text{totbss} & : \text{total between-cluster sum-of-square} \\
\text{tss} & : \text{total sum of squares of the data}
\end{align*}
\]

, and with an attribute ‘meta’ that contains the input components

\[
\begin{align*}
\text{dist.obj} & : \text{(the input) distance matrix} \\
\text{clusters} & : \text{(the input) cluster membership}
\end{align*}
\]

css Nhclust returns a ‘css.multi’ object, which is a data.frame containing the following columns

\[
\begin{align*}
\text{k} & : \text{number of clusters} \\
\text{ev} & : \text{explained variance given k} \\
\text{totbss} & : \text{total between-cluster sum-of-square} \\
\text{tss} & : \text{total sum of squares of the data}
\end{align*}
\]

, and with an attribute ‘meta’ that contains

\[
\begin{align*}
\text{cmethod} & : \text{the clustering method} \\
\text{dist.obj} & : \text{(the input) distance matrix} \\
\text{k} & : \text{(the input) number of clusters} \\
\text{clusters} & : \text{the ‘hclust’ object that is either by input or computed by default}
\end{align*}
\]

See Also

elbow for "elbow" plot using ‘css.multi’ object

elbow

The "Elbow" Method for Clustering Evaluation

Description

Determining the number of clusters in a data set by the "elbow" rule.
Usage

```r
## find a good k given thresholds of EV and its increment.
e1bow(x, inc.thres, ev.thres, precision=3, print.warning=TRUE)
## a wrapper of `elbow' testing multiple thresholds.
e1bow.batch(x, inc.thres=c(0.01, 0.05, 0.1),
ev.thres=c(0.95, 0.9, 0.8, 0.75, 0.67, 0.5, 0.33), precision=3)
## S3 method for class 'elbow'
plot(x, elbow.obj=NULL, main, xlab, ylab, type, pch=20, col.abline="red",
lty.abline=3, if.plot.new=TRUE, print.info=TRUE,
mar=c(4, 5, 3, 3), omi=c(0.75, 0, 0, 0), ...)
```

Arguments

- `x`: a `css.multi` object, generated by `css.hclust`
- `inc.thres`: numeric with value(s) from 0 to 1, the threshold of the increment of EV. A single value is used in `elbow` while a vector of values in `elbow.batch`.
- `ev.thres`: numeric with value(s) from 0 to 1, the threshold of EV. A single value is used in `elbow` while a vector of values in `elbow.batch`.
- `precision`: integer, the number of digits to round for numerical comparison.
- `print.warning`: logical, whether to print warning messages.
- `elbow.obj`: a `elbow` object, generated by `elbow` or `elbow.batch`.
- `main`: an overall title for the plot.
- `ylab`: a title for the y axis.
- `xlab`: a title for the x axis.
- `type`: what type of plot should be drawn. See `help("plot", package="graphics")`.
- `pch`: Either an integer specifying a symbol or a single character to be used as the default in plotting points (see `par`).
- `col.abline`: color for straight lines through the current plot (see option `col` in `par`).
- `lty.abline`: line type for straight lines through the current plot (see option `lty` in `par`).
- `if.plot.new`: logical, whether to start a new plot device or not.
- `print.info`: logical, whether to print the information of `elbow.obj`.
- `mar`: A numerical vector of the form `c(bottom, left, top, right)` which gives the number of lines of margin to be specified on the four sides of the plot (see option `mar` in `par`). The default is `c(4, 5, 3, 3) + 0.1`.
- `omi`: A vector of the form `c(bottom, left, top, right)` giving the size of the outer margins in inches (see option `omi` in `par`).
- `...`: arguments to be passed to method `plot.elbow`, such as graphical parameters (see `par`).
Details

Determining the number of clusters in a data set by the "elbow" rule and thresholds in the explained variance (EV) and its increment.

Value

Both elbow and elbow.batch return a ‘elbow’ object (if a "good" k exists), which is a list containing the following components

- **k**: number of clusters
- **ev**: explained variance given k
- **inc.thres**: the threshold of the increment in EV
- **ev.thres**: the threshold of the EV

, and with an attribute 'meta' that contains

- **description**: A description about the "good" k

See Also

css and css.hclust for computing Clustering Sum-of-Squares.

Examples

```r
## load library
require("GMD")

## simulate data around 12 points in Euclidean space
pointv <- data.frame(x=c(1,2,2,4,4,5,6,7,8,9,9),
y=c(1,2,8,2,4,4,5,9,9,8,1,9))
set.seed(2012)
mydata <- c()
for (i in 1:nrow(pointv)){
  mydata <- rbind(mydata,cbind(rnorm(10,pointv[i,1],0.1),
  rnorm(10,pointv[i,2],0.1)))
}
mydata <- data.frame(mydata); colnames(mydata) <- c("x","y")
plot(mydata,type="p",pch=21, main="Simulated data")

## determine a "good" k using elbow
dist.obj <- dist(mydata[,1:2])
hclust.obj <- hclust(dist.obj)
css.obj <- css.hclust(dist.obj,hclust.obj)
elbow.obj <- elbow.batch(css.obj)
print(elbow.obj)

## make partition given the "good" k
k <- elbow.obj$k; cutree.obj <- cutree(hclust.obj,k=k)
```
mydata$cluster <- cutree.obj

## draw a elbow plot and label the data
dev.new(width=12, height=6)
par(mfcol=c(1,2),mar=c(4,5,3,3),omi=c(0.75,0,0,0))
plot(mydata$x,mydata$y,pch=as.character(mydata$cluster),
      col=mydata$cluster,cex=0.75,main="Clusters of simulated data")
plot.css.obj,elbow.obj,if.plot.new=FALSE)

## clustering with more relaxed thresholds (, resulting a smaller "good" k)
elbow.obj2 <- elbow.batch(css.obj,ev.thres=0.90,inc.thres=0.05)
mydata$cluster2 <- cutree(hclust.obj,k=elbow.obj2$k)
dev.new(width=12, height=6)
par(mfcol=c(1,2), mar=c(4,5,3,3),omi=c(0.75,0,0,0))
plot(mydata$x,mydata$y,pch=as.character(mydata$cluster2),
      col=mydata$cluster2,cex=0.75,main="Clusters of simulated data")
plot(css.obj,elbow.obj2,if.plot.new=FALSE)

equalize.list Make members of a list equal size

Description

Make member bins of a hist object equal size

Usage

equalize.list(x)

Arguments

x a list of numeric vectors

Details

Make members of a list equal size

gdist Generalized Distance Matrix Computation

Description

gdist computes and returns the distance matrix computed by using user-defined distance measure.
Usage

gdist(x, method = "euclidean", MoreArgs = NULL, diag = FALSE, upper = FALSE)

is.dist(d)

Arguments

x a numeric matrix, data frame or ‘dist’ object.

method the distance measure to be used. This can either be one of the methods used in
dist (see help("dist", package = "stats")) or "correlation", "correlation.of.observations" and "correlation.of.variables". In addition, user-defined distance mea-
sure are also allowed, which returns a dist object and should at least have at-
tributes "Size" and "Labels".

MoreArgs a list of other arguments to be passed to gdist.

dia logical value indicating whether the diagonal of the distance matrix should be

upper logical value indicating whether the upper triangle of the distance matrix should be

Value

gdist returns an object of ‘dist’.

is.dist returns a logical value whether an object is ‘dist’.

Examples

## load library
require("GMD")
require(cluster)

## compute distance using Euclidean metric (default)
data(ruspini)
x <- gdist(ruspini)

## see a dendrogram result by hierarchical clustering
dev.new(width=12, height=6)
plot(hclust(x),
```
main="Cluster Dendrogram of Ruspini data",
xlab="Observations"

## convert to a distance matrix
m <- as.matrix(x)

## convert from a distance matrix
d <- as.dist(m)
stopifnot(d == x)

## Use correlations between variables "as distance"
data(USJudgeRatings)

dd <- gdist(x=USJudgeRatings,method="correlation.of.variables")
dev.new(width=12, height=6)
plot(hclust(dd),
     main="Cluster Dendrogram of USJudgeRatings data",
xlab="Variables")
```

---

#### get.sep

*Get row or column lines of separation for heatmap.3*

**Description**

Get row or column lines of separation for heatmap.3 according to clusters

**Usage**

```r
get.sep(clusters, type = c("row", "column", "both"))
```

**Arguments**

- **clusters**: a numerical vector, indicating the cluster labels of observations.
- **type**: string, one of the following: c("row","column","both")

**Details**

Get row or column lines of separation for heatmap.3 according to clusters
**ghist**  

*Generalized Histogram Computation*

**Description**

Generalized Histogram Computation with classes to contain a single histogram or multiple histograms.

**Usage**

```r
ghist(data, n, breaks = if (!.invalid(n)) NULL else "Sturges",
      bins = NULL, digits = 1)
```

```r
gbreaks(data, n)
```

```r
is.ghist(x)
```

```r
as.ghist(x, bins)
```

```r
is.mhist(x)
```

```r
as.mhist(x, bins)
```

```r
mhist2matrix(h)
```

**Arguments**

- `data` a vector of values for which the histogram is desired.
- `n` a single number giving the number of bins for the histogram.
- `breaks` a vector giving the breakpoints between histogram bins, or a character string naming an algorithm to compute the number of bins, or a function to compute the number of bins (see help("dist", package="graphics").
- `bins` character vector, the bin labels.
- `digits` integer, the number of digits to round for breaks.
- `x` an R object.
- `h` an object of class `mhist`

**Details**

- `ghist` generates a single histogram.
- `gbreaks` generate bin boundaries for a histogram.
- `is.ghist` returns TRUE if `x` is an object of class `ghist` and FALSE otherwise.
- `as.ghist` is a generic function. The method for numeric vectors will return a `ghist` object.
- `is.mhist` returns TRUE if `x` is an object of class `mhist` and FALSE otherwise.
as.mhist is a generic function. The method is for numeric list, matrices or data frames and will return a mhist object.

mhist2matrix convert a mhist object into a numeric matrix, filling observations by row.

See Also

plot.mhist mhist.summary plot.mhist.summary

Examples

## load library
require("GMD")

## create two normally-distributed samples
## with unequal means and unequal variances
set.seed(2012)
v1 <- rnorm(1000, mean=-5, sd=10)
v2 <- rnorm(1000, mean=10, sd=5)

## create common bins
n <- 20 # desired number of bins
breaks <- gbreaks(c(v1,v2),n) # bin boundaries
x <-
  list(ghist(v1,breaks=breaks,digits=0),
       ghist(v2,breaks=breaks,digits=0))
mhist.obj <- as.mhist(x)

---

**gmdm**

*Generalized Minimum Distance Matrix*

**Description**

Computing Generalized Minimum Distance Matrix

**Usage**

```r
gmdm(data, labels, pseudocount=0, sliding=TRUE, resolution=1)
```

## S3 method for class 'gmdm'

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

## convert a `gmdm` object into a `dist` object

```r
gmdm2dist(m, diag=FALSE, upper=FALSE)
```

## compute GMDM and convert into a `dist` object

```r
gmdm_dist(data, diag=FALSE, upper=FALSE, ...)
```
Arguments

- **data**: a list of numeric vectors, a numeric matrix or data.frame
- **x**: a gmdm object.
- **m**: a gmdm object.
- **labels**: a character vector of the same length of `x`, giving the names of the numeric vectors.
- **pseudocount**: a numeric value to be allocated for each position to reduce bias; by default `pseudocount = 0`.
- **sliding**: logical, indicating whether sliding is allowed or not for an optimal solution; by default `sliding = TRUE`.
- **resolution**: relative resolution, numeric (>=1), changing the size of the bin by multiplying the value. A larger value (lower resolution) is more computational efficient but missing details.
- **diag**: logical value indicating whether the diagonal of the distance matrix should be printed by `print.dist`.
- **upper**: logical value indicating whether the upper triangle of the distance matrix should be printed by `print.dist`.
- **...**: arguments to be passed to method

Details

Computing Generalized Minimum Distance Matrix

Value

- **gmdm** returns an object of class `gmdm`, a list with components
- **labels**: a string vector, giving the names of distributions
- **data.ori**: a list of numeric vectors, giving the original input
- **data**: a list of numeric vectors, giving the normalized version of the original input
- **dm**: a numeric numeric, the pairwise distance matrix of GM-Distances
- **gap.pair**: a numeric matrix, giving the gap pair of each alignment per row: i.e. relative shifts between distributions of the optimal hit
- **sliding**: logical, indicating whether sliding is performed
- **pseudocount**: a numeric value that is allocated at each position in addition to original values

References

See `citation("GMD")`

See Also

- `plot.gmdm`, `gmdp`
Generalized Minimum Distance between a pair of distributions

Description

Generalized Minimum Distance between a pair of distributions

Usage

```r
gmdp(v1, v2, labels=c("v1","v2"), pseudocount=0, sliding=TRUE, resolution=1)
```

## S3 method for class 'gmdp'

```r
print(x, mode=c("brief","detailed","full"), digits=3, ...)
```

## S3 method for class 'gmdp'

```r
summary(object, ...)
```

Arguments

- `v1`: a numeric vector, giving positional counts as a discrete distribution.
- `v2`: a numeric vector, giving positional counts as a discrete distribution.
- `labels`: a string vector of length 2, giving the names of v1 and v2 respectively.
- `pseudocount`: a numeric value to be allocated for each position to reduce bias; by default `pseudocount = 0`.
- `sliding`: logical, indicating whether sliding is allowed or not for an optimal solution; by default `sliding = TRUE`.
- `resolution`: relative resolution, numeric (>=1), changing the size of the bin by multiplying the value. A larger value (lower resolution) is more computational efficient but missing details.
- `x`: an object of class `gmdp`.
- `object`: an object of class `gmdp`.
- `mode`: a string of the following: `c("brief","detailed","full")`, indicating whether to print in `full` mode (`default`).
- `digits`: integer, indicating the number of decimal places to be printed.
- `...`: arguments to be passed to method.

Details

Generalized Minimum Distance between a pair of distributions
Value

gmdp returns an object of class gmdp, a numeric with an attribute of meta in a list with components:
labels: a string vector, giving the names of distributions
v1.ori: a numeric vector, the first input distribution
v2.ori: a numeric vector, the second input distribution
v1: a numeric vector, the normalized version of the first input distribution
v2: a numeric vector, the normalized version of the second input distribution
distance: numeric, the GM-Distance (GMD)
sliding: logical, indicating whether sliding is performed
pseudocount: a numeric value that is allocated at each position in addition to original values
gap.pair: a numeric matrix, giving one gap pair per row: i.e. relative shifts between distributions of one optimal hit
n.hit: numeric, the number of (equally good) optimal hits

References

See citation("GMD")

See Also

print.gmdp, summary.gmdp, plot.gmdp gmdm

Examples

require(GMD)
gmdp(c(4,1,1,0,0,0,3,1),c(2,1,1,0,0,0,3,3),sliding=FALSE)
x <- gmdp(c(4,1,1,0,0,0,3,1), c(1,1,2,1,1,0,0,0,3,3,5,5),
pseudocount=1, sliding=TRUE)
print(x)
print(x, "full")

Description

Enhanced heatmap representation with dendrograms and partition given the elbow criterion or a desired number of clusters.
1) a dendrogram added to the left side and to the top, according to cluster analysis;
2) partitions in highlighted rectangles, according to the "elbow" rule or a desired number of clusters.
Usage

heatmap.3(x, diss = inherits(x, "dist"), Rowv = TRUE, Colv = TRUE,
dendrogram = c("both", "row", "column", "none"), dist.row, dist.col,
dist.FUN = gdist, dist.FUN.MoreArgs = list(method = "euclidean"),
hclust.row, hclust.col, hclust.FUN = hclust,
hclust.FUN.MoreArgs = list(method = "ward"), scale = c("none", "row",
"column"), na.rm = TRUE, cluster.by.row = FALSE, cluster.by.col = FALSE,
kr = NA, kc = NA, row.clusters = NA, col.clusters = NA,
revR = FALSE, revC = FALSE, add.expr, breaks, x.center,
color.FUN = gplots::bluered, sepList = list(NULL, NULL),
sep.color = c("gray45", "gray50"), sep.lty = 1, sep.lwd = 2, cellnote,
cex.note = 1, notecol = "cyan", na.color = par(bg),
trace = c("none", "column", "row", "both"), tracecol = "cyan", hline,
vline, linecol = tracecol, labRow = TRUE, labCol = TRUE,
srtRow = NULL, srtCol = NULL, sideRow = 4, sideCol = 1,
margin.for.labRow, margin.for.labCol, ColIndividualColors,
RowIndividualColors, cexRow, cexCol, labRow.by.group = FALSE,
labCol.by.group = FALSE, key = TRUE, key.title = "Color Key",
key.xlab = "Value", key.ylab = "Count", keysize = 1.5, mapsize = 9,
mapratio = 4/3, sidesize = 3, cex.key.main = 0.75,
cex.key.xlab = 0.75, cex.key.ylab = 0.75, density.info = c("histogram",
"density", "none"), denscol = tracecol, densadj = 0.25,
main = "Heatmap", sub = "", xlab = "", ylab = "", cex.main = 2,
cex.sub = 1.5, font.main = 2, font.sub = 3, adj.main = 0.5,
mgp.main = c(1.5, 0.5, 0), mar.main = 3, mar.sub = 3, if.plot = TRUE,
plot.row.partition = FALSE, plot.col.partition = FALSE,
cex.partition = 1.25, color.partition.box = "gray45",
color.partition.border = "#FFFFFF", plot.row.individuals = FALSE,
plot.col.individuals = FALSE, plot.row.clusters = FALSE,
plot.col.clusters = FALSE, plot.row.clustering = FALSE,
plot.col.clustering = FALSE, plot.row.individuals.list = FALSE,
plot.col.individuals.list = FALSE, plot.row.clusters.list = FALSE,
plot.col.clusters.list = FALSE, plot.row.clustering.list = FALSE,
plot.col.clustering.list = FALSE, row.data = FALSE, col.data = FALSE,
if.plot.info = FALSE, text.box, cex.text = 1, ...
)

Arguments

x
  data matrix or data frame, or dissimilarity matrix or 'dist' object determined by
  the value of the 'diss' argument. ##diss logical flag: if TRUE (default for dist
  or dissimilarity objects), then x is assumed to be a dissimilarity matrix. If
  FALSE, then x is treated as a matrix of observations by variables.

diss
  logical, whether the x is a dissimilarity matrix

Rowv
  one of the following: TRUE, a 'dend' object, a vector or NULL/FALSE; determines
  if and how the row dendrogram should be reordered.

Colv
  one of the following: "Rowv", TRUE, a 'dend' object, a vector or NULL/FALSE;
determines if and how the column dendrogram should be reordered.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dendrogram</td>
<td>character string indicating whether to draw 'none', 'row', 'column' or 'both' dendrograms. Defaults to 'both'.</td>
</tr>
<tr>
<td>dist.row</td>
<td>a dist object for row observations.</td>
</tr>
<tr>
<td>dist.col</td>
<td>a dist object for column observations.</td>
</tr>
<tr>
<td>dist.FUN</td>
<td>function used to compute the distance (dissimilarity) between both rows and columns. Defaults to gdist.</td>
</tr>
<tr>
<td>dist.FUN.MoreArgs</td>
<td>a list of other arguments to be passed to gdist.</td>
</tr>
<tr>
<td>hclust.row</td>
<td>a hclust object (as produced by hclust) for row observations.</td>
</tr>
<tr>
<td>hclust.col</td>
<td>a hclust object (as produced by hclust) for column observations.</td>
</tr>
<tr>
<td>hclust.FUN</td>
<td>function used to compute the hierarchical clustering when &quot;Rowv&quot; or &quot;Colv&quot; are not dendrograms. Defaults to hclust.</td>
</tr>
<tr>
<td>hclust.FUN.MoreArgs</td>
<td>a list of other arguments to be passed to hclust.</td>
</tr>
<tr>
<td>scale</td>
<td>character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is &quot;none&quot;.</td>
</tr>
<tr>
<td>na.rm</td>
<td>logical, whether NA values will be removed when scaling.</td>
</tr>
<tr>
<td>cluster.by.row</td>
<td>logical, whether to cluster row observations and reorder the input accordingly.</td>
</tr>
<tr>
<td>cluster.by.col</td>
<td>logical, whether to cluster column observations and reorder the input accordingly.</td>
</tr>
<tr>
<td>kr</td>
<td>numeric, number of clusters in rows; suppressed when row.clusters is specified. DEFAULT: NULL.</td>
</tr>
<tr>
<td>kc</td>
<td>numeric, number of clusters in columns; suppressed when col.clusters is specified. DEFAULT: NULL.</td>
</tr>
<tr>
<td>row.clusters</td>
<td>a numerical vector, indicating the cluster labels of row observations.</td>
</tr>
<tr>
<td>col.clusters</td>
<td>a numerical vector, indicating the cluster labels of column observations.</td>
</tr>
<tr>
<td>revR</td>
<td>logical indicating if the row order should be 'rev'ersed for plotting.</td>
</tr>
<tr>
<td>revC</td>
<td>logical indicating if the column order should be 'rev'ersed for plotting, such that e.g., for the symmetric case, the symmetry axis is as usual.</td>
</tr>
<tr>
<td>add.expr</td>
<td>expression that will be evaluated after the call to image. Can be used to add components to the plot.</td>
</tr>
<tr>
<td>breaks</td>
<td>numeric, either a numeric vector indicating the splitting points for binning x into colors, or a integer number of break points to be used, in which case the break points will be spaced equally between range(x). DEFAULT: 16 when not specified.</td>
</tr>
<tr>
<td>x.center</td>
<td>numeric, a value of x for centering colors to</td>
</tr>
<tr>
<td>color.FUN</td>
<td>function or function name in characters, for colors in the heatmap</td>
</tr>
<tr>
<td>sepList</td>
<td>a list of length 2 giving the row and column lines of separation.</td>
</tr>
<tr>
<td>sep.color</td>
<td>color for lines of separation.</td>
</tr>
<tr>
<td>sep.lty</td>
<td>line type for lines of separation.</td>
</tr>
</tbody>
</table>
sep.lwd  
line width for lines of separation.

cellnote  
(optional) matrix of character strings which will be placed within each color cell, e.g. cell labels or p-value symbols.

cex.note  
relative font size of cellnote.

notecol  
color of cellnote.

na.color  
Color to use for missing value (NA). Defaults to the plot background color.

trace  
character string indicating whether a solid "trace" line should be drawn across "row"s or down "column"s, "both" or "none". The distance of the line from the center of each color-cell is proportional to the size of the measurement. Defaults to "none".

tracecol  
character string giving the color for "trace" line. Defaults to "cyan";

hline  
Vector of values within cells where a horizontal dotted line should be drawn. only plotted if 'trace' is 'row' or 'both'. Default to the median of the breaks.

vline  
Vector of values within cells where a vertical dotted line should be drawn; only drawn if 'trace' 'column' or 'both'. vline default to the median of the breaks.

linecol  
the color of hline and vline. Defaults to the value of 'tracecol'.

labRow  
character vectors with row labels to use; defaults to rownames(x).

labCol  
character vectors with column labels to use; defaults to colnames(x).

srtRow  
numerical, specifying (in degrees) how row labels should be rotated. See help("par", package="graphics").

srtCol  
numerical, specifying (in degrees) how col labels should be rotated. See help("par", package="graphics").

sideRow  
2 or 4, which side row labels display.

sideCol  
1 or 3, which side row labels display.

margin.for.labRow  
a numerical value gives the margin to plot labRow.

margin.for.labCol  
a numerical value gives the margin to plot labCol.

ColIndividualColors  
(optional) character vector of length ncol(x) containing the color names for a horizontal side bar that may be used to annotate the columns of x.

RowIndividualColors  
(optional) character vector of length nrow(x) containing the color names for a vertical side bar that may be used to annotate the rows of x.

cexRow  
positive numbers, used as 'cex.axis' in for column axis labeling. The default currently only uses number of columns.

cexCol  
positive numbers, used as 'cex.axis' in for the row axis labeling. The default currently only uses number of rows.

labRow.by.group  
logical, whether group unique labels for rows.

labCol.by.group  
logical, whether group unique labels for columns.

key  
logical indicating whether a color-key should be shown.
key.title character, title of the color-key ["Color Key"]
key.xlab character, xlab of the color-key ["Value"]
key.ylab character, ylab of the color-key ["Count"]
key.size numeric value indicating the relative size of the key
map.size numeric value indicating the relative size of the heatmap.
map.ratio the width-to-height ratio of the heatmap.
side.size numeric value indicating the relative size of the sidebars.
cex.key.main a numerical value giving the amount by which main-title of color-key should be magnified relative to the default.
cex.key.xlab a numerical value giving the amount by which xlab of color-key should be magnified relative to the default.
cex.key.ylab a numerical value giving the amount by which ylab of color-key should be magnified relative to the default.
density.info character string indicating whether to superimpose a 'histogram', a 'density' plot, or no plot ('none') on the color-key.
denscol character string giving the color for the density display specified by 'density.info', defaults to the same value as 'tracecol'.
dens.adj Numeric scaling value for tuning the kernel width when a density plot is drawn on the color key. (See the 'adjust' parameter for the 'density' function for details.) Defaults to 0.25.
main an overall title for the plot. See help("title", package="graphics").
sub a subtitle for the plot, describing the distance and/or alignment gap (the "shift").
xlab a title for the x axis. See help("title", package="graphics").
ylab a title for the y axis. See help("title", package="graphics").
cex.main a numerical value giving the amount by which main-title should be magnified relative to the default.
cex.sub a numerical value giving the amount by which sub-title should be magnified relative to the default.
font.main An integer which specifies which font to use for main-title.
font.sub An integer which specifies which font to use for sub-title.
adj.main The value of 'adj' determines the way in which main-title strings are justified.
mgp.main the margin line (in 'mex' units) for the main-title.
mar.main a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the main-title.
mar.sub a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the sub-title.
if.plot logical, whether to plot. Reordered matrix is returned without graphical output if FALSE.
plot.row.partition logical, whether to plot row partition.
plot.col.partition
  logical, whether to plot column partition.

cep.partition
  a numerical value giving the amount by which partition should be magnified relative to the default.

color.partition.box
  color for the partition box.

color.partition.border
  color for the partition border.

plot.row.individuals
  logical, whether to make a plot of row observations.

plot.col.individuals
  logical, whether to make a plot of column observations.

plot.row.clusters
  logical, whether to make a summary plot of row clusters.

plot.col.clusters
  logical, whether to make a summary plot of column clusters.

plot.row.clustering
  logical, whether to make a summary plot of overall row clustering.

plot.col.clustering
  logical, whether to make a summary plot of overall column clustering.

plot.row.individuals.list
  a list of expressions that is used to plot.row.individuals

plot.col.individuals.list
  a list of expressions that is used to plot.col.individuals

plot.row.clusters.list
  a list of expressions that is used to plot.row.clusters

plot.col.clusters.list
  a list of expressions that is used to plot.col.clusters

plot.row.clustering.list
  a list of expressions that is used to plot.row.clustering

plot.col.clustering.list
  a list of expressions that is used to plot.col.clustering

row.data  (optional) data used to plot.row.individuals, plot.row.clusters or plot.row.clustering

col.data  (optional) data used to plot.col.individuals, plot.col.clusters or plot.col.clustering

if.plot.info
  logical, whether to plot text.box.

text.box
  character plotted when if.plot.info is TRUE.

cex.text
  a numerical value giving the amount by which text.box should be magnified relative to the default.

... arguments to be passed to method heatmap.3.

e help("image", package="graphics").
Details

Enhanced heatmap representation with partition and summary statistics (optional). This is an enhanced version of `heatmap.2` function in the Package `{gplots}`. The enhancement includes: 1) Improved performance with optional input of precomputed dist object and hclust object. 2) Highlight of specific cells using rectangles. For instance, the cells of clusters of interests. (Examples should be included in future.) 3) Add-on plots in addition to the heatmap, such as cluster-wise summary plots and overall clustering summary plots, to the right of or under the heatmap.

Value

A reordered matrix according to row or/and col dendrogram(s) and indices that used for reordering.

Examples

```r
## Example1: mtcars
## load library
require("GMD")

## load data
data(mtcars)

## heatmap on raw data
x <- as.matrix(mtcars)

dev.new(width=10,height=8)
heatmap.3(x) # default, with reordering and dendrogram

## Not run:
heatmap.3(x, Rowv=FALSE, Colv=FALSE) # no reordering and no dendrogram
heatmap.3(x, dendrogram="none") # reordering without dendrogram
heatmap.3(x, dendrogram="row") # row dendrogram with row (and col) reordering
heatmap.3(x, dendrogram="row", Colv=FALSE) # row dendrogram with only row reordering
heatmap.3(x, dendrogram="col") # col dendrogram
heatmap.3(x, dendrogram="col", Rowv=FALSE) # col dendrogram with only col reordering
heatmapOut <-
  heatmap.3(x, scale="column") # scaled by column
names(heatmapOut) # view the list that is returned
heatmap.3(x, scale="column", x.center=0) # colors centered around 0
heatmap.3(x, scale="column",trace="column") # trun "trace" on

## End(Not run)

## coloring cars (row observations) by brand
brands <- sapply(rownames(x), function(e) strsplit(e,"\[\[\]\]")[[1]][1])
names(brands) <- c()
brands.index <- as.numeric(as.factor(brands))
RowIndividualColors <- rainbow(max(brands.index))[brands.index]
heatmap.3(x, scale="column", RowIndividualColors=RowIndividualColors)

## coloring attributes (column features) randomly (just for a test :)
```

Add Legends to Plots

This function can be used to add legends to plots. Note that a call to the function locator(1) can be used in place of the x and y arguments.
Arguments

x
the x coordinates to be used to position the legend.

y
the y coordinates to be used to position the legend. x and y can be specified by keyword or in any way which is accepted by `xy.coords`; See ‘Details’.

legend
a character or expression vector of length ≥ 1 to appear in the legend. Other objects will be coerced by `as.graphicsAnnot`.

fill
if specified, this argument will cause boxes filled with the specified colors (or shaded in the specified colors) to appear beside the legend text.

col
the color of points or lines appearing in the legend.

border
the border color for the boxes (used only if fill is specified).

lty
the line types for lines appearing in the legend.

lwd
the line widths for lines appearing in the legend. One of lty and lwd must be specified for line drawing.

pch
the plotting symbols appearing in the legend, either as vector of 1-character strings, or one (multi character) string. Must be specified for symbol drawing.

angle
angle of shading lines.

density
the density of shading lines, if numeric and positive. If NULL or negative or NA color filling is assumed.

bty
the type of box to be drawn around the legend. The allowed values are "o" (the default) and "n".

bg
the background color for the legend box. (Note that this is only used if bty != "n").

box.lwd
the line type for the legend box.

box.lty
the line width for the legend box.

box.col
the color for the legend box.

pt.bg
the background color for the points, corresponding to its argument bg.

cex
character expansion factor relative to current par("cex").

pt.cex
expansion factor(s) for the points.

pt.lwd
line width for the points, defaults to the one for lines, or if that is not set, to par("lwd").
Bin-wise summary of histograms

Details

see `legend` in package:graphics for details; Note: Old versions of graphics::legend do not have ‘border’ option.
Usage

mhist.summary(h, ...)

## S3 method for class 'mhist.summary'
plot(x,bins,plot.ci=TRUE,col=NULL,
     ci.color="orchid1",tcl=-0.25,omi=c(0.5,0.5,1.0,0.25),mar=c(3,3,3,1),
     mgp=c(2,0.5,0),if.plot.new=TRUE, ...)

Arguments

h a "mhist" object as produced by as.mhist
x a mhist.summary object as produced by mhist.summary
bins character vector, the bin labels; if non-specific, bins are numbered/labeled starting with one.
plot.ci logical, indicating whether plot error bars that represent the 0.50 confidence interval (CI)
col color of the histogram
ci.color color of the error bars
tcl the length of tick marks as a fraction of the height of a line of text. See option tcl in help("par", package="graphics").
omi a vector of the form 'c(bottom, left, top, right)' giving the size of the outer margins in inches. See option omi in help("par", package="graphics").
mar a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. See option mar in help("par", package="graphics").
mgp the margin line (in 'mex' units) for the axis title, axis labels and axis line.
if.plot.new logical, whether starting a new device or not.
... arguments to be passed to method plot.mhist.summary.
See help("barplot2", package="gplots").

Details

Bin-wise summary of a mhist object of histograms

Value

mhist.summary returns a mhist.summary object

See Also

mhist plot.mhist plot.gmdp plot.gmdm
plot.gmdm

S3 method for class ‘gmdm’

Description

S3 method for class gmdm

Usage

## S3 method for class ‘gmdm’
plot(x, labels, colors, type = NULL, main, ylab = "Fraction",
     xlab = "Position", label.length.max = 8, label.line.max = 3,
     cex.text = 1, cex.tickmark = 0.75, if.plot.new = TRUE, ...)

Arguments

x
  an object of class gmdm.

labels
  a string vector of the same length as x$data, giving the names of the numeric vectors in x$data.

colors
  the colors of the discrete distributions; the default is "Dark2" colors in ColorBrewer palettes if not specified.

type
  type of plot, as in help("plot", package="graphics").

main
  an overall title for the plot. See help("title", package="graphics"); the default title is used if not specified.

ylab
  a title for the y axis. See help("title", package="graphics").

xlab
  a title for the x axis. See help("title", package="graphics").

label.length.max
  numeric, giving the maximum string width allowed in diagonal labels.

label.line.max
  numeric, giving the maximum number of lines allowed in diagonal labels.

cex.text
  a numerical value giving the amount by which plot text should be magnified relative to the default.

cex.tickmark
  a numerical value giving the amount by which tickmarks should be magnified relative to the default.

if.plot.new
  logical, indicating whether to start a new plot device.

... arguments to be passed to methods, see gmdp.

Details

S3 method for class gmdm

References

See help(GMD)
plot.gmdp

See Also
gmdm, gmdp

Examples

```r
## Example1: CAGE
require("GMD") # load library
data(cage) # load data

## construct a distance matrix and visualize it
short.labels <- gsub("(\.+)([^\."\]'\"]+)',"\1\"",names(cage)) # get short labels
x <- gmdm(cage[1:6],labels=short.labels[1:6])
plot(x)

## Not run:
## Example2: ChIP-seq

require("GMD") # load library

data(chipseq_mES) # load data
data(chipseq_hCD4T) # load data

## pairwise distance and alignment based on GMD metric
plot(gmdm(chipseq_mES,sliding=FALSE))

## clustering on spatial distributions of histone modifications
x <- gmdm(chipseq_hCD4T,sliding=FALSE,resolution=10)
heatmap.3(x,revC=TRUE)

## End(Not run)
```

---

plot.gmdp

Plot function for class gmdp

Description
Plot Function for Class gmdp

Usage

```r
## S3 method for class 'gmdp'
plot(x, labels = NULL, colors = NULL, main,
     ylab = "Fraction", xlab = "Position", xlim = NULL, type = NULL,
     if.text.gmd = TRUE, if.text.gap = TRUE, ...)
```
Arguments

x an object of class gmdp.
labels a string vector of the same length of x$labels, giving the names of the numeric vectors in x.
colors the colors of the discrete distributions. See help("plot.mhist", package="GMD").
main an overall title for the plot.
ylab a title for the y axis. See help("plot.mhist", package="GMD").
xlab a title for the x axis. See help("plot.mhist", package="GMD").
xlim numeric vectors of length 2, giving the x coordinates ranges.
type type of plot, as in help("plot", package="graphics").
if.text.gmd logical, indicating whether GM-Distance is reported in the subtitle.
if.text.gap logical, indicating whether gap is reported in the subtitle.
... arguments to be passed to methods. See help("plot.mhist", package="GMD").

Details

Plot Function for Class gmdp

References

See help(GMD)

See Also

gmdp

Examples

require("GMD") # load library
data(cage) # load data

## measure pairwise distance
x <- gmdp(cage["Pfkfb3 (T02R00AEC2D)",cage["Csf1 (T03R0572174D)"]])
print(x) # print a brief version by default
print(x, mode="full") # print a full version by default

## show alignment
plot(x,labels=c("Pfkfb3","Csf1"),beside=FALSE)

## show another alignment
plot(gmdp(cage["Hig1 (T09R0743763C)",cage["Cd72 (T04R028B8BC9)"]],
labels=c("Hig1 (T09R0743763C)","Cd72 (T04R028B8BC9)"),
beside=FALSE)
plot.mhist

S3 method for class 'mhist'.

Description

S3 method for class mhist

Usage

## S3 method for class 'mhist'
plot(x, beside = TRUE, labels = NULL, colors = NULL,
     main = NULL, sub = NULL, ylab = NULL, xlab = NULL, xticks = NULL,
     xlabels = NULL, vlinePos = NULL, x.las = 1, xticks.type = c("pretty",
     "original"), xlim = NULL, ylim = NULL, type = NULL, font.type = 1,
     font.family = c("sans", "serif", "mono"), cex.main = 1.75,
     cex.sub = cex.main * 0.9, cex.lab = 1.25, cex.tickmark = 0.75,
     cex.legend = 1.5, tcl = -0.25, omi = c(0.5, 0.5, 1, 0.25), mar = c(4,
     1, 0, 1), mgp = c(0, 0.5, 0), bin.unit = 0.8, legend.lab = labels,
     legend.pos = c("topright", "top", "topleft"), ...)

Arguments

x a numeric matrix or data frame, representing distributions by rows (bins by columns); or a list of numeric vectors as distributions.

beside logical, whether plot histograms side-by-side.

labels a string vector of labels for the histograms in x; should have the same number as of the histograms.

colors the colors for the histograms; by default they are set to colors generated from palette Dark2. Colors will be recycled if the size is smaller than the number of the histograms.

main an overall title for the plot. See help("title", package="graphics").

sub a subtitle for the plot, describing the distance and/or alignment gap (the "shift").

ylab a title for the y axis. See help("title", package="graphics").

xlab a title for the x axis. See help("title", package="graphics").

xticks a string vector indicating the tickmark labels at x-axis. Default: NULL.

xlabels character, labels at x-axis.

vlinePos numeric, position for vertical lines.

x.las numeric in 0,1,2,3; the style of axis labels. See option las in help("par", package="graphics").

xticks.type string in "pretty","original", whether plot the xticks in a pretty way or as is.

xlim range of x values, as in help("plot", package="graphics").

ylim range of y values, as in help("plot", package="graphics").
type type of plot, as in help("plot", package="graphics").
font.type  the name of a font type for drawing text. See font in par. DEFAULT: font.type = 1, corresponding to plain text.
font.family the name of a font family for drawing text. See family in par; DEFAULT: font.family = "sans", corresponding to serif typeface.
cex.main a numerical value giving the amount by which main-title should be magnified relative to the default.
cex.sub a numerical value giving the amount by which sub-title should be magnified relative to the default.
cex.lab a numerical value giving the amount by which xlab and ylab should be magnified relative to the default.
cex.tickmark a numerical value giving the amount by which tickmarks should be magnified relative to the default.
cex.legend a numerical value giving the amount by which legends should be magnified relative to the default.
tcl the length of tick marks as a fraction of the height of a line of text. See option tcl in help("par", package="graphics").
omi a vector of the form c(bottom, left, top, right) giving the size of the outer margins in inches. See option omi in help("par", package="graphics").
mar a numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. See option mar in help("par", package="graphics").
mgp the margin line (in 'mex' units) for the axis title, axis labels and axis line. See option mgp in help("par", package="graphics").
bin.unit numeric, indicating the width of a group of bar(s) in unit of x axis.
legend.lab legend labels, a string vector of the same length of distributions in x, using labels by default. No legend is displayed when it is NA.
legend.pos string, a keyword to be used to position the legend. See help("legend", package="graphics").
... arguments to be passed to method plot.mhist, such as graphical parameters (see par).

Details
Given a list, matrix or data.frame of histograms, plot multiple histograms side-by-side or as subplots.

References
See help(GMD)

See Also
mhist mhist.summary plot.mhist summary plot.gmdp plot.gmdm
Examples

```r
# load library
require("GMD")

# create two normally-distributed samples
# with unequal means and unequal variances
set.seed(2012)
v1 <- rnorm(1000, mean=-5, sd=10)
v2 <- rnorm(1000, mean=10, sd=5)

# create common bins
n <- 20 # desired number of bins
breaks <- gbrc(ce(v1,v2),n) # bin boundaries
x <-
  list(ghist(v1,breaks=breaks,digits=0),
       ghist(v2,breaks=breaks,digits=0))
whist.obj <- as.whist(x)

# plot histograms side-by-side
plot(whist.obj, mar=c(1.5,1,1,0),
     main="Histograms of simulated normal distributions")

# plot histograms as subplots,
# with corresponding bins aligned
plot(whist.obj, beside=FALSE, mar=c(1.5,1,1,0),
     main="Histograms of simulated normal distributions")
```

---

**ts2df**

*Convert time series to data frame*

**Description**

A copy of wq:::ts2df; see ts2df in package:wq for details

**Usage**

```r
ts2df(x, mon1 = 1, addYr = FALSE, omit = FALSE)
```

**Arguments**

- `x` monthly time series vector
- `mon1` starting month number, i.e., first column of the data frame
- `addYr` rows are normally labelled with the year of the starting month, but addYr = TRUE will add 1 to this year number
- `omit` if TRUE, then rows with any NA will be removed.
Details

see ts2df in package:wq for details. Note: wq_0.3-4 asks for R (>= 2.12.0); but GMD supports R (>= 2.9.0).
Index

*Topic **classes**
  gmdm, 14
  gmdp, 16
*Topic **datasets**
  cage, 4
  chipseq, 5
*Topic **plot**
  plot.gmdm, 28
  plot.gmdp, 29
  plot.mhist, 31
*Topic **methods**
  plot.gmdm, 28
  plot.gmdp, 29
  plot.mhist, 31
*Topic **package**
  GMD-package, 2
  as.ghist (ghist), 13
  as.graphicsAnnot, 25, 26
  as.mhist (ghist), 13
  bedgraph.to.depth, 4
  cage, 3, 4, 4, 5, 6
  cagel (cage), 4
  chipseq, 3, 5, 5, 6
  chipseqHC4T (chipseq), 5
  chipseq.mES (chipseq), 5
  css, 3, 6, 9
  css.hclust, 9
  elbow, 3, 7, 7
  equalize.list, 10
  expression, 25
  gb breaks (ghist), 13
  gdist, 3, 10
  get.sep, 12
  ghist, 3, 13
  GMD (GMD-package), 2
  GMD-package, 2
  gmdm, 3, 5, 6, 14, 17, 29
  gmdm2dist (gmdm), 14
  gmdm_dist (gmdm), 14
  gmdp, 3, 5, 6, 15, 16, 29, 30
  heatmap, 3, 3, 17
  is.dist (gdist), 10
  is.ghist (ghist), 13
  is.mhist (ghist), 13
  legend, 24
  locator, 24
  mhist, 27, 32
  mhist (ghist), 13
  mhist.summary, 14, 26, 32
  mhist2matrix (ghist), 13
  plot.elbow (elbow), 7
  plot.gmdm, 15, 27, 28, 32
  plot.gmdp, 17, 27, 29, 32
  plot.mhist, 14, 27, 31
  plot.mhist.summary, 14, 32
  plot.mhist.summary (mhist.summary), 26
  plotmath, 26
  points, 25
  print.gmdm (gmdm), 14
  print.gmdp, 17
  print.gmdp (gmdp), 16
  strwidth, 26
  summary.gmdm, 17
  summary.gmdp (gmdp), 16
  ts2df, 33
  xy.coords, 25