centerScale

center and scale a vector to zero median, unit mad

Description

center and scale a vector to zero median, unit mad

Usage

centerScale(x)

Arguments

x       a numeric vector

Value

a shifted and scaled version of x with zero median, unit mad

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

summary(centerScale(runif(200)))
edd

new expression density diagnostics interface

Description

this will replace edd.unsupervised; has more sensible parameters

Usage

edd(eset, distList=eddDistList, tx=c(sort,flatQQNormY)[[1]],
    refDist=c("multiSim", "theoretical")[[1]],
    method=c("knn", "nnet", "test")[[1], nRowPerCand=100, ...)

Arguments

- eset: eset – instance of Biobase ExpressionSet class
- distList: distList – list comprised of eddDist objects
- tx: tx – transformation of data and reference prior to classification
- refDist: refDist – type of reference distribution system to use
- method: method – type of classifier to use. knn is k-nearest neighbors, nnet is neural net, test is max p-value from ks.test
- nRowPerCand: nRowPerCand – number of realizations for a multiSim reference system
- ...: ...– parameters to classifiers

Details

Classifies genes according to distributional shape, by comparing observed expression distributions to a collection of references, which may be simulated or evaluated theoretically.

The distList argument is important. It enumerates the catalog of distributions for classification of gene expression vectors by distributional shape. See the HOWTO-edd vignette for information on how this list is constructed and how it can be extended.

The tx argument specifies how the data are processed for comparison to the reference catalog. This is a function on a vector returning a vector, but the input and the output need not have the same length. The default value of tx is sort, which entails that the order statistics are treated as multivariate data for classification.

The refDist argument selects the type of reference catalog. Options are ‘multiSim’, for which the reference consists of nRowPerCand realizations of each catalog entry, and ‘theoretical’, for which the reference consists of one vector of quantiles for each catalog entry.

The method argument selects the type of classifier. It would be desirable to allow this to be a function, but there is insufficient structure on classifier argument and return value structure to permit this at present; see the e1071 package for some work on handling various classifiers programmatically (e.g., tune).

Value

a character vector or factor depending on the classifier
eddDist-class

Author(s)
Vince Carey <stvjc@channing.harvard.edu>

See Also
ExpressionSet

Examples

```r
require(Biobase)
data(sample.ExpressionSet)
# should filter to genes with reasonable variation
table(edd(sample.ExpressionSet, meth="nnet", size=10, decay=.2))
library(golubEsets)
data(Golub_Merge)
madvec <- apply(exprs(Golub_Merge),1,mad)
minvec <- apply(exprs(Golub_Merge),1,min)
keep <- (madvec > median(madvec)) & (minvec > 300)
gmfilt <- Golub_Merge[keep==TRUE,]
ALL <- gmfilt$ALL.AML=="ALL"
gall <- gmfilt[,ALL==TRUE]
gaml <- gmfilt[,ALL==FALSE]
alldists <- edd(gall, meth="nnet", size=10, decay=.2)
amldists <- edd(gaml, meth="nnet", size=10, decay=.2)
table(alldists,amldists)
amldists2 <- edd(gaml, meth="nnet", refDist="theoretical", size=10, decay=.2)
table(amldists,amldists2)
```

eddDist-class

Class "eddDist"

Description

Objects from this class can be used to simulate or tabulate reference distributions for edd

Objects from the Class

Objects can be created by calls of the form `new("eddDist", ...)`. These objects include a simple stub (like "norm", which can be modified to obtain the name of a generator (prepend "r"), cdf (prepend "p"), etc.) in R.

Slots

- stub: Object of class "character" stub of a distribution system in R, to which "r" is prepended to get a generator, "p" to get a cdf, "q" to get a quantile function...
- parms: Object of class "numeric" named vector of parameters for a member of the family
- median: Object of class "numeric" median of the distribution (sometimes has to be computed by simulation)
- mad: Object of class "numeric" MAD of the distribution (sometimes has to be computed by simulation)
- tag: Object of class "character" an informative character tag
- latexTag: Object of class "character" an informative character tag in latex format, which can use subscripts, greek letters, etc.
Methods

CDFname signature(x = "eddDist"): prepend "p" to stub(x)

genName signature(x = "eddDist"): prepend "r" to stub(x)

Mad signature(x = "eddDist"): accessor

med signature(x = "eddDist"): accessor

parms signature(x = "eddDist"): accessor

qfName signature(x = "eddDist"): prepend "q" to stub(x)

qfun signature(e = "eddDist"): construct a quantile function from the object

stub signature(x = "eddDist"): accessor

tag signature(x = "eddDist"): accessor

latexTag signature(x = "eddDist"): accessor

testVec signature(x = "numeric", edd = "eddDist", is.centered = "logical"): apply ks.test to a given vector x against the dist specified by the eddDist components

Examples

library(edd)
eddDistList[[1]]
qfun(eddDistList[[1]])

eddObsolete

Expression Density Diagnostics

Description

Classify cohort distributions of gene expression values.

Usage

eddObsolete(eiset,
  ref=c("multiCand", "uniCand", "test", "nnet")[1],
  k=10, l=6, nnsize=6, nniter=200)

Arguments

eset instance of Biobase class ExpressionSet.

ref one of `multiCand`, `uniCand`, `test` or `nnet`. see details.

k k setting for knn – number of nearest neighbors to poll.

l l setting for knn – minimum number of concordant assents.

nnsize size parameter for nnet.

nniter iter setting for nnet.
**Details**

Four options are available for classifying expression densities. Data on each gene are shifted and scaled to have median zero and mad 1. They are then compared to shapes of reference distributions (standard Gaussian, chisq(1), lognorm(0,1), t(3), .75N0,1+.25N4,1, .25N0,1+.75N4,1, Beta(2,8), Beta(8,2), U(0,1)) after each of these has been transformed to have median 0 and mad 1. Classification proceeds by one of four methods, selected by setting of the ‘ref’ argument. Suppose there are S samples in the ExpressionSet.

- **multiCand** – 100 samples of size S are drawn from each reference distribution and then scaled to med 0, mad 1. The knn(k,l) procedure is used to classify the genes based on proximity to representatives in this set.

- **uniCand** – one representative of size S is created from each reference distribution, using the theoretical quantiles. knn(1,0) is used to classify genes based on proximity to these representatives.

- **test** – classification of each gene is based on maximum p-value of Kolmogorov-Smirnov tests vs each reference distribution. If the p-value never exceeds .1, ‘doubt’ is declared.

- **nnet** – 100 samples of size S are drawn from each reference distribution and then scaled to med 0, mad 1. A neural net is fit to this dataset and the associated labels. The net is then applied to the scaled gene expression data and the predictions are used for classification.

**Value**

the vector of classifications, with NAs for nonclassifiable genes

**Author(s)**

VJ Carey

**Examples**

```r
require(Biobase)
data(sample.ExpressionSet)
print(summary(eddObsolete(sample.ExpressionSet,k=10,l=2)))

# 6 x 20 x 50 test problem
set.seed(1234)
test <- matrix(NA,nr=120,nc=50)
test[1:20,] <- rnorm(1000)
test[21:40,] <- rt(1000,3)
test[41:60,] <- rexp(1000,4)
test[61:80,] <- rmixnorm(1000,.750,0,1,4,1)
test[81:100,] <- runif(1000)
test[101:120,] <- rlnorm(1000)
labs <- c(rep("n01",20),rep("t3",20),
        rep("exp",20),rep("mix1",20),rep("u01",20),rep("ln01",20))
phenoData <- new("AnnotatedDataFrame")
pData(phenoData) <- data.frame(1:50)
varLabels(phenoData) <- list("Col1")
TT <- new("ExpressionSet", exprs=test, phenoData = phenoData)

multrun <- eddObsolete(TT, k=10, l=2)
print(table(given=labs, multiCand=multrun))
netrun <- eddObsolete(TT, ref="nnet")
print(table(given=labs, netout=netrun))
```
newrun <- edd(TT, meth="nnet", size=10, decay=.2)
print(table(given=labs, newout=newrun))
newrun <- edd(TT, meth="test")
print(table(given=labs, newout=newrun))

flatQQNorm

QQ difference plot

Description

standard normal transforms to horizontal line at 0

Usage

flatQQNorm(y)

Arguments

y numeric vector

Value

list with elements x and y describing the trace of the qq difference plot

Examples

set.seed(1234)
plot(flatQQNorm(rnorm(40)), ylim=c(-3,3), ylab="QQNorm - line of identity ")

latEDtable

use latex tags for dimnames of table created from edd classification

Description

use latex tags for dimnames of table created from edd classification

Usage

latEDtable(x, baselist=eddDistList, reorder=NULL)

Arguments

x x – a table (possibly two dimensional) of results of applying edd
baselist baselist – a list of eddDist objects
reorder reorder – a numeric vector describing how to re order the table rows/columns

Details

for use with xtable rendering, table will give results with margin names in lexicographic order; reorder can be used to get a different order.
Value

a matrix with dimnames computed from the latexTag slots of the corresponding eddDist objects

Author(s)

Vince Carey <stvjc@channing.harvard.edu>

Examples

```r
require(Biobase)
data(sample.ExpressionSet)
# should filter to genes with reasonable variation
rawTab <- table(edd(sample.ExpressionSet, meth="nnet", size=10, decay=.2))
latEDtable(rawTab)
library(xtable)
xtable(latEDtable(rawTab))

# realTags <- sapply(eddDistList, tag)
reo <- match(realTags, names(rawTab))
xtable(latEDtable(rawTab, reorder=reo))
```

makeCandmat.raw

create and store reference distributions for edd

Description

create and store reference distributions for edd

Usage

```r
makeCandmat.raw (nPerRow=20, nRowPerCand=20, dists=eddDistList, centerScale=TRUE)
```

Arguments

- `nPerRow` size of each reference sample
- `nRowPerCand` number of samples per candidate
- `dists` list of eddDist objects specifying reference distributions
- `centerScale` logical indicating that simulated data should be centered and scaled

Value

matrix with rows realized from reference distributions

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

```r
makeCandmat.raw(5,2,eddDistList[1:3])
```
maxKSp  

classify densities by maximum KS p-value

Description

compares a sample to reference distributions, identifying the closest reference by maximum KS p-value.

Usage

maxKSp( x, is.centered=TRUE, dists=eddDistList, thresh=.1 )

Arguments

x matrix of samples, distributions to be classified by row
is.centered have data been brought to median 0, mad 1
dists list of instances of class eddDist
thresh p-value above which some test must lie, or else 'outlier' is declared

Value

a vector of classifications

Examples

X <- rbind( rnorm(30), runif(30), rchisq(30,1))
tX <- t(apply(X,1,centerScale))
apply(tX,1,maxKSp)

plotED  

visualize an eddDist object

Description

plots an eddDist model; can superimpose density sketch from transformed data.

Usage

plotED(x, y, data=NULL, is.centered=FALSE, ...)

Arguments

x eddDist object
y not used
data optional vector of data to be superimposed in the form of a kernel density estimate
is.centered is.centered: logical indicating that the data vector has zero median and unit mad
... ...: options passed to curve
Examples

```r
# show the first 8 supplied reference dists
if (interactive()){
  omf <- par()$mfrow
  oas <- par()$ask
  on.exit(par(mfrow=omf,ask=oas))
  par(mfrow=c(4,2))
  par(ask=TRUE)
}
set.seed(1234)
for (i in 1:8) plotED(eddDistList[[i]])
# illustrate the superposition
if (interactive()) par(mfrow=c(1,1))
x <- rnorm(30,3,4)
plotED(N01,data=x) # relocates/scales x
y <- 12*rbeta(30,2,8)+4
plotED(B28,data=y)
```
Index

*Topic classes
   eddDist-class, 3
*Topic models
   centerScale, 1
eedd, 2
eddObsolete, 4
flatQQNorm, 6
latexTag, 1
makeCandmat.raw, 7
maxKSp, 8
plotED, 8
B28 (eddDist-class), 3
B22 (eddDist-class), 3
CDFname (eddDist-class), 3
CDFname, eddDist-method (eddDist-class), 3
centerScale, 1
CS1 (eddDist-class), 3
ctr (centerScale), 1
dmix1n (makeCandmat.raw), 7
dmix2n (makeCandmat.raw), 7
dmixnorm (eddDist-class), 3
edd, 2
eddDist-class, 3
eddDistList (eddDist-class), 3
eddObsolete, 4
ExpressionSet, 2–4
flatQQNorm, 6
flatQQNormY (flatQQNorm), 6
fq.matrows (eddObsolete), 4
genName (eddDist-class), 3
genName, eddDist-method (eddDist-class), 3
latexTag, 6
latexTag, eddDist-method (eddDist-class), 3
LN01 (eddDist-class), 3
Mad (eddDist-class), 3
Mad, eddDist-method (eddDist-class), 3
makeCandmat.raw, 7
makeCandmat.theor (eddObsolete), 4
maxKSp, 8
med (eddDist-class), 3
med, eddDist-method (eddDist-class), 3
MIXN1 (eddDist-class), 3
MIXN2 (eddDist-class), 3
mkt (eddObsolete), 4
N01 (eddDist-class), 3
parms (eddDist-class), 3
parms, eddDist-method (eddDist-class), 3
plotED, 8
plotlim (eddDist-class), 3
plotlim, eddDist-method (eddDist-class), 3
pmix1n (makeCandmat.raw), 7
pmix2n (makeCandmat.raw), 7
pmixnorm (eddDist-class), 3
qfName (eddDist-class), 3
qfName, eddDist-method (eddDist-class), 3
qfun (eddDist-class), 3
qfun, eddDist-method (eddDist-class), 3
qmix1n (makeCandmat.raw), 7
qmix2n (makeCandmat.raw), 7
qmixnorm (eddDist-class), 3
rmixnorm (makeCandmat.raw), 7
rmixnorm.alt (eddDist-class), 3
s.rmix1norm (eddObsolete), 4
s.rmix2norm (eddObsolete), 4
stub (eddDist-class), 3
stub, eddDist-method (eddDist-class), 3
INDEX

T3(eddDist-class), 3
tag(eddDist-class), 3
tag, eddDist-method
  (eddDist-class), 3
testcl(eddObsolete), 4
testVec(eddDist-class), 3
testVec, eddDist-method
  (eddDist-class), 3
testVec, numeric, eddDist, logical-method
  (eddDist-class), 3
U01(eddDist-class), 3