Expression density diagnostics in Bioconductor: Details

Vince Carey `stvjc@channing.harvard.edu`

October 31, 2011

Contents

1 Introduction

The `edd` function of *edd* 1.2 assists in exploratory data analysis of microarray expression data. A HOWTO document is supplied with the *edd* package and provides a brief treatment of use of the *edd* function. This document covers the computational and statistical details.

The objective of `edd` is to deliver to the user a classification of shapes of gene-specific distributions of expression over a collection of arrays. The primary input to `edd` is an `ExpressionSet`. The classification is carried out as follows.

- Gene-specific expression vectors (rows of the ExpressionSet's exprs content) are transformed to zero median and unit MAD (median absolute deviation).

- A reference catalog of transformed distributions is constructed, defining the set of distributional shapes of interest.
- Each transformed expression vector is associated with an element of the reference catalog. (or with 'doubt' or 'outlier').

Options to *edd* that specify the methods of constructing catalogs or associating data with catalog elements will be spelled out in this document.

Assessment of distributional shape is a common activity in exploratory data analysis, carried out with boxplots, histograms, density estimation and other tools. The novel problem in the microarray setting is to deal efficiently with thousands of distributions. If a manageable number of 'classes' of distributions can be identified, discovery and inference should be enhanced.

In this document, we review some basic activities related to graphically depicting distinct distributional shapes. Section 2 provides very elementary tools for plotting cumulative distribution functions (CDFs), and for plots that facilitate contrasts of distributions (QQ and QQ difference plots). Section 3 discusses creation of reference catalogs of distributions with distinct shapes. Section 4 describes application of pattern recognition algorithms to the problem of distribution shape classification.

Note that this document spends considerable time on graphical considerations that have become secondary in dealing with expression density diagnostics. HOWTO-edd gives a more concise treatment of how to work directly with the software.

2 Graphics for assessment of distributional shape

A basic tool in our approach to classifying distributions is a method for depicting departure from Gaussianity, or, more generally, depicting features of distributions so that they may be qualitatively distinguished using simple graphics.

To begin, we present some cdf plots for a few familiar distributions. Each distribution is transformed to have mean zero and unit median absolute deviation (MAD). This allows us to focus on graphics that expose distributional 'shape'. Figure 1 presents discretized CDFs for the transformed distributions.

Figure 2 provides QQnormal plots of the same distributions. The motivation is that a 45 degree line is obtained for N(0,1), departures from 45 degree line are evidence of nonGaussianity.

The QQ difference plots (Figures 3 and 4) express departures from normality more effectively – a horizontal line is obtained for the standard Gaussian distribution, and departures from the horizontal line are easily detected and have systematic interpretation:

Formally, these 'flat QQ-normal' plots are defined as

$$[x, y] = [\Phi^{-1}(p_j), y_j - \Phi^{-1}(p_j)],$$

with $p_j \approx \text{card}\{y_k < y_j\}/n$. If y_j are approximately Gaussian, this graph is close to a horizontal line at ordinate zero.

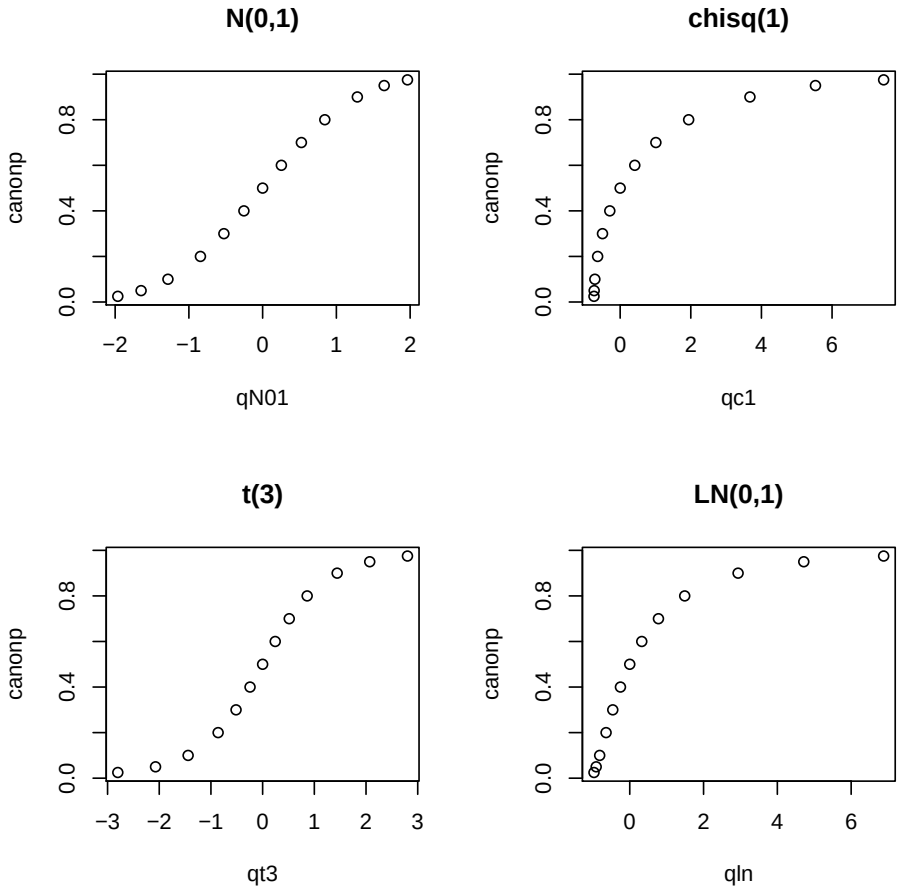


Figure 1: CDF plots for 4 distributions transformed to median 0 MAD 1.

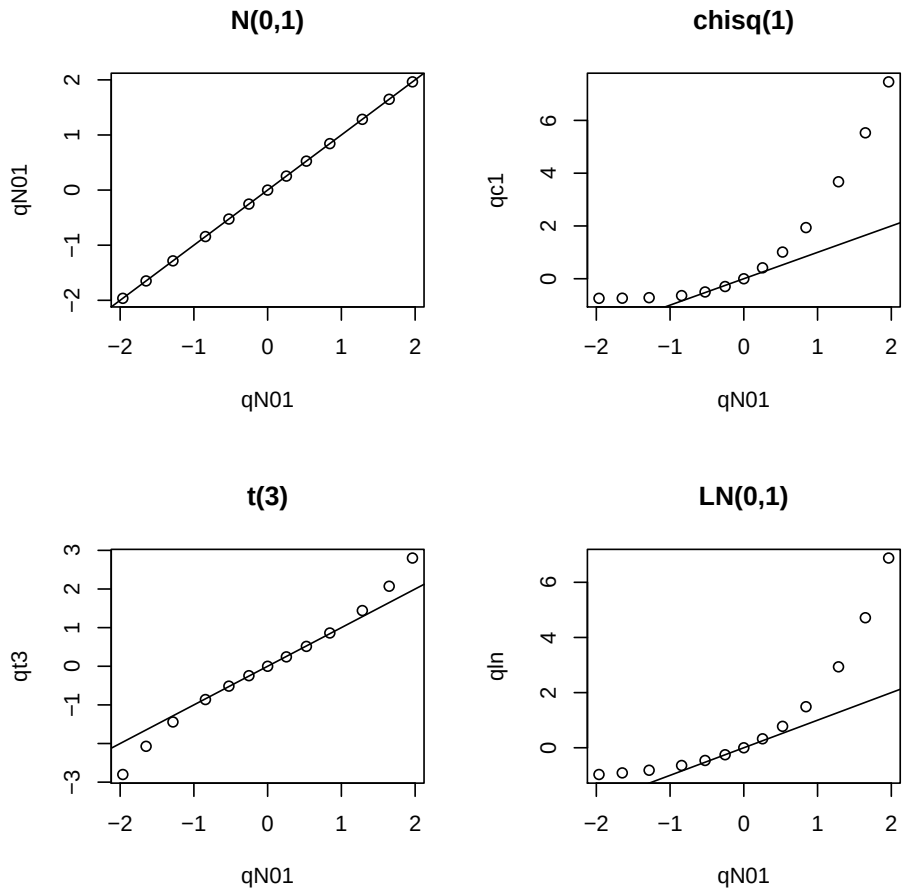


Figure 2: QQ-normal plots for the transformed distributions.

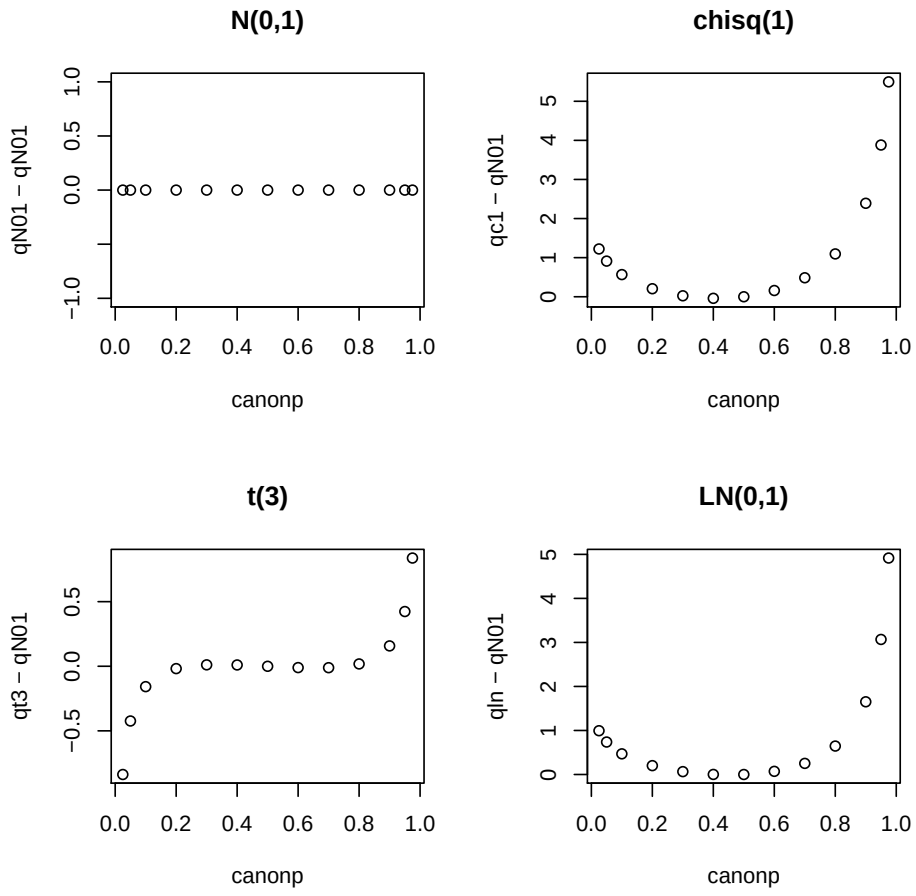


Figure 3: QQ difference plots.

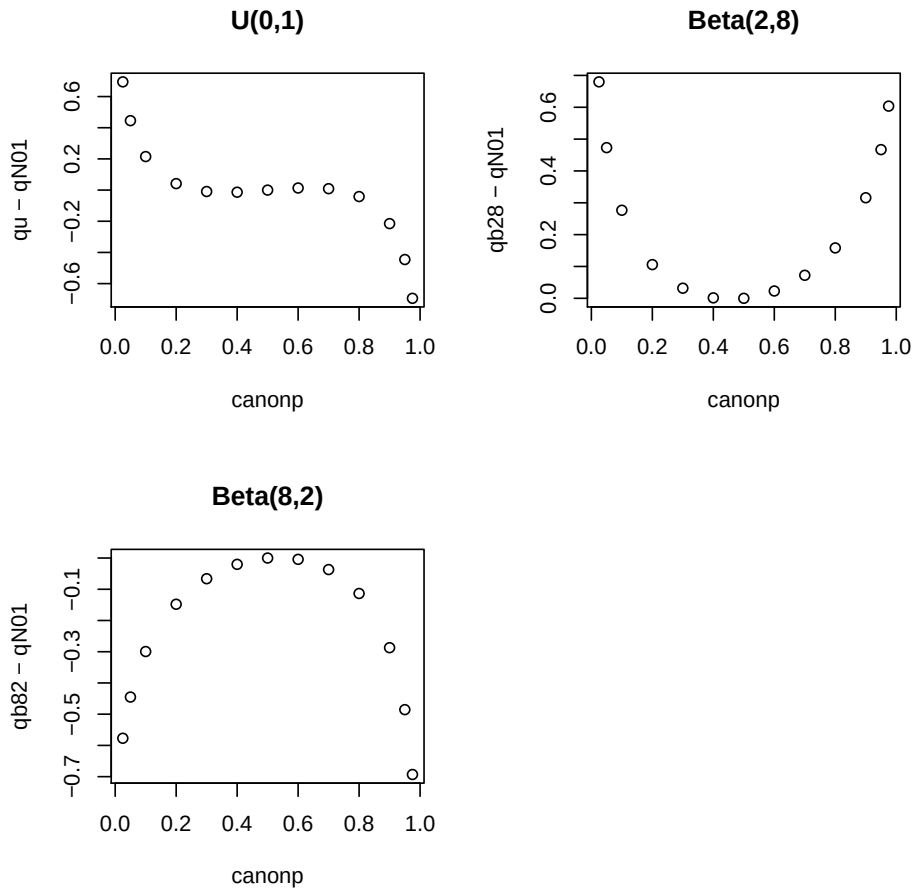


Figure 4: QQ difference plots, continued.

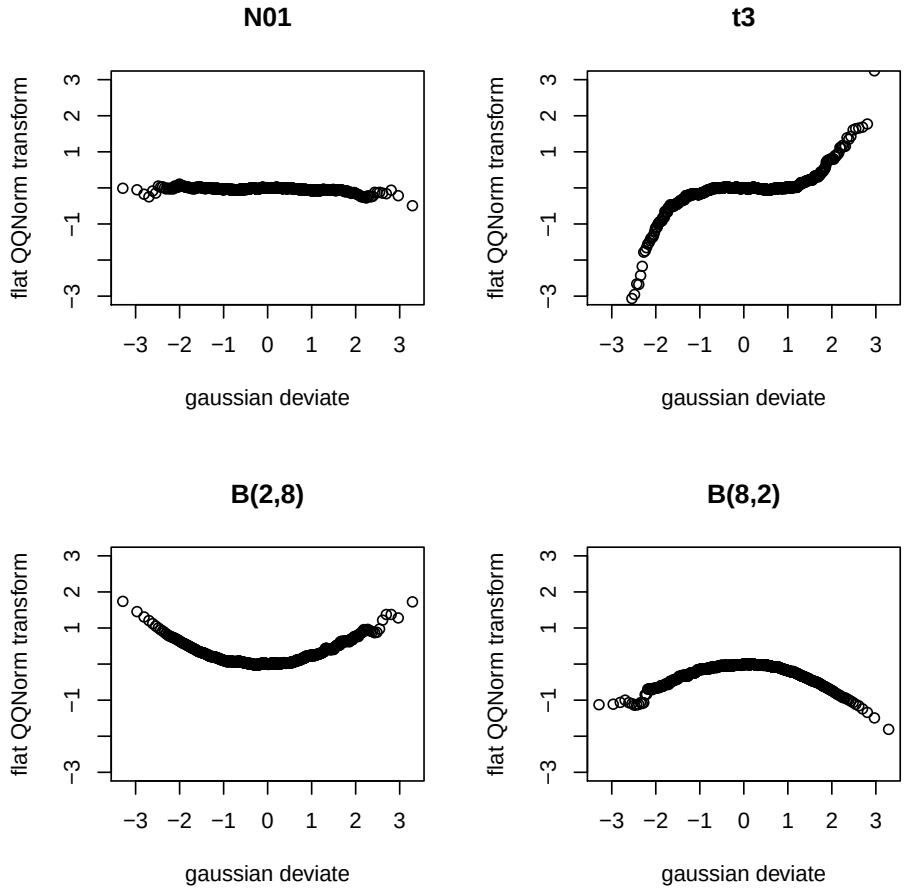


Figure 5: Flat QQnormal plots for 1000 centered and rescaled realizations from each of four parametric distributions.

To conclude this section, we apply the flat QQ normal plotting method to several simulated datasets to illustrate the efficacy of the transformation in detection of departures from Gaussian distributional shape. See Figure 5.

3 Catalogs of distributional shapes

3.1 Underlying data structure

The `eddDist` class has been defined to encapsulate information on distributions that can be used in *edd*.

```
> print(getSlots("eddDist"))

       stub       parms      median         mad         tag     plotlim
"character"   "numeric"   "numeric"   "numeric" "character"   "numeric"
   latexTag
"character"
```

The stub is a string that can be prepended with ”p”, ”d”, ”r”, ”q” to obtain names of R functions that compute cdf, density, samples, or quantiles from distributions. The parms vector specifies the parameters of this distribution, it should be named numeric. The median and mad are sometimes not computable analytically and are obtained by simulation and stored here for reference. The tag slot holds a convenient string name that should clearly indicate the distribution at hand, and latexTag uses mathematical notation if necessary for use with latex rendering. The plotlim is a numeric 2-vector prescribing the x limits for a plot of the density of the distribution.

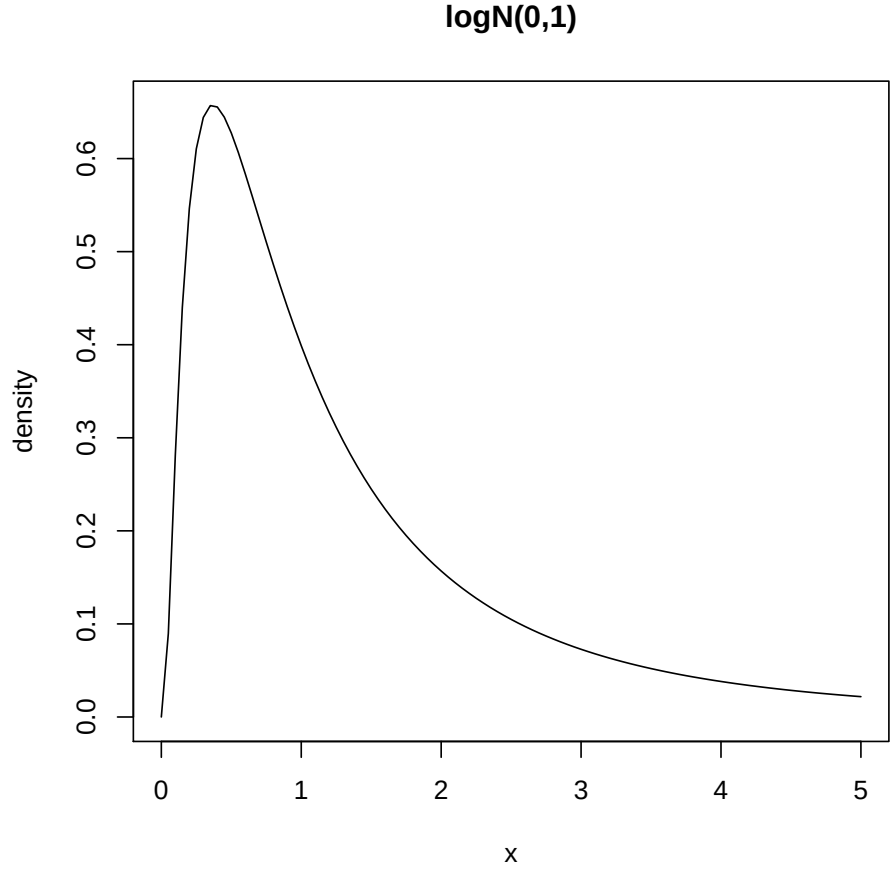
A list of these objects is supplied with the library:

```
> print(names(eddDistList))

 [1] "N01"   "T3"    "LN01"  "CS1"   "B82"   "U01"   "B28"   "MIXN1" "MIXN2"
[10] "MIXN3"

> print(eddDistList[1:2])

$N01
An object of class "eddDist"
Slot "stub":
[1] "norm"

Slot "parms":
mean   sd
   0    1

Slot "median":
[1] 0

Slot "mad":
```

```
[1] 1

Slot "tag":
[1] "N(0,1)"

Slot "plotlim":
[1] -3  3

Slot "latexTag":
[1] "$\\Phi$"


$T3
An object of class "eddDist"
Slot "stub":
[1] "t"

Slot "parms":
df
 3

Slot "median":
[1] 0

Slot "mad":
[1] 1.15

Slot "tag":
[1] "t(3)"

Slot "plotlim":
[1] -4  4

Slot "latexTag":
[1] "$t_3$"
```

Visualization of a distribution can be accomplished with the `plotED` function.

```
> plotED(eddDistList[[3]])
```



3.2 A catalog defined by parametric cdfs

The eddDistList is a catalog defined by parametric cdfs, because the R function given by `paste("p",stub(x),sep="")` for eddDist `x` is a computable cdf. It may thus be employed in KS testing of goodness of fit.

3.3 A catalog defined by transformed quantiles

The function `makeCandmat.theor` obtains quantiles from each distribution defined by eddDist objects in the eddDistList parameter, transformed to have median 0 and mad 1. The number of quantiles to be computed is given by the first argument. In application this will usually match the dimension of the expression vectors.

```
> print(makeCandmat.theor( 5, eddDistList[1:3] ))

           [,1]       [,2] [,3]      [,4]     [,5]
N01  -1.1797611 -0.4972006    0 0.4972006 1.179761
```

```
T3   -1.2776114 -0.4805167    0 0.4805167 1.277611
LN01 -0.7870998 -0.4451921    0 0.7319457 2.560906

>
```

3.4 A catalog defined by multiple transformed realizations

The function `makeCandmat.raw` obtains samples from each distribution defined by eddDist objects in the eddDistList parameter. The generator is constructed to sample from the distribution after transformation to have median 0 and mad 1. The sample size is determined by the first argument, and the number of representative samples to be obtained for each eddDistList element is given in the second argument.

```
> print(makeCandmat.raw( 5, 2, eddDistList[1:3] ))

                [,1]       [,2]       [,3]       [,4]        [,5]
N(0,1)     0.19945461  1.7412724 -0.7962975 -0.3032587  0.90100891
N(0,1)     0.05644715  0.6539273  0.7558425 -0.3350831  0.79649685
t(3)      -0.28292624 -1.8323103 -1.3191323 -0.1834414  0.90203666
t(3)      -1.81856812  0.3646908  0.6204789 -1.0269202  0.04631424
logN(0,1)  0.67045624  1.7696872 -0.5694661  0.4258884 -0.14052508
logN(0,1)  4.54924335 -0.6633869 -1.0545520  2.2462033 -0.07089315

>
```

4 edd: associating expression vectors with catalog elements

The `edd` function is used to define catalog construction and to specify the method by which expression vectors (rows of `exprs(eset)`) are classified.

4.1 Test-based association

An `eddDist` object contains sufficient information to allow the computation of the `ks.test` function for general goodness of fit testing. The `maxKSp` function computes the maximum p-value of these tests over all the elements of the eddDistList.

To compute a single p-value for the goodness of fit of a given expression vector to the distributional shape specified in an eddDist object, the `testVec` method is used.

```
setMethod("testVec", c("numeric","eddDist","logical"),
  function(x,eddd,is.centered) {
    if (is.centered) x <- Mad(eddd)*x + med(eddd)
```

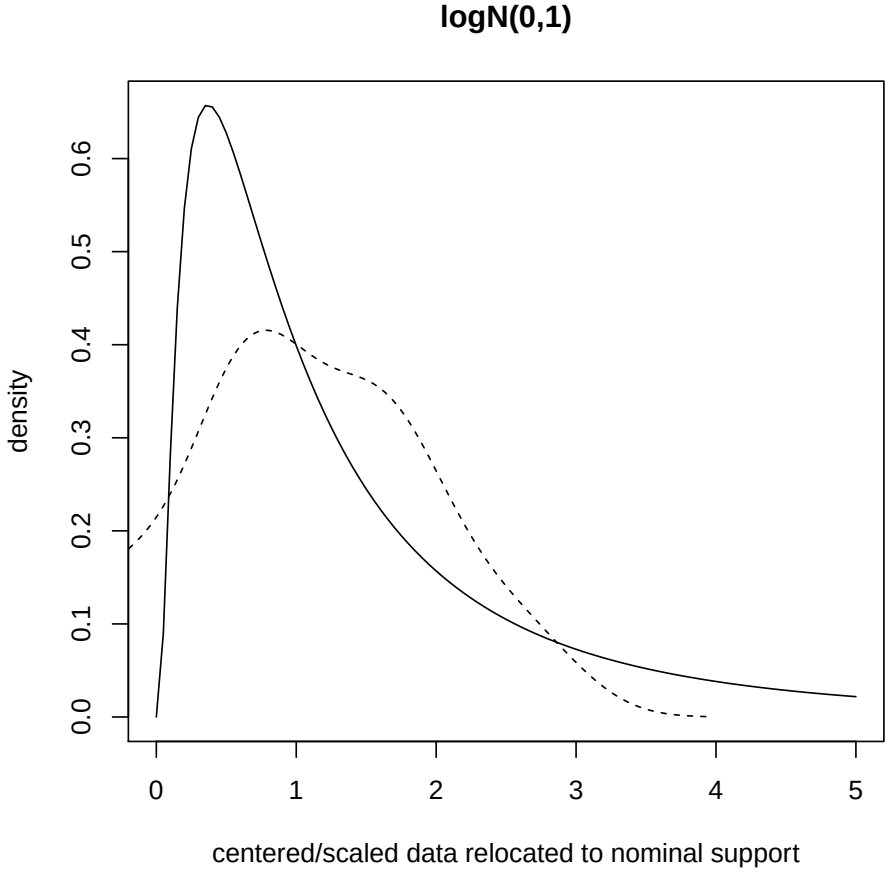
```
    argl <- list(x, CDFname(eddd))
    if (length(Parms <- parms(eddd))>0)
       for (j in 1:length(Parms)) argl[[2+j]] <- Parms[j]
    do.call("ks.test", argl)
})
```

The idea is that (if `is.centered` is TRUE) the data have been transformed to have median 0 and mad 1. The data vector is transformed once more to have the same median and mad as the parametric distribution specified in the `eddDist` object.

```
> x <- rnorm(50)
> print(testVec(x, N01, FALSE))

        One-sample Kolmogorov-Smirnov test

data:  c(-0.607128449988428, 0.442979066209402, -1.70574750718205, 0.531401798933671,
D = 0.094, p-value = 0.7331
alternative hypothesis: two-sided

> print(testVec(x, LN01, FALSE))

        One-sample Kolmogorov-Smirnov test

data:  c(-0.607128449988428, 0.442979066209402, -1.70574750718205, 0.531401798933671,
D = 0.5798, p-value = 8.882e-16
alternative hypothesis: two-sided
```

We see that the p value for the appropriate model (N(0,1)) is fairly high, and that for lognormal is very small. To visualize the discrepancy, we have `plotED`:

```
> plotED(LN01, data=x)
```



To use test-based association with `edd`, specify method="test" and provide a threshold that the maximum p value must exceed for a classification to occur. If no p value exceeds the threshold, then no catalog member is a satisfactory match to the shape of the expression density, and 'outlier' is declared. [If two p-values exceeded the threshold and were close to each other, we should declare 'doubt'. This is not yet handled.]

4.2 k-NN association

If `edd` is used with method="knn" and parameters k and l are provided, k-nearest neighbor classification is performed, treating the expression vector and the elements of the reference catalog as n-dimensional multivariate data. The tx parameter of `edd` determines that the order statistics of the expression vector and catalog elements are used (if tx=sort), or that the comparisons are conducted after transformation of both data and catalog elements to the flatQQnorm space (if tx=flatQQNormY).

4.3 Neural net association

If `edd` is used with method="nnet" and a size (and potentially other parameters) is specified, then classification of the expression vectors occurs using prediction from a neural net fit to the reference catalog. The role of the tx parameter is as described in the previous subsection.