

# HowTo Use the Bioconductor `edd` package

Vince Carey `stvjc@channing.harvard.edu`

April 23, 2010

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Important caveat</b>	<b>2</b>
<b>3</b>	<b>Distributional shapes in Golub's data</b>	<b>2</b>
3.1	Filtering out genes with low variation . . . . .	2
3.2	Forming stratum-specific ExpressionSets . . . . .	2
3.3	Running <code>edd</code> . . . . .	3
3.4	Assessing the results . . . . .	5
<b>4</b>	<b>Extending the reference catalog</b>	<b>7</b>

## 1 Introduction

*edd* is a package that assists with one aspect of exploratory data analysis for microarrays. The basic question addressed in *edd* is the variety of shapes of gene-specific distributions of expression in collections of microarrays. Use of the package is most sensible when there are numerous arrays obtained under the same experimental condition or for a given clinical condition. The key idea is that marginal gene-specific distributions may have a relatively number of different qualitative shapes, some of which may be of considerable substantive interest (e.g., multimodal shapes), and some of which may be of methodologic importance (e.g., when one group of subjects has a skewed distribution for a gene, and another has a symmetric distribution for the same gene, use of a log transform is counterindicated).

In this brief HOWTO, we illustrate directly the use of the *edd* package. We will investigate the diversity of distributions in the two main groups of Golub's leukemia dataset.

## 2 Important caveat

The `edd` function will transform all gene-specific expression distributions to have common location and scale. This process can make noise have the appearance of signal. Before using `edd`, remove all genes that have small variability. See the next section for an example of this filtering process.

## 3 Distributional shapes in Golub's data

First we attach the necessary libraries and data frames. `edd` will require the *golubEsets* library.

```
> library(edd)
> library(golubEsets)
> library(xtable)
> data(Golub_Merge)
```

### 3.1 Filtering out genes with low variation

Next we filter the Golub data to require reasonable dispersion (confine attention to upper half sample defined by size of MAD) and reasonable expression (confine attention to genes with minimum expression level 300).

```
> 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, ]
```

### 3.2 Forming stratum-specific ExpressionSets

Finally we split the dataset into the ALL and AML samples:

```
> ALL <- gmfilt$ALL.AML == "ALL"
> gall <- gmfilt[, ALL == TRUE]
> gaml <- gmfilt[, ALL == FALSE]
> show(gall)
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 540 features, 47 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: 39, 40, ..., 27 (47 total)
```

```

varLabels and varMetadata description:
  Samples: Sample index
  ALL.AML: Factor, indicating ALL or AML
  ....: ...
  Source: Source of sample
  (11 total)
featureData: none
experimentData: use 'experimentData(object)'
pubMedIds: 10521349
Annotation: hu6800

```

### 3.3 Running edd

We will apply edd using an nnet classifier with the default reference catalog. See the edd-Details vignette for information about the reference catalog.

```

> set.seed(12345)
> alldists <- edd(gall, meth = "nnet", size = 10, decay = 0.2)

```

```

# weights: 579
initial value 2078.664026
iter 10 value 1087.941678
iter 20 value 727.152339
iter 30 value 566.312901
iter 40 value 474.859578
iter 50 value 427.967636
iter 60 value 379.962366
iter 70 value 356.614004
iter 80 value 344.562105
iter 90 value 341.268459
iter 100 value 338.252505
final value 338.252505
stopped after 100 iterations

```

```

> amldists <- edd(gaml, meth = "nnet", size = 10, decay = 0.2)

```

```

# weights: 359
initial value 2194.061591
iter 10 value 1280.052365
iter 20 value 968.639680
iter 30 value 855.835315
iter 40 value 759.012398
iter 50 value 707.685549

```

```

iter 60 value 685.109477
iter 70 value 671.800991
iter 80 value 659.021478
iter 90 value 654.918225
iter 100 value 651.330656
final value 651.330656
stopped after 100 iterations

```

An example of the results is given by the classification calls for the first 5 genes in the filtered ExpressionSet:

hum_alu_at	AFFX-HUMGAPDH/M33197_3_at	AFFX-HSAC07/X00351_5_at
".75N(0,1)+.25N(4,1)"	"t(3)"	"t(3)"
AFFX-HSAC07/X00351_3_at	AFFX-M27830_M_at	
"N(0,1)"	"logN(0,1)"	

We can use edd with other classification methods.

```

> set.seed(123)
> alldistsKNN <- edd(gall, meth = "knn", k = 1, l = 0)
> alldistsTEST <- edd(gall, meth = "test", thresh = 0.3)

```

The agreement between nnet and knn procedures is not exact. See table 1. Choice between these methods and selection of tuning parameters is context-dependent.

```

> cap <- "Comparison of distribution shape classification by nnet (rows) and by knn (
> print(try(xtable(latEDtable(table(alldists, alldistsKNN), reorder = greo),
+   digits = rep(0, length(table(alldists)) + 1), caption = cap,
+   label = "conc1"))))

```

The test procedure is the only one at present that allows an outcome of 'doubt'.

```

> print(table(alldistsTEST))

```

alldistsTEST			
.25N(0,1)+.75N(4,1)	.75N(0,1)+.25N(4,1)	B(2,8)	B(8,2)
9	91	169	26
logN(0,1)	N(0,1)	outlier	t(3)
40	68	4	104
U(0,1)	X^2(1)		
26	3		

	$\Phi$	$t_3$	$LN_{\{0,1\}}$	$\chi^2_{2,1}$	$\beta_{\{8,2\}}$	
$\Phi$	46	7	0	0		0
$t_3$	26	52	10	0		0
$LN_{\{0,1\}}$	0	3	71	14		0
$\chi^2_{2,1}$	0	0	1	1		0
$\beta_{\{8,2\}}$	3	0	0	0		8
$U_{\{0,1\}}$	0	0	0	0		0
$\beta_{\{2,8\}}$	24	0	5	0		0
$\frac{3}{4}\Phi + \frac{1}{4}\Phi_{\{4,1\}}$	0	0	4	1		0
$\frac{1}{4}\Phi + \frac{3}{4}\Phi_{\{4,1\}}$	0	1	0	0		1

Table 1: Comparison of distribution shape classification by nnet (rows) and by knn (columns) methods in edd.

### 3.4 Assessing the results

We can assess the relative frequencies of the different shapes in the ALL samples with a table, see Table 2.

```
> cap <- "Frequencies of distributional shapes in filtered ALL data."
> print(xtable(latEDtable(table(alldists), reorder = greo), digits = rep(0,
+   length(table(alldists)) + 1), caption = cap, label = "marg1"))
```

$\Phi$	$t_3$	$LN_{\{0,1\}}$	$\chi^2_{2,1}$	$\beta_{\{8,2\}}$	$U_{\{0,1\}}$	$\beta_{\{2,8\}}$	$\frac{3}{4}\Phi + \frac{1}{4}\Phi_{\{4,1\}}$
66	139	100	2	13	3	161	

Table 2: Frequencies of distributional shapes in filtered ALL data.

We can use barplots also; see Figure 1.

Discordance between distributional shapes in gene expression for the AML and ALL groups can be assessed using the cross-classification, see Table 3.

```
> cap <- "Rows are gene-specific distribution shapes for ALL, columns for AML, and ce"
> print(xtable(latEDtable(table(alldists, amldists), reord = greo),
+   cap = cap, label = "disco1"))
```

Let's see what these discordances mean. To begin, let's get some indices for genes with bimodally shaped expression distribution for ALL, but approximately gaussian expression distribution for AML:

```
> print((1:540)[alldists == ".75N(0,1)+.25N(4,1)" & amldists ==
+   "N(0,1)"][1:5])
```

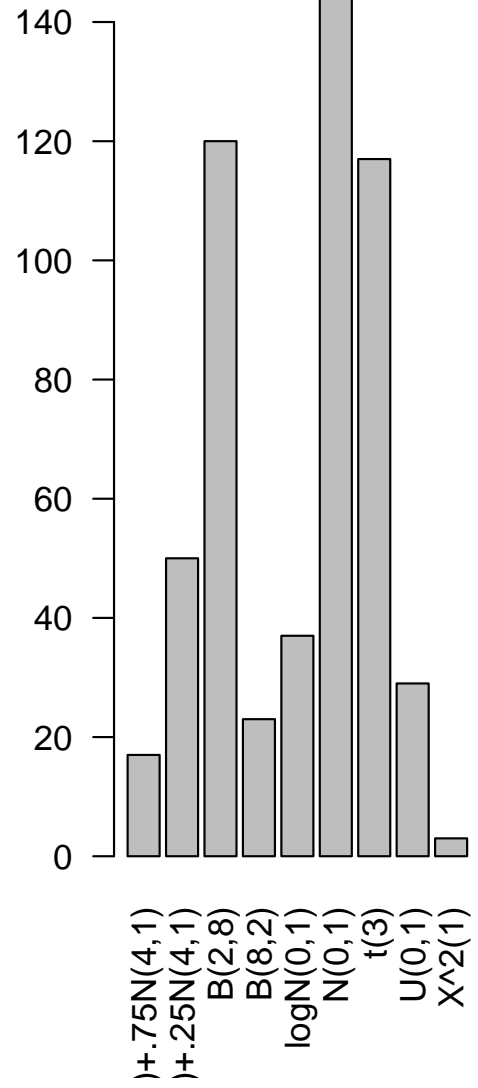
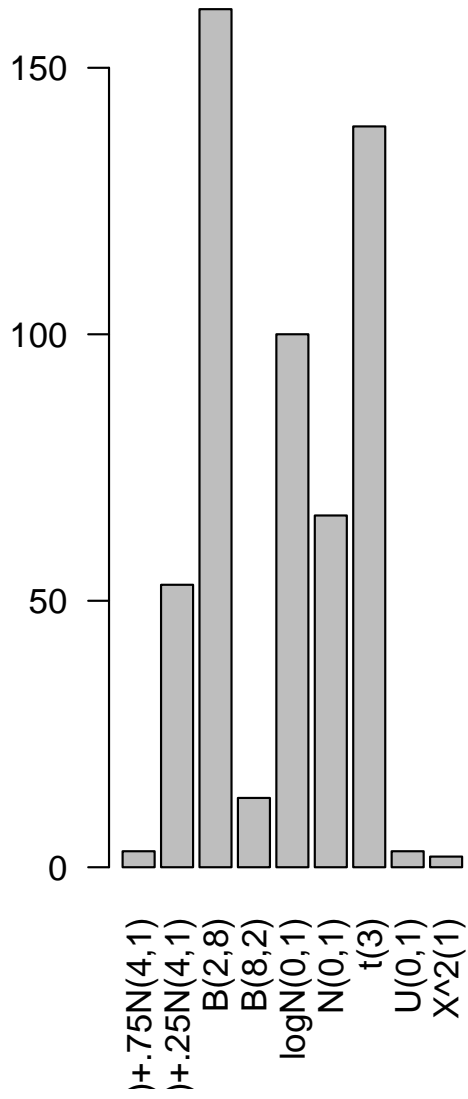


Figure 1: Compositions of distributional shapes within strata.

	$\Phi$	$t_3$	$LN_{0,1}$	$\chi^2_{2,1}$	$\beta_{8,2}$	
$\Phi$	24	13	2	0		5
$t_3$	33	41	8	0		6
$LN_{0,1}$	22	19	12	1		1
$\chi^2_{2,1}$	0	1	0	0		0
$\beta_{8,2}$	2	4	0	0		5
$U_{0,1}$	0	0	0	0		0
$\beta_{2,8}$	44	24	11	2		6
$\frac{3}{4}\Phi + \frac{1}{4}\Phi_{4,1}$	19	13	4	0		0
$\frac{1}{4}\Phi + \frac{3}{4}\Phi_{4,1}$	0	2	0	0		0

Table 3: Rows are gene-specific distribution shapes for ALL, columns for AML, and cell entries are counts of genes.

[1] 37 65 78 135 141

We consider the gene with probe D87953\_at. The top left panel gives the model (solid density trace) and a kernel density estimate applied to the expression levels among ALL patients, and the top right is the corresponding histogram.

While the specific mixture model used as reference is not a perfect fit to the ALL data, the neural net classifier was sensitive to the bimodality. The Gaussian model does not seem particularly appropriate for the AML data, but was the closest match in the reference catalog.

## 4 Extending the reference catalog

The reference catalog supplied with edd has components

```
> names(eddDistList)
```

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

There is nothing sacred about this set. Let's consider its scope (we'll look at 8 of nine reference distributions):

From the example above we see that it might be useful to have a mixture of Gaussians with modes separated by 6SD. To add such a model we construct an instance of the eddDist class:

```
> MIXN3 <- new("eddDist", stub = "mixnorm", parms = c(p1 = 0.75,
+   m1 = 0, s1 = 1, m2 = 6, s2 = 1), median = 0.43, mad = 1.55,
+   tag = ".75N(0,1)+.25N(6,1)", plotlim = c(-3, 11), latexTag = "$\\frac{3}{4}\\Phi + \\frac{1}{4}\\Phi_{4,1}$")
> eddDistList[["MIXN3"]] <- MIXN3
> set.seed(12345)
> alldists2 <- edd(gall, meth = "nnet", size = 10, decay = 0.2)
```

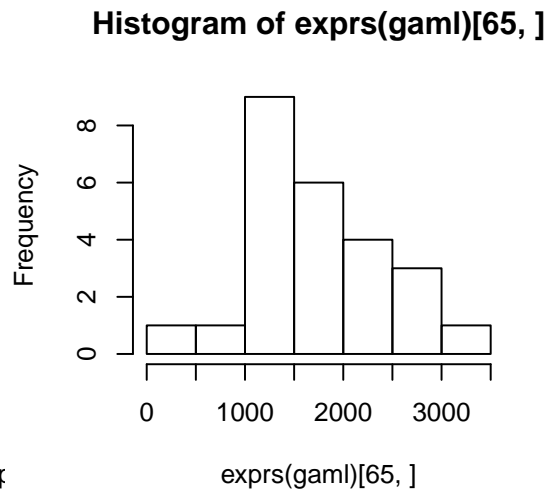
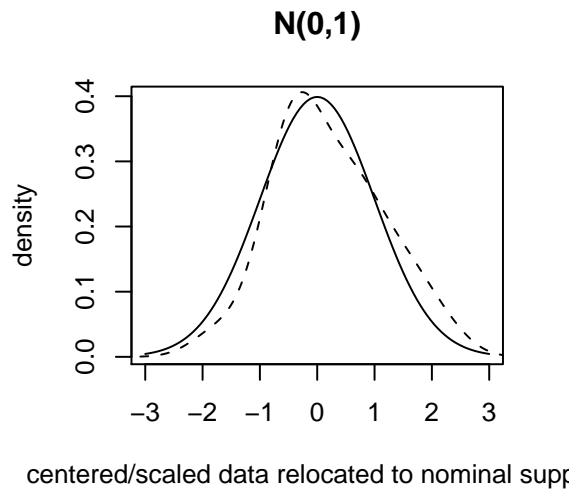
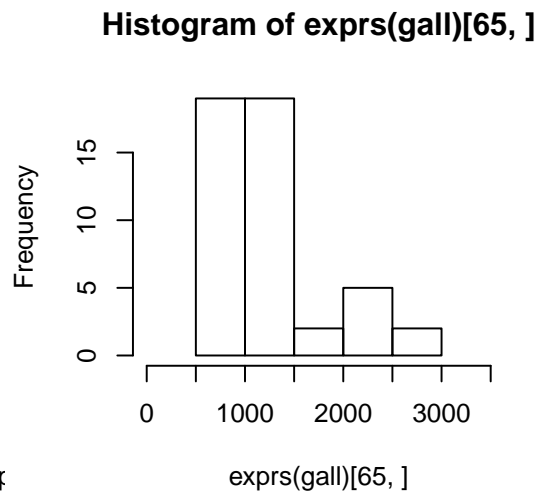
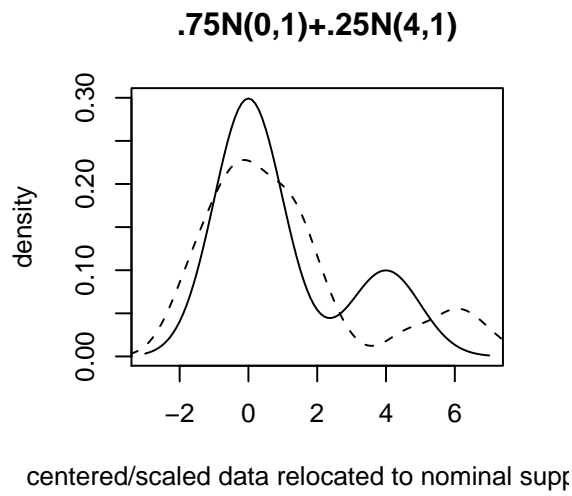


Figure 2: Two models for D87953\_at in ALL and AML patients.



```
> par(mfrow = c(4, 2))
> for (i in 1:8) plotED(eddDistList[[i]])
```

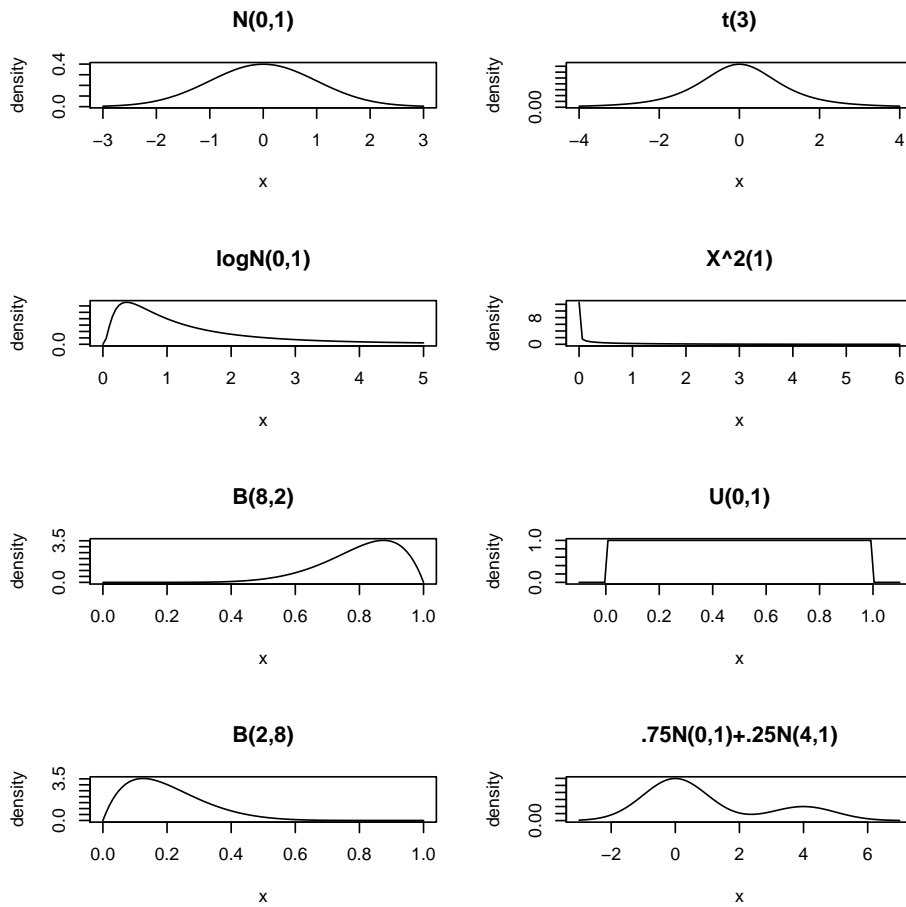


Figure 3: Eight of the reference distributions in the `eddDistList` supplied with *edd*.

```

# weights:  579
initial  value 2078.664026
iter   10 value 1087.941678
iter   20 value 727.152339
iter   30 value 566.312901
iter   40 value 474.859578
iter   50 value 427.967636
iter   60 value 379.962366
iter   70 value 356.614004
iter   80 value 344.562105
iter   90 value 341.268459
iter  100 value 338.252505
final   value 338.252505
stopped after 100 iterations

```

```
> print(alldists2[65])
```

```
[1] ".75N(0,1)+.25N(4,1)"
```

The symbol MIXN3 used to name the list element is arbitrary, as are the values of the tag and latexTag slots. But the user should choose meaningful values for those items. The new reference distribution is used for classification of probe D87953\_at. The two fits for the different mixtures are shown in Figures 4, 5.

```
> plotED(MIXN3, data = exprs(gall)[65, ])
```

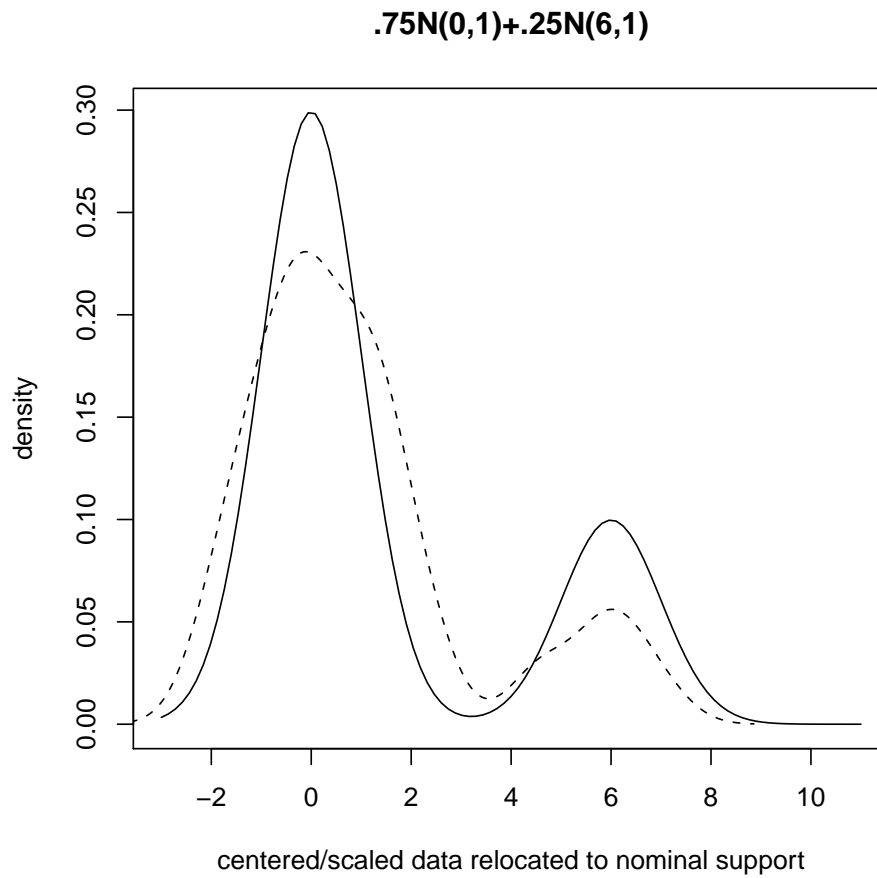


Figure 4: Reference catalog element: mixture with modes separated by 6SD. Superimposed is the kernel smooth of centered/scaled and then translated data for D87953\_at.

```
> plotED(MIXN1, data = exprs(gall)[65, ])
```

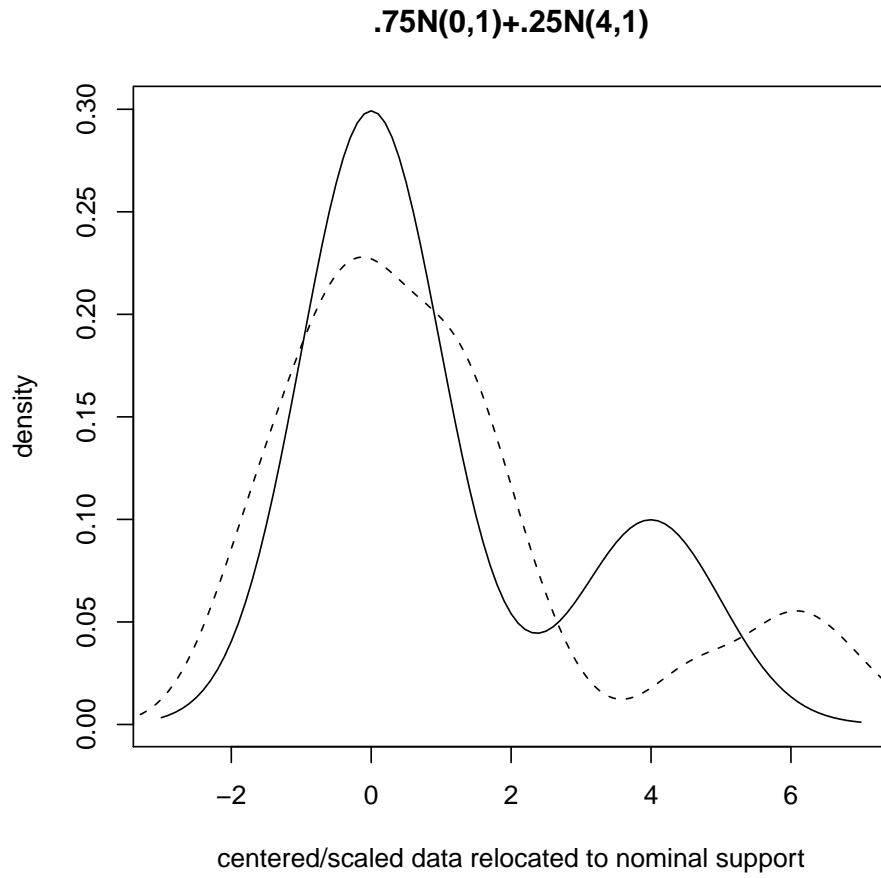


Figure 5: Reference catalog element: mixture with modes separated by 3SD. Superimposed is the kernel smooth of centered/scaled and then translated data for D87953\_at.