Local FDR Simulation Example

Bradley Efron, Brit Turnbull and Balasubramanian Narasimhan
Department of Statistics
Stanford University
Stanford, CA 94305

August 19, 2006

This simulation example involves 2000 "genes", each of which has yielded a test statistic z_i, with $z_i \approx N(\mu_i, 1)$, independently for $i = 1, 2, \ldots, 2000$.

Here μ_i is the "true score" of gene i, which we observe only noisily. 1800 (90%) of the μ_i values are zero; the remaining 200 (10%) are from a $N(3, 1)$ distribution. The data are contained in the dataset `lfdrsim`, where the z_i are the column `zex`.

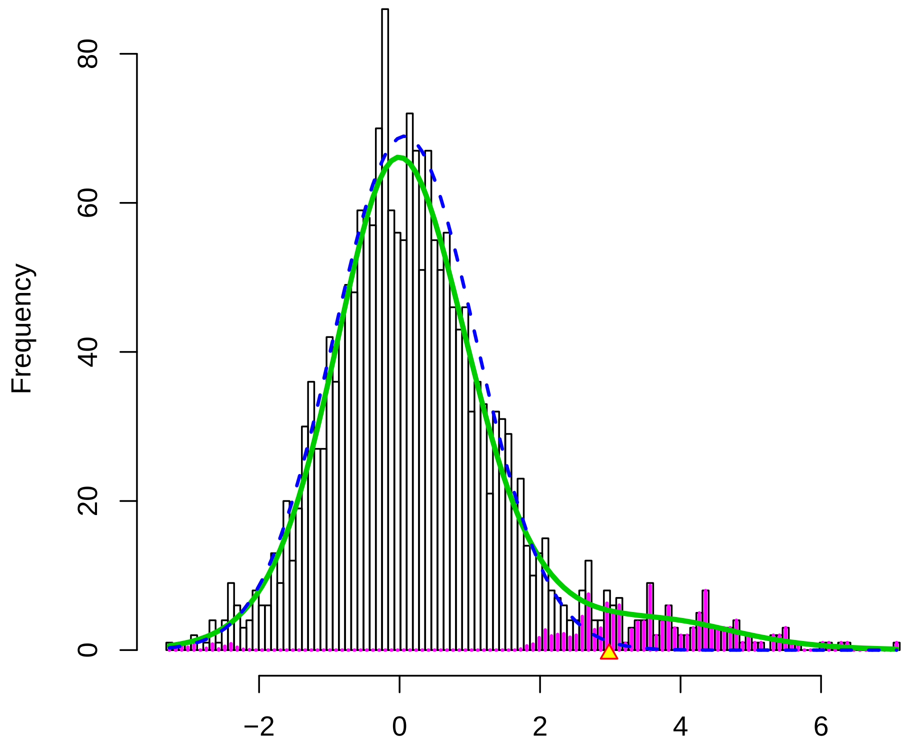
```
> library(locfdr)
```

```
Loading required package: splines
```

```
> data(lfdrsim)
> zex <- lfdrsim[, 2]
```

If we are confident that the null z_i's are distributed as $N(0, 1)$, we run `locfdr` with `nulltype=0`. Otherwise, we use the default `nulltype=1`, which uses empirical estimates of the null density parameters.

```
> w <- locfdr(zex)
```



MLE: delta: 0.071 sigma: 1.016 p0: 0.933
CME: delta: 0.011 sigma: 0.966 p0: 0.908

In the figure, the green solid line is the spline-based estimate of the mixture density f. The blue dashed line is the empirical null subdensity $p_0 f_0$, estimated by default by maximum likelihood (nulltype=1). Whichever nulltype is specified, `locfdr` returns a matrix `fp0` containing parameters of all three nulltypes and corresponding estimates of the proportion p_0 of cases that are null, along with standard errors. In this example, the null distribution is $N(0, 1)$, and both the MLE and central matching estimates come close to this.

```
> w$fp0

           delta      sigma          p0
thest 0.00000000 1.00000000 0.934884830
theSD 0.00000000 0.00000000 0.016381300
mlest 0.07133733 1.01567574 0.932555728
mleSD 0.02761442 0.02721782 0.009518058
cmest 0.01137651 0.96576676 0.908318708
cmeSD 0.04211370 0.03380724 0.013813796
```

The function `locfdr` returns, in the output `mat`, the bin centers `x`, and, at each `x`, the following values:

fdr local false discovery rate based on the specified nulltype

Fdrleft, Fdrright tail false discovery rates

f the mixture density estimate calculated using the type and df arguments, scaled to sum to the number of z_i's.

f0 the null density estimate calculated using the nulltype argument (using nulltype=1 if nulltype=0 is specified)

f0theo the null density estimate calculated using the theoretical null $N(0,1)$

fdrtheo the local false discovery rate based on the theoretical null $N(0,1)$

counts the number of z_i's in the bin

lfdrse the delta-method estimate of the standard error of the log of the local false discovery rate for the specified nulltype

p1f1 the estimated subdensity of the non-null z_i's

```
> w$mat[1:5, ]
```

```
             x       fdr   Fdrleft  Fdrright         f        f0    f0theo
[1,] -3.277130 0.4754348 0.4754348 0.9325557 0.5902186 0.3009048 0.3260307
[2,] -3.189391 0.5222393 0.5010207 0.9326907 0.7117024 0.3985595 0.4329734
[3,] -3.101651 0.5695273 0.5282337 0.9328368 0.8579789 0.5239820 0.5705853
[4,] -3.013912 0.6167842 0.5568976 0.9329928 1.0338087 0.6837521 0.7461681
[5,] -2.926172 0.6634879 0.5867905 0.9331566 1.2447492 0.8856050 0.9682989
       fdrtheo counts    lfdrse      p1f1
[1,] 0.5164208      1 0.3988950 0.3096081
[2,] 0.5687493      0 0.3698064 0.3400234
[3,] 0.6217304      1 0.3411065 0.3693365
[4,] 0.6747682      1 0.3129513 0.3961718
[5,] 0.7272533      2 0.2855029 0.4188731
```

The `fdr` in the result contains the local false discovery rate for each z_i. One might use this vector to create a list of Interesting cases.

```
> which(w$fdr < 0.2)
```

```
 [1]    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
[16]   16   17   18   19   20   21   23   24   25   26   27   28   29   30   31
[31]   32   33   35   37   38   39   41   42   43   45   46   47   48   49   51
[46]   52   54   56   57   58   59   60   61   62   63   66   67   69   70   71
[61]   73   74   75   77   78   79   83   85   88   89   90   92   95   96   98
[76]  100  103  104  106  107  109  112  113  118  121  122  125  127  128  132
[91]  133  135  136  137  141  151  160  161  162  165  168  170 1732 1898
```

Here 0.2 is a rule-of-thumb cut-off. In the simulated data, the first 200 cases have nonzero μ_i. So we can find the true tail FDR.

```
> sum(which(w$fdr < 0.2) > 200)/sum(w$fdr < 0.2)
```

```
[1] 0.01923077
```

The estimated tail FDR can be found from the `mat` output.

```
> w$mat[which(w$mat[, "fdr"] < 0.2)[1], "Fdrright"]
```

```
[1] 0.03515483
```

The tail FDR is the mean local fdr over the entire tail and is therefore smaller than the local fdr cutoff.