User's Guide to the FDR Library

Introduction

This document provides initial guidance on how to use a library of routines implements several methods that use p-values to estimate or control the false discovery rate (FDR; Benjamini and Hochberg 1995) and related multiple testing error measures. The library has been developed and is maintained by the Biostatistics Department of St. Jude Children's Research Hospital in Memphis, TN.

Getting Started

Download the desired version (R or S-plus) of the library and then use the source command to make the routines available for use in the session.

Basic Use of the Library

The end-user will be primarily interested in two functions: `compute.fdr` and `fdr.plots`. The function `compute.fdr` actually performs the FDR calculations; the function `fdr.plots` accepts the results of `compute.fdr` to generate some plots to illustrate and assess the quality of the results.

The function `compute.fdr` has one required argument, `p`, which can be a vector or matrix of p-values. If `p` is a vector, then `compute.fdr` will apply the selected FDR method to `p`. If `p` is a matrix, then `compute.fdr` will apply the selected FDR method to each column of `p` separately.

The function `compute.fdr` has three optional arguments: `fdr.method`, `opts`, and notes. The argument `fdr.method` accepts a string that specifies which FDR method will be applied. Presently (Jan 23, 2006), `compute.fdr` can implement six methods: Benjamini and Hochberg (1995), Benjamini and Yekutieli (2001), Storey (2002), Pounds and Morris (2003), Pounds and Cheng (2004), and Cheng et al. (2004). Pounds (2006) briefly describes and compares these methods. The table below indicates what value of `fdr.method` to specify to implement each of these methods.

Available Options for `fdr.method`

Set `fdr.method` =	**To Implement**
"BH95"	Benjamini and Hochberg (1995)
"BY01"	Benjamini and Yekutieli (2001)
"St02"	Storey (2002)
"PM03"	Pounds and Morris (2003)

“PC04”	Pounds and Cheng (2004)
“Ch04”	Cheng et al. (2004)

The `opts` argument of `compute.fdr` allows the user to specify certain tuning parameters for some of the algorithms, such as the bandwidths for St02 and PC04 and the knot sequence for Ch04. More information is provided below in the section “Speciyfing Options.” Defaults have been provided for these options.

The notes argument of `compute.fdr` accepts a string or string vector. The notes arguments allows the user to associate some comments with the results of the analysis. For example, the user may want to remember that the p-values were generated using a rank-sum test. The user could then set notes = “P-values generated by rank-sum comparison of control and experimental groups.” This string will be included in the notes component of the result, along with any notes that `compute.fdr` generates during its calculations.

The function `compute.fdr` returns a list object with several components:

Components of the Object Returned by `compute.fdr`

Name	**Description**
`p`	the value of p supplied to `compute.fdr`
`fdr.method`	a string giving the name of the method applied
`fdr`	local FDR estimates (Benjamini and Hochberg 2000) for the corresponding entries of `p`
`q`	q-values (or FDR-adjusted p-values) for the corresponding entries of `p`
`ord`	a vector or matrix giving the indices to order `p`, `fdr`, `q`, `cdf`, `ebp`, `pdf`, `fp`, `fn`, and `toterr` in order of ascending p-value. If these components are matrices, then each column of `ord` can be used to order the entries of the corresponding column by order of p-value in the corresponding column.
`fdr.method`	echoes the value of the argument `fdr.method`
`pi`	included only if applicable, gives the estimated proportion of tests having a true null hypothesis. If `p` is a matrix, then pi is a vector with each entry giving the estimate for the corresponding column of `p`.

`cdf` a cumulative distribution function (CDF) estimate used to form the denominator of `fdr`

`ebp` if applicable, gives the estimated empirical Bayes posterior probability (Pounds and Morris 2003) that the null hypothesis is true for the corresponding entries of `p`

`pdf` included only if applicable, gives the probability distribution function of the fitted model for the corresponding entries of `p`.

`fp` gives the estimated proportion of all results (within the corresponding column of `p`) with p-value less than or equal the corresponding entry of `p` that are expected to be false positives, i.e. the numerator of `fdr`

`fn` if applicable, gives the estimated proportion of all results (within the corresponding column of `p`) with p-value greater than the corresponding entry of `p` that are expected to be false negatives

`toterr` gives the sum of `fp` and `fn`

`notes` a string giving some notes about the analysis that was performed, including notes provided by the user through the notes argument.

`a` For Pounds and Morris (2003) method, the maximum likelihood estimate of the parameter *a* of the beta-uniform mixture model

`lambda` For Pounds and Morris (2003) method, the maximum likelihood estimate of the parameter λ of the beta-uniform mixture model

`adj.eps` For Pounds and Morris (2003) method, the value of the `adj.eps` option (see “Specifying Options” section below).

`n.adj` For Pounds and Morris (2003) method, the value of the `n.adj` option (see “Specifying Options” section below).

`lam` For Storey’s (2002) method, the value of the parameter λ used to estimate the null proportion

`robust` For Storey’s (2002) method, the value of the `robust` option (see “Specifying Options” section below)

`LC` For Pounds and Cheng (2004) method, the value of `loess.control` used in the loess regression (see “Specifying Options” section below).

`dplaces`	For Pounds and Cheng (2004) method, the value of `dplaces` used for pre-rounding purposes (see “Specifying Options” section below).
`p.pi`	For Pounds and Cheng (2004) or Cheng et al (2004) methods, gives the p-value at which the p-value PDF estimate is minimized
`qpos`	For Cheng et al (2004), specify the position of spline knots for FDR estimation

Below is a very simple example of how to use `compute.fdr`. It also shows how to use `fdr.plots`, which simply accepts the results of fdr.compute as an argument.

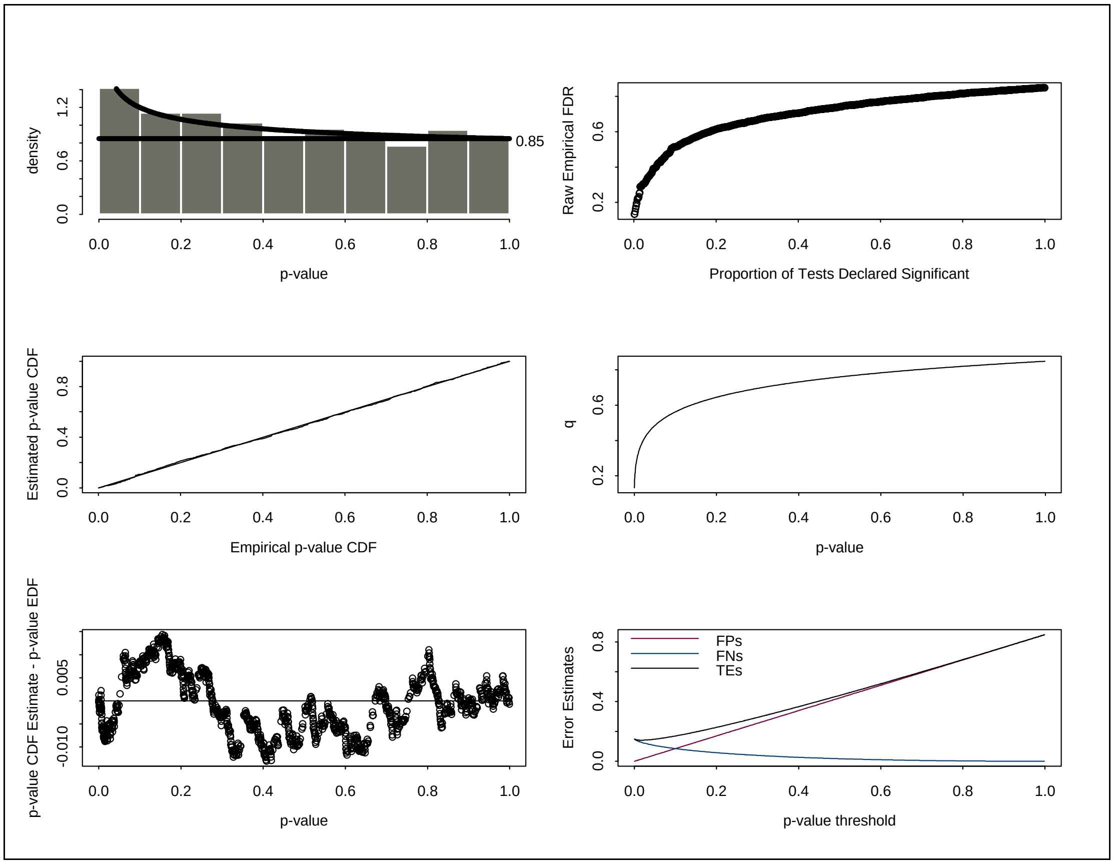
Example of the use of `compute.fdr`

```
# generate p-values to demonstrate use of compute.fdr
pvals<-rbeta(1000,0.8,1)

# Perform calculations using Pounds and Morris (2003)
          method, save results in fdr.demo
fdr.demo<-compute.fdr(pvals,fdr.method=”PM03”)

# Look at q-value curve
plot(fdr.demo$p,fdr.demo$q)

# Look at other plots, using
fdr.plots(fdr.demo)

# Create a table, ordered by ascending p-value
demo.table<-cbind(p=fdr.demo$p,q=fdr.demo$q)[fdr.demo$ord,]

# Look at first 100 rows of the table
demo.table[1:100,]
```

Interpreting the Figures Produced by `fdr.plots`

The function `fdr.plots` produces several figures, depending on which figures are applicable given the value of `fdr.method` in the object returned by `compute.fdr`.

Example of Figures Produced by `fdr.plots`



The top left figure gives a p-value histogram with the probability density function of the fitted model plotted against it. It also includes a horizontal line indicating the estimate of the null proportion, i.e., the proportion of tests for which the null hypothesis is true. This figure can help in assessing the fit of the p-value model and whether the estimate of the null proportion is reasonable. The horizontal line should be at or above the height of the shortest bar for conservative FDR estimation or control. Also, the fitted model should agree with the histogram to indicate good model fit. This plot is produced only for methods that use an estimate of the null proportion or fit a model to the p-values.

The center left figure plots the estimated p-value CDF against the empirical p-value CDF. This plot is analogous to the quantile-quantile plot, but instead plots the estimated and empirical CDFs against one another. Good model fit is shown if the curve falls along the line y=x. The CDF-CDF plot was chosen instead of the more popular quantile-quantile plot as a display of model fit because it is more easily computed. This figure is included only for methods that use a quantity other than the p-value EDF in the denominator of the local FDR estimates (Benjamini and Hochberg 2000).

The bottom left figure plots the difference between the empirical and estimated p-value CDF against the p-value. This plot is intended to make subtle differences between the empirical p-value CDF and estimated p-value CDF more apparent. The vertical axis of this graph is very important, because even small differences can appear large in this plot. The FDR estimates may have some liberal bias in intervals where the curve is less than 0, because the denominator used by the method is smaller than the p-value EDF. The FDR estimates may have some conservative bias in intervals where the curve is greater than 0, because the denominator used by the method is greater than the p-value EDF. This plot is only produced for methods that fit a model, parametric or nonparametric, to the observed p-value distribution.

The top right figure plots the local FDR estimates (Benjamini and Hochberg 2000) against the p-value. This figure can highlight instability in the local FDR estimates, which can lead to liberal bias in q-value estimates (Pounds and Cheng 2004). In this particular example, the local FDR estimate is smooth because the model-based approach of Pounds and Morris (2003) was used. When methods based on the p-value EDF are employed, this plot may show instability for small p. In such cases, a model based approach may help to reduce the instability and yield more reliable q-value estimates.

The center right figure plots the q-value estimate against the p-value threshold. This may be the most useful figure for illustrating the final results of a method.

The bottom right figure plots the estimated proportion of tests resulting in false positives, false negatives, and the total number of errors as a function of the p-value threshold. This plot is only produced for methods that estimate the null proportion. It can help illustrate the threshold selection by minimizing the total error criterion (Cheng et al 2004; Pounds and Cheng 2005).

Specifying Options

The `opts` argument of `compute.fdr` allows the user to specify values for tuning parameters of some methods, such as the bandwidth in Storey's (2002) method for estimating the null proportion, the bandwidth in the LOESS regression in the method of Pounds and Cheng (2004), and knot selection for the method of Cheng et al (2004).

For Storey's (2002) method, the `opts` argument must be a list object with components `lambda` and `robust`. The lambda component gives the value of λ in Storey's (2002) method for computing the estimate of the null proportion. The robust option is a logical value, either `T` or `F`. The defaults are `lambda=NULL` (data-driven selection of `lambda)` and `robust=F`.

For the method of Pounds and Morris (2003), the `opts` argument must be a list object with components `adj.eps` and `n.adj`. These are used to slightly modify p-values if any p-value equals (the model PDF is undefined at p=0). The method will be applied to a new set of p-values given by `(n.adj*p+adj.eps)/(n.adj+2*adj.eps)`, where `p` is the set of original p-values, which will prevent numerical problems arising from a p-value that numerically equals zero. The defaults are `adj.eps=0.05` and `n.adj=length(p).`

For the method of Pounds and Cheng (2004), the `opts` argument must be a list object with components LC and dplaces. The dplaces argument specifies the number of decimal places for pre-rounding of p-values. The LC component is a list with the structure of the loess.control argument of the loess function that is built into S-plus. The defaults are dplaces=4 and LC=loess.control(), which returns the S-plus defaults for loess regression.

For the method of Cheng et al (2004), the `opts` argument must be a list object with component `qpos`. The `qpos` component contains a vector giving the location of the knots for the spline-based FDR estimation.

References

Benjamini Y and Hochberg Y. (1995) "Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, 57, 289-300.

Benjamini Y and Hochberg Y. (2000) On the adaptive control of the false discovery rate in multiple testing with independent statistics. *J. Edu. Behav. Stat.*, 25, 60-83.

Benjamini Y and Yekutieli D. (2001) The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29, 1165-1188.

Cheng C, Pounds S, Boyett JM, Pei D, Kou M-L, and Roussel MF (2004). Significance Threshold Selection Criteria for Massive Multiple Comparisons with Applications to DNA Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, 3: 36. [http://www.bepress.com/sagmb/vol3/iss1/art36]

Pounds S (2006) Estimation and Control of Multiple Testing Error Rates for the Analysis of Microarray Data. *Briefings in Bioinformatics*, in press.

Pounds S and Cheng C (2004). Improving False Discovery Rate Estimation. *Bioinformatics*, 20: 1737-1745.

Pounds S and Cheng C (2005). Statistical Development and Evaluation of Gene Expression Data Filters. *Journal of Computational Biology*, 12: 482-495.

Pounds S and Morris S (2003). Estimating the Occurrence of False Positives and False Negatives in Microarray Studies by Approximating and Partitioning the Empirical Distribution of p-values. *Bioinformatics*, 19: 1236-1242.

Storey JD (2002) "A direct approach to false discovery rates." *Journal of the Royal Statistical Society B*, 64, 479-498.