Bioconductor's `qvalue` package
Version 1.99.0

John D. Storey and Andrew J. Bass
Princeton University
http://genomine.org/contact.html

March 27, 2015

Contents

1 Introduction

The `qvalue` package performs false discovery rate (FDR) estimation from a collection of p-values or from a collection of test-statistics with corresponding simulated null statistics. This package produces esitmates of three key quantities: q-values, the proportion of true null hypotheses (denoted by π_0), and local false discovery rates.

When carrying out multiple hypothesis tests, one typically starts either with a set of p-values or test-statistics. Either quantity yields a natual ordering of tests from most significant to least significant. For example, using p-values one would order the tests from smallest p-value (most significant) to largest p-value (least significant). As another example, using F-statistics one would order the tests from largest F-statistic (most significant) to smallest F-statistic (least significant).

One may then ask: "If I draw a significance threshold somewhere along this list, how much can I trust the top of the list, i.e., those I choose to call statistically singiciant?" Another possible question is: "Where should I draw a line of significance along this list so that we can expect that at most 10% of the list I call significant is composed of false positives?" We may also wish to know the reliability of a set of tests called significant for all possible thresholds simultaneously or we may want to estimate the probability that any given test is a true null hypothesis.

The `qvalue` package forms various estimates that allow one to answer these and other questions. The quantity of interest is the false discovery rate – sometimes abbreviated as FDR – which is roughly defined to be the expected proportion of false discoveries (also known as false positives) among all tests that are called significant.

An overview of the FDR and its well-established methods and theory may be found in Storey (2011) [5] (preprint freely available at http://genomine.org/papers/Storey_FDR_2011.pdf). We recommend this paper for users of `qvalue` unfamiliar with FDR, q-value, and local FDR estimation.

2 Citing this package

The statistical techniques implemented in the package come from the following publications.

Proposed the key strategy and derived the main estimators used in this package.
STOREY, J. D. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B 64* (2002), 479–498.

Developed and proved theorems showing a direct relationship between FDR and Bayesian classification, giving a direct Bayesian version and interpretation of the quantities estimated in this package.
STOREY, J. D. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics 31* (2003), 2013–2035.

Proposed that the FDR and q-value estimators from Storey (2002) [3] be used in a wide range of genomics studies as a way to determine statistical significance.
STOREY, J. D., AND TIBSHIRANI, R. Statistical significance for genome-wide experiments. *Proceedings of the National Academy of Sciences 100* (2003), 9440–9445.

Unified the point estimation apporach of Storey (2002) [3] with the more traditional sequential p-values method approaches from the multiple hypothesis testing literature (e.g., Benjamini and Hochberg 1995 [1]), and proved a number of theorems establishing that the methods in this package provide conservative FDR estimation and control for fixed FDR levels, fixed significance thresholds, and over all levels or thresholds simultaneously.
STOREY, J. D., TAYLOR, J. E., AND SIEGMUND, D. Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B 66* (2004), 187–205.

Provides a concise summary of the main methods and theory on FDR.
STOREY, J. D. False discovery rates. In *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Springer, 2011.

3 Getting help

Many questions about `qvalue` will hopefully be answered by this documentation and references therein. As with any R package, detailed information on functions, their arguments and values, can be obtained in the help files. To view the help for `qvalue` within R, type

```r
help(package = "qvalue")
```

If you identify bugs related to basic usage please contact the authors directly. Otherwise, any questions or problems regarding `qvalue` will most efficiently be addressed on the Bioconductor support site, https://support.bioconductor.org/.

4 Quick start guide

Given a set of p-values, the q-values can be calculated by using the `qvalue` function:

```r
data(hedenfalk)
pvalues <- hedenfalk$p
qobj <- qvalue(pvalues)
qvalues <- qobj$qvalues
```

Additionally, q-values can be calculated given a set of observed and statistics generated from the null distribution:

```r
data(hedenfalk)
obs.stat <- hedenfalk$stat
null.stat <- hedenfalk$stat0
pvalues <- empPvals(stat = obs.stat, stat0 = null.stat)
qobj <- qvalue(pvalues)
qvalues <- qobj$qvalues
```

The following sections of the manual go through a case study to show additional features of the `qvalue` package.

5 Case study: differential gene expression

We demonstrate the functionality of this package using gene expression data from the breast cancer study of Hedenfalk et al. (2001) [2]. The data set is included with the `qvalue` package:

```r
data(hedenfalk)
names(hedenfalk)

## [1] "p"     "stat"  "stat0"
```

The list has three variables: `p`, `stat`, `stat0`. The `p` variable contains the p-values of the data set, the `stat` contains the alternative statistics and the `stat0` contains the null statistics.

The comparison was made between two types of genetic mutation that are associated with an increased

risk of breast cancer, BRCA1 and BRCA2. There were 7 and 8 cDNA arrays for BRCA1 and BRCA2, respectively. The example considered here is restricted to 3170 genes as described in Storey and Tibshirani (2003) [7].[1]

5.1   Checking the p-value histogram

To get a feel for the data, it is essential to view the p-value histogram:

```
hist(hedenfalk$p, nclass = 20)
```
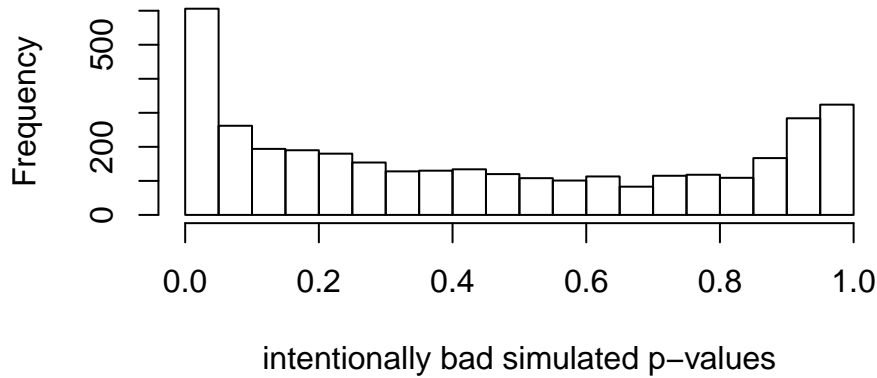
Histogram of hedenfalk$p
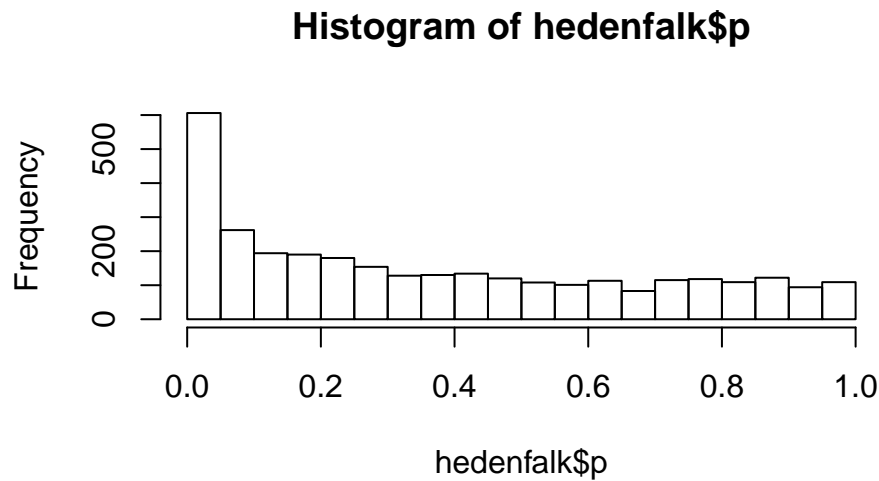
The p-values are relatively flat at the end of the histogram which is important to determine an accurate π_0 estimate. Suppose the p-value histogram instead looked like this simulated set of p-values:

---

[1]The original data and code for pre-processing can be found at http://genomine.org/qvalue.

Problematic p–values

The "U-shaped" p-value histogram is a red flag. The key assumption behind the estimation peformed in this package is that null p-values follow a Uniform(0,1) distribution, which would result in a p-value histogram where the right take is fairly flat as in the Hedenfalk p-values. U-shaped p-value histograms can indicate that a one-sided test was performed on data where there is signal in both directions, or it can indicate that there is dependence among the variables in the data. In the latter case, we suggest considering the `sva` Bioconductor package. In either case, it is usually possible to compute the p-values using a different model or method that will yield p-values that better match the underlying assumptions of the methods implemented in this package.

5.2 Calculating q-values

Once you've examined the distribution of the p-values and confirmed they are well-behaved, the function `qvalue` can be used to calculate the q-values:

```
qobj <- qvalue(p = hedenfalk$p)
```

Several arguments can be used in the function `qvalue`. The following lists the main arguments:

- `p`: A vector of p-values. This is the only necessary input.

- `lambda`: The values of the tuning parameter to be considered in estimating π_0. These must be in [0,1] and are set to lambda=seq(0, 0.95, 0.05) by default. Optional; see Storey (2002) [3] for more information.

- `pi0.method`: Either "smoother" or "bootstrap"; the method for automatically choosing tuning parameter lambda in the estimate of π_0. If the lambda argument above is only given one value, then this option is ignored. The default option is "smoother".

- `fdr.level`: The level at which to control the false discovery rate. Optional; if this is selected, a vector of TRUE and FALSE is returned in the `fdr.level` slot that specifies whether each q-value is less than fdr.level or not.

- **pfdr**: An indicator of whether it is desired to make the estimate more robust for small p-values. This uses the point estimate of "positive false discovery rate" (pFDR). Optional; see Storey (2002) [3] for more information.

The user has the most influence on choosing how to estimate π_0, the overall proportion of true null hypotheses, via `lambda` and `pi0.method`. If no options are selected, then by default the smoother method (`pi0.method = "smoother"`) proposed in Storey and Tibshirani (2003) [7] is used. This is recommended over the bootstrap method, but can be unpredictable for a small number of p-values or in pathological situations. An alternative less susceptible to these scenarios is the bootstrap method (`pi0.method = "bootstrap"`) proposed in Storey, Taylor & Siegmund (2004) [6].

If one selects `lambda=0` (which estimates π_0 as 1) and `fdr.level = 0.05`, then this produces a list of significant tests equivalent to the Benjamini and Hochberg (1995) [1] methodology at level $\alpha = 0.05$ (where, of course, 0.05 can be substituted for any number in $(0, 1]$). This can be viewed as a special conservative case of the Storey (2002) [3] methodology.

5.3 The `qvalue` object

Running the function `qvalue` in the previous section returns a `qvalue` object. A `qvalue` object can be summarized by using the `summary` function:

```
summary(qobj)

##
## Call:
## qvalue(p = hedenfalk$p)
##
## pi0: 0.669926
##
## Cumulative number of significant calls:
##
##            <1e-04 <0.001 <0.01 <0.025 <0.05 <0.1
## p-value        15     76   265    424   605  868
## q-value         0      0     1     73   162  319
## local FDR       0      0     3     30    85  167
##             <1
## p-value   3170
## q-value   3170
## local FDR 2241
```

The summary provides a nice way of viewing the π_0 estimate and the number of significant tests at various cutoffs for different metrics. The cutoffs printed in the `summary` function can be controlled by changing the `cuts` argument.

The object contains several fields:

```
names(qobj)

## [1] "call"       "pi0"        "qvalues"
## [4] "pvalues"    "lfdr"       "pi0.lambda"
## [7] "lambda"     "pi0.smooth"
```

6

A description of each is described below.

- `call`: The function call.

- `pi0`: An estimate of the proportion of null p-values.

- `qvalues`: A vector of the estimated q-values.

- `pvalues`: A vector of the original p-values.

- `lfdr`: A vector of estimated local FDR values.

- `significant`: If fdr.level is specified, and indicator of whether the q-value fell below fdr.level (taking all such q-values to be significant controls FDR at level fdr.level).

- `pi0.lambda`: An estimate of the proportion of null p-values at each `lambda` value.

- `lambda`: A vector of `lambda` values utilized in forming a set of π_0 estimates.

One important number that is obtained with the software is an estimate of the overall proportion of true null hypotheses, π_0:

```
pi0 <- qobj$pi0
```

An estimate of the proportion of true alternative tests is one minus this number. This can be a useful number to know, even if all the truly significant tests cannot all be explicitly identified.

Another important estimate that is returned is the local FDR:

```
lfdr <- qobj$lfdr
```

The p-values, estimated q-values, and estimated local FDR values can also be respectively listed by the following commands:

```
pvalues <- qobj$pvalues
qvalues <- qobj$qvalues
local_FDR <- qobj$lfdr
```

Each entry of `qobj$qvalues` is an estimate of the respective test's q-value, which measures the proportion of false positives incurred (called the false discovery rate) when that particular test is called significant. Calling all tests significant with q-value $\leq \alpha$ controls the FDR at level α. Each entry of `qobj$lfdr` is an estimate of the posterior probability that the null hypothesis is true given the test's p-value, called the local FDR.

5.4   Visualizing results

The `hist` or `plot` function can be used to visualize the final results. The function `plot` allows one to view several useful plots:

- The estimated π_0 versus the tuning parameter λ

- The q-values versus the p-values

- The number of significant tests versus each q-value cut-off

- The number of expected false positives versus the number of significant tests

Applying `plot` to the `hedenfalk qvalue` object, we get:

```
plot(qobj)
```
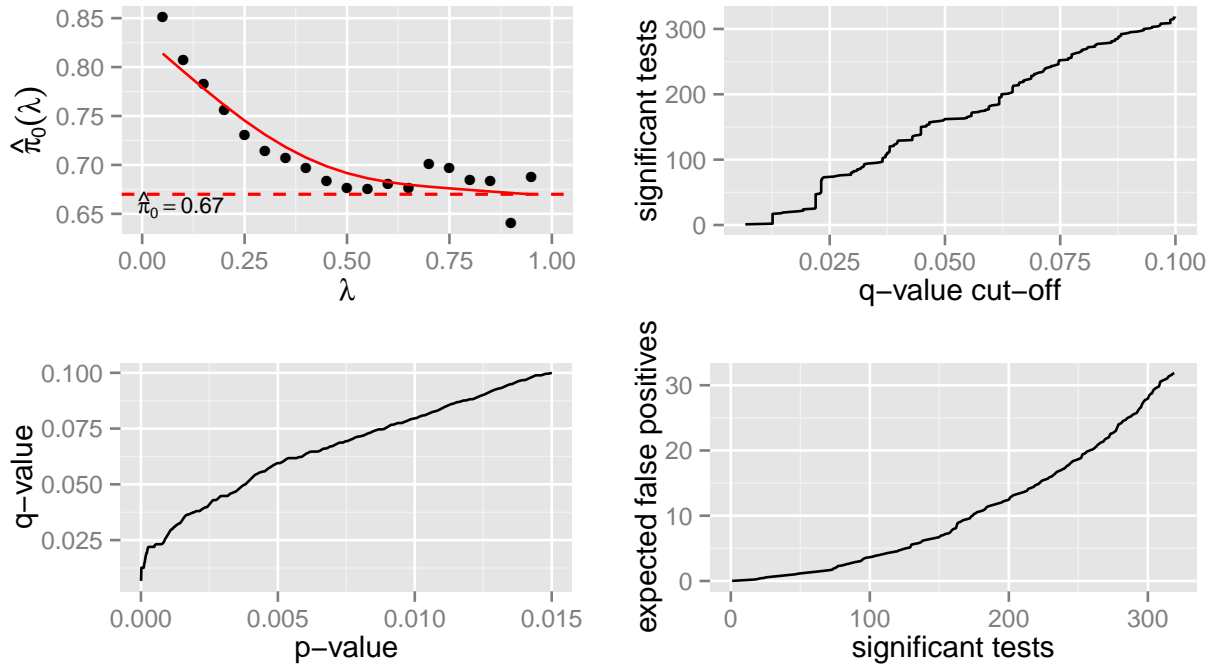

The main purpose of the first plot is to gauge the reliability of the π_0. The tuning parameter λ has to be deal with in order to obtain an overal estimate of π_0. The variable λ is called `lambda` in the package; it can be fixed or automatically chosen. The estimated π_0 is plotted versus the tuning parameter λ. As λ gets larger, the bias of the estimate decreases, yet the variance increases. See Storey (2002) [3] for more on this. Comparing your final estimate of π_0 to this plot gives a good sense as to its quality. A smoother is fit to the plot in order to elucidate the trend of the estimates. The remaining plots show how many tests are significant, as well as how many false positives to expect for each q-value cut-off.

Additionally, running `hist` on a q-value object can be used to view the histogram of p-values along with line plots of both q-values and local FDR values versus the p-values:

```
hist(qobj)
```

p–value density histogram
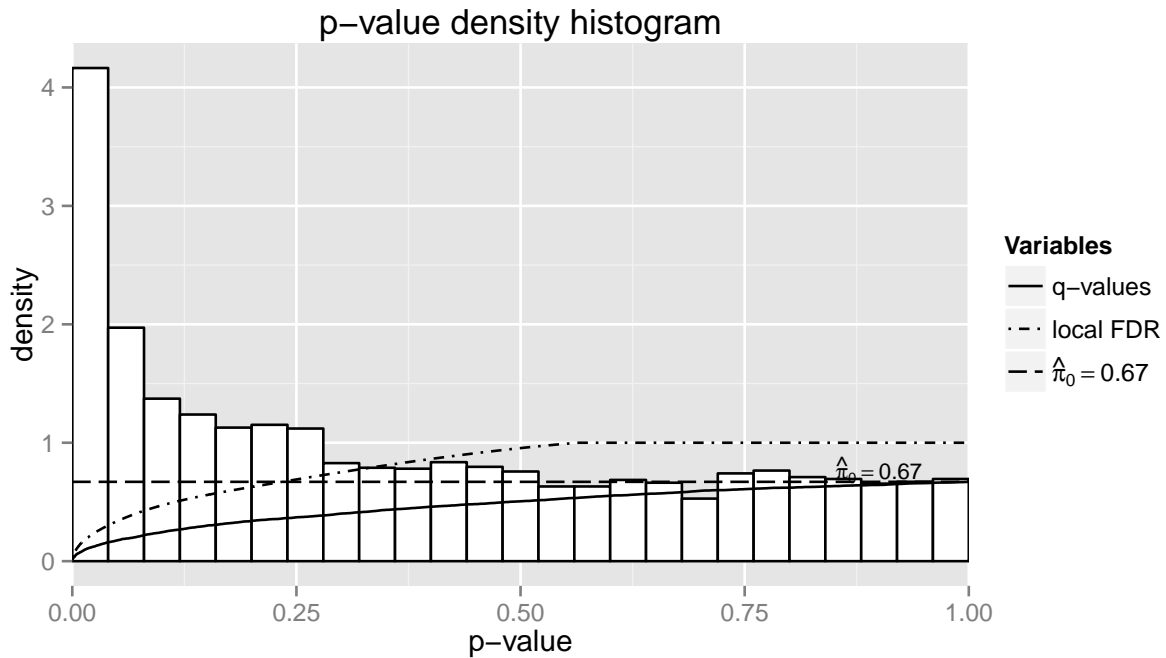
6 Point-and-click implementation

A Shiny implementation of the package written by Andrew Bass can be found at `http://qvalue.princeton.edu`.

7 Frequently asked questions

1. This package produces "adjusted p-values", so how is it possible that my adjusted p-values are smaller than my original p-values?

First, the q-value is not an "adjusted p-values" *per se*, but rather a population quantity with an explicit definition. The package produces estimates of q-values and the local FDR, both of which are very different from p-values. The package does not perform a Bonferroni correction on p-values, which returns "adjusted p-values" that are larger than the original p-values. Second, the maximum possible q-value is π_0, the proportion of true null hypotheses. The maximum possible p-value is 1. When considering a large number of hypothesis tests where there is a nontrivial fraction of true alternative p-values, we will have both an estimate $\pi_0 < 1$ and we will have some large p-values close to 1. Therefore, the maximal estimated q-value will be less than or equal to the estimated π_0 but there will also be a number of p-values larger than the estimated π_0. It must be the case then that at some point p-values become larger than estimated q-values. This is expected and it is not a bug.

Acknowledgements

References

[1] BENJAMINI, Y., AND HOCHBERG, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B 85* (1995), 289–300.

[2] HEDENFALK, I., DUGGAN, D., CHEN, Y., RADMACHER, M., BITTNER, M., SIMON, R., MELTZER, P., GUSTERSON, B., ESTELLER, M., KALLIONIEMI, O., WILFOND, B., BORG, A., AND TRENT, J. Gene expression profiles in hereditary breast cancer. *New England Journal of Medicine 344* (2001), 539–548.

[3] STOREY, J. D. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B 64* (2002), 479–498.

[4] STOREY, J. D. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics 31* (2003), 2013–2035.

[5] STOREY, J. D. False discovery rates. In *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Springer, 2011.

[6] STOREY, J. D., TAYLOR, J. E., AND SIEGMUND, D. Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B 66* (2004), 187–205.

[7] STOREY, J. D., AND TIBSHIRANI, R. Statistical significance for genome-wide experiments. *Proceedings of the National Academy of Sciences 100* (2003), 9440–9445.