Bioconductor's `qvalue` package
Version 2.43.0

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

October 30, 2025

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 empirical null statistics. This package produces estimates 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 natural 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 significant?" 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) [1] (preprint freely available at http://genomine.org/papers/Storey_FDR_2011.pdf). We recommend this paper for users of `qvalue` who want a quick start and are 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. We ask that you cite the most appropriate paper(s) from this list when reporting results from the `qvalue` package.

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

J. D. Storey. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics*, 31:2013–2035, 2003.
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.

J. D. Storey and R. Tibshirani. Statistical significance for genome-wide experiments. *Proceedings of the National Academy of Sciences*, 100:9440–9445, 2003.
Proposed that the FDR and q-value estimators from Storey (2002) [2] be used in a wide range of genomics studies as a way to determine statistical significance.

J. D. Storey, J. E. Taylor, and D. Siegmund. Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B*, 66:187–205, 2004.
Unified the point estimation approach of Storey (2002) [2] with the more traditional sequential p-values method approaches from the multiple hypothesis testing literature (e.g., Benjamini and Hochberg 1995 [6]), 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.

J. D. Storey. False discovery rates. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*. Springer, 2011. http://genomine.org/papers/Storey_FDR_2011.pdf.

Provides a concise summary of the main methods and theory on FDR.

The format of a citation to the `qvalue` package itself is obtained from:

```
citation("qvalue")
```

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

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

If you identify bugs related to basic usage please contact the authors directly, preferably via GitHub at https://github.com/jdstorey/qvalue. 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 `qvalue` object can be calculated by using the `qvalue` function:

```
library(qvalue)
data(hedenfalk)
pvalues <- hedenfalk$p
qobj <- qvalue(p = pvalues)
```

Additionally, the `qvalue` object can be calculated given a set of empirical null statistics:

```
library(qvalue)
data(hedenfalk)
obs_stats <- hedenfalk$stat
null_stats <- hedenfalk$stat0
pvalues <- empPvals(stat = obs_stats, stat0 = null_stats)
qobj <- qvalue(p = pvalues)
```

Once the `qvalue` object is created, estimates of the q-values, the proportion of true null hypotheses π_0, and the local false discovery rates can be accessed from `qobj`:

```
qvalues <- qobj$qvalues
pi0 <- qobj$pi0
lfdr <- qobj$lfdr
```

The object can be summarized and visualized by:

```
summary(qobj)
hist(qobj)
plot(qobj)
```

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) [7]. The test-statistics and p-values from our analysis are included with the `qvalue` package:

```
data(hedenfalk)
names(hedenfalk)

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

A test of differential gene expression was performed between two types of genetic mutations 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) [4].[1]

The list `hedenfalk` has three variables: `p`, `stat`, `stat0`. The `p` variable is a vector of p-values from the tests of differential expression (one per gene for a total of 3170); `stat` is a vector also of length 3170 that contains the observed test-statistics calculated on the original data (the statistic is the absolute values of a two-sample t-test); and `stat0` is a 3170×100 matrix that contains the empirical "null statistics", which were generated by permuting the BRCA1 and BRCA2 group labels 100 times and recalculating the absolute t-statistics for each permutation.

5.1 Calculating p-values

One will typically already have a vector of p-values calculated using an appropriate method before utilizing the `qvalue` package. Sometimes users will instead have a vector of "observed statistics" that have been calculated on the original data, and then simulated or data-resampled (e.g., bootstrap, permutation) "null statistics". As long as the statistics are constructed such that the larger a statistic is, the more evidence there is against the null hypothesis in favor of the alternative hypothesis (e.g., the larger it is the "more extreme" it is), then there is a function in `qvalue` called `empPvals` that allows one to efficiently calculate p-values to be input into `qvalue`:

```
null_stats <- hedenfalk$stat0
obs_stats <- hedenfalk$stat
pvalues <- empPvals(stat = obs_stats, stat0 = null_stats,
    pool = FALSE)
```

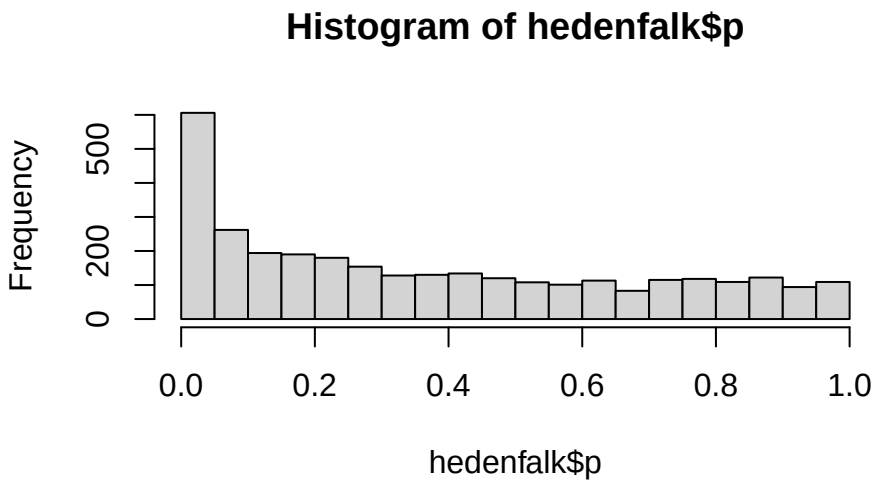
The documentation on `empPvals`, which can be accessed via `?empPvals`, explains how to use this function to calculate test-specific or test-nonspecific pooled p-values using this function.

5.2 Checking the p-value histogram

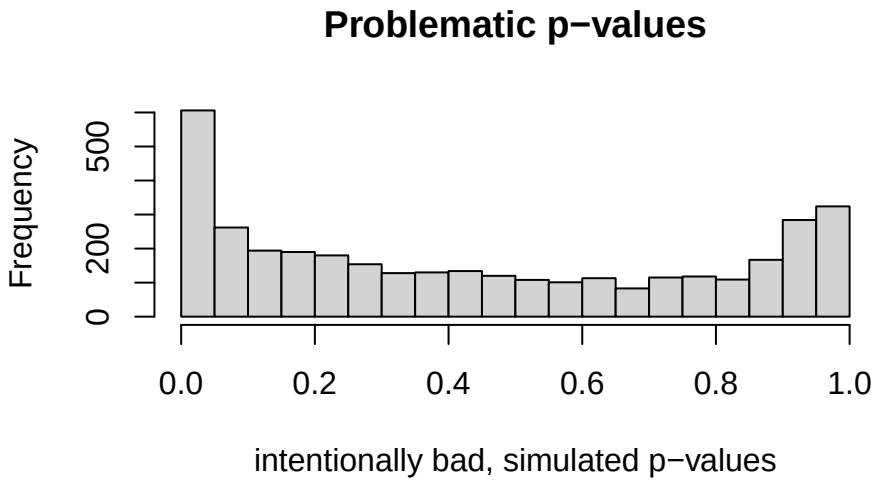
Before running `qvalue`, we strongly recommend that you view a histogram of the p-values:

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

```
hist(hedenfalk$p, nclass = 20)
```



The p-values are relatively flat at the right tail of the histogram. This is an important step in determining whether the true null p-values are distributed according to a Uniform(0,1) distribution. Suppose the p-value histogram instead looked like this simulated set of p-values:



The "U-shaped" p-value histogram is a red flag. An important assumption behind the estimation performed 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 tail is fairly flat as in the Hedenfalk et al. 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.

5.3 The qvalue function

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`:

- `p`: A vector of p-values. This is the only necessary input.
- `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) [2] for more information.
- `...`: Arguments passed to the functions `pi0est` and `lfdr`, which can include:
  - `lambda` (passed to `pi0est`): 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.
  - `pi0.method` (passed to `pi0est`): Either `"smoother"` or `"bootstrap"`; the method for automatically handling the tuning parameter in the estimation of π_0.
  - `trunc` (passed to `lfdr`): If `TRUE`, local FDR estimates > 1 are set to 1. Default is `TRUE`.

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) [4] is used. An alternative is the bootstrap method (`pi0.method = "bootstrap"`) proposed in Storey, Taylor & Siegmund (2004) [5].

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) [6] 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) [2] methodology.

5.3.1 The qvalue object

The object contains several relevant fields:

```
names(qobj)

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

- `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 estimated 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.

5.3.2 Summarizing results

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     29    85  167
##             <1
## p-value   3170
## q-value   3170
## local FDR 2239
```

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

5.3.3 The π_0 estimate

One very important statistic 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 is quite a useful number to know, even if all the truly significant tests cannot all be explicitly identified. If the π_0 estimate is the statistic of interest, the function `pi0est` can be used directly:

```
pi0 <- pi0est(p = hedenfalk$p, lambda = seq(0.1,
    0.9, 0.1), pi0.method = "smoother")
```

```
names(pi0)

## [1] "pi0"        "pi0.lambda" "lambda"
## [4] "pi0.smooth"
```

The `pi0` slot provides the overall π_0 estimate, `pi0.lambda` are the estimated proportion of null p-values at each `lambda` values, and `pi0.smooth` are the estimated proportion of null p-values at each `lambda` from the spline fit.

5.3.4 The q-values

The q-value is the minimum FDR incurred when calling a test significant. The q-values can be extracted from the `qvalue` object by:

```
qvalues <- qobj$qvalues
```

We advocate reporting the estimated q-value for each test. However, sometimes one wants to estimate the FDR incurred for a given p-value cut-off, or estimate the p-value cut-off required to control the FDR at a certain level. For example, if one wants to estimate the false discovery rate when calling all p-values ≤ 0.01 significant, then type:

```
max(qvalues[qobj$pvalues <= 0.01])

## [1] 0.07932935
```

This calculates the maximum estimated q-value among all p-values ≤ 0.01, which is equivalent to estimating the false discovery rate when calling all p-values ≤ 0.01 significant. When considering all p-values ≤ 1, the maximum q-value will be the estimate of π_0. This is because the best estimate of the false discovery rate, when considering all tests, will be the estimate of the rejection region.

If one wants to control the false discovery rate at a pre-determined level α, then calling all tests significant with estimated q-values $\leq \alpha$ accomplishes this under certain mathematical assumptions, including some cases where the p-values are dependent ([5]). If `fdr.level` is set to α in `qvalue`, then a vector indicating whether each q-value is $\leq \alpha$ can be obtained by:

```
qobj_fdrlevel <- qvalue(p = hedenfalk$p, fdr.level = 0.1)
qobj$significant
```

The more likely case is that one will want to investigate the overall behavior of the estimated q-values before making such a decision.

5.3.5 The local false discovery rates

The local FDR is a useful counterpart to the q-values ([1]). The estimated local FDR of a given test is an empirical Bayesian posterior probability that the null hypothesis is true, conditional on the observed p-value. To extract the local FDR estimates, type:

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

The function `lfdr` can be used to calculate the local FDR estimates directly:

```
localFDR <- lfdr(p = hedenfalk$p)
```

When calling `lfdr` directly, the user can provide a `pi0` value if desired. If no `pi0` value is given, then `pi0est`

is called, and arguments can be passed to `pi0est` via the `...` argument in `lfdr`. Type `?lfdr` for further details on the various inputs for the `lfdr` function.

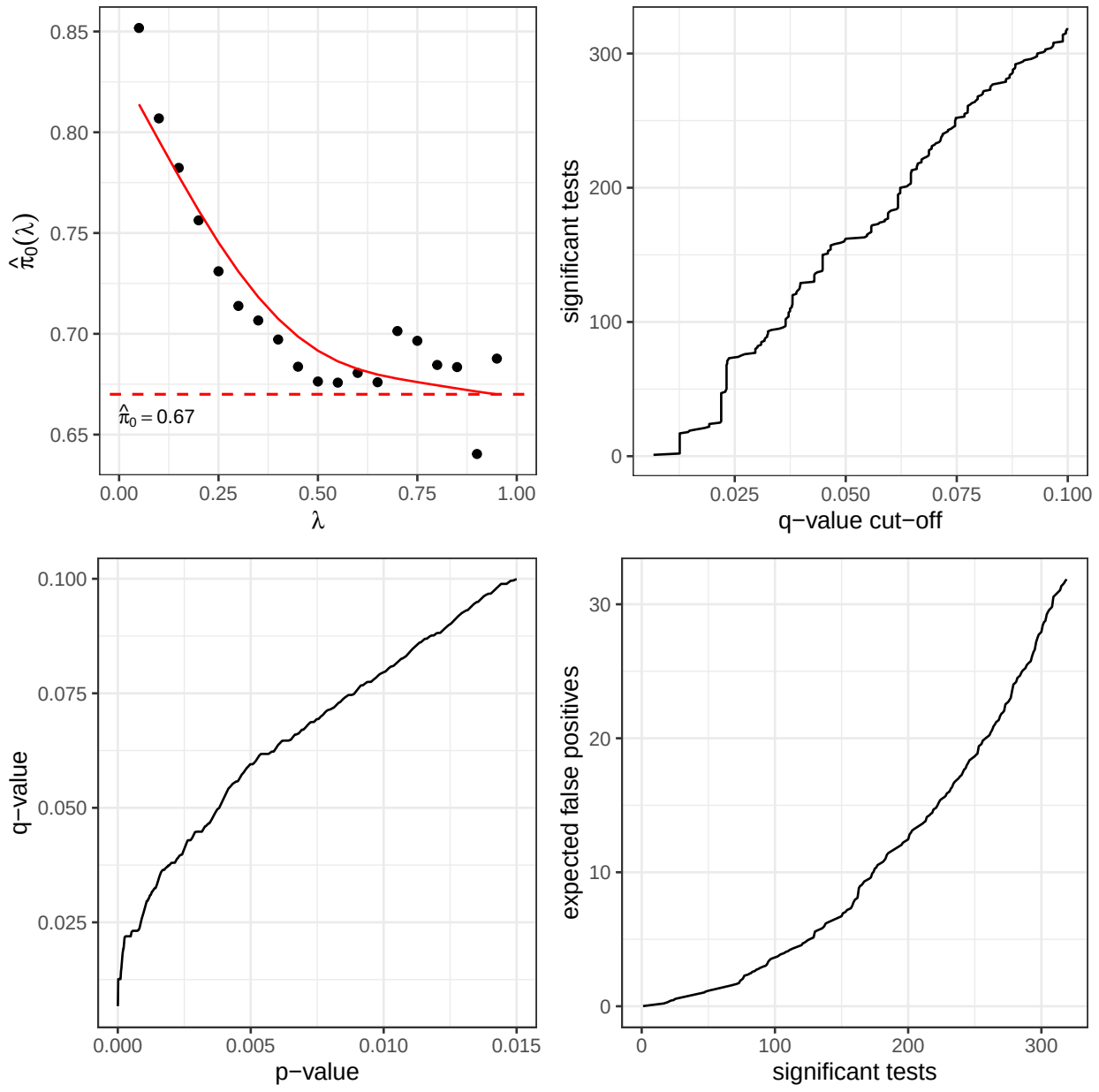
5.4 Visualizing results

The `hist` and `plot` functions can be used to visualize the results from `qvalue`. 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)
```

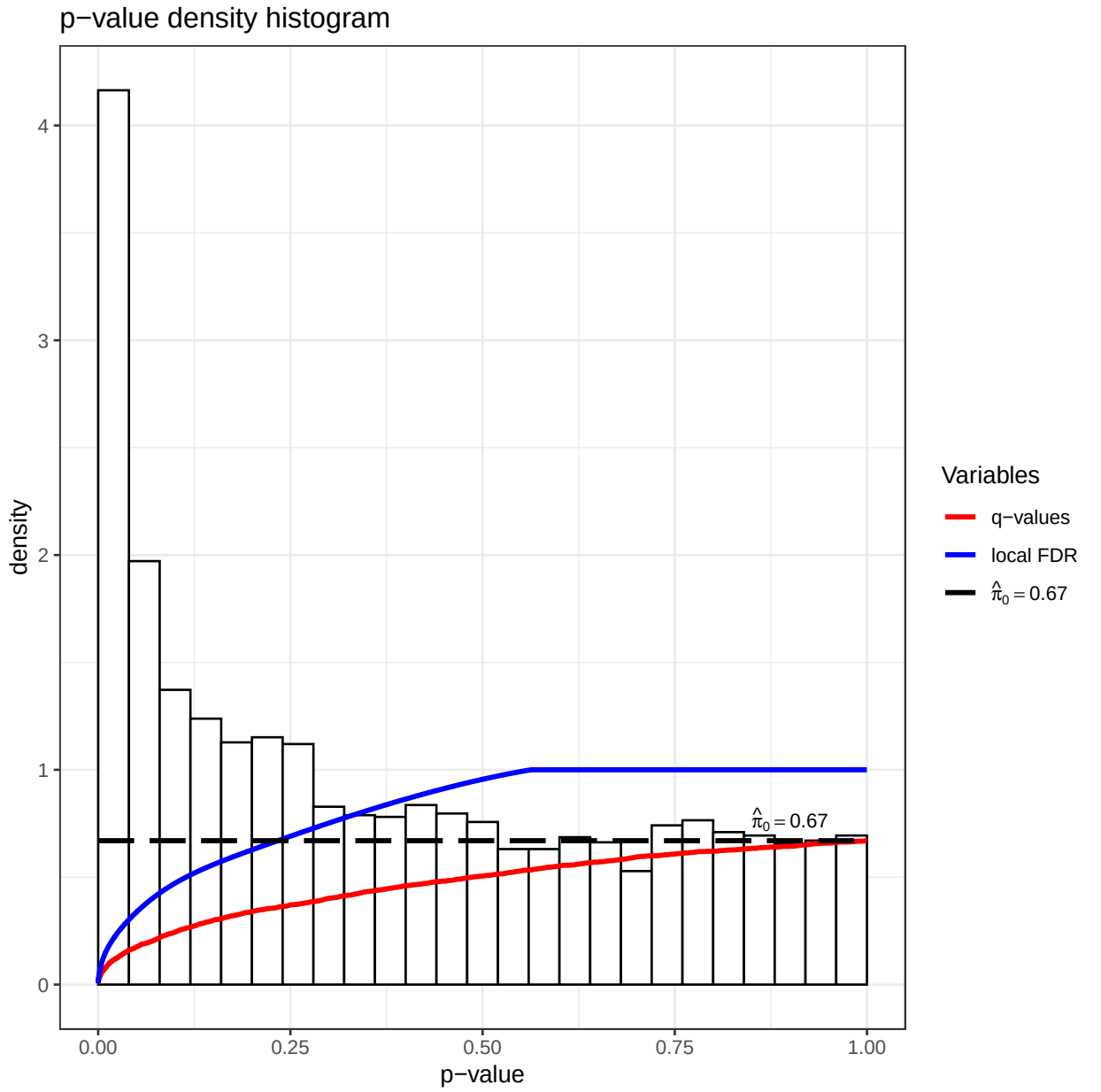


The main purpose of the upper-left plot is to gauge the reliability of the π_0 estimate, where the estimated π_0 is plotted versus the tuning parameter λ. The variable λ is called `lambda` in the package; it can be fixed or automatically handled. As λ gets larger, the bias of the estimate decreases, yet the variance increases. When `pi0.method = "smoother"` is utilized, the fitted smoother is also shown in this plot. See Storey (2002) [2] for more on λ and its role in estimating π_0. Comparing your final estimate of π_0 to this plot gives a good sense as to its quality.

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)
```



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?

The q-value is not an adjusted p-value, but rather a population quantity with an explicit definition (see [1, 2, 3]). 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. 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.

8 Obtaining updates on `qvalue`

The easiest way to obtain the most recent (development) version of `qvalue` is to visit https://github.com/jdstorey/qvalue/. Bug fixes will appear on GitHub earlier than on Bioconductor, and suggested changes can also be made there.

Acknowledgements

This software development has been supported in part by funding from the National Institutes of Health.

References

[1] J. D. Storey. False discovery rates. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*. Springer, 2011. http://genomine.org/papers/Storey_FDR_2011.pdf.

[2] J. D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B*, 64:479–498, 2002.

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

[4] J. D. Storey and R. Tibshirani. Statistical significance for genome-wide experiments. *Proceedings of the National Academy of Sciences*, 100:9440–9445, 2003.

[5] J. D. Storey, J. E. Taylor, and D. Siegmund. Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B*, 66:187–205, 2004.

[6] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 85:289–300, 1995.

[7] I. Hedenfalk, D. Duggan, Y.D. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O.P. Kallioniemi, B. Wilfond, A. Borg, and J. Trent. Gene expression profiles in hereditary breast cancer. *New England Journal of Medicine*, 344:539–548, 2001.