QVALUE: The Manual
Version 1.1

John D. Storey
Department of Biostatistics
Department of Genome Sciences
University of Washington
jstorey@u.washington.edu

March 2003; Revised June 2003

Table of Contents

1. Introduction

This document provides instructions for how to use the q-value functions I have made available in the R software package, as well as a short tutorial on false discovery rates and q-values. If you are unfamiliar with false discovery rates and q-values, then read Section 5 on page 6 first.

2. Citations

The basic methods used in the software come from the following article; please cite this article when reporting results based on the software:

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

If you are applying this methodology in genomics, then it may also be useful to cite:

> Storey JD and Tibshirani R. (2003) Statistical significance for genome-wide experiments. *Proceeding of the National Academy of Sciences*, in press.

If you discuss the strong control, conservative point estimation, or the simultaneous conservative consistency of the methods, then it may also be useful to cite:

> Storey JD, Taylor JE, and Siegmund D. (2002) Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B*, in press.

Several other references are useful to keep in mind. Benjamini & Hochberg (1995) proposed the false discovery rate and provided a step-wise p-value method to control it. Storey (2001) defined the q-value, so the original q-value definition and the study of its properties come from this paper.

3. How to Use QVALUE in R

Step 1. Download R from `http://cran.r-project.org/` and install it. R is a freely available statistical package written by Ihaka & Gentleman (1996) that closely parallels the commercial software S-plus. It’s a great bioinformatics and data analysis tool!

Step 2. Go to `http://faculty.washington.edu/~jstorey/qvalue/` and download the file `qvalue.R`. If you have never saved a ‘.R’ file before then do the following: under *Save as type* select *Default* or *All files* (whichever choice is available), then simply save the file as `qvalue.R` in your favorite directory.

Step 3. Save your p-values into a text file, preferably with one p-value per line. In this manual, we will call this file `pvalues.txt`. This can be done in Excel, for example, by creating a worksheet with the p-values listed in a single column. Then save the file as a tab-delimited text file.

Step 4. Start R. Select `File → Source R Code`. Select the file `qvalue.R` that you downloaded. This will load all the q-value functions into R.

Step 5. Select `File → Change directory`. Select the directory where you have stored `pvalues.txt`. Now type the following commands:

```
> p <- scan(“pvalues.txt”)
> qobj <- qvalue(p)
> qplot(qobj)
> qwrite(qobj, filename=“myresults.txt”)
```

The first line saves the p-values into the R object `p`. The second command saves the output from the main q-value function into the R object `qobj`. The third command makes four useful plots that can be used to assess which significance cut-offs make sense for your study. These plots are labeled and self-explanatory, and they are also discussed in the next section. The fourth command writes the results to a file called `myresults.txt`, which will be written in the same directory as `pvalues.txt`. The file contains the function call used and the estimate of π_0, where π_0 is the *overall* proportion of true null hypotheses. (The false discovery rate is the proportion of true null hypotheses among those called significant, and π_0 is the proportion of true null hypotheses among all tests. See Section 5 on page 6.) Starting on the third line, the file lists each p-value and corresponding estimated q-value, one per line in the same order as `pvalues.txt`.

Several other arguments can be used in the function `qvalue`. The following lists all the possible arguments, with a description of each:

- `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) for more information.
- `pi0.meth`: 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. Optional; the choice `"smoother"` is the default choice.
- `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 that specifies whether each q-value is less than `fdr.level` or not.
- `robust`: An indicator of whether it is desired to make the estimate more robust for small p-values. This uses the point estimate of the “positive false discovery rate” (pFDR). Optional; see Storey (2002) for more information.

The most delicate aspect of this software is choosing how to estimate π_0 via `lambda` and `pi0.meth`. If no options are selected, then by default the smoother method (`pi0.meth=〝smoother″`) proposed in Storey and Tibshirani (2003) is used. My experience indicates that this often works better than the bootstrap method, but can backfire for a small number of p-values or in pathological situations. An overall safer option is the bootstrap method (`pi0.meth=〝bootstrap″`) proposed in Storey, Taylor & Siegmund (2002). If one selects `lambda=0`, then this produces the estimate implicit in the Benjamini and Hochberg (1995) methodology. In particular, setting `lambda=0` estimates π_0 to be 1. This can be viewed as a special conservative case of the Storey (2002) methodology, so at the very least I recommend using `pi0.meth=〝bootstrap″` rather than setting `lambda` to some predetermined number, such as `lambda=0`. Here are three examples of using `qvalue` with non-default options:

```
> qobj <- qvalue(p, lambda=seq(0.2,0.8,0.01), robust=TRUE)
> qobj <- qvalue(p, lambda=0, fdr.level=0.05)
> qobj <- qvalue(p, pi0.meth=″bootstrap″)
```

The function `qplot` has an option to change the range of q-values for which the plots can be made. If one wants to view a range of q-values in, say 0 to 0.3, then type:

```
> qplot(qobj, rng=0.3)
```

The function `qwrite` currently has no options other than designating the file name, as was done above.

I advocate reporting the estimated q-value for each test. However, sometimes one wants to estimate the false discovery rate incurred for a given p-value cut-off, or estimate the p-value cut-off to control the false discovery rate at a certain level. Below are instructions on how to do this in `QVALUE`.

Estimating the false discovery rate for a given p-value cut-off. If one wants to estimate the false discovery rate when calling all p-values less than or equal to 0.01 significant, then type:

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

This calculates the maximum estimated q-value among all p-values less than or equal to 0.01, which is equivalent to estimating the false discovery rate when calling all p-values less than or equal to 0.01 significant. Clearly, if a cut-point different than 0.01 is desired, then replace 0.01 in the above command with that number.

Estimating a p-value cut-off for a given false discovery rate level. If one wants to estimate the p-value cut-off for controlling the false discovery rate at level 0.05, then type:

```
> max(qobj$pvalues[qobj$qvalues <= 0.05])
```

This calculates the largest p-value with estimated q-value less than or equal to 0.05. If we set `lambda=0`, then this is equivalent to the Benjamini & Hochberg (1995) step-wise p-value method. If π_0 is estimated (rather than set to 1), then this is equivalent to the false discovery rate controlling procedure proposed in Storey, Taylor & Siegmund (2002). Clearly, if a level of false discovery rate control other than 0.05 is desired, then replace 0.05 in the above command with the desired number.

4. How to Use the Q-values to Make Decisions

Here, we give some concise guidelines for interpreting the output of the software. A more thorough discussion in the context of genomics can be found in Storey & Tibshirani (2003).

One very important number that is obtained with the software is an estimate of the overall proportion of true null hypotheses π_0. This estimate can be accessed by:

```
> qobj$pi0
```

Clearly, an estimate of *the proportion of significant 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. The p-values and q-values can also be respectively listed by the following commands:

```
> qobj$pvalues
> qobj$qvalues
```

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 (Storey, Taylor & Siegmund 2002). In other words, we guarantee in some sense that

$$\frac{\#\text{ of false positives}}{\#\text{ of } \textit{significan t} \text{ tests}} \leq \alpha$$

by calling all tests significant with q-values $\leq \alpha$. This command is available in `qvalue`: if `fdr.level` is set to α in `qvalue`, then a vector indicating whether each q-value is less than or equal to α can be obtained by:

```
> 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. The function `qplot` allows one to view several useful plots:

1. The estimated π_0 versus the tuning parameter λ
2. The q-values versus the p-values
3. The number of significant tests versus each q-value cut-off
4. The number of expected false positives versus the number of significant tests

The main purpose of the first plot is to gauge how reliable the estimate of π_0 is. Basically, a tuning parameter λ has to be chosen to estimate π_0. The variable λ is called `lambda` in the software; as stated above, 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) 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. A thorough discussion of these plots can be found in Storey & Tibshirani (2003).

Finally, note that the most informative approach is to report the estimated q-value with each test, rather than making potentially arbitrary decisions about cut-offs for significance.

6. What is a Q-value? (A primer)

The q-value is similar to the well known p-value. It gives each hypothesis test a measure of significance in terms of a certain error rate. The p-value of a test measures the minimum *false positive rate* that is incurred when calling that test significant. Likewise, the q-value of a test measures the minimum *false discovery rate* that is incurred when calling that test significant.

Whereas the p-value is commonly used for performing a single significance test, the q-value is useful for assigning a measure of significance to each of *many* tests performed *simultaneously*. (An example is testing thousands of genes for differential expression using DNA microarray data.) For each of these tests, there is a *null hypothesis* tested against an *alternative hypothesis*. A measure of significance therefore roughly measures how much a single test deviates from the null. The false positive rate and false discovery rate accomplish this quite differently.

A *false positive* is the term used to describe rejecting the null hypothesis (i.e., calling the test significant) when it is really true. Suppose we have defined a rule for calling tests significant. The false positive rate of the rule can then be loosely described by:

$$\text{false positive rate} \approx \frac{\text{\# of false positives}}{\text{\# of } \textit{true null} \text{ tests}}.$$

Therefore, the false positive rate measures the proportion of true null hypotheses that were (incorrectly) called significant by this rule.

A *false discovery* is also a false positive, however, the different terminology stresses the fact that we are concerned with false positives among the significant tests (i.e., the discoveries). The false discovery rate is the expected proportion of false positives among the tests found to be significant. The false discovery rate can then be loosely described by:

$$\text{false discovery rate} \approx \frac{\text{\# of false positives}}{\text{\# of } \textit{significan t} \text{ tests}}.$$

The false positive rate and the false discovery rate therefore tell us two very different things about a method for calling tests significant. For a single test, the false positive rate can be useful for measuring how likely it is for a truly null case to be as significant as what has been observed. However, for many tests this is not as useful. For example, suppose we decide that we can live with a false positive rate of 5%. Then about 5% of the time, we will call a truly null hypothesis significant. If we perform 1000 tests at a 5% false positive rate, then we can expect up to 50 false positives. This will typically be too many in practical situations.

When performing many significance tests, the false discovery rate gives more useful information. If we are willing to incur a false discovery rate of 5%, then this means that among all tests we call significant, about 5% of them will be false positives. If there are

100 significant tests, then this results in about 5 false positives; 500 significant tests results in about 25 false positives, etc.

If all tests are called significant then the false positive rate = 1 since all tests are called significant, and therefore all true null hypotheses are called significant. On the other hand, the false discovery rate is

$$\pi_0 \equiv \frac{\#\text{ of } \textit{true null} \text{ tests}}{\#\text{ of total tests}}$$

when all tests are called significant. The quantity π_0 is the overall proportion of true null hypotheses in the study. This is a useful number to consider as well as $\pi_1 \equiv 1 - \pi_0$, which is the proportion of significant results in the study. An estimate π_0 of is provided in the software.

In most significance testing situations, the null hypothesis is defined in such a way that either the null distribution of the test statistic is known (e.g., the null distribution of a t-test is the *t* distribution when the data are normal) or the null distribution can be simulated (e.g., via permutations or the bootstrap). Regardless of the method used, the false positive rate is easily measured, making it straightforward to obtain p-values. The false discovery rate, on the other hand, involves information about the false null hypotheses. Therefore to make a precise false discovery rate calculation, we would have to know which tests are truly significant and what their alternative distributions are.

False discovery rates methods can be described without loss of generality in terms of p-value. Specifically, Storey (2001) has shown that the q-value is the same whether we estimate it from the original statistics or from their corresponding p-values. Therefore, the software available here calculates q-values based on p-values. Because of the ease at which p-values can be obtained, this is a useful way to make the methods widely available.

Storey (2002) has developed methods for estimating false discovery rates that can be applied in a variety of ways. Rather than using only information from the null distribution, it utilizes information from all the p-values at once. For a given p-value threshold, say 5%, the false discovery rate is estimated in such a way that on average this estimate will exceed the true false discovery rate. This is a good property – we don't want to report a smaller false discovery rate than truly exists. Recently, it has been shown that this same estimate can be used to pick a false discovery rate beforehand, say 1%, and find the p-value threshold that guarantees on average that the true false discovery rate will be less than or equal to the desired level (Storey, Taylor & Siegmund 2002). This is also a desirable property.

Fixing a significance threshold beforehand or fixing the false discovery rate beforehand may be useful in some situations. What is most general and useful however, is a test-specific false discovery rate measure. This essentially allows us to look at all possible thresholds at once, as well as providing each test with a measure of significance that can be easily interpreted. *This is exactly what the q-value accomplishes*. For a given test, we estimate the q-value by calculating the minimum estimated false discovery rate among all thresholds at which the false discovery rate is called significant. Conditions under which the q-values are simultaneously conservative have been given in Storey, Taylor & Siegmund (2002). If this property holds, then one can consider all q-values simultaneously without worrying about incurring bias. A neat Bayesian posterior probability view of q-values has been shown in Storey (2001), which gives the origin of its name.

See my recent talk at `http://faculty.washington.edu/~jstorey/qvalue/talk.pdf` for another short introduction to false discovery rates and q-values, including a brief summary of the formulas used in this software.

7. References

Benjamini Y and Hochberg Y. (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing.

Ihaka R and Gentleman R. (1996) R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5**: 299-314.

Storey JD. (2001) The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics*, in press.

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

Storey JD, Taylor JE, and Siegmund D. (2002) Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B*, in press.

Storey JD and Tibshirani R. (2003) Statistical significance for genome-wide experiments. *Proceeding of the National Academy of Sciences*, in press.