The Stata Journal (2010)
10, Number 4, pp. 568–584

Frequentist q-values for multiple-test procedures

Roger B. Newson
National Heart and Lung Institute
Imperial College London
London, UK
r.newson@imperial.ac.uk

Abstract. Multiple-test procedures are increasingly important as technology increases scientists' ability to make large numbers of multiple measurements, as they do in genome scans. Multiple-test procedures were originally defined to input a vector of input p-values and an uncorrected critical p-value, interpreted as a familywise error rate or a false discovery rate, and to output a corrected critical p-value and a discovery set, defined as the subset of input p-values that are at or below the corrected critical p-value. A range of multiple-test procedures is implemented using the `smileplot` package in Stata (Newson and the ALSPAC Study Team 2003, *Stata Journal* 3: 109–132; 2010, *Stata Journal* 10: 691–692). The `qqvalue` command uses an alternative formulation of multiple-test procedures, which is also used by the R function `p.adjust`. `qqvalue` inputs a variable of p-values and outputs a variable of q-values that are equal in each observation to the minimum familywise error rate or false discovery rate that would result in the inclusion of the corresponding p-value in the discovery set if the specified multiple-test procedure was applied to the full set of input p-values. Formulas and examples are presented.

Keywords: st0209, qqvalue, smileplot, multproc, p.adjust, R, multiple-test procedure, data mining, familywise error rate, false discovery rate, Bonferroni, Šidák, Holm, Holland, Copenhaver, Hochberg, Simes, Benjamini, Yekutieli

1 Introduction

Multiple-test procedures are one of the key themes in twenty-first-century biostatistics so far because technology gives scientists the power to measure unprecedented numbers of comparisons in genome scans, epigenome scans, and metabolome scans. A multiple-test procedure takes the following as input: a vector of p-values that corresponds to multiple comparisons testing multiple null hypotheses, and an uncorrected critical p-value, which is usually interpreted either as a maximum permissible familywise error rate (FWER) or as a maximum permissible false discovery rate (FDR). The multiple-test procedure outputs a corrected critical p-value that is used to define a discovery set as the subset of input p-values at or below the corrected critical p-value. A number of multiple-test procedures have been implemented in Stata using the `smileplot` package (Newson and the ALSPAC Study Team 2003, 2010).

Frequentist multiple-test procedures are a generalization of the concept of confidence regions beyond scalar and even vector parameters to a set-valued parameter, namely, the set of null hypotheses that are true. If the input uncorrected critical p-value $\alpha \in (0, 1)$ is an FWER, then we can be $100(1 - \alpha)\%$ confident that all the null hypotheses in the discovery set are false. If the input uncorrected critical p-value $\alpha = \beta \times \gamma$ is an FDR, then we can be $100(1 - \beta)\%$ confident that over $100(1 - \gamma)\%$ of the null hypotheses in the discovery set are false. Of course, the discovery set may be empty, in which case 100% of the null hypotheses in it are false.

Conventionally, a multiple-test procedure has been implemented by writing a program that inputs a vector of p-values and an uncorrected critical p-value and outputs a corrected critical p-value and a discovery set. The `multproc` command of the `smileplot` package introduced by Newson and the ALSPAC Study Team (2003) does just that.

The R function `p.adjust` (Smyth and the R Core Team 2010) uses an alternative way of implementing multiple-test procedures. This function inputs a vector of p-values and a specified multiple-test procedure. It outputs a new vector of q-values (parallel to the input vector), sometimes known as adjusted p-values. For each input p-value, the corresponding q-value is the lowest input uncorrected critical p-value (FWER or FDR) that would cause the input p-value to be included in the discovery set if the specified multiple-test procedure was applied to the full vector of p-values. This q-value may be one if there is no FWER or no FDR less than one for which the corresponding null hypothesis would be rejected.

The Stata `qqvalue` package is modeled broadly on the R function `p.adjust`; it generates q-values for an input variable of p-values and a specified multiple-test procedure. The name `qqvalue` originally stood for "quasi–q-value", which was my initial choice of terminology and was intended to prevent confusion between the vector of adjusted p-values output by `p.adjust` and the scalar corrected critical p-value output by the `multproc` command of `smileplot`. The term q-value was originally introduced as an empirical Bayesian concept by Storey (2003), who aimed to control the positive FDR by estimating from the vector of input p-values the prior probability that a null hypothesis is true. The q-values calculated by `p.adjust` and `qqvalue`, by contrast, are the nearest frequentist equivalent of Storey's q-values. They are minimum FWERs or FDRs for rejection of individual input p-values, just as Storey's original q-values are minimum positive FDRs for rejection of individual input p-values. In view of this difference, I originally added the prefix "quasi–", but was advised by Gordon Smyth (the author of `p.adjust`) that the prefix was not really necessary because it is now common to use the term q-value for the values computed by `p.adjust`. I therefore now conform to this usage but use the term "frequentist q-value" when making a distinction from the original Bayesian q-value.

The remainder of this article documents and details the `qqvalue` package. Section 2 documents the command itself. Section 3 presents and details the methods and formulas used. Section 4 gives some examples of the use of `qqvalue` in practice.

2 The qqvalue command

2.1 Syntax

`qqvalue` *varname* [*if*] [*in*] [`,` `method(`*method*`)` `bestof(`#`)` `qvalue(`*newvar*`)` `npvalue(`*newvar*`)` `rank(`*newvar*`)` `svalue(`*newvar*`)` `rvalue(`*newvar*`)` `float` `fast`]

where *method* is one of
`bonferroni` | `sidak` | `holm` | `holland` | `hochberg` | `simes` | `yekutieli`

`by` *varlist*`:` can be used with `qqvalue`; see [D] **by**. If `by` *varlist*`:` is used, then all generated variables are calculated using the specified multiple-test procedure within each by-group defined by the variables in the *varlist*.

2.2 Description

`qqvalue` is similar to the R package `p.adjust`. It inputs a single variable, assumed to contain p-values calculated for multiple comparisons, in a dataset with one observation per comparison. It outputs a new variable—calculated by inverting a multiple-test procedure specified by the user—containing the q-values corresponding to these p-values. Each q-value represents, for each corresponding p-value, the minimum uncorrected p-value threshold for which that p-value would be in the discovery set, assuming that the specified multiple-test procedure was used on the same set of input p-values to generate a corrected p-value threshold. These minimum uncorrected p-value thresholds may represent FWERs or FDRs, depending on the procedure used. `qqvalue`'s options may be used to output other variables that contain the various intermediate results used in calculating the q-values. The multiple-test procedures available for `qqvalue` are a subset of those available using the `multproc` command of the `smileplot` package (Newson and the ALSPAC Study Team 2010).

2.3 Options

`method(`*method*`)` specifies the multiple-test procedure method to be used for calculating the q-values from the input p-values. The *method* may be `bonferroni`, `sidak`, `holm`, `holland`, `hochberg`, `simes`, or `yekutieli`. These method names specify that the q-values will be calculated from the input p-values by inverting the multiple-test procedure specified by the `method()` option of the same name for the `multproc` command of the `smileplot` package (Newson and the ALSPAC Study Team 2010). The default is `method(bonferroni)`.

`bestof(`#`)` specifies an integer. If the `bestof()` option is specified and # is greater than the number of input p-values, then the q-values are calculated assuming that the input p-values are a subset (usually the smallest number of input p-values) of

a superset of p-values. If the `method()` option specifies a one-step method (such as `bonferroni` or `sidak`), then the q-values do not depend on the other p-values in the superset, but only on the number of p-values in the superset. If the `method()` option specifies a step-down method (such as `holm` or `holland`), then it is assumed that all the other p-values in the superset are greater than the largest of the input p-values. If the `method()` option specifies a step-up method (such as `hochberg`, `simes`, or `yekutieli`), then it is assumed that all the other p-values in the superset are equal to one, which implies that the q-values will be conservative and will define an upper bound to the respective q-values that would have been calculated if we knew the other p-values in the superset. If `bestof()` is unspecified (or nonpositive), then the input p-values are assumed to be the full set of p-values calculated. The `bestof()` option is useful if the input p-values are known (or suspected) to be the smallest of a greater set of p-values that we do not know. This often happens if the input p-values are from a genome scan reported in the literature.

`qvalue(`*newvar*`)` specifies the name of a new output variable containing the q-values calculated from the input p-values. The new output variable is generated using the multiple-test procedure specified by the `method()` option.

`npvalue(`*newvar*`)` specifies the name of a new output variable to be generated. It contains in each observation the total number of p-values in the sample of observations specified by the `if` and `in` qualifiers or in the by-group containing that observation if the `by:` prefix is specified.

`rank(`*newvar*`)` is the name of a new variable to be generated. It contains in each observation the rank of the corresponding p-value from the lowest to the highest. Tied p-values are ranked according to their position in the input dataset. If the `by:` prefix is specified, then the ranks are defined within the by-group.

`svalue(`*newvar*`)` specifies the name of a new output variable to be generated, which contains the s-values calculated from the input p-values. The s-values are an intermediate result; they are calculated in the course of calculating the q-values and are used mainly for validation. They are calculated from the input p-values by inverting the formulas used for the rank-specific critical p-value thresholds, which are calculated by the `multproc` command of the `smileplot` package. These rank-specific p-value thresholds are returned in the generated variable specified by the `critical()` option of `multproc`. The s-values may be greater than one.

`rvalue(`*newvar*`)` specifies the name of a new output variable to be generated, which contains the r-values calculated from the input p-values. The r-values are an intermediate result; they are calculated in the course of calculating the q-values and are used mainly for validation. They are calculated from the s-values by truncating the s-values to a maximum of one. The q-values are calculated from the r-values using a procedure that is dependent on the multiple-test procedure specified by the `method()` option. If the multiple-test procedure is a one-step procedure (such as `bonferroni` or `sidak`), then the q-values are equal to the corresponding r-values. If the multiple-test procedure is a step-down procedure (such as `holm` or `holland`), then the q-value for each p-value is equal to the cumulative maximum of all the

r-values corresponding to *p*-values of rank equal to or less than that *p*-value. If the multiple-test procedure is a step-up procedure (such as `hochberg`, `simes`, or `yekutieli`), then the *q*-value for each *p*-value is equal to the cumulative minimum of all the *r*-values corresponding to *p*-values of rank equal to or greater than that *p*-value.

`float` specifies that the output variables specified by the `qvalue()`, `rvalue()`, and `svalue()` options be created as variables of type `float`. If `float` is absent, then these variables are created as variables of type `double`. Whether or not `float` is specified, all generated variables are stored to the lowest precision possible without loss of information.

`fast` is an option for programmers. It specifies that `qqvalue` will not take any action to restore the original data in the event of failure or if the user presses **Break**.

3 Methods and formulas

The methods used are a development of those used by the `multproc` command of the `smileplot` package, which is documented in Newson and the ALSPAC Study Team (2003, 2010). I will therefore use a notation that is as consistent as possible with that source. I will use uppercase and lowercase symbols to denote different quantities and to reduce confusion in readers who refer both to that article and to this article.

We assume that there is a sequence of m distinct parameters $\theta_1, \ldots, \theta_m$, estimated using estimates $\widehat{\theta}_1, \ldots, \widehat{\theta}_m$ and having the values $\theta_1^{(0)}, \ldots, \theta_m^{(0)}$ under their respective null hypotheses. Typically, $\theta_i^{(0)}$ is zero for difference parameters such as median differences or is one for ratio parameters such as median ratios. We denote by $P_1, \ldots, P_m$ the observed *p*-values for testing the m null hypotheses. Each P_i has the property that if $0 \leq \alpha \leq 1$, then

$$\Pr\left(P_i \leq \alpha \;\middle|\; \theta_i = \theta_i^{(0)} \right) \leq \alpha$$

We denote by $R_1, \ldots, R_m$ the ranks (in ascending order) of $P_1, \ldots, P_m$ and denote by $Q_1, \ldots, Q_m$ the *p*-values in ascending order so that for each i, $Q_{R_i} = P_i$. (The Q_i are not the *q*-values, which we will define in due course.)

The methods used by the `multproc` command of the `smileplot` package aim to define a credible (or acceptable) subset of indices $C \subseteq (1, \ldots, m)$ such that the null hypotheses $(\theta_i = \theta_i^{(0)} : i \in C)$ are acceptable and the complementary set of null hypotheses $(\theta_i = \theta_i^{(0)} : i \notin C)$ are rejected. This is done by defining an uncorrected *p*-value threshold, p_{unc}; calculating a corrected *p*-value threshold, p_{cor}, from p_{unc} and $Q_1, \ldots, Q_m$; and defining the acceptable subset C to be the subset of indices i such that $P_i > p_{\text{cor}}$. The methods used by `qqvalue`, by contrast, are derived by inverting the methods used by `multproc` because they start from an individual input *p*-value and derive the minimum uncorrected *p*-value threshold, which if used would have made the corrected *p*-value threshold at least as large as the individual input *p*-value.

The multiple-test procedures used by `qqvalue` and selected using the `method()` option are a subset of those used by `multproc`. They are listed in table 1 and classified in three ways: the form of the algorithm used (one-step, step-down, or step-up), the interpretation of the uncorrected overall critical p-value (FWER or FDR), and the correlation assumed between the P_i (independence, nonnegative, or arbitrary).

Table 1. Multiple-test procedures specified by the `method()` option of `qqvalue`

`method()`	Step type	FWER/FDR	Correlation assumed
`bonferroni`	one-step	FWER	arbitrary
`sidak`	one-step	FWER	nonnegative
`holm`	step-down	FWER	arbitrary
`holland`	step-down	FWER	nonnegative
`hochberg`	step-up	FWER	independence
`simes`	step-up	FDR	nonnegative
`yekutieli`	step-up	FDR	arbitrary

3.1 Formulas for one-step, step-down, and step-up methods

The formulas used by `multproc` are given in Newson and the ALSPAC Study Team (2003, section 3.1). Each of the methods of `multproc` works by specifying a nondecreasing sequence of individual critical p-values $c_1, \ldots, c_m$, which correspond to the ordered input p-values $Q_1, \ldots, Q_m$. The formulas used by each method for deriving these thresholds c_i as functions of p_{unc}, i, and m are listed in that subsection.

Once these c_i are specified, each `multproc` method selects an overall corrected critical p-value, p_{cor}, from the c_i in one of three ways, namely, one-step, step-down, or step-up. In the one-step case, the c_i are all equal to a common value, p_{cor}, defined in a way that is not dependent on i. In the step-down case, p_{cor} is set to the minimum c_i such that $Q_i > c_i$ if such a c_i exists or to the maximum critical p-value c_m otherwise. In the step-up case, p_{cor} is set to the maximum c_i such that $Q_i \leq c_i$ if such a c_i exists or to the minimum critical p-value c_1 otherwise.

The q-values computed by `qqvalue` are derived by inverting the formulas of `multproc`. The technique can be summarized in the phrase "sorted p-values generate s-values generate r-values generate q-values". For each given method, this technique is executed in three steps:

1. Invert the formula used for calculating c_i as a function of p_{unc} to give a formula for calculating p_{unc} as a function of c_i. If we substitute the sorted p-value Q_i for c_i in this formula, then the result will be denoted s_i. s_i will be expressed on an uncorrected p-value scale but may be one or greater if no FWER or FDR less than one will generate a threshold $c_i \geq Q_i$.

2. Define $r_i = \min(s_i, 1)$ as the minimum uncorrected critical p-value that generates a threshold that Q_i can pass below. If we are willing to live with a FWER or FDR of 1, at which 100% of discoveries may be false, then any p-value may be included in the discovery set.

3. Define the set of q-values q_i from the set of r-values r_i, using a formula that depends on whether the procedure is one-step, step-down, or step-up. For a one-step procedure, this formula is

$$q_i = r_i \tag{1}$$

For a step-down procedure, it is

$$q_i = \max(r_j : j \leq i) \tag{2}$$

For a step-up procedure, it is

$$q_i = \min(r_j : j \geq i) \tag{3}$$

For each i, q_i will then be the q-value corresponding to the sorted p-value Q_i. Therefore, for each i, the q-value corresponding to P_i will be q_{R_i}.

The formulas for deriving the s_i from the Q_i are derived by inverting a subset of those in Newson and the ALSPAC Study Team (2003, section 3.1). They are given as follows, together with references for the original multiple-test procedures:

One-step methods

1. `bonferroni`

$$s_i = mQ_i$$

2. `sidak` (Šidák 1967)

$$s_i = 1 - (1 - Q_i)^m$$

Step-down methods

1. `holm` (Holm 1979)

$$s_i = (m - i + 1)Q_i$$

2. `holland` (Holland and Copenhaver 1987)

$$s_i = 1 - (1 - Q_i)^{m-i+1}$$

Step-up methods

1. `hochberg` (Hochberg 1988)

$$s_i = (m - i + 1)Q_i$$

The s_i are the same as those for the step-down Holm method.

2. `simes` (Simes [1986]; Benjamini and Hochberg [1995]; Benjamini and Yekutieli [2001, first method])
$$s_i = \frac{m}{i} Q_i$$
3. `yekutieli` (Benjamini and Yekutieli [2001, second method])
$$s_i = \frac{m}{i} Q_i \sum_{j=1}^{m} j^{-1}$$

All these expressions for s_i are increasing in Q_i and increasing in m and nonincreasing (or constant in the case of one-step procedures) in i. The corresponding expressions for $r_i = \min(s_i, 1)$ will therefore be nondecreasing in Q_i and in m, and will be nonincreasing in i.

3.2 Incomplete sets of input p-values

We have assumed so far that the variable input to `qqplot` contains the full set of p-values from a project. In practice, this may not be the case. Scientists who report genome scans frequently give only a short list of those associations with the lowest $k < m$ p-values and do not report the rest (and so do scientists in other fields, who are less likely to admit it). Readers are then left with the problem of how much confidence to have in their "discoveries".

Fortunately, reports of genome scans usually contain an indication of how many associations were really measured. (Unfortunately, this is usually not the case in many other fields.) This can be helpful, given the formulas of the previous subsection. Formulas (1), (2), and (3) imply that for each sorted p-value, Q_i, the corresponding q-value, q_i, depends only on Q_i in the case of one-step procedures, depends on p-values equal to or less than Q_i in the case of step-down procedures, and depends on p-values equal to or greater than Q_i in the case of step-up procedures. This statement implies that q-values can be computed for any subset of p-values in the case of one-step procedures or for the lowest k p-values in the case of step-down procedures without knowing the other p-values. In the case of step-up procedures (which are usually more powerful), life is less simple. However, even in this case, (3) implies that we can still compute conservative estimates of the q-values for the lowest k p-values, which are guaranteed to be upper bounds for the corresponding true q-values, by assuming (conservatively) that all the other p-values in the full set are equal to one.

The `bestof()` option of `qqvalue` allows us to compute conservative q-values for an input variable containing a subset of k p-values by supplying the number m of p-values present in the full set. These conservative q-values will be correct for any subset of k p-values in the case of one-step procedures, correct for the lowest k p-values in the case of step-down procedures, and conservative for the lowest k p-values in the case of step-up procedures. We therefore may be able to show that we can be confident in a list of the highlights of a genome scan as long as we know how large the genome scan was.

3.3 q-values versus discovery sets

A long list of multiple-test procedures was implemented in Stata using the `smileplot` package of Newson and the ALSPAC Study Team (2003, 2010). This package implemented the procedures by generating scalar corrected critical p-values and corresponding discovery set indicator variables. Since then, R users, and now also Stata users, have gained the option of using some of the same procedures to generate q-values. What are the advantages of the two policies?

Multiple-test procedures were originally developed and justified in terms of discovery sets. This is especially the case with multiple-test procedures that control the FDR, such as those of Benjamini and Yekutieli (2001), which are implemented using the options `method(simes)` and `method(yekutieli)` of `smileplot` and `qqvalue`. The Simes procedure, in particular, has the advantageous property that the power to detect an effect of a given size does not necessarily tend to zero as the number of comparisons tends to infinity, in contrast to the case with most other multiple-test procedures (see Genovese and Wasserman [2002]). Discovery sets that are defined to control the FDR also have two very useful multiplicative properties:

- If we control the FDR at $\alpha = \beta \times \gamma$, then we can be $100(1 - \beta)\%$ confident that over $100(1 - \gamma)\%$ of the discovery set will correspond to false null hypotheses (see Newson and the ALSPAC Study Team [2003]).
- If we carry out a preliminary study to find a candidate discovery set (controlling the FDR at β) and then carry out a follow-up study on an independent set of subjects (containing only comparisons from that candidate discovery set and controlling the FDR at γ), then the "overall" FDR of the process generating the follow-up discovery set, prior to the preliminary study, is $\alpha = \beta \times \gamma$ (see Benjamini and Yekutieli [2005]).

The first of these results specifies a trade-off between how confident we can be and how much we can be confident about. The second of these results specifies a similar trade-off between how conservative we need to be in the preliminary study and how conservative we need to be in the follow-up study. Both of these results are entirely evidence-based and objectivist-frequentist, and they are derived without using any authority-based subjectivist claims of having prior knowledge.

In view of these properties of discovery sets, my first impulse was to adopt a standard practice of defining a nested list of three discovery sets that correspond to FDRs of 0.25, 0.05, and 0.01; then to identify these discovery sets by adding one, two, or three stars to the p-value in the table of results; then to add three footnotes to the table, with one, two, and three stars, respectively; and finally to indicate the corrected p-value thresholds under the respective FDRs.

However, the list of FDRs adopted by our research group might not be the same as the lists of FDRs adopted by other research groups, and readers might prefer to have a common analog scale of significance for results from all research groups. Moreover, the

second result seems to assume (implausibly) that scientists conform rigorously and inflexibly to a study plan to the point of defining FDR thresholds prior to the preliminary study and canceling the follow-up study if the discovery set from the preliminary study is empty. Furthermore, if we have an output variable of q-values, then we can define as many discovery sets as we like by selecting observations with q-values at or below our chosen FDRs. For these reasons, I would currently argue that q-values represent an advance on nested discovery sets and that `qqvalue` should probably supersede `smileplot` for most purposes.

It should be stressed that the field of multiple-test procedures is currently in a state of rapid development and that there is not necessarily a consensus on the subject, even among statisticians.

4 Examples

`qqvalue`, like `smileplot`, requires an input dataset with one observation per parameter and also requires data on p-values (and possibly other attributes) for the parameters. In Stata, such datasets are typically created using the official Stata `statsby` command (see [D] **statsby**) or, alternatively, using the `parmest` package of Newson (2003). In our examples, we will assume that such a dataset (or resultsset) has been created and that it contains a variable containing the input p-values.

4.1 Epigenetic assay data in the ALSPAC study

The Avon Longitudinal Study of Parents and Children (ALSPAC) is a multipurpose birth cohort study based at Bristol University, England. The study involves over 14,000 pregnancies in the Avon area of England in the early 1990s, the children from which have been followed through childhood. For further information, refer to the study website at http://www.alspac.bris.ac.uk.

A nested pilot study in ALSPAC subjected the cord blood DNA of 174 subjects (69 girls and 105 boys) to methylation assays. DNA methylation levels (as percentages) were measured at 1,505 methylation sites in the human genome. A methylation site is a position in the genome where a single DNA base can be either methylated (typically implying that a gene is switched off) or unmethylated (typically implying that a gene is switched on). The science of gene switching, including methylation, is known as epigenetics. Each of the 1,505 methylation assays performed on cord blood samples measured the percent of all copies of the appropriate methylation site that were methylated. The methylation data were considered to be useful at 1,495 of these sites.

The methylation levels at these 1,495 sites were distributed non-normally in ways that varied greatly from site to site, being positively skewed at some sites, negatively skewed at other sites, bimodal at others, and semidiscrete at others, with a vast majority of zeros (indicating no methylation) and a small minority of positive values (indicating some methylation). There did not seem to be a unified model whose parameters we

might fit to the data at all sites. I therefore decided to use the methods of Newson (2006b) and Newson (2006a) to generate confidence intervals and p-values for Somers' D and unequal-variance confidence intervals for Theil–Sen median slopes and Hodges–Lehmann median differences. These methods are all implemented using the `somersd` package (Newson 2006a,b).

As a preliminary analysis, I compared methylation levels at each of the 1,495 sites, between the 105 boys and the 69 girls. I used Somers' D and the Hodges–Lehmann median difference, which have distinct confidence intervals sharing a common p-value. Both of these parameters were restricted to comparisons within laboratory batches to remove the influence of batch effects. The estimates, confidence intervals, and p-values were stored in an output dataset (or resultsset) with one observation per methylation site.

q-values for the Simes procedure were then computed using the following Stata code:

```
. qqvalue p, method(simes) qvalue(qq)
. format qq %8.2g
. summarize p qq, detail
```

```
                          P-value
-------------------------------------------------------------
      Percentiles      Smallest
 1%     7.41e-11       3.43e-15
 5%     .0017592       6.52e-14
10%     .0732356       2.87e-13       Obs                1495
25%     .3035019       4.59e-13       Sum of Wgt.        1495
50%      .579294                      Mean           .5381529
                       Largest       Std. Dev.       .304321
75%     .7946141              1
90%     .9225728              1       Variance       .0926113
95%      .966077              1       Skewness       -.310372
99%     .9948297              1       Kurtosis       1.889998

                 q-value by method(simes)
-------------------------------------------------------------
      Percentiles      Smallest
 1%     7.15e-09       5.13e-12
 5%      .035067       4.87e-11
10%     .7131457       1.43e-10       Obs                1495
25%            1       1.72e-10       Sum of Wgt.        1495
50%            1                      Mean           .9052502
                       Largest       Std. Dev.      .2553094
75%            1              1
90%            1              1       Variance       .0651829
95%            1              1       Skewness      -2.859704
99%            1              1       Kurtosis       9.78171
```

Most of the q-values are as high as 1, but some are tiny, which implies that the corresponding p-values would still be in the Simes discovery set even if the FDR was controlled very stringently.

I then plotted the q-values against the position of the corresponding methylation site in the human genome. The human genome has 22 nonsex chromosomes, numbered from 1 to 22, and 2 sex chromosomes, denoted X and Y. Each chromosome has a very long

linear DNA sequence, and each methylation site has a position (or coordinate) on its chromosome. I therefore defined, for each methylation site on each of the chromosomes 1–22 and X, a relative position on a scale from 0 (for the first methylation site on the chromosome) to 100 (for the last methylation site on the chromosome). (There were no methylation sites on the Y chromosome.)

The integer variable denoting the chromosome for each methylation site had the variable name `chromosome`, and the continuous variable denoting the methylation site's relative position had the variable name `mrelpos`. To make the plot, we use the commands `regaxis` and `logaxis`, which are components of the `regaxis` package.[1] The `regaxis` package is very useful in defining axis scales and tick positions, especially for variables such as p-values and q-values that are plotted on a log scale. The Stata code for making the plot is as follows:

```
. regaxis mrelpos, include(0 100) cycle(25) lticks(xlabs)
. logaxis qq, base(10) include(1) lrange(yrange) lticks(ylabs)
> maxticks(12)
. scatter qq mrelpos, msize(2)
> by(chrom, compact row(4) total)
> xlabel(`xlabs´, labsize(4) angle(270))
> yaxis(1 2)
> yscale(reverse log range(`yrange´)) ylab(`ylabs´, labsize(4) angle(0))
> ylabel(0.05, axis(2) labsize(4) angle(0))
> yline(0.05, lpattern(shortdash))
> plotregion(marg(2 2 0.5 0))
```

1. The `regaxis` package can be downloaded from Statistical Software Components at http://econpapers.repec.org/scripts/search/search.asp?ft=regaxis.

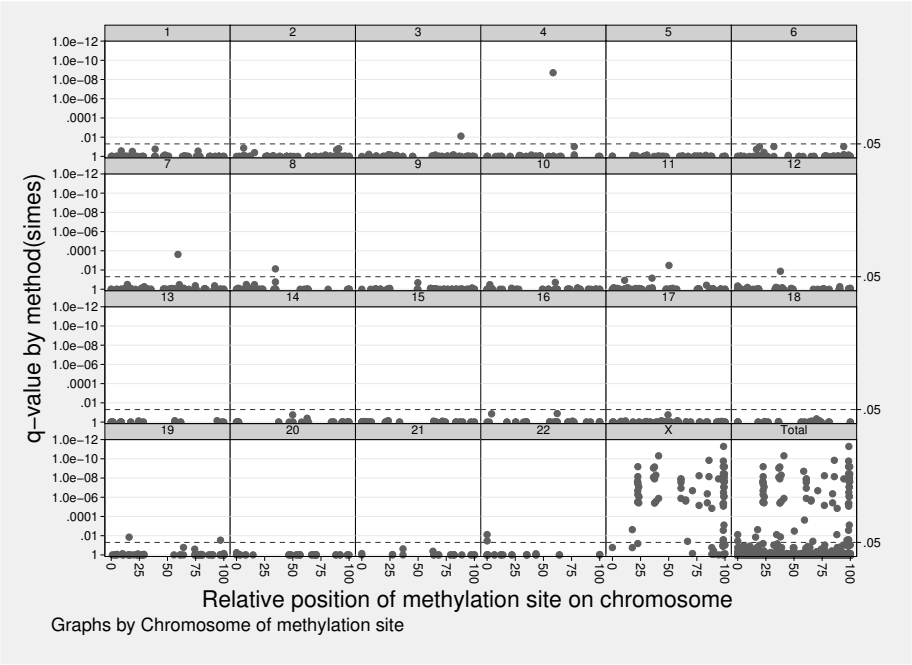

Figure 1. q-values for boy–girl methylation differences at 1,495 sites

The result of this code is given in figure 1, which shows one panel for each of the 23 chromosomes plus one for all methylation sites on all chromosomes. The horizontal axis gives the relative position of the methylation site, and the vertical axis gives the corresponding q-value on a reverse log scale. We see that even allowing for multiple comparisons, there is a large number of statistically significant boy–girl differences in methylation, and that most (but not all) of these are on the X chromosome. This finding does not surprise epigeneticists because a girl has two X chromosomes per cell, of which one is inactivated by methylation, whereas a boy has only one X chromosome per cell, which is not inactivated.

As a comparison, we also used the `multproc` command of the `smileplot` package of Newson and the ALSPAC Study Team (2003, 2010) to define a Simes corrected critical p-value corresponding to an FDR of 0.05. We plotted the p-values of the methylation sites against their positions in the genome, with vertical-axis reference lines at the uncorrected and corrected critical p-values. The result is given as figure 2, which has vertical-axis reference lines at the uncorrected critical p-value of 0.05 and at the corrected critical p-value of 0.00254181. The message of the two figures is qualitatively similar. However, figure 1 is arguably more informative because there you can see at a glance the discovery set under any FDR, rather than the discovery set only at the FDR of 0.05.

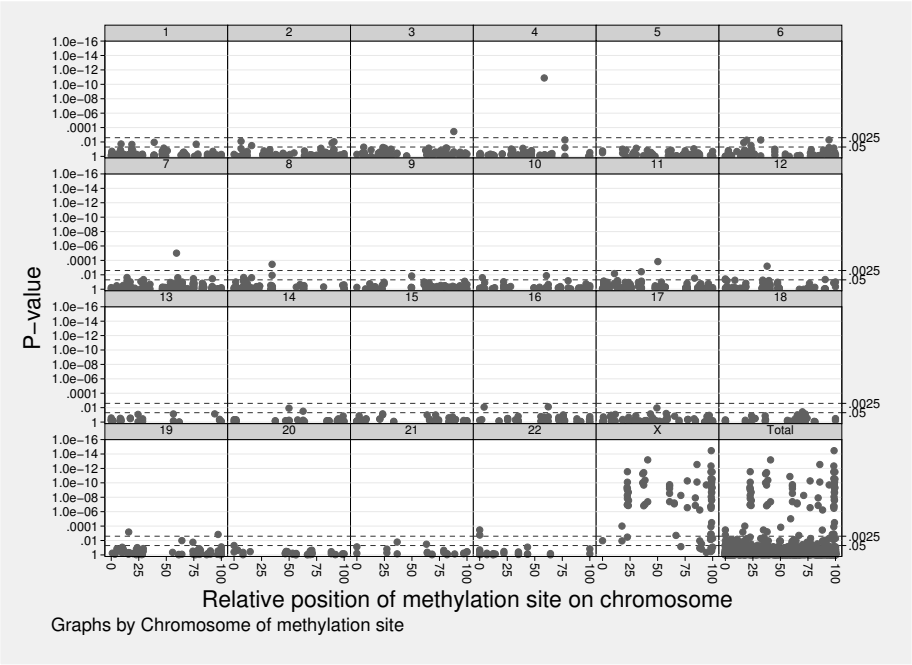

Figure 2. p-values for boy–girl methylation differences at 1,495 sites

4.2 Polymorphisms associated with autism spectrum disorders

In Wang et al. (2009), several research groups combined their genome scan data on the association of autism spectrum disorders with a total of 486,864 single-nucleotide polymorphisms (SNPs). The highlight of their results was a subset of associations (with the lowest p-values) between autism spectrum disorders and six SNPs in the 5p14.1 region of chromosome 5. This region lies between two genes that encode the amino acid sequences of cadherin molecules, which seem to play a role in cell–cell adhesion during the formation of connections between neurons in the developing brain. The authors gave the p-values for these six most significant SNPs.

These p-values were entered into a Stata dataset with one observation for each of the six SNPs and the following variables: `snp` (the name of the SNP), `position` (position of the SNP on chromosome 5), `alleles` (the DNA bases of the more and less frequent alleles of the SNP), and `pcomb` (the p-value for the association, which was determined using combined data from all scans).

We use `pcomb` as the input variable for `qqvalue`, and we output three q-value variables that were generated using the option `bestof(486864)` and the `method()` options `simes`, `yekutieli`, and `bonferroni`, respectively. The Stata code and its output are as follows:

```
. qqvalue pcomb, method(simes) bestof(486864) qv(qqcomb1)
. qqvalue pcomb, method(yekutieli) bestof(486864) qv(qqcomb2)
. qqvalue pcomb, method(bonferroni) bestof(486864) qv(qqcomb3)
. format qqcomb1 qqcomb2 qqcomb3 %8.2g
. list, noobs
```

snp	position	alleles	pcomb	qqcomb1	qqcomb2	qqcomb3
rs4307059	26003460	C/T	2.10e-10	.0001	.0014	.0001
rs7704909	25934678	C/T	9.90e-10	.00018	.0024	.00048
rs12518194	25987318	G/A	1.10e-09	.00018	.0024	.00054
rs4327572	26008578	T/C	2.70e-09	.00033	.0045	.0013
rs1896731	25934776	C/T	4.80e-08	.0047	.064	.023
rs10038113	25938100	C/T	7.40e-08	.006	.082	.036

We see that, although these six SNPs are the most significant of 486,864 investigated, their association with autistic spectrum disorders is still at least suggestive, even if we use the `yekutieli` or `bonferroni` methods, whose q-values are in the variables `qqcomb2` and `qqcomb3`, respectively. The associations are even more impressive if we use the more powerful `simes` method, whose q-values are in the variable `qqcomb1`.

5 Acknowledgments

I would like to thank my Imperial College colleagues Professor Peter Burney, for suggesting that something like q-values might be a good idea, and Adaikalavan Ramasamy, for drawing my attention to the `p.adjust` package in R. In addition, I would like to thank Gordon Smyth of the Walter and Eliza Hall Institute of Medical Research, Victoria, Australia, for writing the current version of `p.adjust`, for some very helpful correspondence when I was certifying `qqvalue`, and for some equally helpful advice on the terminology to use.

I would also like to thank my collaborators in the ALSPAC Study Team (Institute of Child Health, University of Bristol, United Kingdom) for allowing the use of their data in this paper. The whole ALSPAC Study Team comprises interviewers, computer technicians, laboratory technicians, clerical workers, research scientists, volunteers, and managers who continue to make the study possible. The ALSPAC study could not have been undertaken without the cooperation and support of the mothers and midwives who took part or without the financial support of the Medical Research Council, the Department of Health, the Department of the Environment, the Wellcome Trust, and other funders. The ALSPAC study is part of the World Health Organization–initiated European Longitudinal Study of Pregnancy and Childhood. My own work at Imperial College London is financed by the United Kingdom Department of Health.

6 References

Benjamini, Y., and Y. Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)* 57: 289–300.

Benjamini, Y., and D. Yekutieli. 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics* 29: 1165–1188. Also downloadable from Yoav Benjamini's website at http://www.math.tau.ac.il/~ybenja/.

———. 2005. Quantitative trait loci analysis using the false discovery rate. *Genetics* 171: 783–790.

Genovese, C., and L. Wasserman. 2002. Operating characteristics and extensions of the false discovery rate procedure. *Journal of the Royal Statistical Society, Series B* 64: 499–517.

Hochberg, Y. 1988. A sharper Bonferroni procedure for multiple tests of significance. *Biometrika* 75: 800–802.

Holland, B. S., and M. D. Copenhaver. 1987. An improved sequentially rejective Bonferroni test procedure. *Biometrics* 43: 417–423.

Holm, S. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6: 65–70.

Newson, R. 2003. Confidence intervals and p-values for delivery to the end user. *Stata Journal* 3: 245–269.

———. 2006a. Confidence intervals for rank statistics: Percentile slopes, differences, and ratios. *Stata Journal* 6: 497–520.

———. 2006b. Confidence intervals for rank statistics: Somers' *D* and extensions. *Stata Journal* 6: 309–334.

Newson, R., and the ALSPAC Study Team. 2003. Multiple-test procedures and smile plots. *Stata Journal* 3: 109–132.

———. 2010. Software Updates: st0035_1: Multiple-test procedures and smile plots. *Stata Journal* 10: 691–692.

Šidák, Z. 1967. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association* 62: 626–633.

Simes, R. J. 1986. An improved Bonferroni procedure for multiple tests of significance. *Biometrika* 73: 751–754.

Smyth, G., and the R Core Team. 2010. p.adjust. Part of the R package stats. http://www.r-project.org/.

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

Wang, K., H. Zhang, D. Ma, M. Bucan, J. T. Glessner, B. S. Abrahams, D. Salyakina, M. Imielinski, J. P. Bradfield, P. M. A. Sleiman, C. E. Kim, C. Hou, E. Frackelton, R. Chiavacci, N. Takahashi, T. Sakurai, E. Rappaport, C. M. Lajonchere, J. Munson, A. Estes, O. Korvatska, J. Piven, L. I. Sonnenblick, A. I. Alvarez-Retuerto, E. I. Herman, H. Dong, T. Hutman, M. Sigman, S. Ozonoff, A. Klin, T. Owley, J. A. Sweeney, C. W. Brune, R. M. Cantor, R. Bernier, J. R. Gilbert, M. L. Cuccaro, W. M. McMahon, J. Miller, M. W. State, T. H. Wassink, H. Coon, S. E. Levy, R. T. Schultz, J. I. Nurnberger, J. L. Haines, J. S. Sutcliffe, E. H. Cook, N. J. Minshew, J. D. Buxbaum, G. Dawson, S. F. A. Grant, D. H. Geschwind, M. A. Pericak-Vance, G. D. Schellenberg, and H. Hakonarson. 2009. Common genetic variants on 5p14.1 associate with autistic spectrum disorders. *Nature* 459: 528–533.

About the author

Roger B. Newson is a lecturer in medical statistics at Imperial College London, United Kingdom. He works principally in asthma research. He wrote the `qqvalue`, `smileplot`, `parmest`, `somersd`, and `regaxis` Stata packages.