The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–16

Frequentist q–values for multiple–test procedures

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

Abstract. Multiple–test procedures are increasingly important as technology increases the ability of scientists to make large numbers of multiple measurements, as in genome scans. They were originally defined to input a vector of input P–values and an uncorrected critical P–value, interpreted as a familywise error rate (FWER) or a false discovery rate (FDR), 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. The `qqvalue` package, downloadable from SSC, uses an alternative formulation of multiple–test procedures, also used by the R function `p.adjust`. It inputs a variable of P–values and outputs a variable of q–values, equal, in each observation, to the minimum FWER or FDR 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: st0001, `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 21st century biostatistics so far, as technology gives scientists the power to measure unprecedented numbers of comparisons in genome scans, epigenome scans, and metabolome scans. A multiple–test procedure takes, as input, a vector of P–values, corresponding to multiple comparisons testing multiple null hypotheses, and an uncorrected critical P–value, usually interpreted either as a maximum permissible familywise error rate (FWER) or as a maximum permissible false discovery rate (FDR). It outputs a corrected critical P–value, 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, downloadable from SSC and described in Newson and the ALSPAC Study Team (2003).

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 a 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 a 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, multiple–test procedures have been implemented by writing a program to input a vector of P–values and an uncorrected critical P–value, and to output a corrected critical P–value and a discovery set, as is done by the `multproc` module of the `smileplot` package introduced by Newson and the ALSPAC Study Team (2003). An alternative way of implementing multiple–test procedures is used by the R function `p.adjust` (Smyth and the R Core Team (2010)). This inputs a vector of P–values, and a specified multiple–test procedure, and outputs a new vector, parallel to the input vector, of q–values, 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) which 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 1, if there is no FWER or FDR less than 1 for which the corresponding null hypothesis would be rejected.

The Stata `qqvalue` package, downloadable from SSC, is modelled broadly on the R function `p.adjust`, and 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` module of `smileplot`. The term "q–value" was originally introduced as an empirical Bayesian concept by Storey (2003), who aimed to control the positive false discovery rate (pFDR) 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, and are minimum FWERs or FDRs for rejection of individual input P–values, just as Storey's original q–values are minimum pFDRs 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, as 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 docoments and discusses the `qqvalue` package. Section 2 documents the package itself. Section 3 presents, and discusses, the methods and formulas used. Section 4 gives some examples of the use of `qqvalue` in practice.

2 The `qqvalue` package

2.1 Syntax

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

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

`by` *varlist*`:` can be used with `qqvalue`. (See help for `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 1 observation per comparison. It outputs a new variable, containing the q–values corresponding to these P–values, calculated by inverting a multiple–test procedure specified by the user. These q–values represent, 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 familywise error rates or false discovery rates, depending on the procedure used. Optionally, `qqvalue` may output other variables, containing 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` module of the `smileplot` package, which can be downloaded from SSC.

2.3 Options

`method(`*method_name*`)` specifies the multiple–test procedure method to be used for calculating the q–values from the input P–values. The *method_name* 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` module of the `smileplot` package, which can be downloaded from SSC. If `method()` is unset, then it is set to `bonferroni`.

`bestof(`*#*`)` specifies an integer number. 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) 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 1, implying that the q–values will be conservative, and 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 non-positive), 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, which we do not know. This often happens if the input P–values are from a genome scan reported in the literature.

`qvalue(`*newvarname*`)` specifies the name of a new output variable to be generated, containing the q–values calculated from the input P–values, using the multiple–test procedure specified by the `method()` option.

`npvalue(`*newvarname*`)` specifies the name of a new output variable to be generated, containing, 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(`*newvarname*`)` is the name of a new variable to be generated, containing, 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(`*newvarname*`)` specifies the name of a new output variable to be generated, containing the s–values calculated from the input P–values. The s–values are an intermediate result, 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 calculated by the `multproc` module of the `smileplot` package. These rank–specific P–value thresholds are returned in the generated variable specified by the `critical()` option of `multproc`. Note that the s–values may be greater than 1.

`rvalue(`*newvarname*`)` specifies the name of a new output variable to be generated, containing the r–values calculated from the input P–values. The r–values are an intermediate result, 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 1. The q–values are calculated from the r–values using a procedure 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 will 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 so that it can 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` option of the `smileplot` package, documented in Newson and the ALSPAC Study Team (2003). We will therefore use a notation as consistent as possible with that source, using upper– and lower–case symbols to denote different quantities, in order 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 $\hat{\theta}_1, \ldots, \hat{\theta}_m$, and having the values $\theta_1^{(0)}, \ldots, \theta_m^{(0)}$ under their respective null hypotheses. (Typically, $\theta_i^{(0)}$ is 0 for difference parameters such as median differences, or 1 for ratio parameters such as median ratios.) 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 $0 \leq \alpha \leq 1$,

$$\Pr\left(P_i \, \leq \, \alpha \;\middle|\; \theta_i = \theta_i^{(0)} \right) \, \leq \, \alpha. \tag{1}$$

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, $Q_{R_i} = P_i$. (Note that the Q_i are *not* the q–values, which we will define in due course.)

The methods used by the `multproc` module 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 $p_{\rm unc}$, calculating a corrected P–value threshold $p_{\rm cor}$ from $p_{\rm unc}$ and $Q_1, \ldots, Q_m$, and defining the acceptable subset C to be the subset of indices i such that $P_i > p_{\rm cor}$. The methods used by `qqvalue`, by contrast, are derived by inverting the methods used by `multproc`, as 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()`

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	Non–negative
`holm`	Step–down	FWER	Arbitrary
`holland`	Step–down	FWER	Non–negative
`hochberg`	Step–up	FWER	Independence
`simes`	Step–up	FDR	Non–negative
`yekutieli`	Step–up	FDR	Arbitrary

option, are a subset of those used by `multproc`. They are listed in Table 1, and classified in 3 ways. These are 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, non–negative, or 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), in Subsection 3.1. Each of the methods of `multproc` works by specifying a non–decreasing sequence of individual critical P–values $c_1, \ldots, c_m$, corresponding 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 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, and 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 3 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 1 or greater, if no FWER or FDR less than 1 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 depending on whether the procedure is one–step, step–down, or step–up. For a one–step procedure, this formula is
$$q_i = r_i \tag{2}$$
and, for a step–down procedure, it is
$$q_i = \max\{r_j : j \leq i\} \tag{3}$$
and for a step–up procedure, it is
$$q_i = \min\{r_j : j \geq i\} \tag{4}$$
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), Subsection 3.1. They are given as follows, together with references for the original multiple–test procedures:

One–step methods

1. `bonferroni`.
$$s_i = mQ_i \tag{5}$$
2. `sidak` (Šidák (1967)).
$$s_i = 1 - (1 - Q_i)^m \tag{6}$$

Step–down methods

1. `holm` (Holm (1979)).
$$s_i = (m - i + 1)Q_i \tag{7}$$
2. `holland` (Holland and Copenhaver (1987)).
$$s_i = 1 - (1 - Q_i)^{m-i+1} \tag{8}$$

Step–up methods

1. `hochberg` (Hochberg (1988)).
$$s_i = (m - i + 1)Q_i \tag{9}$$
Note that the s_i are the same as those for the step–down Holm procedure.

2. `simes` (Simes (1986); Benjamini and Hochberg (1995); Benjamini and Yekutieli (2001), first method).

$$s_i = \frac{m}{i} Q_i \tag{10}$$

3. `yekutieli` (Benjamini and Yekutieli (2001), second method).

$$s_i = \frac{m}{i} Q_i \sum_{j=1}^{m} j^{-1} \tag{11}$$

Note that all of these expressions for s_i are increasing in Q_i and increasing in m. and non–increasing (constant in the case of one–step procedures) in i. The corresponding expressions for $r_i = \min(s_i, 1)$ will therefore be non–decreasing in Q_i and in m, and non–increasing 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 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.) The reader is then left with the problem of how much confidence to have in these "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 (2), (3) and (4) 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 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, Formula (4) implies that we can still compute conservative estimates of the q–values for the lowest k P–values, 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 1.

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). This package implemented them 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 false discovery rate (FDR), such as those of Benjamini and Yekutieli (2001), imlemented 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 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, followed by 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 tradeoff between how confident we can be and how much we can be confident about. The second of these results specifies a similar tradeoff 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 derived without using any authority–based subjectivist claims to have "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, corresponding to FDRs of 0.25, 0.05, and 0.01, and to identify these discovery sets by adding one, two or three stars to the P–value in the table of results, and to add three footnotes to the table, with one, two and three stars, respectively, indicating 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 cancelling the follow–up study if the discovery set from the preliminary study is empty. And, *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 1 observation per parameter, and 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), downloadable from SSC. In our examples, we will assume that such a dataset (or resultsset) has been created, and 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 multi-purpose birth cohort study based at Bristol University, England, involving 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 was performed in ALSPAC, in which the cord blood DNA of 174 subjects (69 girls and 105 boys) was subjected to methylation assays, measuring DNA methylation levels (percent) at 1505 methylation sites in the human genome. A methylation site is a position in the genome where a single DNA base can either be 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 1505 methylation assays measured the percent of all copies of the appropriate methylation site, in the cord blood sample, that were methylated. The methylation data was considered to be useful at 1495 of these sites.

The distributions of the methylation levels at these 1495 sites was 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 "semi–discrete" at others again, 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. We therefore decided to use the methods of Newson (2006a) and Newson (2006b), 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, which can be downloaded from SSC.

As a preliminary analysis, we compared methylation levels, at each of the 1495 sites, between the 105 boys and the 69 girls, using 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
. summ 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
```

We note that most of the q–values are as high as 1, but that some are tiny, implying that the corresponding P–values would still be in the Simes discovery set, even if the FDR was controlled very stringently.

We then plotted the q–values against the position of the corresponding methylation site in the human genome. The human genome has 22 non–sex 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 co–ordinate) on its chromosome. We 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 used the modules `regaxis` and `logaxis`, which are components of the `regaxis` package, which can be downloaded from SSC. The `regaxis` package is very useful in defining axis scales and tick positions, especially for variables that are plotted on a log scale, such as P–values and q–values. The Stata code for making the plot was as follows:

```
. regaxis mrelpos, inc(0 100) cy(25) ltick(xlabs)
. logaxis qq, base(10) inc(1) lrange(yrange) ltick(ylabs) maxt(12)
. scatter qq mrelpos, msize(2) ///
>   by(chrom, compact row(4) total) ///
>   xlab(`xlabs´, labsize(4) angle(270)) ///
>   yaxis(1 2) ///
>   yscal(reverse log range(`yrange´)) ylab(`ylabs´, labsize(4) angle(0)) ///
>   ylab(0.05, axis(2) labsize(4) angle(0)) ///
>   yline(0.05, lpattern(shortdash)) ///
>   plotregion(marg(2 2 0.5 0))
```

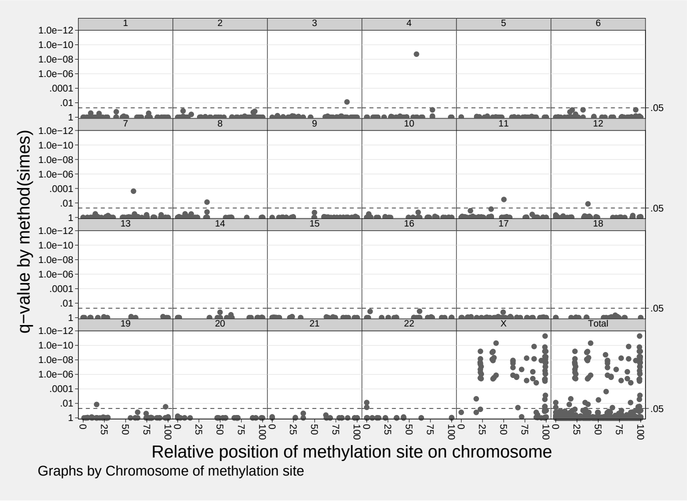

Figure 1: q–values for boy–girl methylation differences at 1495 sites.

The result of this code is given in Figure 1. There is one panel for each of the 23 chromosomes, and 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

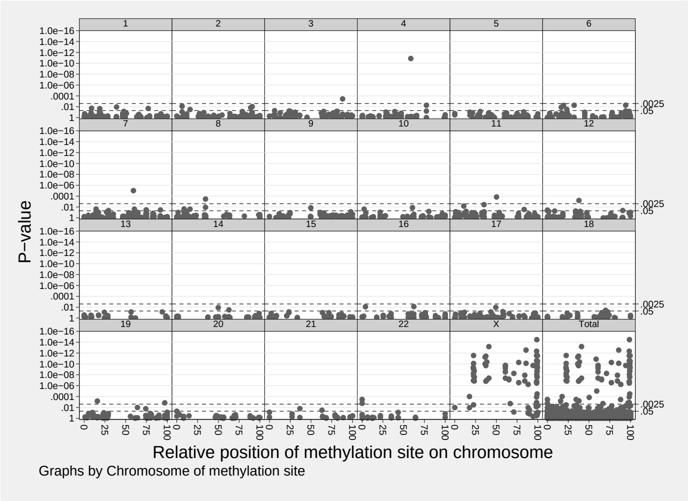

Figure 2: P–values for boy–girl methylation differences at 1495 sites.

comparisons, there are a large number of statistically significant boy–girl differences in methylation, and most (but not all) of these are on the X–chromosome. This 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` module of the `smileplot` package of Newson and the ALSPAC Study Team (2003) to define a Simes corrected critical P–value corresponding to a FDR of 0.05, and 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 .05 and at the corrected critical P–value of .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.

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 486864 single–nucleotide polymorphisms (SNPs). The highlight of their results was a subset of associations, with the lowest P–values, between autism spectrum disorders and 6 SNPs in the 5p14.1 region of Chromosome 5. This region is between 2 genes encoding 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 6 most significant SNPs.

These P–values were entered into a Stata dataset, with 1 observation for each of the 6 SNPs, and 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, using combined data from all scans).

We used `pcomb` as the input variable for `qqvalue`, and output 3 q–value variables, generated using the option `bestof(486864)` and the `method()` options `simes`, `yekutieli` and `bonferroni`, respectively. The Stata code, and its output, was 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 the SNPs are the most significant of 486864 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. And the associations are even more impressive if we use the more powerful `simes` method, whose q–values are in the variable `qqcomb1`.

5 Acknowledgements

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. We would also like to thank our collaborators in the ALSPAC Study Team (Institute of Child Health, University of Bristol, Bristol, UK) 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 co-operation and support of the mothers and midwives who took part, or 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 WHO-initiated European Longitudinal Study of Pregnancy and Childhood (ELSPAC). My own work at Imperial College London is financed by the UK 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. *The Annals of Statistics* 29: 1165–1188.

———. 2005. Quantitative trait loci analysis using the false discovery rate. *Genetics* 171(2): 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(3): 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(3): 245–269.

———. 2006a. Confidence intervals for rank statistics: Somers' D and extensions. *Stata Journal* 6(3): 309–334.

———. 2006b. Confidence intervals for rank statistics: Percentile slopes, differences, and ratios. *Stata Journal* 6(4): 497–520.

Newson, R., and the ALSPAC Study Team. 2003. Multiple–test procedures and smile plots. *Stata Journal* 3(2): 109–132.

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. *The Annals of Statistics* 31(6): 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, UK, working principally in asthma research. He wrote the `qqvalue`, `smileplot`, `parmest`, `somersd` and `regaxis` packages.