1

Multiple Testing Procedures and Applications to Genomics

Katherine S. Pollard, Sandrine Dudoit, Mark J. van der Laan

October 5, 2004

Abstract

The Bioconductor R package multtest implements widely applicable resampling-based single-step and stepwise multiple testing procedures (MTP) for controlling a broad class of Type I error rates, in testing problems involving general data generating distributions (with arbitrary dependence structures among variables), null hypotheses, and test statistics [1, 2, 3, 4, 5]. The current version of multtest provides MTPs for null hypotheses concerning means, differences in means, and regression parameters in linear and Cox proportional hazards models. Both bootstrap and permutation estimators of the test statistics (t- or F-statistics) null distribution are available. Procedures are provided to control Type I error rates defined as tail probabilities and expected values of arbitrary functions of the numbers of Type I errors, $V_n$, and rejected hypotheses, $R_n$. These error rates include: the generalized family-wise error rate, $gFWER(k) = Pr(V_n > k)$, or chance of at least $(k+1)$ false positives (the special case $k = 0$ corresponds to the usual family-wise error rate, FWER); tail probabilities $TPPFP(q) = Pr(V_n/R_n > q)$ for the proportion of false positives among the rejected hypotheses; the false discovery rate, $FDR = E[V_n/R_n]$. Single-step and step-down common-cut-off (maxT) and common-quantile (minP) procedures, that take into account the joint distribution of the test statistics, are implemented to control the FWER. In addition, augmentation procedures are provided to control the $gFWER$, TPPFP, and FDR, based on any initial FWER-controlling procedure. The results of a multiple testing procedure are summarized using rejection regions for the test statistics, confidence regions for the parameters of interest, and adjusted $p$-values. The modular design of the multtest package allows interested users to readily extend the package's functionality,
by inserting additional functions for test statistics and testing procedures. The S4 class/method object-oriented programming approach was adopted to summarize the results of a MTP. The multiple testing procedures are applied to the Acute Lymphoblastic Leukemia (ALL) dataset of Chiaretti et al. [6], available in the R package ALL, to identify genes whose expression measures are associated with (possibly censored) biological and clinical outcomes such as: cytogenetic test status (normal vs. abnormal), tumor molecular subtype (BCR/ABL, NEG, ALL1/AF4, E2A/PBX1, p15/p16, NUP-98), and time to relapse.

1.1 Introduction

1.1.1 Motivation

Current statistical inference problems in areas such as genomics, astronomy, and marketing routinely involve the simultaneous test of thousands, or even millions, of null hypotheses. Examples of testing problems in genomics include:

- the identification of differentially expressed genes in microarray experiments, i.e., genes whose expression measures are associated with possibly censored responses or covariates interest;
- tests of association between gene expression measures and Gene Ontology (GO) annotation (www.geneontology.org);
- the identification of transcription factor binding sites in ChIP-Chip experiments, where chromatin immunoprecipitation (ChIP) of transcription factor bound DNA is followed by microarray hybridization (Chip) of the IP-enriched DNA [7];
- the genetic mapping of complex traits using single nucleotide polymorphisms (SNP).

The above testing problems share the following general characteristics:

- inference for high-dimensional multivariate distributions, with complex and unknown dependence structures among variables;
- broad range of parameters of interest, such as, regression coefficients in model relating patient survival to genome-wide transcript levels or DNA copy numbers, pairwise gene correlations between transcript levels;
- many null hypotheses, in the thousands or even millions;
- complex dependence structures among test statistics, e.g., Gene Ontology directed acyclic graph (DAG).
Motivated by these applications, we have developed resampling-based single-step and step-down multiple testing procedures (MTP) for controlling a broad class of Type I error rates, in testing problems involving general data generating distributions (with arbitrary dependence structures among variables), null hypotheses, and test statistics [1, 2, 3, 4, 5]. In particular, Dudoit et al. [2] and Pollard & van der Laan [5] derive single-step common-cut-off and common-quantile procedures for controlling arbitrary parameters of the distribution of the number of Type I errors, such as the generalized family-wise error rate, $gFWER(k)$, or chance of at least $(k+1)$ false positives. van der Laan et al. [3] focus on control of the family-wise error rate, $FWER = gFWER(0)$, and provide step-down common-cut-off and common-quantile procedures, based on maxima of test statistics (maxT) and minima of unadjusted $p$-values (minP), respectively. Dudoit & van der Laan [1] and van der Laan et al. [4] propose a general class of augmentation multiple testing procedures (AMTP), obtained by adding suitably chosen null hypotheses to the set of null hypotheses already rejected by an initial MTP. In particular, given any FWER-controlling procedure, they show how one can trivially obtain procedures controlling tail probabilities for the number ($gFWER$) and proportion (TPPFP) of false positives among the rejected hypotheses.

A key feature of our proposed MTPs is the test statistics null distribution (rather than data generating null distribution) used to derive rejection regions (i.e., cut-offs) for the test statistics and resulting adjusted $p$-values [1, 2, 3, 4, 5]. For general null hypotheses, defined in terms of submodels for the data generating distribution, this null distribution is the asymptotic distribution of the vector of null value shifted and scaled test statistics. Resampling procedures (e.g., based on the non-parametric or model-based bootstrap) are proposed to conveniently obtain consistent estimators of the null distribution and the resulting test statistic cut-offs and adjusted $p$-values [2, 3, 5].

The Bioconductor R package multtest provides software implementations of the above multiple testing procedures.

1.1.2 Outline

The present chapter provides a summary of our proposed multiple testing procedures (1, 2, 3, 4, 5; Section 1.2), discusses their software implementation in the Bioconductor R package multtest (Section 1.3), and describes their application to the ALL dataset of Chiaretti et al. [6] (Section 1.4).

Specifically, given a multivariate dataset (stored as a matrix, data.frame, or microarray object of class exprSet) and user-supplied choices for the test statistics, Type I error rate and its target level, resampling-based estimator of the test statistics null distribution, and procedure for error rate control, the main user-level function MTP returns unadjusted and adjusted $p$-values, cut-off vectors for the test statistics, and estimates and confidence regions.
for the parameters of interest. Both bootstrap and permutation estimators of the test statistics null distribution are available and can optionally be output to the user. The variety of models and hypotheses, test statistics, Type I error rates, and MTPs currently implemented are discussed in Section 1.3.2. The S4 class/method object-oriented programming approach was adopted to represent the results of a MTP. Several methods are defined to produce numerical and graphical summaries of these results (Section 1.3.3). A modular programming approach, which utilizes function closures, allows interested users to readily extend the package’s functionality, by inserting functions for new test statistics and testing procedures (Section 1.3.4). Ongoing efforts are discussed in Section 1.5.

1.2 Methods

1.2.1 Multiple hypothesis testing framework

Hypothesis testing is concerned with using observed data to test hypotheses, i.e., make decisions, regarding properties of the unknown data generating distribution. Below, we discuss in turn the main ingredients of a multiple testing problem, namely: data, null and alternative hypotheses, test statistics, multiple testing procedure (MTP) to define rejection regions for the test statistics, Type I and Type II errors, and adjusted \( p \)-values. The crucial choice of a test statistics null distribution is addressed in Section 1.2.2. Specific proposals of MTPs are given in Sections 1.2.3 – 1.2.5.

Data. Let \( X_1, \ldots, X_n \) be a random sample of \( n \) independent and identically distributed (i.i.d.) random variables, \( X \sim P \in M \), where the data generating distribution \( P \) is known to be an element of a particular statistical model \( M \) (i.e., a set of possibly non-parametric distributions).

Null and alternative hypotheses. In order to cover a broad class of testing problems, define \( M \) null hypotheses in terms of a collection of submodels, \( M(m) \subseteq M, m = 1, \ldots, M \), for the data generating distribution \( P \). The \( M \) null hypotheses are defined as \( H_0(m) = I(P \in M(m)) \) and the corresponding alternative hypotheses as \( H_1(m) = I(P \notin M(m)) \).

In many testing problems, the submodels concern parameters, i.e., functions of the data generating distribution \( P, \Psi(P) = \psi = (\psi(m) : m = 1, \ldots, M) \), such as means, differences in means, correlations, and parameters in linear models, generalized linear models, survival models, time-series models, dose-response models, etc. One distinguishes between two types of testing problems: one-sided tests, where \( H_0(m) = I(\psi(m) \leq \psi_0(m)) \), and two-sided tests, where \( H_0(m) = I(\psi(m) = \psi_0(m)) \). The hypothesized null values, \( \psi_0(m) \), are frequently zero.
Let \( \mathcal{H}_0 = \mathcal{H}_0(P) \equiv \{ h : H_0(h) = 1 \} = \{ m : P \in \mathcal{M}(m) \} \) be the set of \( h \in \mathcal{H} \) true null hypotheses, where we note that \( \mathcal{H}_0 \) depends on the data generating distribution \( P \). Let \( \mathcal{H}_1 = \mathcal{H}_1(P) \equiv \mathcal{H}_0(P) = \{ m : H_1(m) = 1 \} = \{ m : P \notin \mathcal{M}(m) \} \) be the set of \( h \equiv |\mathcal{H}_1| = M - h_0 \) false null hypotheses, i.e., true positives. The goal of a multiple testing procedure is to accurately estimate the set \( \mathcal{H}_0 \), and thus its complement \( \mathcal{H}_1 \), while controlling probabilistically the number of false positives at a user-supplied level \( \alpha \).

**Test statistics.** A testing procedure is a data-driven rule for deciding whether or not to reject each of the \( M \) null hypotheses \( H_0(m) \), i.e., declare that \( H_0(m) \) is false (zero) and hence \( P \notin \mathcal{M}(m) \). The decisions to reject or not the null hypotheses are based on an \( M \)-vector of test statistics, \( T_n = (T_n(m) : m = 1, \ldots, M) \), that are functions of the data, \( X_1, \ldots, X_n \). Denote the typically unknown (finite sample) joint distribution of the test statistics \( T_n \) by \( Q_n = Q_n(P) \).

Single-parameter null hypotheses are commonly tested using \( t \)-statistics, i.e., standardized differences,

\[
T_n(m) = \frac{\text{Estimator} - \text{Null value}}{\text{Standard error}} = \frac{\sqrt{n} \psi_n(m) - \psi_0(m)}{\sigma_n(m)}. \tag{1.1}
\]

In general, the \( M \)-vector \( \psi_n = (\psi_n(m) : m = 1, \ldots, M) \) denotes an asymptotically linear estimator of the parameter \( M \)-vector \( \psi = (\psi(m) : m = 1, \ldots, M) \) and \( (\sigma_n(m)/\sqrt{n} : m = 1, \ldots, M) \) denote consistent estimators of the standard errors of the components of \( \psi_n \). For tests of means, one recovers the usual one-sample and two-sample \( t \)-statistics, where the \( \psi_n(m) \) and \( \sigma_n(m) \) are based on sample means and variances, respectively. In some settings, it may be appropriate to use (unstandardized) difference statistics, \( T_n(m) \equiv \sqrt{n}(\psi_n(m) - \psi_0(m)) \). Test statistics for other types of null hypotheses include \( F \)-statistics, \( \chi^2 \)-statistics, and likelihood ratio statistics.

**Example: ALL microarray dataset.** Suppose that, as in the analysis of the ALL dataset of Chiaretti et al. [6] (Section 1.4), one is interested in identifying genes that are differentially expressed in two populations of ALL cancer patients, those with normal cytogenetic test status and those with abnormal test. The data consist of random \( J \)-vectors \( X \), where the first \( M \) entries of \( X \) are microarray expression measures on \( M \) genes of interest and the last entry, \( X(J) \), is an indicator for cytogenetic test status (1 for normal, 0 for abnormal). Then, the parameter of interest is an \( M \)-vector of differences in mean expression measures in the two populations, \( \psi(m) = E[X(m)|X(J) = 0] - E[X(m)|X(J) = 1], m = 1, \ldots, M \). To identify genes with higher mean expression measures in the abnormal compared to the normal cytogenetics subjects, one can test the one-sided null hypotheses \( H_0(m) = I(\psi(m) \leq 0) \) vs. the alternative hypotheses \( H_1(m) = I(\psi(m) > 0) \).
Multiple testing procedure. A multiple testing procedure (MTP) provides rejection regions, $C_n(m)$, i.e., sets of values for each test statistic $T_n(m)$ that lead to the decision to reject the null hypothesis $H_0(m)$. In other words, a MTP produces a random (i.e., data-dependent) subset $R_n$ of rejected hypotheses that estimates $H_1$, the set of true positives,

$$R_n = \mathcal{R}(T_n, Q_{0n}, \alpha) \equiv \{ m : H_0(m) \text{ is rejected} \} = \{ m : T_n(m) \in C_n(m) \},$$

(1.3)

where $C_n(m) = \mathcal{C}(T_n, Q_{0n}, \alpha)(m)$, $m = 1, \ldots, M$, denote possibly random rejection regions. The long notation $\mathcal{R}(T_n, Q_{0n}, \alpha)$ and $\mathcal{C}(T_n, Q_{0n}, \alpha)(m)$ emphasizes that the MTP depends on: (i) the data, $X_1, \ldots, X_n$, through the $M$-vector of test statistics, $T_n = (T_n(m) : m = 1, \ldots, M)$; (ii) a test statistics null distribution, $Q_{0n}$ (Section 1.2.2); and (iii) the nominal level $\alpha$, i.e., the desired upper bound for a suitably defined false positive rate.

Unless specified otherwise, it is assumed that large values of the test statistic $T_n(m)$ provide evidence against the corresponding null hypothesis $H_0(m)$, that is, we consider rejection regions of the form $C_n(m) = (c_n(m), \infty)$, where $c_n(m)$ are to-be-determined cut-offs, or critical values.

Type I and Type II errors. In any testing situation, two types of errors can be committed: a false positive, or Type I error, is committed by rejecting a true null hypothesis, and a false negative, or Type II error, is committed when the test procedure fails to reject a false null hypothesis. The situation can be summarized by Table 1.1 below, where the number of Type I errors is $V_n \equiv \sum_{m \in \mathcal{H}_0} I(T_n(m) \in C_n(m)) = |R_n \cap \mathcal{H}_0|$ and the number of Type II errors is $U_n \equiv \sum_{m \in \mathcal{H}_1} I(T_n(m) \notin C_n(m)) = |R_n^c \cap \mathcal{H}_1|$. Note that both $U_n$ and $V_n$ depend on the unknown data generating distribution $P$ through the unknown set of true null hypotheses $\mathcal{H}_0 = \mathcal{H}_0(P)$. The numbers $h_0 = |\mathcal{H}_0|$ and $h_1 = |\mathcal{H}_1| = M - h_0$ of true and false null hypotheses are unknown parameters, the number of rejected hypotheses $R_n \equiv \sum_{m=1}^M I(T_n(m) \in C_n(m)) = |R_n|$ is an observable random variable, and the entries in the body of the table, $U_n, h_1 - U_n, V_n$, and $h_0 - V_n$, are unobservable random variables (depending on $P$, through $\mathcal{H}_0(P)$).
Table 1.1. Type I and Type II errors in multiple hypothesis testing.

<table>
<thead>
<tr>
<th>Null hypotheses</th>
<th>not rejected</th>
<th>rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>$</td>
<td>R_n \cap H_0</td>
</tr>
<tr>
<td>false</td>
<td>$U_n =</td>
<td>R_n \cap H_1</td>
</tr>
</tbody>
</table>

Ideally, one would like to simultaneously minimize both the chances of committing a Type I error and a Type II error. Unfortunately, this is not feasible and one seeks a trade-off between the two types of errors. A standard approach is to specify an acceptable level $\alpha$ for the Type I error rate and derive testing procedures, i.e., rejection regions, that aim to minimize the Type II error rate, i.e., maximize power, within the class of tests with Type I error rate at most $\alpha$.

**Type I error rates.** When testing multiple hypotheses, there are many possible definitions for the Type I error rate (and power). Accordingly, we adopt a general definition of Type I error rates, as parameters, $\theta_n = \theta(F_{V_n,R_n})$, of the joint distribution $F_{V_n,R_n}$ of the numbers of Type I errors $V_n$ and rejected hypotheses $R_n$. Such a general representation covers the following commonly-used Type I error rates.

1. **Generalized family-wise error rate** (gFWER), or probability of at least $(k + 1)$ Type I errors, $k = 0, \ldots, (h_0 - 1)$,
   
   $gFWER(k) \equiv Pr(V_n > k) = 1 - F_{V_n}(k). \quad (1.4)$

   When $k = 0$, the gFWER is the usual family-wise error rate, FWER, controlled by the classical Bonferroni procedure.

2. **Per-comparison error rate** (PCER), or expected proportion of Type I errors among the $M$ tests,
   
   $PCER \equiv \frac{1}{M} E[V_n] = \frac{1}{M} \int v dF_{V_n}(v). \quad (1.5)$

3. **Tail probabilities for the proportion of false positives** (TPPFP) among the rejected hypotheses,
   
   $TPPFP(q) \equiv Pr(V_n/R_n > q) = 1 - F_{V_n/R_n}(q), \quad q \in (0, 1), \quad (1.6)$

   with the convention that $V_n/R_n \equiv 0$, if $R_n = 0$. 

4. **False discovery rate (FDR)**, or expected value of the proportion of false positives among the rejected hypotheses, 

\[
FDR \equiv E[V_n/R_n] = \int qdF_{V_n/R_n}(q),
\]

again with the convention that \(V_n/R_n \equiv 0\) if \(R_n = 0\) [8].

Note that while the gFWER is a parameter of only the marginal distribution \(F_{V_n}\) for the number of Type I errors \(V_n\) (tail probability, or survivor function, for \(V_n\)), the TPPFP is a parameter of the joint distribution of \((V_n, R_n)\) (tail probability, or survivor function, for \(V_n/R_n\)). Error rates based on the proportion of false positives (e.g., TPPFP and FDR) are especially appealing for the large-scale testing problems encountered in genomics, compared to error rates based on the number of false positives (e.g., gFWER), as they do not increase exponentially with the number of hypotheses. The above four error rates are part of the broad class of Type I error rates considered in Dudoit & van der Laan [1] and defined as tail probabilities \(\Pr(g(V_n, R_n) > q)\) and expected values \(E[g(V_n, R_n)]\) for an arbitrary function \(g(V_n, R_n)\) of the numbers of false positives \(V_n\) and rejected hypotheses \(R_n\). The gFWER and TPPFP correspond to the special cases \(g(V_n, R_n) = V_n\) and \(g(V_n, R_n) = V_n/R_n\), respectively.

**Adjusted p-values.** The notion of p-value extends directly to multiple testing problems, as follows. Given a MTP, \(\mathcal{R}_n = \mathcal{R}(T_n, Q_0n, \alpha)\), the adjusted p-value, \(\tilde{P}_0n(m) = P(T_n, Q_0n)(m)\), for null hypothesis \(H_0(m)\), is defined as the smallest Type I error level \(\alpha\) at which one would reject \(H_0(m)\), that is,

\[
\tilde{P}_0n(m) = \inf \{\alpha \in [0,1] : \text{Reject } H_0(m) \text{ at MTP level } \alpha\} = \inf \{\alpha \in [0,1] : m \in \mathcal{R}_n\} = \inf \{\alpha \in [0,1] : T_n(m) \in \mathcal{C}_n(m)\}, \quad m = 1, \ldots, M.
\]

As in single hypothesis tests, the smaller the adjusted p-value, the stronger the evidence against the corresponding null hypothesis. The main difference between unadjusted (i.e., for the test of a single hypothesis) and adjusted p-values is that the latter are defined in terms of the Type I error rate for the entire testing procedure, i.e., take into account the multiplicity of tests. For example, the adjusted p-values for the classical Bonferroni procedure for FWER control are given by \(\tilde{P}_0n(m) = \min(M \tilde{P}_0n(m), 1)\), where \(\tilde{P}_0n(m)\) is the unadjusted p-value for the test of single hypothesis \(H_0(m)\).

We now have two representations for a MTP, in terms of rejection regions for the test statistics and in terms of adjusted p-values

\[
\mathcal{R}_n = \{m : T_n(m) \in \mathcal{C}_n(m)\} = \{m : \tilde{P}_0n(m) \leq \alpha\}.
\]

Again, as in the single hypothesis case, an advantage of reporting adjusted p-values, as opposed to only rejection or not of the hypotheses, is that the
level $\alpha$ of the test does not need to be determined in advance, that is, results of the multiple testing procedure are provided for all $\alpha$. Adjusted $p$-values are convenient and flexible summaries of the strength of the evidence against each null hypothesis, in terms of the Type I error rate for the entire MTP (gFWER, TPPFP, FDR, or any other suitably defined error rate).

**Stepwise multiple testing procedures.** One usually distinguishes between two main classes of multiple testing procedures, single-step and stepwise procedures. In *single-step procedures*, each null hypothesis is evaluated using a rejection region that is independent of the results of the tests of other hypotheses. Improvement in power, while preserving Type I error rate control, may be achieved by *stepwise procedures*, in which rejection of a particular null hypothesis depends on the outcome of the tests of other hypotheses. That is, the (single-step) test procedure is applied to a sequence of successively smaller nested random (i.e., data-dependent) subsets of null hypotheses, defined by the ordering of the test statistics (common cut-offs) or unadjusted $p$-values (common-quantile cut-offs). In *step-down procedures*, the hypotheses corresponding to the most significant test statistics (i.e., largest absolute test statistics or smallest unadjusted $p$-values) are considered successively, with further tests depending on the outcome of earlier ones. As soon as one fails to reject a null hypothesis, no further hypotheses are rejected. In contrast, for *step-up procedures*, the hypotheses corresponding to the least significant test statistics are considered successively, again with further tests depending on the outcome of earlier ones. As soon as one hypothesis is rejected, all remaining more significant hypotheses are rejected.

**Confidence regions.** For the test of single-parameter null hypotheses and for any Type I error rate of the form $\theta(F_{V_n})$, Dudoit & van der Laan [1] and Pollard & van der Laan [5] provide results on the correspondence between single-step MTPs and $\theta$–specific confidence regions.

### 1.2.2 Test statistics null distribution

**Test statistics null distribution.** One of the main tasks in specifying a MTP is to derive rejection regions for the test statistics such that the Type I error rate is controlled at a desired level $\alpha$, i.e., such that $\theta(F_{V_n}, R_n) \leq \alpha$, for finite sample control, or $\limsup_{n} \theta(F_{V_n}, R_n) \leq \alpha$, for asymptotic control. However, one is immediately faced with the problem that the true distribution $Q_n = Q_n(P)$ of the test statistics $T_n$ is usually unknown, and hence, so are the distributions of the numbers of Type I errors, $V_n = \sum_{m \in H_0} I(T_n(m) \in C_n(m))$, and rejected hypotheses, $R_n = \sum_{m=1}^{M} I(T_n(m) \in C_n(m))$. In practice, the test statistics true distri-
bution \( Q_n(P) \) is replaced by a null distribution \( Q_0 \) (or estimate thereof, \( Q_{0n} \)), in order to derive rejection regions, \( C(T_n, Q_0, \alpha)(m) \), and resulting adjusted \( p \)-values, \( \tilde{P}(T_n, Q_0)(m) \).

The choice of null distribution \( Q_0 \) is crucial, in order to ensure that (finite sample or asymptotic) control of the Type I error rate under the assumed null distribution \( Q_0 \) does indeed provide the required control under the true distribution \( Q_n(P) \). For proper control, the null distribution \( Q_0 \) must be such that the Type I error rate under this assumed null distribution dominates the Type I error rate under the true distribution \( Q_n(P) \). That is, one must have \( \theta(F_{V_n, R_n}) \leq \theta(F_{V_0, R_0}) \), for finite sample control, and \( \lim \sup_n \theta(F_{V_n, R_n}) \leq \theta(F_{V_0, R_0}) \), for asymptotic control, where \( V_0 \) and \( R_0 \) denote, respectively, the numbers of Type I errors and rejected hypotheses under the assumed null distribution \( Q_0 \).

For error rates \( \theta(F_{V_n}) \), defined as arbitrary parameters of the distribution of the number of Type I errors \( V_n \), we propose as null distribution the asymptotic distribution \( Q_0 \) of the vector of null value shifted and scaled test statistics \[ Z_n(m) \equiv \sqrt{\min \left( 1, \frac{\tau_0(m)}{\text{Var}[T_n(m)]} \right)} \left( T_n(m) + \lambda_0(m) - E[T_n(m)] \right). \] (1.10)

For the test of single-parameter null hypotheses using \( t \)-statistics, the null values are \( \lambda_0(m) = 0 \) and \( \tau_0(m) = 1 \). For testing the equality of \( K \) population means using \( F \)-statistics, the null values are \( \lambda_0(m) = 1 \) and \( \tau_0(m) = 2/(K-1) \), under the assumption of equal variances in the different populations. Dudoit et al. [6] and van der Laan et al. [7] prove that this null distribution does indeed provide the desired asymptotic control of the Type I error rate \( \theta(F_{V_n}) \), for general data generating distributions (with arbitrary dependence structures among variables), null hypotheses (defined in terms of submodels for the data generating distribution), and test statistics (e.g., \( t \)-statistics, \( F \)-statistics).

For a broad class of testing problems, such as the test of single-parameter null hypotheses using \( t \)-statistics (as in Equation (1.1)), the null distribution \( Q_0 \) is an \( M \)-variate Gaussian distribution with mean vector zero and covariance matrix \( \Sigma^*(P) \): \( Q_0 = Q_0(P) \equiv N(0, \Sigma^*(P)) \). For tests of means, where the parameter of interest is the \( M \)-dimensional mean vector \( \Psi(P) = \psi = E[X] \), the estimator \( \psi_n \) is simply the \( M \)-vector of sample averages and \( \Sigma^*(P) \) is the correlation matrix of \( X \sim P, \text{Cor}[X] \). More generally, for an asymptotically linear estimator \( \psi_n, \Sigma^*(P) \) is the correlation matrix of the vector influence curve (VC).

Note that the following important points distinguish our approach from existing approaches to Type I error rate control. Firstly, we are only concerned with Type I error control under the true data generating distribution \( P \). The notions of weak and strong control (and associated subset pivotality, Westfall & Young [9], p. 42–43) are therefore irrelevant to our approach.
Secondly, we propose a null distribution for the test statistics \( T_n \sim Q_0 \), and not a data generating null distribution \( X \sim P_0 \in \cap_{m=1}^{M} M(m) \). The latter practice does not necessarily provide proper Type I error control, as the test statistics' assumed null distribution \( Q_0(P_0) \) and their true distribution \( Q_0(P) \) may have different dependence structures (in the limit) for the true null hypotheses \( H_0 \).

**Bootstrap estimation of the test statistics null distribution.** In practice, since the data generating distribution \( P \) is unknown, then so is the proposed null distribution \( Q_0 = Q_0(P) \). Resampling procedures, such as bootstrap Procedure \( \mathbb{P} \) below, may be used to conveniently obtain consistent estimators \( Q_{0n} \) of the null distribution \( Q_0 \) and of the resulting test statistic cut-offs and adjusted \( p \)-values.
Procedure 1 [Bootstrap estimation of the null distribution $Q_0$]

1. Let $P^*_{n}$ denote an estimator of the data generating distribution $P$. For the non-parametric bootstrap, $P^*_{n}$ is simply the empirical distribution $P_n$, that is, samples of size $n$ are drawn at random, with replacement from the observed data $X_1, \ldots, X_n$. For the model-based bootstrap, $P^*_{n}$ is based on a model $M$ for the data generating distribution $P$, such as the family of $M$-variate Gaussian distributions.

2. Generate $B$ bootstrap samples, each consisting of $n$ i.i.d. realizations of a random variable $X^* \sim P^*_{n}$.

3. For the $b$th bootstrap sample, $b = 1, \ldots, B$, compute an $M$-vector of test statistics, $T^#_{n}(\cdot, b) = (T^#_{n}(m, b) : m = 1, \ldots, M)$. Arrange these bootstrap statistics in an $M \times B$ matrix, $T^#_{n} = (T^#_{n}(m, b))$, with rows corresponding to the $M$ null hypotheses and columns to the $B$ bootstrap samples.

4. Compute row means, $E[T^#_{n}(m, \cdot)]$, and row variances, $\text{Var}[T^#_{n}(m, \cdot)]$, of the matrix $T^#_{n}$, to yield estimates of the true means $E[T_n(m)]$ and variances $\text{Var}[T_n(m)]$ of the test statistics, respectively.

5. Obtain an $M \times B$ matrix, $Z^#_{n} = (Z^#_{n}(m, b))$, of null value shifted and scaled bootstrap statistics $Z^#_{n}(m, b)$, by row-shifting and scaling the matrix $T^#_{n}$ using the bootstrap estimates of $E[T_n(m)]$ and $\text{Var}[T_n(m)]$ and the user-supplied null values $\lambda_0(m)$ and $\tau_0(m)$. That is, compute

$$Z^#_{n}(m, b) \equiv \sqrt{\min\left(1, \frac{\tau_0(m)}{\text{Var}[T^#_{n}(m, \cdot)]}\right)} \times \left(T^#_{n}(m, b) + \lambda_0(m) - E[T^#_{n}(m, \cdot)]\right).$$

6. The bootstrap estimate $Q_{0n}$ of the null distribution $Q_0$ is the empirical distribution of the $B$ columns $Z^#_{n}(\cdot, b)$ of matrix $Z^#_{n}$.

Dudoit et al. [2] and van der Laan et al. [3] show that single-step and step-down procedures based on consistent estimators of the null distribution $Q_0$ also provide asymptotic control of the Type I error rate. The reader is referred to these two articles and to Dudoit & van der Laan [1] for details on the choice of null distribution and various approaches for estimating this null distribution.

Having selected a suitable test statistics null distribution, there remains the main task of specifying rejection regions for each null hypothesis, i.e., cut-offs for each test statistic. Among the different approaches for defining
rejection regions, we distinguish between single-step vs. stepwise procedures, and common cut-offs (i.e., the same cut-off $c_0$ is used for each test statistic) vs. common-quantile cut-offs (i.e., the cut-offs are the $\delta_0$-quantiles of the marginal null distributions of the test statistics). The next three subsections discuss three main approaches for deriving rejection regions and corresponding adjusted $p$-values: single-step common-cut-off and common-quantile procedures for control of general Type I error rates $\theta(F_{V_n})$ (Section 1.2.3); step-down common-cut-off (maxT) and common-quantile (minP) procedures for control of the FWER (Section 1.2.4); augmentation procedures for control of the gFWER and TPPFP, based on an initial FWER-controlling procedure (Section 1.2.5).

### 1.2.3 Single-step procedures for control of general Type I error rates $\theta(F_{V_n})$

Dudoit et al. [2] and Pollard & van der Laan [5] propose single-step common-cut-off and common-quantile procedures for controlling arbitrary parameters $\theta(F_{V_n})$ of the distribution of the number of Type I errors. The main idea is to substitute control of the parameter $\theta(F_{V_n})$, for the unknown, true distribution $F_{V_n}$ of the number of Type I errors, by control of the corresponding parameter $\theta(F_{R_0})$, for the known, null distribution $F_{R_0}$ of the number of rejected hypotheses. That is, consider single-step procedures of the form $R_n = \{m : T_n(m) > c_{n}(m)\}$, where the common cut-off $c_{n}(m)$ are chosen so that $\theta(F_{R_0}) \leq \alpha$, for $R_0 = \sum_{m=1}^{M} I(Z(m) > c_{n}(m))$ and $Z \sim Q_0$. Among the class of MTPs that satisfy $\theta(F_{R_0}) \leq \alpha$, Dudoit et al. [2] and Pollard & van der Laan [5] propose two procedures, based on common cut-offs and common-quantile cut-offs, respectively. The procedures are summarized below and the reader is referred to the articles for proofs and details on the derivation of cut-offs and adjusted $p$-values.

**Single-step common-cut-off procedure.** The set of rejected hypotheses for the $\theta$-controlling single-step common-cut-off procedure is of the form $R_n = \{m : T_n(m) > c_0(m)\}$, where the common cut-off $c_0$ is the smallest (i.e., least conservative) value for which $\theta(F_{R_0}) \leq \alpha$.

For $gFWER(k)$ control (special case $\theta(F_{V_n}) = 1 - F_{V_n}(k)$), the procedure is based on the $(k + 1)st$ ordered test statistic. Specifically, the adjusted $p$-values are given by

$$\tilde{p}_{0n}(m) = Pr_{Q_0}(Z^*(k+1) \geq t_n(m)), \quad m = 1, \ldots, M,$$

where $Z^*(m)$ denotes the $m$th ordered component of $Z = (Z(m) : m = 1, \ldots, M) \sim Q_0$, so that $Z^*(1) \geq \ldots \geq Z^*(M)$. For FWER control ($k = 0$), the procedure reduces to the single-step maxT procedure, based on the maximum test statistic, $Z^*(1)$. 
**Single-step common-quantile procedure.** The set of rejected hypotheses for the $\theta$–controlling single-step common-quantile procedure is of the form $R_n \equiv \{ m : T_n(m) > c_0(m) \}$, where $c_0(m) = Q_{0,m}^{-1}(\delta_0)$ is the $\delta_0$–quantile of the marginal null distribution $Q_{0,m}$ of the $m$th test statistic, i.e., the smallest value $c$ such that $Q_{0,m}(c) = Pr_{Q_0}(Z(m) \leq c) \geq \delta_0$ for $Z \sim Q_0$. Here, $\delta_0$ is chosen as the smallest (i.e., least conservative) value for which $\theta(F_{R_0}) \leq \alpha$.

For $gFWER(k)$ control, the procedure is based on the $(k + 1)$st ordered unadjusted $p$-value. Specifically, let $\bar{Q}_{0,m} \equiv 1 - Q_{0,m}$ denote the survivor functions for the marginal null distributions $Q_{0,m}$ and define unadjusted $p$-values $P_{0}(m) \equiv \bar{Q}_{0,m}(Z(m))$ and $P_{0n}(m) \equiv \bar{Q}_{0,n}(T_n(m))$, for $Z \sim Q_0$ and $T_n \sim Q_n$, respectively. Then, the adjusted $p$-values for the common-quantile procedure are given by

$$\tilde{p}_0(m) = Pr_{Q_0}(P_0^*(k + 1) \leq p_{0n}(m)), \quad m = 1, \ldots, M,$$

where $P_0^*(m)$ denotes the $m$th ordered component of the $M$–vector of unadjusted $p$-values $(P_0(m) : m = 1, \ldots, M)$, so that $P_0^*(1) \leq \ldots \leq P_0^*(M)$. For FWER control ($k = 0$), one recovers the single-step min$P$ procedure, based on the minimum unadjusted $p$-value, $P_0^*(1)$.

### 1.2.4 Step-down procedures for control of the family-wise error rate

van der Laan et al. [3] propose step-down common-cut-off (maxT) and common-quantile (minP) procedures for controlling the family-wise error rate, FWER. These procedures are similar in spirit to their single-step counterparts in Section 1.2.3 (special case $\theta(F_{V_n}) = 1 - F_{V_n}(0)$), with the important step-down distinction that hypotheses are considered successively, from most significant to least significant, with further tests depending on the outcome of earlier ones. That is, the test procedure is applied to a sequence of successively smaller nested random (i.e., data-dependent) subsets of null hypotheses, defined by the ordering of the test statistics (common cut-offs) or unadjusted $p$-values (common-quantile cut-offs).

**Step-down common-cut-off (maxT) procedure.** Rather than being based solely on the distribution of the maximum test statistic over all $M$ hypotheses, the step-down common cut-offs and corresponding adjusted $p$-values are based on the distributions of maxima of test statistics over successively smaller nested random subsets of null hypotheses. Specifically, let $O_n(m)$ denote the indices for the ordered test statistics $T_n(m)$, so that $T_n(O_n(1)) \geq \ldots \geq T_n(O_n(M))$. The step-down common-cut-off procedure is then based on the distributions of maxima of test statistics over the nested subsets of ordered hypotheses $\mathcal{O}_n(h) \equiv \{O_n(h), \ldots, O_n(M)\}$. The
adjusted p-values for the step-down maxT procedure are given by
\[ \tilde{p}_{0n}(o_n(m)) = \max_{h=1,\ldots,m} \left\{ P_{Q_0} \left( \max_{l \in \Theta_n(h)} Z_l \geq t_n(o_n(h)) \right) \right\}, \quad (1.14) \]
where \( Z = (Z(m) : m = 1, \ldots, M) \sim Q_0 \). Taking maxima of the probabilities over \( h \in \{1, \ldots, m\} \) enforces monotonicity of the adjusted p-values and ensures that the procedure is indeed step-down, that is, one can only reject a particular hypothesis provided all hypotheses with more significant (i.e., larger) test statistics were rejected beforehand.

**Step-down common-quantile (minP) procedure.** Likewise, the step-down common-quantile cut-offs and corresponding adjusted p-values are based on the distributions of minima of unadjusted p-values over successively smaller nested random subsets of null hypotheses. Specifically, let \( O_n(m) \) denote the indices for the ordered unadjusted p-values \( P_0^m(1) \leq \cdots \leq P_0^m(M) \). The step-down common-quantile procedure is then based on the distributions of minima of unadjusted p-values over the nested subsets of ordered hypotheses \( O_n(h) \equiv \{O_n(h), \ldots, O_n(M)\} \). The adjusted p-values for the step-down minP procedure are given by
\[ \tilde{p}_{0n}(o_n(m)) = \max_{h=1,\ldots,m} \left\{ P_{Q_0} \left( \min_{l \in \Theta_n(h)} P_0(l) \leq p_{0n}(o_n(h)) \right) \right\}, \quad (1.15) \]
where \( P_0(m) = Q_{0,m}(Z(m)) \) and \( Z = (Z(m) : m = 1, \ldots, M) \sim Q_0 \).

### 1.2.5 **Augmentation multiple testing procedures**

Dudoit & van der Laan [1] and van der Laan et al. [4] discuss augmentation multiple testing procedures (AMTP), obtained by adding suitably chosen null hypotheses to the set of null hypotheses already rejected by an initial MTP. Specifically, given any initial procedure controlling the generalized family-wise error rate, augmentation procedures are derived for controlling Type I error rates defined as tail probabilities and expected values for arbitrary functions \( g(V_n, R_n) \) of the numbers of Type I errors and rejected hypotheses (e.g., proportion \( g(V_n, R_n) = V_n / R_n \) of false positives among the rejected hypotheses). Adjusted p-values for the AMTP are shown to be simply shifted versions of the adjusted p-values of the original MTP. The important practical implication of these results is that any FWER-controlling MTP and its corresponding adjusted p-values, provide, without additional work, multiple testing procedures controlling a broad class of Type I error rates and their adjusted p-values. One can therefore build on the large pool of available FWER-controlling procedures, such as the single-step and step-down maxT and minP procedures discussed in Sections 1.2.3 and 1.2.4 above.
Augmentation procedures for controlling tail probabilities of the number (gFWER) and proportion (TPPFP) of false positives, based on an initial FWER-controlling procedure, are treated in detail in van der Laan et al. and are summarized below. The gFWER and TPPFP correspond to the special cases $g(V_n, R_n) = V_n$ and $g(V_n, R_n) = V_n/R_n$, respectively.

Denote the adjusted $p$-values for the initial FWER-controlling procedure by $\tilde{P}_0^+ (m)$. Order the $M$ null hypotheses according to these $p$-values, from smallest to largest, that is, define indices $O_n(m)$, so that $\tilde{P}_0^+ (O_n(1)) \leq \ldots \leq \tilde{P}_0^+ (O_n(M))$. Then, for a nominal level $\alpha$ test, the initial FWER-controlling procedure rejects the $R_n$ null hypotheses

$$R_n \equiv \{ m : \tilde{P}_0^+ (m) \leq \alpha \}.$$  

(1.16)

**Augmentation procedure for controlling the gFWER.** For control of gFWER($k$) at level $\alpha$, given an initial FWER-controlling procedure, reject the $R_n$ hypotheses specified by this MTP, as well as the next $A_n = \min\{k, M - R_n\}$ most significant null hypotheses. The adjusted $p$-values $\tilde{P}_0^+ (O_n(m))$ for the new gFWER-controlling AMTP are simply $k$-shifted versions of the adjusted $p$-values of the initial FWER-controlling MTP:

$$\tilde{P}_0^+ (O_n(m)) = \begin{cases} 0, & \text{if } m = 1, \ldots, k; \\ \tilde{P}_0^+ (O_n(m - k)), & \text{if } m = k + 1, \ldots, M. \end{cases} \quad (1.17)$$

That is, the first $k$ adjusted $p$-values are set to zero and the remaining $p$-values are the adjusted $p$-values of the FWER-controlling MTP shifted by $k$. The AMTP thus guarantees at least $k$ rejected hypotheses.

**Augmentation procedure for controlling the TPPFP.** For control of TPPFP($q$) at level $\alpha$, given an initial FWER-controlling procedure, reject the $R_n$ hypotheses specified by this MTP, as well as the next $A_n$ most significant null hypotheses,

$$A_n = \max \left\{ m \in \{0, \ldots, M - R_n\} : \frac{m}{m + R_n} \leq q \right\} = \min \left\{ \left\lfloor \frac{q R_n}{1 - q} \right\rfloor, M - R_n \right\}, \quad (1.18)$$

where the floor $\lfloor x \rfloor$ denotes the greatest integer less than or equal to $x$, i.e., $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$. That is, keep rejecting null hypotheses until the ratio of additional rejections to the total number of rejections reaches the allowed proportion $q$ of false positives. The adjusted $p$-values $\tilde{P}_0^+ (O_n(m))$ for the new TPPFP-controlling AMTP are simply shifted versions of the adjusted $p$-values of the initial FWER-controlling MTP, that is,

$$\tilde{P}_0^+ (O_n(m)) = \tilde{P}_0^+ (O_n([ (1 - q)m ])), \quad m = 1, \ldots, M, \quad (1.19)$$
where the ceiling $\lceil x \rceil$ denotes the least integer greater than or equal to $x$, i.e., $\lceil x \rceil - 1 < x \leq \lceil x \rceil$.

**FDR-controlling procedures.** Given any TPPFP-controlling procedure, van der Laan et al. [4] derive two simple (conservative) FDR-controlling procedures. The more general and conservative procedure controls the FDR at nominal level $\alpha$, by controlling $TPPFP(\alpha/2)$ at level $\alpha/2$. The less conservative procedure controls the FDR at nominal level $\alpha$, by controlling $TPPFP(1 - \sqrt{1 - \alpha})$ at level $1 - \sqrt{1 - \alpha}$. In what follows, we refer to these two MTPs as "conservative" and "restricted", respectively. The reader is referred to the original article for details and proofs of FDR control (Section 2.4, Theorem 3).

1.3 Software implementation: *multtest* package

1.3.1 Overview

The MTPs proposed in Sections 1.2.3–1.2.5 are implemented in the latest version of the Bioconductor R package *multtest* (version 1.5.0, Bioconductor release 1.5). New features include: expanded class of tests (e.g., for regression parameters in linear models and in Cox proportional hazards models); control of a wider selection of Type I error rates (e.g., gFWER, TPPFP, FDR); bootstrap estimation of the test statistics null distribution; augmentation multiple testing procedures; confidence regions for the parameter vector of interest. Because of their general applicability and novelty, we focus in this section on MTPs that utilize a bootstrap estimated test statistics null distribution and that are available through the package’s main user-level function: MTP. Note that for many testing problems, MTPs based on permutation (rather than bootstrap) estimated null distributions are also available in the present and earlier versions of *multtest*. In particular, permutation-based step-down maxT and minP FWER-controlling MTPs are implemented in the functions mt.maxT and mt.minP, respectively, and can also be applied directly through a call to the MTP function.

We stress that all the bootstrap-based MTPs implemented in *multtest* can be performed using the main user-level function MTP. Most users will therefore only need to be familiar with this function. Other functions are provided primarily for the benefit of more advanced users, interested in extending the package’s functionality (Section 1.3.4). For greater detail on *multtest* functions, the reader is referred to the package documentation, in the form of help files, e.g., ?MTP, and vignettes, e.g., openVignette("multtest").

One needs to specify the following main ingredients when applying a MTP: the data, $X_1, \ldots, X_n$; suitably defined test statistics, $T_n$, for each of the null hypotheses under consideration (e.g., one-sample $t$-statistics, re-
bust rank-based F-statistics, t-statistics for regression coefficients in Cox proportional hazards model); a choice of Type I error rate, \( \theta(F_{\nu_n}, R_n) \), providing an appropriate measure of false positives for the particular testing problem (e.g., \( TPPFP(0.10) \)); a proper joint null distribution, \( Q_0 \) (or estimate thereof, \( Q_{0n} \)), for the test statistics (e.g., bootstrap null distribution as in Procedure 1); given the previously defined components, a multiple testing procedure, \( R_n = R(T_n, Q_{0n}, \alpha) \), for controlling the error rate \( \theta(F_{\nu_n}, R_n) \) at a target level \( \alpha \). Accordingly, the multtest package has adopted a modular and extensible approach to the implementation of MTPs, with the following four main types of functions.

- Functions for computing the test statistics, \( T_n \). These are internal functions (e.g., meanX, coxY), i.e., functions that are generally not called directly by the user. As shown in Section 1.3.2 below, the type of test statistic is specified by the test argument of the main user-level function MTP. Advanced users, interested in extending the class of tests available in multtest, can simply add their own test statistic functions to the existing library of such internal functions (see Section 1.3.4 below, for a brief discussion of the closure approach for specifying test statistics).

- Functions for obtaining the test statistics null distribution, \( Q_0 \), or an estimate thereof, \( Q_{0n} \). The main function currently available is the internal function boot.resample, implementing the non-parametric version of bootstrap Procedure 1 (Section 1.2.2).

- Functions for implementing the multiple testing procedure, \( R(T_n, Q_{0n}, \alpha) \), i.e., for deriving rejection regions, confidence regions, and adjusted \( p \)-values. The main function is the user-level wrapper function MTP, which implements the single-step and step-down maxT and minP procedures for FWER control (Sections 1.2.3 and 1.2.4). The functions fwer2gfwerr, fwer2tppfr, and fwer2fdr implement, respectively, gFWER-, TPPFP-, and FDR-controlling augmentation multiple testing procedures, based on adjusted \( p \)-values from any FWER-controlling procedure, and can be called via the typeone argument to MTP (Section 1.2.5).

- Functions for numerical and graphical summaries of a MTP. As described in Section 1.3.3 below, a number of summary methods are available to operate on objects of class MTP, output from the main MTP function.

1.3.2 Resampling-based multiple testing procedures: MTP function

The main user-level function for resampling-based multiple testing is MTP. Its input/output and usage are described next.
1. Multiple Testing Procedures and Applications to Genomics

> library(Biobase)
> library(multtest)
> library(genefilter)

> args(MTP)

function (X, W = NULL, Y = NULL, Z = NULL, Z.incl = NULL, Z.test = NULL,
    na.rm = TRUE, test = "t.twosamp.unequalvar", robust = FALSE,
    standardize = TRUE, alternative = "two.sided", psi0 = 0,
    typeone = "fwer", k = 0, q = 0.1, fdr.method = "conservative",
    alpha = 0.05, nulldist = "boot", B = 1000, method = "ss.maxT",
    get.cr = FALSE, get.cutoff = FALSE, get.adjp = TRUE, keep.nulldist = FALSE,
    seed = NULL)

NULL

INPUT.

Data. The data, \(X\), consist of a \(J\)-dimensional random vector, observed on
each of \(n\) sampling units (patients, cell lines, mice, etc). These data
can be stored in a \(J \times n\) matrix, data.frame, or exprs slot of an object
of class exprSet *** REF: Chapter on Biobase introducing exprSet.
In some settings, a \(J\)-vector of weights may be associated with each
observation, and stored in a \(J \times n\) weight matrix, \(W\) (or an \(n\)-vector \(W\),
if the weights are the same for each of the \(J\) variables). One may also
observe a possibly censored continuous or polychotomous outcome,
\(Y\), for each sampling unit, as obtained, for example, from the pheno-
Data slot of an object of class exprSet. In some studies, \(L\) additional
covariates may be measured on each sampling unit and stored in \(Z\),
an \(n \times L\) matrix or data.frame. When the tests concern parameters
in regression models with covariates from \(Z\) (e.g., values \texttt{lm.XvsZ},
\texttt{lm.YvsXZ}, and \texttt{coxph.YvsXZ}, for the argument test, described below),
the arguments \(Z.incl\) and \(Z.test\) specify, respectively, which
covariates (i.e., which columns of \(Z\), including \(Z.test\)) should be included
in the model and which regression parameter is to be tested
(only when \texttt{test="lm.XvsZ"}). The covariates can be specified either
by a numeric column index or character string. If \(X\) is an instance of
the class exprSet, \(Y\) can be a column index or character string referring
to the variable in the data.frame \texttt{pData(X)} to use as outcome. Likewise, \(Z.incl\) and \(Z.test\) can be column indices or character strings
referring to the variables in \texttt{pData(X)} to use as covariates. The data
components (\(X, W, Y, Z, Z.incl,\) and \(Z.test\)) are the first six arguments
to the MTP function. Only \(X\) is a required argument; the others
are by default NULL. The argument \texttt{na.rm} allows one to control the
treatment of "Not Available" or NA values. It is set to TRUE, by default,
so that an observation with a missing value in any of the data
objects’ \(j\)th component (\(j = 1, \ldots, J\)) is excluded from computation
of any of the relevant test statistics.
Test statistics.
The test statistics should be chosen based on the parameter of interest (e.g., location, scale, or regression parameters) and the hypotheses one wishes to test. In the current implementation of `multtest`, the following test statistics are available through the argument `test`, with default value `t.twosamp.unequalvar`, for the two-sample Welch $t$-statistic.

- `t.onesamp`: One-sample $t$-statistic for tests of means.
- `t.twosamp.equalvar`: Equal variance two-sample $t$-statistic for tests of differences in means.
- `t.twosamp.unequalvar`: Unequal variance two-sample $t$-statistic for tests of differences in means (also known as two-sample Welch $t$-statistic).
- `t.pair`: Two-sample paired $t$-statistic for tests of differences in means.
- `f`: Multi-sample $F$-statistic for tests of equality of population means.
- `f.block`: Multi-sample $F$-statistic for tests of equality of population means in a block design.
- `lm.XvsZ`: $t$-statistic for tests of regression coefficients for variable $Z$.test in linear models each with outcome $X[j,1]$ ($j = 1, \ldots, J$), and possibly additional covariates $Z.incl$ from the matrix $Z$ (in the case of no covariates, one recovers the one-sample $t$-statistic, `t.onesamp`).
- `lm.YvsXZ`: $t$-statistic for tests of regression coefficients in linear models with outcome $Y$ and each $X[j,1]$ ($j = 1, \ldots, J$) as covariate of interest, with possibly other covariates $Z.incl$ from the matrix $Z$.
- `coxph.YvsXZ`: $t$-statistic for tests of regression coefficients in Cox proportional hazards survival models with outcome $Y$ and each $X[j,1]$ ($j = 1, \ldots, J$) as covariate of interest, with possibly other covariates $Z.incl$ from the matrix $Z$.

Robust, rank-based versions of the above test statistics can be specified by setting the argument `robust` to `TRUE` (the default value is `FALSE`). Consideration should be given to whether `standardized` (Equation 1.1) or `unstandardized` difference statistics are most appropriate (see Pollard & van der Laan [5] for a comparison). Both options are available through the argument `standardize`, by default `TRUE`. The type of alternative hypotheses is specified via the `alternative` argument: default value of `two.sided`, for two-sided test, and values of `less` or `greater`, for one-sided tests. The (common) null value for the parameters of interest is specified through the `psi0` argument, by default zero.
**Type I error rate.** The \texttt{MTP} function controls by default the family-wise error rate (FWER), or chance of at least one false positive (argument \texttt{typeone=“fwer”}). Augmentation procedures (Section 1.2.5), controlling other Type I error rates such as the gFWER, TPPFP, and FDR, can be specified through the argument \texttt{typeone}. Related arguments include \texttt{k} and \texttt{q}, for the allowed number and proportion of false positives for control of gFWER(\texttt{k}) and TPPFP(\texttt{q}), respectively, and \texttt{fdr.method}, for the type of TPPFP-based FDR-controlling procedure (i.e., "conservative" or "restricted" methods). The nominal level of the test is determined by the argument \texttt{alpha}, by default 0.05. Testing can be performed for a range of nominal Type I error rates by specifying a vector of levels \texttt{alpha}.

**Test statistics null distribution.** In the current implementation of \texttt{MTP}, the test statistics null distribution is estimated by default using the non-parametric version of bootstrap Procedure 1 (argument \texttt{nulldist=“boot”}). The bootstrap procedure is implemented in the internal function \texttt{boot.resample}, which calls C to compute test statistics for each bootstrap sample. The values of the shift (\(\lambda_0\)) and scale (\(\tau_0\)) parameters are determined by the type of test statistics (e.g., \(\lambda_0 = 0\) and \(\tau_0 = 1\) for \(t\)-statistics). Permutation null distributions are also available via \texttt{nulldist=“perm”}. The number of resampling steps is specified by the argument \texttt{B}, by default 1,000.

**Multiple testing procedures.** Several methods for controlling the chosen Type I error rate are available in \texttt{multtest}.

- **FWER-controlling procedures.** For FWER control, the \texttt{MTP} function implements the single-step and step-down (common-cut-off) maxT and (common-quantile) minP MTPs, described in Sections 1.2.3 and 1.2.4 and specified through the argument \texttt{method} (internal functions \texttt{ss.maxT}, \texttt{ss.minP}, \texttt{sd.maxT}, and \texttt{sd.minP}). The default MTP is the single-step maxT procedure (\texttt{method=“ss.maxT”}), since it requires the least computation.
- **gFWER-, TPPFP-, and FDR-controlling augmentation procedures.** As discussed in Section 1.2.5 any FWER-controlling MTP can be trivially augmented to control additional Type I error rates, such as the gFWER and TPPFP. Two FDR-controlling procedures can then be derived from the TPPFP-controlling AMTP. The AMTPs are implemented in the functions \texttt{fwer2gfwer}, \texttt{fwer2tpfp}, and \texttt{fwer2fdr}, that take FWER adjusted \(p\)-values as input and return augmentation adjusted \(p\)-values for control of the gFWER, TPPFP, and FDR, respectively. Note that the aforementioned AMTPs can be applied directly via the \texttt{typeone} argument of the main function \texttt{MTP}.
Output control. Various arguments are available to control output, i.e., specify which combination of the following quantities should be returned: confidence regions (argument `get.cr`); cut-offs for the test statistics (argument `get.cutoff`); adjusted p-values (argument `get.adjp`); test statistics null distribution (argument `keep.nulldist`). Note that parameter estimates and confidence regions only apply to the test of single-parameter null hypotheses (i.e., not the F-tests). In addition, in the current implementation of `MTP`, parameter confidence regions and test statistic cut-offs are only provided when `typeone="fwer"`, so that `get.cr` and `get.cutoff` should be set to `FALSE` when using the error rates gFWER, TPPFP, or FDR.

Note that the `multtest` package also provides several simple, marginal FWER-controlling MTPs, such as the Bonferroni, Holm [10], Hochberg [11], and Šidák [12] procedures, and FDR-controlling MTPs, such as the Benjamini & Hochberg [8] and Benjamini & Yekutieli [13] procedures. These procedures are available through the `mt.rawp2adjp` function, which takes a vector of unadjusted p-values as input and returns the corresponding adjusted p-values.

OUTPUT.

The S4 class/method object-oriented programming approach was adopted to summarize the results of a MTP (Section 1.3.4). Specifically, the output of the `MTP` function is an instance of the class `MTP`. A brief description of the class and associated methods is given next. Please consult the documentation for details, e.g., using `class ? MTP` and `methods ? MTP`.

```r
> slotNames("MTP")

[1] "statistic" "estimate" "sampsizes" "rawp" "adjp" "conf.reg"
[7] "cutoff" "reject" "nulldist" "call" "seed"
```

**statistic:** The numeric $M$–vector of test statistics, specified by the values of the MTP arguments `test`, `robust`, `standardize`, and `psi0`. In many testing problems, $M = J = \text{nrow}(X)$.

**estimate:** For the test of single-parameter null hypotheses using $t$-statistics (i.e., not the $F$-tests), the numeric $M$–vector of estimated parameters.

**sampsizes:** The sample size, i.e., $n = \text{ncol}(X)$.

**rawp:** The numeric $M$–vector of unadjusted p-values.

**adjp:** The numeric $M$–vector of adjusted p-values (computed only if the `get.adjp` argument is `TRUE`).

**conf.reg:** For the test of single-parameter null hypotheses using $t$-statistics (i.e., not the $F$-tests), the numeric $M \times 2 \times \text{length}(\alpha)$
array of lower and upper simultaneous confidence limits for the parameter vector, for each value of the nominal Type I error rate \( \alpha \) (computed only if the get.cr argument is TRUE).

cutoff: The numeric \( M \times \text{length}(\alpha) \) matrix of cut-offs for the test statistics, for each value of the nominal Type I error rate \( \alpha \) (computed only if the get.cutoff argument is TRUE).

reject: The \( M \times \text{length}(\alpha) \) matrix of rejection indicators (TRUE for a rejected null hypothesis), for each value of the nominal Type I error rate \( \alpha \).

nulldist: The numeric \( M \times B \) matrix for the estimated test statistics null distribution (returned only if keep.nulldist=TRUE; option not currently available for permutation null distribution, i.e., \text{nulldist}="perm"). By default (i.e., for \text{nulldist}="boot"), the entries of nulldist are the null value shifted and scaled bootstrap test statistics, as defined by Procedure I.

call: The call to the function \text{MTP}.

seed: An integer for specifying the state of the random number generator used to create the resampled datasets. The seed can be reused for reproducibility in a repeat call to \text{MTP}. This argument is currently used only for the bootstrap null distribution (i.e., for \text{nulldist}="boot").  

See \texttt{? set.seed} for details.

### 1.3.3 Numerical and graphical summaries

The following methods are defined to operate on \text{MTP} instances and summarize the results of a MTP.

print: The print method returns a description of an object of class \text{MTP}, including the sample size \( n \), the number \( M \) of tested hypotheses, the type of test performed (value of argument test), the Type I error rate (value of argument typeone), the nominal level of the test (value of argument \( \alpha \)), the name of the MTP (value of argument method), the call to the function \text{MTP}. In addition, this method produces a table with the class, mode, length, and dimension of each slot of the \text{MTP} instance.

summary: The summary method provides numerical summaries of the results of a MTP and returns a list with the following three components.

- rejections: A \text{data.frame} with the number(s) of rejected hypotheses for the nominal Type I error rate(s) specified by the \text{alpha} argument of the function \text{MTP} (NULL values are returned...
if all three arguments \texttt{get.cr}, \texttt{get.cutoff}, and \texttt{get.adjp} are \texttt{FALSE}).

- **index**: A numeric $M$–vector of indices for ordering the hypotheses according to first \texttt{adjp}, then \texttt{rawp}, and finally the absolute value of \texttt{statistic} (not printed in the summary).

- **summaries**: When applicable (i.e., when the corresponding quantities are returned by \texttt{MTP}), a table with six number summaries of the distributions of the adjusted $p$-values, unadjusted $p$-values, test statistics, and parameter estimates.

**plot**: The \texttt{plot} method produces the following graphical summaries of the results of a MTP. The type of display may be specified via the \texttt{which} argument.

1. Scatterplot of number of rejected hypotheses vs. nominal Type I error rate.
2. Plot of ordered adjusted $p$-values; can be viewed as a plot of Type I error rate vs. number of rejected hypotheses.
3. Scatterplot of adjusted $p$-values vs. test statistics (also known as “volcano plot”).
4. Plot of unordered adjusted $p$-values.
5. Plot of confidence regions for user-specified parameters, by default the 10 parameters corresponding to the smallest adjusted $p$-values (argument \texttt{top}).
6. Plot of test statistics and corresponding cut-offs (for each value of \texttt{alpha}) for user-specified hypotheses, by default the 10 hypotheses corresponding to the smallest adjusted $p$-values (argument \texttt{top}).

The argument \texttt{logscale} (by default equal to \texttt{FALSE}) allows one to use the negative decimal logarithms of the adjusted $p$-values in the second, third, and fourth graphical displays. Note that some of these plots are implemented in the older function \texttt{mt.plot}.

[: Subsetting method, which operates selectively on each slot of an \texttt{MTP} instance to retain only the data related to the specified hypotheses.

\texttt{as.list}: Converts an object of class \texttt{MTP} to an object of class \texttt{list}, with an entry for each slot.

### 1.3.4 Software design

The following features of the programming approach employed in \texttt{multtest} may be of interest to users, especially those interested in extending the functionality of the package.

**Function closures.** The use of \textit{function closures}, in the style of the \texttt{gene-filter} package, allows uniform data input for all MTPs and facilitates the
extension of the package’s functionality by adding, for example, new types of test statistics. Specifically, for each value of the MTP argument test, a closure is defined which consists of a function for computing the test statistic (with only two arguments, a data vector x and a corresponding weight vector w, with default value of NULL) and its enclosing environment, with bindings for relevant additional arguments, such as null values psi0, outcomes Y, and covariates Z. Thus, new test statistics can be added to multtest by simply defining a new closure and adding a corresponding value for the test argument to MTP (existing internal test statistic functions are located in the file R/statistics.R).

Class/method object-oriented programming. Like many other Bioconductor packages, multtest has adopted the S4 class/method object-oriented programming approach of Chambers [14]. In particular, a new class, MTP, is defined to represent the results of multiple testing procedures, as implemented in the main MTP function. As discussed above, in Section 1.3.3, several methods are provided to operate on instances of this class.

Calls to C. Because resampling procedures, such as the non-parametric bootstrap implemented in multtest, are computationally intensive, care must be taken to ensure that the resampling steps are not prohibitively slow. The use of closures for the test statistics, however, prevents writing the entire program in C. In the current implementation, we have chosen to define the closure and compute the observed test statistics in R, and then call C (using the R random number generator) to apply the closure to each bootstrap resampled dataset. This approach puts the for loops over bootstrap samples (B) and hypotheses (M) in the C environment, thus speeding up this computationally expensive part of the program. Further optimization for speed may be investigated for future releases.

1.4 Applications: ALL microarray dataset

1.4.1 ALL data package and initial gene filtering

We illustrate some of the functionality of the multtest package using the Acute Lymphoblastic Leukemia (ALL) microarray dataset of Chiaretti et al. [6], available in the data package ALL. This dataset consists of 21 phenotypes (i.e., patient level responses and covariates) and 12,625 Affymetrix gene expression measures (chip series HGU95Av2), for each of 128 ALL patients. For greater detail, please consult the ALL package documentation. The main object in this package is ALL, an instance of the class exprSet, which contains the expression measures, phenotypes, and gene annotation information. The genes-by-subjects matrix of expression measures is provided in the exprs slot of ALL and the phenotype data are stored in the
phenoData slot. Note that the expression measures have been obtained using the three-step robust multichip average (RMA) pre-processing method, implemented in the package *affy*. In particular, the expression measures have been subject to a base 2 logarithmic transformation.

```r
> library(ALL)
> library(hgu95av2)
> data(ALL)
> class(ALL)
[1] "exprSet"
attr(,"package")
[1] "Biobase"

> slotNames(ALL)
[1] "exprs"  "se.exprs"  "phenoData"  "description"  "annotation"
[6] "notes"

> show(ALL)

Expression Set (exprSet) with
12625 genes
128 samples
phenoData object with 21 variables and 128 cases

varLabels
  cod: Patient ID
  diagnosis: Date of diagnosis
  sex: Gender of the patient
  age: Age of the patient at entry
  BT: does the patient have B-cell or T-cell ALL
  remission: Complete remission(CR), refractory(REF) or NA. Derived from CR
  CR: Original remission data
  date.cr: Date complete remission if achieved
  t(4;11): did the patient have t(4;11) translocation. Derived from citog
  t(9;22): did the patient have t(9;22) translocation. Derived from citog
  cyto.normal: Was cytogenetic test normal? Derived from citog
  citog: original citogenetics data, deletions or t(4;11), t(9;22) status
  mol.biol: molecular biology
  fusion protein: which of p190, p210 or p190/210 for bcr/able
  mdr: multi-drug resistant
  kinet: ploidy: either diploid or hyperd.
  ccr: Continuous complete remission? Derived from f.u
  relapse: Relapse? Derived from f.u
  transplant: did the patient receive a bone marrow transplant? Derived from f.u
  f.u.: follow up data available
  date last seen: date patient was last seen
> names(varLabels(ALL))

```
[1] "cod"       "diagnosis"  "sex"       "age"
[5] "BT"        "remission"  "CR"        "date.cr"
[9] "t(4;11)"   "t(9;22)"    "cyto.normal" "citog"
[13] "mol.biol"  "fusion protein" "mdr"     "kinet"
[17] "ccr"       "relapse"    "transplant" "f.u"
[21] "date last seen"
```

> X <- exprs(ALL)
> pheno <- pData(ALL)

Our goal is to identify genes whose expression measures are associated with (possibly censored) biological and clinical outcomes such as: cytogenetic test status (normal vs. abnormal), tumor molecular subtype (BCR/ABL, NEG, ALL1/AF4, E2A/PBX1, p15/p16, NUP-98), and time to relapse. However, before applying the multiple testing procedures discussed in Section 1.2, above, we perform initial gene filtering as in Chiaretti et al. [6] and retain only those genes for which (i) at least 20% of the subjects have a measured intensity of at least 100 and (ii) the coefficient of variation (i.e., the ratio of the standard deviation to the mean) of the intensities across samples is between 0.7 and 10. These two filtering criteria can be readily applied using functions from the `genefilter` package.

```r
> ffun <- filterfun(pOverA(p = 0.2, A = 100), cv(a = 0.7, b = 10))
> filt <- genefilter(2^X, ffun)
> filtX <- X[filt, ]
> dim(filtX)
[1] 431 128
> filtALL <- ALL[filt, ]
```

1.4.2 Association of expression measures and cytogenetic test status: two-sample t-statistics

**Step-down minP FWER-controlling MTP with two-sample Welch t-statistics and bootstrap null distribution**

The phenotype data include an indicator variable, `cyto.normal`, for cytogenetic test status (1 for normal vs. 0 for abnormal). To identify genes with higher mean expression measures in the abnormal compared to the normal cytogenetics subjects, one-sided two-sample t-tests can be performed. We choose to use the Welch t-statistic and to control the FWER using the bootstrap-based step-down minP procedure with \( B = 10000 \) bootstrap iterations.
> seed <- 99
> cyto.boot <- MTP(X = filtALL, Y = "cyto.normal", alternative = "less",
+    B = 10000, method = "sd.minP", seed = seed)

Let us examine the results of the MTP stored in the object cyto.boot.

> class(cyto.boot)
[1] "MTP"
attr(.,"package")
[1] "multtest"

> slotNames(cyto.boot)
[1] "statistic" "estimate" "sampsize" "rawp" "adjp" "conf.reg"
[7] "cutoff" "reject" "nulldist" "call" "seed"

> print(cyto.boot)

Multiple Testing Procedure

Object of class: MTP
sample size = 128
number of hypotheses = 431

test statistics = t.twosamp.unequalvar
type I error rate = fwer
nominal level alpha = 0.05
multiple testing procedure = sd.minP

Call: MTP(X = filtALL, Y = "cyto.normal", alternative = "less", B = 10000,
    method = "sd.minP", seed = seed)

Slots:

<table>
<thead>
<tr>
<th>Class</th>
<th>Mode</th>
<th>Length</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>statistic</td>
<td>numeric</td>
<td>numeric</td>
<td>431</td>
</tr>
<tr>
<td>estimate</td>
<td>numeric</td>
<td>numeric</td>
<td>431</td>
</tr>
<tr>
<td>sampsize</td>
<td>numeric</td>
<td>numeric</td>
<td>1</td>
</tr>
<tr>
<td>rawp</td>
<td>numeric</td>
<td>numeric</td>
<td>431</td>
</tr>
<tr>
<td>adjp</td>
<td>numeric</td>
<td>numeric</td>
<td>431</td>
</tr>
<tr>
<td>conf.reg</td>
<td>array</td>
<td>logical</td>
<td>0 0,0,0</td>
</tr>
<tr>
<td>cutoff</td>
<td>matrix</td>
<td>logical</td>
<td>0 0,0</td>
</tr>
<tr>
<td>reject</td>
<td>matrix</td>
<td>logical</td>
<td>431,1</td>
</tr>
<tr>
<td>nulldist</td>
<td>matrix</td>
<td>logical</td>
<td>0 0,0</td>
</tr>
<tr>
<td>call</td>
<td>call</td>
<td>call</td>
<td>7</td>
</tr>
<tr>
<td>seed</td>
<td>integer</td>
<td>numeric</td>
<td>626</td>
</tr>
</tbody>
</table>

> summary(cyto.boot)
MTP: sd.minP
Type I error rate: fwer

Level Rejections
1 0.05 15

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.922</td>
<td>-1.083</td>
</tr>
<tr>
<td>-0.7589</td>
<td>-0.2015</td>
</tr>
<tr>
<td>-0.03815</td>
<td>-0.008719</td>
</tr>
<tr>
<td>-0.1192</td>
<td>-0.03249</td>
</tr>
<tr>
<td>0.5946</td>
<td>0.1467</td>
</tr>
<tr>
<td>2.5620</td>
<td>0.6709</td>
</tr>
</tbody>
</table>

The following commands may be used to obtain a list of genes that are differentially expressed in normal vs. abnormal cytogenetics patients at nominal FWER level $\alpha = 0.05$, i.e., genes with adjusted $p$-values less than or equal to 0.05. Functions from the annotate and annaffy packages may then be used to obtain annotation information on these genes (e.g., gene names, PubMed abstracts, GO terms) and to generate HTML tables of the results. *** REF: Chapter on annotation.

```r
> cyto.diff <- cyto.boot@adjp <= 0.05
> sum(cyto.diff)
[1] 15
> cyto.AffyID <- geneNames(filtALL)[cyto.diff]
> mget(cyto.AffyID, env = hgu95av2GENENAME)

$"1433_g_at"
[1] " MAD, mothers against decapentaplegic homolog 3 (Drosophila)"

$"32542_at"
[1] " four and a half LIM domains 1"

$"32562_at"
[1] " endoglin (Osler-Rendu-Weber syndrome 1)"

$"33232_at"
[1] " cysteine-rich protein 1 (intestinal)"

$"33284_at"
[1] " myeloperoxidase"

$"33891_at"
[1] " chloride intracellular channel 4"

$"37539_at"
```
Various graphical summaries of the results may be obtained using the `plot` method, by selecting appropriate values of the argument `which` (Figure [1.1]).

```r
> par(mfrow = c(2, 2))
> plot(cyto.boot)
```

**Marginal FWER-controlling MTPs with two-sample Welch t-statistics and bootstrap null distribution**

Given a vector of unadjusted \( p \)-values, the `mt.rawp2adjp` function computes adjusted \( p \)-values for the marginal FWER-controlling MTPs of Bonferroni, Holm [10], Hochberg [11], and Šidák [12], discussed in detail in Dudoit et al. [15]. The `mt.plot` function may then be used to compare the different procedures in terms of their adjusted \( p \)-values.

```r
> marg <- c("Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD")
> cyto.marg <- mt.rawp2adjp(rawp = cyto.boot@rawp, proc = marg)
> comp.marg <- cbind(cyto.boot@adjp, cyto.marg$adjp[order(cyto.marg$index), +   -1])
```
Figure 1.1. Cytogenetic test status — Step-down minP FWER-controlling MTP. By default, four graphical summaries are produced by the \texttt{plot} method for instances of the class \texttt{MTP}.

```r
> par(mfrow = c(1, 1))
> mt.plot(adjp = comp.marg, teststat = cyto.boot@statistic, proc = c("SD minP", + marg), leg = c(0.1, 400), col = 1:6, lty = 1:6, lwd = 3)
> title("Comparison of marginal and step-down minP FWER-controlling MTPs")
```

In this dataset, most of the FWER-controlling MTPs perform similarly, making very few rejections at nominal Type I error rates near zero. As expected, the bootstrap-based step-down minP procedure, which takes into account the joint distribution of the test statistics, leads to slightly more rejections than the marginal methods (Figure 1.2). The results also illustrate that stepwise MTPs are less conservative than their single-step analogues (e.g., Holm and Hochberg vs. Bonferroni; step-down Šidák vs. single-step Šidák).

\textit{Step-down minP FWER-controlling MTP with two-sample Welch t-statistics and permutation null distribution}

Because the sample sizes are not equal for the two cytogenetic groups and the expression measures may have different covariance structures in the two populations, we expect the bootstrap and permutation null distributions to yield different sets of rejected hypotheses (Pollard & van der Laan \cite{5}). To compare the two approaches, we apply the permutation-based step-down minP procedure, first using the old \texttt{mt.minP} function and then using the
new MTP function (which calls mt.minP). Please note that while the MTP and mt.minP functions produce the same results, these are presented in a different manner. In particular, for the new function MTP, the results (e.g., test statistics, parameter estimates, unadjusted $p$-values, adjusted $p$-values, cut-offs) are given in the original order of the null hypotheses, while in the mt.minP function, the hypotheses are sorted first according to their adjusted $p$-values, next their unadjusted $p$-values, and finally their test statistics. In addition, the new function MTP implements a broader range of MTPs and has adopted the S4 class/method design for representing and summarizing the results of a MTP.
> summary(cyto.perm.new)

MTP: sd.minP
Type I error rate: fwer

Level Rejections
1 0.05 0

<table>
<thead>
<tr>
<th>Min. 1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>adjp</td>
<td>0.5360</td>
<td>1.0000</td>
<td>1.00000</td>
<td>0.9918</td>
</tr>
<tr>
<td>rawp</td>
<td>0.0036</td>
<td>0.2316</td>
<td>0.48980</td>
<td>0.4840</td>
</tr>
<tr>
<td>statistic</td>
<td>-2.9220</td>
<td>-0.7589</td>
<td>-0.03815</td>
<td>-0.1192</td>
</tr>
<tr>
<td>estimate</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NaN</td>
</tr>
</tbody>
</table>

> sum(cyto.perm.new@adjp <= 0.05)
[1] 0

> sum(cyto.perm.new@adjp <= 0.05 & cyto.boot@adjp <= 0.05)
[1] 0

At nominal FWER level $\alpha = 0.05$, the permutation step-down minP procedure identifies 0 genes as differentially expressed between patients with normal and abnormal cytogenetic test status. In contrast, the bootstrap version of the step-down minP procedure identifies 15 differentially expressed genes.

*Step-down minP FWER-controlling MTP with robust two-sample t-statistics and bootstrap null distribution*

The Wilcoxon rank sum statistic (also known as the Mann-Whitney statistic) is a robust alternative to the usual two-sample $t$-statistic. Note that only $B = 1000$ bootstrap iterations are used in this example, although more are recommended in practice.

> cyto.wilcox <- MTP(X = filtALL, Y = "cyto.normal", robust = TRUE, + alternative = "less", B = 1000, method = "sd.minP", seed = seed)

> sum(cyto.wilcox@adjp <= 0.05)
[1] 16

> sum(cyto.wilcox@adjp <= 0.05 & cyto.boot@adjp <= 0.05)
[1] 10

At nominal FWER level $\alpha = 0.05$, the bootstrap step-down minP MTP based on the robust Wilcoxon test statistic identifies 16 genes as differentially expressed, compared to 15 genes for the same MTP based on the Welch $t$-statistic. 10 genes are identified by both procedures.
1.4.3 Augmentation procedures for gFWER, TPPFP, and FDR control

In the context of microarray gene expression data analysis or other high-dimensional inference problems, one is often willing to accept some false positives, provided their number is small in comparison to the number of rejected hypotheses. In this case, the FWER is not a suitable choice of Type I error rate and one should consider other rates that lead to larger sets of rejected hypotheses. The augmentation procedures of Section 1.2.5 implemented in the function `MTP`, allow one to reject additional hypotheses, while controlling an error rate such as the generalized family-wise error rate (gFWER), the tail probability of the proportion of false positives (TPPFP), or the false discovery rate (FDR). We illustrate the use of the `fwer2gfwer`, `fwer2tppfp`, and `fwer2fdr` functions, but note that the gFWER, TPPFP, and FDR can also be controlled directly using the `MTP` function with appropriate choices of arguments `typeone`, `k`, `q`, and `fdr.method`.

gFWER control

```r
define k = c(5, 10, 50, 100)
calculate cyto.gfwer <- fwer2gfwer(adjp = cyto.boot@adjp, k = k)
create matrix comp.gfwer <- cbind(cyto.boot@adjp, cyto.gfwer)
create MTP tps <- paste("gFWER(", c(0, k), ")", sep = "")
create MT plot mt.plot(adjp = comp.gfwer, teststat = cyto.boot@statistic, proc = tps, +       leg = c(0.1, 400), col = 1:5, lty = 1:5, lwd = 3)
title("Comparison of gFWER(k)-controlling AMTPs based on SD minP MTP")
```

For gFWER-controlling AMTPs, Figure 1.3 illustrates that the number of rejected hypotheses increases linearly with the number `k` of allowed false positives, for nominal levels `α` such that the initial FWER-controlling MTP does not reject more than `M = k` hypotheses. That is, the curve for the gFWER(`k`)–controlling AMTP is obtained from that of the initial FWER-controlling procedure by a simple vertical shift of `k`.

TPPFP control

```r
define q = c(0.05, 0.1, 0.5)
calculate cyto.tppfp <- fwer2tppfp(adjp = cyto.boot@adjp, q = q)
create matrix comp.tppfp <- cbind(cyto.boot@adjp, cyto.tppfp)
create MTP tps <- c("FWER", paste("TPPFP(", q, ",")", sep = "")
create MT plot mt.plot(adjp = comp.tppfp, teststat = cyto.boot@statistic, proc = tps, +       leg = c(0.1, 400), col = 1:4, lty = 1:4, lwd = 3)
title("Comparison of TPPFP(q)-controlling AMTPs based on SD minP MTP")
```

For TPPFP control. Figure 1.4 shows that, as expected, the number of rejections, while controlling `TPPFP(q)` at a given level `α`, increases with the allowed proportion `q` of false positives, though not linearly. Furthermore,
for the ALL dataset, the increases in the number of rejections are not very large.

**FDR control**

Given any TPPFP-controlling MTP, van der Laan et al. [4] derive two simple (conservative) FDR-controlling MTPs. Here, we compare these two FDR-controlling approaches, based on a TPPFP-controlling augmentation of the step-down minP procedure, to the marginal Benjamini & Hochberg [8] and Benjamini & Yekutieli [13] procedures, implemented in the function `mt.rawp2adjp`.

```r
> cyto.fdr <- fwer2fdr(adjp = cyto.boot@adjp, method = "both")$adjp
> cyto.marg.fdr <- mt.rawp2adjp(rawp = cyto.boot@rawp, proc = c("BY", "BH"))
> comp.fdr <- cbind(cyto.fdr, cyto.marg.fdr$adjp[order(cyto.marg.fdr$index), -1])
> mtps <- c("AMTP Cons", "AMTP Rest", "BY", "BH")
> mt.plot(adjp = comp.fdr, teststat = cyto.boot@statistic, proc = mtps,
> leg = c(0.1, 400), col = c(2, 2, 3, 3), lty = rep(1:2, 2),
> lwd = 3)
> title("Comparison of FDR-controlling MTPs")
```

Figure 1.5 shows that for most values of the nominal FDR level $\alpha$, the usual Benjamini & Hochberg ("BH") MTP leads by far to the largest number of rejected hypotheses. The Benjamini & Yekutieli ("BY") MTP, a conservative version of the Benjamini & Hochberg MTP (with $\sim \log M$ penalty on the $p$-values), leads to much fewer rejections. The AMTPs based on conservative bounds for the FDR ("AMTP Cons" and "AMTP Rest") are much more conservative than the Benjamini & Hochberg MTP and only lead to an increased number of rejections for very high nominal FDR levels.

### 1.4.4 Association of expression measures and tumor molecular subtype: multi-sample $F$-statistics

To identify genes with differences in mean expression measures between different tumor molecular subtypes (BCR/ABL, NEG, ALL1/AF4, E2A/PBX1, p15/p16, NUP-98), one can perform a family of $F$-tests. Tumor subtypes with fewer than 10 subjects are merged into one group. Adjusted $p$-values and test statistic cut-offs (for nominal levels $\alpha$ of 0.01 and 0.1) are computed as follows for the bootstrap-based single-step maxT FWER-controlling procedure.

```r
> mb <- as.character(phenomol.biol)
> table(mb)
> other <- c("E2A/PBX1", "NUP-98", "p15/p16")
```
Figure 1.3. Cytogenetic test status — gFWER-controlling AMTPs. Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing gFWER-controlling AMTPs, based on the bootstrap step-down minP FWER-controlling procedure, with different allowed numbers \( k \) of false positives.

Let us examine the results of the MTP.

```r
> mb[mb %in% other] <- "other"
> table(mb)
> mb.boot <- MTP(X = filtX, Y = mb, test = "f", alpha = c(0.01, + 0.1), B = 10000, get.cutoff = TRUE, seed = seed)

Let us examine the results of the MTP.

> summary(mb.boot)

MTP: ss.maxT
Type I error rate: fwer

Level Rejections
1 0.01 427
2 0.10 429

          Min. 1st Qu.  Median   Mean 3rd Qu. Max.
adjp 0.000000 0.000 0.000 0.001107 0.000 0.3412
rawp 0.000000 0.000 0.000 0.000000 0.000 0.0000
statistic 0.06911 1.816 4.375 5.922000 8.283 36.1300
estimate NA NA NA NaN NaN NA
```
Figure 1.4. *Cytogenetic test status — TPPFP-controlling AMTPs.* Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing TPPFP-controlling AMTPs, based on the bootstrap step-down minP FWER-controlling procedure, with different allowed proportions $q$ of false positives.

```r
> mb.diff <- mb.boot@adjp <= 0.01
> sum(mb.diff)
[1] 427

> sum(mb.boot@statistic >= mb.boot@cutoff[, "alpha=0.01"] & mb.diff)
[1] 427
```

For control of the FWER at nominal level $\alpha = 0.01$, the bootstrap-based single-step maxT procedure with $F$-statistics identifies 427 genes (i.e., almost all the 431 filtered genes) as having significant differences in mean expression measures between tumor molecular subtypes. This set can be identified through either adjusted $p$-values or cut-offs for the test statistics. The plot of test statistics and corresponding cut-offs in Figure 1.6 illustrates that the $F$-statistics for the 10 genes with the smallest adjusted $p$-values are much larger than expected by chance under the null distribution.

```r
> plot(mb.boot, which = 6)
```
Figure 1.5. Cytogenetic test status — FDR-controlling MTPs. Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing four FDR-controlling MTPs.

1.4.5 Association of expression measures and time to relapse: Cox t-statistics

The bootstrap-based MTPs implemented in the main MTP function (nulldist="boot") allow the test of hypotheses concerning regression parameters in models for which the subset pivotality condition may not hold (e.g., logistic and Cox proportional hazards models). The phenotype information in the ALL package includes the original remission status of the ALL patients (remission variable in the data.frame pData(ALL)). There are 88 subjects who experienced original complete remission (remission="CR") and who were followed up for remission status at a later date. We apply the single-step maxT procedure to test for a significant association between expression measures and time to relapse amongst these 88 subjects, adjusting for sex. Note that most of the code below is concerned with extracting the (censored) time to relapse outcome and covariates from slots of the exprSet instance ALL. B = 5000 bootstrap iterations are used in this example.

```r
> cr.ind <- pheno$remission == "CR"
> cr.pheno <- pheno[cr.ind, ]
> times <- strptime(cr.pheno$"date last seen", "%m/%d/%Y") - strptime(cr.pheno$"date.cr", "%m/%d/%Y")
```
Figure 1.6. *Tumor molecular subtype — Single-step maxT FWER-controlling MTP.* Plot of F-statistics and corresponding cut-offs for the 10 genes with the smallest adjusted p-values, based on the bootstrap single-step maxT FWER-controlling procedure (plot method, which=6).

```r
> time.ind <- !is.na(times)
> times <- times[time.ind]
> cens <- ((1:length(times)) %in% grep("CR", cr.pheno[time.ind, +
   "f.u"]) )
> rel.times <- Surv(times, !cens)
> patients <- (1:ncol(filtX))[cr.ind][time.ind]
> relX <- filtX[, patients]
> relZ <- pheno[,]

> cox.boot <- MTP(X = relX, Y = rel.times, Z = relZ, Z.incl = "sex",
   + Z.test = NULL, test = "coxph.YvsXZ", B = 5000, get.cr = TRUE,
   + seed = seed)

> summary(cox.boot)

MTP: ss.maxT
Type I error rate: fwer

   Level Rejections
   1  0.05  6

Min. 1st Qu.  Median    Mean 3rd Qu.  Max.
```
For control of the FWER at nominal level $\alpha = 0.05$, the bootstrap-based single-step maxT procedure identifies 6 genes whose expression measures are significantly associated with time to relapse. Equivalently, Figure 1.7 illustrates that the level $\alpha = 0.05$ confidence regions corresponding to these 6 genes do not include the null value $\psi_0 = 0$ for the Cox regression parameters (indicated by red horizontal line). The confidence intervals for the next four genes barely cover $\psi_0 = 0$.

1.5 Discussion

The multtest package implements a broad range of resampling-based multiple testing procedures. Ongoing efforts are as follows.
1. Extending the class of available tests, by adding test statistic closures for tests of correlations, quantiles, and parameters in generalized linear models (e.g., logistic regression).

2. Extending the class of resampling-based estimators for the test statistics null distribution (e.g., parametric bootstrap, Bayesian bootstrap). A closure approach may be considered for this purpose.

3. Providing parameter confidence regions and test statistic cut-offs for other Type I error rates than the FWER.

4. Implementing the new augmentation multiple testing procedures proposed in Dudoit & van der Laan [1] for controlling tail probabilities $Pr(g(V_n, R_n) > q)$ for an arbitrary function $g(V_n, R_n)$ of the numbers of false positives $V_n$ and rejected hypotheses $R_n$.

5. Providing a formula interface for a symbolic description of the tests to be performed (cf. model specification in `lm`).

6. Providing an `update` method for objects of class `MTP`. This would allow reusing available estimates of the null distribution to implement different MTPs for a given Type I error rate and to control different Type I error rates.
7. Extending the MTP class to keep track of results for several MTPs.

8. Increasing the computational efficiency of the bootstrap estimation of the test statistics null distribution.

References


1. Multiple Testing Procedures and Applications to Genomics
