FarmTest: An R Package for Factor-Adjusted Robust Multiple Testing

Koushiki Bose[*], Jianqing Fan[†], Yuan Ke[‡], Xiaoou Pan[§] and Wen-Xin Zhou[¶]

Abstract

We provide a publicly available library `FarmTest` implemented in the `R` programming system. This library implements a factor-adjusted robust multiple testing principle proposed by Fan *et al.* (2019) for large-scale simultaneous inference on mean effects. To capture the dependency among a large pool of variables, we use a factor model that takes explicitly into account for the dependency. Specifically, we consider three different scenarios with observable factors, latent factors and a mixture of observable and latent factors. The non-factor case, which corresponds to the standard multiple mean testing problem, is also included. The library implements a series of adaptive Huber methods integrated with fast data-driven tuning schemes to estimate model parameters and construct test statistics that are robust against heavy-tailed and/or asymmetric error distributions. Extensions to two-sample simultaneous mean testing problems are also included. Examples and datasets are contained in the library to illustrate the use of each scenario. Numerical experiments demonstrate the effectiveness of the FarmTest method, especially with highly correlated and heavy-tailed data.

Keywords: Large-scale multiple testing, dependence, robustness, factor model, Huber regression, false discovery rate.

1. Introduction

In the era of big data, large-scale multiple testing problems arise from a wide range of fields, including biological sciences such as genomics and neuroimaging, social science, signal processing, marketing analytics, and financial economics. When testing multitudinous statistical hypotheses simultaneously, researchers appreciate statistically significant evidence against the null hypothesis with a guarantee of controlled false discovery rate (FDR) (Benjamini and Hochberg 1995). Since

[*]Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, USA. E-mail: koush.bose@gmail.com.

[†]Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, USA. E-mail: jqfan@princeton.edu.

[‡]Department of Statistics, University of Georgia, Athens, GA 30602, USA. E-mail: yuan.ke@uga.edu.

[§]Department of Mathematics, University of California, San Diego, La Jolla, CA 92093, USA. E-mail: xip024@ucsd.edu.

[¶]Department of Mathematics, University of California, San Diego, La Jolla, CA 92093, USA. E-mail: wez243@ucsd.edu.

the seminal work of Benjamini and Hochberg (1995), multiple testing with FDR control has been extensively studied and successfully used in many applications. We refer to Cai and Sun (2017) for a selective survey of some significant developments in large-scale inference. Most of the existing testing procedures are tailored to independent or weakly dependent hypotheses or tests. See, Storey (2002), Genovese and Wasserman (2004) and Lehmann and Romano (2005), to name a few. The independence assumption, however, is restricted in real applications as correlation effects are ubiquitous in high dimensional measurements. In genomic studies, for instance, multiple genes may belong to the same regulatory pathway or there may exist gene-gene interactions. In neuroimaging analysis, the effective degrees of freedom are generally much smaller than the number of voxels, due to spatial correlation and continuity (Medland *et al.* 2014). Ignoring such strong dependency and directly applying standard FDR controlling procedures can lead to inaccurate false discovery control, loss of statistical power and unreliable scientific conclusions.

Over the past decade, a multi-factor model has proven to be an effective tool for modeling cross-sectional dependence, with applications in genomics, neuroscience and financial economics. Related references in the context of multiple testing include Leek and Storey (2008), Friguet *et al.* (2009), Fan *et al.* (2012), Desai and Storey (2012) and Fan and Han (2017). In Friguet *et al.* (2009) and Desai and Storey (2012), the authors suggested to subtract out the realized common factors estimated from a strict factor model using an EM algorithm. Fan *et al.* (2012) considered a general setting for estimating the FDP, where the test statistics follow a multivariate normal distribution with an arbitrary but known covariance structure. Later, Fan and Han (2017) proposed a data-driven estimator of the FDP, starting with estimating the unknown covariance matrix. In these papers, the construction of test statistics and p-values heavily rely on the assumed joint normality of factors and noise, which is arguably another folklore regarding high dimensional data. For example, the distribution of the normalized gene expressions often deviates from Gaussian, regardless of the normalization methods used (Purdom and Holmes 2005). In finance, the non-Gaussian character of the distribution of price changes has been repeatedly observed in various market data (Mandelbrot 1963; Cont 2001). Therefore, it is imperative to develop large-scale multiple testing tools that adjust cross-sectional dependence properly and are robust to heavy-tailedness at the same time.

Recently, Fan *et al.* (2019) developed a Factor-Adjusted Robust Multiple Test (FarmTest) procedure for large-scale simultaneous inference with highly correlated and heavy-tailed data. This is the first paper in multiple testing that recognizes the value of robustness against both strong cross-sectional dependence and heavy-tailed sampling distribution. More specifically, let $\boldsymbol{X} = (X_1, \ldots, X_p)^\intercal$ be a random vector with mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\intercal$. We are interested in the p hypotheses $H_{0j} : \mu_j = 0$ and wish to find a multiple comparison procedure able to reject individual hypotheses while controlling the FDR. The FarmTest method models the dependency among X_j's through an approximate multi-factor model, namely $X_j = \mu_j + \boldsymbol{b}_j^\intercal \boldsymbol{f} + u_j$, where $\boldsymbol{f}$ is a zero-mean random vector capturing the dependence structure of $\boldsymbol{X}$. The package applies to either observable or unobservable factor $\boldsymbol{f}$. The former includes the non-factor case which corresponds to the standard multiple mean testing problem. For the latter, the package is able to estimate the factors in a data-driven way. Test statistics are then calculated by subtracting out the realized common factors. Therefore, multiple comparisons are applied to these weakly dependent factor-adjusted test statistics. Also, adjusting the factors before testing reduces signal-to-noise ratios, which enhances statistical power. Since a data-driven eigenvalue ratio method is used to estimate the number of (latent) factors, the testing procedure still

works when the dependence is weak and therefore is rather flexible. For other dependency structures that can be exploited for efficient multiple testing, see Section 3.5 in Cai and Sun (2017).

This article describes an R library named `FarmTest`, which implements the FarmTest procedure(s) developed in Fan *et al.* (2019). It is a user-friendly tool to conduct large-scale hypothesis testing, especially when one or several of the following scenarios are present: the dimensionality is far larger than the sample size available; the data is heavy-tailed and/or asymmetric; there is strong cross-sectional dependence among the data. A simple call of `FarmTest` package only requires the input of a data matrix and the null hypotheses to be tested. It outputs the hypotheses that are rejected, along with the p-values and some estimated parameters which may be of use in further analysis. Testing can be carried out for both one-sample and two-sample problems.

Another key feature of our package is that it makes uses of advanced robust estimation tools for fitting regression models (Zhou *et al.* 2018; Sun *et al.* 2019) and covariance estimation (Ke *et al.* 2019). When data is generated from a heavy-tailed distribution, test statistics that are based on the least squares method are sensitive to outliers, which often causes significant false discoveries and suboptimal power (Zhou *et al.* 2018). The effect of heavy-tailedness is amplified by high dimensionality; even moderate-tailed distributions can generate very large outliers by chance, making it difficult to separate the true signals from spurious variables. As a result, large-scale multiple testing based on non-robust statistics may engender an excessive false discovery rate, which arguably is one of the causes of the current crisis in reproducibility in science. Moreover, to choose the multiple tuning parameters in robust regression and covariance estimation, we employ the recently developed data-driven procedures (Chen and Zhou 2019; Ke *et al.* 2019), which are particularly designed for adaptive Huber regression and are considerably faster than the K-fold cross-validation method used in Fan *et al.* (2019).

We illustrate the effectiveness of our method via a simple numerical experiment; see Figure 1. Suppose we observe independent data vectors $\{\boldsymbol{X}_i\}_{i=1}^n$ from a five-factor model:

$$\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip})^\intercal = \boldsymbol{\mu} + \mathbf{B}\boldsymbol{f}_i + \boldsymbol{u}_i, \quad i = 1, \ldots, n,$$

where $\boldsymbol{f}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_5)$ are the common factors, and the entries of the loading matrix $\mathbf{B} \in \mathbb{R}^{p\times 5}$ are iid from a uniform distribution, $\mathcal{U}(-2, 2)$. The idiosyncratic errors, $\boldsymbol{u}_i$'s, are independently generated from a t-distribution with 2 degrees of freedom. The sample size n and dimension p take values 100 and 500, respectively. The mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\intercal$ is set to be $\mu_j = 1$ for $1 \le j \le 125$ and 0 otherwise. The testing problem in focus is to identify the non-zero μ_j's. Figure 1 displays the histograms of estimated means through different methods. When using the sample/empirical means, there is a large overlap between the null and non-null groups, implying that the signals and noise can hardly be separated. After applying both factor-adjustment and robust estimation procedure, the alternatives are considerably better separated from the nulls, allowing us to differentiate the two groups more easily. Applying either factor-adjustment or robust method helps separate the two groups, but not as much as applying the integrated procedure. This toy example highlights the key merit of our method.

Before moving forward to the next section, we remark that the R software (R Core Team 2017) provides several functions and packages for multiple testing. However, most existing tools do not address the above two issues. The built-in hypothesis testing function in R, named `t.test`, neither adjusts for strong dependence in the data, nor estimates the parameters in focus robustly. Func-

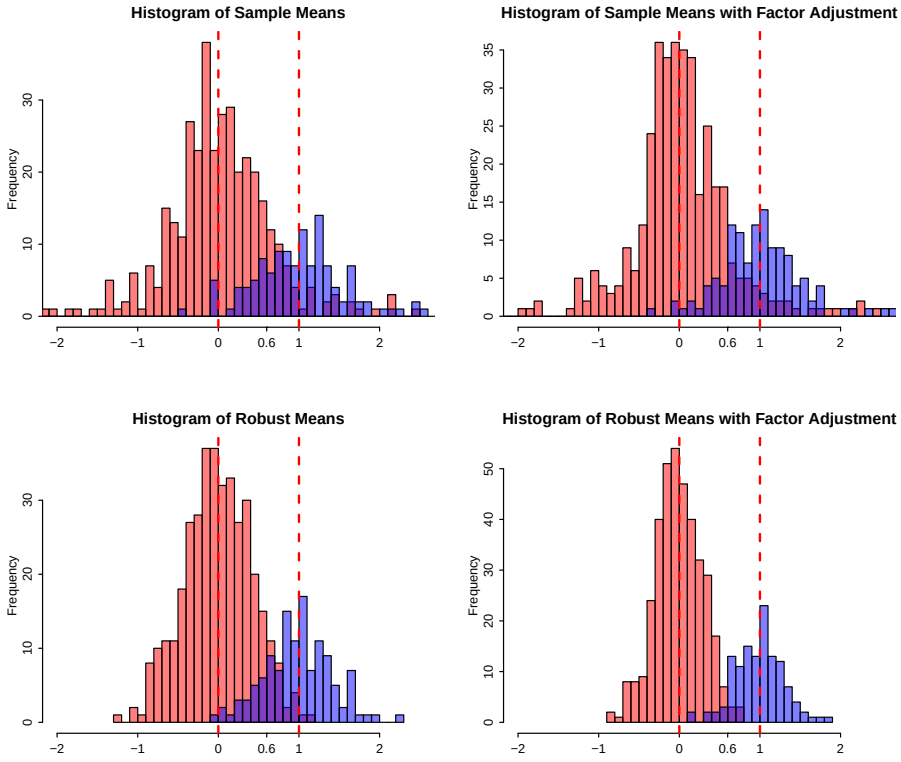

Figure 1: Histograms of estimated means using different procedures. The solid line represents the value of the true null, while the dashed line represents the value of the true signal. The colors separate the means into true and false null clusters. The overlap between these two groups makes the testing problem difficult. The means for the factor-adjusted robust method are not only better separated but also well-clustered around their true values.

tions like `p.adjust` in the `stats` package or `qvalue` (Storey *et al.* 2015) only adjust user-input p-values for multiple testing and do not address the problem of estimating the p-values themselves. The package `multcomp` (Hothorn *et al.* 2008) provides simultaneous testing tools for general linear hypotheses in parametric models under independence and normality assumptions. The package `multtest` (Pollard *et al.* 2005) is developed to implement non-parametric bootstrap and permutation resampling-based multiple testing procedures. The `multtest` can calculate test statistics based on ranked data which is robust against outliers but yields biased mean estimators. In addition, `multtest` cannot explicitly model the dependence structure in data. The package `mutoss` (MuToss Coding Team *et al.* 2015) is designed to apply many existing multiple hypothesis testing procedures with FDR control and p-value correction. Nevertheless, none of the tools in `mutoss` is suitable to deal with both strong dependency and heavy-tailedness. Moreover, existing packages are often difficult to navigate since users need to combine many functions to perform multiple tests.

The rest of the paper is organized as follows. In Section 2, we introduce the main methodologies for the multiple comparison problem. Section 3 presents detailed algorithms that are designed for three different scenarios. In Section 4, we demonstrate the usage of `FarmTest` package and its main functions. In Sections 5 and 6, we examine the performance of the package through simulated and empirical datasets.

2. Factor-adjusted robust multiple testing

In this section, we revisit the problem of simultaneous inference on the mean effects under a factor model and discuss the main ideas behind the FarmTest method developed by Fan *et al.* (2019).

2.1. Multiple testing with false discovery rate control

Suppose we observe n independent data vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ from a p-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_p)^\intercal$. Further, let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\intercal$ and $\boldsymbol{\Sigma} = (\sigma_{jk})_{1 \le j,k \le p}$ denote the mean vector and covariance matrix of $\boldsymbol{X}$, respectively. In the language of hypothesis testing, we are interested in one of the following three types of hypotheses:

$$H_{0j} : \mu_j = h_j^0 \quad \text{versus} \quad H_{1j} : \mu_j \neq h_j^0; \tag{2.1}$$

$$H_{0j} : \mu_j \le h_j^0 \quad \text{versus} \quad H_{1j} : \mu_j > h_j^0; \tag{2.2}$$

$$H_{0j} : \mu_j \ge h_j^0 \quad \text{versus} \quad H_{1j} : \mu_j < h_j^0; \tag{2.3}$$

for $j = 1, \ldots, p$. In the default setting, $h_j^0 = 0$ for all j.

Here we take the two-sided test (2.1) as an example to discuss the false discovery rate (FDR) control. For $1 \le j \le p$, let T_j be a generic test statistic for the jth hypothesis. Given a prespecified threshold $z > 0$, we reject the jth null hypothesis if $|T_j| \ge z$. The FDR is defined as the expected value of the false discovery proportion (FDP):

$$\mathrm{FDR}(z) = \mathbb{E}\{\mathrm{FDP}(z)\}, \quad \mathrm{FDP}(z) = \frac{V(z)}{\max\{R(z), 1\}}, \tag{2.4}$$

where $R(z) = \sum_{j=1}^{p} 1(|T_j| \ge z)$ is the number of total rejections and $V(z) = \sum_{j:\mu_j = h_j^0} 1(|T_j| \ge z)$ is the number of false discoveries. If the FDP(z) were known, the rejection threshold will be $z_\alpha = \inf\{z \ge$

$0 : \mathrm{FDP}(z) \le \alpha\}$ in order to achieve FDP control. Notice that $R(z)$ is observable given the data while $V(z)$ is an unobserved random quantity that needs to be estimated.

Assume that there are $p_0 = \pi_0 p$ true nulls and $p_1 = (1-\pi_0)p$ true alternatives. Suppose the constructed test statistic T_j is close in distribution to standard normal for every $j = 1, \ldots, p$, if the test statistics are weakly dependent. Heuristically the number of false discoveries $V(z)$ is close to $2p_0\,\Phi(-z)$ for any $z \ge 0$. A conservative way is to replace $V(z)$ by $2p\,\Phi(-z)$. Assuming the normal approximation is sufficiently accurate, $2p\,\Phi(-z)$ provides an overestimate of the number of false discoveries, resulting in an underestimate of the $\mathrm{FDP}(z)$. A more accurate method is to estimate the unknown proportion of null hypotheses $\pi_0 = p_0/p$ from the data. Let $\{P_j = 2\Phi(-|T_j|)\}_{j=1}^p$ be the approximate p-values. For a predetermined $\lambda \in [0, 1)$, Storey (2002) suggest to estimate π_0 by

$$\widehat{\pi}_0(\lambda) = \frac{\sum_{j=1}^p 1(P_j > \lambda)}{(1-\lambda)p}, \tag{2.5}$$

because larger p-values are more likely to come from the true null hypotheses. Consequently, a data-driven rejection threshold is $\widehat{z}_\alpha = \inf\{z \ge 0 : \widehat{\mathrm{FDP}}(z) \le \alpha\}$, where $\widehat{\mathrm{FDP}}(z) = 2\widehat{\pi}_0(\lambda)p\,\Phi(-z)/R(z)$.

2.2. Factor-adjusted test statistics

In this section, we discuss the construction of test statistics under strong cross-sectional dependency captured by common factors. Specifically, we allow the p coordinates of $\boldsymbol{X}$ to be strongly correlated through an approximate factor model of the form $\boldsymbol{X} = \boldsymbol{\mu} + \mathbf{B}\boldsymbol{f} + \boldsymbol{u}$, where $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p)^\intercal \in \mathbb{R}^{p\times K}$ represents the factor loading matrix, $\boldsymbol{f} = (f_1, \ldots, f_K)^\intercal \in \mathbb{R}^K$ is the common factor, and $\boldsymbol{u} = (u_1, \ldots, u_p)^\intercal \in \mathbb{R}^p$ denotes a vector of idiosyncratic errors uncorrelated with $\boldsymbol{f}$. The observed samples thus follow

$$\boldsymbol{X}_i = \boldsymbol{\mu} + \mathbf{B}\boldsymbol{f}_i + \boldsymbol{u}_i, \quad i = 1, \ldots, n, \tag{2.6}$$

where $(\boldsymbol{f}_i, \boldsymbol{u}_i)$'s are independent copies of $(\boldsymbol{f}, \boldsymbol{u})$. Assume that both $\boldsymbol{f}$ and $\boldsymbol{u}$ have zero means. Further, denote by $\boldsymbol{\Sigma}_f$ and $\boldsymbol{\Sigma}_{\boldsymbol{u}} = (\sigma_{u,jk})_{1\le j,k\le p}$ the covariance matrices of $\boldsymbol{f}$ and $\boldsymbol{u}$, respectively.

Our package allows the common factor $\boldsymbol{f}$ to be either observable or unobservable. In the former case, we observe $\{(\boldsymbol{X}_i, \boldsymbol{f}_i)\}_{i=1}^n$ so that model (2.6) is reduced to a multi-response linear regression problem; for the latter, we only observe $\{\boldsymbol{X}_i\}_{i=1}^n$ and therefore need to recover the latent factors. The latent factor model has known identifiability issues; see Bai and Li (2012) for a set of possible solutions. For simplicity, we adopt the following identification conditions on $\mathbf{B}$ and $\boldsymbol{f}$:

$$\boldsymbol{\Sigma}_f = \mathbf{I}_K \quad \text{and} \quad \mathbf{B}^\intercal\mathbf{B} \text{ is diagonal.} \tag{2.7}$$

In Section 3, we construct robust test statistics in both cases and then state the algorithms.

2.3. Robust estimation

As another key feature, the FarmTest method is robust against heavy-tailed sampling distributions. Under such scenarios, the ordinary least squares estimators can be suboptimal. Recently, Fan *et al.* (2017) and Sun *et al.* (2019) proposed the adaptive Huber regression method, the core of which is Huber's M-estimator (Huber 1964) with a properly calibrated robustification parameter that adapts to the sample size, dimensionality and noise level. They showed that the adaptive Huber estimator

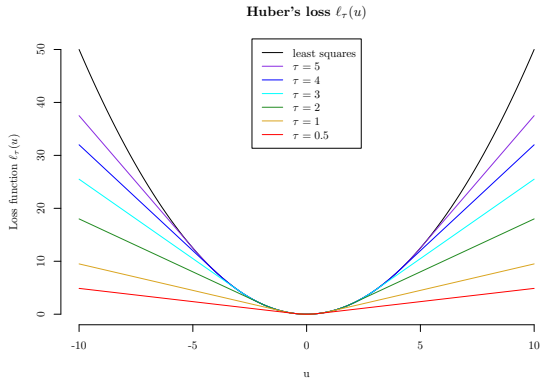

Figure 2: Huber's loss function for various choices of the tuning parameter τ. The least-squares (ℓ_τ with $\tau = \infty$)-loss is also shown for comparison.

admits a sub-Gaussian-type deviation bound under mild moment conditions. This package exploits this approach to estimate the unknown parameters and to construct test statistics. Recall that the Huber loss is defined as

$$\ell_\tau(u) = \begin{cases} u^2/2 & \text{if } |u| \le \tau, \\ \tau|u| - \tau^2/2 & \text{if } |u| > \tau, \end{cases} \tag{2.8}$$

where $\tau > 0$ is the robustification parameter.

The robustification parameter τ controls the blending of the quadratic and linear components of the loss. Figure 2 reveals the shape of the Huber loss with different values of τ. When $\tau \to \infty$, the Huber loss approaches the quadratic loss that leads to the least squares estimator. On the other hand, when $\tau \to 0$, the Huber loss approaches the L_1 loss (after properly normalized), which corresponds to the least absolute deviation (LAD) estimator. The LAD estimator is robust to outliers but is also biased if the distribution is asymmetric.

3. Algorithms

In this section, we formally describe the algorithms for the FarmTest procedure. The `FarmTest` library is implemented in `C++` and called from `R` by user-friendly interfaces. We revisit and discuss procedures for the two scenarios with observable and unobservable/latent factors (Zhou *et al.* 2018; Fan *et al.* 2019). Notice that the two scenarios are inherently different in terms of estimating unknown parameters and constructing test statistics. The algorithms with observable and latent factors are presented in Section 3.1 and 3.2, respectively. Although we only focus on one-sample tests (2.1)–(2.3), the method can be easily extended to two-sample problems. We discuss such an extension in Sections 3.1 and 3.2. Moreover, multiple testing with partially observed factors is another interesting problem, and will be discussed in Section 3.3. Section 3.4 discusses the selection of tuning parameters.

3.1. Observable factors

Suppose we observe independent data vectors $\{(\boldsymbol{X}_i, \boldsymbol{f}_i)\}_{i=1}^n$ from model (2.6). The testing procedure for the hypotheses in (2.1)–(2.3) is described in Algorithm 1. Algorithm 1 automatically selects the robustification parameters $\{\tau_j, \upsilon_j\}_{j=1}^p$ following the data-driven method proposed by Ke *et al.* (2019). See Section 3.4 for more details. To enhance the finite sample performance, alternatively we can use the weighted/multiplier bootstrap (Zhou *et al.* 2018; Chen and Zhou 2019) to compute p-values for all the marginal hypotheses. For $b = 1, \ldots, B$, we obtain the corresponding bootstrap draw of $(\widehat{\mu}_j, \widehat{\boldsymbol{b}}_j)$ via

$$(\widehat{\mu}_{b,j}^\flat, \widehat{\boldsymbol{b}}_{b,j}^\flat) = \operatorname*{argmin}_{\mu, \boldsymbol{b}} \sum_{i=1}^n w_{b,ij} \ell_{\tau_j}(X_{ij} - \mu - \boldsymbol{f}_i^\top \boldsymbol{b}),$$

where $\{w_{b,ij}, i = 1, \ldots, n, j = 1, \ldots p\}$ are independent and identically distributed (iid) random variables that are independent from the data and satisfy $\mathbb{E}(w_{b,ij}) = 1$ and $\mathrm{var}(w_{b,ij}) = 1$. For two-sided alternatives, the bootstrap p-values are then defined as $P_j^\flat = (1/B) \sum_{b=1}^B I(|\widehat{\mu}_{b,j}^\flat - \widehat{\mu}_j| \geq |\widehat{\mu}_j|)$, followed by Steps 5–7 in Algorithm 1.

Algorithm 1 FarmTest with known factors (Zhou *et al.* 2018)

Input: Data $\{(\boldsymbol{X}_i, \boldsymbol{f}_i)\}_{i=1}^n$, null hypotheses $\{h_j^0\}_{j=1}^p$, and $\alpha, \lambda \in (0, 1)$

1: For $j = 1, \ldots, p$, obtain the Huber estimators $(\widehat{\mu}_j, \widehat{\boldsymbol{b}}_j) \in \operatorname{argmin}_{\mu, \boldsymbol{b}} \sum_{i=1}^n \ell_{\tau_j}(X_{ij} - \mu - \boldsymbol{f}_i^\top \boldsymbol{b})$.

2: Estimation of residual variances $\sigma_{u,jj}$'s: compute

(i) $\widehat{\boldsymbol{\Sigma}}_f = (1/n) \sum_{i=1}^n \boldsymbol{f}_i \boldsymbol{f}_i^\top$, $\widehat{\theta}_j \in \operatorname{argmin}_\theta \sum_{i=1}^n \ell_{\upsilon_j}(X_{ij}^2 - \theta)$ for $j = 1, \ldots, p$;

(ii) $\widehat{\sigma}_{u,jj} = \widehat{\theta}_j - \widehat{\mu}_j^2 - \widehat{\boldsymbol{b}}_j^\top \widehat{\boldsymbol{\Sigma}}_f \widehat{\boldsymbol{b}}_j$ if $\widehat{\theta}_j > \widehat{\mu}_j^2 + \widehat{\boldsymbol{b}}_j^\top \widehat{\boldsymbol{\Sigma}}_f \widehat{\boldsymbol{b}}_j$; otherwise $\widehat{\sigma}_{u,jj} = \widehat{\theta}_j$.

3: Construct test statistics $T_j = \sqrt{n/\widehat{\sigma}_{u,jj}}(\widehat{\mu}_j - h_j^0)$ for $j = 1, \ldots, p$.

4: Compute p-values $\{P_j\}_{j=1}^p = \begin{cases} \{2\Phi(-|T_j|)\}_{j=1}^p & \text{for (2.1)}, \\ \{\Phi(-T_j)\}_{j=1}^p & \text{for (2.2)}, \\ \{\Phi(T_j)\}_{j=1}^p & \text{for (2.3)}. \end{cases}$

5: Estimate the proportion of true alternatives: $\widehat{\pi}_0(\lambda) = \frac{\mathrm{Card}\{P_j > \lambda\}}{(1-\lambda)p}$.

6: Order the p-values as $P_{(1)} \leq \cdots \leq P_{(p)}$.
Compute the rejection threshold $t := \max\{1 \leq j \leq p : P_{(j)} \leq \frac{\alpha j}{\widehat{\pi}_0(\lambda) p}\}$

7: Reject each hypothesis in the set $\{1 \leq j \leq p : P_j \leq P_{(t)}\}$.

Output: Rejected hypotheses, p-values, other estimated parameters

Next we discuss an extension of Algorithm 1 to the two-sample problem, which is also implemented in the package. Suppose we observe two independent samples $\{(\boldsymbol{X}_i, \boldsymbol{f}_i^X)\}_{i=1}^{n_1}$ and $\{(\boldsymbol{Y}_i, \boldsymbol{f}_i^Y)\}_{i=1}^{n_2}$ from the models

$$\boldsymbol{X} = \boldsymbol{\mu}^X + \mathbf{B}^X \boldsymbol{f}^X + \boldsymbol{u}^X \text{ and } \boldsymbol{Y} = \boldsymbol{\mu}^Y + \mathbf{B}^Y \boldsymbol{f}^Y + \boldsymbol{u}^Y. \tag{3.1}$$

We are interested in the p hypotheses $H_{0j} : \mu_j^X - \mu_j^Y = h_j^0$ versus $H_{1j} : \mu_j^X - \mu_j^Y \neq h_j^0$ or versus some one-sided alternatives. To begin with, applying Step 1 in Algorithm 1 separately to each dataset to obtain the estimates $\{(\widehat{\mu}_j^X, \widehat{\mu}_j^Y)\}_{j=1}^p$ and $\{(\widehat{\sigma}_{u,jj}^X, \widehat{\sigma}_{u,jj}^Y)\}_{j=1}^p$. Next, define the two-sample counterparts

of the test statistics in Step 2 as

$$T_j = \frac{\widehat{\mu}_j^X - \widehat{\mu}_j^Y - h_j^0}{\sqrt{\widehat{\sigma}_{u,jj}^X/n_1 + \widehat{\sigma}_{u,jj}^Y/n_2}}, \quad j = 1, \ldots, p.$$

After that, we follow Steps 3–7 as in Algorithm 1 to obtain the p-values and rejected hypotheses.

3.2. Latent factors

In this section, suppose we are given independent observations $\{\boldsymbol{X}_i\}_{i=1}^n$. The strong dependency among the coordinates of $\boldsymbol{X}_i$ is captured by a latent factor $\boldsymbol{f}_i$ (Leek and Storey 2008). Due to the need of recovering latent factors from the data, the corresponding testing procedure is more involved. We summarize the major steps in Algorithm 2. All the tuning parameters required for Algorithm 2, $\{\tau_j, \upsilon_j\}_{j=1}^p$ and $\{\upsilon_{jk}\}_{1\le j<k\le p}$, are automatically selected from the data; see Section 3.4.

An extension of Algorithm 2 to the two-sample problem is also included in the library. Suppose we observe two independent samples $\{\boldsymbol{X}_i\}_{i=1}^{n_1}$ and $\{\boldsymbol{Y}_i\}_{i=1}^{n_2}$, and wish to test the hypotheses $H_{0j} : \mu_j^X - \mu_j^Y = h_j^0$ versus $H_{1j} : \mu_j^X - \mu_j^Y \neq h_j^0$ or some one-sided alternatives. In this case, Steps 1–5 in Algorithm 2 are applied separately to each dataset to obtain the estimates $\{(\widehat{\mu}_j^X, \widehat{\mu}_j^Y)\}_{j=1}^p$, $\{(\widehat{\sigma}_{u,jj}^X, \widehat{\sigma}_{u,jj}^Y)\}_{j=1}^p$, $\widehat{\mathbf{B}}^X$, $\widehat{\mathbf{B}}^Y$ and $\bar{\boldsymbol{f}}^X$, $\bar{\boldsymbol{f}}^Y$. After replacing the test statistics in Step 6 with

$$T_j = \frac{(\widehat{\mu}_j^X - \langle \widehat{\boldsymbol{b}}_j^X, \bar{\boldsymbol{f}}^X \rangle) - (\widehat{\mu}_j^Y - \langle \widehat{\boldsymbol{b}}_j^Y, \bar{\boldsymbol{f}}^Y \rangle) - h_j^0}{\sqrt{\widehat{\sigma}_{u,jj}^X/n_1 + \widehat{\sigma}_{u,jj}^Y/n_2}}, \quad j = 1, \ldots, p,$$

one can follow Steps 7–10 to obtain the p-values and rejected hypotheses.

3.3. Partially observable factors

Motivated by applications to comparative microarray experiments (Leek and Storey 2008; Friguet *et al.* 2009) and mutual fund selection (Lan and Du 2019), we further discuss the case where both explanatory variables and latent factors are present. The statistical model is of the form

$$\boldsymbol{X}_i = \boldsymbol{\mu} + \mathbf{B}\boldsymbol{f}_i + \mathbf{C}\boldsymbol{g}_i + \boldsymbol{u}_i, \quad i = 1, \ldots, n,$$

where $\boldsymbol{f}_i \in \mathbb{R}^K$ is a vector of explanatory variables and $\boldsymbol{g}_i \in \mathbb{R}^L$ represents the latent factor. Here $L \ge 1$ may be user-specified or is unknown. For multiple comparison of the mean effects under this model, the `FarmTest` package can be used in a two-stage fashion. In the first stage, apply Algorithm 1 to fit model (2.6) with observed data $\{(\boldsymbol{X}_i, \boldsymbol{f}_i)\}_{i=1}^n$ and compute fitted residuals $\boldsymbol{X}_i^{\text{res}} = \boldsymbol{X}_i - \widehat{\mathbf{B}}\boldsymbol{f}_i$; in the second stage, run Algorithm 2 on $\{\boldsymbol{X}_i^{\text{res}}\}_{i=1}^n$ to conduct factor-adjusted multiple testing.

3.4. Selection of tuning parameters

The FarmTest procedure (Fan *et al.* 2019) involves multiple tuning parameters, including the number of factors K (if not specified by the user) and robustification parameters for fitting factor models. For the former, we apply the eigenvalue ratio method (Lam and Yao 2012; Ahn and Horenstein

Algorithm 2 FarmTest with latent factors (Fan *et al.* 2019)

Input: Data $\{\boldsymbol{X}_i\}_{i=1}^n$, null hypotheses $\{h_j^0\}_{j=1}^p$, and $\alpha, \lambda \in (0,1)$

1: For $j = 1, \ldots, p$, compute

- $\widehat{\mu}_j = \operatorname{argmin}_\mu \sum_{i=1}^n \ell_{\tau_j}(X_{ij} - \mu)$, $\widehat{\theta}_j = \operatorname{argmin}_\theta \sum_{i=1}^n \ell_{\nu_j}(X_{ij}^2 - \theta)$,
- $\widehat{\sigma}_{jj} = \begin{cases} \widehat{\theta}_j - \widehat{\mu}_j^2 & \text{if } \widehat{\theta}_j > \widehat{\mu}_j^2, \\ \widehat{\theta}_j & \text{otherwise.} \end{cases}$

2: Define the paired data $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_N\} = \{\boldsymbol{X}_1 - \boldsymbol{X}_2, \boldsymbol{X}_1 - \boldsymbol{X}_3, \ldots, \boldsymbol{X}_{n-1} - \boldsymbol{X}_n\}$, where $N = n(n-1)/2$. For $1 \le j < k \le p$, compute

- $\widehat{\sigma}_{jk} = \operatorname{argmin}_\theta \sum_{i=1}^N \ell_{\nu_{jk}}(Y_{ij}Y_{ik}/2 - \theta)$, and $\widehat{\sigma}_{kj} = \widehat{\sigma}_{jk}$.

3: Define the covariance matrix estimator $\widehat{\boldsymbol{\Sigma}} = (\widehat{\sigma}_{jk})_{1 \le j,k \le p}$.

- Let $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_p$ be the ordered eigenvalues of $\widehat{\boldsymbol{\Sigma}}$ and denote by $v_1, v_2, \ldots, v_p$ the corresponding eigenvectors.
- Calculate $K = \operatorname{argmax}_{1 \le k \le \min(n,p)/2} \frac{\lambda_k}{\lambda_{k+1}}$. The step is omitted if K is specified by the user in advance.
- Calculate $\widehat{\mathbf{B}} = (\widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_p)^\intercal = (\lambda_1^{1/2}\boldsymbol{v}_1, \ldots, \lambda_K^{1/2}\boldsymbol{v}_K) \in \mathbb{R}^{p \times K}$.

4: $\bar{\boldsymbol{f}} = \operatorname{argmin}_{\boldsymbol{f} \in \mathbb{R}^K} \sum_{j=1}^p \ell_\gamma(\bar{X}_j - \widehat{\boldsymbol{b}}_j^\intercal \boldsymbol{f})$, where $\bar{X}_j = (1/n)\sum_{i=1}^n X_{ij}$.

5: For $j = 1, \ldots, p$, compute $\widehat{\sigma}_{u,jj} = \begin{cases} \widehat{\sigma}_{jj} - \|\widehat{\boldsymbol{b}}_j\|_2^2 & \text{if } \widehat{\sigma}_{jj} > \|\widehat{\boldsymbol{b}}_j\|_2^2, \\ \widehat{\sigma}_{jj} & \text{otherwise.} \end{cases}$

6: Construct test statistics $T_j = \sqrt{n/\widehat{\sigma}_{u,jj}}(\widehat{\mu}_j - \widehat{\boldsymbol{b}}_j^\intercal \bar{\boldsymbol{f}} - h_j^0)$, $j = 1, \ldots, p$.

7: Compute p-values $P_j = \begin{cases} \{2\Phi(-|T_j|)\}_{j=1}^p & \text{for (2.1)}, \\ \{\Phi(-T_j)\}_{j=1}^p & \text{for (2.2)}, \\ \{\Phi(T_j)\}_{j=1}^p & \text{for (2.3)}. \end{cases}$

8: Estimate the proportion of true alternatives: $\widehat{\pi}_0(\lambda) = \frac{\text{Card}\{P_j > \lambda\}}{(1-\lambda)p}$.

9: Order the p-values as $P_{(1)} \le \ldots \le P_{(p)}$.
Compute the rejection threshold $t := \max\{1 \le j \le p : P_{(j)} \le \frac{\alpha j}{\widehat{\pi}_0(\lambda)p}\}$

10: Reject each hypothesis in the set $\{1 \le j \le p : P_j \le P_{(t)}\}$.

Output: Rejected hypotheses, p-values, other estimated parameters

2013) to estimate K, that is,

$$\widehat{K} = \underset{1\le k\le K_{\max}}{\operatorname{argmax}} \frac{\lambda_k(\widehat{\mathbf{\Sigma}})}{\lambda_{k+1}(\widehat{\mathbf{\Sigma}})},$$

where $\widehat{\mathbf{\Sigma}}$ is a generic covariance matrix estimator with eigenvalues $\lambda_1(\widehat{\mathbf{\Sigma}}) \ge \cdots \ge \lambda_p(\widehat{\mathbf{\Sigma}})$, and $K_{\max}$ be a prescribed upper bound. In the library, we take $K_{\max} = \min(n, p)/2$. This method is chosen as it does not involve other hyperparameters (except $K_{\max}$). When the factors are unobservable, the estimation of K is essentially an un-supervised learning problem. We choose K to be the smallest nonnegative integer such that the residuals $\boldsymbol{X}_i - \mathbf{B}\boldsymbol{f}_i$ are weakly correlated. Therefore, slight over-estimation of K does not affect much the testing results. If K is set to be zero, the FarmTest library directly applies a robust multiple testing procedure based on Huber's M-estimation partnered with multiplier bootstrap. See Zhou *et al.* (2018) for more details.

The robustification parameter in the Huber loss plays an important role in controlling the bias-robustness tradeoff. According to the theoretical analysis in Zhou *et al.* (2018), the optimal choice of τ_j in Algorithm 1 depends on the variance of X_j. Due to heterogeneity, we have p different τ_j's that need to be selected from the data. Furthermore, the covariance estimation step in Algorithm 2 entails as many as $p(p-1)/2$ parameters υ_{jk}. In this case, it is computationally expensive to use cross-validation (Fan *et al.* 2019) when the dimension is large. Recently, Ke *et al.* (2019) proposed a fast data-driven approach, which estimates the regression coefficients/covariances and calibrates the tuning parameter simultaneously by solving a system of equations. Numerical studies therein suggest that the new data-driven method is considerably faster than cross-validation while performs equally as well. Another library named `tfHuber` that implements this procedure for mean and covariance matrix estimation as well as standard and ℓ_1-regularized Huber regression can be found in our GitHub page: https://github.com/XiaoouPan/tfHuber.

4. Package overview

The `FarmTest` package has four core functions. The main function `farm.test` carries out the entire FarmTest procedure, and outputs the testing results along with several useful estimated model parameters. A user-friendly print function that summarizes the test outcome is equipped with `farm.test`. The function `farm.fdr` conducts multiple testing with FDR control using the method proposed in Storey (2002), see Steps 4–6 in Algorithm 1 or Steps 8–10 in Algorithm 2. The other two functions, `farm.mean` and `farm.cov`, implement the tuning-free principle (Ke *et al.* 2019) for estimating the mean vector and covariance matrix, which are of independent interest. In this section, we focus primarily on introducing the `farm.test` function, and demonstrate its usage with numerical experiments.

The FarmTest package is publicly available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=FarmTest`. To avoid computationally expensive tuning via cross-validation, we apply a tuning-free principle and develop an updated version that can be found at `https://github.com/XiaoouPan/FarmTest`. We will soon submit this updated version to CRAN. At the moment, it can be installed into `R` environment via `devtools` package from the GitHub repository:

```
R> install.packages("devtools")
```

```
R> library(devtools)
R> devtools::install_github("XiaoouPan/FarmTest")
```

4.1. A showcase example

We first present an example by applying the package to a synthetic dataset. To begin with, we use the `rstiefel` package to simulate a uniformly distributed random orthonormal matrix as the loading matrix $\mathbf{B}$ after rescaling. With sample size $n = 120$, dimension $p = 400$ and number of factors $K = 5$, we generate data vectors $\{\boldsymbol{X}_i\}_{i=1}^n$ from model (2.6), where the factors $\boldsymbol{f}_i \in \mathbb{R}^K$ follow a standard multivariate normal distribution and the noise vectors $\boldsymbol{u}_i \in \mathbb{R}^p$ are drawn from a multivariate t_3 distribution with zero mean and identity covariance matrix. For the mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\intercal$, we set the first $p_1 = 100$ coordinates to be 1 and the rest to be 0.

```
R> library(FarmTest)
R> library(rstiefel)
R> library(mvtnorm)
R> n <- 120
R> p <- 400
R> K <- 5
R> set.seed(100)
R> B <- rustiefel(p, K) %*% diag(rep(sqrt(p), K))
R> FX <- rmvnorm(n, rep(0, K), diag(K))
R> p1 <- 100
R> strength <- 1
R> mu <- c(rep(strength, p1), rep(0, p - p1))
R> U <- rmvt(n, diag(p), 3)
R> X <- rep(1, n) %*% t(mu) + FX %*% t(B) + U
```

4.2. Function call with default parameters

Using the data generated above, let us call the main function `farm.test` with all default optional parameters, and then summarize the outputs via its print function.

```
R> output <- farm.test(X)
R> output

One-sample FarmTest with unknown factors
n = 120, p = 400, nFactors = 5
FDR to be controlled at: 0.05
Alternative hypothesis: two.sided
Number of hypotheses rejected: 102
```

As shown in the snapshot above, the function `farm.test` correctly estimates the number of factors, and rejects 102 hypotheses with 2 false discoveries. For this individual experiment, the FDP and power are 0.020 and 1, respectively as calculated below. Here the power is referred to as the ratio between the number of correct rejections and the number of nonnulls p_1.

Output	Implication	Data type	R class
means	estimated means	p-vector	matrix
stdDev	estimated standard deviations	p-vector	matrix
loadings	estimated loading matrix	$(p \times K)$-matrix	matrix
eigenVal	eigenvalues of estimated covariance	p-vector	matrix
eigenRatio	eigenvalue ratios of estimated covariance	$(\min\{n, p\}/2)$-vector	matrix
nFactors	(estimated) number of factors	positive integer	integer
tStat	test statistics	p-vector	matrix
pValues	p-values	p-vector	matrix
significant	indicators of significance	boolean p-vector	matrix
reject	indices of rejected hypotheses	vector	integer
type	whether factor is known	string	character
n	sample size	positive integer	integer
p	data dimension	positive integer	integer
h0	null hypothesis	p-vector	numeric
alpha	nominal FDR level	numerical number	numeric
alternative	alternative hypothesis	string	character

Table 1: Objects in the output list of farm.test function with their implications, and description of data type and class in R language.

```
R> FDP <- sum(output$reject > p1) / length(output$reject)
R> FDP

[1] 0.01960784

R> power <- sum(output$reject <= p1) / p1
R> power

[1]  1
```

All the outputs are incorporated into a list, which can be quickly examined by `names()` function. See Table 1 for detailed descriptions of the outputs.

```
R> names(output)

 [1] "means"       "stdDev"      "loadings"    "eigenVal"    "eigenRatio"  "nFactors"
 [7] "tStat"       "pValues"     "significant" "reject"      "type"        "n"
[13] "p"           "h0"          "alpha"       "alternative"
```

With these above objects, users may present the testing results in their own way. To illustrate, we create a table encapsulating `means`, `tStat`, `pValues`, `significant`, which represent estimated means, test statistics, p-values and significance indicators.

```
R> result <- cbind(output$means, output$tStat, output$pValues, output$significant)
R> colnames(result) <- c("means", "statistics", "p-values", "significant")
R> head(result)
```

```
         means statistics     p-values significant
[1,] 1.1109811   8.901542 5.507617e-19           1
[2,] 0.8521871   6.008699 1.870184e-09           1
[3,] 0.8596745   6.713502 1.900078e-11           1
[4,] 0.9306973   7.047471 1.821988e-12           1
[5,] 0.7343990   5.438463 5.374206e-08           1
[6,] 0.9383129   7.349271 1.992903e-13           1
```

To visualize the testing results, in Figure 3 we present several plots based on the outputs. From the histograms of estimated means and test statistics, we see that data are generally categorized into two groups, one of which has $\widehat{\mu}_j$ concentrated around 1 and test statistics bounded away from 0. It is therefore relatively easy to identify alternatives/signals from the nulls. From the eigenvalue ratio plot, we see that the fifth ratio (highlighted as a red dot) is evidently above the others, thus determining the number of factors. The scree plot, on the other hand, reveals that the top 5 eigenvalues (above the red dashed line) together explain the vast majority of the variance, indicating that the proportion of common variance (due to common factors) is high.

4.3. Function call with options

In this section, we illustrate `farm.test` function with other options that allow us to call it more flexibly.

FarmTest with known factors

When the factors are observable, we can simply put the $n \times K$ factor matrix into argument `fX`, and the output is formatted the same as before. As a remark, among all the items listed in Table 1, `eigenVal` and `eigenRatio`, which are eigenvalues and eigenvalue ratios of estimated covariance matrix, are not available in this case; see Algorithm 1.

```
R> output <- farm.test(X, fX = FX)
R> output
```

```
One-sample FarmTest with known factors
n = 120, p = 400, nFactors = 5
FDR to be controlled at: 0.05
Alternative hypothesis: two.sided
Number of hypotheses rejected: 98
```

One-sided test

Consider one-sided alternatives $H_{1j} : \mu_j \geq 0$, $j = 1, \ldots p$ with a nominal FDR level 1%. We modify the arguments `alternative` and `alpha` as follows:

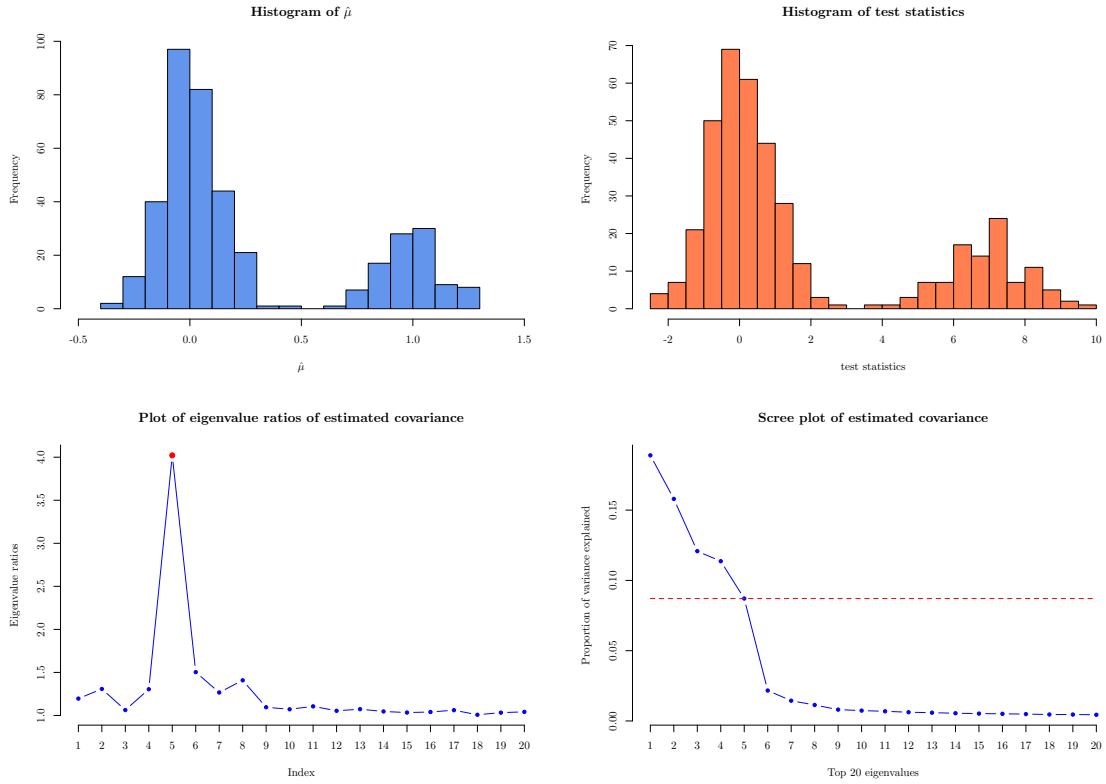

Figure 3: Upper panel: histograms of estimated means and test statistics. Lower panel: eigenvalue ratio plot with the largest ratio highlighted and scree plot of the eigenvalues of the estimated covariance matrix.

```
R> output <- farm.test(X, alternative = "greater", alpha = 0.01)
R> output

One-sample FarmTest with unknown factors
n = 120, p = 400, nFactors = 5
FDR to be controlled at: 0.01
Alternative hypothesis: greater
Number of hypotheses rejected: 101
```

None-zero null hypotheses

Users can specify null hypotheses bypassing any vector with length p into argument `h0`. In the next example, we consider the p null hypotheses as all the means are equal to 1, so that the number of true nonnulls becomes 300.

```
R> output <- farm.test(X, h0 = rep(1, p), alpha = 0.01)
R> output

One-sample FarmTest with unknown factors
n = 120, p = 400, nFactors = 5
FDR to be controlled at: 0.01
Alternative hypothesis: two.sided
Number of hypotheses rejected: 300
```

User-specified number of factors

When the factors are unknown, users can also specify the number of factors based on some subjective grounds. In this case, Step 3 in Algorithm 2 is avoided. For example, we run the function with the number of factors chosen to be `KX = 2`, which is less than the true parameter 5. This misspecification results in a loss of power with two true alternatives unidentified.

```
R> output <- farm.test(X, KX = 2)
R> power <- sum(output$reject <= p1) / p1
R> power

[1] 0.98
```

As a special case, if we declare `KX = 0` in the function, a robust multiple test without factor-adjustment is conducted.

```
R> output <- farm.test(X, KX = 0)
R> output
```

```
One-sample robust multiple test without factor-adjustment
n = 120, p = 400
FDR to be controlled at: 0.05
Alternative hypothesis: two.sided
Number of hypotheses rejected: 97
```

Two-sample test

Finally, we present an example of two-sample FarmTest. Using the same sampling distributions for the factor loading matrix, factors and noise vectors, we generate another sample $\{Y_i\}_{i=1}^m$ from model (3.1) with $m = 150$.

```
R> m <- 150
R> set.seed(200)
R> BY <- rustiefel(p, K) %*% diag(rep(sqrt(p), K))
R> FY <- rmvnorm(m, rep(0, K), diag(K))
R> uY <- rmvt(m, diag(p), 3)
R> Y <- FY %*% t(BY) + uY
```

Then `farm.test` function can be called with an additional argument `Y`.

```
R> output <- farm.test(X, Y = Y)
R> output

Two-sample FarmTest with unknown factors
X.n = 120, Y.n = 150, p = 400, X.nFactors = 5, Y.nFactors = 5
FDR to be controlled at: 0.05
Alternative hypothesis: two.sided
Number of hypotheses rejected: 103
```

The output is formatted similarly as in Table 1, except that `means`, `stdDev`, `loadings`, `eigenVal`, `eigenRatio`, `nFactors` and `n` now consist of two items for samples `X` and `Y`.

```
R> names(output$means)

[1] "X.mean" "Y.mean"
```

5. Simulations

In this section, we assess and compare the performance of `farm.test` function in the `FarmTest` package with the following methods:

- t-test using the R built-in function `t.test`;

- WMW-test (Wilcoxon-Mann-Whitney) using the `onesamp.marginal` function in the `mutoss` package (MuToss Coding Team *et al.* 2015);
- RmTest (Robust Multiple test) without factor-adjustment by claiming `KX = 0` in the `farm.test` function.

For t-test and WMW-test, the functions we call produce vectors of p-values, to which the `farm.fdr` function in the `FarmTest` package is applied.

In all the numerical experiments, we consider two-sided alternatives with a nominal FDR level $\alpha = 5\%$. The true number of factors is 5. Factors and loadings are generated the same way as in Section 4.1. To add dependency among idiosyncratic errors, the covariance matrix of $\boldsymbol{u}$, denoted by $\boldsymbol{\Sigma}_u$, is taken to be a block-diagonal symmetric matrix with block size 5×5. Within each block, the diagonal entries are all equal to 3 and the off-diagonal entries are generated from $\mathcal{U}[0, 1]$. In the simulations, we drop the case where the generated $\boldsymbol{\Sigma}_u$ is not positive-definite. The distribution of $\boldsymbol{u}$ is specified in two models as follows.

- **Model 1**. $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_u)$: centered multinormal distribution with covariance matrix $\boldsymbol{\Sigma}_u$;
- **Model 2**. $\boldsymbol{u} \sim t_3(\mathbf{0}, \boldsymbol{\Sigma}_u)$: multivariate t-distribution with degrees of freedom 3 and covariance matrix $\boldsymbol{\Sigma}_u$.

For each model, we consider various combinations of sample size n and dimensionality p, specifically, $n \in \{60, 80, 100, 120, 140\}$ and $p \in \{200, 400, 600, 800, 1000\}$. The number of true alternatives p_1 is taken to be $0.2\,p$, and the signal strength is set as $4\sqrt{\log(p)/n}$.

Figures 4 and 5 depict the FDR and power curves for either "fixed n growing p" or "fixed p growing n" based on 200 simulations. Across various settings, FarmTest consistently maintains high empirical power with FDR well controlled around the nominal level. In contrast, the competing methods may lose as many as 10% to 30% powers, which can be ascribed to not accounting for the common factors. In summary, we conclude that the `FarmTest` package provides an efficient implementation of the FarmTest method, which carries out multiple testing for multivariate data with heavy-tailed distribution and a strong dependency structure.

6. Real data example

In this section, we apply the `FarmTest` package to test the mean effects of stock returns. In capital asset pricing theory, the stock's risk-adjusted mean return or "alpha" is a quantity of interest since it indicates the excessive return incurred from investing in a particular stock. If the efficient equity market hypothesis holds, we expect "alpha" to be zero. Hence, detecting non-zero alphas can help investors to identify market inefficiencies, that is, whether certain stocks exhibit an abnormal rate of return or are mispriced. As discussed in Cont (2001), both cross-sectional dependency and heavy tailedness are silent features of stock returns. Traditional multiple testing tools may fail to efficiently detect true non-zero alphas while keeping the false discovery rate under control.

In this study, we test the annual mean effects of stocks in the S&P500 index. The data is available on COMPUSTAT and CRSP databases. Figure 6 displays the histogram of excess kurtosises of all

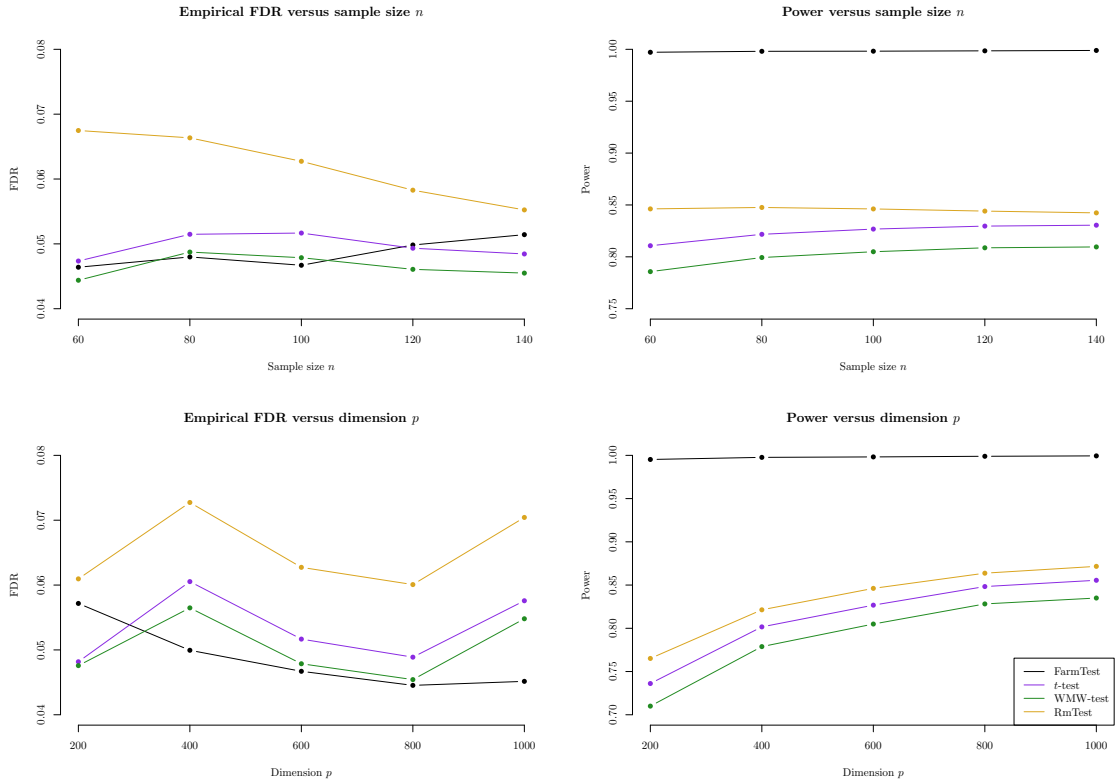

Figure 4: Comparison of FarmTest with three other methods in terms of FDR and power under Model 1 (multivariate normal distribution). In the upper panel, p is fixed at 600 and n grows from 60 to 140; in the lower panel, n is fixed at 100 and p ranges from 200 to 1000.

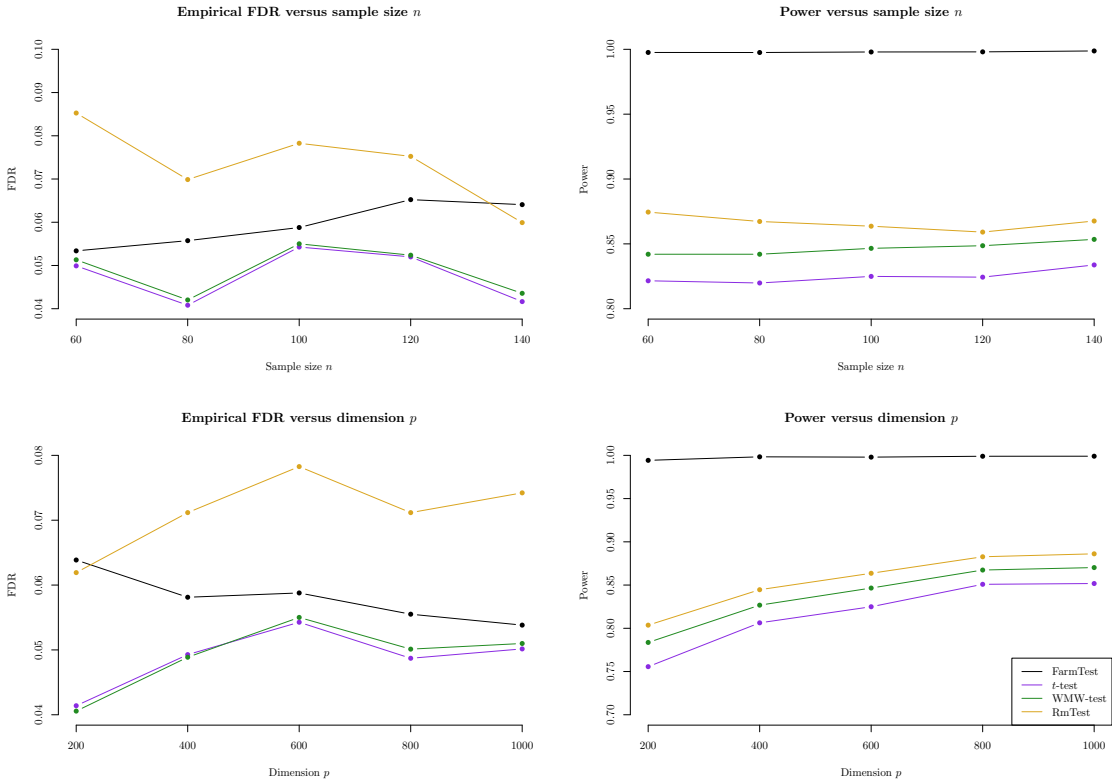

Figure 5: Comparison of FarmTest with three other methods in terms of FDR and power under Model 2 (multivariate t-distribution). In the upper panel, p is fixed at 600 and n grows from 60 to 120; in the lower panel, n is fixed at 100 and p ranges from 200 to 1000.

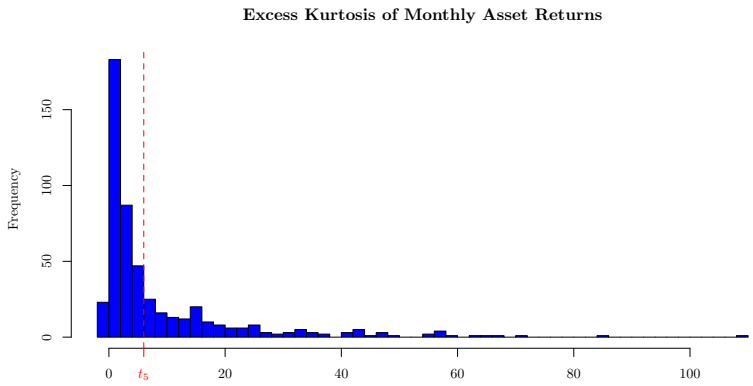

Figure 6: Histogram of excess kurtosises of monthly returns of the stocks in the S&P500 index from 2008 to 2016. The red dashed line marks the excess kurtosis of t_5-distribution.

the stocks that have continuous membership in the S&P500 index from 2008 to 2016. Most stocks have excess kurtosises greater than zero, indicating tails heavier than that of a normal distribution. Besides, more than 33% of the stocks are severely heavy-tailed as their excess kurtosises exceed 6, the excess kurtosis of the t_5-distribution. We collect monthly returns of stocks from the S&P500 index over rolling windows: for each month between 2008 and 2016, we collect monthly returns of stocks who have continuous records over the past year. The average number of stocks collected each year is 598. For each rolling window, we conduct multiple testing using the four methods considered in the previous section, that is, FarmTest, t-test, WMW-test, and RmTest.

The nominal FDR level is set as $\alpha = 1\%$. Within each rolling time window, we have $p \approx 600$ and $n = 12$. The numbers of discoveries of each method are depicted chronologically in Figure 7, and Table 2 displays several key summary statistics. Since the t-test barely discovers any stock throughout the whole procedure, we only present the results for the other three methods in Figure 7. It is interesting to observe that across different time rolling windows, the testing outcomes of the WMW-test are relatively stable and time-insensitive. FarmTest, on the other hand, selects much fewer stocks in the year of 2009, coinciding to some extent with the financial crisis during which the market volatility is much higher. RmTest typically selects most stocks, which is partly due to the lack of FDR control under strong dependency. A major, noticeable impact of dependence is that it results in clusters of rejections: if a test is rejected, then there are likely to be further rejections for tests that are highly correlated with this one. This phenomenon is in accord with our simulation results, showing that FarmTest simultaneously controls the FDR and maintains high power while the other methods either make too many false discoveries or fail to detect true signals.

7. Discussion

We provide the implementation of FarmTest, a flexible large-scale multiple testing method that is robust against strongly dependent and/or heavy-tailed data. The factor-adjustment procedure helps to construct weakly dependent test statistics, and also enhances statistical power by reducing the signal-to-noise ratio. Moreover, by exploiting the idea of adaptive Huber regression, the testing

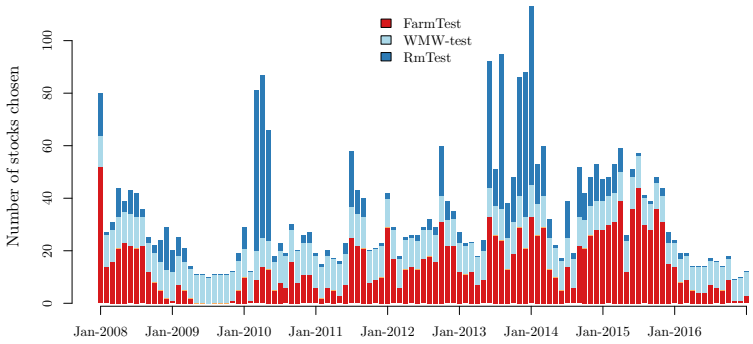

Figure 7: Stack bar plot of the numbers of discoveries via FarmTest, WMW-test and RmTest from 2008 to 2016, using rolling windows of one year. Within each time window, we report the number of stocks in the S&P500 index that show significant statistical evidence against null hypotheses that there are no excessive returns, with FDR controlled at 1%.

Method	Mean	Std. Dev.	Median	Min	Max
FarmTest	14.477	11.070	12	0	52
WMW-test	10.991	1.005	11	8	12
RmTest	8.147	14.414	3	0	68

Table 2: Summary statistics of the number of discoveries via FarmTest, WMW-test and RmTest between 2008 and 2016 using rolling windows of size 12 (months).

procedure is robust against heavy-tailed data. We demonstrate the efficacy of our package on both real and simulated data sets.

References

Ahn SC, Horenstein AR (2013). "Eigenvalue Ratio Test for the Number of Factors." *Econometrica*, **81**, 1203–1227.

Bai J, Li K (2012). "Statistical Analysis of Factor Models of High Dimension." *The Annals of Statistics*, **40**, 436–465.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, **51**, 289–300.

Cai TT, Sun W (2017). "Large-Scale Global and Simultaneous Inference: Estimation and Testing in Very High Dimensions." *Annual Review of Economics*, **9**, 411–439.

Chen X, Zhou WX (2019). "Robust Inference via Multiplier Bootstrap." *The Annals of Statistics*, in press.

Cont R (2001). "Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues." *Quantitative Finance*, **1**, 223–236.

Desai KH, Storey JD (2012). "Cross-Dimensional Inference of Dependent High-Dimensional Data." *Journal of the American Statistical Association*, **107**, 1143–1164.

Fan J, Han X (2017). "Estimation of the False Discovery Proportion with Unknown Dependence." *Journal of the Royal Statistical Society B*, **79**, 135–151.

Fan J, Han X, Gu W (2012). "Estimating False Discovery Proportion under Arbitrary Covariance Dependence." *Journal of the American Statistical Association*, **107**, 1019–1035.

Fan J, Ke Y, Sun Q, Zhou WX (2019). "FARM-Test: Factor-Adjusted Robust Multiple Testing with Approximate False Discovery Control." *Journal of the American Statistical Association*. URL https://doi.org/10.1080/01621459.2018.1527700.

Fan J, Li Q, Wang Y (2017). "Estimation of High Dimensional Mean Regression in the Absence of Symmetry and Light Tail Assumptions." *Journal of the Royal Statistical Society B*, **79**, 247–265.

Friguet C, Kloareg M, Causeur D (2009). "A Factor Model Approach to Multiple Testing under Dependence." *Journal of the American Statistical Association*, **104**, 1406–1415.

Genovese C, Wasserman L (2004). "A Stochastic Process Approach to False Discovery Control." *The Annals of Statistics*, **32**, 1035–1061.

Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." *Biometrical Journal*, **50**(3), 346–363.

Huber PJ (1964). "Robust Estimation of a Location Parameter." *The Annals of Mathematical Statistics*, **35**, 73–101.

Ke Y, Minsker S, Ren Z, Sun Q, Zhou WX (2019). "User-Friendly Covariance Estimation for Heavy-Tailed Distributions." *Statistical Science*, **34**(3), 454–471.

Lam C, Yao Q (2012). "Factor Modeling for High Dimensional Time-Series: Inference for the Number of Factors." *The Annals of Statistics*, **40**, 694–726.

Lan W, Du L (2019). "A Factor-Adjusted Multiple Testing Procedure with Application to Mutual Fund Selection." *Journal of Business and Economic Statistics*, **37**, 147–157.

Leek JT, Storey JD (2008). "A General Framework for Multiple Testing Dependence." *Proceedings of the National Academy of Sciences of the United States of America*, **105**, 18718–18723.

Lehmann EL, Romano JP (2005). "Generalizations of the Familywise Error Rate." *The Annals of Statistics*, **33**, 1138–1154.

Mandelbrot B (1963). "The Variation of Certain Speculative Prices." *Journal of Business*, **36**, 394–419.

Medland S, Jahanshad N, Neale B, Thompson P (2014). "The Variation of Certain Speculative Prices." *Nature Neuroscienc*, **17**, 791–800.

MuToss Coding Team, Blanchard G, Dickhaus T, Hack N, Konietschke F, Rohmeyer K, Rosenblatt J, Scheer M, Werft W (2015). `mutoss`*: Unified Multiple Testing Procedures*. R package version 0.1-10, URL https://CRAN.R-project.org/package=mutoss.

Pollard K, Dudoit S, van der Laan M (2005). *Multiple Testing Procedures: R* ***multtest*** *Package and Applications to Genomics, in Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer.

Purdom E, Holmes SP (2005). "Error Distribution for Gene Expression Data." *Statistical Applications in Genetics and Molecular Biology*, **4**, 16.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Storey JD (2002). "A Direct Approach to False Discovery Rates." *Journal of the Royal Statistical Society B*, **64**(3), 479–498.

Storey JD, Bass AJ, Dabney A, Robinson D (2015). ***qvalue****: Q-value Estimation for False Discovery Rate Control*. R package version 2.4.2, URL http://github.com/jdstorey/qvalue.

Sun Q, Zhou WX, Fan J (2019). "Adaptive Huber Regression." *Journal of the American Statistical Association*. URL https://doi.org/10.1080/01621459.2018.1543124.

Zhou WX, Bose K, Fan J, Liu H (2018). "A New Perspective on Robust *M*-Estimation: Finite Sample Theory and Applications to Dependence-Adjusted Multiple Testing." *The Annals of Statistics*, **46**, 1904–1931.