Decentralized nonparametric multiple testing

Subhadeep Mukhopadhyay

To cite this article: Subhadeep Mukhopadhyay (2018): Decentralized nonparametric multiple testing, Journal of Nonparametric Statistics, DOI: 10.1080/10485252.2018.1508678

To link to this article: https://doi.org/10.1080/10485252.2018.1508678

View supplementary material

Published online: 09 Aug 2018.

Submit your article to this journal

View Crossmark data
Decentralized nonparametric multiple testing

Subhadeep Mukhopadhyay
Department of Statistical Science, Temple University, Philadelphia, Pennsylvania, USA

ABSTRACT
Consider a big data multiple testing task, where, due to storage and computational bottlenecks, one is given a very large collection of p-values by splitting into manageable chunks and distributing over thousands of computer nodes. This paper is concerned with the following question: How can we find the full data multiple testing solution by operating completely independently on individual machines in parallel, without any data exchange between nodes? This version of the problem tends naturally to arise in a wide range of data-intensive science and industry applications whose methodological solution has not appeared in the literature to date; therefore, we feel it is necessary to undertake such analysis. Based on the nonparametric functional statistical viewpoint of large-scale inference, started in Mukhopadhyay, S. (2016), ‘Large Scale Signal Detection: A Unifying View’, Biometrics, 72, 325–334, this paper furnishes a new computing model that brings unexpected simplicity to the design of the algorithm which might otherwise seem daunting using classical approach and notations.

1. The open problem
Consider a multiple testing task with number of hypotheses in the millions, or even billions, as in high-throughput genomics, neuroscience, astronomy, marketing and other data-intensive applications. In this paper, we are interested in cases where these massive collection of p-values (corresponding to the null hypotheses) are distributed across multiple machines by breaking them into manageable chunks, as shown in Figure 1. Given this set of partitioned p-values \( P_j = \{u_{j1}, \ldots, u_{jn_j}\}, \quad (j = 1, \ldots, K) \), suppose the goal of a data scientist is to get the full data (oracle) multiple testing result controlling overall false discovery rate (fdr), without shipping all the p-values to a centralised computing machine, as this would clearly be unrealistic due to huge volume (too expensive to store), computational bottleneck, and possible privacy restrictions. Driven by practical need, the interest for designing Decentralized Large-Scale Inference Engine has enormously increased in the last few years, due to their ability to scale cost effectively as the data volume continued to increase by leveraging modern distributed storage and computing environments.
Figure 1. The data structure and setting of decentralised large-scale inference problem. Massive collection of $p$-values distributed across large number of computer nodes.

There is, however, apparently no explicit algorithm currently available in the literature to tackle this innocent-looking problem of breaking the multiple testing computation into many pieces, each of which can be processed completely independently on individual machines in parallel.

Remark 1.1: Here we give two real-data examples from genetics, where millions to billions of hypotheses are tested routinely to perform multiple hypotheses testing procedures. The first one is GWAS studies (MacArthur et al. 2016), which require procedures that can perform tens of billions of tests for finding significant interaction between the pairs of single-nucleotide polymorphisms (SNPs) within a reasonable time-frame. The second example is eQTL studies (Xia et al. 2011), usually consist of $10^9$ tests. No doubt there are innumerable examples like this in modern data-intensive sciences and engineering applications, which necessitate a distributed multiple testing architecture.

Remark 1.2: To get a glimpse of the challenge of developing distributed large-scale inference methods, consider a specific multiple testing method, say the Benjamini Hochberg’s (BH) FDR controlling procedure, which starts by calculating the global-rank of each $p$-value:

$$\text{Global-rank of } u_{ji} = \# \text{ } p\text{-values } \leq u_{ji} \text{ in the full } - \text{ data } \bigcup_{j=1}^{k} \mathcal{P}_j.$$  

The computation of global-ranks, from the partitioned $p$-values, without any communications between the machines, is a highly non-trivial problem. Difficulty with similar caliber also arises in implementing local false discovery type algorithms.

The aim of this paper is to provide a general framework for designing Decentralized Large-Scale Inference algorithms, by adopting the nonparametric functional statistical viewpoint proposed in Mukhopadhyay (2016). The key to our theory is a
new modelling principle, called the ‘Superposition property,’ as a basis for addressing the big data challenge in a way that is easy to implement and understand (also teach).

2. The method

In this paper, we suggest a new modelling theory for designing the desired scalable simultaneous inference architecture. At its core, there is a key representation scheme based on Superposition principle. To get there, however, we need to introduce a few modern notations and basic definitions.

2.1. Background and notations

We begin by recalling the basic notations and some theoretical background as given in Mukhopadhyay (2016), which will be used throughout the paper. Let $Z_i$’s are the test statistics for the corresponding hypothesis testing problem $H_i (i = 1, \ldots, N)$ and the goal is to detect false null hypotheses by testing them simultaneously. More broadly, we can think $Z_1, Z_2, \ldots, Z_N$ as a mixed random sample, with the majority of the observations coming from null distribution $F_0$, and a small proportion from unknown signal distribution $H$:

$$F = \eta F_0 + (1 - \eta) H, 0 < \eta \leq 1.$$ 

Note that here $H$ is arbitrary, i.e. it could be a mixture of unknown distributions of any complexity.

**Definition 2.1:** Our nonparametric theory of large-scale inference starts by defining comparison distribution function between $F_0$ and $F$ (with respective densities $f_0$ and $f$) by $D(u; F_0, F) := F(F_0^{-1}(u))$ and the corresponding comparison density:

$$d(u; F_0, F) = \frac{f(F_0^{-1}(u))}{f_0(F_0^{-1}(u))}, 0 < u < 1.$$

Consider testing $N$ independent null hypothesis $H_1, \ldots, H_N$ based on corresponding p-values $u_1, \ldots, u_N$, where $u_i$ is equals to $F_0(z_i)$ or $1 - F_0(z_i)$ depending on whether we want left-tailed or right-tailed $p$-values. If all the null-hypotheses are true (i.e. under $H_0 : F = F_0$) we would expect $\hat{D}(u_i; F_0, F) = \hat{F}(F_0^{-1}(u_i)) \approx u_i$ where $\hat{F}(z; Z) = N^{-1} \sum_{i=1}^{N} I(Z_i \leq z)$. Thus, intuitively, one can suspect that the collection of $p$-values $\{u_i : \hat{D}(u_i)/u_i > \gamma\}$, for a suitably chosen threshold $\gamma$, data-dependent or constant, potentially correspond to the true signals or false null hypotheses. Based on this intuition, the following theorem presents an equivalent representation of the BH procedure (Benjamini and Hochberg 1995) in our notation:

**Theorem 2.1 (Mukhopadhyay 2016):** Let $u_{(1)} \leq u_{(2)} \leq \cdots \leq u_{(N)}$ be the sorted $p$-values of $H_{(1)}, \ldots, H_{(N)}$. Then the procedure that rejects $H_{(1)}, \ldots, H_{(k)}$ where

$$k = \text{argmax}_i \left\{ \frac{\hat{D}(u_{(i)})}{u_{(i)}} \cdot \frac{\eta}{\alpha} \right\}$$

controls FDR at the level $\alpha$, regardless of the distribution of the test statistic corresponds to false null hypothesis.
Another popular method, Higher Criticism (Donoho and Jin 2004), also admits comparison distribution representation. Reject $H_i$ for $i = 1, \ldots, k$ where

$$k = \left\{ 1 \leq i \leq \alpha_0 N : \arg \max \sqrt{N} \frac{\check{D}(u(i)) - u(i)}{u(i)(1 - u(i))} \right\}, \quad \alpha_0 \in (0, 1).$$

Furthermore, frequentist and Bayesian large-scale inference algorithms can be connected using the theory of reproducing kernel Hilbert space (RKHS) of the limiting Brownian bridge process of the comparison distribution. Efron’s empirical Bayes local false discovery (Efron et al. 2001) can alternatively be represented in terms of the $p$-values using our notation as

$$\text{fdr}(u) = \Pr(\text{null} \mid U = u) = \eta/d(u; F_0, F), \quad 0 < u < 1$$

which leads to following procedure: reject all $H_i$ if $\check{d}(u; F_0, F) > \eta/2\alpha$. Efron (2007) showed that under certain condition on the alternatives, this method controls size, or Type I errors at the desired level $\alpha$.

Thus a harmonious unification between different cultures of multiple testing is possible by recasting it into a nonparametric comparison density function approximation problem, thereby allowing a more convenient and concise description of the existing techniques.

### 2.2. Towards decentralised model

The functional statistical reformulation discussed in the earlier section provides us with the first impetus towards decentralising multiple testing computing. Nonetheless, to develop the explicit strategy (of estimating comparison density), we need more. Traditional off-the-shelf nonparametric density estimation algorithms (e.g. kernel density estimation technique) faces stiff modelling challenges; see Supplementary Appendix B1 for more discussion. To address this, we introduce a specialized nonparametric model, called skew-Beta model, that is amenable to distributed computing. This is a critical in parallelising the computation across a large number of machines, with no loss of accuracy.

**Definition 2.2:** The Skew-Beta comparison density model is given by:

$$d(u; F_0, F) = f_B(u; \gamma, \beta) \left\{ 1 + \sum_j \text{LP}[j; F_B, D] T_j(u; F_B) \right\}, \quad 0 < u < 1,$$

where beta density and distribution are denoted by $f_B$ and $F_B$, respectively; $T_j(u; F_B)$ are called LP-polynomials—Legendre polynomials of rank-transformed random variables, given by $\text{Leg}_j(F_B(u))$. LP polynomials constitute a complete basis in the Hilbert space $L_2^\tau(F_B)$, which dictates the optimality of the stochastic expansion (3).

**Theorem 2.2:** The generalised LP-Fourier coefficients of the skew-Beta nonparametric model (3) admit the following representation:

$$\text{LP}[j; F_B, D] = \mathbb{E}[\text{Leg}_j(F_B(U)); D] = \int_0^1 \text{Leg}_j(F_B(u))dD(u; F_0, F).$$
This suggests that the unknown coefficients of the model (3) can be rapidly computed by taking the mean of the Leg\(_j\) score functions evaluated at the beta-transformed p-values:

\[
\text{LP}[j; F_B, \tilde{D}] = N^{-1} \sum_{i=1}^{N} \text{Leg}_j[F_B(u_i; \gamma, \beta)].
\]

**Remark 2.1:** The method described so far is applicable when one can access all the p-values \(\{u_1, \ldots, u_N\}\) at once on a single computer. We call this framework *Centralized Simultaneous Inference Model*. However, as we have argued this may not be scalable and flexible setting in the ‘big data’ era. Next, we address this limitation by developing a theory of computation that can operate in parallel on the p-values distributed across multiple computers to yield the oracle full data multiple testing solution.

Let \(K\) denote the number of partitions or the number of CPUs, each containing \(n_l\) p-values \(\{u_{l1}, \ldots, u_{ln_l}\}\). The full data comparison distribution function can be expressed as

\[
D(u; F_0, \tilde{F}) = \tilde{F}(Q(u; F_0)) = N^{-1} \sum_{i=1}^{k} \sum_{j=1}^{n_l} \text{Leg}_j[F_B(u_{lj})] = \sum_{l=1}^{K} \pi_l D(u; F_0, \tilde{F}_l),
\]

where \(\pi_l = n_l/N\), and \(Q(u; F_0)\) is the quantile function for null \(F_0\). Often we will be using a shorthand notation \(\tilde{D}_l\) for \(D(u; F_0, \tilde{F}_l)\) (by a slight abuse of notation) for compactness.

**Theorem 2.3:** The full data LP-Fourier coefficients (4) admit the following notable distributed representation

\[
\text{LP}[j; F_B, \tilde{D}] = N^{-1} \sum_{l=1}^{K} \sum_{i=1}^{n_l} \text{Leg}_j(F_B(u_{lj})) = \sum_{l=1}^{K} \pi_l \text{LP}[j; F_B, \tilde{D}_l],
\]

(5)

where \(\text{LP}[j; F_B, \tilde{D}_l] = n_l^{-1} \sum_{i=1}^{n_l} \text{Leg}_j(F_B(u_{lj}))\).

**Remark 2.2 (Large-scale inference for big data via local modelling):** As a consequence of Theorem 2.3, one can perform ‘local modelling’– modelling by computing the LP-coefficients \(\text{LP}[j; F_B, \tilde{D}_l]\) based on the local p-values \(\{u_{l1}, \ldots, u_{ln_l}\}\) in parallel, to yield the full data ‘global’ LP-coefficients. This allows us to scale multiple testing problems for very large datasets on a cluster of machines, leveraging big data processing platforms such as Apache Hadoop or Spark.

Substituting (5) into (3), we have the following representation of the comparison density

\[
d(u; F_0, \tilde{F}) = f_B(u; \gamma, \beta) \left[ 1 + \sum_{l=1}^{K} \pi_l \sum_{j=1}^{m} \text{LP}[j; F_B, \tilde{D}_l] \text{Leg}_j(F_B(u; \gamma, \beta)) \right]
\]

\[
= \sum_{l=1}^{K} \pi_l f_B(u; \gamma, \beta) \left[ 1 + \sum_{j=1}^{m} \text{LP}[j; F_B, \tilde{D}_l] \text{Leg}_j(F_B(u; \gamma, \beta)) \right].
\]

(6)
2.3. Superposition principle

Define the locally estimated comparison density for the $l$th partition as
\[
d(u; F_0, \tilde{F}_l) = f_B(u; \gamma, \beta) \left[ 1 + \sum_{j=1}^{m} \operatorname{LP}[j; F_B, \tilde{D}_l] \operatorname{Leg}_j(F_B(u; \gamma, \beta)) \right].
\] (7)

Combining the LP representations (6) and (7), we get the following remarkable decomposition formula.

**Theorem 2.4 (The superposition principle):** Under LP-expansion, the oracle full-data based comparison density estimate can be represented as the weighted sum of the locally estimated comparison densities:
\[
d(u; F_0, \tilde{F}) = \sum_{l=1}^{K} \pi_l d(u; F_0, \tilde{F}_l), \quad 0 < u < 1.
\] (8)

**Remark 2.3:** The modelling paradigm based on the principle of superposition suggests that we can estimate the global (full data) comparison density function by properly stitching together all the ‘local snapshots’ $d(u; F_0, \tilde{F}_l)$ in a completely parallelised manner. Furthermore, it doesn’t matter how the $p$-values are partitioned, as the final aggregated result (8) will always agree in the end. This idea of *decomposition over distributed data-blocks* is interesting in its own right as a means of developing parallelizable algorithms for statistical modelling.

**Remark 2.4 (Signal heterogeneity index):** The shape of the individual comparison density estimates $d(u; F_0, \tilde{F}_l)$ in (8) are highly informative in revealing how heterogeneous (signal-rich) the different $p$-value partitions are. In fact, the homogeneous data-distribution hypothesis $H_0 : F_1 = \cdots = F_K$, can equivalently be rephrased in terms of equality of component comparison densities $H'_0 : d_1 = \cdots = d_K$. Consequently, if data partitions result in significantly different estimates of $d_l$, that would indicate the possibility of heterogeneity; thus, it is naturally tempting to come up with a rapidly computable measure of *signal heterogeneity index*. For each partition define the $H$-statistic:
\[
H_l \leftarrow \sum_{j=1}^{m} \left| \operatorname{LP}_l[j; U, \tilde{D}_l] \right|^2 = \sum_{j=1}^{m} \left| n_l^{-1} \sum_{i=1}^{n_l} \operatorname{Leg}_j(u_{il}) \right|^2; \quad (l = 1, \ldots, K).
\] (9)

The rationale comes from the following theorem.

**Theorem 2.5:** For any arbitrary $G$ with support $[0, 1]$, the skew-$G$ LP represented comparison density is given by $d(u; G, F) = g(u)\{1 + \sum_j \operatorname{LP}[j; G, D] T_j(u; G)\}$. Define the for Chi-square divergence between $D$ and $G$ too be $\chi^2(D||G) = \int [d(u)/g(u) - 1]^2 dG$. Then Chi-square divergence, which measures how close $d$ is from $g$, admits the following expression:
\[
\sum_j \left| \operatorname{LP}[j; G, D] \right|^2 \mathbb{E}_G[\operatorname{Leg}_j^2(U)] + \sum_{j \neq k} \operatorname{LP}[j; G, D] \operatorname{LP}[k; G, D] \mathbb{E}_G \left[ \operatorname{Leg}_j(U)\operatorname{Leg}_k(U) \right].
\] (10)
Important point to note: To measure the departure from uniformity (the null $p$-value distribution) by selecting $G$ to be uniform distribution $U[0, 1]$ in Theorem 2.5, the general expression (10) drastically simplifies as Legendre polynomials are orthogonal with respect to the uniform measure, thereby boiling down to $\sum_j |LP[j; U, D]|^2$, which can be readily computed using (9) for different partitions. The $H$-statistic values can be used to find high-priority (discovery-prone) partitions for careful investigations.

**Remark 2.5 (Data-driven weighted multiple testing):** At this point, an astute reader may be wondering whether we can also use $H_l$ as our data-driven weights to increase the power of the multiple testing procedures. Indeed, these group-specific heterogeneity indices can be used for constructing weights by properly normalising them:

$$w_l = (\pi_l)^{-1} \frac{H_l}{\sum_I H_l} \text{ such that } \sum_{l=1}^K \pi_l w_l = 1.$$  

The empirical detection power can be increased significantly in a heterogeneous case by running partition-specific scanning with different thresholds. For example, The rejection region of (1) can be modified based on data-driven weights as $\mathcal{R} = \bigcup_{l=1}^K \mathcal{R}_l$ where

$$
\mathcal{R}_l = \text{ Collection of } p - \text{ values in the } l\text{th partition } \leq \max_{1 \leq i \leq n_l} \left\{ u_{(li)} : \frac{\tilde{D}(u_{(li)})}{u_{(li)}} > \frac{\pi_0}{w_l \alpha} \right\}  
$$

This refined weighted version is expected to increase the power (Westfall, Kropf, and Finos 2004; Ignatiadis, Klaus, Zaugg, and Huber 2016) of the proposed distributed multiple testing procedure compared to their unweighted counterparts. This demonstrates how heterogeneity can be leveraged for designing powerful large-scale distributed signal detection algorithms. Obviously, instead of data-driven nonparametric weights, domain scientists can also assign weights to each of the partitions using prior scientific knowledge, or they can use some kind of fusion of both data-driven and science-driven weights.

**2.4. The algorithm**

We outline the steps of our algorithm derived from the theory and ideas described in the previous section.

*Algorithm: Decentralized Nonparametric Multiple Testing Engine*

**Step 1.** We start with the collection of $p$-values $\{u_{i1}, \ldots, u_{i_{n_l}}\}_{l=1}^K$ distributed across $K$ machines; $N = \sum_{l=1}^K n_l$ denotes the total number of the $p$-values (which could be in billions and thus can exceed the capacity of a single machine).

**Step 2.** For $j = 1, 2$ compute $M_j = \sum_{l=1}^K \pi_l M_j[\tilde{D}_l]$, where $M_j[\tilde{D}_l]$ denotes the $j$th order sample moment of the $p$-values present in the $l$th partition $n_l^{-1} \sum_{i=1}^{n_l} u_{li}^j$, and $\pi_l = n_l/N$.  


Step 3. Compute the method of moment estimators of the parameters of beta distribution

\[ \hat{\gamma} = \frac{M_1(M_1 - M_2)}{M_2 - M_1^2}; \quad \hat{\beta} = \frac{(1 - M_1)(M_1 - M_2)}{M_2 - M_1^2}. \]

Step 4. For \( l = 1, \ldots, K \): At each partition separately compute

Step 4a. \( \text{LP}[j; F_B, \tilde{D}_l] = n^{-1}_l \sum_{i=1}^{n_l} \text{Leg}_j(F_B(u_{li}; \hat{\gamma}, \hat{\beta})); \)

Step 4b. \( \text{LP}[j; U, \tilde{D}_l] = n^{-1}_l \sum_{i=1}^{n_l} \text{Leg}_j(u_{li}); \)

Step 4c. \( H_l = \sum_{j=1}^{m} |\text{LP}[j; U, \tilde{D}_l]|^2. \)

Step 5. Using Theorem 2.3, for \( j = 1, \ldots, m \) compute \( \text{LP}[j; F_B, \tilde{D}] = \sum_{l=1}^{K} \pi_l \text{LP}[j; F_B, \tilde{D}_l]. \)

Step 6. Return the estimated full data comparison density:

\[ \hat{d}(u; F_0, F) = f_B(u; \hat{\gamma}, \hat{\beta}) \left\{ 1 + \sum_{j=1}^{m} \text{LP}[j; F_B, \tilde{D}] \text{Leg}_j(F_B(u; \hat{\gamma}, \hat{\beta})) \right\}, \quad \text{for } 0 < u < 1 \]

where recall that \( f_B \) and \( F_B \) respectively denote beta density and distribution function. Estimate the smooth nonparametric model by selecting the ‘significantly large’ LP-coefficients using the method proposed in Mukhopadhyay (2016, Section 3.3). At this point one can even estimate the proportion of true null hypothesis by applying the Minimum Deviance Algorithm of Mukhopadhyay (2016, Section 3.4) on \( \hat{d}(u; F_0, F). \)

Step 7. Implement (1)–(2): they are upgraded nonparametrically smooth versions of BH, HC, and local FDR. See Appendix B2 for more details.

Step 8. For more insights, return heterogeneity indices \( H_1, \ldots, H_K \). Partitions with higher \( H \)-index get prioritised. Display the chart consisting of pairs of points \((l, H_l)\); see Section 3 for more details.

Step 9. Further enhancement: Improve the power of the decentralised multiple testing procedure (Step 7) by using partition-specific thresholds. Compute data-driven weights \( w_l = (\pi_l)^{-1}(H_l/\sum_{i=1}^{K} H_l) \) \((l = 1, \ldots, K)\) and incorporate into (11).

Remark 2.6: The proposed algorithm immediately allows parallelisation and a MapReduce type implementation. In particular, the ‘Map()’ function consists of Steps 2 and 4 (local modelling and parallel execution); and in the ‘Reduce()’ stage we perform (combining) Steps 3, 5, and 9 (requires no data exchange between nodes). As a result, our modelling framework represents a significant step forward, for it enables massive scalability to perform simultaneous inference on genuinely large datasets distributed over a cluster of commodity machines.

3. Examples

Two examples will be discussed one real data and the other one a simulated study.
Example 3.1: Prostate cancer data (Singh et al. 2002) consists of 102 patient samples (50 normal and 52 as prostate tumor samples) and $N = 6033$ gene expression measurements. We aim to detect interesting genes that are differentially expressed in the two samples. For this purpose, we compute $p$-values based on two-sample $t$-test for each gene. Instead of having all the $p$-values in one centralised machine, we assume that they are distributed across $K$ processors based on the following partitioning scheme: sort the $p$-values and separate the lowest 1% of $p$-values ($\sim 60$ $p$-values) and randomly divide them into three blocks of equal size $L_1$, $L_2$ and $L_3$.

Do the same for the top 1% of the $p$-values and create $U_i$ ($i = 1, 2, 3$) – each containing 20 $p$-values. Randomly shuffle the rest of the $p$-values and bin them equally into $K = 200$ partitions $P_1, \ldots, P_{200}$; and finally construct

\[ P_1 \cup L_1, P_2 \cup L_2, P_3 \cup L_3, P_4, \ldots, P_{197} \cup U_1, P_{198} \cup U_2, P_{200} \cup U_3. \]

Figure 2 shows the distribution of $p$-values in the first and last three machines, which represents the active components in (8). By design, all the remaining 194 partitions have uniformly distributed $p$-values i.e. $d_l \equiv 1$. In what follows, we present a three-tier analysis pipeline:

Level 1. We compute the LP-Fourier coefficients $LP[j; F_B, \tilde{D}_l]$ in parallel mode at each of the computing nodes of the cluster. By combining all of them using Step 5, our algorithm yields the following full-data comparison density estimate for the distributed prostate data:

\[
\hat{d}(u; \Phi, F) = .75 \left[ 1 + 0.0589 \text{Leg}_6 \left( F_B(u; \hat{\gamma} = .861, \hat{\beta} = .862) \right) \right]
\]

\[ u^{-1.38}(1 - u)^{-1.37}, \quad 0 < u < 1. \]

Although it is self-evident, it is important to point out that, irrespective of the partitioning scheme, our algorithm is guaranteed to reproduce the same full-data result. We can now use this estimate at each partition to identify the discoveries by using (1)–(2) at the desired fdr level (also see Appendix B2). For example, straightforward computation by applying (2) at $\alpha = .2$ finds 65 non-null genes (32 in the left tail and 33 in the right), spread over first and last three partitions. Whereas using two-sided $p$-values, $u_i = 2\Phi(-|z_i|)$ the smooth-BH procedure by plug-in $\hat{D}(u_{ih}) = \int_0^{u_{ih}} \hat{d}(v; \Phi, F)dv$ in (1) declares 63 genes (30 in the left tail and 33 in the right) to be significant.

Level 2. The goal here is to identify the signal-rich $p$-value sources using H-index. Recall the first and last three partitions contain the discoveries, and are thus expected to have large H-statistic (9) value. The top left panel of Figure 3 plots the pair of points ($l, H_l$), which we call ‘Control H-chart.’ Use this chart to monitor and quickly spot the ‘informative’ batch of $p$-values. For partitioned prostate data, as expected, the H-chart indicates that the first and last 3 groups are the primary source of discoveries (rejected null hypotheses).

Note that two different partitions may have similar value of H-indices, while the statistical characteristics might be very different. For example, in the prostate data the partitions $\{U_j\}_j$ contain genes with large positive t-statistic (upper-tail); in contrast, the partitions $\{L_j\}_j$ contain smallest (negative) t-statistic (lower-tail). Yet, as shown in Fig 3 (top left panel), the magnitudes of the H-statistic for both of the groups are comparable, in fact
Figure 2. (color online) Distribution of first and last three partitioned $p$-values for prostate data. Last row shows the $\hat{d}_l$ ($l = 1, 100, 200$) along with the full-data $\hat{d}$, computed using the superposition rule $\hat{d} = \sum_{l=1}^{K} \pi_l \hat{d}_l$.

almost equal! Naturally at this point, an investigator may be interested in more refined grouping of the $p$-value sources based on signal characteristics. To illustrate this point we introduce our second example.

Example 3.2: Generate 9800 samples from $\mathcal{N}(0,1)$ and divide them equally across $K = 200$ pieces. In each of the first four partitions we add 25 samples from $\mathcal{N}(2,1)$, and in the next four partitions we add samples generated from $\mathcal{U}(2,4)$. Thus, among 200 partitions, only the first eight contain the discoveries, albeit of two kinds. The H-chart shown the bottom left corner of Figure 3 correctly separates the eight informative $p$-value sources from the rest.
Define the principal signal-profile coordinate of the $i$th partition as by $\lambda_j u_{ij}$ for $j = 1, \ldots, m$. The two-dimensional exploratory graph in the right panel of Figure 3 is formed using the points $(\lambda_1 u_{i1}, \lambda_2 u_{i2})$, for $i = 1, \ldots, k$ by taking the dominant two terms of the SVD. Different partitions or groups are displayed as points that allow us to separate the groups according to the statistical nature of signals.

**Remark 3.1 (Efficiency and accuracy dilemma):** (i) the whole analysis can be done without moving any $p$-values across the data silos – a cost-effective and accelerated computation. (ii) The proposed decentralised technique is an exact method. The Superposition principle along with Theorems 2.3–2.5 should be interpreted as identities that hold for any arbitrary partitions (partition-invariance), i.e. irrespective of how you break $N$ hypotheses into $K$ parts!
4. Conclusions

Without losing the organic character of the general theory of nonparametric multiple testing proposed in Mukhopadhyay (2016), we successfully derived its non-trivial extension that allows transition from centralised to decentralised capability to scale for massive datasets with billions of tests. This shift is necessary in order to fully realise the potential for ever-increasing amounts of distributed big datasets, which has become the de facto standard in science, industry, and business. The core principles and ideas presented in this paper provide a comprehensive framework by embracing small (centralized) and massive (distributed) scale multiple testing cultures in a way that is intuitive and easy-to-implement; as a result, they have the potential to radically simplify theory, practice, and teaching. Prostate cancer data and simulated examples are used to illustrate the main steps (and more importantly the interpretations) of our algorithm. Obviously more complicated and large datasets could be used, but this should suffice to get the point across.

Notes

1. BH (Benjamini and Hochberg 1995) and HC (Donoho and Jin 2004) procedures start by ordering the $p$-values from smallest to largest incurring at least $O(N \log N)$ computational cost and other method like local fdr Efron, Storey, and Tibshirani is of even greater complexity ($N^2$), thereby making legacy multiple testing algorithms infeasible for such massive scale inference problems.

Acknowledgments

The author would like to thank two anonymous reviewers for their helpful and constructive comments that greatly contributed to improving the final version of the paper.

Disclosure statement

No potential conflict of interest was reported by the author.

ORCID

Subhadeep Mukhopadhyay http://orcid.org/0000-0002-6841-0132

References


