BIOMETRICS 000, 000–000 DOI: 000
000 0000

Fast Approximation of Small p-values in Permutation Tests by Partitioning the Permutations

Brian D. Segal,* Thomas Braun, Michael Elliott, and Hui Jiang

Department of Biostatistics, University of Michigan
1415 Washington Heights, Ann Arbor, MI 48109-2029
**email:* bdsegal@umich.edu

SUMMARY: Researchers in genetics and other life sciences commonly use permutation tests to evaluate differences between groups. Permutation tests have desirable properties, including exactness if data are exchangeable, and are applicable even when the distribution of the test statistic is analytically intractable. However, permutation tests can be computationally intensive. We propose both an asymptotic approximation and a resampling algorithm for quickly estimating small permutation p-values (e.g. $< 10^{-6}$) for the difference and ratio of means in two-sample tests. Our methods are based on the distribution of test statistics within and across partitions of the permutations, which we define. In this article, we present our methods and demonstrate their use through simulations and an application to cancer genomic data. Through simulations, we find that our resampling algorithm is more computationally efficient than another leading alternative, particularly for extremely small p-values (e.g. $< 10^{-30}$). Through application to cancer genomic data, we find that our methods can successfully identify up- and down-regulated genes. While we focus on the difference and ratio of means, we speculate that our approaches may work in other settings.

KEY WORDS: Computational efficiency; Genomics; Multiple hypothesis tests; Resampling methods; Two-sample tests

1. Introduction and Motivation

Many researchers in the life sciences use permutation tests, for example, to test for differential gene expression (Doerge and Churchill, 1996, Morley et al., 2004, Stranger et al., 2005, 2007, Raj et al., 2014), and to analyze brain images (Nichols and Holmes, 2002, Bartra et al., 2013, Simpson et al., 2013). These tests are useful when the sample size is too small for large sample theory to apply, or when the distribution of the test statistic is analytically intractable. Permutation tests are also exact, meaning that they control the type I error rate exactly for finite sample size (Lehmann and Romano, 2006). However, permutation tests can be computationally intensive, especially when estimating small p-values for many tests. In this paper, we present computationally efficient methods for approximating small permutation p-values (e.g. $< 10^{-6}$) for the difference and ratio of means in two-sample tests, though we speculate that our methods will also work for other smooth function of the means.

We denote the two groups of sample data as $\boldsymbol{x} = (x_1, \ldots, x_{n_x})'$ and $\boldsymbol{y} = (y_1, \ldots, y_{n_y})'$, with respective sample sizes n_x and n_y. We denote the full data as $\boldsymbol{z} = (\boldsymbol{x}', \boldsymbol{y}')'$, with total sample size $N = n_x + n_y$. Writing $\boldsymbol{z} = (z_1, \ldots, z_N)'$, we have that $z_i = x_i, i = 1, \ldots, n_x$, and $z_{n_x+j} = y_j, j = 1, \ldots, n_y$. In our setting, z_i are scalar values for all $i = 1, \ldots, N$. We use π to denote a permutation of the indices of $\boldsymbol{z}$, i.e. $\pi : \{1, \ldots, N\} \to \{1, \ldots, N\}$ is a bijection, and we denote the permuted dataset corresponding to π as $\boldsymbol{z}^* = (z_1^*, \ldots, z_N^*)'$, where $z_{\pi(i)}^* = z_i, i = 1, \ldots, N$. We use the term *correspondence* throughout this paper, so for clarity, we define our use of the term in Definition 1.

Definition 1 (Correspondence): Let $\boldsymbol{z} = (z_1, \ldots, z_N)'$ be the N-dimensional vector of observed data, and let $\pi : \{1, \ldots, N\} \to \{1, \ldots, N\}$ be a bijection (permutation) of the indices of $\boldsymbol{z}$. We say that the N-dimensional vector $\boldsymbol{z}^* = (z_1^*, \ldots, z_N^*)'$ *corresponds* to permutation π if $z_{\pi(i)}^* = z_i$ for all $i = 1, \ldots, N$.

It will also be useful to write the permuted dataset as $\boldsymbol{z}^* = (\boldsymbol{x}^{*\prime}, \boldsymbol{y}^{*\prime})'$, where $\boldsymbol{x}^* = (z^*_1, \dots, z^*_{n_x})'$ and $\boldsymbol{y}^* = (z^*_{n_x+1}, \dots, z^*_N)'$ are the permuted group samples.

Let T be a test statistic, such that larger values are more extreme, and let $t = T(\boldsymbol{x}, \boldsymbol{y})$ be the observed test statistic. Similar to Lehmann and Romano (2006, p. 636), we denote the permutation p-value as $\hat{p} = \Pr(T \geqslant t|\boldsymbol{z}) = |\Psi|^{-1} \sum_{\pi \in \Psi} I[T(\boldsymbol{x}^*, \boldsymbol{y}^*) \geqslant t]$, where Ψ is the set of all permutations of the indices of $\boldsymbol{z}$, $|\Psi| = N!$ is the number of elements in Ψ, I is an indicator function, and for each π, $(\boldsymbol{x}^{*\prime}, \boldsymbol{y}^{*\prime})'$ is the corresponding permuted dataset. The randomization hypothesis (Lehmann and Romano, 2006, Definition 15.2.1) asserts that under the null hypothesis, the distribution of T is invariant under permutations $\pi \in \Psi$. This allows, for example, for the null hypothesis $H_0 : z_i \overset{\text{iid}}{\sim} P, i = 1, \dots, N$, or more generally, for exchangeability, $H_0 : P(Z_1 = z_1, \dots Z_N = z_n) = P(Z_1 = z^*_1, \dots, Z_N = z^*_N)$ for all permuted datasets $\boldsymbol{z}^*$.

The set Ψ is typically too large to evaluate fully, so Monte Carlo methods are usually used to approximate $\hat{p}$. When resampling with replacement, also known as simple Monte Carlo resampling, the Monte Carlo estimate of $\hat{p}$ is $\tilde{p} = (B + 1)^{-1} \left(\sum_{b=1}^{B} I\left[T_b \geqslant t\right] + 1 \right)$, where B is the number of resamples, and $T_b = T(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for $(\boldsymbol{x}^{*\prime}, \boldsymbol{y}^{*\prime})'$ corresponding to the b^{th} randomly sampled permutation π_b. We refer to the above estimate as the adjusted $\tilde{p}$, because it adjusts the estimate to ensure it stays within its nominal level (Lehmann and Romano, 2006). However, for simplicity and to be consistent with other computationally efficient methods, particularly that of Yu et al. (2011), we use the unadjusted $\tilde{p}$, in which we remove the '+1' from the numerator and denominator.

While there may be many reasons for obtaining accurate small p-values, perhaps they are most often obtained in multiple testing settings, which are common in genetics. For example, in the analysis we present in Section 6, we analyze 15,386 genes for differential expression. With a Bonferroni correction and a type I error rate of $\alpha = 0.05$, to control the family-wise error rate (FWER), we would need to estimate p-values $< 0.05/15,386 \approx 3.25 \times 10^{-6}$. While one might want to use a different correction to control the FWER, false discovery rate (FDR), or other criteria, we would still need to calculate small p-values before implementing typical step-up or step-down procedures (for example, Holm (1979) to control FWER, or Benjamini and Hochberg (1995) to control FDR). These p-values, in combination with content area expertise and other statistical quantities, such as effect size, can be useful for prioritizing genes for further laboratory and statistical analysis.

As noted by Kimmel and Shamir (2006) and Yu et al. (2011), with simple Monte Carlo resampling, to estimate p-values on the order of $\hat{p} = 10^{-6}$ with a precision of $\sigma_{\hat{p}} = \hat{p}/10$, we need on the order of $B = 10^8$ resamples when using simple Monte Carlo resampling. For example, to separately estimate 5,000 p-values that are each on the order of 10^{-6}, we would need a total of $5,000 \times 10^8 = 5 \times 10^{11}$ resamples.

Several researchers have developed methods for reducing the computational burden of permutation tests, including Robinson (1982), Mehta and Patel (1983), Booth and Butler (1990), Kimmel and Shamir (2006), Conneely and Boehnke (2007), Li et al. (2008), Han et al. (2009), Knijnenburg et al. (2009), Pahl and Schäfer (2010), Zhang and Liu (2011), Jiang and Salzman (2012), and Zhou and Wright (2015). For comparisons with our method, we focus on the stochastic approximation Monte Carlo (SAMC) algorithm developed by Liang et al. (2007) and tailored to p-value estimation by Yu et al. (2011). Of the available methods, we found that SAMC was the most appropriate comparison, because: 1) we could directly apply it to the test static in our motivating application (see Section 6), 2) it is intended for very small p-values, and 3) it does not require derivations, so is more likely to be used in practice.

In this article, we propose alternative methods for quickly approximating small permutation p-values for the difference and ratio of the means in two-sample tests. Our approaches partition the permutations such that $\tilde{p}$ has a predictable trend across the partitions. Taking advantage of this trend, we develop both a closed form asymptotic approximation to the permutation p-value, as well as a computationally efficient resampling algorithm.

We find through simulations that our resampling algorithm is more computationally efficient than the SAMC algorithm, which in turn is 100 to 500,000 times more computationally efficient than simple Monte Carlo resampling (Yu et al., 2011). However, SAMC is a more general algorithm and can be used for a greater variety of statistics. The increase in efficiency is most notable for our algorithm when estimating extremely small p-values (e.g. $< 10^{-30}$). Our asymptotic approximation tends to be less accurate than our resampling algorithm but does not require resampling.

Before presenting our methods, we briefly explain the underlying properties that make them possible. The two basic components underlying our methods are 1) the partitions, which we define, and the distribution of permutations across these partitions, and 2) the limiting behavior of test statistics within each partition, and the trend in p-values across the partitions. We address the first component in Section 2 and the second in Section 3.

In Section 4, we introduce methods for estimating permutation p-values that take advantage of the properties discussed in Sections 2 and 3. In Section 5, we investigate the behavior of these methods through simulations and compare against the SAMC algorithm (additional simulations and comparisons against other methods are in the Web Appendices). Then in Section 6, we use our proposed methods to analyze cancer genomic data. In Section 7, we end with a discussion of limitations and possible extensions. As noted under Supplementary material, we have implemented our methods in the R package `fastPerm`.

2. Partitioning the permutations

2.1 *Defining the partitions*

Let the smaller of the two sample sizes be $n_{\min} = \min(n_x, n_y)$. We define the distance between permutation π and the observed ordering of the indices $(1, 2, 3, \ldots, N)$ as the number of observations that are exchanged between $\boldsymbol{x}$ and $\boldsymbol{y}$ under the action of π. To be precise, let $\omega(\pi)$ be the set of indices that π places in one of the first n_x positions, i.e. $\omega(\pi) = \{i \in \{1, \ldots, N\} : \pi(i) \leqslant n_x\}$. Then we define the distance, denoted as $d(\pi)$, between permutation π and the observed ordering, as

$$d\,(\pi) = n_x - |\omega(\pi) \cap \{1, 2, \ldots, n_x\}|. \qquad (1)$$

We define partition m, denoted as $\Pi(m)$, as the set of all permutations a distance of m away from the observed ordering, i.e. $\Pi(m) = \{\pi : d\,(\pi) = m\}$, $m = 0, 1, \ldots, n_{\min}$. As described below, our proposed methods focus on the permutation distributions of test statistics when resampling is restricted to permutations from a single partition.

To see why this definition of distance is useful, and to foreshadow our method, suppose that $\mu_x \neq \mu_y$, and note that as observations are exchanged between $\boldsymbol{x}$ and $\boldsymbol{y}$, the empirical distributions of the permuted samples $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ tend to become more similar. Consequently, test statistics that measure changes in the mean tend to become less extreme. For example, suppose that $n = n_x = n_y$ with n even, and let $\boldsymbol{z}^* = (\boldsymbol{x}^{*\prime}, \boldsymbol{y}^{*\prime})'$ be a permuted dataset corresponding to a permutation $\pi \in \Pi(n/2)$. Then half of the observations in $\boldsymbol{x}^*$ are from $\boldsymbol{x}$ and half are from $\boldsymbol{y}$, and the same is true for $\boldsymbol{y}^*$. Consequently, we would expect $\bar{x}^* \approx \bar{y}^*$, where $\bar{x}^*$ and $\bar{y}^*$ are the means of the permuted samples.

To make this explicit, and again assuming that $n = n_x = n_y$, let $\boldsymbol{\delta}_x^\pi = (\delta_{x,1}^\pi, \ldots, \delta_{x,n}^\pi)'$ and $\boldsymbol{\delta}_y^\pi = (\delta_{y,1}^\pi, \ldots, \delta_{y,n}^\pi)'$ be $n \times 1$ indicator vectors designating which observations are exchanged between $\boldsymbol{x}$ and $\boldsymbol{y}$ under the action of permutation π:

$$\delta^{\pi}_{x,i} = \begin{cases} 1 \text{ if } \pi(i) > n \\ 0 \text{ if } \pi(i) \leqslant n \end{cases}, i = 1, \ldots, n, \qquad \delta^{\pi}_{y,j} = \begin{cases} 1 \text{ if } \pi(n+j) \leqslant n \\ 0 \text{ if } \pi(n+j) > n \end{cases}, j = 1, \ldots, n.$$

Under the action of permutation π, $\bar{x}^* = n^{-1}\left[(\mathbf{1} - \boldsymbol{\delta}^{\pi}_x)'\boldsymbol{x} + \left(\boldsymbol{\delta}^{\pi}_y\right)'\boldsymbol{y}\right]$, where $\mathbf{1}$ is an $n \times 1$ vector of ones. Assuming uniform distribution of the permutations π, $\mathbb{E}\left[\boldsymbol{\delta}^{\pi}_x | \pi \in \Pi(m)\right] = (m/n)\mathbf{1}$, an $n \times 1$ vector with all elements equal to m/n. Consequently, $\mathbb{E}[\bar{x}^* | \pi \in \Pi(m), \boldsymbol{x}, \boldsymbol{y}] = \bar{x} + (m/n)(\bar{y} - \bar{x})$ and $\mathbb{E}[\bar{y}^* | \pi \in \Pi(m), \boldsymbol{x}, \boldsymbol{y}] = \bar{y} + (m/n)(\bar{x} - \bar{y})$.

Then, for example, with the test statistic $T = \bar{x} - \bar{y}$, we have that $\mathbb{E}[T(\boldsymbol{x}^*, \boldsymbol{y}^*) | \pi \in \Pi(m), \boldsymbol{x}, \boldsymbol{y}] = (\bar{x} - \bar{y})(1 - 2m/n)$, where $\boldsymbol{x}^*, \boldsymbol{y}^*$ are the permuted samples corresponding to a permutation $\pi \in \Pi(m)$, $m = 0, \ldots, n$. This shows that the expected value of T is zero when for both $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ half of the observations are from $\boldsymbol{x}$ and half are from $\boldsymbol{y}$, i.e. in the $m = n/2$ partition. Similarly, the magnitude of T is $|\bar{x} - \bar{y}|$ when either none or all of the observations are exchanged between $\boldsymbol{x}$ and $\boldsymbol{y}$ (partitions $m = 0$ and $m = n$, respectively). This example demonstrates that test statistics tend to be less extreme when the permuted group samples, $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$, each contain a mixture of elements from the observed group samples, $\boldsymbol{x}$ and $\boldsymbol{y}$. Similar results hold for unbalanced sample sizes.

2.2 *Distribution of the partitions*

Uniform sampling of the permutations π leads to a non-uniform distribution of the partitions $\Pi(m)$. The probability of drawing a permutation from partition m under uniform sampling, which we denote as $f(m), m = 1, \ldots, n_{\min}$, is given by

$$\begin{aligned} f(m) &\propto |\Pi(m)| & (\pi \sim \text{Uniform}) \\ &= \binom{n_x}{m}\binom{n_y}{m}, \end{aligned}$$

where the last line follows directly from the definition of $\Pi(m)$. The normalizing constant is $\sum_{j=0}^{n_{\min}} \binom{n_x}{j}\binom{n_y}{j} = \binom{N}{n_{\min}}$, so

$$f\,(m) = \binom{N}{n_{\min}}^{-1} \binom{n_x}{m}\binom{n_y}{m}. \tag{2}$$

As described in Section 4, in our proposed methods, we use f to weight the partition-specific p-values in order to obtain an overall p-value.

We note that in practice, directly using (2) to calculate $f(m)$ is not possible for large n_x and n_y, because the binomial coefficients become too large to represent on most computers. However, by noting the relationship between the gamma function and factorials, we can compute (2) for large sample sizes with the equivalent form:

$$\begin{aligned} f\,(m) = \exp\{&\log\Gamma(n_x+1) - \log\Gamma(n_x - m + 1) \\ &+ \log\Gamma(n_y+1) - \log\Gamma(n_y - m + 1) - 2\log\Gamma(m+1) \\ &- \log\Gamma(N+1) + \log\Gamma(N - n_{\max} + 1) + \log\Gamma(n_{\max}+1)\}, \end{aligned}$$

where $\log\Gamma$ is the log gamma function.

3. Trend in p-values across the partitions

In this section, we describe the trend in p-values across the partitions both with asymptotic and simulated results. The results described in this section are given in greater detail in Web Appendix A and are the basis for our proposed methods.

Let T be a two-sided test statistic that is a function of the means, such that larger values are more extreme. In particular, we study $T = |\bar{x} - \bar{y}|$ and $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$. T is a random variable, and we could calculate its value for all permutations of the data to get its permutation distribution.

We use two notations for the arguments to T: $T(\boldsymbol{x}, \boldsymbol{y})$ and $T(m)$. $T(\boldsymbol{x}, \boldsymbol{y})$ denotes the test statistic computed with data $\boldsymbol{x}, \boldsymbol{y}$, e.g. $T(\boldsymbol{x}, \boldsymbol{y}) = |\bar{x} - \bar{y}|$, and $T(m)$ denotes the test statistic computed with some permuted dataset $\boldsymbol{z}^*$, where $\boldsymbol{z}^*$ corresponds to a permutation

$\pi \in \Pi(m)$. This notation facilitates further analysis in Web Appendix A. We note that $\Pr(T(m) > t|\boldsymbol{z}) = \Pr(T(\boldsymbol{x}^*, \boldsymbol{y}^*) > t|\boldsymbol{z}, \pi \in \Pi(m))$, i.e., $T(m) = T(\boldsymbol{x}^*, \boldsymbol{y}^*)$ restricted to permutations in partition m. To be concrete, we could in principle compute the partition-specific permutation p-value, $\Pr(T(m) > t|\boldsymbol{z})$, as $\hat{p}(m) = |\Pi(m)|^{-1} \sum_{\pi \in \Pi(m)} I[T(\boldsymbol{x}^*, \boldsymbol{y}^*) \geqslant t]$, where for each $\pi \in \Pi(m)$, $(\boldsymbol{x}^{*\prime}, \boldsymbol{y}^{*\prime})'$ is the corresponding permuted dataset.

While we are primarily interested in two-sided statistics T in this paper, it helps to first note results for their one-sided counterparts, which we denote by R. In particular, $R = \bar{x} - \bar{y}$ and $R = \bar{x}/\bar{y}$. Similar to before, let $R(m) = R(\boldsymbol{x}^*, \boldsymbol{y}^*)$ restricted to permutations in partition m. As shown in Corollary 2 of Web Appendix A, under certain regularity conditions and sufficiently large sample sizes, $R(m) \sim N(\nu(m), \sigma^2(m))$, where $\nu(m)$ and $\sigma^2(m)$ are functions of the partition m as well as the sample means and variances of $\boldsymbol{x}$ and $\boldsymbol{y}$. The regularity conditions are standard assumptions for finite sample central limit theorems and the delta method, requiring that the tails of the distributions of the data are not too large and that the derivative of R exists at the means.

As described in Corollary 3 of Web Appendix A, a direct consequence of the limiting normality of $R(m)$ is that for n_x and n_y sufficiently large,

$$\Pr(T(m) \geqslant t|\boldsymbol{z}) \approx 2 - \Phi[\xi(\min\{m, 2m_{\max} - m\})] - \Phi\left[\xi^{\text{conj}}(\min\{m, 2m_{\max} - m\})\right], \quad (3)$$

where Φ is the standard normal cumulative density function (CDF), $m_{\max} = \arg\max_m f(m)$, and ξ and ξ^{conj} are functions of the partition m and data $\boldsymbol{z}$, whose forms depend on the statistic T. The functions ξ and ξ^{conj} are identical in form but reverse the role of the means of the permuted samples $\bar{x}^*$ and $\bar{y}^*$. This accounts for the two-sided form of T. Equation 3 is the basis for our asymptotic approximation, which is described in Section 4.1.

The proof of (3) involves the fact that $\Pr(T(m) \geqslant t|\boldsymbol{z})$, as a function of m, is approximately symmetric about $m_{\max}$. This symmetry is exact when $n_x = n_y$ and less accurate as the group sample sizes become imbalanced. Consequently, the accuracy of the approximation in (3) is best for equal group sample sizes and worsens as the group sample sizes become more imbalanced.

The result in (3) and the form for ξ and ξ^{conj} shown in Web Appendix A for $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$ give the smooth pattern shown in Figure 1 for $n_x = n_y = 100, \mu_x = \sigma_x^2 = 4$, and $\mu_y = \sigma_y^2 = 2$. In the case where $n_x \neq n_y$, the center of the trend shifts but is otherwise similar.

The smooth trend shown in Figure 1 is primarily an observation, though it holds with striking similarity for both $T = |\bar{x} - \bar{y}|$ and $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$ for a wide range of group sample sizes and parameter values. This observation is the basis for our resampling algorithm described in Section 4.2.

Figure 2 shows simulated results with $B = 10^3$ resamples within each partition for data coming from the following distributions with $n_x = n_y = 100$: Poisson with rates $\lambda_x = 4$ and $\lambda_y = 2$; exponential with rates $\lambda_x = 2$ and $\lambda_y = 1$; log normal with means $\mu_x = 2$ and $\mu_y = 1$ and variances $\sigma_x^2 = \sigma_y^2 = 1$, where μ and σ^2 are the means and variances of the log; and negative binomial with size $r_x = r_y = 3$ and probability of success $p = r/(r + \mu)$, where the means are $\mu_x = 4$ and $\mu_y = 2$. For visual comparison between theoretical and simulated results, Figure 1b shows the theoretical values cut off at 10^{-3}.

Note that the p-value for the $m = 0$ partition is always 1, as the only permutation in that partition is the observed test statistic. The same holds for partition $m = n_{\text{min}}$ when $n_x = n_y$.

[Figure 1 about here.]

[Figure 2 about here.]

4. Proposed methods

In this section, we propose two methods for approximating small permutation p-values: 1) a closed-form asymptotic approximation, and 2) a computationally efficient resampling algorithm. First, we note that we can express the permutation p-value as

$$\Pr(T \geqslant t|\boldsymbol{z}) = \sum_{m=0}^{n_{\min}} \Pr\left(T(m) \geqslant t\,|\boldsymbol{z}\right) f\left(m\right). \tag{4}$$

Both the asymptotic and resampling-based approaches involve approximations for the $\Pr\left(T(m) \geqslant t\,|\boldsymbol{z}\right)$ terms in (4). The asymptotic approach uses (3) to approximate these terms, whereas the resampling algorithm uses the trend across the partitions to predict the terms.

If multiplicity corrections are needed, researchers can apply step-up or step-down procedures to the p-values produced by our method (e.g. Holm (1979) to control FWER, or Benjamini and Hochberg (1995) to control FDR).

4.1 *Asymptotic approximation*

Our asymptotic approximation to the permutation p-value is given by $\hat{p}_{\text{asym}} = \sum_{m=0}^{n_{\min}} h(m)f(m)$, where $f(m)$ is given by (2) and

$$h(0) = 1$$

$$h(m) = 2 - \Phi\left[\xi(\min\{m, 2m_{\max} - m\})\right] - \Phi\left[\xi^{\text{conj}}(\min\{m, 2m_{\max} - m\})\right], m \in [1, n_{\min} - 1]$$

$$h(n_{\min}) = \begin{cases} 1 & \text{if } n_x = n_y \\ 2 - \Phi\left[\xi(\min\{m, 2m_{\max} - m\})\right] - \Phi\left[\xi^{\text{conj}}(\min\{m, 2m_{\max} - m\})\right] & \text{otherwise} \end{cases}$$

To see why $h(0) = 1$ always and $h(n_{\min}) = 1$ when $n_x = n_y$, note that the p-value is always 1 in the $m = 0$ partition, because this partition only contains the observed permutation. The same is true for the $n_{\min}$ partition when $n_x = n_y$, as T is a two-sided statistic.

Regarding notation, we use a hat in $\hat{p}_{\text{asym}}$ as opposed to a tilde to emphasize that we are not using Monte Carlo methods.

4.2 *Resampling algorithm*

As noted in Section 3, we could in principle estimate each $\Pr(T(m) \geqslant t|\boldsymbol{z})$ term in (4) with Monte Carlo methods, but this would be more computationally intensive than directly estimating $\Pr(T \geqslant t|\boldsymbol{z})$ without conditioning on the partition. This is because for small p-values, $\Pr(T(m) \geqslant t|\boldsymbol{z})$ terms for m near $m_{\max}$ (the middle partition when $n_x = n_y$) are very small, so we would need to use an extremely large number of resamples to estimate these values (e.g. see Figure 1a).

However, by taking advantage of the trend in p-values across the partitions, we can avoid directly calculating $\Pr(T(m) \geqslant t|\boldsymbol{z})$ for m near $m_{\max}$. Instead, we use simple Monte Carlo resampling to estimate $\Pr(T(m) \geqslant t|\boldsymbol{z})$ sequentially for $m = 1, 2, \ldots, m_{\text{stop}}$, where m_{stop} is the stopping partition, which, as described below, is determined dynamically. We then use a Poisson model to predict the $\Pr(T(m) \geqslant t|\boldsymbol{z})$ terms for the remaining partitions (as well as for partitions $m = 1, \ldots, m_{\text{stop}}$) under the assumption that the log of the partition-specific p-values is linear in m.

We then take a weighted sum across the predicted partition-specific p-values, as in (4), to obtain an overall p-value. We denote the resulting p-value as $\tilde{p}_{\text{pred}}$, where the tilde emphasizes the use of Monte Carlo methods and the subscript emphasizes that the estimate is based on predicted counts within each partition.

As described in Algorithm 1, we set the number of Monte Carlo resamples within partitions at B_{pred} (e.g. we use $B_{\text{pred}} = 10^3$) and estimate $\Pr(T(m) > t|\boldsymbol{z})$ for $m = 1, \ldots, m_{\text{stop}}$, where m_{stop} is the first partition in which none of the resampled statistics are larger than the observed statistic. We stop at partition m_{stop} because the exponential decrease in p-values across the partitions, shown in Figure 1a, makes it nearly certain that we would not obtain a p-value greater than zero in partitions larger than m_{stop} using only $B_{\text{pred}} = 10^3$ resamples. In other words, it would be a waste of resources to continue sampling from additional partitions. Furthermore, since the trend is symmetric about $m_{\max}$, we can estimate the p-values in partitions $m = m_{\max} + 1, \ldots, n_{\min}$ using the p-values in partitions $m = 1, \ldots, m_{\max}$.

Regarding the Poisson model, this is a natural choice for count data (the number of resampled statistics larger than the observed statistic within each partition) and also enforces a log-linear trend. Furthermore, we found that Poisson regression worked best in the simulations. In addition to our current approach of using a slope and intercept term in the Poisson model, we experimented with using higher order polynomials and B-splines and selecting the optimal order or degrees of freedom based on AIC. However, we found that this approach was too sensitive to noise in the data and sometimes gave highly erroneous results (e.g. p-values > 1).

In Algorithm 1, we represent vector indices by square brackets $[\cdot]$ and begin the index at zero because our partitions begin at $m = 0$. We use the vector $\boldsymbol{c}$ to store the count of permuted test statistics in each partition that are as large or larger than the observed test statistic as obtained with simple Monte Carlo resampling and use $\boldsymbol{c}_{\text{pred}}$ to store predicted counts based on a fitted model. We use B_{pred} to denote that number of resamples within each partition.

Algorithm 1 $\tilde{p}_{\text{pred}}$

```
set $m \leftarrow 1$ and $\boldsymbol{c}[0] \leftarrow B_{\text{pred}}$
while ($m \leqslant m_{\text{max}}$ and $\boldsymbol{c}[m-1] > 0$) do
    for $b = 1, \ldots, B_{\text{pred}}$, sample $\pi_b \in \Pi(m)$ uniformly and calculate $T_b(m) = T(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for $\boldsymbol{x}^*, \boldsymbol{y}^*$ corresponding to $\pi_b$
    set $\boldsymbol{c}[m] \leftarrow \sum_b I[T_b(m) \geqslant t]$ and update $m \leftarrow m + 1$
end while
set $m_{\text{stop}} \leftarrow m - 1$ and $m_{\text{reg}} \leftarrow \max_m \{m \in \{1 \ldots, m_{\text{max}}\} : \boldsymbol{c}[m] > 0\}$
regress $\boldsymbol{c}[0 : m_{\text{reg}}]$ on $(0, \ldots, m_{\text{reg}})$ using a Poisson model with slope and intercept terms
predict $\boldsymbol{c}_{\text{pred}}$ for $m = 1, \ldots, n_{\text{min}}$ with fitted model, s.t. $\boldsymbol{c}_{\text{pred}}$ is symmetric about $m_{\text{max}}$
set $\boldsymbol{c}_{\text{pred}}[0] \leftarrow B_{\text{pred}}$, and if $n_x = n_y$, then set $\boldsymbol{c}_{\text{pred}}[n_x] \leftarrow B_{\text{pred}}$
return $\tilde{p}_{\text{pred}} \equiv (1/B_{\text{pred}}) \sum_{m=0}^{n_{\text{min}}} \boldsymbol{c}_{\text{pred}}[m] f(m)$
```

Our proposed algorithm runs in $O(B_{\text{pred}} m_{\text{stop}})$ time. As described in Web Appendix F, we provide functions for estimating m_{stop}, and thus run-time, prior to running the algorithm.

5. Simulations

To investigate the behavior of our proposed methods, we conducted simulations with the statistics $T = |\bar{x} - \bar{y}|$ and $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$. Given the extremely small p-values in our simulations, it was not feasible to compute the true permutation p-values for comparison. Instead, we used asymptotically equivalent p-values and large sample sizes.

In Web Appendix C, we show results from additional simulations for 1) small sample sizes, and 2) data generated under the null hypothesis, in which case we approximated the true permutation p-value with simple Monte Carlo resampling, and 3) data generated as Gamma random variables. In Web Appendix D, we also show simulations with the moment-corrected correlation (MCC) method of Zhou and Wright (2015) using the statistic $T = |\bar{x} - \bar{y}|$, and compare our method with saddle point approximations (Robinson, 1982) by analyzing two small datasets ($n_x = n_y = 8$ and $n_x = 7, n_y = 10$), also using the statistic $T = |\bar{x} - \bar{y}|$. In Web Appendix E, we show simulation results using our method with a studentized statistic to test null hypotheses regarding a single parameter as opposed to the full distribution, as described by Chung et al. (2013). The results in Web Appendices C and D show that the accuracy of our method is comparable to alternative methods, and the results in Web Appendix E show that by using a studentized statistic, our method can be extended to null hypotheses specifying equality in the means ($H_0 : \mu_x = \mu_y$), as opposed to equality in the entire distributions ($H_0 : P_x = P_y$).

5.1 *Difference in means*

In this section, we consider the test statistic $T = |\bar{x} - \bar{y}|$ with normally distributed data of equal variance. Since the t-test is asymptotically equivalent to the permutation test in this setting (Lehmann and Romano, 2006, p. 642-643), we used the t-test as a baseline for comparison. We simulated data with both equal and unequal sample sizes ($n_x = n_y$ and $n_x \neq n_y$). In both cases, we generated data $x_i, i = 1, \ldots, n_x$ and $y_j, j = 1, \ldots, n_y$ as realizations of the respective random variables $X_i \overset{\text{iid}}{\sim} N(\mu_x, 1)$ and $Y_j \overset{\text{iid}}{\sim} N(\mu_y, 1)$ for various parameter values. For each combination of parameter values, we generated 100 datasets.

For equal sample sizes, we set $n = n_x = n_y = 100$, 500, or 1,000. For unequal sample sizes, we set $n_y = 500$, and $n_x = 50$, 200, or 350. In both cases we set $\mu_y = 0$ and $\mu_x = 0.75$ or 1. For each dataset, we applied our methods and did a t-test with the `t.test` function in R (R Core Team, 2015) (two-sided with equal variance). For our resampling algorithm, we used $B_{\text{pred}} = 10^3$ resamples in each partition.

For comparison, we also ran the SAMC algorithm using the R package `EXPERT` written by Yu et al. (2011). We set the number of iterations (also resamples) in the initial round at 5×10^4 and the number of iterations in the final round at 10^6. Following the advice of Yu et al. (2011), we set the gain factor sequence to begin decreasing after the $1,000^{th}$ iteration, the proportion of data to be updated at each iteration at 0.05, and the number of regions at 101 for the initial run and 301 for the final run.

Results are shown in Figures 3 and 4. In the Figures, p_t denotes the p-value from a two-sided t-test with equal variance, and p denotes the p-value from either our methods or SAMC. The dashed line has a slope of 1 and intercept of 0, and indicates agreement between methods. The SAMC algorithm did not produce values for smaller p-values due to numerical problems, so these points are missing from Figures 3 and 4 (385 missing points in Figure 3, and 179 missing points in Figure 4). In order to estimate these points with the `EXPERT` implementation of the SAMC algorithm, we would need to increase the number of iterations.

[Figure 3 about here.]

[Figure 4 about here.]

As Figures 3 and 4 show, our resampling algorithm and asymptotic approximation are able to estimate extremely small p-values, which the SAMC algorithm is not able to estimate even though we set it to use approximately two orders of magnitude more resamples than our resampling algorithm. While our asymptotic approximation has less variance than our resampling algorithm, the asymptotic approximation appears to have more bias. We note that the scales are not the same in Figures 3 and 4, but in both cases, the p-values are smaller than what would typically be estimated with resampling methods.

Figures 3b and 4b also demonstrate that our algorithm uses fewer permutations when estimating smaller p-values than when estimating larger p-values. This occurs because the trend in partition-specific p-values across the partitions tends to be steeper for smaller overall p-values, which leads to earlier stopping times.

5.2 *Ratio of means*

In this section, we consider the test statistic $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$, both for $n_x = n_y$ and $n_x \neq n_y$. We generated data $x_i, i = 1, \ldots, n_x$ and $y_j, j = 1, \ldots, n_y$ as realizations of the respective random variables $X_i \overset{\text{iid}}{\sim} \text{Exp}(\lambda_x)$ and $Y_j \overset{\text{iid}}{\sim} \text{Exp}(\lambda_y)$, where $\text{Exp}(\lambda)$ is an exponential distribution with rate λ, i.e. $\mathbb{E}[X_i] = 1/\lambda_x$. We chose this setup because 1) having data with non-negative support ensures non-zero denominators in the ratio statistic, and 2) the resulting ratio statistic follows a beta prime distribution, also called a Pearson type VI distribution (Johnson et al., 1995, p. 248), which provides an approximate baseline for comparison (see Web Appendix B).

For equal sample sizes, we set $n = n_x = n_y = 100$, 500, or 1,000. For unequal sample sizes, we set $n_y = 500$, and $n_x = 50$, 200, or 350. In both cases we set $\lambda_x = 1$ and $\lambda_y = 1.75$ or 2.25. For all parameter combinations, we generated 100 datasets.

For each dataset, we applied our methods and computed the p-value from the beta prime distribution. For our resampling algorithm, we used $B_{\text{pred}} = 10^3$ resamples in each partition. We also computed p-values using the delta method (see Web Appendix G) and ran the SAMC algorithm with the same specifications as described in Section 5.1.

Results are shown in Figures 5 and 6. In the Figures, p_β denotes the p-value from the beta prime distribution, and p denotes the p-value from either our methods, the delta method (see Web Appendix G), or SAMC. The dashed line has a slope of 1 and intercept of 0, and indicates agreement between methods. As before, the SAMC algorithm did not produce values for smaller p-values, so these points are missing from Figures 5 and 6 (246 missing points in Figure 5, and 33 missing points in Figure 6).

[Figure 5 about here.]

[Figure 6 about here.]

As Figures 5 and 6 show, both our resampling algorithm and asymptotic approximation appear to have more bias in this setting than for the difference in means, though in this case, the asymptotic approximation is biased downward instead of upward. Our resampling algorithm tends to be biased upward.

As before, the SAMC algorithm had trouble estimating extremely small p-values with the number of iterations we allowed it. In the case of equal sample sizes, the SAMC algorithm began to have problems for p-values around 10^{-30}. In the case of unequal sample sizes, the SAMC algorithm appears to have performed similarly to our resampling algorithm, albeit with one to two orders of magnitude more resamples. Figures 5 and 6 also show that p-values from the delta method (see Web Appendix G) are not reliable, even for large sample sizes.

Similar to Section 5.1, Figures 5b and 6b show that our resampling algorithm uses fewer resamples for smaller p-values. Also, as before, the scale of the p-values is not the same in Figures 5 and 6, but in both cases, they are smaller than what would typically be estimated with resampling methods.

6. Application to cancer genomic data

To further demonstrate our methods, we analyzed RNA-seq data collected as part of The Cancer Genome Atlas (TCGA) (National Cancer Institute, 2015). In particular, we were interested in identifying genes that were differentially expressed in two different types of lung cancers: lung adenocarcinoma (LUAD), and lung squamous cell carcinoma (LUSC).

We downloaded normalized gene expression data from the TCGA data portal. As described by TCGA, to produce the normalized gene expression data, tissue samples from patients with LUSC and LUAD were sequenced using the Illumina RNA Sequencing platform. The raw sequencing reads from all patient samples were processed and analyzed using the SeqWare Pipeline 0.7.0 and MapspliceRSEM workflow 0.7 developed by the University of North Carolina. Sequencing reads were aligned to the human reference genome using MapSplice (Wang et al., 2010), and gene level expression values were estimated using RSEM (Li and Dewey, 2011) with gene annotation file GAF 2.1. For each sample, RSEM gene expression estimates were normalized to set the upper quartile count at 1,000 for gene level estimates. For the analyses in this section, we used the normalized RSEM gene expression estimates.

For both LUAD and LUSC, TCGA contains normalized expression estimates for 20,531 genes (the same genes for both cancers). There were 548 subjects with LUAD observations, and 541 with LUSC observations. To ensure that our results would be biologically meaningful, we restricted our analysis to genes for which at least 50% of the subjects had expression levels above the 25^{th} percentile of all normalized gene expression levels (6.57). This reduced our analysis to 15,386 genes.

Let $P_{x,g}$ and $P_{y,g}$ be the underlying distributions that generated the normalized expression levels in LUAD and LUSC, respectively, for gene g. To test the two-sided hypothesis of $H_0 : P_{x,g} = P_{y,g}$ versus the alternative $H_1 : \mu_x/\mu_y \neq 1$, we used the fold-change statistic $T = \max(\bar{x}_g/\bar{y}_g, \bar{y}_g/\bar{x}_g)$. Here, μ_x and μ_y are the means of $P_{x,g}$ and $P_{y,g}$, respectively.

First, we conducted simple Monte Carlo permutation tests on all 15,386 genes with $B = 10^3$ resamples. This left us with 10,302 genes with p-values less than 10^{-3}, the minimum estimate possible with only $B = 10^3$ resamples. We then used our resampling algorithm to estimate p-values for the 10,302 genes that passed our preliminary screen.

Table 1 shows the results for the fifteen genes with the smallest p-values, as well as the deviance and AIC from the Poisson regression fit during the resampling algorithm. We report both the estimate from the initial, single run of our algorithm, as well as the $10^{th}, 50^{th}$, and 90^{th} quantiles from an additional 1,000 runs. Note that Table 1 reports the observed ratio of mean(LUAD)/mean(LUSC), not the max of the ratios that we used in the permutation test. Of the top 15 genes, none had elevated levels of LUAD. Point estimates for all genes are available as supplementary material.

Eleven of the these fifteen genes, shown in bold (*DSG3, KRT5, DSC3, CALM3, TP63, ATP1B3, KRT6B, TRIM29, PVRL1, FAT2*, and *KRT6C*), were also identified by Zhan et al. (2015) as being among the most effective genes for distinguishing between LUAD and LUSC. Like us, Zhan et al. (2015) used the TCGA dataset, though they based their analysis on the area under the curve from a Wilcoxon rank-sum test.

[Table 1 about here.]

We emphasize that in presenting Table 1, we are not trying to promote the use of p-values as the sole source of information for making scientific decisions, such as ranking the importance of genes. Instead, we present Table 1 and make comparisons with the findings of Zhan et al. (2015) as a way of verifying the reasonableness of our results. Zhan et al. (2015) used different methods to analyze the TCGA data, so we do not expect our results to be exactly the same, but it is encouraging that our results appear to agree to some extent.

We also want to point out that our resampling algorithm can approximate extremely small p-values, but that in doing so, there is a large amount variability in the estimates. However, we think these estimates could still be used as an approximation of the order of magnitude, and note that they would be infeasible to estimate with existing Monte Carlo methods, including the SAMC algorithm.

7. Discussion

As we have demonstrated through simulations and an application to cancer genomic data, our methods can quickly approximate small permutation p-values (e.g. $< 10^{-6}$) for two-sample tests, where the test statistic is the difference or ratio of means. The computational efficiency of our resampling algorithm is particularly notable when estimating extremely small p-values (e.g. $< 10^{-30}$).

As is suggested in the example of Section 2, our methods can only detect changes in the mean. If $P_x \neq P_y$ but $\mu_x = \mu_y$, then the statistics $T = |\bar{x} - \bar{y}|$ and $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$ cannot detect differences. We also note that while our development focuses on the null hypothesis $P_x = P_y$, the simulations in Appendix E suggest that our methods extend to less restrictive null hypotheses, such as those considered by Janssen (1997) and Chung et al. (2013).

As shown in the Section 5 and the Web Appendices, the accuracy of our resampling method is comparable to alternative methods, such as SAMC and MCC, though SAMC and MCC are applicable in situations where our methods are not. In particular, MCC can handle any statistic that can be expressed as, or is permutationally equivalent to, an inner product. In addition to these methods, researchers may want to consider the method of Fieller (1954) for obtaining confidence intervals for the ratio of means, and the approaches described by Cui and Churchill (2003) for using t-tests and ANOVA to analyze the mean log ratio.

While the reliability of our resampling algorithm will vary based on the empirical distribution of the data, in general, we recommend having at least 15-20 observations in each group for p-values near 1×10^{-6} and at least 70-90 observations in each group for p-values near 1×10^{-30} (see Web Appendix F). As demonstrated in Section 6, there can be considerable variability in estimating extraordinarily small p-values (e.g. 1×10^{-200}).

For these extraordinarily small p-values, we recommend that our method be used only to approximate the order of magnitude of the permutation p-value.

In choosing between our resampling algorithm and asymptotic approximation, we recommend using the resampling algorithm when possible for small p-values, as it appears to perform better in simulations. However, as demonstrated in the Web Appendix, our asymptotic method may be preferable for large p-values, as it appears to be more conservative under the null. Both approaches work best for equal sample sizes, and we suggest caution when using with small and highly imbalanced samples.

Depending on a researcher's needs, our algorithm could be useful as a fast approximation of small p-values. This might be helpful, for example, in a screening study involving many genes, in which a researcher wants to quickly get a sense for which genes have p-values that are likely to be below a small threshold. It might also be helpful as a preliminary analysis to approximate the order of magnitude of a p-value, which could help a researcher to determine whether it would be feasible to follow-up with other Monte Carlo methods, such as SAMC, and if so, how many iterations they would need to use. For some situations, such as our analysis in Section 6, this could save considerable time and resources.

We want to emphasize that our methods are most useful for approximating small permutation p-values. For large p-values, our resampling algorithm is less computationally efficient than simple Monte Carlo resampling. In the context of genomics data, before using our methods, we recommend that researchers use simple Monte Carlo resampling with a small number of resamples (e.g. 10^3) to identify which genes have p-values below a certain threshold (e.g. 10^{-3}). However, this is not a requirement.

This paper focuses on two-sample tests, and we plan to explore extensions to multiple samples in future work. As one way to handle multiple samples, we could conduct a union-intersection test (Casella and Berger, 2002, p. 380). For example, say we have k samples

$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$, and we wish to test the hypothesis $H_0 : \cap_{i \neq j} P_{x_i} = P_{x_j}$ versus the alternative $H_1 : \cup_{i \neq j} \mu_{x_i} \neq \mu_{x_j}$, where μ_i is the mean of P_{x_i}. Then we could use Algorithm 1 to compute p-values for all pairwise differences (or all pairwise ratios), and then take the minimum p-value. As another alternative, we could extend Algorithm 1 to use an omnibus statistic, similar to the ANOVA F-test, and use a multi-sample version of (2). For example, we might use $T = \sum_i n_i |\bar{x}_i - \bar{x}|/n$ where $\bar{x}_i$ and n_i are the mean and sample size, respectively, for group i, $\bar{x}$ is the overall mean, and $n = \sum_i n_i$. However, the extension of (2) to multiple samples is non-trivial. It is also unclear whether the p-values from the multi-sample case would follow the same trends across the partitions as in the two-sample case.

Returning to the two-sample case, while we have focused on the difference and ratio of the means, preliminary efforts to explain the nearly log-linear trend in p-values across the partitions suggests that the same pattern might hold for other smooth functions of the means. In future work, we plan to explore this further. We also plan to investigate potential diagnostics for assessing the reliability of the algorithm's output, possibly based on the AIC from the Poisson regression. Finally, we note that alternative Monte Carlo methods could be incorporated into our resampling algorithm. For example, the SAMC algorithm could be used in place of simple Monte Carlo resampling within each partition. This might further reduce run-time and increase accuracy.

8. Supplementary material

We have implemented our method in the R package `fastPerm` available at https://github.com/bdsegal/fastPerm. All code for the simulations and analyses in this paper are available at https://github.com/bdsegal/code-for-fastPerm-paper. Web Appendices referenced in Sections 1 and 3-7 are available with this paper at the Biometrics website on Wiley Online Library.

Acknowledgments

We thank the associate editor and two referees for their insightful comments and suggestions.

References

Bartra, O., McGuire, J. T., and Kable, J. W. (2013). The valuation system: a coordinate-based meta-analysis of BOLD fMRI experiments examining neural correlates of subjective value. *Neuroimage* **76,** 412–427.

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

Booth, J. G. and Butler, R. W. (1990). Randomization distributions and saddlepoint approximations in generalized linear models. *Biometrika* **77,** 787–796.

Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.

Chung, E., Romano, J. P., et al. (2013). Exact and asymptotically robust permutation tests. *The Annals of Statistics* **41,** 484–507.

Conneely, K. N. and Boehnke, M. (2007). So many correlated tests, so little time! rapid adjustment of p-values for multiple correlated tests. *The American Journal of Human Genetics* **81,** 1158–1168.

Cui, X. and Churchill, G. A. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome biology* **4,** 210.

Doerge, R. W. and Churchill, G. A. (1996). Permutation tests for multiple loci affecting a quantitative character. *Genetics* **142,** 285–294.

Fieller, E. C. (1954). Some problems in interval estimation. *Journal of the Royal Statistical Society. Series B (Methodological)* pages 175–185.

Han, B., Kang, H. M., and Eskin, E. (2009). Rapid and accurate multiple testing correction and power estimation for millions of correlated markers. *PLoS Genetics* **5,** 1000456.

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

Janssen, A. (1997). Studentized permutation tests for non-iid hypotheses and the generalized Behrens-Fisher problem. *Statistics & probability letters* **36,** 9–21.

Jiang, H. and Salzman, J. (2012). Statistical properties of an early stopping rule for resampling-based multiple testing. *Biometrika* **99,** 973–980.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1995). *Continuous univariate distributions*, volume 2. Wiley Series in Probability and Mathematical Statistics, 2nd edition.

Kimmel, G. and Shamir, R. (2006). A fast method for computing high-significance disease association in large population-based studies. *The American Journal of Human Genetics* **79,** 481–492.

Knijnenburg, T. A., Wessels, L. F., Reinders, M. J., and Shmulevich, I. (2009). Fewer permutations, more accurate p-values. *Bioinformatics* **25,** i161–i168.

Lehmann, E. L. and Romano, J. P. (2006). *Testing statistical hypotheses.* Springer Science & Business Media.

Li, B. and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* **12,** 1.

Li, Q., Zheng, G., Li, Z., and Yu, K. (2008). Efficient approximation of p-value of the maximum of correlated tests, with applications to genome-wide association studies. *Annals of Human Genetics* **72,** 397–406.

Liang, F., Liu, C., and Carroll, R. J. (2007). Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association* **102,** 305–320.

Mehta, C. R. and Patel, N. R. (1983). A network algorithm for performing Fisher's exact test in r×c contingency tables. *Journal of the American Statistical Association* **78,** 427–434.

Morley, M., Molony, C. M., Weber, T. M., Devlin, J. L., Ewens, K. G., Spielman, R. S., and Cheung, V. G. (2004). Genetic analysis of genome-wide variation in human gene expression. *Nature* **430,** 743–747.

National Cancer Institute (2015). The cancer genome atlas. http://cancergenome.nih.gov.

Nichols, T. E. and Holmes, A. P. (2002). Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human Brain Mapping* **15,** 1–25.

Pahl, R. and Schäfer, H. (2010). PERMORY: an LD-exploiting permutation test algorithm for powerful genome-wide association testing. *Bioinformatics* **26,** 2093–2100.

Raj, T., Rothamel, K., Mostafavi, S., Ye, C., Lee, M. N., Replogle, J. M., Feng, T., Lee, M., Asinovski, N., Frohlich, I., et al. (2014). Polarization of the effects of autoimmune and neurodegenerative risk alleles in leukocytes. *Science* **344,** 519–523.

R Core Team (2015). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Robinson, J. (1982). Saddlepoint approximations for permutation tests and confidence intervals. *Journal of the Royal Statistical Society. Series B (Methodological)* pages 91–101.

Simpson, S., Lyday, R., Hayasaka, S., Marsh, A., and Laurienti, P. (2013). A permutation testing framework to compare groups of brain networks. *Frontiers in Computational Neuroscience* **7,** 171.

Stranger, B. E., Forrest, M. S., Clark, A. G., Minichiello, M. J., Deutsch, S., Lyle, R., Hunt, S., Kahl, B., Antonarakis, S. E., Tavaré, S., et al. (2005). Genome-wide associations of gene expression variation in humans. *PLoS Genetics* **1,** e78.

Stranger, B. E., Forrest, M. S., Dunning, M., Ingle, C. E., Beazley, C., Thorne, N., Redon, R., Bird, C. P., de Grassi, A., Lee, C., et al. (2007). Relative impact of nucleotide and copy number variation on gene expression phenotypes. *Science* **315,** 848–853.

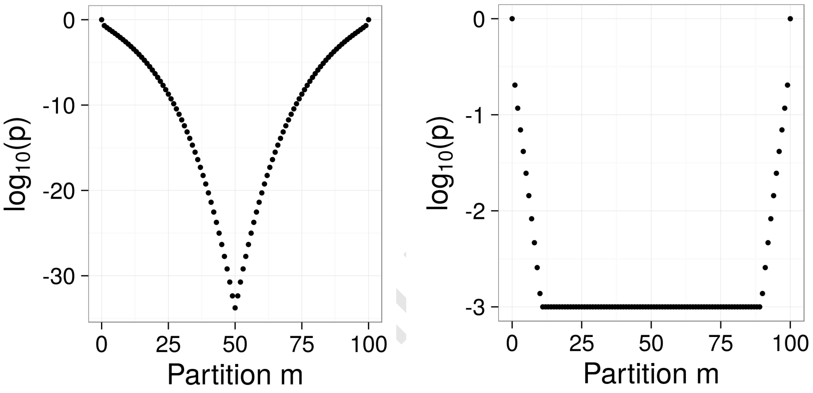
Wang, K., Singh, D., Zeng, Z., Coleman, S. J., Huang, Y., Savich, G. L., He, X., Mieczkowski, P., Grimm, S. A., Perou, C. M., et al. (2010). Mapsplice: accurate mapping of RNA-Seq reads for splice junction discovery. *Nucleic Acids Research* **38,** e178–e178.

Yu, K., Liang, F., Ciampa, J., and Chatterjee, N. (2011). Efficient p-value evaluation for resampling-based tests. *Biostatistics* pages 1–11.

Zhan, C., Yan, L., Wang, L., Sun, Y., Wang, X., Lin, Z., Zhang, Y., Shi, Y., Jiang, W., and Wang, Q. (2015). Identification of immunohistochemical markers for distinguishing lung adenocarcinoma from squamous cell carcinoma. *Journal of Thoracic Disease* **7,** 1398–1405.

Zhang, Y. and Liu, J. S. (2011). Fast and accurate approximation to significance tests in genome-wide association studies. *Journal of the American Statistical Association* **106,** 846–857.

Zhou, Y.-H. and Wright, F. A. (2015). Hypothesis testing at the extremes: fast and robust association for high-throughput data. *Biostatistics* pages 1–15.



(a) Theoretical trend

(b) Theoretical trend cut off at 10^{-3}

Figure 1: Theoretical trend in p-values with $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$ for $n_x = n_y = 100, \mu_x = \sigma_x^2 = 4$, and $\mu_y = \sigma_y^2 = 2$.

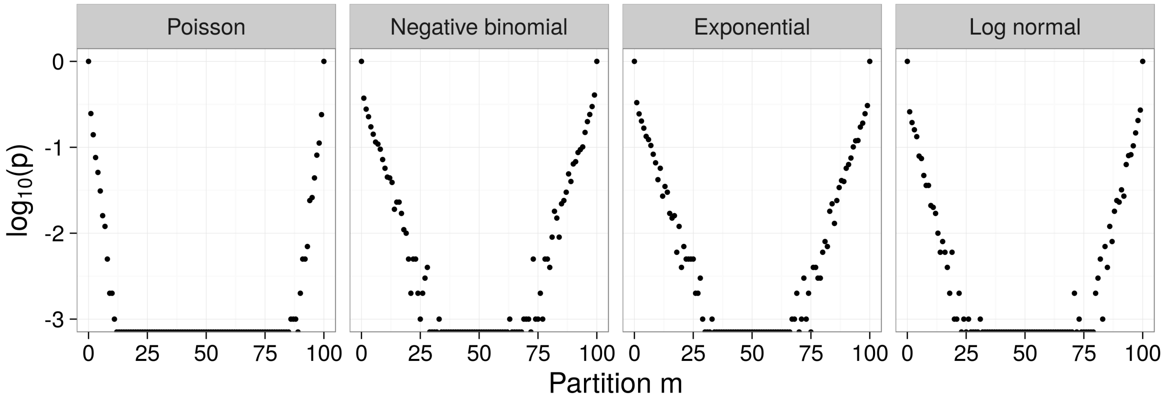
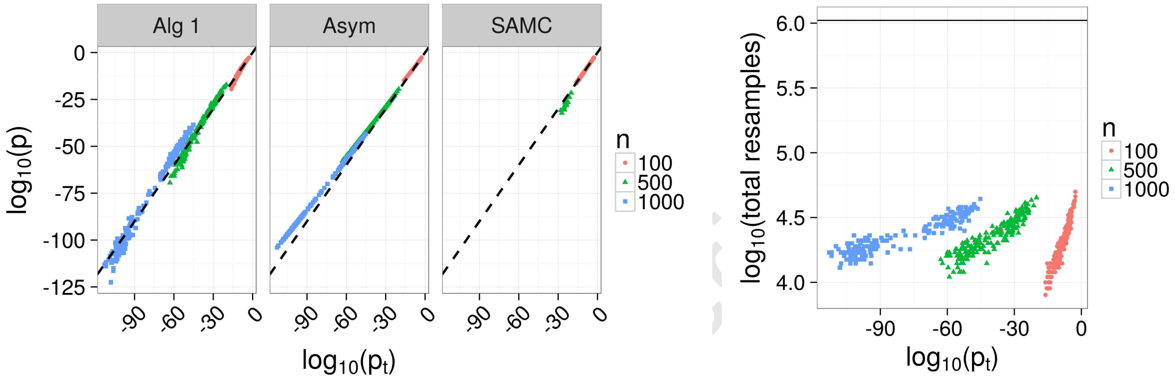


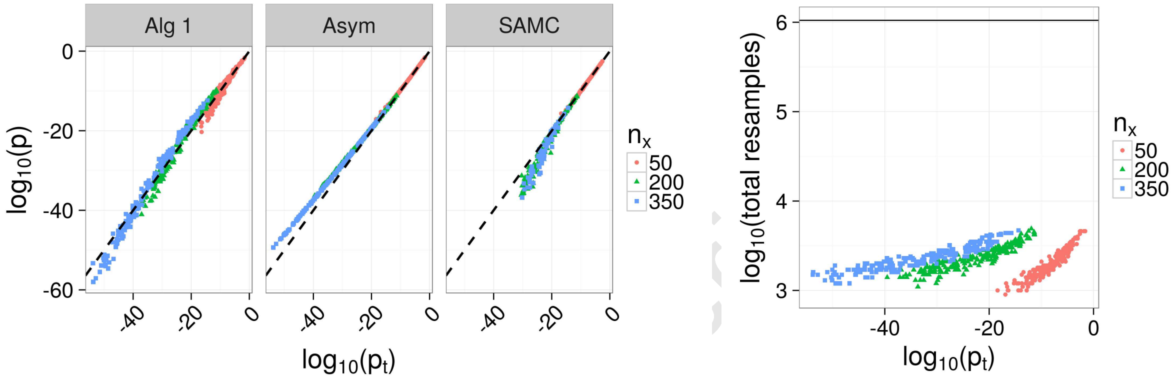
Figure 2: Simulated trend in p-values with $B = 10^3$ resamples within each partition and $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$



(a) p-values

(b) Number of resamples in Alg 1

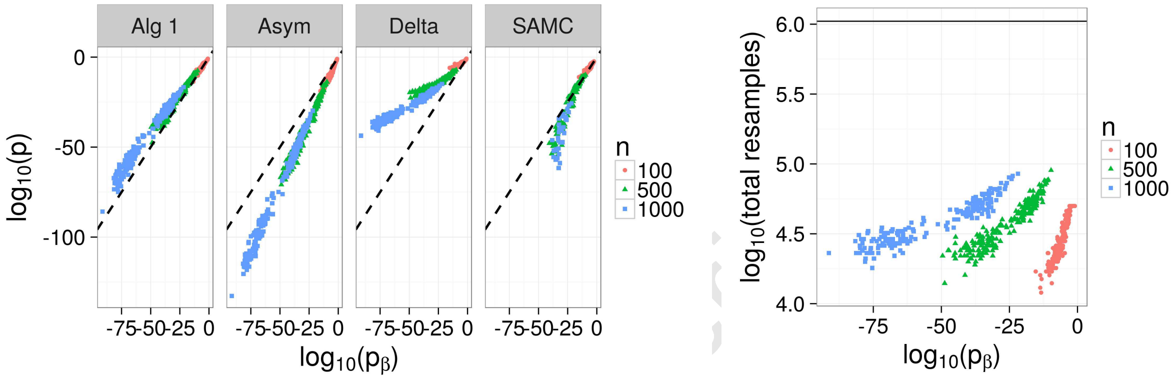
Figure 3: Simulation results using the statistic $T = |\bar{x} - \bar{y}|$ with equal sample sizes of $n = n_x = n_y = 100, 500, 1{,}000$. *Alg 1* is our resampling algorithm with $B_{\text{pred}} = 10^3$ resamples in each partition, *Asym* is our asymptotic approximation, *SAMC* is the SAMC algorithm, and p_t is a two-sided t-test with equal variance. The diagonal dashed line has slope of 1 and intercept of 0, and indicates agreement between methods. The horizontal line in 3b shows the number of iterations used in the SAMC algorithm (set in advance, and independent of p-value). The SAMC algorithm did not produce values for 385 tests (points missing). This figure appears in color in the electronic version of this article.



(a) p-values

(b) Number of resamples in Alg 1

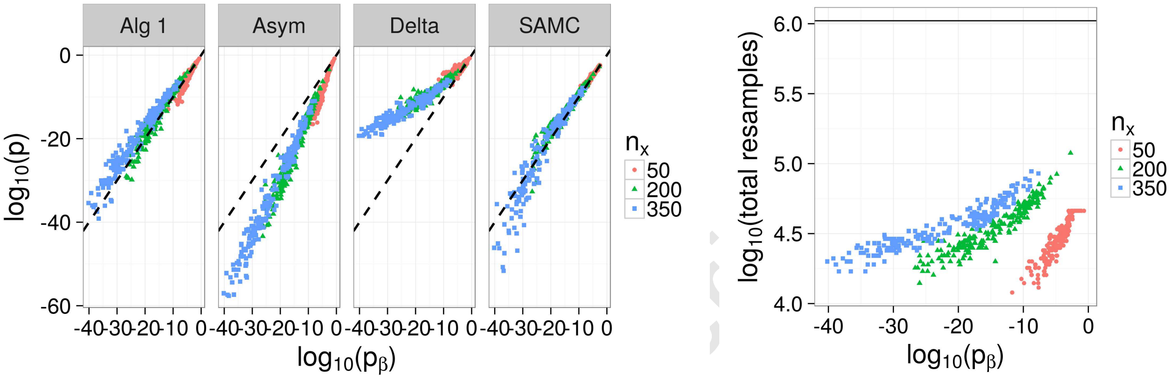
Figure 4: Simulation results using the statistic $T = |\bar{x} - \bar{y}|$ with unequal sample sizes, where $n_y = 500$ and $n_x = 50, 200, 350$. *Alg 1* is our resampling algorithm with $B_{\text{pred}} = 10^3$ resamples in each partition, *Asym* is our asymptotic approximation, *SAMC* is the SAMC algorithm, and p_t is a two-sided t-test with equal variance. The diagonal dashed line has slope of 1 and intercept of 0, and indicates agreement between methods. The horizontal line in 4b shows the number of iterations used in the SAMC algorithm (set in advance, and independent of p-value). The SAMC algorithm did not produce values for 179 tests (points missing). This figure appears in color in the electronic version of this article.



(a) p-values

(b) Number of resamples in Alg 1

Figure 5: Simulation results using the statistic $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$ with equal sample sizes of $n = n_x = n_y = 100$, 500, 1,000. *Alg 1* is our resampling algorithm with $B_{\text{pred}} = 10^3$ resamples in each partition, *Asym* is our asymptotic approximation, *Delta* is the delta method, *SAMC* is the SAMC algorithm, and p_β is the two-sided p-value from the beta prime distribution. The diagonal dashed line has slope of 1 and intercept of 0, and indicates agreement between methods. The horizontal line in 5b shows the number of iterations used in the SAMC algorithm (set in advance, and independent of p-value). The SAMC algorithm did not produce values for 246 tests (points missing). This figure appears in color in the electronic version of this article.



(a) p-values

(b) Number of resamples in Alg 1

Figure 6: Simulation results using the statistic $T = \max(\bar{x}/\bar{y}, \bar{y}/\bar{x})$ with unequal sample sizes, where $n_y = 500$ and $n_x = 50, 200, 350$. *Alg 1* is our resampling algorithm with $B_{\text{pred}} = 10^3$ resamples in each partition, *Asym* is our asymptotic approximation, *Delta* is the delta method, *SAMC* is the SAMC algorithm, and p_β is the two-sided p-value from the beta prime distribution. The diagonal dashed line has slope of 1 and intercept of 0, and indicates agreement between methods. The horizontal line in 6b shows the number of iterations used in the SAMC algorithm (set in advance, and independent of p-value). The SAMC algorithm did not produce values for 33 tests (points missing). This figure appears in color in the electronic version of this article.

Table 1: Fifteen genes with the smallest p-values, and other output from our algorithm with $B_{\text{pred}} = 10^3$ resamples in each partition. *Single run* is the value of $\log_{10}(\tilde{p}_{\text{pred}})$ from the initial run of our resampling algorithm. The quantiles are from 1,000 replicates. For the single run, m_{stop} is the partition at which our algorithm stopped, and deviance and AIC are from the Poisson regression fit during the algorithm. Genes shown in bold were identified by Zhan et al. (2015) as being among the most effective genes for distinguishing between LUAD and LUSC using the area under the curve from a Wilcoxon rank-sum test.

	$\log_{10}(\tilde{p}_{\text{pred}})$					
Gene name	Single run	Quantiles $(10^{th}, 50^{th}, 90^{th})$	$\frac{\text{mean(LUAD)}}{\text{mean(LUSC)}}$	m_{stop}	Deviance	AIC
DSG3	-212	(-217, -208, -200)	0.0100	5	40.1	68.1
KRT5	-210	(-223, -214, -205)	0.0107	4	12.5	38.2
DSC3	-197	(-212, -205, -197)	0.0175	6	41.5	72.1
CALML3	-195	(-198, -188, -179)	0.0138	6	57.8	90
TP63	-193	(-199, -192, -186)	0.0308	6	24.2	55.1
ATP1B3	-193	(-196, -188, -181)	0.225	5	28.6	57.7
S1PR5	-190	(-190, -181, -173)	0.0775	6	98.4	131
KRT6B	-185	(-189, -181, -173)	0.0173	5	45.4	76.1
TRIM29	-183	(-188, -181, -174)	0.0788	6	39.3	72
JAG1	-180	(-186, -179, -172)	0.170	5	60.7	92.2
PVRL1	-180	(-183, -177, -171)	0.110	6	8.33	39.2
CLCA2	-178	(-188, -180, -172)	0.0138	7	51.6	86.8
BNC1	-178	(-197, -188, -181)	0.0244	7	76.8	112
FAT2	-177	(-186, -179, -173)	0.0339	7	53.5	89
KRT6C	-177	(-188, -181, -174)	0.0183	6	84.8	119