Towards a post-clustering test for differential expression

Jesse M. Zhang, Govinda M. Kamath, and David N. Tse *

Dept of Electrical Engineering, Stanford University, Stanford, CA, USA

5 November 2018

Abstract

Single-cell technologies have seen widespread adoption in recent years. The datasets generated by these technologies provide information on up to millions or more individual cells; however, the identities of the cells are often only determined computationally. Single-cell computational pipelines involve two critical steps: organizing the cells in a biologically meaningful way (clustering) and identifying the markers driving this organization (differential expression analysis). Because clustering algorithms *force* separation, performing differential expression analysis after clustering on the same dataset will generate artificially low p-values, potentially resulting in false discoveries. In this work, we introduce the truncated normal (TN) test, a test based on the truncated normal distribution that significantly corrects for this problem. We present a data-splitting-based framework that leverages the TN test to return reasonable p-values for arbitrary clustering schemes. We demonstrate the efficacy of our solution on simulated and real datasets, and we provide our code at `https://github.com/jessemzhang/tn_test`.

1 Introduction

In recent years, single-cell technologies have accelerated basic biology research, shedding light on a wide variety of biological phenomena. Modern advances in single-cell technologies can cheaply generate genomic profiles of up to millions of individual cells [1, 2, 3, 4, 5]. Depending on the type of assay, these profiles can describe cell features such as RNA expression, transcript compatability counts [6], epigenetic features [7], or nuclear RNA expression [8]. Because the cell types of individual cells often cannot be known prior to the computational step, a key step in single-cell computational pipelines [9, 10, 11, 12, 13] is clustering: organizing individual cells into biologically meaningful populations. Furthermore, computational pipelines use differential expression analysis to identify the key features that distinguish a population from other populations, for example a gene based on its relative expression level.

Existing computational workflows often perform clustering and differential expression analysis on the same dataset. This reusing of the same dataset is more colloquially known as "data snooping." Because clustering algorithms *force* separation regardless of the underlying truth, running differential expression on the resulting clusters will yield artificially low p-values. While several differential expression methods exist [14, 12, 15, 16, 17, 18], as a motivating example we consider the classic Student's t-test introduced in 1908 [19]. We note that none of these tests correct for

*Corresponding Author. Email: dntse@stanford.edu

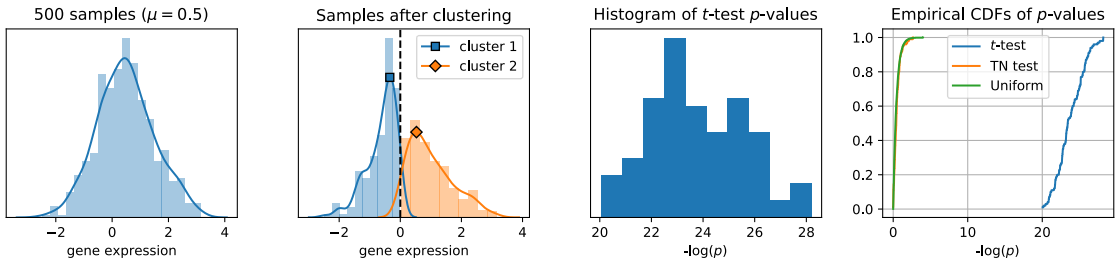

Figure 1: Artificially low p-values due to clustering. Although the 500 samples are drawn from the same $\mathcal{N}(\mu, 1)$ distribution, our simple clustering approach will always generate two clusters that seem significantly different under the t-test. In this work, we explore an approach for correcting the selection bias due to clustering. In other words, we attempt to close the gap between the blue and green curves in the rightmost plot. We introduce the TN test, which generates significantly more reasonable p-values.

the data snooping problem as they were not designed to account for the clustering process. The t-test was devised for controlled experiments where the hypothesis to be tested was defined before the experiments were carried out. For example, to test the efficacy of a drug, the researcher would randomly assign individuals to case and control groups, administer the placebo or the drug, and take a set of measurements. Because the populations were clearly defined a priori, so was the null hypothesis. Therefore, a t-test would yield valid p-values. In other words, under the null hypothesis where no effect exists, the mean measurement should be the same across the two populations, and the p-value should be uniformly distributed between 0 and 1.

For single-cell analysis, however, the populations are often obtained *after* the measurements are taken, via clustering, and therefore we can expect the t-test to return significant p-values even if the null hypothesis was true. Figure 1 shows how a measurement, such as expression of a gene, is deemed significantly different between two clusters even though all samples came from the same normal distribution. The clustering introduces a **selection bias** that would result in several false discoveries if uncorrected.

In this work, we introduce the truncated normal (TN) test, an approximate test based on the truncated normal distribution that corrects for a significant portion of the selection bias generated by clustering. We condition on the clustering event using the hyperplane that separates the clusters. By incorporating this hyperplane into our null model, we can obtain a uniformly distributed p-value even in the presence of clustering. To our knowledge, the TN test is the first test to correct for clustering bias while addressing the differential expression question: *is this feature significantly different between the two clusters?* For the rest of this work, we assume that the feature of interest is some gene expression level.

We then proceed to provide a data-splitting based framework that allows us to generate valid p-values for differential expression of genes for clusters obtained from any clustering algorithm.

The TN test was motivated by existing theory in selective inference [20, 21]. Additionally, the TN test can be used to determine whether or not a clustering is significant, an area where existing tests [22, 23] have also demonstrated promising results.

2 Methods

2.1 Clustering model

To motivate our approach, we consider the simplest model of clustering: samples are drawn from one of two clusters, and the clusters can be separated using a linear separator. As we later show, the linear separability assumption is often true for high-dimensional datasets such as single-cell datasets. For the rest of this section, we assume that the hyperplane a is given and independent from the data we are using for differential expression analysis. For example, we can assume that in a dataset of n independent and identically distributed samples and d genes, we had set aside n_1 samples to generate the two clusters and identify a, thus allowing us to classify future samples without having to rerun our clustering algorithm. We run differential expression analysis using the remaining $n_2 = n - n_1$ samples while *conditioning* on the selection event. More specifically, our test accounts for the fact that a particular a was chosen to govern clustering. We will later demonstrate empirically that the resulting test we develop suffers from significantly less selection bias.

For pedagogical simplicity, we start by assuming that our samples are 1-dimensional ($d = 1$) and our clustering algorithm divides our samples into two clusters based on the sign of the (mean-centered) observed expressions. Let Y represent the negative samples and Z represent the positive samples. We assume that our samples come from normal distributions with known variance 1 prior to clustering, and we condition on our clustering event by introducing truncations into our model. Therefore Y and Z have *truncated* normal distributions due to clustering:

$$f_Y(y; \mu_L) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y - \mu_L)^2\right) \frac{\mathbb{I}(y \leq 0)}{\Phi(-\mu_L)}$$
$$f_Z(z; \mu_R) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(z - \mu_R)^2\right) \frac{\mathbb{I}(z > 0)}{\Phi(\mu_R)}.$$

Here, the $\mathbb{I}$ terms are indicator functions denoting how truncation is performed, and the Φ terms are normalization factors to ensure that f_Y and f_Z integrate to 1. Φ represents the CDF of a standard normal random variable. μ_L and μ_R denote the means of the untruncated versions of the distributions. We want to test if the gene is differentially expressed between two populations Y and Z, i.e. if $\mu_L = \mu_R$.

2.2 Derivation of the test statistic

The joint distribution of our n samples of Y with our m samples of Z can be expressed in exponential family form as

$$f_{Y,Z}(y, z; \mu_L, \mu_R) = \exp\left(\mu_L n\bar{y} + \mu_R m\bar{z} - \psi(\mu_L, \mu_R)\right) h(y, z),$$

where $\bar{y}$ and $\bar{z}$ represent the sample means of Y and Z, respectively. ψ is the cumulant generating function, and h is the carrying density. Please see the Appendix for more details. To test for differential expression, we want to test if $\mu_L = \mu_R$, which is equivalent to testing if $\mu_R - \mu_L = 0$. With a slight reparametrization, we let $\theta = (\mu_R - \mu_L)/2$ and $\mu = (\mu_R + \mu_L)/2$, resulting in the expression:

$$f_{Y,Z}(y, z; \mu, \theta) = \exp\left(\mu(n\bar{y} + m\bar{z}) + \theta(m\bar{z} - n\bar{y}) - \psi'(\mu, \theta)\right) h(y, z).$$

Algorithm 1 1D TN test when variance = 1 and clustering is performed based on sign of expression

Input: Two groups of samples Y, Z

Output: p-value

1. Using maximum likelihood, estimate μ_L, μ_R, the mean parameters of the truncated Gaussian distributions

2. To obtain the null distribution, set $\mu_L = \mu_R = (\mu_L + \mu_R)/2$, then obtain estimates of $\mu_Y, \mu_Z, \sigma_Y^2, \sigma_Z^2$, the means and the variances of truncated distributions

3. Perform an approximate test with the statistic

$$\frac{m(\bar{z} - \mu_Z) - n(\bar{y} - \mu_Y)}{\sqrt{m\sigma_Z^2 + n\sigma_Y^2}}$$

   where m and n represent the number of samples of Z and Y, respectively. This statistic is approximately $\mathcal{N}(0, 1)$ distributed.

We can test design tests for $\theta = \theta_0$ using its sufficient statistic, $m\bar{z} - n\bar{y}$ [24]. From the Central Limit Theorem (CLT), we see that the test statistic

$$\frac{m(\bar{z} - \mu_Z) - n(\bar{y} - \mu_Y)}{\sqrt{m\sigma_Z^2 + n\sigma_Y^2}} \xrightarrow{\text{CLT}} \mathcal{N}(0, 1).$$

Intuitively, this test statistic compares $m\bar{z} - n\bar{y}$, the gap between the observed means, to $m\mu_Z - n\mu_Y$, the gap between the excepted means. For differential expression, we set $\theta_0 = 0$. Because $\mu_Y, \mu_Z, \sigma_Y^2, \sigma_Z^2$ under the null $\mu_L = \mu_R$ are unknown, we estimate them from the data by first estimating μ_L and μ_R via maximum likelihood. Although the estimators for μ_L and μ_R have no closed-form solutions due to the Φ terms, the joint distribution can be represented in exponential family form. Therefore the likelihood function is concave with respect to μ_L and μ_L, and we can obtain estimates $\hat{\mu}_L, \hat{\mu}_R$ via gradient ascent. We then set $\hat{\mu} = (\hat{\mu}_L + \hat{\mu}_R)/2$. This procedure is summarized in Algorithm 1. We note that approximation errors are accumulated from the CLT approximation and errors in the maximum likelihood estimation process, and therefore the limiting distribution of the test statistic should have wider tails. Despite this, we show later that this procedure corrects for a large amount of the selection bias.

2.3 TN test for d dimensions and unknown variance

In this section, we generalize our 1-dimensional result to d dimensions and non-unit variance. Our samples now come from the multivariate truncated normal distributions

$$f_Y(y; \mu_L, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(y - \mu_L)^T \Sigma^{-1} (y - \mu_L)\right) \frac{\mathbb{I}(a^T y \leq 0)}{\Phi\left(-\frac{a^T \mu_L}{\sqrt{a^T \Sigma a}}\right)}$$

$$f_Z(z; \mu_R, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(z - \mu_R)^T \Sigma^{-1} (z - \mu_R)\right) \frac{\mathbb{I}(a^T z > 0)}{\Phi\left(\frac{a^T \mu_R}{\sqrt{a^T \Sigma a}}\right)}$$

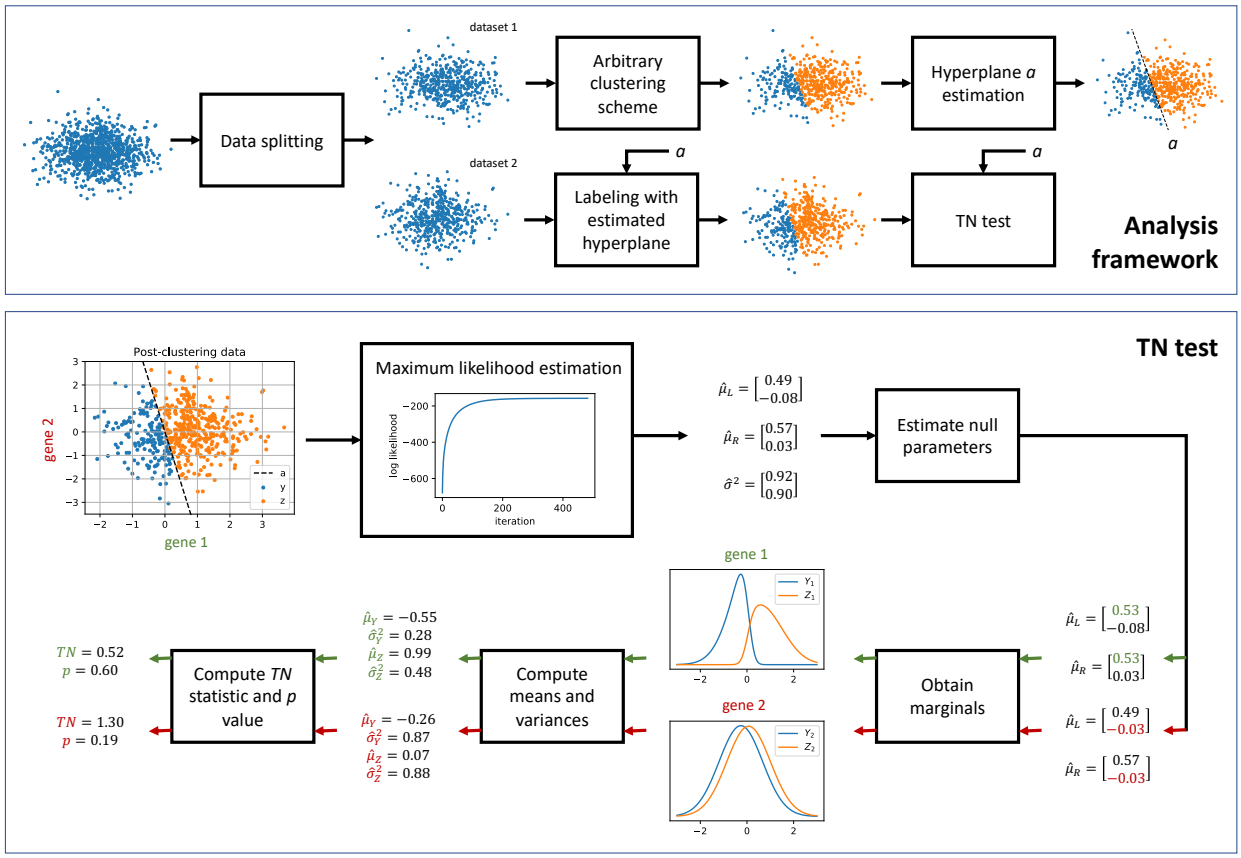

Figure 2: Overview of the TN test with 1000 samples drawn independently from a unimodal Gaussian distribution. Genes 1 and 2 are drawn independently from $\mathcal{N}(0.5, 1)$ and $\mathcal{N}(0, 1)$, respectively. **Analysis framework**. The samples are split into two datasets. An arbitrary clustering algorithm is performed on dataset 1, and a hyperplane is fitted to the cluster labels. The hyperplane, which is independent from dataset 2, is then used to assign labels to dataset 2. Differential expression analysis is performed on dataset 2 using the TN test, which requires knowledge of the hyperplane to correct for selection bias introduced by the clustering event. **TN test.** μ_L and μ_R are the means of the untruncated normal distributions that generated Y and Z. The covariance matrix is assumed to be diagonal and equal across the two untruncated distributions, and σ^2 represents the diagonal entries along the matrix. Separating hyperplane a is assumed to be given, but can be estimated from a held-out portion of data using a support vector machine. Under the t-test, the p-values obtained for dataset 2's genes 1 and 2 are 5.3×10^{-88} and 1.0×10^{-6}, respectively.

where μ_L, μ_R, Σ denote the means and covariance matrix of the untruncated versions of the distributions. We assume that all samples are drawn independently, and Σ is diagonal: $\Sigma_{ij} = \sigma_i^2$ if $i = j$ else $\Sigma_{ij} = 0$. The joint distribution of our n samples of Y with our m samples of Z can be expressed in exponential family form as

$$f_{Y,Z}(y, z; \eta) = \exp\left(-\frac{1}{2}\eta_1^T \left(\|y\|_F^2 + \|z\|_F^2\right) + \eta_2^T n\bar{y} + \eta_3^T m\bar{z} - \psi(\eta)\right) h(y, z)$$

Algorithm 2 TN test

Input: Two groups of samples Y, Z, a separating hyperplane a
Output: p-value

1. Using maximum likelihood, estimate the mean and variance parameters of the truncated Gaussian distributions on either side of the hyperplane: μ_L, μ_R, Σ
2. For gene g, obtain the marginal distributions $f_{Y_g}(y_g), f_{Z_g}(z_g)$ under the null (i.e. setting $\mu_{Lg} = \mu_{Rg} = (\mu_{Lg} + \mu_{Rg})/2$)
3. Using numerical integration, obtain estimates of $\mu_{Y_g}, \mu_{Z_g}, \sigma^2_{Y_g}, \sigma^2_{Z_g}$, the means and the variances of $f_{Y_g}(y_g)$ and $f_{Z_g}(z_g)$
4. Perform an approximate test with the statistic

$$\frac{m(\bar{z}_g - \mu_{Z_g}) - n(\bar{y}_g - \mu_{Y_g})}{\sqrt{m\sigma^2_{Z_g} + n\sigma^2_{Y_g}}}$$

where m and n represent the number of samples of Z and Y, respectively.

---

where ψ is the cumulant generating function, h is some carrying density, and $\|\cdot\|_F$ denotes the Frobenius norm. The natural parameters η_1, η_2, η_3 are equal to

$$\eta_1 = \begin{bmatrix} 1/\sigma_1^2 \\ \vdots \\ 1/\sigma_k^2 \end{bmatrix}, \quad \eta_2 = \Sigma^{-1}\mu_L, \quad \eta_3 = \Sigma^{-1}\mu_R.$$

To test differential expression of gene g, we can test if $\eta_{2g} = \eta_{3g}$, which is equivalent to testing $\mu_{Lg}/\sigma_g^2 = \mu_{Rg}/\sigma_g^2$ or $\mu_{Lg} = \mu_{Rg}$. In similar spirit to the 1-dimensional case, we perform a slight reparameterization, letting $\theta_g = (\eta_{3g} - \eta_{2g})/2$ and $\mu_g = (\eta_{2g} + \eta_{3g})/2$:

$$f_{Y,Z}(y,z) = \exp\left(-\frac{\eta_1^T}{2}(\|y\|_F^2 + \|z\|_F^2) + \sum_{j\neq g}(\eta_{2j} n\bar{y}_j + \eta_{3j} m\bar{z}_j) + \mu_g(n\bar{y}_g + m\bar{z}_g) + \theta_g(m\bar{z}_g - n\bar{y}_g) - \psi'\right) h.$$

We again design tests for θ_g using its sufficient statistic, $m\bar{z}_g - n\bar{y}_g$:

$$TN = \frac{m(\bar{z}_g - \mu_{Z_g}) - n(\bar{y}_g - \mu_{Y_g})}{\sqrt{m\sigma^2_{Z_g} + n\sigma^2_{Y_g}}} \xrightarrow{\text{CLT}} \mathcal{N}(0,1).$$

During the testing procedure, we want to evaluate if $\theta_g = 0$ (i.e. if gene g has significantly different mean expression between the two populations). With $\theta_g = 0$ as our null hypothesis, we compute the corresponding parameters $\mu_{Z_g}, \mu_{Y_g}, \sigma^2_{Z_g}, \sigma^2_{Y_g}$ under the null, allowing us to evaluate the probability of seeing a TN statistic at least as extreme as the one observed for the actual data.

Like in the 1-dimensional case, we use maximum likelihood to estimate η_1, η_2, and η_3, leveraging the fact that the likelihood function is concave because the joint distribution is an exponential

family. After estimating the natural parameters, we can easily recover Σ, μ_L, and μ_R. To obtain estimates for $\mu_{Z_g}, \mu_{Y_g}, \sigma^2_{Z_g}, \sigma^2_{Y_g}$ under the null, we first set $\mu_{Lg} = \mu_{Rg} = (\mu_{Lg} + \mu_{Rg})/2$. We then use numerical integration to obtain the first and second moments of gene g's marginal distributions:

$$f_{Y_g}(y_g) = \Phi\left(\frac{-a_g y_g - a_{-g}^T \mu_{L,-g}}{\sqrt{\sum_{i \neq g} a_i^2 \sigma_i^2}}\right) \frac{\exp\left(-\frac{1}{2\sigma_g^2}(y_g - \mu_{Lg})^2\right)}{\Phi\left(-\frac{a^T \mu_L}{\sqrt{\sum_{i=1}^d a_i^2 \sigma_i^2}}\right)\sqrt{2\pi\sigma_g^2}}$$

$$f_{Z_g}(z_g) = \Phi\left(\frac{a_g z_g + a_{-g}^T \mu_{R,-g}}{\sqrt{\sum_{i \neq g} a_i^2 \sigma_i^2}}\right) \frac{\exp\left(-\frac{1}{2\sigma_g^2}(z_g - \mu_{Rg})^2\right)}{\Phi\left(\frac{a^T \mu_R}{\sqrt{\sum_{i=1}^d a_i^2 \sigma_i^2}}\right)\sqrt{2\pi\sigma_g^2}}.$$

The TN test procedure is summarized in Algorithm 2 and Figure 2. More details regarding the above derivations are given in the Appendix.

2.4 TN test for p-value correcting with arbitrary clustering

Algorithm 3 Clustering and TN test framework

Input: Samples X

Output: p-value

1. Split X into two partitions X_1, X_2
2. Run your favorite clustering algorithm on X_1 to generate labels, choosing two clusters for downstream differential expression analysis
3. Use X_1 and the labels to determine a, the separating hyperplane (e.g. using an SVM)
4. Divide X_2 into Y, Z using the obtained hyperplane
5. Run TN test using Y, Z, a

We describe a full framework (Figure 2) for clustering the dataset X and obtaining corrected p-values via the TN test. Using a data-splitting approach, we run an arbitrary clustering algorithm on one portion of the data, X_1, to generate 2 clusters. For differential expression analysis, we estimate the separating hyperplane a using a linear binary classifier such as the support vector machine (SVM). This hyperplane is used to assign labels to the remaining samples in X_2, yielding Y and Z. Finally, we can run a TN test using Y, Z, and a. This approach is summarized in Algorithm 3. Note that in the case of $k > 2$ clusters, we can assign all points in X_2 using our collection of $\binom{k}{2}$ hyperplanes.

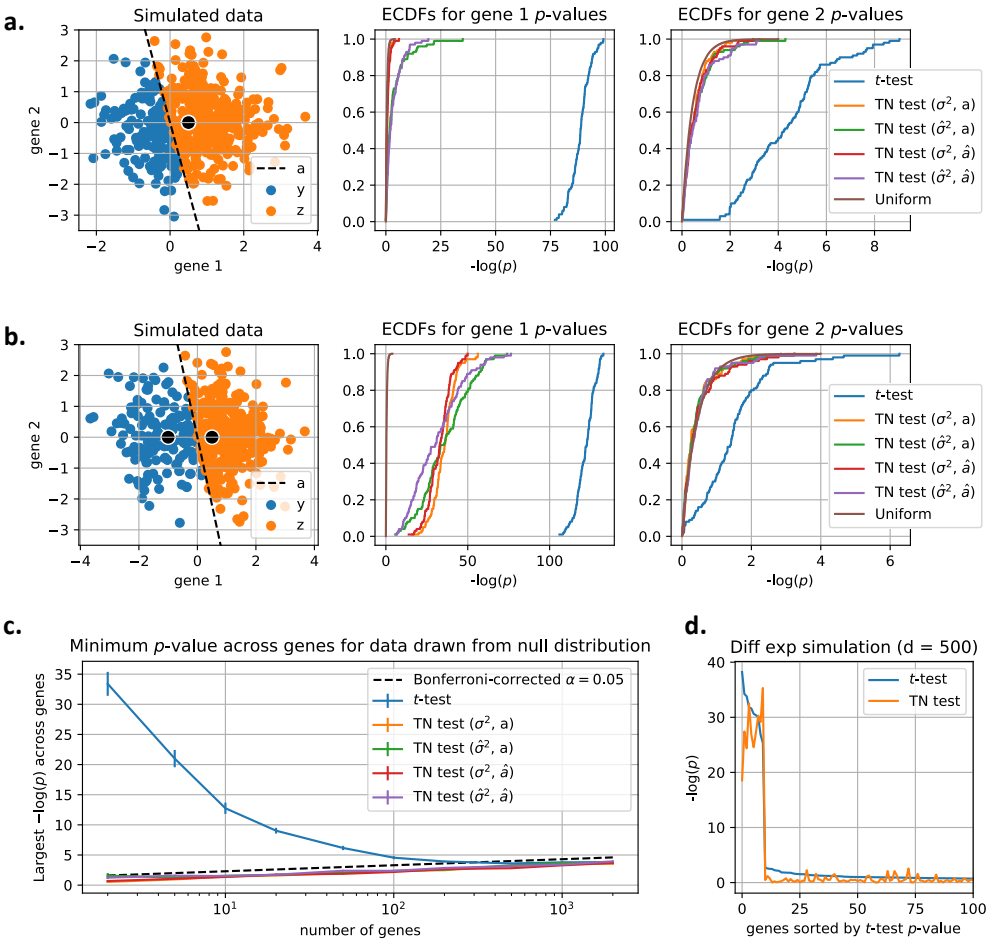

Figure 3: Results on simulated data drawn from truncated normal distributions. **a.** 500 samples are drawn from the same distribution, and genes 1 and 2 are drawn from $\mathcal{N}(0.5, 1)$ and $\mathcal{N}(0, 1)$, respectively. The clustering step splits the dataset into groups of 156 and 344 samples, and a exactly captures the clustering rule. We see that although neither gene is differentially expressed in the underlying distribution, the t-test consistently returns small p-values across 100 simulation runs. We present four version of the TN test, all of which significantly correct for the clustering step. $\hat{\sigma}^2$ indicates that the variance was unknown and therefore estimated from the data. $\hat{a}$ indicates that the hyperplane was estimated from a held-out 10% of the samples using an SVM. **b.** The experiment from **a** is repeated except gene 1 is drawn from a $\mathcal{N}(-1, 0)$ distribution instead. The number of samples in each group and the separating hyperplane remain the same. **c.** We explore how the minimum p-value across genes change with d, the number of genes. For a particular number of genes, 200 samples are drawn from a $\mathcal{N}(0, I)$ distribution, and a is chosen randomly. This simulation is repeated 10 times for each value of d. α indicates the chosen level of significance. **d.** For $d = 500$, we run a 200-sample simulation experiment (100 in each cluster) where 10 genes are differentially expressed. 10 values of μ_L were set to -1, and the corresponding entries in μ_R were set to 1. All other entires of μ_L, μ_R were set to 0, and $\sigma^2 = 1$.

3 Results

3.1 Performance on simulated truncated normal data

We first explore the performance of our proposed method on synthetic data sampled from normal distributions prior to clustering, resulting in data sampled from truncated normal distributions post-clustering. Figure 3a shows results for the 2-dimensional case where no differential expression should be observed (the untruncated means of both clusters are identical). The covariance matrix is identity and we let σ^2 denote the diagonal entries of the matrix. Note that for this example, gene 1 needs a larger correction factor than gene 2 because the separating hyperplane is less aligned with the gene 1 axis. We see that when both the variance σ^2 and separating hyperplane a are known, the TN test completely corrects for the selection event. As we introduce more uncertainty (i.e. if we need to estimate σ^2 or a or both), the correction factor shrinks; however, the gap is still significantly better than for the t-test case. To estimate a, we fit a linear regression model to 10% of the dataset, and we work with the remaining 90% of the dataset after relabeling it based on our estimate of a. Figure 3b repeats the experiment for the case where gene 1 is differentially expressed. The TN test again corrects for the selection bias in gene 2, but we still obtain significant p-values for gene 1 though not nearly as extreme as for the t-test case. We also observe a general loss of statistical power if we need to estimate σ^2 or a. Loss of statistical power is visualized in Figure 3b as a shift of the empirical CDF (ECDF) left towards larger p-values. For example, if we need to estimate a, our dataset used for differential expression grows smaller, and therefore the test loses some power.

Figure 3c extends the above result to greater values of d, the total number of genes. For each value of d, samples are drawn from the same $\mathcal{N}(0, I)$ distribution, and the separating hyperplane is randomly chosen. We see that as we increase d, the minimum TN test p-value across all d genes exactly follows the family-wise error rate (FWER) curve. FWER represents the probability of making at least 1 false discovery, and naturally increases with d, the number of tests we do. This highlights the validity of the TN test, which is behaving as expected when samples are drawn from the null distribution. In comparison, the t-test returns extreme p-values especially for lower values of d. As d increases, however, the marginal distributions become roughly normal, and the selection bias incurred by our simple clustering approach disappears. While the TN test provides less gain in higher (≥ 200) dimensions, we note that for real datasets, cluster identities are often driven by an effectively small amount of genes. Cell identities are not driven by a large amount of *independent* features, and this is why several single-cell pipelines perform dimensionality reduction before clustering.

We also simulate date from underlying distributions where differential expression should be observed. Figure 3d shows that the TN test assigns low p-values to differentially expressed genes and high p-values to the other genes.

3.2 Performance on single-cell RNA-Seq data

We consider the peripheral blood mononuclear cell (PBMC) dataset of 2700 cells generated using recent techniques developed by 10x Genomics [4], and this dataset was also used in a tutorial for the Seurat single-cell package [9]. Figure 4a shows the result of processing the dataset using Seurat, which preprocesses the dataset before running a graph-based clustering algorithm [25, 26, 27] yielding 9 clusters.

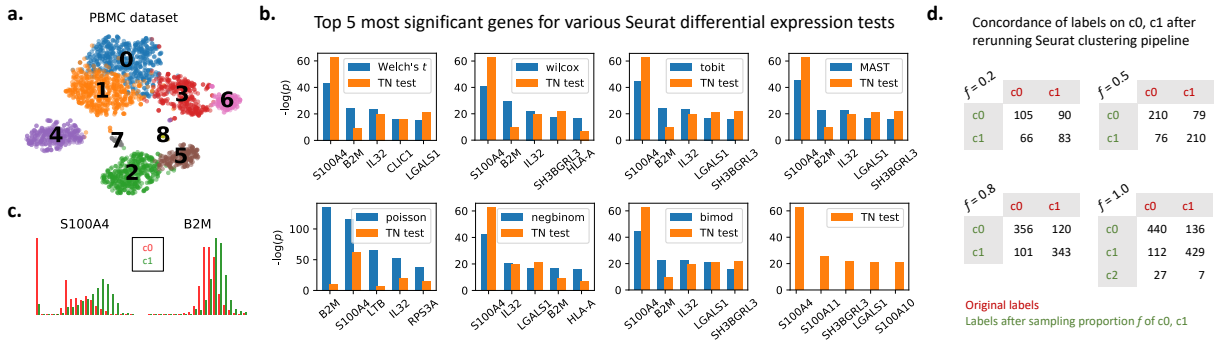

Figure 4: TN test on subset of 2700-cell PBMC dataset. **a.** t-SNE plot of the 2700 PBMC dataset colored by clusters found using Seurat [9]. **b.** The analysis pipeline in Figure 2 is run on the cells in clusters 0 and 1 to generate TN test p-values. The Seurat clustering pipeline is used to recover 2 clusters on dataset 1, half of the 1151 cells, and seven differential expression methods provided by Seurat (see Appendix for details) are also run on dataset 1. **c.** The expression profile of two genes across all 1151 cells with the new labels are shown: *S100A4*, the most significant gene according to several tests, and *B2M*, a gene corrected by the TN test. **d.** State-of-the-art single-cell clustering pipelines such as Seurat can generate different clustering results on the same cells.

We restrict our attention to clusters 0 and 1 and run the analysis framework described by Figure 2. With 579 and 572 cells in clusters 0 and 1, we randomly split the pool of 1151 cells in half into datasets 1 and 2. We recluster dataset 1 using Seurat, using clustering parameters that would result in 2 clusters. We use SVM to obtain a hyperplane that perfectly separates the two clusters, and we use this hyperplane to assign labels to samples in dataset 2. We subsequently perform a TN test using the hyperplane and dataset 2. To compare the TN test to methods that do not account for clustering bias, we run the entire Seurat pipeline (including differential expression analysis) on dataset 1. Seurat offers several differential expression approaches, and for each of 7 of the approaches (see Appendix for more details), we compare the obtained p-values to the TN test p-values. Figure 4b shows while the TN test agrees with the 7 differential expression tests on several genes (e.g. *S100A4*), it also disagrees on some other genes. The gene with the most heavily corrected p-value was *B2M*. While several of the Seurat-provided tests would detect a significant change ($p < 10^{-20}$), the TN-test accounts for the fact that this difference in expression may be due to a clustering artifact ($p = 2.2 \times 10^{-10}$). Figure 4c shows the expression profiles of *S100A4* and *B2M* for all 1151 cells with new labels, highlighting how the detected difference for *S100A4* may actually be significant while the detected difference for *B2M* may have been induced by clustering.

We also note that state-of-the-art single-cell clustering pipelines such as Seurat can generate different clustering results on the same dataset (Figure 4d). Importantly, different clustering results imply different null hypotheses when we reach the differential expression analysis step, which further undermines the validity of the "discovered" differentiating markers. Although data splitting reduces the number of samples available for clustering, we see that sacrificing a portion of the data can correct for biases introduced by clustering. Additionally, we note that the method we propose here allows arbitrary clustering schemes to be used on the first portion of the dataset but restricts to only linear separators on the second portion of the dataset. In high dimensions, we can obtain linear separators for most clusters found in practice.

4 Acknowledgements

We thank Jonathan Taylor, Martin Zhang, and Vasilis Ntranos of Stanford University for helpful discussions about selective inference and applications of the method. GMK and JMZ are supported by the Center for Science of Information, an NSF Science and Technology Center, under grant agreement CCF-0939370. JMZ and DNT are supported in part by the National Human Genome Research Institute of the National Institutes of Health under award number R01HG008164.

References

[1] Klein, A. M. *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161**, 1187–1201 (2015).

[2] Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–1214 (2015).

[3] Fan, H. C., Fu, G. K. & Fodor, S. P. Combinatorial labeling of single cells for gene expression cytometry. *Science* **347**, 1258367 (2015).

[4] Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8**, 14049 (2017).

[5] Cao, J. *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* **357**, 661–667 (2017).

[6] Ntranos, V., Kamath, G. M., Zhang, J. M., Pachter, L. & David, N. T. Fast and accurate single-cell rna-seq analysis by clustering of transcript-compatibility counts. *Genome biology* **17**, 112 (2016).

[7] Buenrostro, J. D. *et al.* Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* **523**, 486 (2015).

[8] Habib, N. *et al.* Massively parallel single-nucleus rna-seq with dronc-seq. *Nature methods* **14**, 955 (2017).

[9] Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36**, 411 (2018).

[10] Qiu, X. *et al.* Single-cell mrna quantification and differential analysis with census. *Nature methods* **14**, 309 (2017).

[11] McCarthy, D. J., Campbell, K. R., Lun, A. T. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics* **33**, 1179–1186 (2017).

[12] Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32**, 381 (2014).

[13] Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 15 (2018).

[14] McDavid, A. *et al.* Data exploration, quality control and testing in single-cell qpcr-based gene expression experiments. *Bioinformatics* **29**, 461–467 (2012).

[15] Finak, G. *et al.* Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome biology* **16**, 278 (2015).

[16] Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology* **15**, 550 (2014).

[17] Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature methods* **11**, 740 (2014).

[18] Zhang, J. M., Fan, J., Fan, H. C., Rosenfeld, D. & David, N. T. An interpretable framework for clustering single-cell rna-seq datasets. *BMC bioinformatics* **19**, 93 (2018).

[19] Student. The probable error of a mean. *Biometrika* 1–25 (1908).

[20] Taylor, J. & Tibshirani, R. J. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences* **112**, 7629–7634 (2015).

[21] Fithian, W., Sun, D. & Taylor, J. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597* (2014).

[22] Tibshirani, R., Walther, G. & Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**, 411–423 (2001).

[23] Liu, Y., Hayes, D. N., Nobel, A. & Marron, J. Statistical significance of clustering for high-dimension, low–sample size data. *Journal of the American Statistical Association* **103**, 1281–1293 (2008).

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

[25] Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31**, 1974–1980 (2015).

[26] Levine, J. H. *et al.* Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell* **162**, 184–197 (2015).

[27] Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* **2008**, P10008 (2008).