A Framework for Single Cell RNA Sequencing Data Analysis

Yiliang Zhang[1,*], Molei Liu[1], Yue Li[1], Xiannian Zhang[3], Yanyi Huang[3,4], Hao Ge[2,3]

[1]School of Mathematical Science, Peking University, Beijing, China; [2]Beijing International Center For Mathematical Research, Peking University, Beijing, China;[3] Biodynamic Optical Imaging Center, Peking University, Beijing, China; [4] College of Engineering, Peking University, Beijing, China

Abstract

High-throughput technologies are widely used, for example to assay genetic variants, gene and protein expression, and epigenetic modifications. Systematic errors, including batch effects, have been widely reported as a major challenge in high-throughput technologies. Even though some efforts have been made to remove such bias, seldom of the previous methods take the overdispersion of zero(which indicates dropout events) appeared in the single-cell RNA sequencing count data into consideration. However, the zero counts is supposed to have great relation with batch effects and gene expression level. In this article, we propose a framework for single-cell RNA sequencing data analysis based on zero-inflated poisson model which can account and adjust for batch effect with the information for gene expression level and dropout events.

Introduction

High-throughput technologies are widely used, for example to assay genetic variants, gene and protein expression, and epigenetic modifications[15]. Systematic errors, including batch effects, have been widely reported as a major challenge in high-throughput technologies. Batch effects occur because measurements are affected by laboratory conditions, reagent lots and personnel differences. This becomes a major problem when batch effects are correlated with an outcome of interest and lead to incorrect conclusions.

[13], which is among the earliest microarray experiments, firstly observed batch effects. They describe batch effects by "It's well known among aficionados that comparison of the 'same' experiment performed a few weeks apart reveals considerably wider variation than seen when a single sample is tested by repeated hybridization." After the concept of batch effect is introduced into the field, [1] proposed the first method to adjust for batch effects based on

singular value decomposition(SVD). The SVD method is applied in [18] to enlarge the total dataset from two different array sets on gene-expression patterns of soft tissue tumours. The first paper specific only on batch effect came out in 2004, before which the method to adjust batch effects are only SVD. This paper use DWD to adjust batch effects[3].Johnson and Li propose a method based on empirical bayes[11] and a method to detect whether there are batch effects in a data set was proposed in [19]. The methods to adjust for batch effects and technical bias are keeping been proposed in these year such as Surrogate Variable Analysis(SVA)[16], cross-platform normalization(XPN)[20] and covariance adjustment[14].

Even though these efforts have been made to remove such bias, seldom of the previous methods take the overdispersion of zero(which indicates dropout events) appeared in the single-cell RNA sequencing count data into consideration. In [8], the author argues that proportion of detected gene is a major source of technical cell-to-cell noise. The author illustrate their points with examining data from five published studies. [9] observed the effects brought by dropout event and proposed Toolkit for Analysis of Single Cell(TASC) model based on zero-inflated model[12] by use of external RNA spike-in.They develop an empirical bayes procedure that borrows information across cells.

In this article, we propose a framework for single-cell RNA sequencing data analysis also based on zero-inflated poisson model which can account and adjust for batch effect with the information for gene expression level and dropout events. Compared with [9], our model didn't use external spike-in whose quality is often considered to be affected by systematic bias. In addition, our method is more efficient in computation.

In the following sections, we firstly introduce how to modeling the zero-inflated model and how to estimate the parameters with Monte-Carlo EM algorithm. Then, we introduce our procedure on quality control, differential expression analysis and visualization. At last, we apply our method on real experiment data.

Methods

Modeling of observed count

We use Zero-inflated Poisson(ZIP) regression to model the scRNA-seq data. There is a indicator Z_{cg} for gene g in cell c which suggests dropout event occurs if $Z_{cg} = 0$ and dropout does not occur if $Z_{cg} = 1$. We assume the read count of gene g in cell c, Y_{cg} subjects to a poisson distribution with parameter λ_{cg} when dropout event does not occur, i.e. $Z_{cg} = 1$ while Y_{cg} is zero when dropout event occurs, i.e. $Z_{cg} = 0$. So, our model for observed read count data can be write as:

$$Y_{cg} \sim \begin{cases} Poisson(\lambda_{cg}) & (Z_{cg} = 1); \\ 0 & (Z_{cg} = 0). \end{cases}$$

So that

$$P(Y_{cg} = 0) = P(Z_{cg} = 0) + P(Z_{cg} = 1) \exp(-\lambda_{cg})$$
$$P(Y_{cg} = k) = P(Z_{cg} = 1) \frac{\lambda_{cg}^k}{k!} \exp(-\lambda_{cg}), k = 1, 2, \ldots.$$

Modeling of non-dropout

We use poisson distribution to approximate the non-dropout read count. λ_{cg}, the parameter of the poisson distribution, depends on the gene's true expression level, denoted by $\mu_{G(c)g}$. The function, $G(c)$: cell $\longrightarrow$ biological group, defines the map from cell to biological group. In other words, $\mu_{G(c)g}$ is the essential absolute molecule count level of biological group $G(c)$ for arbitrary gene g. Thus, λ_{cg} can be written as:

$$\log(\lambda_{cg}) = \log(\mu_{G(c)g}) + \log(r_c) + \log(l_g) + b_c,$$

where the total read counts of cell c is denoted by r_c and the length of gene g is denoted by l_g. We include total read counts and length of gene into account because the longer the gene is and the more read counts the cell is detected will lead to more read count for every gene of the cell, which is confounded with true expression level $\mu_{G(c)g}$. We add total read counts and gene length into our model in order to remove the confounded effect. b_c is the batch effect of cell c. Motivated by [8], we assume that λ_{cg} is proportional to e^{b_c} in our model. b_c also has impact on dropout rates and we will introduce it in the next section.

Modeling of dropout rates

For gene g in cell c, let Z_{cg} denotes the indicator for dropout. $Z_{cg} = 1$ indicates dropout event does not occur for gene g in cell c. We assume that Z_{cg} is a random variable and subject to a Bernoulli distribution whose parameters are the following factors:(1)r_c, total read count in cell c; (2)l_g, length of gene g; (3)$B(c)$, batch of cell c; (4)b_c, batch effect for cell c, which is a cell specific covariate.

We form a GLMM to jointly model these factors. To be specific, we can write the distribution of Z_{cg} as:

$$Z_{cg} \sim Bernoulli\left(\Phi(\gamma + \alpha \log(r_c) + \beta \log(l_g) + \theta^T \mathbf{f}(b_c))\right),$$

where $\mathbf{f}(\cdot)$ is a group of polynomial basis, i.e. $b_c, b_c^2, \ldots$, and $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution. $\{\alpha, \beta, \gamma, \theta^T\} \subset \Theta$ belongs to the set of coefficients for the model. The batch effect is captured by the random effect term $b_c \sim \mathcal{N}(\nu_{B(c)}, \sigma^2_{B(c)})$. $\nu_{B(c)}$ and $\sigma^2_{B(c)}$ are the mean and variance of the cell specific batch effect b_c from batch $B(c)$ respectively. In other words, the b_cs of the cells in an identical batch are independently generated from a same normal distributed population with mean $\nu_{B(c)}$ and variance $\sigma^2_{B(c)}$. In addition, to ensure identifiability, we assume $\sum_c \nu_{B(c)} = 0$.

Estimation of the coefficients

The number of parameters that can be estimated in the model depends on how we formulate the the polynomial basis of b_c. We cannot directly estimate the parameters with maximum likelihood estimation(MLE) since it's hard to maximize the log-likelihood for the model. The log-likelihood function can be found in the supplementary material for this article. The sum of exponentials and the random effect complicates the maximization of $L(\Theta; Y)$, which is the likelihood of the model.

But suppose we knew which zeros came from dropout event and which came from the Poisson; that is, suppose we could observe $Z_{cg} = 1$ when Y_{cg} is from the Poisson state and $Z_{cg} = 0$ when Y_{cg} is from dropout event, zero state. What's more, if we also knew the random effect of the batch effect of all the cells, then the likelihood would change into a simple one(see the supplementary material of this article for details)

$$L(\Theta; Y, \mathbf{Z}, \mathbf{b}) = \mathbf{L}(\mu_{\mathbf{G(c)},\mathbf{g}}; \mathbf{Y}, \mathbf{Z}, \mathbf{b}) + \mathbf{L}(\alpha, \beta, \gamma, \theta; \mathbf{Z}, \mathbf{b}) + \mathbf{G} \cdot \mathbf{L}(\nu_{\mathbf{B(c)}}, \sigma^{\mathbf{2}}_{\mathbf{B(c)}}; \mathbf{b}) + \mathbf{r}(\mathbf{Y}, \mathbf{Z}, \mathbf{b}),$$

where $\mathbf{Z}$ is the set of all the dropout indicators Z_{cg} and $\mathbf{b}$ is the set of all the batch effect b_c. $r(Y, \mathbf{Z}, \mathbf{b})$ is the function with on relation with Θ which can be ignored in the procedure of MLE. $L(\mu_{G(c),g}; \mathbf{Y}, \mathbf{Z}, \mathbf{b})$ is the log-likelihood function when $\mathbf{Z}$ and $\mathbf{b}$ is known covariates for its response Y. $L(\alpha, \beta, \gamma, \theta; \mathbf{Z}, \mathbf{b})$ is the log-likelihood function when $\mathbf{b}$ is a known covariate for its response $\mathbf{Z}$. $L(\nu_{B(c)}, \sigma^2_{B(c)}\mathbf{b})$ is the log-likelihood function when $\mathbf{b}$ is the response of the parameters. The log-likelihood with complete data is easier to maximize than the log-likelihood with incomplete data because $L(\mu_{G(c),g}; \mathbf{Y}, \mathbf{Z}, \mathbf{b})$, $L(\alpha, \beta, \gamma, \theta; \mathbf{Z}, \mathbf{b})$ and $L(\nu_{B(c)}, \sigma^2_{B(c)}; \mathbf{b})$ can be maximized separately. So, we can implement EM algorithm[7] to estimate the parameters.

With the EM algorithm, the incomplete log-likelihood is maximized iteratively by alternating between estimating log-likelihood by its posterior expectation (E step)under the current estimates of Θ and Y, then, given that the latent variable is removed by posterior expectation, maximizing the posterior expectation of the log-likelihood function whose dependent variable exclude the latent variables(M step). The algorithm stop when Θ converges. The estimates from the final iteration are the MLE's Θ for the log-likelihood.

However, since there are two latent variables, $\mathbf{Z}$ and $\mathbf{b}$, in this model, we cannot find the closure form solution of the joint posterior distribution of $\mathbf{Z}$ and $\mathbf{b}$. Meanwhile, the posterior expectation of log-likelihood is hard to be directly calculated because the complicated form of the complete data log-likelihood(see the supplementary material). These difficulties motivated us introduce *data augmentation* and *Monte-Carlo EM*[21] into our framework.

Data Augmentation. Now, we introduce another latent variable

$$\eta_{cg} \sim N(\pi_{cg}, 1),$$

where $\pi_{cg} = \gamma + \alpha \log(r_c) + \beta \log(l_g) + \theta^T \mathbf{f}(b_c)$. η_{cg} is independently drawn from $N(\pi_{cg}, 1)$. Then we have $P(\eta_{cg} > 0) = \Phi(\pi_{cg})$. So Z_{cg} can be treated as $Z_{cg} = \mathbf{1}_{\{\eta_{cg}>0\}}$. It can be proved that

$(\eta_{cg}|Z_{cg}, b_c, \alpha, \beta, \gamma, \theta)$ is a truncated normal distribution(TN)(See the supplementary material)

$$\eta_{cg}|Z_{cg}, b_c, \alpha, \beta, \gamma, \theta \sim \begin{cases} TN(\pi_{cg}, 1, -\infty, 0), & \text{if } Z_{cg} = 0; \\ TN(\pi_{cg}, 1, 0, \infty), & \text{if } Z_{cg} \neq 0. \end{cases}$$

Then $(\eta_{cg}|Y_{cg}, b_c, \alpha, \beta, \gamma, \theta)$ is subject to

$$\begin{cases} q_{cg}TN(\pi_{cg}, 1, -\infty, 0) + (1 - q_{cg})TN(\pi_{cg}, 1, 0, \infty), & \text{if } Y_{cg} = 0; \\ TN(\pi_{cg}, 1, 0, \infty), & \text{if } Y_{cg} \neq 0. \end{cases}$$

with $q_{cg} = \frac{1-p_{cg}}{p_{cg}e^{-\lambda_{cg}}+1-p_{cg}}$, where $p_{cg} = \Phi(\pi_{cg})$. With the help of data augmentation, we steer clear of the computational complexity brought by logistic regression when maximizing the posterior expectation of $L(\alpha, \beta, \gamma, \theta; \mathbf{Z}, \mathbf{b})$(See the supplementary material) and The truncated normal distribution can be efficiently sampled with rejection sampling. Since the prior on $\alpha, \beta, \theta, \gamma$ is flat, $(\alpha, \beta, \theta, \gamma|b_c, \eta_{cg})$ is a normal distribution and easier to estimate.

E step. We use Monte-Carlo EM algorithm to fit the model, i.e., implement Monte-Carlo method to approximate the posterior expectation of log-likelihood.

$$E_{\mathbf{Z},\eta,\mathbf{b}|\mathbf{Y},\mathbf{\Theta}}\left[L\left[\mathbf{Y}, \mathbf{b}, \eta, \mathbf{Z}; \mathbf{\Theta}\right]|\hat{\Theta}^{(t)}, \mathbf{Y}\right] \approx \frac{1}{K}\sum_{k=1}^{K} E_{Z,\eta|\mathbf{Y},\mathbf{\Theta},\mathbf{b_k}}\left[L\left[\mathbf{Y}, \mathbf{b_k}, \eta, \mathbf{Z}; \mathbf{\Theta}\right]|\hat{\Theta}^{(t)}, \mathbf{Y}, \mathbf{b_k}\right],$$

where $\mathbf{b_k} = (b_{1k}, b_{2k}, \ldots, b_{Ck})$ subject to $(\mathbf{b}|\mathbf{Y}, \hat{\Theta}^{(t)})$, is the marginal posterior distribution of $\mathbf{b}$ under the data $\mathbf{Y}$ and the current estimates $\hat{\Theta}^{(t)}$. K is the number of samples. In each iteration of E-step, $\mathbf{b_k}$ is firstly and independently sampled from $(\mathbf{b}|\mathbf{Y}, \hat{\mathbf{\Theta}}^{(\mathbf{t})})$ via Metropolis-Hastings algorithm[6].With the sampled $\mathbf{b}_k$, $(\eta_{cg}|Y_{cg}, b_c, \alpha, \beta, \gamma, \theta)$ follows truncated normal distribution, whose expectation is easy to compute. In addition, posterior expectation of Z_{cg} is computed by Bayes formula given $\mathbf{b}_k$, parameters, and the observed data. Thus the expectation-maximization can be divided into three parts:(1).$\sum_{k=1}^{K} E_{Z|\mathbf{Y},\mathbf{\Theta},\mathbf{b_k}}\left[L\left[\mathbf{Y}, \mathbf{b_k}, \mathbf{Z}; \mu\right]|\hat{\Theta}^{(t)}, \mathbf{Y}, \mathbf{b_k}\right]$, (2). $\sum_{k=1}^{K} E_{\eta|\mathbf{Y},\mathbf{\Theta},\mathbf{b_k}}\left[L\left[\eta, \mathbf{b_k}; \gamma, \alpha, \beta, \theta\right]|\hat{\Theta}^{(t)}, \mathbf{Y}, \mathbf{b_k}\right]$ and (3). $\sum_{k=1}^{K} L(\mathbf{b}_k; \nu, \sigma)$. We can maximize them separately in the M step.

M step for μ. When analyzing data from different biological groups, $\widehat{\mu}_{cg}$ can be obtained via averaging weighted by the posterior mean of Z_{cg} and e^{b_c}. To be specific, we have

$$\begin{aligned} &\sum_{k=1}^{K} E_{Z|Y_{cg},\Theta,b_{ck}}\left\{\sum_{c\in G(c)}\left[L\left[Y_{cg}, b_{ck}, Z_{cg}; \mu_{G(c)g}\right]|\hat{\Theta}^{(t)}, Y_{cg}, b_{ck}\right]\right\} \\ =&\log\left(\mu_{G(c)g}\right)\sum_{c\in G(c)}\left[y_{cg}\sum_{k=1}^{K}\widetilde{Z_{cgk}}\right] - \mu_{G(c)g}l_g\sum_{c\in G(c)}\left[r_c\sum_{k=1}^{K}\widetilde{Z_{cgk}}e^{b_{ck}}\right], \end{aligned}$$

where $\widetilde{Z_{cgk}}$ is the posterior expectation of Z_{cg} given $\mathbf{b}_k$, parameters, and the observed data(See the supplementary material). $\mu_{G(c)g}$ maximize the log-likelihood function for every biological group $G(c)$ and gene g. So, the estimator of $\mu_{G(c)g}$ is given by

$$\widehat{\mu_{G(c)g}} = \frac{\sum_{c\in G(c)}\left[y_{cg}\sum_{k=1}^{K}\widetilde{Z_{cgk}}\right]}{l_g\sum_{c\in G(c)}\left[r_c\sum_{k=1}^{K}\widetilde{Z_{cgk}}e^{b_{ck}}\right]}.$$

M step for γ, α, β and θ. With the help of data augmentation, $L([\eta, \mathbf{b_k}; \gamma, \alpha, \beta, \theta]\,|\hat{\Theta}^{(t)}, \mathbf{Y}, \mathbf{b_k})$ replaces $L([\mathbf{Z}, \mathbf{b_k}; \gamma, \alpha, \beta, \theta]\,|\hat{\Theta}^{(t)}, \mathbf{Y}, \mathbf{b_k})$ which is very easy to parameterize(See the supplementary material). Since $\eta_{cg} \sim N(\pi_{cg}, 1)$, the maximization of the expectation of the log-likelihood $L([\eta, \mathbf{b_k}; \gamma, \alpha, \beta, \theta]\,|\hat{\Theta}^{(t)}, \mathbf{Y}, \mathbf{b_k})$ is equivalent to the least square estimate. The estimators of γ, α, β and θ can be found in the supplementary material.

M step for ν and σ. To find the estimator of $\nu_{B(c)}$ and $\sigma^2_{B(c)}$, we can simply implement restrained MLE with the data set $\mathbf{b} = \{b_{11}, b_{12}, \ldots, b_{1K}, b_{21}, \ldots, \ldots, b_{ck}, \ldots b_{CK}\}$ and the restriction of identifiability $\sum_c \nu_{B(c)} = 0$.

Quality Control

Quality control is implemented in our model to detect the abnormity cell in the experiment after the parameters are estimated. In most previous work, principal component analysis(PCA) is the most common method of quality control. However, PCA cannot tell us how abnormity a cell is since we judge whether a cell is abnormity with PCA by visual inspection, which is very subjective. we propose a method of quality control here based on the random batch effect in the ZIP model.

In the E-step of the Monte-Carlo EM algorithm, we have sampled a set of posterior $\mathbf{b}_k$s from $(\mathbf{b}|\mathbf{Y}, \hat{\mathbf{\Theta}})$ for each cell c in each batch $B(c)$. So, the data set $\mathbf{b}$, which is a set of number, contains $C \times K$ b_{ck}s. And then, we can derive the empirical p-value for each cell and conduct quality control with these $\mathbf{b}_k$s. The empirical p-value for b_{ck} is given by

$$p_c^k = \frac{\#\left\{c', k' : |b_{c'k'}| \geq b_{ck}\right\}}{CK}, \text{for } c' = 1, 2, \ldots, C; k' = 1, 2, \ldots, K.$$

For each cell c, define $m_c = \#\left\{k : p_c^k < 0.05\right\}$. We use Fisher's exact test [5] to test whether m_c/K is significantly larger than 0.05.

Differential Expression

We propose a likelihood ratio test method to detect differential expression of gene g between two biological groups A and B, i.e., $H_0 : \mu_{Ag} = \mu_{Bg}$ v.s. $H_1 : \mu_{Ag} \neq \mu_{Bg}$. Now, we denote

$\mathbf{y}_g = (Y_{1g}, \ldots, Y_{Cg})$. First fit the model under H_0 and get posterior estimations of p_{cg} and b_c. Under H_0, when the parameters are known $\hat{\mathcal{L}}_{1g} - \hat{\mathcal{L}}_{0g}$ is subject to $\chi^2(1)$. However, since the parameters are estimated by us, $\hat{\mathcal{L}}_{1g} - \hat{\mathcal{L}}_{0g}$ is subject to $T\chi^2(1)$, where T is an unknown constant. We need to estimate T then find the p-value for each gene. In the following steps we treat p_{cg} and b_c are fixed.

1. Let $\hat{\mathcal{L}}_{0g}$ and $\hat{\mathcal{L}}_{1g}$ be the maximized likelihood for $\mathbf{y}_g$ achieved under H_0 and H_1.

2. Randomly assign group label $x_c \in \{A, B\}$ for each cell, fit the model for $\mathbf{y}_g$ under H_1 and compute the maximized likelihoods. Repeat for M times and denote the maximized likelihoods by $\hat{\mathcal{L}}_{1g}^{(m)}, m = 1, 2, \ldots, M$.

3. Under null hypothesis, $\frac{\hat{\mathcal{L}}_{1g} - \hat{\mathcal{L}}_{0g}}{T} \sim \chi^2(1)$, where T is an unknown constant. Estimate T by $\sum_{m=1}^{M} \left[\hat{\mathcal{L}}_{1g}^{(m)} - \hat{\mathcal{L}}_{0g}\right] / M$, then we can compute p-value for this test.

After deriving the p-value for each individual gene g, we apply Benjamini-Hochberg procedure to implement FDR control and get the set of differentially expressed genes.

Visualization

We implement a weighted version of PCA to do visualization by using weighted covariance matrix. Entries of the weighted covariance matrix is calculated as

$$\hat{\Sigma}(i,j) = \frac{\sum_g (y_{ig}^* - \bar{y}_i^*)(y_{jg}^* - \bar{y}_j^*)\hat{p}_{ig}\hat{p}_{jg}}{\sum_g \hat{p}_{ig}\hat{p}_{jg}}, 1 \le i, j \le C,$$

where $y_{ig}^* = y_{ig}/E[e^{b_i}|Y, \widehat{\Theta}]$, and y_{ig} is normalized or log normalized expression level (RPKM or log(RPKM+1)) and $\hat{p}_{ig}$ is the posterior probability that $Z_{ig} = 1$. We use this matrix to replace the covariance matrix in traditional PCA.

We implement PCA on $\hat{\Sigma}$ to get the low-dimensional embedding of the data. With weighted covariance matrix specified, we can also obtain dissimilarity matrix directly and implement manifold learning method like ISOmap to visualize data.

Applications

To evaluate our method's performance, we implement it on embryonic cells' scRNA-seq data[4]. In the article, there are totally 22958 genes and 54 Cells from two biological group (8-cell and 16-cell) and two batches (run-00193, run-0088) as is shown in Table 1. For pre-processing, we delete genes with extreme high expression level (top 5%), and genes detected in less than 10% cells, which are common procedures in scRNA-seq data analysis. And 11724 genes are remained after the pre-processing.

	16-cell	8-cell
run-88 (batch 1)	27	10
run-193 (batch 2)	8	9

Table 1: Design of the scRNA-seq experiment.

Quality Control

We fit our model on the pre-processed data with our introduced zero-inflated poisson model. Linear function b_c is adopted for the dropout probability, which means $f(b_c) = b_c$. So there only contain one parameter in *theta*. It takes less than 5 minutes to run the algorithm, while it takes [10] hours to fit their model. No alter for convergence of the sampling method and the EM procedure appears. After the model is fitted, we have:

1. Estimator for θ, turns out to be less than 0, which indicates the negative correlation between the cells' detection rate and mean expression level discussed in [8].

2. Our proposed quality control procedure selects three cells, with p-values less than 10^{-10} in the fisher's exact test. These three cells are exactly the three with the lowest detection rates (proportion of nonzero genes).

Differential Expression

Our method for differential expression detection is implemented after deleting the three cells of low quality and refitting our model. After the model is fitted again with three cells discarded, we have:

1. Empirical distribution of the simulated log-likelihood ratio under null hypothesis seems to be proportional to $\chi^2(1)$ distribution. Figure 1 and figure 2 show the histogram of the log-likelihood ratio $\hat{\mathcal{L}}_{1g}^{(m)} - \hat{\mathcal{L}}_{0g}$ for two randomly selected genes.

2. Our method finds only 4 genes with Benjamini-Hochberg corrected p-values less than 0.001. While DEseq[2] finds 2 significant differential expressed genes and DEseq2[17] (adjusting for batch) finds 3 significant differential expressed genes with the same pre-processing procedure. The 4 genes detected by our method are neither detected by DEseq nor DEseq2.

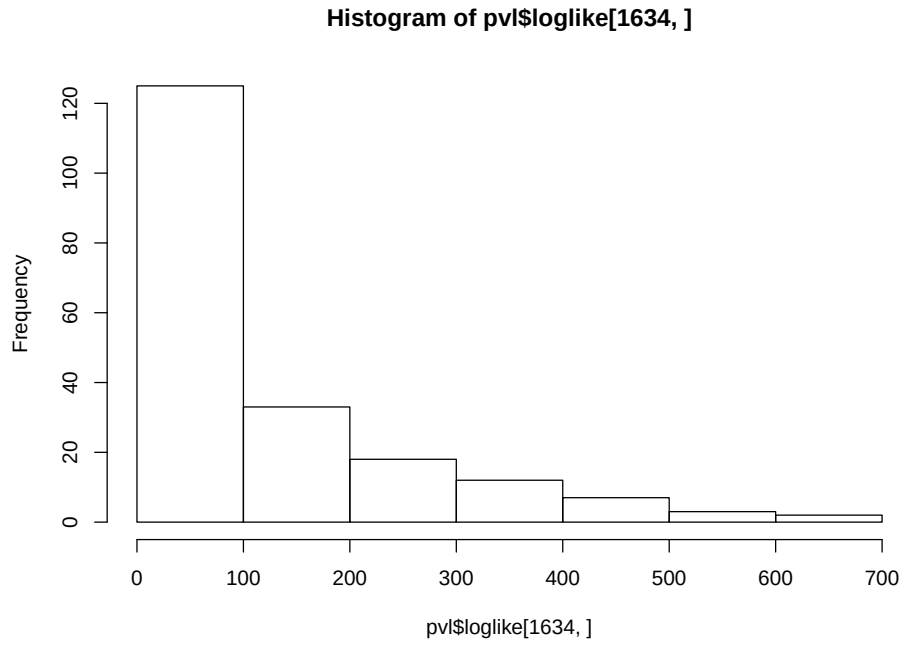

Figure 1: Histogram for the distribution of log-likelihood ratio for one specific gene

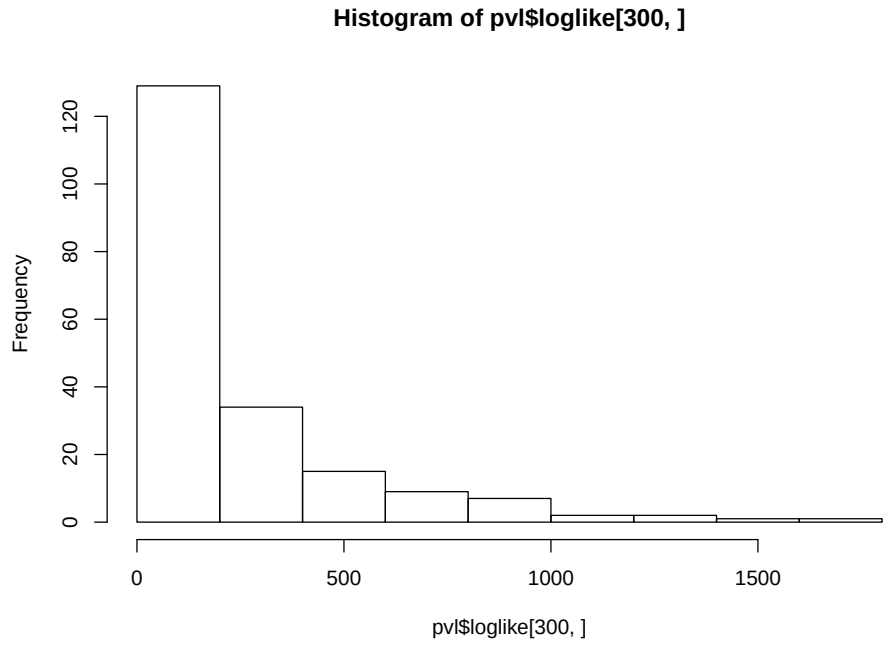

Figure 2: Histogram for the distribution of log-likelihood ratio for one specific gene

Visulization

We implement weighted PCA and manifold learning method Isomap to visualize the data with the abnormity cells and genes removed. Results are compared with PCA method and Isomap method on the pre-processed normalized raw data. Comparing figure 3 and figure 4 provides evidence that the batch effects are adequately adjusted for in these data. Isomap algorithm is also implemented joint with adjustment for our method. The dissimilarity matrix is computed by the weighted covariance matrix. The comparison of the raw data and the adjusted data is shown in figure 5 and figure 6.

Figure 3: PCA on normalized raw data

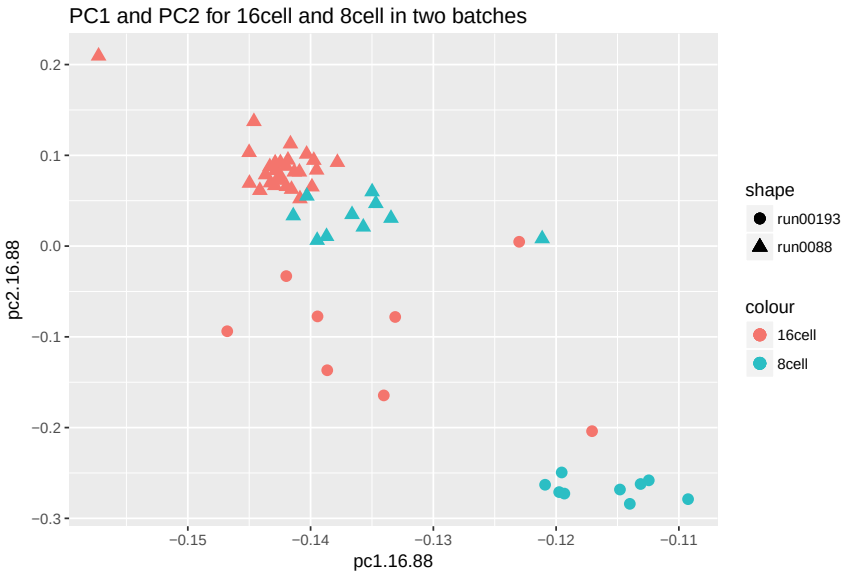

Figure 4: Our weighted PCA method.

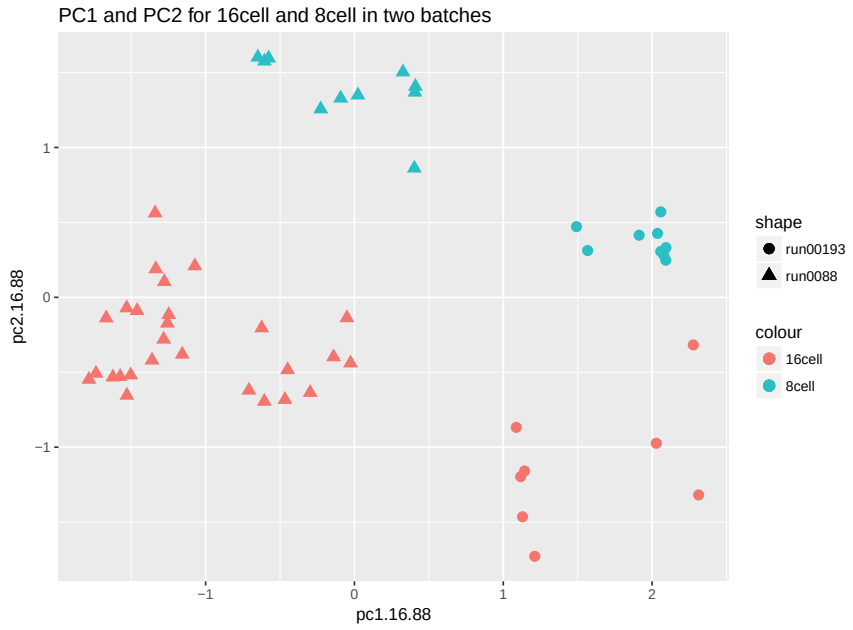

Figure 5: ISOmap on normalized raw data, with parameters of dissimilarity tuned by us.

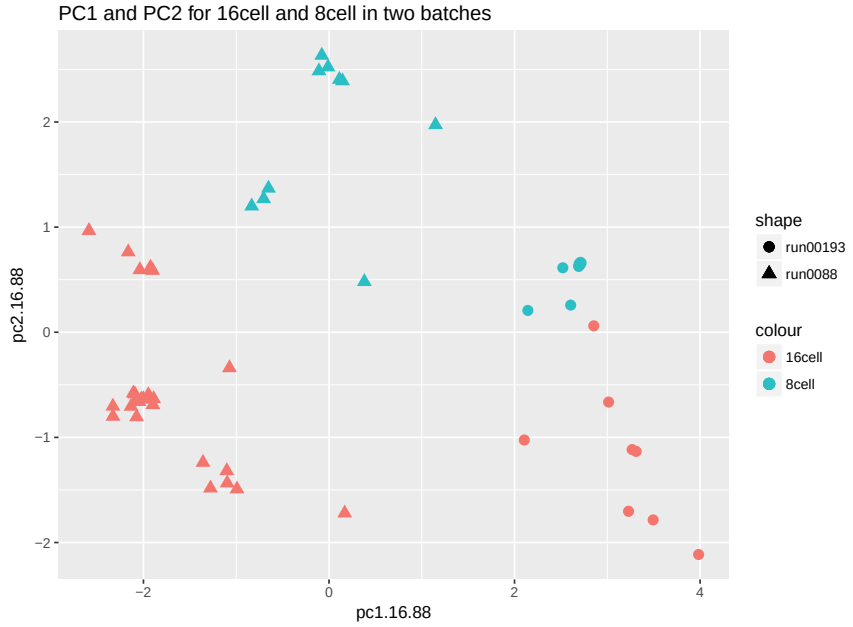

Figure 6: ISOmap on data adjusted by our method, with parameters of dissimilarity.

Conclusions

Systematic bias in RNA sequencing is a very common problem faced by researchers in the area of biological engineering. Our zero-inflated poisson model is a powerful method to account and adjust for the batch effects taking the overdispersion of dropout events into consideration. The ZIP model can somehow remove the batch effect in the real world data. With EM algorithm and data augmentation, we proof that ZIP model is feasible and computational acceptable. Quality control, differential expression analysis and visualization are easy to implemented after the model is fitted and the batch effects are adjusted.

However, the ZIP model still have some space to be improved

1. Conservative on accounting and adjusting for batch effect. A possible solution is to add more penalize to $(b_c - \nu_{B(c)})^2$ since we regard normal prior is equivalent to a penalization.

2. Though modelling technical excess zero, Poisson assumption of the true count data seems to be still improper. The true count data is overdispersed than Poisson. These could cause our method to be poor in power.

3. Beta-Poisson or zero-inflated Poisson seem to be a way out, but would introduce too much computational burden.

Acknowledgments

We thank Hao Ge, Yanyi Huang and Xiannian Zhang for helpful discussions. Molei Liu, Yue Li and Yiliang Zhang worked on the model together.

References

[1] O. Alter, P. O. Brown, and D. Botstein. Singular value decompositin for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18):10101–6, 2000.

[2] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome biology*, 11(10):R106, 2010.

[3] M. Benito, J. Parker, Q. Du, J. Wu, X. Dong, C. M. Perou, and J. S. Marron. Adjustment of systematic microarray data bases. *Bioinformatics*, 20(1):105–114, 2004.

[4] Q. Deng, D. Ramsköld, B. Reinius, and R. Sandberg. Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343(6167):193–196, 2014.

[5] R. A. Fisher. On the interpretation of 2from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[6] E. Greenberg. Understanding the metropolis-hastings algorithm. *American Statistician*, 49(4):327–335, 1995.

[7] M. R. Gupta and Y. Chen. *Theory and Use of the EM Algorithm*. Now Publishers Inc., 2011.

[8] S. C. Hicks, M. Teng, and R. A. Irizarry. On the widespread and critical impact of systematic bias and batch effects in single-cell rna-seq data. *Nordisk Medicin*, 94(1):32–41, 2015.

[9] C. Jia, D. Kelly, J. Kim, M. Li, and N. Zhang. Accounting for technical noise in single-cell rna sequencing analysis. *bioRxiv*, page 116939, 2017.

[10] C. Jia, D. Kelly, J. Kim, M. Li, and N. Zhang. Accounting for technical noise in single-cell rna sequencing analysis. *bioRxiv*, 2017.

[11] W. E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

[12] D. Lambert. Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992.

[13] E. S. Lander. Array of hope. *Nature Genetics*, 21(1 Suppl):3–4, 1999.

[14] J. A. Lee, K. K. Dobbin, and J. Ahn. Covariance adjustment for batch effect in gene expression data. *Statistics in Medicine*, 33(15):2681–95, 2014.

[15] J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010.

[16] J. T. Leek and J. D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet*, 3(9):e161, 2007.

[17] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

[18] T. O. Nielsen, R. B. West, S. C. Linn, O. Alter, M. A. Knowling, J. X. O'Connell, S. Zhu, M. Fero, G. Sherlock, and J. R. Pollack. Molecular characterisation of soft tissue tumours: a gene expression study. *Lancet*, 359(9314):1301–7, 2002.

[19] S. E. Reese, K. J. Archer, T. M. Therneau, E. J. Atkinson, C. M. Vachon, M. D. Andrade, J. P. A. Kocher, and J. E. Eckelpassow. A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics*, 29(22):2877–83, 2011.

[20] A. A. Shabalin, H. Tjelmeland, C. Fan, C. M. Perou, and A. B. Nobel. Merging two gene-expression studies via cross-platform normalization. *Bioinformatics*, 24(9):1154–1160, 2008.

[21] G. C. G. Wei and M. A. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.