Insider Immune to Batch Effects in Bulk RNA-seq Analysis

Kai, ZHAO

October 19, 2022

1 Background

The batch effects make RNA-seq analysis challenging, especially single-cell RNA-seq analysis. A number of recent studies work on this issue in the single-cell setting [1]–[3]. Even though a similar issue in bulk RNA-seq receives less attention, it still is a vital issue in bulk RNA-seq analysis worth studying. A recent study proposes a statistical method to address this concern in bulk RNA-seq analysis [4]. To my knowledge, our method can also handle batch effects from bulk RNA-seq data.

2 Origins of batch effects

Let us begin with understand the origins of batch effects in bulk RNA-seq data. For the convenience of illustration, here batch effects will refer to batch effects in the bulk RNA-seq setting without specification. Batch effects may have different origins, for example, laboratories, technical platform of sequencing, or biological conditions (e.g., donor, tissues, gender, development stages etc.). Technical batch effects (e.g., laboratories, technical platform of sequencing) usually is random and orthonormal to biological subspace [3]. However, the batch effect originating from biological conditions can interact with biological subspace, since different biological conditions can bring in heterogeneity in RNA expression. This type of batch effects may distort the underlying biological pattern of our interest.

3 Why Insider immune to batch effects

Here I seek to illustrate that why insider is immune to batch effects in bulk RNA-seq setting. To recall our approach, the log transformed expression level of gene m in tissue h of donor i with phenotype j, z_{ijhm}, is modelled by

$$\hat{z}_{ijhm} = d_i^T v_m + p_j^T v_m + t_h^T v_m + w_{jh}^T v_m, \tag{1}$$

where i, h, j, m stands for donor i, tissue h, phenotype j, and gene m, respectively. Here $d_i, t_h, p_j, w_j h, v_m$ are vectors of length K, and w_{jh} is introduced to capture interactions between phenotype j and tissues h. The objective

function following our modelling is formulated as below

$$\begin{aligned}\mathcal{L}(d,t,p,v) =& \frac{1}{2}\sum_{i,j,h,m}\left[z_{ijhm} - d_i^T v_m - p_j^T v_m - t_h^T v_m - w_{jh}^T v_m\right]^2 + \\ & \frac{1}{2}\lambda\left[(\sum_i ||d_i||_2^2 + \sum_j ||p_j||_2^2 + \sum_h ||t_h||_2^2 + \|w_{jh}\|_2^2)\right] + \\ & \frac{1}{2}\lambda(1-\alpha)\sum_k ||v_m||_2^2 + \lambda\alpha\sum_k ||v_m||_1\end{aligned} \tag{2}$$

3.1 How gene representations tackle batch effects

Here I use computing the parameters for the m-th gene as an example to illustrate our problem. When optimizing the objective function defined Equation 2 with respect to v_m, our problem has the following form

$$\mathcal{L}(v_m) = \frac{1}{2}\|\mathbf{z_m} - \mathbf{U}\mathbf{v_m}\|_2^2 + \frac{1}{2}\lambda(1-\alpha)\|\mathbf{v_m}\|_2^2 + \lambda(1-\alpha)\|\mathbf{v_m}\|_1, \tag{3}$$

where $\mathbf{U} = \mathbf{X_D}\mathbf{D} + \mathbf{X_P}\mathbf{P} + \mathbf{X_T}\mathbf{T} + \mathbf{X_W}\mathbf{W}$, and $\mathbf{z_m}$ is the expression levels of gene m across the N RNA-seq samples. To further simplify the problem, we are trying to solve the following linear regression problem

$$\hat{\mathbf{z}}_\mathbf{m} = \mathbf{U}\mathbf{v_m},$$

with the elastic net penalty introduced on $\mathbf{v_m}$.

Here we may ignore the penalty for a while for illustration. Suppose $\mathbf{z_m}$ is affected by technical batch effects due to factors, such as different laboratories or sequencing platforms. These effects are usually small in well quality-controlled RNA samples.

On the one hand, they are much smaller than the biological variation of gene $\mathbf{m}$. This is the case for all highly variable or active genes in RNA-seq data. After log-transformation, they become even subtle, so their leverages are subtle in our regression problem. In our analyses, there are several hundreds of observations $\mathbf{z_m}$ and linear regression is robust to noises. In the situation, we can reach the conclusion that $\mathbf{v_m}$ basically capture the biological variation of gene m.

On the other hand, if gene m is seldom expressed, then there are lots of zeros and few small non-zeros values in $\mathbf{z_m}$. Since we introduced elastic penalty, which encourage sparsity in $\mathbf{v_m}$, $\mathbf{v_m}$ should have lots of zeros and few small non-zeros. The effects of batch effect is well-controlled in this situation. Note in general the effects of genes seldom expressed have little contributions to the total variation of RNA-seq data matrix.

In summarize, $\mathbf{v_m}$ basically capture the variation from biological subspaces and the influence of batch effects on $\mathbf{v_m}$ is subtle and ignorable. The latent representations of covariates are linear combination of components of $\mathbf{V}$, thus immune to technical batch effects.

3.2 How latent representations of covariates handle batch effects

To re-emphasis, the latent representations of covariates are linear combinations of components of $\mathbf{V}$, thus immune to technical batch effects. Thus, in this subsection, I focus on how latent representations of covariates handle batch effects originating biologically. Here I use the latent representation for phenotypes as an example, and I believe

what will be discussed can be applied to other covariates. To learn the loading of the phenotype representation for phenotype j, we need to solve the problem below

$$\text{argmin}_{p_j} \frac{1}{2}\|\mathbf{Y_j} - \mathbf{X_j p_j}\|_2^2 + \frac{1}{2}\lambda\|\mathbf{p_j}\|_2^2,$$

where Y_j is obtained by stacking $Z_{i \in S_j}$ into a long vector of length $n(S_j) * M$, and similarly $\mathbf{X_j}$ is obtained via concating $n(S_j)$ number of Vs together, which is a $n(S_j) * M$ by K matrix. Here S_j is the set of row indices of RNA-seq samples with phenotype j, $n(S_j)$ is the number of elements in S_j, and M is the number of columns of Z. In the ridge regression problem, we derive a closed form for p_j.

The latent representations of p_j controls batch effects, since Y_j consists gene expression of several tissues from a number of donors, so p_j computed from it can control the batch effect from donor and tissue by minimizing the loss function across different tissues from different donors. Therefore, I believe the influence of batch effects on p_j is well controlled. If there is one sample for computing the latent representation of p_j, then the latent representations for p_j is greatly affected by the single sample.

In conclusion, if our latent representations for specific covariate is obtained by regressing across other covariates, then the batch effect from other covariates is controlled in computing our latent representations for the covariates.

In my opinion, the above explanation can also be applied to our covariates, such as, donor, tissue, etc.

References

[1] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[2] J. D. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Z. Macosko, "Single-cell multi-omic integration compares and contrasts features of brain cell identity," *Cell*, vol. 177, no. 7, pp. 1873–1887, 2019.

[3] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni, "Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors," *Nature biotechnology*, vol. 36, no. 5, pp. 421–427, 2018.

[4] R. Molania, M. Foroutan, J. A. Gagnon-Bartsch, L. C. Gandolfo, A. Jain, A. Sinha, G. Olshansky, A. Dobrovic, A. T. Papenfuss, and T. P. Speed, "Removing unwanted variation from large-scale rna sequencing data with prps," *Nature Biotechnology*, pp. 1–14, 2022.