Computational and Structural Biotechnology Journal xxx (xxxx) xxx

ELSEVIER

COMPUTATIONAL ANDSTRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing

Wenan Chen [a,1], Silu Zhang [b,1], Justin Williams [c], Bensheng Ju [c], Bridget Shaner [c], John Easton [c], Gang Wu [a], Xiang Chen [c,*]

[a] Center for Applied Bioinformatics, St. Jude Children's Research Hospital, Memphis, TN, United States
[b] Department of Diagnostic Imaging, St. Jude Children's Research Hospital, Memphis, TN, United States
[c] Department of Computational Biology, St. Jude Children's Research Hospital, Memphis, TN, United States

ARTICLE INFO

Article history:
Received 30 September 2019
Received in revised form 24 March 2020
Accepted 25 March 2020
Available online xxxx

Keywords:
scRNA-seq
Differential expression analysis
Batch effects
Latent batch effects
Aggregation-based methods
Fixed effect model
Mixed effect model
Surrogate variable based methods

ABSTRACT

Accounting for batch effects, especially latent batch effects, in differential expression (DE) analysis is critical for identifying true biological effects. Single-cell RNA sequencing (scRNA-seq) is a powerful tool for quantifying cell-to-cell variation in transcript abundance and characterizing cellular dynamics. Although many scRNA-seq DE analysis methods accommodate known batch variables, their performance has not been systematically evaluated. Moreover, the challenge of accounting for latent batch variables in scRNA-seq DE analysis is largely unmet. In contrast, many methods have been developed to account for batch variables (either known or latent) in other high-dimensional data, especially bulk RNA-seq. We extensively evaluate eleven methods for batch variables in different scRNA-seq DE analysis scenarios, with a primary focus on latent batch variables. We demonstrate that for known batch variables, incorporating them as covariates into a regression model outperformed approaches using batch-corrected matrix. For latent batches, fixed effects models have inflated FDRs, whereas aggregation-based methods and mixed effects models have significant power loss. Surrogate variable based methods generally control the FDR well while achieving good power with small group effects. However, their performance (except SVA) deteriorated substantially in scenarios involving large group effects and/or group label impurity. In these settings, SVA achieves relatively good performance despite occasionally inflated FDR (up to 0.2). Finally we make following recommendations for scRNA-seq DE analysis: 1) incorporating known batch variables instead of using batch-corrected data; 2) employing SVA for latent batch correction and 3) better methods are still needed to fully unleash the power of scRNA-seq.

1. Introduction

Single-cell RNA sequencing (scRNA-seq) brings single-cell level resolution to the analysis of transcriptomics. The technique has been applied in many areas, such as novel cell population discovery, cell heterogeneity dissection, and cell lineage construction [1,2]. There are two main quantification schemes for scRNA-seq: read count and unique molecular identifier (UMI) count. The UMI count has the advantage of avoiding application biases introduced by sequencing library construction, which can be approximated by a negative binomial model [3–5]. As with other high-dimensional data, accounting for the batch effects in an analysis is critical for revealing the real biological effects [6]. While the batch-effect concern is universal for all scRNA-seq analyses (recently reviewed in [7]), it is probably more prominent for differential expression (DE) analysis of scRNA-seq data, because cells from different experimental groups/conditions are typically captured separately, and this produces large collections of cells with batch effects (technical variations) embedded with underlying biological differences [8,9]. When the batch effects completely overlap with the group differences, it is difficult to distinguish their individual effects. With the fall in cost of scRNA-seq, a better design emerged with multiple batches/replicates for each group [8–10].

Several methods have been proposed to account for known batch effects in DE analysis in scRNA-seq data by incorporating

* Corresponding author at: Department of Computational Biology, St. Jude Children's Research Hospital, 262 Danny Thomas Place, Mail Stop 1135, Memphis, TN 38105, USA.
E-mail address: xiang.chen@stjude.org (X. Chen).
[1] These authors contribute equally.

https://doi.org/10.1016/j.csbj.2020.03.026
2001-0370/

batch variables as covariates in a regression model [5,11,12]. Other approaches have been developed to directly output a batch corrected matrix for downstream analysis, mostly for visualization/clustering (reviewed in [7]). Although several methods (ComBat [13], MNNCorrect [14], zinbwave [15], scMerge [16]) achieved relative good performance compared to others in a limited comparison [7], their performance in DE analysis has not been systematically evaluated.

Many methods have been developed to account for the unknown/latent batch variables for high-throughput platforms, such as SVA [17,18], RUV [19], dSVA [20], BCconf [21], and CorrConf [22]. However, scRNA-seq platforms, especially droplet-based platforms [3,4,23], generate shallow transcriptome profiles (with many zero entries and a low signal-to-noise ratio) for hundreds to thousands of single cells. Given these distinctive characteristics, the effectiveness of the general methods has not been established for scRNA-seq data. Recently, a few batch-correction methods have been proposed for DE analysis of scRNA-seq data. These include aggregation-based methods [24], nested fixed effects models [10], and nested mixed effects models [9]. The aggregation-based methods pool all cells from a batch to produce a pseudo-bulk sample and then analyze the pooled data by using approaches designed for bulk RNA-seq. Nested fixed-effect methods treat the batch effects as fixed effects nested within each group and then test the group effects for each gene. Alternatively, the batch effects can be modeled in mixed effects models, in which all cells from each batch share a random effect. Although the nested fixed-effect models and nested mixed-effect models were designed for scRNA-seq, they belong to the single-gene based methods, which ignore potential common information shared among all genes, which in turn might result in a loss of power.

Most scRNA-seq platforms produce either read count or UMI count based gene expression matrices. Although a high abundance of zeroes in the expression matrix is common with both schemes, we have shown that the UMI count can be modeled by simpler models. Moreover, the negative binomial model is a good approximation model and zero-inflated models are not needed for UMI counts [5].

In this study, we evaluated the performance of eleven representative methods (with various parameter configurations) accounting for known/latent batch effects in DE analysis in extensive simulations in UMI count based scRNA-seq datasets. We compared the performance of selected methods in an scRNA-seq dataset for Rh41 cells with multiple batches.

2. Methods

2.1. Comparison scheme and criteria

A schematic diagram of the comparison is shown in Fig. 1. We simulated two different batch effects scenarios and considered different numbers of cells as well as impurity of the group labels. FDR, statistical power, F_1-score and area under the curve (AUC) of the precision-recall curve were used to compare different methods accounting for batch effects in the DE analysis. For the AUC calculation, we restricted to the area with precision >0.8 and normalized its area to 1. The first scenario simulated two groups with three matched batches, i.e., samples were simultaneously collected for both groups for each batch. The second scenario also included two groups, each with three independent batches, i.e., all samples were collected independently. We further simulated different magnitudes of group effects and different sample sizes of cells. Finally, we simulated a group with impurity, i.e., a small portion of cells within each batch were mislabeled. This scheme represented the experimental design using fluorescence-activated cell sorting (FACS), in which 95% purity is considered high and acceptable [25]. The fold change and the total number of cells in each simulation setting are summarized in Table 1.

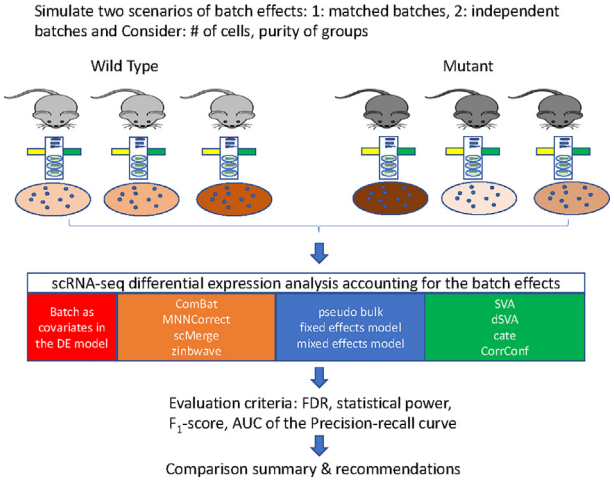

Fig. 1. Schematic diagram of the evaluation of different methods accounting for the batch effects in scRNA-seq DE analysis. Two different batch effects scenarios were simulated. Eleven major methods with different configurations were compared in terms of FDR and statistical power to detect the DE genes. We provide comparison summaries and recommendations.

2.2. Simulation of matched batches

For matched batches, we started from an scRNA-seq data set for Rh41 cells from three different batches [26]. After filtering out genes expressed at only low levels (average UMI count < 0.1 in any batch), a total of 9831 genes remained. Filtering of genes expressed at low levels was used only in the data simulation step; this simplified the model to permit a focus on comparing method performance with no need for concerns about false positives being introduced by genes with very low expression levels [11]. Genes were sorted based on the average gene count, and we selected approximately 20% of the genes in pairs for which the fold change between the two genes in the pair was close to a specified fold-change value. These gene pairs were selected so as to cover the entire expression spectrum. We randomly sampled 10% to 40% of the cells from each batch and swapped the expression vectors of the pre-selected gene pairs. In this way, we simulated the DE of genes between the selected cells and the remaining cells. In addition, we used Splatter [27] to simulate data with batch effects in an experiment whose design was similar to the matched-batch scenario. We used the default setting and provided one batch of Rh41 cells for parameter estimation. The group probability was set to 0.25 and 0.75, with three batches per group.

2.3. Simulation of independent batches

For independent batches, we followed the simulation strategy described by Lun and Marioni [24]. In this scheme, six independent batches/plates were generated, three for each group. We simulated the gene count matrix by generating counts from the negative binomial (NB) distribution. The parameters, such as the mean and dispersion of each gene and the variance of batch effects (assuming a log-normal distribution with zero mean), were estimated from the Rh41 dataset. Instead of assuming that each gene had the independent batch variables used by Lun and Marioni [24], we assumed that all genes shared the same batch variable among cells in the same batch, although we allowed different scales of batch effect in different genes by multiplying a different constant by

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

Table 1
Different simulation settings.

Batches	Group effect	Total number of cells	Impurity level	Number of replicates
Matched	Small, FC = 1.5	Small, 600	0	50
Matched	Small, FC = 1.2	Large, 12,000	0	50
Independent	Small, FC = 1.5	Small, 600	0	50
Independent	Small, FC = 1.2	Large, 12,000	0	50
Matched	Large, FC = 20	Small, 600	0	10
Matched	Large, FC = 20	Large, 12,000	0	10
Independent	Large, FC = 20	Small, 600	0	10
Independent	Large, FC = 20	Large, 12,000	0	10
Matched	Large, FC = 25	Small, 600	5%	10
Matched	Large, FC = 25	Large, 12,000	5%	10
Splatter	Default setting	Small, 600	0	50
Splatter	Default setting	Large, 6000	0	50

the batch variable for each gene. For each gene, the model can be summarized as follows:

$$f(E(y_{ijk})) = \mu + g_i + s * b_{j(i)} \quad (1)$$

where y_{ijk} denotes the expression count from sample k in batch j of group i, μ is the overall mean, g_i denotes the group effect, $b_{j(i)}$ denotes the batch variable j within group i, s is a gene-specific scaling factor, f represents the link function, and g_i is the group effect. We used the log function as the link function in the negative binomial-based simulation. Here we have omitted the gene-specific subscript for simplicity.

We chose the constant s for each gene so that the variance of the batch effect among six batches was proportional to the estimated variance of the batch effect of each gene. The number of cells per batch/plate was 50 in one group and 150 in another group in the small sample-size scenarios (giving 600 cells in total) and 1000 in one group and 3000 in another group in the large sample-size scenarios (giving 12,000 cells in total). As in the simulation of matched batches, 9831 genes were simulated, of which 20% were DE genes. We excluded simulated data sets for which the batch variables fully aligned with the group label (e.g., all positive batch variables were in one group and all negative batch effects in the other) because it would be difficult to distinguish batch and group effects.

2.4. Simulation of group impurity

We simulated the impurity scenario for the matched batches. To create mislabeling for a specified fraction of cells in each group, we switched the group label.

2.5. Evaluated methods

The methods and parameter configurations evaluated are summarized in Table 2 and are briefly described below.

2.5.1. DE analysis in general

Following the practice of Lun and Marioni [24], we used edgeR [28] for the DE analysis and included the estimated surrogate variables for batch effects as the covariates. Overall, edgeR is an efficient DE algorithm that directly uses the UMI count. Except when using aggregation-based methods, we set prior.df to 0 to infer independently the dispersion of each gene based on scRNA-seq data. We evaluated two methods for library size estimation: the total UMI per cell and the scran [29] inferred library size. For methods that return a batch corrected matrix, we used the function f. p value from the R package sva [17,18] to calculate the p-values based on the corrected matrix.

2.5.2. Analysis with known batch variables

The true batch variables were provided to each method assuming known batches. The method batch_scran was used as the reference for comparison with all other methods.

2.5.3. Methods outputting the batch corrected matrix

ComBat [13] uses a linear model to model the normalized gene expression matrix, which includes the variables of interest, such as the group variable, and the batch effects as covariates. Each gene has its own batch specific mean parameter as well as a batch specific variance parameter. Once these parameters are estimated for each gene, an empirical Bayesian adjustment across all genes are used to provide a more stable estimation of these gene specific parameters. The output of the method is a batch corrected matrix.

MNNCorrect [14] assumes similar cells in two batches can be mapped using the mutual nearest neighbors, then their differences in the gene expression vector space representing the batch effect and can be corrected by keeping one batch as a reference and subtracting the difference from the other batch. It assumes the batch variable is almost orthogonal to the group variable. The output is a batch corrected expression matrix. Note that the group information is not used in MNNCorrect, different from later surrogate variable based methods.

The method scMerge [16] identifies cell clusters within each batch and maps cell clusters of different batches using mutual nearest clusters to identify shared "cell type" across batches. Then these "cell type" labels can be included in the RUV model [19] as covariates of interest and other latent batch information estimated from the RUV model is subtracted from the expression matrix. This is called the unsupervised version because the group information or the "cell type" information is not supplied to the method. For the supervised version, the group information or the "cell type" information is directly supplied. In this case, it would be similar as an application of the RUV method to produce a batch corrected matrix. The package scMerge also provides a method to identify stably expressed genes across different batches.

The method zinbwave [15] allows modeling of the gene expression count using both gene specific and cell specific variables. The method uses a zero inflated negative binomial model to account for potential excess of zeros. The gene or cell specific variables can be either known or latent. It can optionally output a normalized expression matrix. Besides, it can also estimate the latent batch variables representing the existing but uncaptured variation from known variables of interest. Because in this study we focused on UMI counts, which has been shown that the negative binomial distribution is adequate to model their distribution [5], we set the parameter zeroinflation to false.

Note that MNNCorrect and scMerge can only be applied in the matched batch scenario because for independent batches, each

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

Table 2
Evaluated methods, package versions, and parameter configurations.

Methods	Version	Batch type	Description
scImpute	0.0.9	–	Cluster is set to 6 to reflect 6 batches. Impute threshold is default 0.5
batch, batch_scran	edgeR: 3.23.5scran: 1.10.2	known	Include the batches directly in the DE analysis using edgeR. "_scran" means scran is used to estimate the size factor, otherwise the total UMI count is used.
ComBat	3.34.0	known matched	Use the default parametric adjustments. The input is the log transformed matrix. f.pvalue from package sva is used to calculate the p-values based on the corrected matrix. This can only be applied on known matched batches.
MNNCorrect	1.2.4	known matched	Correct all the genes based on the 2,000 high variable genes selected using the function modelGeneVar.
scMerge	1.2.0	known matched	Unsupervised gene selection is used by choosing the top 2,000 stably expressed genes using the function scSEGIndex. kmeansK is set to two clusters per batch. For the supervised version, the group information is used as the "cell type", this is similar as using the RUV method.
zinbwave	1.8.0	latent	zinbwave_normalized fits a default intercept model and then uses the corrected matrix for DE analsyis. zinbwave fits a model with the group variable as the covariates and uses the extracted 20 components as surrogate batch variables. We set the zero inflation to false so only negative binomial distribution is used.
CorrConf	2.1	latent	The name has the pattern CorrConf<_k20><_scran><_ns>, "_k20" means setting the number of surrogate variables to 20, otherwise is automatically estimated by ChooseK. "_scran" means scran is used to estimate the size factor, otherwise the total UMI count is used. "_ns" means using the original count matrix without summing, otherwise 20 cells are summed into a "summed cell" to form the new count matrix.
cate	1.0.4	latent	Similar method name pattern as CorrConf. When the number of surrogate variables is not specified, CBCV from CorrConf is used to automatically estimate the number used.
dSVA	1.0	latent	Similar method name pattern as CorrConf. When the number of surrogate variables is not specified, it is automatically estimated.
SVA	3.29.1	latent	Similar method name as CorrConf. When the number of surrogate variables is not specified, it is automatically estimated.
pseudo_bulk	3.23.5	latent	Aggregate all cell counts within each batch to generate a pseudo bulk sample. Then perform the DE analysis using quasi-likelihood (QL) based method using edgeR.
fixed_effect	3.23.5	latent	The batch effects are nested within each group using the formula in edgeR ~ group + group:batch. We set the contrast to contr.sum and test whether group effect is 0. The likelihood based test is used. Scran is used to estimate the size factor.
mixed model	SAS 9.4	latent	The counts are modeled using negative binomial distribution, and the batch effects are modeled using a random Gaussian distribution in SAS. Four different combinations of test options are used: laplace_ChiSq, quad_ChiSq, PL_default_F, PL_KR_F. laplace_ChiSq is shown as mixed_model in the results. laplace and quad means the approach uses Laplace approximation and adaptive quadrature, respectively, when using the maximum likelihood estimation. PL means pseudo-likelihood estimation, default_F means the default F test, KR_F means the F test with the Kenward and Roger adjustment on the degree of freedom. quad_ChiSq and PL_KR_F failed to finish on several data sets, and we use the rest for FDR and power estimation.

batch contains only a single group label / "cell type". ComBat cannot run on independent batches because the batch variable is confounded with the group variable.

2.5.4. Aggregation based methods

Lun and Marioni proposed to aggregate/sum counts from all cells in each batch into one pseudo-bulk sample [24]. They then used quasi-likelihood for the test, as in a bulk RNA-seq analysis. We have called this method pseudo_bulk.

2.5.5. Fixed effects model

This method was proposed by Cole et al. [10]. We ignored the subscript that specified the gene. For each gene, the batches were nested within each group and a fixed effects model similar to Eq. (1) was used, with the scale parameter being absorbed into the batch variables:

$$g(E(y_{ijk})) = \mu + g_i + b_{j(i)} \tag{2}$$

where y_{ijk} denotes the expression count from sample k in batch j of group i, μ is the overall mean, g_i denotes the group effect, and $b_{j(i)}$ denotes the nested batch effect j within group i. The null hypothesis is $g_i = 0, i = 1, .., G$, where G is the total number of groups.

To make Eq. (2) identifiable, the following constraints were added:

$$\sum_{i=1}^{G} g_i = 0 \tag{3}$$

$$\sum_{j=1}^{B_i} b_{j(i)} = 0, i = 1, \cdots, G \tag{4}$$

where B_i is the number of batches within group i. The constraint (4) implied that the average batch effects were the same across groups.

It can be shown that the fixed effects model is equivalent to putting one variable for each batch in the model and testing whether the average effects across batches of each group are the same. Specifically, this model can be written as:

$$g(E(y_{ijk})) = p_{j(i)} \tag{5}$$

This model has the same number of free parameters as in Eq. (2), with $p_{j(i)} = \mu + g_i + b_{j(i)}$. The null hypothesis is equivalent to $\frac{1}{B_1}\sum_{j=1}^{B_1} p_{j(1)} = \frac{1}{B_i}\sum_{j=1}^{B_i} p_{j(i)}, i = 2, .., G$. With the above null hypothesis, it is clear that that when there is no group effect but the average batch effects are different, the null hypothesis will still be rejected, which results in inflated type I error.

The model for the matched batches can be represented as follows:

$$g(E(y_{ijk})) = \mu + g_i + b_j \tag{6}$$

with the constraints

$$\sum_{i=1}^{G} g_i = 0 \tag{7}$$

$$\sum_{j=1}^{B} b_j = 0 \tag{8}$$

where b_j is the batch effect for each batch $j, j = 1, \cdots, B$. Thus, the nested fixed effects model includes the matched-batch model as a reduced model. This explains the good performance of this nested model when applied to the data for simulated matched batches.

However, when the batches are independent and few, the assumption of the same average batch effect among groups might be violated, leading to an increase in false positives, as shown in the simulations.

2.5.6. Mixed effects model

The model is similar to that in Eq. (2). The difference is that it assumes the batch effect $b_{j(i)}$ to be a random variable, and these are usually assumed to follow a normal distribution. Therefore, there is no hard assumption that the average batch effect in the given data is the same across groups, even though, on the population level (when the number of batches is infinite), we assume the average to be the same. We used a negative binomial distribution for the count and fitted the mixed model using SAS PROC GLIMMIX. We evaluated different options in the fitting, including maximum likelihood estimation using Laplace approximation or adaptive quadrature, and pseudo-likelihood estimation with the default F test or the F test with the Kenward and Roger adjustment on the degree of freedom. Because of the high computational complexity, mixed effects models were executed only on 10 replicates in small sample-size scenarios. Moreover, a fraction of the data set failed to converge and was excluded from the FDR/power calculations.

2.5.7. Surrogate variable based methods

These methods aim to estimate the surrogate variables based on the data matrix with high-dimensional features (gene expression in this application) to uncover the unobserved batch effects. The primary assumption is that only a small set of genes are differentially expressed between distinct groups (i.e., there is a sparsity of DE genes). In this study, we evaluated SVA [17,18], cate [30], dSVA [20], and CorrConf [22], which were either widely adopted approaches or recently published methods that were claimed to have good performance. Briefly, SVA iteratively estimates the probability of each gene being affected only by the batch effect and not by the group effect and then performs a weighted singular value decomposition on the data matrix to estimate the surrogate variables. The cate method first estimates the coefficients/loadings of batch effects by using a factor analysis and then estimates the batch variables by using a robust regression under the sparse group-effect assumption. dSVA first performs singular value decomposition on the residual matrix after regressing out the variables of interest and then estimates the batch variables by using a regression that has connections to the restricted least squares method. CorrConf is an extension of the method BCconf [21], which corrects a bias in the cate method, especially when the confounding batch effect is weak. Because CorrConf can also be applied to independent samples and estimates the number of surrogate variables faster than does BCconf, only CorrConf was included in the comparison.

Because all surrogate variable based methods implicitly or explicitly assume a Gaussian distribution for the data matrix, we transformed the gene expression data matrix before applying these methods. Specifically, we used $log_2(TPM + 0.1)$ as the input to different methods, where *TPM* represents transcripts (UMI count) per million. Finally, in the DE analysis, the estimated surrogate variables were used as covariates for the batch effects, with edgeR being used with the likelihood ratio test. For the simulated data with a large number of cells (approximately 2000) in each batch, we sorted cells by total UMI within each batch and summed 20 cells into a new aggregated pseudo-cell. Empirical evidence indicated that the pseudo-cells achieved similar or better efficiency in the surrogate variable estimation and similar or improved DE analysis performance in simulations, as compared to the raw cell-count matrix (see Results). The library sizes were estimated using scran or the raw total UMI. The number of surrogate variables included in the DE analysis was either estimated by each method or fixed at 20.

When the number of cells is large (>10,000), generating pseudo-cells by aggregating a predefined number of cells (20 in our evaluation) can both improve FDR control when using surrogate variable based methods and substantially reduce the computational burden (see Results section for details). Although the exact reason for the improved performance is not known, we hypothesize that cell aggregation reduces the data sparsity, which improves the fit to the normal distribution, a common assumption for surrogate variable based methods [17,18,20–22,30].

2.6. Data analysis in Rh41 cells

The protocol described by Chen et al. [5] was followed to sort Rh41 cells into two groups by FACS using the CD44 cell-surface marker. These groups were designated $CD44^{low}$ and $CD44^{high}$. The sorting and scRNA-seq experiments were performed on three independent cultures of Rh41 cells and generated three matched/-paired batches (giving six scRNA-seq datasets in total). For scRNA-seq data, we applied a loose threshold to filter genes: at least 10 cells with nonzero values out of >20,000 cells in the data. We also generated bulk RNA-seq datasets (independent of the scRNA-seq datasets) by using the same sorting protocol. Two evaluation schemes were used. In the first evaluation, we applied different methods to the scRNA-seq data from two batches, assuming unknown batch information, and used the DE genes identified in the remaining batch for validation. As both the $CD44^{low}$ and $CD44^{high}$ populations used for validation were derived from a single batch, no batch correction was needed for DE analysis. In the second evaluation, we performed DE analysis on all three batches, again assuming unknown batch information, and compared the results to those for the DE genes derived from the bulk RNA-seq analysis (using edgeR with TMM normalization [31] and with the paired information). We evaluated the power to recover DE genes detected in bulk RNA-seq analysis with FDR cutoffs of 0.05 and 0.1.

3. Results

3.1. Representative configurations of evaluated methods

Among all evaluated parameter configurations (Figs. S1–S8), we identified a good representative configuration for each method for comparison purposes. We found that scran-inferred size factors reduced the FDR in most cases, especially for independent batches. Therefore, all the representative configurations used scran except for the pseudo_bulk and mixed effects models implemented in SAS, and those methods output the batch corrected matrix. Even though scran estimation of the size factor is generally beneficial for DE analysis, we have identified certain scenarios in which scran normalization leads to an inflated FDR, which suggests that more improvements are needed for proper size-factor estimation, especially in the context of batch effect estimation.

Because of the high abundance of zeros in scRNA-seq data, it is often assumed that imputation will help overcome this drawback and provide more transcriptomic information. Consequently, we evaluated a hypothesis that adding an imputation step before batch effects removal would further improve DE analysis. To this end, we imputed the count matrix by using scImpute [32] then performed DE analysis with the true batch information. We compared the results of the analysis with and without imputation. Surprisingly, our comparison revealed that, instead of improving performance, scImpute either reduced the power or inflated the

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

type I error (Tables S1–S5). Consequently, we evaluated all methods by using the raw counts.

For surrogate variable based methods, there were substantial differences in the number of surrogate variables reported by the individual methods. Moreover, using these automatically inferred surrogate variables often resulted in poor performance (especially in the small sample-size scenarios). To provide a meaningful comparison, we reported the performance by using 20 surrogate variables for all surrogate variable based methods; this empirically achieved a good tradeoff between controlling the FDR and maintaining the power. For large sample-size scenarios, using surrogate variable based methods with the raw data was computationally expensive and yielded no significant improvement in performance when compared to the pseudo-cell strategy (Figs. S1–S8). Therefore, the pseudo-cell aggregated data was used for all representative surrogate variable methods. Table 3 summarized the average FDR and relative power of these representative methods with different simulation settings. The average F1-score and AUC were reported in Table S1–S5.

3.2. Methods with known batches

Fig. 2 shows the results of small group effects using large number of cells. Compared to batch_scran which accounts for the batches in a regression model, methods that output batch corrected matrix (ComBat, MNNCorrect, scMerge, scMerge, scMerge_supervised, zinbwave_normalized) either had inflated FDR, or reduced power (Fig. 2a and c). Similar suboptimal performance can be seen using F_1-score or the AUC of the precision-recall curve (Fig. 2e and g). This observation is expected because the authors of these packages cautioned potential suboptimal performance in DE analysis (e.g., MNNCorrect) or recommended to use the corrected matrix for visualization and clustering analysis (e.g., zinbwave), as evaluated by Tran et al. [7]. Moreover, even with knowledge of the true batch variable, these methods (ComBat, MNNCorrect, scMerge, scMerge_supervised) had either similar or often worse performance than surrogate methods that estimated the batch variable, such as SVA_k20_scran.

The results of small sample size (Figs. S9–S11) show similar patterns as that of large sample size. Due to the requirement of true batch information (ComBat, MNNCorrect, scMerge) and their inferior performance (ComBat, MNNCorrect, scMerge and zinbwave), we did not focus on these methods in the analysis for latent batches.

3.3. Evaluation of latent batches of large sample size

3.3.1. Small group effects

In matched-batch scenarios (Fig. 2a, c, e and g), all methods achieved good FDR control (Fig. 2a). The pseudo_bulk method showed substantial power loss, whereas other methods achieved power comparable to batch_scran (Fig. 2c). Similarly, the pseudo_-

Table 3
FDR and relative power of representative methods.

Methods	Small group effect										Large group effect	
	Matched		Independent		Splatter		Matched		Independent		Impure	
	S	L	S	L	S	L	S	L	S	L	S	L
FDR												
batch_scran	0.041	0.042	0.042	0.043	0.064	0.054	0.047	0.073	0.039	0.044	0.045	0.073
scImpute_batch_scran	0.044	0.049	0.187	0.132	0.525	**0.511**	**0.324**	NA	**0.216**	NA	**0.204**	NA
ComBat	0.107	0.078	NA	NA	0.071	0.062	0.117	0.119	NA	NA	0.142	0.142
MNNCorrect	0.034	0.048	NA	NA	0.042	0.048	**0.502**	**0.540**	NA	NA	**0.497**	**0.524**
scMerge	**0.306**	**0.453**	NA	NA	**0.795**	NA	**0.576**	**0.606**	NA	NA	**0.506**	**0.825**
zinbwave_normalized	**0.355**	**0.650**	**0.205**	**0.457**	0.046	0.040	**0.357**	**0.716**	0.102	**0.339**	**0.502**	**0.760**
zinbwave	**0.324**	**0.738**	**0.236**	**0.526**	0.064	0.070	**0.316**	**0.818**	0.169	**0.615**	**0.419**	**0.842**
CorrConf_k20_scran	0.063	0.045	0.046	0.042	0.074	0.048	0.108	0.119	0.062	0.088	0.122	0.051
cate_k20_scran	0.097	0.061	0.068	0.058	0.090	0.049	0.094	0.154	0.054	0.057	**0.365**	0.112
dSVA_k20_scran	0.095	0.057	0.064	0.057	0.072	0.047	0.108	0.152	0.058	0.121	**0.237**	0.060
SVA_k20_scran	0.044	0.051	0.042	0.075	0.069	0.044	0.049	0.104	0.039	0.125	0.048	0.157
pseudo_bulk	0.000	0.000	0.033	0.001	0.007	0.000	0.002	0.000	0.014	0.069	0.003	0.000
fixed_effect	0.050	0.042	**0.243**	**0.503**	0.088	0.059	0.055	0.069	0.155	**0.422**	0.056	0.072
mixed_effect	0.056	NA	0.085	NA	0.028	NA	NA	NA	NA	NA	NA	NA
Relative power												
batch_scran	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
scImpute_batch_scran	0.842	0.969	1.161	0.992	1.834	1.182	1.000	NA	1.000	NA	1.000	NA
ComBat	0.964	0.940	NA	NA	0.996	0.968	1.000	1.000	NA	NA	1.000	1.000
MNNCorrect	**0.641**	0.814	NA	NA	0.998	0.966	1.000	1.000	**0.000**	NA	1.000	1.000
scMerge	**0.001**	**0.090**	NA	NA	**0.007**	NA	1.000	1.000	**0.000**	NA	1.000	0.873
zinbwave_normalized	1.073	**0.718**	1.004	**0.683**	1.022	**0.754**	1.000	1.000	1.000	1.000	1.000	1.000
zinbwave	1.182	0.980	1.138	1.003	1.007	1.003	1.000	1.000	1.000	1.000	1.000	1.000
CorrConf_k20_scran	1.012	0.950	0.967	0.987	0.973	0.981	0.948	0.987	**0.770**	1.000	**0.013**	**0.404**
cate_k20_scran	1.079	0.920	1.046	0.997	1.004	0.997	1.000	1.000	1.000	1.000	**0.021**	1.000
dSVA_k20_scran	1.085	0.995	1.045	1.001	0.996	0.996	1.000	0.999	0.959	1.000	**0.014**	0.874
SVA_k20_scran	0.897	0.956	**0.723**	0.998	0.906	0.972	1.000	1.000	1.000	1.000	1.000	1.000
pseudo_bulk	**0.000**	**0.000**	**0.317**	**0.069**	**0.483**	**0.404**	1.000	1.000	0.998	0.998	1.000	1.000
fixed_effect	0.973	1.010	1.108	0.995	1.025	1.001	1.000	1.000	1.000	1.000	1.000	1.000
mixed_effect	**0.694**	NA	0.956	NA	**0.743**	NA	NA	NA	NA	NA	NA	NA

S: small number of cells; L: large number of cells; NA: not computed.
For FDR, regular font indicates FDR $\leq$ 0.08, underlined font indicates 0.08 < FDR $\leq$ 0.2, bold font indicates FDR > 0.2
For power, regular font indicates relative power $\geq$0.9, underlined font indicates relative power $\geq$0.8 but <0.9, bold font indicates relative power <0.8.

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

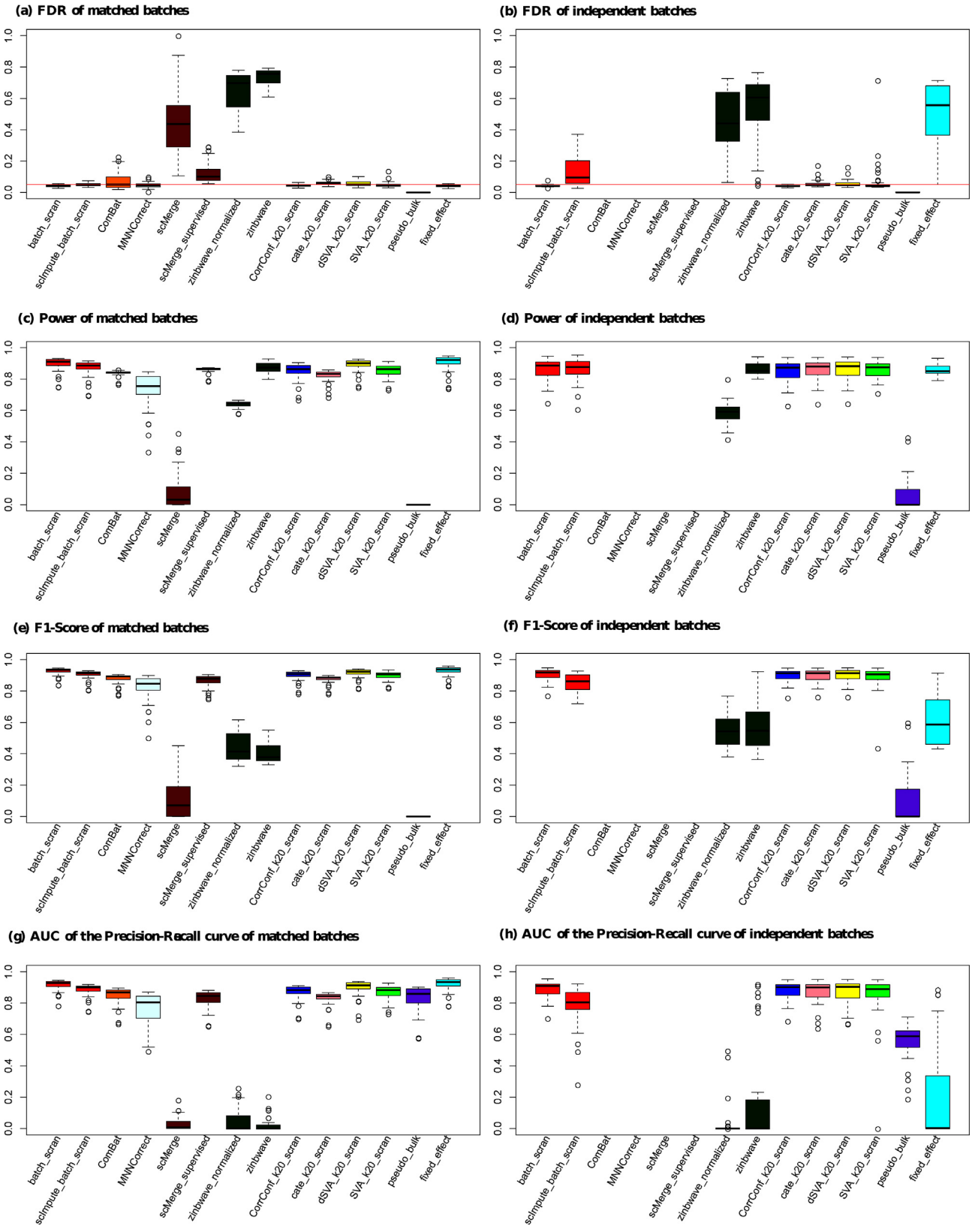

Fig. 2. FDR and power from the simulation of the large sample size and small group effects. a) FDR of matched batches; b) power of matched batches; c) FDR of independent batches; d) power of independent batches; e) F_1-score of matched batches; f) F_1-score of matched batches; g) AUC of the Precision-Recall curve of matched batches of independent batches; h) AUC of the Precision-Recall curve of matched batches of independent batches. The FDR, power, F_1-score, and AUC of each method is plotted as a boxplot based on replications. For the FDR, the redline is the nominal threshold of 0.05. A large deviation from this line indicates either inflation or deflation of the FDR.

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

Figure 3

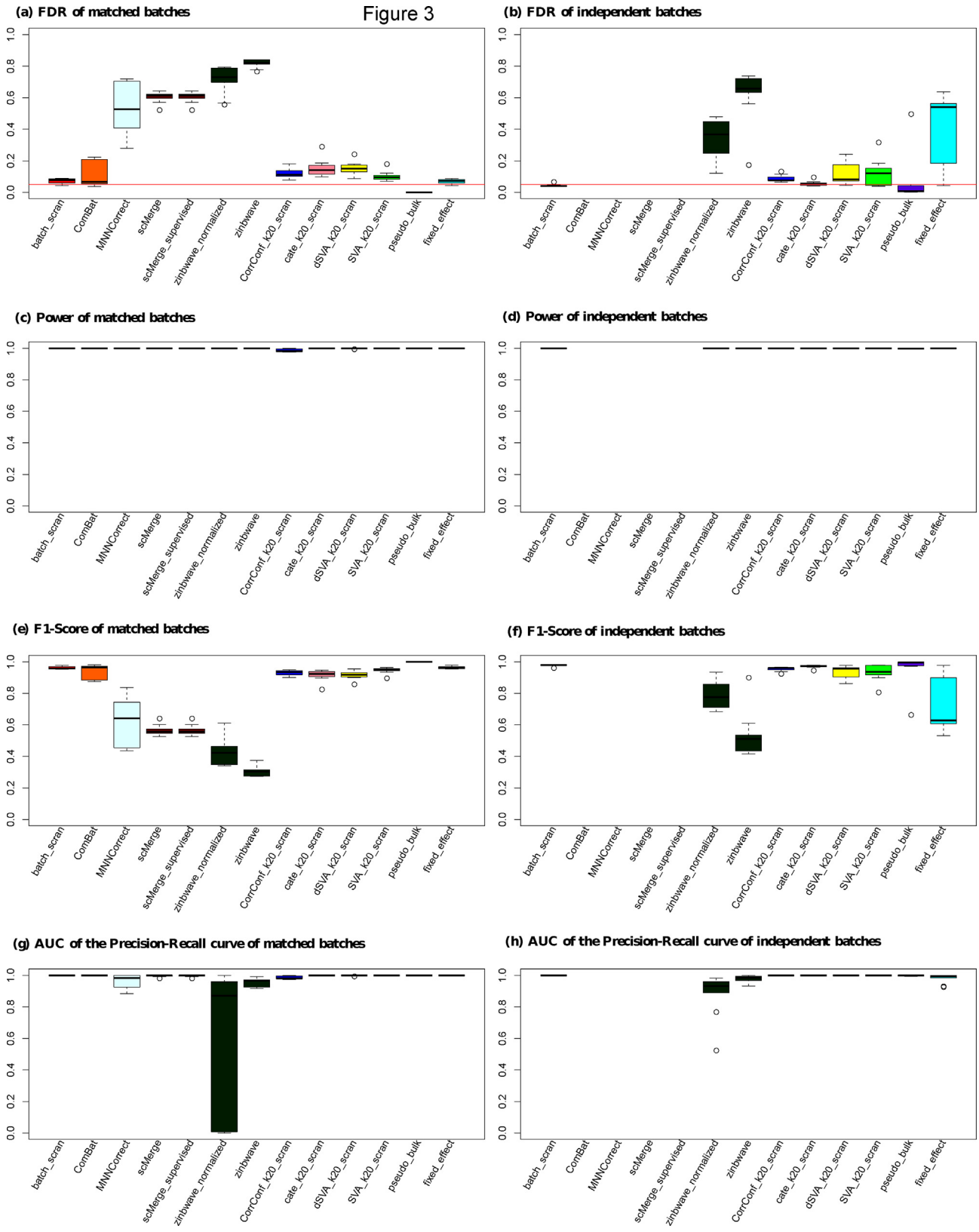

Fig. 3. FDR and power from the simulation of the large sample size and large group effects. a) FDR of matched batches; b) power of matched batches; c) FDR of independent batches; d) power of independent batches; e) F_1-score of matched batches; f) F_1-score of matched batches; g) AUC of the Precision-Recall curve of matched batches of independent batches; h) AUC of the Precision-Recall curve of matched batches of independent batches. The FDR, power, F_1-score, and AUC of each method is plotted as a boxplot based on replications. For the FDR, the redline is the nominal threshold of 0.05. A large deviation from this line indicates either inflation or deflation of the FDR.

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

bulk method showed the worst performance in terms of the F_1-score (Fig. 2e). However, it only had a minor loss in AUC (Fig. 2g), which is consistent with the observation from Lun et al. [24]. This observation suggested that although the peudo_bulk approach produced a largely correct gene rank, it is over-conservative in measuring the significance. The Splatter-based simulation yielded similar results (Figs. S7 and S8).

In scenarios with independent batches, the surrogate variable based methods (CorrConf, cate, dSVA, and SVA) achieved good performance in FDR control, power, F_1-score and AUC although SVA occasionally showed inflated FDR (Fig. 2b, d, f and h). Conversely, the fixed effects model showed FDR inflation (Fig. 2b), as well as a clear loss in F_1-score and AUC (Fig. 2f and i). The pseudo_bulk method again suffers from substantial loss in power (Fig. 2d), F_1-score (Fig. 2f) and a lower AUC (Fig. 2h).

3.3.2. Large group effects

When the group effects are large, all the evaluated methods accounting for latent batches achieved near-perfect power in recovering DE genes in the matched-batch scenarios. Although the surrogate methods showed moderate FDR inflation (Fig. 3a and c), they still achieved close to optimal performance in terms of F_1-score and AUC. A similar trend was found in the independent-batch scenario (Fig. 3b and d), with the following exceptions: fixed effects models showed severe FDR inflation, whereas one surrogate method (cate) controlled the FDR properly.

3.3.3. Group impurity

This scenario approximated a DE analysis in which the group label was not 100% accurate. An incorrect group label can result from impurity in a FACS experiment or from incorrect group assignment in a clustering analysis, which are common occurrences in real data analysis. Fig. 4 shows the FDR and power when approximately 5% of the cells in each batch are incorrectly labeled. We evaluated the matched-batch scenario. The aggregation method (pseudo_bulk) and the fixed effects method performed well in this setting. CorrConf and dSVA showed substantially reduced power, because both of these methods captured the true group label information in the estimated surrogate variables, which subsequently resulted in a major reduction in power to recover DE genes after (improperly) accounting for surrogate variables (Fig. 4). The cate method maintained the power well, perhaps because it uses robust regression when estimating the batch information. However, in applications with the raw data (without aggregating pseudo-cells), the power of CorrConf, cate, and dSVA was close to 0 (Fig. S5), which indicated that the true group labels were almost perfectly captured, although it should be noted that the annotated (impure) group information was included in the inference of the surrogate variables. Conversely, by selecting genes that were probably not differentially expressed between groups, SVA (one of the surrogate variable methods) remained unaffected by the mislabeling, showing little change in terms of FDR control and detection power.

3.4. Evaluation results of latent batches in small sample-size scenarios

The results with small sample sizes were generally consistent with those with large sample sizes. Therefore, we focused on results specific to the simulation of a small number of cells.

3.4.1. Small group effects

The results for matched batches and independent batches are shown in Fig. S9. Mixed effects models were included because the computational burden was manageable. The mixed effects models showed loss of power, especially for the matched batches and in Splatter-based simulations (Fig. S7). For other methods, the results were similar to those obtained using large numbers of cells, except that the FDR was moderately inflated for several surrogate based methods. This inflation might have been caused

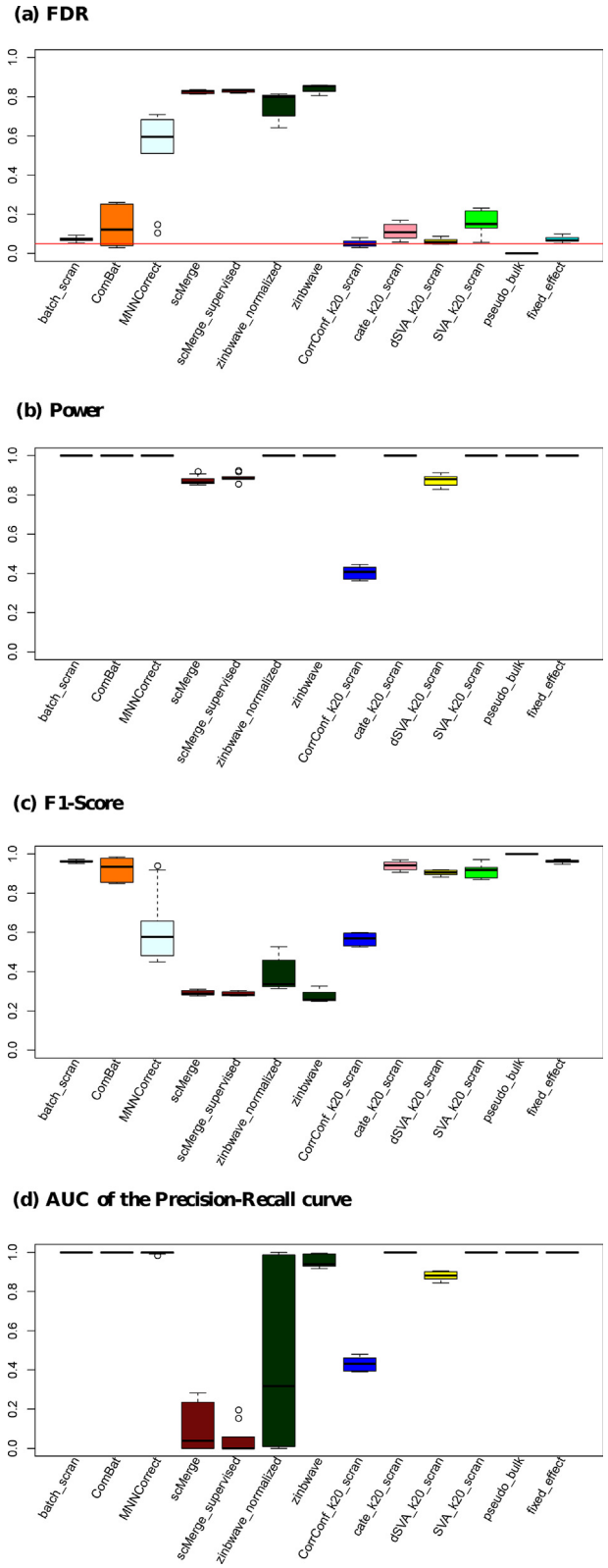

Fig. 4. FDR (a), power (b), F_1-score (c) and AUC of the Precision-Recall curve (d) from the simulation of the large sample size and impure group labels with matched batches. The FDR, power, F_1-score, and AUC of each method is plotted as a boxplot based on replications. For the FDR, the redline is the nominal threshold of 0.05. A large deviation from this line indicates either inflation or deflation of the FDR.

by a less accurate estimation of the batches with a small sample size.

Our analysis revealed that, for individual genes, certain mixed effects models (e.g., quad_ChiSq) can have an inflated FDR, especially in scenarios with independent batches. This might be caused by the large number of batches required by these methods in order for them to estimate accurately the batch effects based on a single gene. Our simulation, which approximated practical scRNA-seq data, had only three batches per condition. The observed FDR inflation was consistent with the results of McNeish et al. [33].

3.4.2. Large group effects

Results for matched batches and independent batches are shown in Fig. S10. By including those DE genes in the surrogate variable inference, CorrConf and dSVA lost power with independent batches, indicating that the inferred surrogate variables captured both the batch and the group information to some extent. SVA and cate seems to be robust in this scenario, achieving near-optimal F_1-score and AUC.

3.4.3. Group impurity

Similar to the results for large sample-size scenarios, all surrogate variable based methods except SVA showed essentially zero power, indicating perfect capture of the true group information in the estimated surrogate variables (Fig. S11).

3.5. Simulation result summary

For known batch information, incorporating the batch information as covariates in a regression model outperformed approaches working on the batch corrected matrix. Among methods designed for latent batch correction, the surrogate variable based methods, such as SVA_k20_scran, achieved a relatively good balance between FDR control (which was slightly inflated in certain scenarios) and good power in scenarios with small group effects. CorrConf and dSVA exhibited power loss in scenarios with large group effects. Moreover, CorrConf, cate, and dSVA may have substantial power loss with group impurity. These are potentially due to the capture of the group information in the estimated surrogate variables. By focusing on genes likely not differentially expressed (among groups), SVA was robust to this concern, although it could have a moderately inflated FDR. The pseudo_bulk aggregation method was usually over-conservative with respect to FDR control, resulting in substantial power loss with relatively small group effects. The fixed effects model worked well when the assumption (e.g., that the batch effects were the same for two groups) was satisfied; otherwise, it could result in a highly inflated FDR. The mixed effects model alleviated the problem of inflated FDR in the fixed effects model but also lost power, especially with matched batches.

Recommendations: due to the robustness of SVA under different scenarios, we recommend SVA for adjusting for latent batch effects. When users are confident that the group information is highly accurate, cate is also a good candidate for adjusting for latent batch effects. More details about the advantages, limitations, and recommendations are summarized in Table 4.

The FDR, Power, F_1-score and AUC plots for all configurations among the evaluated approaches are summarized in Figs. S1–S4 and Tables S1–S3.

3.6. DE analysis of $CD44^{high}$ and $CD44^{low}$ subpopulations of Rh41 cells

We applied the methods to a dataset derived from three batches of Rh41 cells sorted into $CD44^{high}$ and $CD44^{low}$ subpopulations. First, for each method, we compared the DE genes detected in two batches of data with the DE genes detected in the third batch (Table 5). In this setting, batch_scran (with true batch information provided) detected the most DE genes (10090 with 7711 confirmed in the validation set, F_1 score = 0.776), followed by SVA_scran (9904, with 7432 matched, F_1 score = 0.755). SVA_scran is also accurate in this setting, with a precision (0.750) approaching that of the batch_scran (0.764). In contrast, CorrConf_scran (3260, with 2320 matched, F_1 score = 0.356) and dSVA_scran (3139, with 2430 matched, F_1 score = 0.376) detected substantially fewer DE genes, probably as a result of impurity of the sorted populations [26]. Although the aggregation method (pseudo_bulk) has higher precision (0.938) when compared to other approaches, it detects far fewer DE genes (403, with 378 matched, F_1 score = 0.074), which

Table 4
Summary of evaluated methods.

Methods	Advantage	Limitation	Recommend application
ComBat, MNNCorrect, scMerge	Good for combining data sets from different sources for visualization and clustering	It is suboptimal to use the batch corrected matrix for DE analysis	Clustering, visualization of data from different sources/batches
zinbwave	Useful for modeling non-UMI based scRNA-seq	Large inflated FDR or reduced power in DE analysis with latent batches	DE analysis for non-UMI based scRNA-seq with no need for latent batch correction
CorrConf	Good control of FDR and high power when the group effects are small	Inflated FDR or reduced power when the group effects are large or the group is impure	DE analysis for moderate effects or the group information is highly accurate. Can be used together with SVA for a robust check
cate	Good or slightly inflated FDR and high power when the group effects are small	Inflated FDR or reduced power when the group effects are large or the group is impure	DE analysis when the group information is highly accurate. Can be used together with SVA for a robust check
dSVA	Good or slightly inflated FDR and high power when the group effects are small	Inflated FDR or reduced power when the group effects are large or the group is impure	DE analysis for moderate effects or the group information is highly accurate. Can be used together with SVA for a robust check
SVA	Good control of FDR and high power when the group effects are small; it is also little affected by the group label purity	Occasionally not very stable	Good candidate for DE analysis. Can be used together with cate/CorrConf /dSVA for a robust check
pseudo_bulk	Superfast, easy to apply	Low power	Good for identifying strong DE genes
fixed_effect	fast	Need to assume the average batch effects are similar between groups	DE analysis when we are sure the average batch effects per group are similar, such as in a paired/blocked design
mixed_effect	Can have higher power than pseudo bulk	Very slow for a large number of cells, and the power is low	When the cell number per batch is small (e.g., $\langle 1\,0\,0$) and the number of batches is large (e.g., ≥ 5) and a mixed model is strongly preferred because of other modeling aspects.

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

Table 5
Comparison on real data with two batches as discovery and one batch as validation. TPM ≥ 1 is applied to the single-cell results.

Methods	DECount	TPCount	Precision	Recall	F_1 Score
batch_scran	10,090	7711	0.764	0.788	0.776
pseudo_bulk	403	378	0.938	0.039	0.074
CorrConf_scran	3260	2320	0.712	0.237	0.356
cate_scran	7843	5604	0.715	0.573	0.636
dSVA_scran	3139	2430	0.774	0.248	0.376
SVA_scran	9904	7432	0.750	0.759	0.755

# of DE genes in validation set: 9788.

is consistent with the power loss noted in the simulations. Moreover, all DE genes reported by the aggregation method (pseudo_bulk, 403) is also recovered by SVA_scran (Fig. 5a). Similarly, SVA_scran recovers majority of DE genes reported by other evaluated methods (CorrConf_scran: 2654/3260; cate_scran: 7267/7843; dSVA_scran: 2892/3139, Fig. 5a). The recovery is even higher when measured by the DE genes confirmed in the validation set (pseudo_bulk: 378/378; CorrConf_scran: 2211/2320; cate_scran: 5533/5604; dSVA_scran: 2392/2430), suggesting that SVA_scran is a good candidate to account for latent batch effects in real data with potential label impurity.

A similar pattern was observed in the second evaluation, in which we compared the detected DE genes (using all three batches of scRNA-seq data) with the bulk RNA-seq derived DE genes (Table 6 and Fig. 5b). As in the first evaluation, CorrConf_scran and dSVA_scran recovered substantially fewer DE genes than did cate_scran or SVA_scran. The R^2 between the group label and the estimated surrogate variables from CorrConf_scran and dSVA_scran was 0.95 and 0.92, respectively, suggesting that their inferred surrogate variables essentially captured the underlying group information.

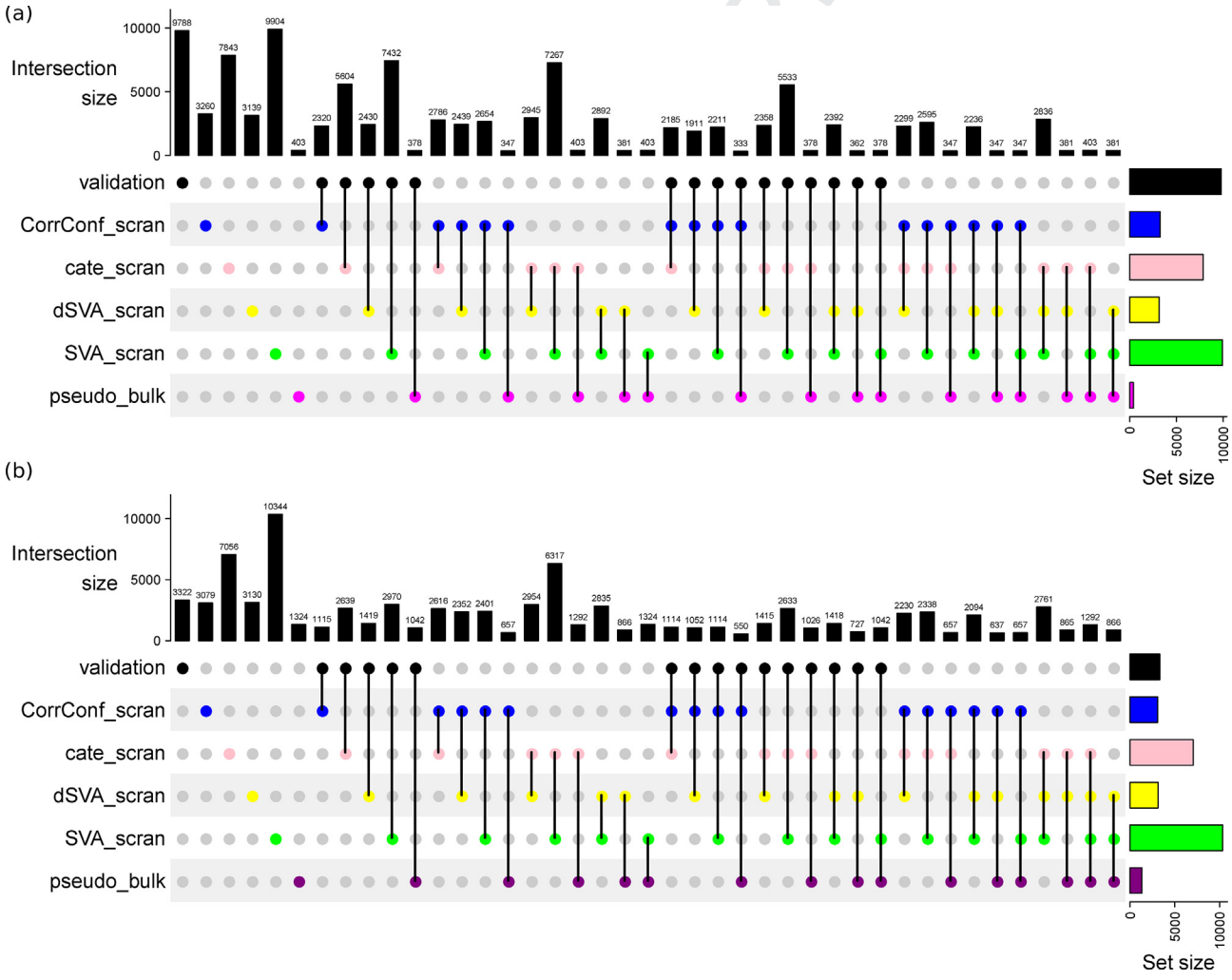

Fig. 5. UpSet plot showing the intersections of DE genes among different methods. In each UpSet plot, the bar height in the top panel indicates the size of a specific intersection. The bubbles below each bar with non-gray color indicate which sets are in the intersection. A line is drawn to connect those non-gray bubbles when there are at least two different sets in the intersection. The columns of bars and bubbles are sorted by the number of sets in the intersection. a) UpSet plot showing the number of DE genes for each method and their intersections when using the third single-cell RNA-seq data set used as the validation data set; b) UpSet plot showing the number of DE genes for each method and their intersections when using the bulk RNA-seq data used as the validation data set.

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

Table 6
Comparison on real data with three batches, using bulk RNA-seq as the ground truth. TPM ≥ 1 is applied to single-cell results and FPKM ≥ 1 is applied to the bulk RNA-seq results, with FDR cutoffs of 0.05 and 0.1.

Methods	DECount	TPCount	Precision	Recall	F_1 Score
FDR in bulk < 0.05 (#DE genes in bulk: 3322)					
batch_scran	10,606	2958	0.279	0.890	0.425
pseudo_bulk	1324	1042	0.787	0.314	0.449
CorrConf_scran	3079	1115	0.362	0.336	0.348
cate_scran	7056	2639	0.374	0.794	0.502
dSVA_scran	3130	1419	0.453	0.427	0.440
SVA_scran	10,344	2970	0.287	0.894	0.435
FDR in bulk < 0.1 (#DE genes in bulk: 4475)					
batch_scran	10,606	3899	0.368	0.871	0.517
pseudo_bulk	1324	1093	0.826	0.244	0.377
CorrConf_scran	3079	1361	0.442	0.304	0.360
cate_scran	7056	3299	0.468	0.737	0.572
dSVA_scran	3130	1711	0.547	0.382	0.450
SVA_scran	10,344	3928	0.380	0.878	0.530

Bulk RNA-seq detected substantially fewer DE genes (3322) when compared to scRNA-seq (10,606 DE genes detected), suggesting that scRNA-seq-based analysis is more sensitive for revealing DE genes, probably as a result of its capture of the variation information within each batch (which consists of thousands of values for each batch in scRNA-seq, as compared to a single value in bulk RNA-seq). Many potentially true DE genes revealed in scRNA-seq-based analysis failed to reach statistical significance in the bulk RNA-seq data analysis, analogous to the power loss of the aggregation method (pseudo_bulk) in the simulation results. Consequently, the precision with which DE genes were detected by single-cell based methods, based on comparisons to the DE genes derived from independent single-cell data, was much higher than the precision obtained when using RNA-seq data. This is consistent with the pattern shown in Table 6. When the FDR cutoff was relaxed to 0.1 for the bulk RNA-seq result, the recall of batch_scran and SVA_scran decreased by only approximately 2%. However, both the precision and the F1 score increased substantially (by ~10% and 0.09, respectively), which means that most of the genes with FDRs between 0.05 and 0.1 in the bulk results achieved FDRs of <0.05 with batch_scran and SVA_scran.

4. Discussion

We evaluated eleven methods which are either widely used or recently developed to account for the batch effects with various parameter configurations in scRNA-seq DE analysis. In general, For unobserved batch variables, when they can be approximated by analyzing the full gene-cell matrix (e.g., large sample size with small group effects), surrogate variable based approaches outperformed single gene based methods, such as aggregation methods and mixed effects models [9,24]. However, simulation results also indicated that the current surrogate variable based methods have not been properly designed/optimized for scRNA-seq data (e.g., CorrConf_k20_scran can show both inflated FDR and reduced power). Furthermore, when there are impurities in the group labels, as is expected in many real applications, methods such as CorrConf, cate, and dSVA might (inadvertently) extract the true underlying group information in the surrogate batch variables. This will substantially reduce the power of detecting biologically meaningful DE genes, which represents a major concern for these methods. Conversely, one of the surrogate viable methods, SVA, is apparently insensitive to this potential problem, probably because it first attempts to identify a list of genes that are unlikely to be affected by the group difference and assigns greater weight to them in later estimations. However, similar to other surrogate variable methods, SVA still exhibits slight FDR inflation (especially with large group effects). If this slight FDR inflation (e.g., up to 0.2) is tolerable, we recommend SVA for correcting either known or latent batches, (with "pseudo-cell" aggregation for large number of cells). Overall, there is no single method that can strictly control the FDR and achieve close to the optimal power of DE gene detection in all simulated scenarios. It is, therefore, necessary to develop new methods, especially ones tailored to the specific features of scRNA-seq data, such as the large sample size, abundance of zeros, and low count values.

We showed that scRNA-seq based imputation is not necessary and often results in suboptimal performance compared to methods that model the discrete counts using the negative binomial distribution. Imputation techniques might be useful for clustering/visualization because these methods, e.g., k-means clustering or Gaussian mixture models, assume data follows a continuous distribution, imputation might help in transforming the data towards a more continuous fashion especially in the log scale, which might benefit the methods for downstream visualization/clustering.

Based on our evaluation, the aggregation approach to form "pseudo-cells" from a small number of cells, e.g., 20, seems to be very useful both for reducing the computational speed as well as maintaining/improving the performance of several surrogate variable based methods. One typical example is SVA. It is likely that the distribution of the log scaled counts can be better modeled as Gaussian distributions after count aggregation, which are the primary assumption employed by all surrogate variable based methods.

Although we focused on DE analysis of two groups in the current evaluation, these methods can be applied for testing equal expressions among multiple groups or for testing other interesting contrasts within the generalized linear (mixed) model framework. For example, once the batch information is estimated, these estimated batch variables can be used as known covariates in the design matrix to adjust for the latent batch effects.

In our comparison, we did not request cells to be derived from a single cell type; therefore, the interpretation of the DE analysis depends on the comparison configuration. For example, a typical scRNA-seq analysis may include cell-type heterogeneity in both groups, which inevitably complicates the DE analysis because both the changes in cell-type proportion and the expression change within a specific subpopulation will generate the DE genes. To perform DE analysis in a specific cell type, we may first perform clustering analysis to identify distinct cell subpopulations by using a clustering method optimized for scRNA-seq data [34], followed by cell-type identification using known marker genes, and we perform DE analysis in the desired cell types while adjusting for the batch effects. We advise caution with respect to identifying the cell types properly so that they are biologically meaningful and comparable across different batches. When combining clustering with DE analysis, we must be cautious to avoid the "data snooping" or selection bias which results in false *P* values [35].

Finally, in the current study, we evaluated the batch correction in only UMI count based scRNA-seq data. Although we expect that read count based scRNA-seq data might show similar patterns (after accounting for zero inflation), additional evaluations are needed.

CRediT authorship contribution statement

Wenan Chen: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Silu Zhang:** Methodology, Software, Writing - original draft. **Justin Williams:** Investigation, Resources, Data curation. **Bensheng Ju:** Investigation, Resources, Data curation. **Bridget Shaner:** Investigation, Resources, Data curation. **John Easton:** Investigation, Resources,

Please cite this article as: W. Chen, S. Zhang, J. Williams et al., A comparison of methods accounting for batch effects in differential expression analysis of UMI count based single cell RNA sequencing, Computational and Structural Biotechnology Journal, https://doi.org/10.1016/j.csbj.2020.03.026

Data curation. **Gang Wu:** Writing - review & editing. **Xiang Chen:** Conceptualization, Methodology, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank Keith A. Laycock, PhD, ELS, for editing the manuscript.

Funding

National Cancer Institute of the National Institutes of Health [P30CA021765]; American Lebanese Syrian Associated Charities (ALSAC).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2020.03.026.

References

[1] Hwang B, Lee JH, Bang D. Single-cell RNA sequencing technologies and bioinformatics pipelines. Exp Mol Med 2018;50:96.
[2] Liu Serena, Trapnell Cole. Single-cell transcriptome sequencing: recent advances and remaining challenges. F1000Res 2016;5.
[3] Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 2015;161:1187–201.
[4] Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 2015;161:1202–14.
[5] Chen W, Li Y, Easton J, Finkelstein D, Wu G, Chen X. UMI-count modeling and differential expression analysis for single-cell RNA sequencing. Genome Biol 2018;19:70.
[6] Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, et al. Tackling the widespread and critical impact of batch effects in high-throughput data. Nat Rev Genet 2010;11:733–9.
[7] Tran HTN, Ang KS, Chevrier M, Zhang X, Lee NYS, Goh M, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biol 2020;21:12.
[8] Hicks SC, Townes FW, Teng M, Irizarry RA. Missing data and technical variability in single-cell RNA-sequencing experiments. Biostatistics 2018;19:562–78.
[9] Tung PY, Blischak JD, Hsiao CJ, Knowles DA, Burnett JE, Pritchard JK, et al. Batch effects and the effective design of single-cell gene expression studies. Sci Rep 2017;7:39921.
[10] Cole MB, Risso D, Wagner A, DeTomaso D, Ngai J, Purdom E, et al. Performance assessment and selection of normalization procedures for single-cell RNA-Seq. Cell Syst 2019;8(315–328):e318.
[11] Soneson C, Robinson MD. Bias, robustness and scalability in single-cell differential expression analysis. Nat Methods 2018;15:255–61.
[12] Finak G, McDavid A, Yajima M, Deng J, Gersuk V, Shalek AK, et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. Genome Biol 2015;16:278.
[13] Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics 2007;8:118–27.
[14] Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol 2018;36:421–7.
[15] Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert JP. A general and flexible method for signal extraction from single-cell RNA-seq data. Nat Commun 2018;9:284.
[16] Lin Y, Ghazanfar S, Wang KYX, Gagnon-Bartsch JA, Lo KK, Su X, et al. scMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets. Proc Natl Acad Sci USA 2019;116:9775–84.
[17] Leek JT, Storey JD. Capturing heterogeneity in gene expression studies by surrogate variable analysis. PLoS Genet 2007;3:1724–35.
[18] Leek JT, Storey JD. A general framework for multiple testing dependence. Proc Natl Acad Sci USA 2008;105:18718–23.
[19] Risso D, Ngai J, Speed TP, Dudoit S. Normalization of RNA-seq data using factor analysis of control genes or samples. Nat Biotechnol 2014;32:896–902.
[20] Lee S, Sun W, Wright FA, Zou F. An improved and explicit surrogate variable analysis procedure by coefficient adjustment. Biometrika 2017;104:303–16.
[21] McKennan C, Nicolae D. Accounting for unobserved covariates with varying degrees of estimability in high dimensional experimental data. arXiv:180100865, 2018..
[22] McKennan C, Nicolae D. Estimating and accounting for unobserved covariates in high dimensional correlated data. arXiv:180805895, 2018..
[23] Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;8:14049.
[24] Lun ATL, Marioni JC. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. Biostatistics 2017;18:451–64.
[25] Cossarizza A, Chang HD, Radbruch A, Akdis M, Andra I, Annunziato F, et al. Guidelines for the use of flow cytometry and cell sorting in immunological studies. Eur J Immunol 2017;47:1584–797.
[26] Cheng C, Easton J, Rosencrance C, Li Y, Ju B, Williams J, et al. Latent cellular analysis robustly reveals subtle diversity in large-scale single-cell RNA-seq data. Nucl Acids Res 2019;47:e143.
[27] Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. Genome Biol 2017;18:174.
[28] Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 2010;26:139–40.
[29] Lun AT, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol 2016;17:75.
[30] Wang JS, Zhao QY, Hastie T, Owen AB. Confounder adjustment in multiple hypothesis testing. Ann Stat 2017;45:1863–94.
[31] Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol 2010;11:R25.
[32] Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nat Commun 2018;9:997.
[33] McNeish D, Stapleton LM. Modeling clustered data with very few clusters. Multivariate Behav Res 2016;51:495–518.
[34] Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat Rev Genet 2019;20:273–82.
[35] Zhang JM, Kamath GM, Tse DN. Valid post-clustering differential analysis for single-cell RNA-Seq. Cell Syst 2019;9(383–392):e386.