Published online 18 September 2017

Nucleic Acids Research, 2017, Vol. 45, No. 22 ***e179***
doi: 10.1093/nar/gkx828

Linnorm: improved statistical analysis for single cell RNA-seq expression data

Shun H. Yip[1,2,3], Panwen Wang[2], Jean-Pierre A. Kocher[2], Pak Chung Sham[1,4,5] and Junwen Wang[2,6,*]

[1]Centre for Genomic Sciences, LKS Faculty of Medicine, The University of Hong Kong, Hong Kong SAR, China, [2]Department of Health Sciences Research and Center for Individualized Medicine, Mayo Clinic, Scottsdale, AZ 85259, USA, [3]School of Biomedical Sciences, LKS Faculty of Medicine, The University of Hong Kong, Hong Kong SAR, China, [4]Department of Psychiatry, LKS Faculty of Medicine, The University of Hong Kong, Hong Kong SAR, China, [5]State Key Laboratory in Cognitive and Brain Sciences, LKS Faculty of Medicine, The University of Hong Kong, Hong Kong SAR, China and [6]Department of Biomedical Informatics, Arizona State University, Scottsdale, AZ 85259, USA

Received July 17, 2017; Revised August 31, 2017; Editorial Decision September 05, 2017; Accepted September 06, 2017

ABSTRACT

Linnorm is a novel normalization and transformation method for the analysis of single cell RNA sequencing (scRNA-seq) data. Linnorm is developed to remove technical noises and simultaneously preserve biological variations in scRNA-seq data, such that existing statistical methods can be improved. Using real scRNA-seq data, we compared Linnorm with existing normalization methods, including NODES, SAMstrt, SCnorm, scran, DESeq and TMM. Linnorm shows advantages in speed, technical noise removal and preservation of cell heterogeneity, which can improve existing methods in the discovery of novel subtypes, pseudo-temporal ordering of cells, clustering analysis, etc. Linnorm also performs better than existing DEG analysis methods, including BASiCS, NODES, SAMstrt, Seurat and DESeq2, in false positive rate control and accuracy.

INTRODUCTION

RNA sequencing (RNA-seq) (1) is a popular method for quantifying expression levels. RNA-seq produces datasets in the form of reads of RNA fragments generated by high-throughput sequencing technologies. For humans, each RNA-seq sample can often consist of more than 20 million reads. Gene expression levels can be deduced from these datasets by counting the number of reads originating from each gene. Single cell RNA-seq (scRNA-seq) technology (2,3) applies RNA-seq on individual cells, allowing the transcriptome of a given cell to be investigated. Since the amount of RNA from a single cell is very limited, scRNA-seq relies heavily on amplification. Such heavy RNA amplification can cause random dropout events, which induce a large number of zero counts in the expression matrix. Therefore, more specialized normalization methods are needed for scRNA-seq data analysis.

Normalization is an important procedure in statistical analysis. Raw data contain both biological variations and technical noises. Ideally, normalization should eliminate all technical noises from the dataset, while ensuring that all biological variations are detected in the downstream analyses. Hence, an ideal normalization method would allow downstream analyses to achieve finely controlled false positive rates (FPRs) and false negative rates (FNRs), and simultaneously attain high accuracy. Therefore, existing software packages such as SAMstrt (4), scran (5), edgeR (6), DESeq2 (7), and limma (8) are paired with normalization methods such as TMM (9), DESeq-sc (10) and voom (11). Current scRNA-seq analysis methods, such as SAMstrt, scran and SCnorm (12), have used the normalization step similar to existing RNA-seq analysis methods. They perform normalization by utilizing the scaling factor, which is a multiplier for each cell's expression. Seurat (13) utilizes the conventional relative expression normalization, but it has a data imputation step to replace zeroes in the dataset. BASiCS (14) utilizes a post hoc correction strategy instead of normalization. While most existing methods are parametric, NODES (BioRxiv: https://doi.org/10.1101/049734) has a non-parametric approach. It converts the expression data into psedo-counts, with goals to eliminate variance across the dataset, such that the stable genes would show zero variance. This removes heterogeneity in the dataset and decreases false positive rates in downstream DEG analysis.

In addition to normalization, data transformation is also needed in statistical analysis. Some downstream analysis methods, such as limma and principal component analysis, are based on the linear model or assume normality

*To whom correspondence should be addressed. Tel: +1 480 301 4644; Fax: +1 480 301 8387; Email: wang.junwen@mayo.edu

in the dataset. Other methods, such as scran and Seurat, utilize the logarithmic transformation, which is often assumed to transform a count dataset toward homoscedasticity and normality. However, this assumption may not be well satisfied in most cases. To solve this issue, RNA-seq analysis methods were developed to model or transform the datasets, such that these assumptions can be fulfilled. For example, voom employs precision weights to model the mean–variance relationship for limma, and the DESeq2 package includes a variance-stabilizing transformation method (DESeq-vst) that transforms bulk RNA-seq data toward homoscedasticity. However, no dedicated data transformation strategy has been developed for scRNA-seq data.

We present Linnorm, a linear model and normality based normalizing transformation method for more precise statistical analysis of scRNA-seq data. By utilizing a critically selected list of homogenously expressed genes as reference, Linnorm calculates a set of parameters for normalization. Linnorm's normalization algorithm is similar to the scaling factor (9,10). With the scaling factor, normalization is done with raw expression data, where a factor can be interpreted as a linear model that goes through the origin, which is the point (0, 0). In comparison, Linnorm performs a prior logarithmic transformation on the expression data, and the dataset is fitted to a linear model that does not need to go through the origin. This allows expression levels to be adjusted both linearly and exponentially. The additional exponential scaling in Linnorm allows a better fit of each cell's expression to the expression mean than the scaling factor, which provides a stronger noise removal effect. Lastly, Linnorm's transformation method focuses on employing the linear model on mean, SD and skewness, which thrives to ensure homoscedasticity and normality in the homogenously expressed genes.

We compared Linnorm with existing scRNA-seq normalization and transformation methods, including NODES(BioRxiv: https://doi.org/10.1101/049734), SCnorm (12), scran (5), Seurat (13) and DESeq-sc (10). We further tested the effects of Linnorm's normalizing transformation on the quality of DEG analysis; and compared it with existing methods that provide *P* value output, including BASiCS (14), SAMstrt (4), NODES and Seurat. Additionally, we compared Linnorm with two popular RNA-seq analysis methods, DESeq2 and TMM. Our results show that Linnorm has key advantages in both technical noise removal and preservation of cell-to-cell differences. This improves performances in multiple distinct statistical analyses in our study. Particularly, Linnorm's good preservation of cell heterogeneity suggests that it can improve common scRNA-seq analyses such as the discovery of novel cell subpopulation, pseudo-temporal ordering of cells (15), clustering analysis, etc.

MATERIALS AND METHODS

Linnorm algorithm

Overview. Linnorm assumes that a set of genes are homogeneously (stably) expressed across different cells/samples. The Linnorm algorithm calculates normalization and transformation parameters by utilizing these stably expressed genes. After normalization and transformation, these genes will have: (i) stable expression values and (ii) approach homoscedasticity and normality. Hence, Linnorm's first step is to identify the stable genes by filtering. After calculating the parameters with the stable genes, the parameters are applied to the entire dataset. All zeroes in each gene are ignored in the following steps.

Initial normalization and transformation. To accurately identify a set of stable genes, an initial normalization and transformation step is applied. In this step, Linnorm utilizes conventional methods to prepare the dataset for modeling.

First, we normalize the dataset by converting it into a relative scale. Let E_{ij} be the expression level of the feature (gene, etc.) i and library (sample) j; let m be the total number of features and n be the total number of samples. We convert each sample into the relative scale by:

$$R_{ij} = \frac{E_{ij}}{\sum_{i=1}^{m} E_{ij}} \; (1 \leq i \leq m, 1 \leq j \leq n) \qquad (1)$$

Filtering. While RNA-seq data have biological replicates, scRNA-seq dataset do not. Hence, we filter the dataset to ensure that the genes being used for modeling are largely homogeneous. Linnorm's filtering algorithm has two steps: (i) low count genes with high amount of zero are filtered and (ii) highly variable genes with high amount of technical noise are filtered based on standard deviation (SD) and skewness. We transform the dataset with logarithm in this step.

Filtering low count genes with high technical noise. Genes with high amount of zero counts are filtered using the minimum non-zero cell portion (MZP) threshold of z, and $0 < z \leq 1$, where only genes with at least z portion of the cells being non-zero would be retained. By default, z is set to 0.75, which ensures at least three non-zero cells even when the sample size is down to 3 or 4, because skewness requires at least three non-zero values to be calculated.

In a dataset that contain very low level of technical noise, the mean versus SD plot will show a negative slope (7,11). However, since the lowest expressing genes would contain multiple raw counts of near 1, they would also have low SD. Therefore, these genes will connect the origin (0, 0) with the highest point in the negatively sloped mean versus SD plot, which forms a hill-shaped plot (Figure 1). To identify the threshold for filtering low count genes, Linnorm gradually increases the filtering threshold until one third of the lowest expressing genes show a negative slope.

Filtering highly variable genes based on SD and Skewness. Linnorm then filters highly variable genes based on their SD and skewness. To identify highly variable genes, a locally weighted scatterplot smoothing (LOWESS) curve is fitted onto the mean versus SD plot. The log-fold change of each gene's SD from its expected value is calculated from the LOWESS results. Since the magnitude of fold change of each gene's SD may not be distributed evenly across different means, the log-fold change of each gene's SD is scaled based on the residuals from a linear regression line between

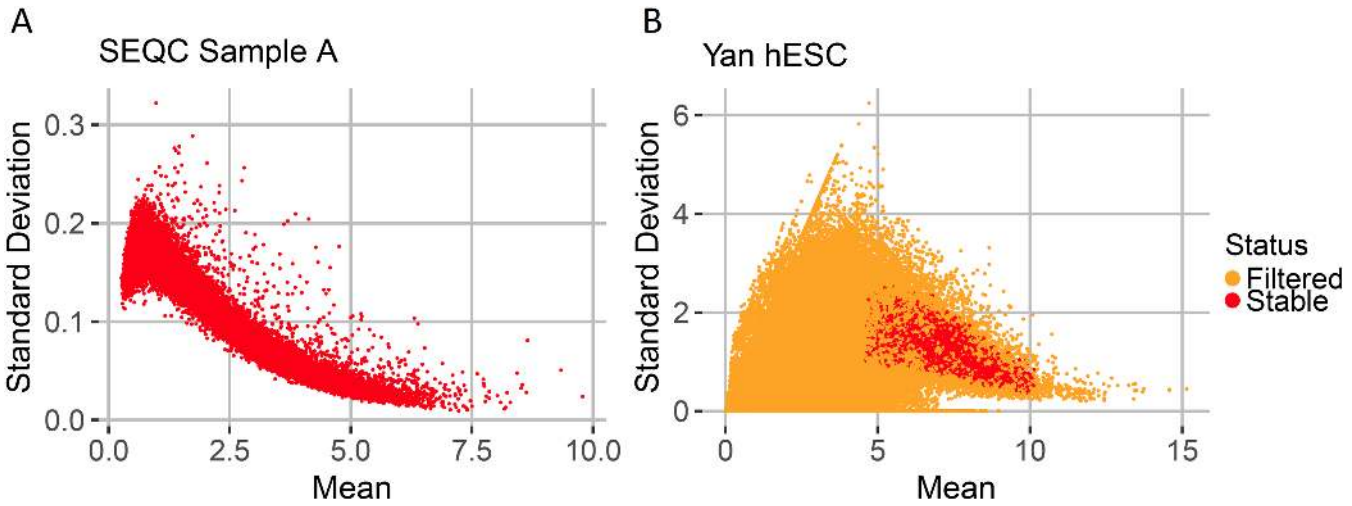

Figure 1. Linnorm's selection of homogeneously expressed genes from scRNA-seq dataset. Mean versus SD plots after log-plus-one CPM transformation, where zeroes are ignored. (**A**) Replicates of the SEQC dataset's Sample A, where genes with any zero counts are filtered. It is a model of a low noise dataset. (**B**) Yan dataset human embryonic stem cell data, where all genes are retained and Linnorm's selected stable genes are highlighted in red. Figure B shows that Linnorm's filtering procedure is capable of selecting a set of low noise stable genes that show a negative slope similar to the low noise model in A.

the mean and the log-fold changes. This scaling method is similar to Seurat. Seurat scaled SD by assigning each features into bins based on their mean, whereas Linnorm utilized linear regression. Compared to Seurat, Linnorm's scaling method is continuous across the mean. After removing outliers using the *boxplot* function in R, genes with significant SD are identified from the population of log-fold changes by using a two-sided Student's *t*-test.

Next, Linnorm identifies genes with significant skewness in the dataset. A LOWESS curve is fitted between mean and skewness. The residual skewness of each gene is calculated. After removing outlying residuals using the *boxplot* function in R, genes with significant skewness are identified from the residuals by using a two-sided Student's *t*-test.

Finally, *P* values of the SD and skewness are combined by using the Empirical Brown's method (16). Genes with significantly low *P* values are filtered. If spike-in genes are provided, the mean and SD used in the above Student's *t*-tests will be calculated using the spike-in genes instead of the whole population.

Calculating the data transformation parameter, $\boldsymbol{\lambda}$. Linnorm transforms the dataset using a modified log-plus-one transformation. Let T_{ij} be the transformed dataset and λ be a transformation parameter.

$$T_{ij} = \ln(\lambda R_{ij} + 1) \tag{2}$$

To find λ that best transforms the dataset, we consider the homoscedasticity and normality. To measure the deviation of T_{ij} from the homoscedasticity and normality assumptions, we calculate the deviation coefficient $F(\lambda)$

$$\begin{aligned} F(\lambda) &= V(\lambda)^2 + S(\lambda)^2 \\ \lambda &= \text{argmin}(\text{V}(\lambda)^2 + \text{S}(\lambda)^2) \end{aligned} \tag{3}$$

Here, we use $V(\lambda)$ to represent the homoscedasticity and $S(\lambda)$ to represent the skewness of the dataset. $V(\lambda)$ and $S(\lambda)$ are combined using the Euclidian distance, where the square root is omitted because it is a monotonic function.

Calculation of $V(\boldsymbol{\lambda})$. $V(\lambda)$ represents the deviation of T_{ij} from homoscedasticity. Let M_i be the mean of each feature (or gene expression) in T_{ij} and D_i be the SD of each feature in T_{ij}. Homoscedasticity suggests that D_i and M_i should have a relationship for all M_i where c is a constant, denoted by the formula:

$$D_i^{optimal} = c \tag{4}$$

Given λ and R_{ij}, we find the relationship between D_i and M_i in T_{ij} by linear regression and obtain the formula:

$$D_i^{expected} = a_d M_i + b_d \tag{5}$$

From Equation (5), we can see that Equation (4) is satisfied when $a_d = 0$. Since $V(\lambda)$ needs to be combined with $S(\lambda)$ in a later step, we normalize the data to a logarithmic form and add 1 to ensure they are in a similar scale.

$$V(\lambda) = \log(|a_d| + 1) + 1 \tag{6}$$

Calculation of $S(\boldsymbol{\lambda})$. A normally distributed dataset has zero skewness, thus $S(\lambda)$ represents the deviation of T_{ij} from zero skewness. Let s_i be the Pearson's moment coefficient of skewness of the feature i in T_{ij}. Then, optimally, s_i and M_i should have a relationship denoted by the equation:

$$s_i^{optimal} = 0 \tag{7}$$

To find the relationship between mean M_i and skewness s_i given λ in T_{ij}, we perform a linear regression and obtain the formula:

$$s_i^{expected} = a_s M + b_s \tag{8}$$

Next, we measure the deviation of Equation (8) from Equation (7) by performing an integral on the absolute value of Equation (8) to obtain:

$$S^{raw}(\lambda) = \int_{M_1}^{M_m} |a_s M + b_s| dM \Big/ (M_m - M_1), \tag{9}$$

where M_m and M_l are the maximum and minimum mean, respectively. Similar to $V(\lambda)$, we normalize it for combination in $F(\lambda)$, such that $S(\lambda) = \log(S^{raw}(\lambda) + 1) + 1$

***Optimization of* λ.** We find the optimized normalization parameter $\lambda^{optimized}$ for which $F(\lambda)$ from Equation (3) is minimized.

To define an initial range of λ, we make an observation on a_d from Equation (6). Let's assume that the input data is in the raw count format. Let u be the maximum total count of the libraries in E_{ij}. We know that

$$a_d \begin{cases} > 0, \ if \lambda = 1 \\ < 0, \ if \lambda = u \end{cases} \tag{10}$$

from previous studies (7,11). Since $a_d^{optimal} = 0$, we obtain Equation (11) by combining it with Equation (10).

$$1 < \lambda < u \tag{11}$$

If the input E_{ij} is in raw count units, then the upper bound of λ should be the total count of the largest sample. Since some machine learning-based expression quantification software tools do not output raw counts, we estimate λ by letting $m^{\min}$ be the mean of the non-zero values of 5% of the genes with the smallest expression mean, and set

$$u = \frac{1}{m^{\min}} \tag{12}$$

In case there is more than one local minimum, Linnorm searches for an optimal λ in this range using an iterated local search algorithm, rather than using an expectation maximization algorithm. Nevertheless, the user-inputted dataset may have been filtered or modified, and we cannot guarantee that u must be larger than the original maximum total count. In this case, Linnorm will enlarge u and continue searching for a global minimum of λ, if it finds $\lambda \approx u$. Finally, we show that a global minimum exists in the Supplementary methods section and Supplementary Figure S1.

Normalization. First, $G_{ij} = \ln(\lambda R_{ij})$ is obtained, which contains all genes from the dataset. Then, it is filtered with the above method for the following steps. Each gene's mean expression value across all cell is calculated. The expression mean and each sample's expression are fitted to linear models, such that n models with the equation $y_i = m_j x_{ij} + c_j$ are calculated, where y represents the expression means and x represents a sample's expression values. Each of the model is shifted toward the identity line based on the normalization strength coefficient, μ, where $0 \leq \mu \leq 1$. m and c in the models can be updated with the equations $m^{updated} = \mu(m - 1) + 1$ and $c^{updated} = c \times \mu$. When μ is zero, Linnorm's normalization would be equivalent to the conventional relative expression normalization. Alternatively, the $\mu = 1$ would signal the software to maximize its technical noise removal effects. By default, μ is set to the middle point of 0.5, which provides a moderate level of normalization strength for a general dataset. However, we encourage users to optimize this parameter using Linnorm's clustering visualization functions, because each dataset contains a different amount of noise that needs to be removed accurately. Next, the parameters m and c will be applied to all genes in the dataset; and we obtain $B_{ij} = \exp(m_j^{updated} G_{ij} + c_j^{updated})$, which completes the normalization step and the logarithmic transformation is reversed. Finally, to complete Linnorm's normalizing transformation, we obtain $T_{ij} = \ln(B_{ij} + 1)$.

In the first step, $G_{ij} = \ln(\lambda R_{ij})$, λ from the transformation section is required. In the case where only data normalization, but not transformation, is needed. The calculation of λ from the transformation step would be skipped. If the user requests CPM output, λ would be replaced by a million. If output is requested to be in the estimated raw count unit, λ would be replaced by the median of total counts.

Even though Linnorm calculates the normalization and transformation parameters with the stable genes only, these parameters are applied to all genes from the original input. Therefore, by default, Linnorm's output expression matrix has the same size as the input expression matrix.

Datasets

scRNA-seq data. The five scRNA-seq datasets used in this study have distinct characteristics. They are summarized in Supplementary Table S1.

Yan dataset. This is a human preimplantation embryo and embryonic stem cell dataset. The average total read count in the expression matrix is 25,228,939 reads. Cell types with more than 10 samples, labeled 4-cell, 8-cell, Morulae, Late blastocyst and hESC were downloaded from Gene Expression Omnibus (GEO) (http://www.ncbi.nlm.nih.gov/geo) under accession no. GSE36552 (17).

Deng dataset. This is a mouse preimplantation embryo and embryonic stem cell dataset. The average total read count in the expression matrix is 15,540,102 reads. Cell types with more than 10 samples, labeled 16cell, 4cell, 8cell, C57twocell, early2cell, earlyblast, late2cell, lateblast, mid2cell and midblast were downloaded from GEO under accession no. GSE45719 (18).

Islam dataset. This dataset consists of 92 samples of scRNA-seq raw count data, 48 of which are embryonic stem cells and 44 are embryonic fibroblasts. The negative controls were not used in our study. The average total read count in the expression matrix is 578,467.6 reads. Raw count expression matrix was downloaded from GEO under accession no. GSE29087 (2).

Patel dataset. This is a glioblastoma dataset from tumors from five individual patients with IDs MGH26, MGH28, MGH29, MGH30 and MGH31. The average total read count in the expression matrix is 1,137,872 reads. Samples labeled MGH26, MGH26-2, MGH28, MGH29, MGH30 and MGH31 were downloaded from GEO under accession no. GSE57872 (19).

Klein dataset. This dataset was generated by the droplet barcoding method with an average total read count of 20,033.40 reads in the expression matrix. Four sets of single cell RNA-seq data, a 0 day mouse embryonic stem (ES) cell sample (File: GSM1599494_ES_d0_main.csv with 933

samples) and three samples following LIF withdrawal for 2, 4 and 7 days (Files: GSM1599497_ES_d2_LIFminus.csv with 303 samples, GSM1599498_ES_d4_LIFminus.csv with 683 samples, GSM1599499_ES_d7_LIFminus.csv with 798 samples), were downloaded from GEO under accession no. GSE65525 (20).

RNA-seq data.

SEQC dataset. The SEQC dataset is a set of RNA-seq raw count data paired with a Taqman dataset, which serves as the gold standard for the gene expression measurements. Its RNA-seq data contains Samples A, B, C and D. The Taqman data contain a subset of 955 genes and 4 replicates for each sample. This dataset was obtained through the bioconductor seqc package (21).

Expression quantification for scRNA-seq data

Raw reads of Yan, Deng and Patel datasets were downloaded from GEO. They were first trimmed using Trimmomatics (22) with default parameters, except Patel dataset where *MINLEN* was set to 20 because of its shorter read lengths. They were quantified by Kallisto (23) using Ensembl (24) hg38 or mm10 assembly using default parameters, except Patel dataset, where the kmer length was set to 19 because of its shorter reads. Estimated raw counts and TPM of Yan, Deng and Patel datasets were obtained from Kallisto.

Hardware and software used in this study

All of the analyses in this paper were run on a computing workstation with 32 Intel(R) Xeon(R) CPU E5–2650 v2 processors and 128 GB of RAM.

Supplementary Table S2 provides the versions of each software tool used in this study.

Methods used in this study

To investigate the $\log_2$FC of housekeeping genes, the *Linnorm.Norm* function from the *Linnorm* package was used, and the transformation step is skipped. For the scran software, we cancelled its default log plus one transformation. All tools were run with default parameters.

For t-SNE K-means clustering, Linnorm, scran, SCnorm, TMM, DESeq-sc and DESeq-vst were used with default parameters. The DESeq-sc parameters are described in Brennecke *et al.* (10); we call it DESeq-sc because this method from the DESeq package was utilized for single cell datasets in Brennecke *et al.* For DESeq-vst, we used the *varianceStabilizingTransformation* function from the *DESeq2* package. Seurat was run with *min.genes* parameter set to 1 to ensure that it will not filter any cells (for fair comparison with the other methods). Similarly, for fair comparison, NODES's *pQ* function is run with *frac=Inf* and *throw_sd=0* to prevent it from removing cells and genes. NODES, SCnorm, TMM and DESeq-sc were transformed with log plus one.

For the DEG analyses, Linnorm, SAMstrt, edgeR, DESeq2 and voom were run with default parameters. Both voom and Linnorm utilized limma. For fair comparison, Seurat was run with *min.genes* argument set to 1 in the *Setup* function and *thresh.use* set to 0 in the *FindMarkers* function, such that all genes were assigned *P* values for the evaluation and no samples were removed. Similarly, for fair comparison, NODES is run with *frac=Inf* in its normalization function, *pQ*, to prevent it from removing samples. BASiCS was run with the *BASiCS_MCMC* function's *N*, *Thin* and *Burn* parameters set to 1000, 10 and 500, respectively.

TMM, edgeR, DESeq2, DESeq-sc, DESeq-vst, SAMstrt, BASiCS, scran, SCnorm and voom were run with raw counts. Linnorm, NODES and Seurat were run with raw count with the Islam and Klein datasets, but TPM otherwise.

FPR/FNR in response to zero counts

We tested each method's FPR and FNR under the null with DEG analyses. We performed this test using Islam's embryonic stem cell dataset, Yan's hESC dataset, Deng's midblast dataset and Patel's MGH26 dataset. DEG analyses were performed by randomly choosing sample sets from the pool of cells with multiple repetitions. Because the cells were selected randomly, no true DEG would be expected. Therefore, by definition, *P* value should equal to the proportion of the genes in the dataset in the following tests. Lastly, we repeated the previous steps multiple times to ensure reliability of our results. The sample set size of 10 was tested 100 times for each dataset.

Evaluation of DEG analysis methods with SEQC datasets

In this test, we performed DEG analyses on Sample A versus B, A versus C, A versus D, B versus C, B versus D and C versus D. If a gene's Taqman expression's log2FC was higher than 1 or lower than –1, then it was considered to be a true DEG in the ROC curve. Other $\log_2$FC are also tested. A sample set size of 10 was tested 20 times by randomly choosing sample sets without replacement, for a total of 120 DEG tests.

RESULTS

Linnorm selects stably expressed genes to substitute for spike-in genes

Existing scRNA-seq normalization tools often rely on spike-in genes. However, their qualities are difficult to control (25,26). Therefore, Linnorm utilizes a novel method to select homogeneously expressed genes for modeling. Figure 1A is a normal RNA-seq dataset where all samples are replicates and genes that contain any zero counts are filtered. Since RNA-seq dataset have a higher accuracy than scRNA-seq data (27) and they have true replicates as opposed to samples of unique cells, we use this dataset as a model of a low noise dataset. Figure 1B highlights the stable genes selected by Linnorm in the Yan dataset. In Figure 1B, there is a group of lowly expressing genes that connect the origin to the highest point in the plot, forming a hill shaped plot. It is because genes with multiple cells with the raw count of near 1 would show lower SDs. These lowly expressed genes are promptly filtered by Linnorm; and the

genes that are retained in the dataset (genes in red color) show a negative slope that is similar to Figure 1A and previous studies (Figure 1B) (7,11). As shown in the Figure, Linnorm's algorithm is capable of selecting stable genes that show a similar negative slope in the mean versus SD plot as the low noise model.

Linnorm transforms the stably expressed genes toward homoscedasticity and normality

Homoscedasticity and normality are often assumed upon logarithmic transformation of a count dataset. Existing methods, such as scran and Seurat, utilize the log-plus-one transformation conventionally to transform scRNA-seq data. Nevertheless, homoscedasticity and normality can be better achieved with Linnorm. Figure 2 compares the homoscedasticity and normality assumptions between the log-plus-one CPM transformation and Linnorm's transformation, using the Klein dataset. Log-plus-one CPM in Figure 2A shows apparent heteroscedasticity with the stable genes and a negative slope similar to Figure 1. While this slope is naturally occurring (7,11) and is utilized by Linnorm to filter low count genes, it violates the homoscedasticity assumption. Hence, similar to DESeq-vst, the goal of Linnorm's transformation step is to minimize this slope. After Linnorm transformation, homoscedasticity is better achieved, where the stable genes have attained homogeneous SDs across the mean (Figure 2B).

Next, the normality assumption is investigated by using skewness (Figures 2C and D). Figure 2C shows that all stable genes are negatively skewed with the log-plus-one CPM transformation. Linnorm's improvement in normality over log-plus-one CPM is observed in Figure 2D, where the stable genes are now relatively closer to the x-axis. In Figures 2C and D, lowly expressing genes are also shown to have inflated skewness. This is because they often contain multiple counts of near 1, which can skew the distribution. Hence, Linnorm's algorithm is only focused on ensuring normality and homoscedasticity in the sufficiently and stably expressed genes.

Linnorm accurately removes technical variations from housekeeping genes

Housekeeping genes are known stable genes. However, variations can be observed in housekeeping genes for technical reasons. Previous studies noted that the upregulation of the MYC gene can increase cell size and the total amount of RNA, which can cause a global shift of expression levels when relative expression estimates are utilized (25,26,28,29). We examine this effect by utilizing the Yan dataset's human embryonic stem cell (hESC) samples. Figure 3A shows the expression levels of the MYC gene across the 32 hESCs, which shows hundreds of fold of expression differences between the lowest and the highest expressing cells. To examine the performance of the normalization methods, we obtain hESC samples with the lowest and highest MYC expression levels, respectively. We examined the $\log_2$ fold change ($\log_2$FC) of the housekeeping genes (30) between the two groups after normalization. Since housekeeping genes are known stable genes, their $\log_2$FC should be close to zero. Figure 3B shows the absolute $\log_2$FC of the 258 housekeeping genes after normalization, where each dot is the $\log_2$FC of a gene. Linnorm, NODES, SCnorm, scran, DESeq-sc and TMM's mean absolute log2FC across the sample set sizes were 0.60, 0.36, 0.80, 0.82, 0.93 and 0.88, respectively. Linnorm's lower average $\log_2$FC than most of the other methods (except NODES) suggests that it has normalized a larger amount of variations from the housekeeping genes.

Linnorm preserves cell heterogeneity

The previous section revealed that Linnorm would eliminate a larger amount of technical variations in the housekeeping genes than most of the other existing methods, which raises a concern of whether Linnorm could preserve cell heterogeneity in the dataset. Next, to investigate whether Linnorm's normalizing transformation preserves cell-to-cell differences in the data, we simulated hidden cell subpopulation analyses with the five scRNA-seq datasets. We performed t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction with *K*-means clustering, where cell type information were blind to each normalization method. Then, we investigated clustering purity with the known cell type information (Supplementary Figure S2), where the clustering purity of 1 indicates that all cells are correctly clustered. In this section, Linnorm's normalization and transformation is tested against the other normalization and transformation methods, including NODES, SCnorm, scran, Seurat, TMM, DESeq-sc and DESeq-vst.

Figure 4 shows Linnorm's clustering purities plotted against the clustering purities of the other methods in five independent datasets. Linnorm outperforms other methods in terms of clustering purity, with most of its data points located on the upper-left side of the black diagonal line. Linnorm's average purity, with or without normalization, is also higher than the other methods (Supplementary Figure S3 and Supplementary Table S3). One-sided Wilcoxon signed rank test is utilized to test whether Linnorm's clustering purities are greater than the other methods. Against NODES, SCnorm, scran, Seurat, TMM, DESeq-sc and DESeq-vst, the *P* values are 4e-04, 4.7e-05, 4.5e-05, 1.2e-02, 2.8e-03, 2.7e-04 and 4.4e-04, respectively. Hence, Linnorm's clustering purity is significantly higher than all other methods. Linnorm shows good preservation of cell heterogeneity in scRNA-seq data, which implies that it can improve other similar analyses, such as the pseudo-temporal ordering of cells.

Computational speed

Computational speed is an important consideration in the development and implementation of Linnorm's algorithm, because scRNA-seq data can have more than hundreds of samples. Figure 5 summaries the computational time of different methods utilizing the Yan dataset. The speed of normalization, transformation and DEG analysis methods are tested separately. To be classified as a dedicated normalization or a dedicated transformation method, a method must include some algorithm that is more complicated than a

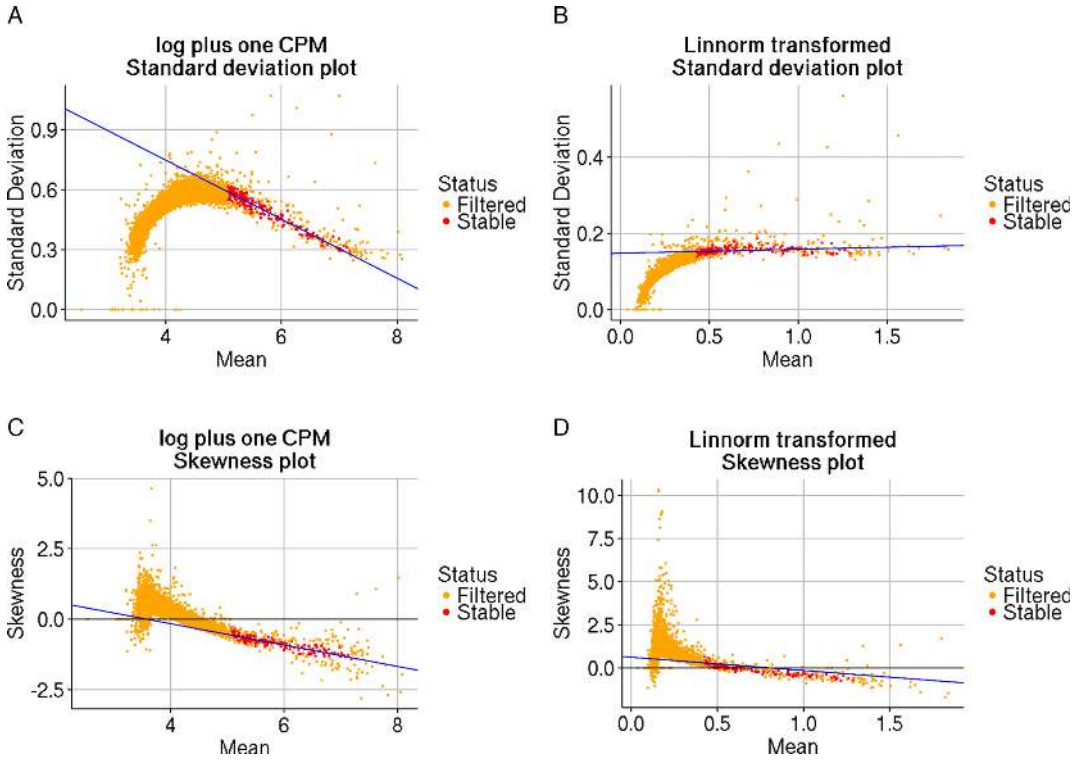

Figure 2. Comparing the homoscedasticity and normality assumptions between the conventional log-plus-one CPM transformation and Linnorm's transformation. The Klein dataset is utilized. (**A**, **B**) Mean versus SD plots for the investigation of homoscedasticity, where SD should be stable across the mean. (**C**, **D**) Mean versus skewness plots for the examination of normality, where the normal distribution has the skewness of 0. All zeroes are ignored in this figure.

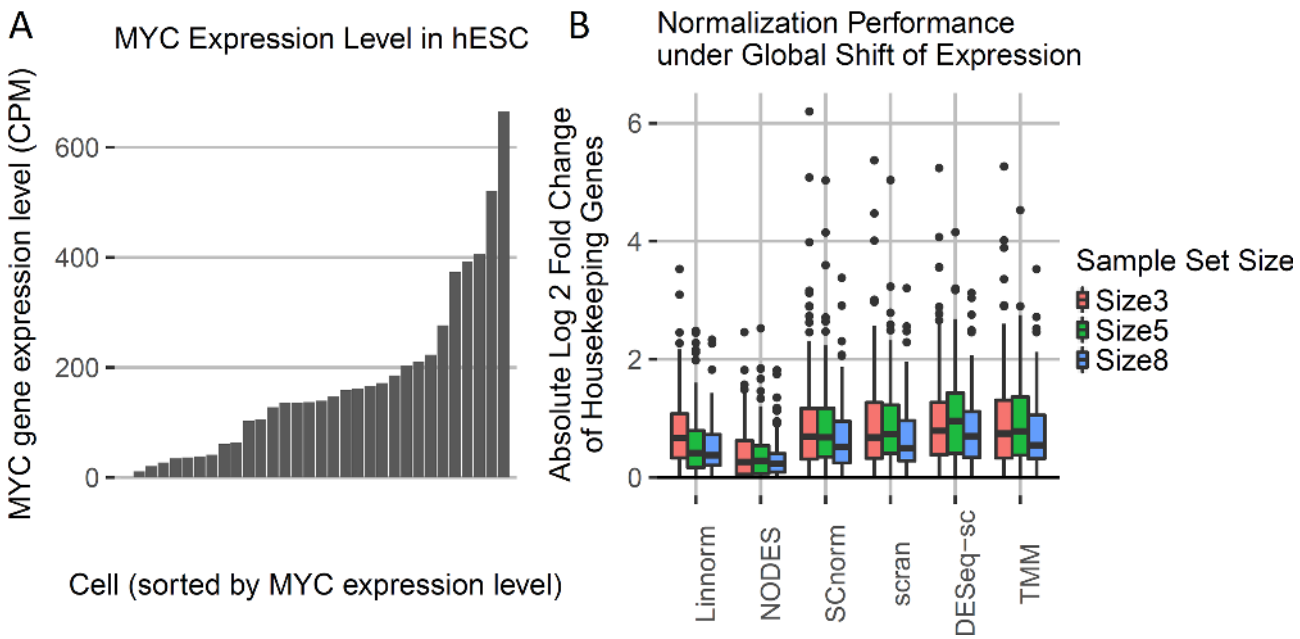

Figure 3. Investigating technical variation removal with $\log_2$FC of housekeeping genes. Technical variations are induced by differentiating MYC expression levels, where it is known to induce a global shift of expression. (**A**) CPM expression level of the MYC gene in the Yan dataset's hESC. (**B**) Average $\log_2$FC of the housekeeping genes between the samples with the lowest MYC expressions and the samples with the highest MYC expressions. Sample set size of three indicates a three samples versus three samples comparison. There is a total of 258 housekeeping genes. A well performing method should have housekeeping genes' $\log_2$FC closer to zero.

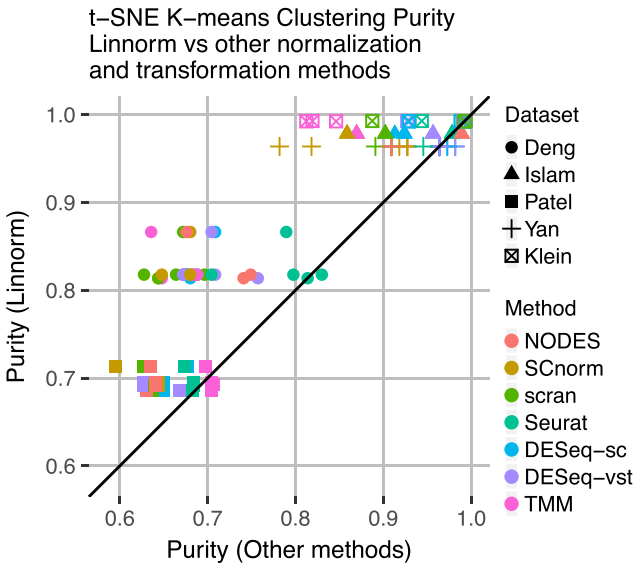

Figure 4. Simulated hidden cell subpopulation analysis using t-SNE *K*-means clustering. Plot of Linnorm's clustering purity versus the other methods' purity. Purity of 1 indicates that all cells with the same cell type are correctly clustered together. A point on the left side of the black identity line would indicate Linnorm's improved performance over one of the existing methods in one of the analyses. The other methods and the five datasets are highlighted in the legend for references. Each dataset was tested with the number of principal components from 2 to 6, giving a total of 25 tests.

CPM/TPM normalization or a simple log or log-plus-one transformation.

For normalization at the sample size of 110, Linnorm took 1.98 s on average and was 6.5, 97, 2.0, 1.1 and 75 times faster than NODES, SCnorm, scran, DESeq-sc and TMM, respectively. For data transformation at the sample size of 110, Linnorm took 7.7 s on average and was 3.2 and 65 times faster than Seurat and DESeq-vst, respectively. For DEG analysis at the sample set size of 55, Linnorm took 11.6 s and was 129, 1.4, 46, 217, 242, 39 and 1.1 times faster than BASiCS, NODES, SAMstrt, Seurat, DESeq2, edgeR and voom, respectively.

Linnorm is more than a hundred percent faster than the tested methods at a larger sample size, because Linnorm is implemented with a time complexity of $O(n \log n)$; and its extensive usage of the linear model, mean, SD and skewness can be simultaneously computed in one pass of the expression matrix under a single loop in C++.

Linnorm controls FPR/FNR well in response to zero counts

In the absence of true differential expression, *P* value equals to the proportion of genes in the dataset. In other words, 5% of the genes would have *P* values of less than 0.05 under the null. This concept is often used to examine FPR in DEG analysis (11,31). (BioRxiv: https://doi.org/10.1101/018739) To utilize this concept to examine each method's FPR across varying amounts of non-zero cells, we utilize the ratio of the number of genes with *P* value less than 0.05 and the number of expected genes given the proportion. We took the logarithm of this ratio, abbreviated as Log Significant to Expected Ratio (LogSTER). By definition, a LogSTER close to zero indicates fine control of FPR and FNR. Since scRNA-seq data contain an abundance of zeroes, we inspect each method's FPR and FNR in response to zero counts. We plot LogSTER against the minimum non-zero cell portion (MZP) threshold in Figure 6. The MZP threshold of 0.2 means that genes without at least 20% of the cells being non-zero would be filtered.

Linnorm shows LogSTERs that are close to zero above the MZP threshold of 0.2, indicating fine control of both FPR and FNR (Figure 6). At the MZP threshold of 0.5, Linnorm, BASiCS, NODES, SAMstrt and Seurat have LogSTERs of 0.01, 2.47, –4.76, 1.17 and 2.36, respectively. NODES has a negative LogSTER, which indicates high FNR and confirms the concern from Figure 3. On the other hand, the other methods show high LogSTER values, which indicate high FPR. Interestingly, RNA-seq DEG analysis methods show better control of FPR and FNR with scRNA-seq data than the existing scRNA-seq methods (Supplementary Figures S4 and S5).

Below the MZP threshold of 0.2, Linnorm calls less genes as significant and attain negative LogSTER values. In this analysis, since each of the two conditions in the DEG analyses contain 10 cells, MZP threshold of 0.2 indicates an average of 2 non-zero values in each condition. Linnorm's lower LogSTER below the MZP threshold of 0.2 is reasonable because SD requires at least two numbers to be calculated, and a good method should not declare genes as significant when there is an insufficient amount of evidence. In Supplementary Figures S4 and S5, we also show that Linnorm can control FPR and FNR well across other datasets and smaller sample sizes.

Linnorm improves accuracy in DEG analysis

Although a method that has finely controlled FPR and FNR would robustly call the correct number of significant genes, it would not necessarily mean that the genes being called were accurate. While the balance of FPR and FNR depend on the amount of noise being normalized from the dataset, accuracy depends on whether technical noises, instead of biological variations, are being accurately normalized. To investigate Linnorm's accuracy, its receiver operating characteristic (ROC) curve performance was examined using the SEQC dataset from the MAQC-III project (21). This RNA-seq dataset is utilized because it is complemented by a gold standard Taqman dataset for the calculation of accuracy. Since RNA-seq datasets are similar to scRNA-seq dataset, but with higher accuracy and less zero counts (27), and scRNA-seq datasets also have genes that contain no zero-counts, a good scRNA-seq method should be compatible with RNA-seq data.

In Figure 7, we define a gene as differentially expressed when its log 2 fold change (log2FC) is larger than 1 or less than –1 as measured by TaqMan qPCR. Linnorm's area under the ROC curve (AUC) above chance level was 131%, 60%, 52%, 7.0% and 13% higher than BASiCS, NODES, SAMstrt, Seurat and DESeq2, respectively. Improvements are also observed across different log2FC thresholds (Figure 7B). The same test is applied to edgeR and voom in the Supplementary Figure S6. edgeR, DESeq2 and voom's performances are in concordance with a previous study (32).

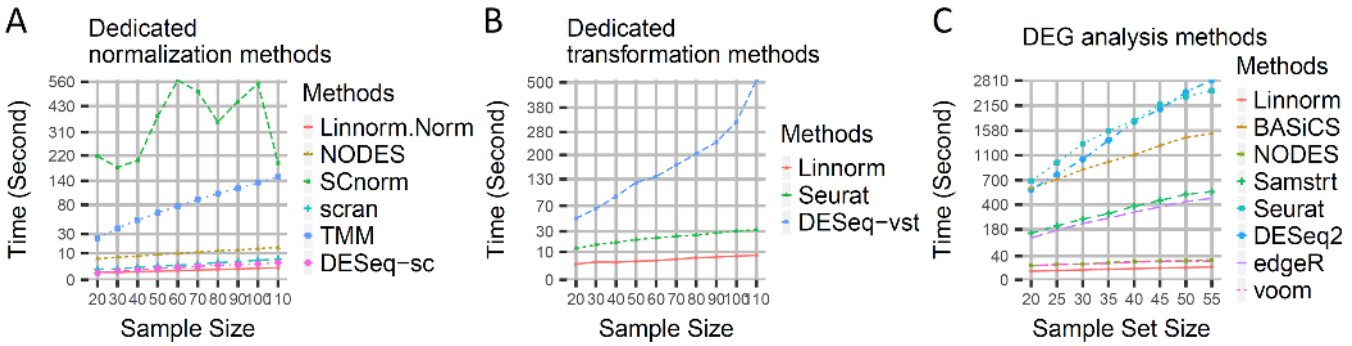

Figure 5. Computational speed across sample sizes with the Yan dataset. Linnorm is colored red. (**A**) Normalization methods. While all other methods are single threaded, SCnorm's algorithm would utilize all available threads in the computing cluster. Hence, its speed would deviate according to the amount of available resources in the system. (**B**) Transformation methods. While Seurat is not a dedicated transformation method, we place it here because its data imputation algorithm outputs data in the transformed format. (**C**) DEG analysis methods.

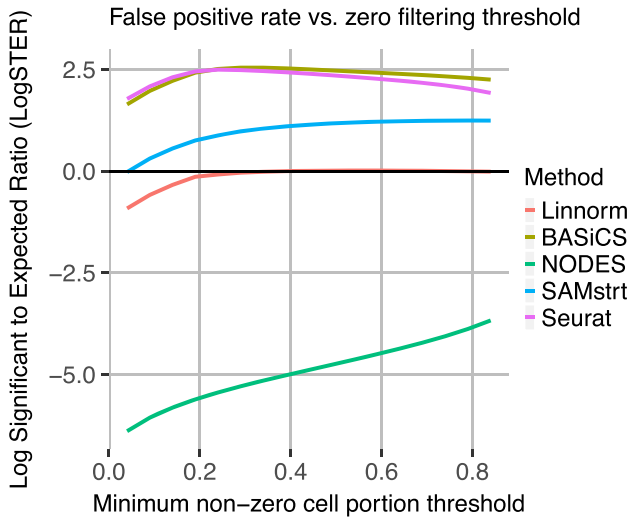

Figure 6. FPR and FNR across zero filtering thresholds in the absence of true differential expression. Without true DEGs, *P* value equals to the proportion of genes in a DEG analysis. In other words, 5% of the genes in the dataset should have a *P* value of <0.05 under the null. Hence, under a *P* value threshold, the number of significant genes should equal to the expected proportion of the genes. We plot the Log Significant To Expected Ratio (LogSTER) against the Minimum non-Zero cell Portion (MZP) threshold. MZP threshold of 0.25 means that genes with at least 25% non-zero values will be retained in the results. LogSTER of zero indicates finely controlled FPR and FNR, where the number of significant genes equals to the number of expected genes. High LogSTER, which indicates more significant genes than expected, would indicate high FPR; where low LogSTER would indicate high FNR. The sample set size of 10 and the *P* value threshold of 0.05 were utilized.

While Linnorm shows higher AUC performance than existing methods, it has a high amount of overlaps with existing methods (Supplementary Figure S5). These results indicate that Linnorm performs well in comparison with other scRNA-seq DEG analysis methods.

DISCUSSION

Stably expressed gene selection is an especially important process in scRNA-seq data modeling. Existing methods often assume that some genes in the dataset show stable expression and thrive to extract them. This is because heterogeneous expression levels can cause contradictions to the assumptions behind the data modeling method. In Linnorm's case, it thrives to transform the dataset toward homoscedasticity and normality. However, even after ignoring zeroes, we show that lowly expressing genes can cause lower SDs and significant skewness in the dataset (Figures 1 and 2). Together with naturally occurring heterogeneous genes that can induce more skewness in the dataset, this can cause violations to Linnorm's assumptions. Since existing scRNA-seq methods also have various assumptions that need to be satisfied, filtering steps are often included in scRNA-seq analysis methods to select a set of stable genes for data modeling.

Compared to RNA-seq data, where biological replicates often exist, each cell in a scRNA-seq dataset is expected to be unique. Because of the high amount of technical noises in scRNA-seq, these issues increase the difficultly of selecting stably expressed genes from a scRNA-seq dataset. Some existing methods avoid this issue by the utilization of spike-in genes or unique molecular identifiers (UMI), which are assumed to have gold standard quality by various algorithms. However, when spike-ins or UMIs are not available, existing methods would often rely on filtering thresholds and utilize the entire remaining dataset for modeling. Nevertheless, some of these filtering thresholds are rudimentary, such as the minimum average expression threshold (MAE) and the minimum non-zero cell threshold (MNZ). In comparison, Linnorm filtering algorithm has more considerations regarding different scenarios. First, Linnorm utilizes the relationship between mean and SD to filter low count genes, instead of the MAE threshold. Regardless of the sample size, we notice that lowly expressed genes near the raw count of near one would connect the origin to the global maximum in the mean versus SD plot (Figure 1). Compared to the MAE threshold, where raw count, CPM, TPM, etc. of one can hold drastically different meanings, Linnorm's method provides a reliable indicator of lowly expressed genes across more scenarios. Additionally, Linnorm filters gene with a high amount of zeroes by utilizing the MZP threshold, instead of the MNZ threshold. Some existing scRNA-seq methods have utilized the MNZ threshold of ~2–10. However, in single cell datasets with hundreds

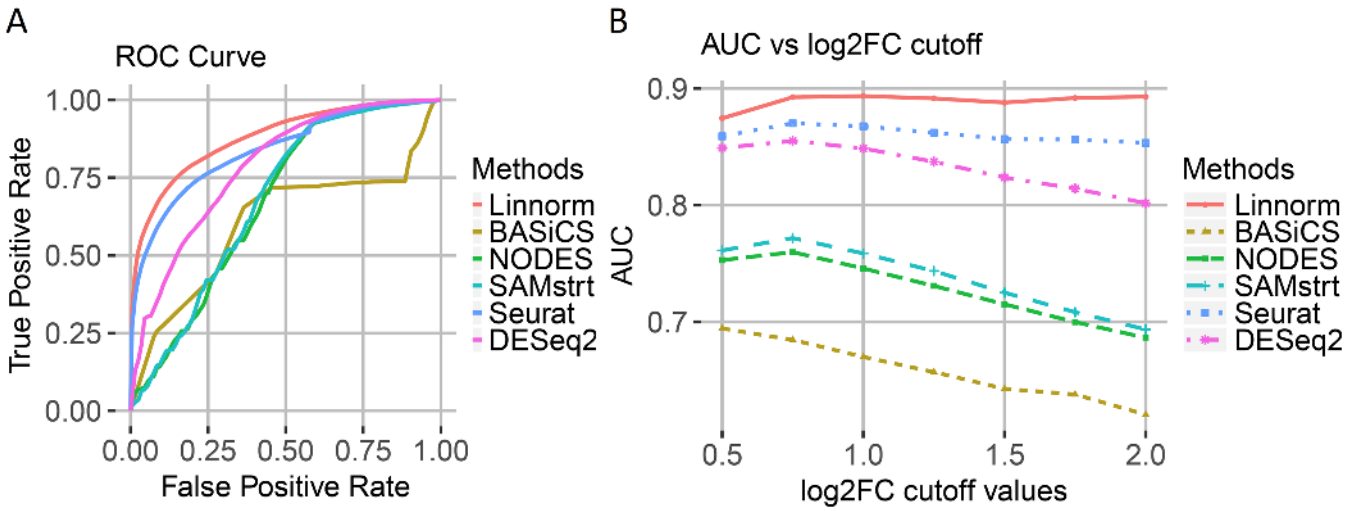

Figure 7. DEG analysis performance with the SEQC dataset. (**A**) Receiver operating characteristic (ROC) curve of SEQC DEG analysis using a sample set size of 10. True DEG are defined as genes with absolute $\log_2$-fold change (log2FC) higher than 1 in the gold standard Taqman dataset. (**B**) Area under the ROC curve across Taqman log2FCs from 0.5 to 2.

of samples, genes with only 2–10 non-zero cells are often overwhelmed by the raw count of near one. This results in unreliable SD and skewness (Figures 1 and 2). Linnorm's utilization of a proportion as the threshold increases its reliability across sample sizes. These improvements in data filtering have enhanced Linnorm's robustness across a wider range of situations.

Compared to existing methods, Linnorm is especially meticulous in the selection of stable genes for modeling. In statistical modeling, datasets with insufficient observations can only be subjected to limited filtering. However, scRNA-seq data often contain thousands, if not tens of thousands, of detected genes or transcripts. This allows the development of Linnorm's unsparing filtering strategy, which often filters >50% of the genes that show at least one expressing cell. Nonetheless, since Linnorm's filtering thresholds are often based on *P* values and proportions, it ensures that hundreds or thousands of genes would often remain in the dataset. Our results show that this is sufficient for scRNA-seq data modeling, which allows improved performances in multiple aspects. In this study, we find that choosing fewer but more accurate observations for modeling is more beneficial than choosing more but less accurate observations, because the number of observation in scRNA-seq data is large.

Traditional normalization methods often employ the use of the scaling factor. This assumes that expression levels are scaled linearly by the differences in sequencing depth (12,25,26,28). Linnorm's modified normalization strategy involves a linear regression analysis of the log-transformed dataset, which is performed between each sample's expression and the expression mean across samples. This strategy allows expression levels not only to be scaled similar to the other methods but also to be adjusted exponentially. This allows a better fit to the expression mean and more noise can be eliminated. When the optimal normalization solution is the scaling factor, Linnorm's normalization can become the scaling factor strategy, as its *m* parameters approach one. Another advantage of this strategy is that it does not eliminate cell heterogeneity. On the contrary, by aligning the expression values between the stable genes only, the variances of some highly variable genes can increase. While Linnorm's stronger noise elimination than existing methods implies its better FPR control (Figure 3), this increase of variances in highly variable genes is crucial for its FNR control and preservation of cell-to-cell differences (Figures 4 and 6). Closer examination of Linnorm's normalization in the gene-by-gene basis has shown that this strategy can increase the accuracy of the expression levels in the heterogeneously expressed genes (Figure 7). Lastly, we provide an adjustable normalization strength parameter, μ, such that users can control and optimize the strength of Linnorm's noise removal effects.

Linnorm transforms the dataset toward homoscedasticity and normality, which allows parametric tests to be applied more reliably. Conventionally, the CPM/TPM unit is often utilized prior to the log-plus-one transformation, where they multiply the relative expression matrix with an arbitrary number of one million. This arbitrary number could cause deviation from homoscedasticity after log-plus-one transformation. To solve this issue, the main goal of Linnorm's transformation step is to choose a better number, than one million, to multiply to the relative expression matrix, such that homoscedasticity and normality can be better achieved in a wider range of situations.

Because the total count in scRNA-seq data is generally low (Supplementary Table S1), λ from the transformation step can be smaller than a million, which results in lower numeric values of mean and SD than CPM (Figure 2). This raises a concern because smaller SDs in some genes may hinder the identification of noise from true variations, decreasing performances in clustering and DEG analyses. However, this is addressed by Linnorm. Since Linnorm's λ value is multiplied to all genes in the dataset, genes with a similar mean are affected similarly. To identify true variations in a dataset, a gene's mean or SD is often compared

within genes or to other genes that share a similar mean. Therefore, in the examination of clustering purity and DEG analysis accuracy, Linnorm is shown to be reliable. Another example can be shown with Seurat, where its 'λ' is set to be a constant of ten thousand, which is smaller than Linnorm's assigned λ in all five scRNA-seq datasets in this study. Nevertheless, Seurat's data transformation and imputation attained the second best result in preserving cell heterogeneity, and it shows inflated FPR in DEG analysis (Figures 4 and 6). This indicates that the smaller numeric values of mean and SD has lesser effects on statistical analyses than the underling normalization and transformation strategies.

Linnorm's transformation can be compared to existing methods that utilize transformation, such as DESeq-vst and voom. DESeq-vst's approach transforms the dataset with a log plus *n* transformation, where *n* is the adjustable parameter. voom sets *n* to ∼0.5 counts prior to a logarithmic transformation on the CPM unit. In comparison, Linnorm's transformation sets *n* to 1 and multiplies the dataset by an adjustable parameter. Unlike DESeq-vst and voom, Linnorm's approach ensures zeroes to remain zero after transformation. Compared to DESeq-vst, Linnorm's most notable improvement is computational time. Linnorm utilizes linear regression on the mean, SD and skewness of the stable genes to adjust the transformation parameter, such that F(λ) from Equation (3) can be calculated in one pass of the filtered expression matrix. To transform 2700 samples from the Klein dataset, Linnorm and DESeq-vst took 23 s and 12.5 h on average, respectively. Linnorm's transformation can also preserve a higher amount of biological variation in the dataset than DESeq-vst (Figure 4, Supplementary Figure S8 and Supplementary Table S7). Compared to voom, Linnorm has a better control of variance when there is a high amount of zero. voom's addition of 0.5 counts in the CPM unit induces negative numbers in the expression matrix. Because of the characteristic of the logarithmic function, these negative numbers would become exponentially smaller as the total count of the dataset become larger. This induces a larger difference between the count of one and the count of zero. This effect is reflected in Supplementary Figure S4. voom's LogSTER approaches zero with higher MZP, which indicates that voom performs optimally when there are less zeroes in a dataset.

Linnorm is shown to be robust with datasets that show distinct properties, including sample sizes that range from 6 to 2717 and average total read count that range from 0.02 million to 25.2 million. Our evaluations demonstrated that Linnorm is better than most normalization methods in removing technical variations, performed the best in preserving cell-to-cell differences for clustering, performed better than existing DEG analysis methods in balancing FPR and FNR in the presence of zero-counts, and has the highest accuracy in DEG analysis. We conclude that Linnorm is a reliable normalization and transformation method for scRNA-seq expression data.

DATA AVAILABILITY

Linnorm is open source and is written in C++ and implemented into an R Package. It is freely available at (http://www.jjwanglab.org/linnorm) and on Bioconductor (https://www.bioconductor.org/packages/release/bioc/html/Linnorm.html).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

Research Grants Council, Hong Kong SAR, China [17121414M]; Mayo Clinic (Mayo Clinic Arizona and Center for Individualized Medicine); National Institute of Health [5R01CA170357, 2P30CA015083, 1U54CA210180]. Funding for open access charge: Mayo Clinic (Mayo Clinic Arizona and Center for Individualized Medicine).
Conflict of interest statement. None declared.

REFERENCES

1. Wang,Z., Gerstein,M. and Snyder,M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10**, 57–63.
2. Islam,S., Kjällquist,U., Moliner,A., Zajac,P., Fan,J.-B., Lönnerberg,P. and Linnarsson,S. (2011) Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Res.*, **21**, 1160–1167.
3. Tang,F., Barbacioru,C., Wang,Y., Nordman,E., Lee,C., Xu,N., Wang,X., Bodeau,J., Tuch,B.B. and Siddiqui,A. (2009) mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods*, **6**, 377–382.
4. Katayama,S., Tohonen,V., Linnarsson,S. and Kere,J. (2013) SAMstrt: statistical test for differential expression in single-cell transcriptome with spike-in normalization. *Bioinformatics*, **29**, 2943–2945.
5. Lun,A.T., Bach,K. and Marioni,J.C. (2016) Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.*, **17**, 75.
6. Zhou,X., Lindsay,H. and Robinson,M.D. (2014) Robustly detecting differential expression in RNA sequencing data using observation weights. *Nucleic Acids Res.*, **42**, e91.
7. Love,M.I., Huber,W. and Anders,S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.
8. Ritchie,M.E., Phipson,B., Wu,D., Hu,Y., Law,C.W., Shi,W. and Smyth,G.K. (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.*, **43**, e47.
9. Robinson,M.D. and Oshlack,A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.*, **11**, R25.
10. Brennecke,P., Anders,S., Kim,J.K., Kolodziejczyk,A.A., Zhang,X., Proserpio,V., Baying,B., Benes,V., Teichmann,S.A., Marioni,J.C. *et al.* (2013) Accounting for technical noise in single-cell RNA-seq experiments. *Nat. Methods*, **10**, 1093–1095.
11. Law,C.W., Chen,Y., Shi,W. and Smyth,G.K. (2014) voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.*, **15**, R29.
12. Bacher,R., Chu,L.-F., Leng,N., Gasch,A.P., Thomson,J.A., Stewart,R.M., Newton,M. and Kendziorski,C. (2017) SCnorm: robust normalization of single-cell RNA-seq data. *Nat. Methods*, **14**, 584–586.
13. Satija,R., Farrell,J.A., Gennert,D., Schier,A.F. and Regev,A. (2015) Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, **33**, 495–502.
14. Vallejos,C.A., Richardson,S. and Marioni,J.C. (2016) Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome Biol.*, **17**, 1.
15. Trapnell,C., Cacchiarelli,D., Grimsby,J., Pokharel,P., Li,S., Morse,M., Lennon,N.J., Livak,K.J., Mikkelsen,T.S. and Rinn,J.L. (2014) Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nat. Biotechnol.*, **32**, 381.

16. Poole,W., Gibbs,D.L., Shmulevich,I., Bernard,B. and Knijnenburg,T.A. (2016) Combining dependent *P*-values with an empirical adaptation of Brown's method. *Bioinformatics*, **32**, i430–i436.
17. Yan,L., Yang,M., Guo,H., Yang,L., Wu,J., Li,R., Liu,P., Lian,Y., Zheng,X., Yan,J. *et al.* (2013) Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat. Struct. Mol. Biol.*, **20**, 1131–1139.
18. Deng,Q., Ramskold,D., Reinius,B. and Sandberg,R. (2014) Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, **343**, 193–196.
19. Patel,A.P., Tirosh,I., Trombetta,J.J., Shalek,A.K., Gillespie,S.M., Wakimoto,H., Cahill,D.P., Nahed,B.V., Curry,W.T., Martuza,R.L. *et al.* (2014) Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, **344**, 1396–1401.
20. Klein,A.M., Mazutis,L., Akartuna,I., Tallapragada,N., Veres,A., Li,V., Peshkin,L., Weitz,D.A. and Kirschner,M.W. (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.
21. SEQC/MAQC-III Consortium. (2014) A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nat. Biotechnol.*, **32**, 903–914.
22. Bolger,A.M., Lohse,M. and Usadel,B. (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, **30**, 2114–2120.
23. Bray,N.L., Pimentel,H., Melsted,P. and Pachter,L. (2016) Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.*, **34**, 525–527.
24. Yates,A., Akanni,W., Amode,M.R., Barrell,D., Billis,K., Carvalho-Silva,D., Cummins,C., Clapham,P., Fitzgerald,S., Gil,L. *et al.* (2016) Ensembl 2016. *Nucleic Acids Res.*, **44**, D710–D716.
25. Bacher,R. and Kendziorski,C. (2016) Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biol.*, **17**, 63.
26. Stegle,O., Teichmann,S.A. and Marioni,J.C. (2015) Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.*, **16**, 133–145.
27. Svensson,V., Natarajan,K.N., Ly,L.-H., Miragaia,R.J., Labalette,C., Macaulay,I.C., Cvejic,A. and Teichmann,S.A. (2017) Power analysis of single-cell RNA-sequencing experiments. *Nat. Methods*.
28. Loven,J., Orlando,D.A., Sigova,A.A., Lin,C.Y., Rahl,P.B., Burge,C.B., Levens,D.L., Lee,T.I. and Young,R.A. (2012) Revisiting global gene expression analysis. *Cell*, **151**, 476–482.
29. Lin,C.Y., Lovén,J., Rahl,P.B., Paranal,R.M., Burge,C.B., Bradner,J.E., Lee,T.I. and Young,R.A. (2012) Transcriptional amplification in tumor cells with elevated c-Myc. *Cell*, **151**, 56–67.
30. Synnergren,J., Giesler,T.L., Adak,S., Tandon,R., Noaksson,K., Lindahl,A., Nilsson,P., Nelson,D., Olsson,B., Englund,M.C. *et al.* (2007) Differentiating human embryonic stem cells express a unique housekeeping gene signature. *Stem Cells*, **25**, 473–480.
31. Soneson,C. and Delorenzi,M. (2013) A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, **14**, 91.
32. Rapaport,F., Khanin,R., Liang,Y., Pirun,M., Krek,A., Zumbo,P., Mason,C.E., Socci,N.D. and Betel,D. (2013) Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biol.*, **14**, R95.