Much ado about nothing: how to normalize single-cell RNA sequencing data with many zero counts

Supplementary Materials

by

Aaron T. L. Lun[1], Karsten Bach[2] and John C. Marioni[1,2]

[1]Cancer Research UK Cambridge Institute, University of Cambridge, Li Ka Shing Centre, Robinson Way, Cambridge CB2 0RE, United Kingdom

[2]EMBL European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom

February 3, 2016

1 Resolving linear dependencies in the constructed system

Consider the application of the deconvolution method on a data set with four cells using a sliding window of size 2. Assuming cells $j = 1$ to 4 were placed consecutively on the ring, this would yield the linear system

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \theta_A \\ \theta_B \\ \theta_C \\ \theta_D \end{bmatrix}$$

for pool-based size factor estimates θ_A to θ_D. Assume that the pool-based factors are estimated accurately and precisely, such that the minimum value of the residual sum of squares for this system can be obtained near the set of true values for the cell-based factors. This system has no unique solution - for the true factors $(\theta_1, \theta_2, \theta_3, \theta_4)^T$, an equally good fit can be obtained with $(\theta_1 + x, \theta_2 - x, \theta_3 + x, \theta_4 - x)^T$ for any real x.

The addition of equations relating each θ_j with its direct estimate θ'_j ensures identifiability, as a value of x will be chosen that minimizes the residual sum of squares of the $(\theta_1 + x, \theta_2 - x, \theta_3 + x, \theta_4 - x)^T$ from the direct estimates. In this example, the residual sum of squares for the possible solutions can be written as

$$\sum_{j \in J_1} (\theta_j + x - \theta'_j)^2 + \sum_{j \in J_2} (\theta_j - x - \theta'_j)^2 = 4x^2 + 2x \left[\sum_{j \in J_1} (\theta_j - \theta'_j) - \sum_{j \in J_2} (\theta_j - \theta'_j) \right] + \sum_{j \in \{J_1, J_2\}} (\theta_j - \theta'_j)^2$$

where $J_1 = \{1, 3\}$ and $J_2 = \{2, 4\}$. For these 4 cells, the above expression is minimized in terms of x at

$$x = -\frac{1}{4} \left[\sum_{j \in J_1} (\theta_j - \theta'_j) - \sum_{j \in J_2} (\theta_j - \theta'_j) \right] .$$

The direct estimate of each size factor is unbiased as stochastic zeroes are not removed, i.e., $E(\theta_j - \theta'_j) = 0$.

As the number of cells in the system increases, the sizes of J_1 and J_2 will increase. For example, deconvolution is typically applied in data sets involving at least 200 cells and a sliding window of size 20, such that both J_1 and J_2 will consist of at least 10 cells. With more cells, the sum of the errors $\theta_j - \theta'_j$ will approach the sum of their expected values, i.e., zero. This means that the minimum residual sum of squares will be obtained at $x = 0$, i.e., solving the linear system will yield the true values of the size factors. Thus, the additional equations will not affect accurate estimation of the size factors in the deconvolution method.

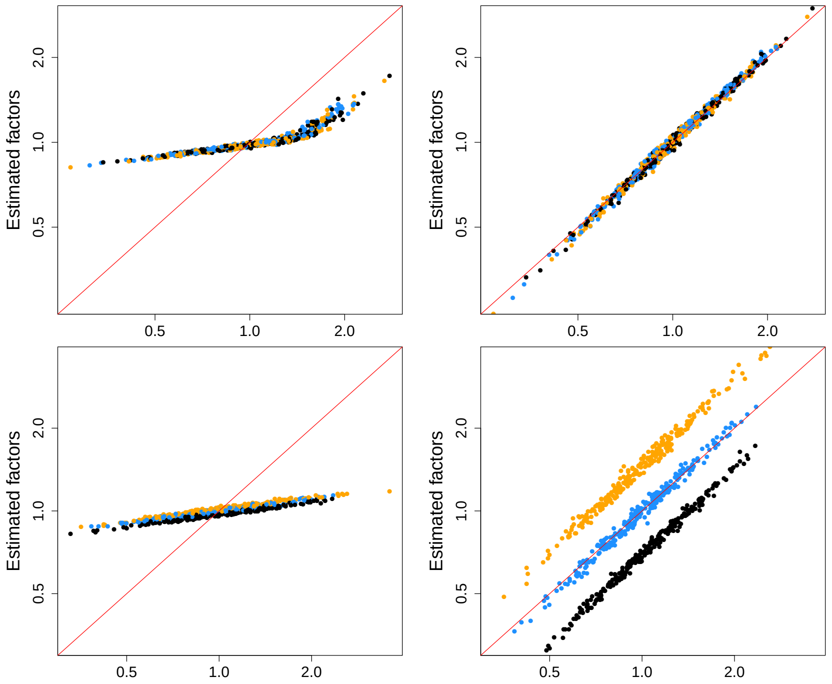
2 Implementation details of the clustering approach

Let ρ_{xy} denote Spearman's rank correlation coefficient between the counts of cells x and y. We define the distance between these cells as $1 - \rho_{xy}$. In this manner, a distance matrix is constructed between all pairs of cells. Hierarchical clustering is performed on this matrix using the hclust function with Ward's clustering criterion. Clusters of cells are defined using a dynamic tree cut from the dynamicTreeCut package v1.62 (https://cran.r-project.org/web/packages/dynamicTreeCut/index.html). This ensures that each cluster contains a minimum number of cells (200 by default) required for stable deconvolution. Correlation-based clustering is appealing as it is insensitive to global scaling of the expression values in each cell. Prior normalization is not required, which avoids a circular dependence between normalization and clustering. Alternatively, one can use known aspects of the data set as clusters, e.g., groupings, batches. Empirical clustering may not be required if such information is available, which reduces computational work and reduces the potential for errors.

By default, the baseline pseudo-cell is chosen from the cluster where the mean library size per cell is equal to the median of the mean library size across all clusters (or, for an even number of clusters, the cluster with the smallest mean library size above the median). This uses the mean library size as a rough proxy for cell similarity. The baseline cluster is likely to be least dissimilar to every other cluster, which reduces the amount of DE during pairwise normalization between pseudo-cells. More intelligent choices of the baseline can be used if the similarities between clusters are known, e.g., from visualization after dimensionality reduction.

In general, cluster-specific normalization requires some caution. Done incorrectly, this may introduce artificial differences between cells in different clusters, such that the statistical rigour of downstream analyses (e.g., to detect DE between clusters) would be compromised. However, such problems are avoided here by using a two-step normalization strategy. The first normalization step removes any systematic differences between cells *within* each cluster, while the second normalization between pseudo-cells removes any differences *between* clusters. The end result is that differences between all cells in all clusters are removed. This is equivalent to the outcome of a hypothetical one-step method that does not use cluster information (and is robust to DE and stochastic zeroes, unlike existing methods). Moreover, normalization accuracy in Figure 4 is unaffected by the use of clustering prior to deconvolution, which suggests that this approach is valid.

Semi-systematic zeroes may be removed prior to deconvolution in each cluster. In particular, genes are removed if they only have zero counts across all cells in the cluster. Such genes provide no information for normalizing between cells in the same cluster, and their removal will not affect the cluster-specific size factor estimates. However, these genes are retained during rescaling of the size factors between clusters. This is because they will have non-zero counts in at least one cluster (assuming that systematic zeroes across the entire data set have already been removed). Removal of such genes will distort the median ratio between pseudo-cells and lead to biased size factor estimates, as described previously for the existing methods.



(a) **(b)**

Figure S1: Performance of DESeq normalization after addition of a pseudo-count. A pseudo-count of unity was added to all counts directly (a) or after scaling the pseudo-count by the relative library size (b), i.e., the pseudo-count added to each cell was equal to the ratio of its library size to the mean library size. Simulations were performed with no DE (first row) and varying magnitudes of DE (second row). Axes are shown on a log-scale. For comparison, each set of size factors was scaled such that the grand mean across cells was the same as that for the true values. The red line represents equality between the rescaled estimates and true factors. Cells in the first, second and third subpopulations are shown in black, blue and orange, respectively.

Table S1: Proportion of the top set of DE genes that were shared between deconvolution and each existing normalization method. Top genes were identified as those with the lowest p-values from edgeR in the analysis with each normalization method. Top sets of size ranging from 100 to 2000 were tested for both the brain and inDrop data sets. Smaller sets were not tested due to avoid ambiguous ranks from p-values of zero.

Data set	*Top*	*Method*		
		DESeq	TMM	Library size
Brain	100	0.44	0.60	0.71
	500	0.60	0.70	0.75
	2000	0.70	0.75	0.80
inDrop	100	0.88	0.94	0.98
	500	0.84	0.92	0.97
	2000	0.83	0.93	0.95

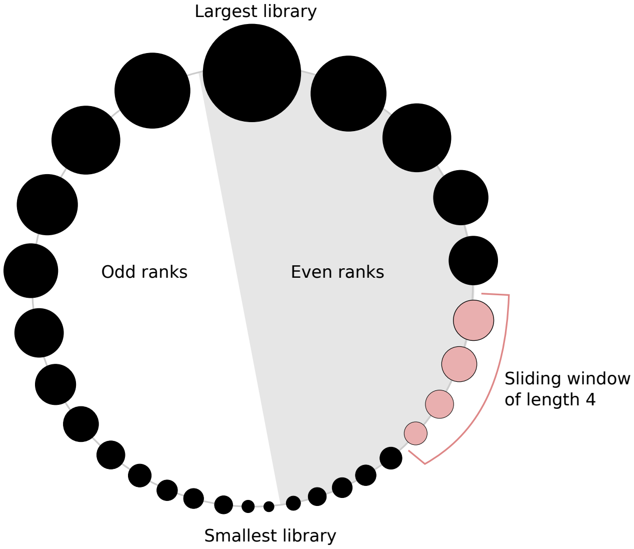


Figure S2: Ring arrangement of cells ordered by library size. Each circle represents a cell where the size of the circle corresponds to the library size of that cell. Even- and odd-ranking cells lie on opposite sides, with the largest and smallest libraries at the top and bottom, respectively. Cells lying in a window of length 4 are highlighted in red. Different instances of the window are obtained by sliding the window across the ring.

Table S2: Proportion of the top set of highly variable genes that were shared between deconvolution and each existing normalization method. Top genes were identified as those with the largest distance-to-median values. For both data sets, top sets of size ranging from 100 to 2000 were compared between methods.

Data set	*Top*	*Method*		
		DESeq	TMM	Library size
Brain	100	0.78	0.81	0.90
	500	0.78	0.80	0.88
	2000	0.67	0.73	0.84
inDrop	100	0.74	0.76	0.90
	500	0.57	0.59	0.85
	2000	0.59	0.61	0.85