Dear Barbara,

Thank you for a rapid and constructive review process on our manuscript. We are grateful for your comments and those of the two reviewers, which highlight the utility of our approach and the relevance of the problem.

We have revised the manuscript to take into account of all the reviewers’ comments and feel that the new version is much improved. In particular, we have added a new analysis (in the main text and Section 6 of the Supplementary Materials) to demonstrate how our approach out-performs existing methods on real data. Additionally, as requested by reviewer 1, we now provide a measure of the uncertainty in our estimates of the size factors. Finally, we have performed additional comparisons, using new simulations that better capture the sparse nature of low-coverage single-cell RNA sequencing data, which illustrate the good performance of our method.

Detailed responses to these and the other comments raised by the reviewers are provided below.

We hope that you enjoy reading our revised manuscript and look forward to hearing from you soon.

Best wishes,

Aaron Lun & John Marioni

***# REVIEWER 1:***

**> [1] The title is witty but unnecessary. The authors should be mindful that the journal is read by an international audience and a simple and straightforward title would be more appealing.**

We agree with the referee and have changed the title to “Pooling across cells to normalize single-cell RNA sequencing data with many zero counts”.

**> [2] The authors provide a thorough and detailed description of the limitations provided by existing normalization methods - in particular DEseq2 and TMM. However, these methods were never designed with single cell data processing so although it is useful to briefly highlight their ineffectiveness with single cell zero count data, I am not sure the authors need to devote so much of the paper to its discussion. It takes nearly six pages and two figures before we reach the interesting bit which is their proposed method.**

We have substantially re-written the manuscript to reflect this concern. In particular, we now provide much less detail on the implementation of existing normalization approaches and refer to the appropriate literature. Moreover, we more clearly state that these approaches were not designed with single-cell RNA sequencing data in mind. However, we still provide an assessment of their performance using our simulated data since they are widely applied in the context of single-cell RNA sequencing studies (e.g., Kolodziejczyk et al. (2015), Li et al. (2015), Deng et al. (2014) and Freeman et al. (2015)).

**> [3] The authors proposed one possible pooling strategy. How much of an effect does the choice of pooling have on size factor estimates? A statement should be made on this issue or a simulation study to investigate the stability of the size factor estimates with respect to pooling selection should be performed.**

This is an interesting point. In the initial design of the deconvolution method, we randomly selected pools of cells to construct the linear system. To our surprise, this actually performed quite well. Eventually, though, we refined the method to use a ring arrangement, which provides a modest improvement in precision. We have added some simulations to the Supplementary Materials showing a two-fold decrease in the median (log-)error when the ring is used instead of random pools.

**> [4] The deconvolution method requires the solution of a linear system in which their as many equations as cells. Is this computationally problematic for studies involving very large cell numbers? The authors could comment on the computational demands of their method.**

This is a good point and we have added a new section, “Computational complexity of the deconvolution method” to the Supplementary Materials. Therein, we note that the computational time and memory required by deconvolution depends on the number of cells. For each cell, an equation is constructed, meaning that the size of the linear system (in terms of the number of equations and coefficients) increases quadratically with the number of cells. Furthermore, common implementations of the QR decomposition have O(n3) time complexity for n cells. We observe that this is approximately consistent with our empirical timings (new Supplementary Figure S6).

In practice, we mitigate against the cubic time complexity by using clustering to break up the linear system (main text and Section 3 of the revised Supplementary Materials). Since the deconvolution strategy is applied within each cluster, the overall time complexity depends on the number of clusters and size of each cluster (a few large clusters will result in cubic complexity, while many small clusters will result in linear complexity). Finally, we note that most clustering algorithms have quadratic time complexity with respect to the number of cells, e.g., when building the distance matrix. However, we expect that clustering will be performed as a routine part of scRNA-seq data analysis, meaning that no additional time is added by using this information in the normalization.

**> [5] The authors use weighted least squares to give a point estimate of the size factors but the method would be much more powerful if uncertainty information could be computed and if this uncertainty could be propagated into downstream analysis.**

We agree that incorporating uncertainty in the size factor estimates into downstream analyses would be helpful. However, in practice, pipelines like DESeq2 and edgeR, which are commonly applied to scRNA-seq data, assume that size/normalization factors are known values. Similarly, monocle (a dedicated single-cell analysis method) requires normalized expression values as input, without any provision for the variability of the normalization procedure. In general, a holistic framework is required to incorporate uncertainty into downstream inferences, e.g., using a Bayesian model as described by Vallejos et al. (2015). This is not compatible with the majority of existing methods that are more modular.

Nevertheless, we now provide an option in the function to obtain standard errors from the linear system, and we have commented on this possibility in the manuscript.

**> [6] A major difficulty that the authors face is demonstrating the effectiveness of the method in real world analysis situations. The authors offer some anecdotal evidence based on the analysis of two single cell data sets but ultimately the conclusion is little more than a statement that different size factor estimates can lead to different DE gene lists. I think more needs to be done to show the strength of the method in real world scenarios even if this is only defined statistically. For example, random subsets of the data could be taken and the DE gene lists compared across subsets to identify the genes that overlap. Does the deconvolution method lead to more stable gene lists that consistently report the same DE genes? This is merely a suggestion but something beyond a statement that "these results suggest that the genes unique to deconvolution are more biologically relevant that those unique to library size normalization" is required to make the analysis more credible.**

This is a very good point. To address this concern we have performed additional analyses to demonstrate the efficacy of our approach as suggested by the reviewer.

In particular, we have added a GLM-based analysis to demonstrate the relative accuracy of the deconvolution method compared to other methods. For a homogeneous population of cells (from real data) we first compute the size factors using the deconvolution and an alternative approach. Subsequently, for each gene, we model the expression counts using a negative binomial GLM. We use the deconvolution (log-)size factors as GLM offsets, while the other set of size factors is used as a covariate. If the normalization is accurate, we would expect the offsets to capture all differences between cells (in this homogeneous population) due to cell-specific biases. Consequently, we would not expect including the covariate term (the other normalization factor) to improve the fit of the model. To assess whether this was the case, we used a likelihood ratio test to compare the null model with no covariate with the alternative model. We repeated this analysis swapping the role of the deconvolution derived size factor with that of the alternative size factor.

We observed that the number of genes where there was evidence to reject the null hypothesis was significantly greater when the deconvolution size factor was used as the covariate (Supplementary Figure S7), strongly suggesting that the deconvolution approach better captures technical biases in the data than the alternative normalization strategies.

We comment on this new analysis in the main text and provide a more rigorous description of the testing procedure in new Section 6 of the Supplementary Materials.

***# REVIEWER 2:***

**> 1. The only comparisons done are against TMM and DESeq's normalization (and library size). I'm not up-to-date on all the normalization approaches that exist, but a high amount of 0 counts come up in other related fields, such as metagenomics. For example, the CSS method:**

**http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4010126/**

**Since TMM and DESeq were not designed with any of the considerations of scRNA-seq, the comparisons set the bar a little bit low.**

We agree with the reviewer that TMM and DESeq were not designed for single-cell RNA sequencing data and, therefore, it is not surprising that they do not perform well in practice. We now emphasise this point in the main text (see also our response to point 2 from reviewer 1). However, in practice, these approaches are widely used in the single-cell RNA sequencing literature (e.g., Kolodziejczyk et al. (2015), Li et al. (2015), Deng et al. (2014) and Freeman et al. (2015)). Consequently, we feel that it is important to formally assess their performance.

The reviewer’s suggestion that other methods used outside the RNA sequencing field is interesting. However, since these methods would require some adaptation for RNA sequencing data and, more importantly, have not been applied previously in this context, we feel that assessing their performance falls outside the scope of this study.

**> 2. Code should be made available (e.g., a github repo) for the simulation and the analyses in the paper. This appears to be almost a standard nowadays.**

We agree, and all code for the manuscript is available on https://github.com/MarioniLab/Deconvolution2016.

**> 3. I would like to see some more description, in the context of a real scRNA-seq dataset, of the chosen simulation parameters. There is a statement on Page 4 that "This recapitulates the spread of abundances in real data", in reference to a log(lambda) ~ Uniform. I have never seen a distribution of expression levels that is remotely uniform (but perhaps I misunderstand something). What I suggest is to make a small number of visual diagnostics (e.g., distributions of expression, dispersion-mean plots) of the simulated data alongside the real data (Supplementary Figures, probably). There should be some minimal correspondence, otherwise the simulation is questionable.**

This is a good point. In the previous submission our simulations were based on high-coverage scRNA-seq data sets that we had previously worked on, rather than on the low-coverage Zeisel and Klein data sets that we consider in the paper. We have rectified this in the revised submission. More specifically, we have updated the simulation design and now provide relevant diagnostic plots in the Supplementary Materials, which show that our simulated range of expression levels recapitulates the distribution of gene expression levels observed in real data sets.

Importantly, these modifications do not affect our conclusions. In fact, the differences between normalization methods in Figures 1 and 4 now demonstrate even more clearly the advantages of the deconvolution approach. Finally, the original simulations have been moved to the Supplementary Materials and illustrate the performance of the various methods on high-coverage data.

**> 4. The authors are very minimalistic in their approach to R software. Sure, it is publicly available from github, but there should at least be a README that tells the user how to install the package and a vignette with some basic examples. This is either sloppy or lazy. For this kind of package (and these authors), surely Bioconductor submission is a minimum requirement.**

The scran package was submitted to Bioconductor on the 7th of March, along with full documentation, tests and a vignette. We anticipate that it will be available in the upcoming release (BioC 3.3) sometime in April.