Supplementary Information for

Gene Expression Distribution Deconvolution in Single Cell RNA Sequencing

Jingshu Wang, Mo Huang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, John Murray, Arjun Raj, Mingyao Li, and Nancy R. Zhang

Nancy R. Zhang
E-mail: nzh@wharton.upenn.edu

This PDF file includes:

Supporting Information Text

Data sources of publicly available datasets. The ERCC UMI count matrix of (1–3) are downloaded from the NCBI GEO website (GSE54006, GSE63473, GSE78779). The raw FASTQ files of the 10x data from (4) is released as ArrayExpress E-MTAB-5480 and we obtain the mapped UMI counts from the original authors. The UMI count matrices of both biological genes and ERCC spike-ins in Klein et al. (5) and Zeisel et al. (6) are downloaded from the NCBI GEO website (GSE65525, GSE60361). The count matrices of Tung et al. (7) are downloaded from the Github page: https://github.com/jdblischak/singleCellSeq. Both the ERCC data and the datasets of 10 purified cell types in Zheng et al. (8) are downloaded from the 10x genome website: https://support.10xgenomics.com/single-cell-gene-expression/datasets. To calculate the expected number of molecule amount for each ERCC spike-in gene across cells, we referred to Table 1 of Svensson et al. (4) which summarized the dilution ratios and volumes of the ERCC spike-ins in each of the publicly available datasets. All the ERCC spike-in datasets and information that are used in the manuscript are also publicly available on our Github repository: https://github.com/jingshuw/DESCEND_manuscript_source_code. The CPM counts of the Drop-seq data in Torre et al. (9) can be downloaded from the NCBI GEO website (GSE99330). Both the raw Drop-seq UMI counts and the RNA FISH counts for the 26 genes profiled by both technologies are available on our Github repository.

A comparison of the CV, Gini coefficient and Fano factor. Both CV and Gini coefficient are scale invariant measures of the dispersion of a distribution. Given a sample $(x_1, x_2, \cdots, x_n)$, the Gini coefficient is defined as

$$G = \frac{\sum_i \sum_j |x_i - x_j|}{2n \sum_i x_i} = \frac{\frac{1}{n^2} \sum_i \sum_j |x_i - x_j|}{2\mu}$$

where $\mu = \sum_i x_i / n$ is the sample mean. On the other hand,

$$\frac{\sqrt{\frac{1}{n^2} \sum_i \sum_j (x_i - x_j)^2}}{2\mu} = \frac{\sqrt{\frac{1}{n^2} \sum_i \sum_j (x_i - \mu - x_j + \mu)^2}}{2\mu}$$
$$= \frac{\sqrt{2\sigma^2}}{2\mu} = \frac{\mathrm{CV}}{\sqrt{2}}$$

where σ^2 is the sample variance. Comparing the two, the Gini is merely a robust version of the CV, replacing the square-norm with the absolute-norm.

In contrast, the Fano factor, defined as σ^2/μ is not scale invariant, thus can not be estimated when the cell specific efficiency constants are unavailable. A selection of HVGs based on Fano factors favors highly expressed gene. For a gene g, let the DESCEND recovered true gene expression be the distribution of $k_g \lambda_{cg}$ where k_g is some unknown scaling factor. In addition, assume that the biological variation of gene g has the decomposition $\sigma_g^2 = \mu_g + \theta_g \mu_g^2$, then the Fano factor of the recovered distribution is

$$\frac{k_g^2 \sigma_g^2}{k_g \mu_g} = k_g(1 + \theta_g \mu_g).$$

indicating that for genes with the similar dispersion parameters, selection based on Fano factors favors more dispersed and highly expressed genes. Also, it depends on the assumption that the efficiency of all the genes are approximately the same, which may not be true.

Dispersion calculation of ERCC data from Svensson et al. (4). (4) contains two datasets at different concentration levels and because of the low efficiency (less than 0.01% on average per cell) of the experiment, their dispersion measured by CV is calculated using a simple moment method.

For a single ERCC spike-in gene, under the "Poisson-alpha" noise model, as the observed count in one cell follows $Y_c \sim \mathrm{Poisson}(\alpha_c \lambda_c)$, then conditional on α_c and assume $\lambda_c \sim \mathrm{Poisson}(\lambda)$ where λ is the expected molecule input amount, we have $\mathrm{Var}(Y_c) = \alpha_c(1 + \alpha_c)\lambda$. as the cell efficiency $\alpha_c \approx 10^{-4}$ for the ERCC data from (4), we have $\mathrm{Var}(Y_c) \approx \alpha_c \lambda$ and $\mathbb{E}(Y_c) = \alpha_c \lambda$.

Thus, if the "Poisson-alpha" noise model is plausible, we should have $\mathrm{Var}\left(\sum_c Y_c\right) = \sum_c \alpha_c \lambda$. We calculate the sample dispersion (rescaled to have mean λ) of this spike-in gene across cells as

$$\frac{\sqrt{\sum_c (Y_c - \alpha_c \lambda)^2 / \sum_c \alpha_c}}{\lambda}$$

and compare it with $1/\sqrt{\lambda}$, the CV of Poisson at mean λ, to check if the relationship $\mathrm{Var}\left(\sum_c Y_c\right) = \sum_c \alpha_c \lambda$ under the "Poisson-alpha" model is fitted or not.

The reason that we use this simple moment method instead of DESCEND to check for over-dispersion for this dataset is that under the scenario of super low cell efficiency with concentrated input molecule amount across cells (CV is mostly 0 in this dataset), DESCEND is biased because of the usage of a penalized likelihood and the moment method provides a more accurate assessment of dispersion for such ERCC dataset.

Practical considerations for discretization of the deconvolved distribution in DESCEND. One component of the DESCEND algorithm is the discretization of the deconvolved distribution to simplify computation. For discretization, we need to consider both the range of discretized points and number of points we use. We find in practice that DESCEND is robust to the number of points and using $m = 50$ points is enough to give satisfying results in most the datasets we have analyzed as the distribution of gene expression is typically smooth, even when in a heterogenous population.

Choosing the range of points is more critical. The lower limit is always 0, but for the upper limit, we need to avoid either making it too large so that most points has almost zero probability or making it too small so that the upper tail of the true expression distribution can not be well estimated. We estimate the upper limit of the distribution from the observed counts. In DESCEND, if at most 95% of the cells have non-zero UMI count for gene g then the upper limit is set as the 98% percentile of $\{Y_{cg}/\alpha_c,\ c = 1, 2, \cdots C\}$. However, in scRNA-seq data, there can be many genes that are very sparse. In DESCEND, if the sparsity of gene g is larger than 95%, then the upper limit is set to be the 80% percentile of $\{Y_{cg}/\alpha_c,\ c = 1, 2, \cdots C, Y_{cg} \neq 0\}$.

Details of the simulation experiments. In the sample-splitting simulation, the 820 Oligodentroctyes cells from Zeisel et al. (6) are randomly split into two equal-sized groups. DESCEND uses the ERCC spike-ins to compute the cell-specific efficiencies as α_c and deconvolve the true absolute gene expression distribution separately for each genes in each cell group.

In the parameteric simulation, we first run DESCEND to the observed UMI counts of 820 cells to get the estimated parameter values. Here, we first estimate the cell size for each cell and then run DESCEND to deconvolve cell size adjusted distribution. The log of cell size is added as the covariate for both the nonzero fraction and nonzero mean. To generate pseudo scRNA-seq UMI counts, We assume that the cell size adjusted gene expression follows a zero-inflated log-Normal distribution for each gene, where the mean and variance match the corresponding parameters of the deconvolved distribution by DESCEND. We create "null" genes by setting the nonzero fraction as 1 for the genes whose estimated nonzero fraction by DESCEND is larger than 0.8. For the coefficients of cell size on the pseudo counts, we keep the DESCEND estimated coefficients of cell size on nonzero mean and nonzero fraction as true parameters for this synthetic data. Only genes whose average UMI counts per cell among originally observed Oligodendroctyes cells is larger than 0.3 are used for generating pseudo RNA-seq counts, resulting in 5045 genes used. The estimated technical noise model is taken as the true model, thus the simulated UMI count data matrix has the same size and the same per-cell efficiency as the original filtered data.

In the down-sampling simulation, we first select the top 150 genes with the highest average UMI counts among the 820 cells. For these genes, the original observed UMI counts are treated as the true underlying gene expression in each cell. Then, we generate down-sampled observed counts from these true counts by adding noise following the "Poisson-alpha" noise model with efficiencies being $\alpha_c \equiv 20\%, 10\%$ and 5%. We compared the estimated parameters using DESCEND from these down-sampled observed counts with the true parameters values (the values calculated from the original observed UMI counts).

Pre-processing and analysis of the RNA FISH data from Torre et al. (9). In this dataset, both Drop-seq and RNA FISH are applied to the same melanoma cell line. For the Drop-seq experiment, cells with library size less than 1000 UMIs are removed (as we use GAPDH to normalize the data for recovering the relative gene expression), and 5763 cells are left with median 1473 UMIs per cell for further analysis. For the RNA FISH experiment, cells with less than 100 or more than 1000 GAPDH UMI read counts are removed, with 79099 cells left for further analysis. Of the 26 genes profiled by RNA FISH, the genes VCF and FOSL1 are removed as we find apparent inconsistency between the Drop-seq data and RNA FISH data for these two genes (The fraction of zero read counts are too high in the RNA FISH data compared with the Drop-seq data). The RNA FISH data is normalized by GAPDH, thus we can analyze the expression of at most 23 genes (excluding VCF, FOSL1 and GAPDH). The RNA FISH data are treated as gold standard, meaning that we ignore the measurement errors and assume that the RNA FISH counts represent the true expression level of the genes. The DESCEND recovered distribution is re-centered to have the same mean as the corresponding RNA FISH distribution as we allow for the linear form the "Poisson-alpha" noise model $\alpha_{cg} = \alpha_c\gamma_g$. The distribution based measurements: nonzero fraction, CV and Gini coefficients are all scaling invariant.

For its Drop-seq dataset, we computed the CV from the raw normalized counts using conditional variance decomposition under the "Poisson-alpha" noise model and compare the values with the DESCEND estimated ones. For a fixed gene g, let $\mu_Y(\sigma_Y), \mu_\alpha(\sigma_\alpha), \mu_\lambda(\sigma_\lambda)$ be the mean (variance) of Y_{cg}, α_c and λ_{cg} across cells respectively. Here α_c is the library size of each cell. Then, we have

$$\begin{aligned}
\sigma_Y^2 &= \mathbb{E}\Big(\operatorname{Var}\left[Y_{cg} \mid \alpha_c\right]\Big) + \operatorname{Var}\Big(\mathbb{E}\left[Y_{cg} \mid \alpha_c\right]\Big) \\
&= \mathbb{E}\Big(\operatorname{Var}\left[\alpha_c\lambda_{cg} \mid \alpha_c\right] + \mathbb{E}\left[\alpha_c\lambda_c \mid \alpha_c\right]\Big) + \operatorname{Var}\left[\alpha_c\mu_\lambda\right] \\
&= \mathbb{E}\left[\alpha_c\right]^2 \operatorname{Var}\left[\lambda_{cg}\right] + \mathbb{E}\left[\alpha_c\right]\mathbb{E}\left[\lambda_{cg}\right] + \mu_\lambda^2 \operatorname{Var}\left[\alpha_c\right] \\
&= \sigma_\alpha^2\sigma_\lambda^2 + \mu_\alpha^2\sigma_\lambda^2 + \sigma_\alpha^2\mu_\lambda^2 + \mu_\alpha\mu_\lambda
\end{aligned}$$

Divide by $\mu_Y^2 = \mu_\alpha^2\mu_\lambda^2$ on both sides, we get

$$(1 + \mathrm{CV}_\alpha^2)\mathrm{CV}_\lambda^2 = \mathrm{CV}_Y^2 - \mathrm{CV}_\alpha^2 - \frac{1}{\mu_Y}$$

from which we can estimate CV_λ^2 as both Y_{cg} and α_c are observable. To compute the Gini coefficient of the normalized counts, we used the R package reldist (10).

To quantify the relationship between cell size and expression burstiness in the RNA FISH data, we use the GAPDH read counts as the proxy of cell sizes, as experiments show that they are highly linearly correlated with the true cell size (11). Thus, we have 23 genes left for further analysis (excluding VCF, FOSL1 and GAPDH). We run a logistic regression to estimated the linear relationship between the log of cell size and the odds ratio of the nonzero fraction. Also, we use R packages gamlss, gamlss.tr (12, 13) and run zero-truncated negative binomial models to estimate the linear relationship between the log of cell size and the log of the nonzero mean for the RNA FISH data.

Analysis of the data from Tung et al. (7). The data in Tung et al. (7) contains three C1 replicates from three human induced pluripotent stem cell lines and UMI were added to all samples. In the original paper, one replicate of the first individual (NA 19098.r2) was removed from the data due to low quality and 564 cells are kept after filtering. Of the 564 cells, the number of cells in each of the eight replicates ranges from 51 to 85. For comparison between the two artificial groups, each group contains randomly selected 50 cells from one replicate of each of the three individuals. For comparison between individuals, we choose the two individuals whose all three replicates are kept and include all cells belonging to them in DESCEND.

Each replicate is a batch. As the batches are perfectly confounded with both the artificial group and the individual labels, we are unable to adjust for the confounding effects in mean expression. However, by adding the batch indicators as covariates, we can remove the batch biases within each testing group, thus can adjust for the confounding effect of batches on the dispersion parameters, such as CV and Gini coefficients.

We apply DESCEND to recover the relative gene expression and add the batches as covariates on the nonzero mean. Because of the limited number of cells in each group, we only look at the most highly expressed 187 genes whose average UMI read counts exceed 50 in order to make the estimation errors manageable. As these are highly expressed genes, most of their CV/Gini coefficients are also very small. In addition, as most of these genes are not bursty, there is no apparent difference when the batches are added as covariates on both the nonzero mean and nonzero fraction.

Analysis of the data from Klein et al. (5). For the mouse embryonic stem cell data from Klein et al. (2015), the single cells are sampled from a differentiating mESC population before and at $2, 4, 7$ days after LIF withdrawal. The number of cells sampled at each of the four times are 933, 303, 683 and 798 with the average library size being approximately 29500, 8500, 4700 and 26500 respectively. When comparing across all four datasets, we only keep genes whose fraction of non-zero read counts are at least 5% and whose average UMI read count are at least 0.15 in every dataset, resulting in 9059 left. For the differential analysis between Day 0 and Day 2, we ask the above criteria to be satisfied only in the two datasets involved, resulting in 13096 genes left for analysis.

In addition to using DESCEND for differential testing of the mean relative expression between Day 0 and Day 2, we also use the R package DESeq2 (14) with default settings. For the GO over-representation analysis of this dataset, we use the R package gProfileR (15). For the tSNE plot of the differentiation at Day 4 and Day 7, we use the R package SeuratV2.1 (16) and follow the standard work flow on their online tutorial: http://satijalab.org/seurat/pbmc3k_tutorial.html.

Analysis of the data from Zeisel et al. (6). The dataset contains read counts of 12234 genes in 3005 cells obtained from the mouse somatosensory cortex and hippocampus CA1 region. The 3005 cells have been further clustered into 7 major cell types: Astrocytes-Ependymal, Endothelial-Mural, Interneurons, Microglia, Oligodendrocytes, CA1 pyramidal and S1 pyramidal, and the number cells in each cell type are 224, 235, 290, 98, 820, 939 and 399 respectively. We recover the cell size adjusted gene expression distribution for each cell type separately. The cell sizes are estimated as the ratio between the library size and the cell efficiency estimated from the ERCC spike-ins. To avoid estimation bias, both the cell efficiency and cell size are treated as covariates on both nonzero fraction and nonzero mean.

For each cell type, we only recover the expression distribution of the genes whose fraction of non-zero read counts are at least 5% and whose average UMI read count are at least 0.3 for estimating the burstiness parameters with acceptable accuracy. The number of genes kept are then 3855, 3496, 7984, 3299, 4951, 7866 and 7683 respectively for each cell type. When we compare across cell types, we only consider the intersection of the genes from the involved cell types. For example, there are 2105 genes which are kept in all cell types.

For the GO over-representation analysis of this dataset, we use the R package clusterProfiler (17) which allows user-defined list of the background genes. We define the list of background genes to be all genes who passes our filtering criteria to avoid possible biases introduced during the filtering process.

Cell type identification. For cell clustering of the datasets from Zeisel et al. (6) and Zheng et al. (8), we follow the steps in the online tutorial of Seurat (Version 2.1). We try different values of the resolution parameter to get the desired number of clusters ($K = 7$ for (6) and $K = 10$ for (8)) in each scenario and compare the clustering accuracy of Seurat with and without DESCEND at the given number of clusters. To compute the adjusted Rand index, we use the R package mclust (18).

Mathematical Details of DESCEND

In this section, we elaborate on some of the statistical details underlying the DESCEND method with a more mathematically rigorous language than the method section of the manuscript. The methods build on the ideas in (19), and it may also be helpful to read that paper in order to understand the technical details of DESCEND.

Model Specification. As we introduced in the "Model Overview" section of the main manuscript, the observed read count Y_{cg} for gene g in cell c is modeled as a convolution of the true gene expression λ_{cg} and independent technical noise:

$$Y_{cg} \sim F_{cg}(\lambda_{cg}) \quad [1]$$

where $F_{cg}(\cdot)$ quantifies the technical variations of the noise. The technical noise distribution F_{cg} aims to capture reverse transcription loss, PCR amplification bias, sequencing loss, batch effects and other library preparation biases. Below, we first describe the possibilities for F_{cg} and then discuss modeling the biological variation of the true gene expression λ_{cg}.

The model for the technical noise. For scRNA-seq with UMI barcoding, the observed data are the counts of UMI barcodes for each gene in each cell. For this type of data, DESCEND currently allows two classes of technical noise models: the "Poisson-alpha" model and the "NB-alpha" model. The Poisson-alpha model assumes

$$Y_{cg} \sim \text{Poisson}(\alpha_{cg}\lambda_{cg}) \quad [2]$$

where α_{cg} is a cell- and gene-specific efficiency constant. For most data sets in our paper, with the exception of (7), we allowed the simplification $\alpha_{cg} = \alpha_c$. Here, we will discuss the general model.

An analytic argument for model Eq. (2) was given by (20), which we recapitulate here. scRNA-seq technology includes three main steps: reverse transcription, then PCR amplification, followed by sequencing. In the first step, the number of copies W_{cg} of gene g in cell c should approximately follow

$$W_{cg} \sim \text{Binomial}(\lambda_{cg}, p_{1cg})$$

where p_{1cg} is the probability of a transcript molecule being reverse transcribed. The amplification step amplifies W_{cg}, but with UMI barcoding, the amplification noise can be ignored. The sequencing step following amplification takes a sample from the UMI barcodes in the library, and thus, the observed UMI barcode count Y_{cg} follows

$$Y_{cg} \sim \text{Binomial}(W_{cg}, p_{2cg})$$

where p_{2cg} is the probability of a transcript being amplified then sequenced. Let $\alpha_{cg} = p_{1cg}p_{2cg}$, then, since the three steps are independent, we have

$$Y_{cg} \sim \text{Binomial}(\lambda_{cg}, \alpha_{cg})$$

which is approximately model Eq. (2). The substitution of the Poisson for the Binomial distribution is reliable when α_{cg} is small, which is true for most scRNA-seq data sets. For large α_{cg}, Y_{cg} under the Binomial distribution should be slightly less dispersed than under the Poisson distribution, and using the Poisson distribution would lead to more conservative inference, which we feel is not a bit issue.

DESCEND can accommodate four forms for α_{cg}:

- constant across genes for each cell: $\alpha_{cg} \equiv \alpha_c$
- linear for each gene across cells: $\alpha_{cg} \propto \alpha_c$ or more specifically, $\alpha_{cg} = \alpha_c\gamma_g$
- log-linear for each gene across cells: $\alpha_{cg} = \alpha_c^{\gamma_g}$ as suggested in (21)
- batch-correction form: $\alpha_{cg} = \alpha_c \exp(U_c^b\beta_{bg})$ where U_c^b is the batch indicator covariates for cell c.

The default form for α_{cg} in DESCEND is the constant form. If the linear form is assumed, the distribution of $\gamma_g\lambda_{cg}$ can still be deconvolved by DESCEND. Though γ_g is unidentifiable, one can still recover scale-invariant properties of the true gene expression distribution, such as nonzero fraction, Gini or CV. The log-linear form allow gene-specific efficiency effects and the batch-correction form allows correction of batch effects on each gene differently. The log-linear and the batch-correction forms look complicated, but, as we shall see, the coefficients in both models can be estimated in DESCEND by treating α_c or U_c^b as covariates. DESCEND also accepts the estimated α_c or γ_g computed by other software, such as (21).

As described in the main manuscript, the constant/ linear forms of the "Poisson-alpha" model capture most of the random variation in ERCC spike-in genes for the 9 public UMI datasets we have examined. It is important to note that the cell-specific α_c factor is able to account for noise introduced in multiple stages of the experiment, including reverse transcription loss sequencing error and cell-level average batch effects, although the adjustment is not gene-specific. In the main manuscript, we also showed using the data from Tung (2017) that the batch-correction form is useful for removing batch effects for biological genes. The appropriate model choice should always be made in consideration of the specific protocol that generated the data.

The efficiency constants α_{cg} discussed above is slightly different from the scaling constant α_c discussed in the "Model Overview" and "Methods" sections of the main manuscript. There, the more general scaling constant α_c either represent the efficiency as described here or the efficency times the total number of mRNA copies in the cell. With gene and cell specific

constant α_{cg}, we from now on also allow for such generalization, where α_{cg} is either the efficiency, or efficiency times the total number of mRNAs in cell c. For the latter scenario, λ_{cg} is then intepreted as the true expression concentration (or relative expression) instead of the true absolute expression levels. We will discuss in detail in the model estimation section how such generalization enables estimation of the technical noise model without presence of ERCC spike-ins.

Another type of technical noise model allowed by DESCEND is the NB-alpha model, which extends the Poisson-alpha model by assuming

$$Y_{cg} \sim \text{Negative Binomial}(\alpha_{cg}\lambda_{cg}, \theta),$$

where the over-dispersion parameter θ quantifies possible over-dispersion beyond Poisson. We have shown on ERCC spike-ins that θ is almost 0 in most datasets, and thus ignore this small level of over-dispersion. Yet, it is possible that some datasets can have relatively large over-dispersion due to unknown sources of technical variation.

We believe carefully done spike-ins are invaluable for checking the appropriateness of the noise model for any scRNA-seq experiment. Note, however, that in DESCEND, if deconvolving the relative expression distribution, spike-ins are used only for model checking and not for the estimation of any noise parameters. Thus, even in the absence of reliable spike-ins, one can still recover the distribution for relative expression.

The model for the true gene expression. As explained in the main manuscript, and in the previous subsection, we allow λ_{cg} to represent either the absolute or relative expression. Hereon, we refer to λ_{cg} as true expression, with its interpretation implicit given by the choice of the scaling constants α_{cg}.

First, consider the scenario when no covariates are present. We denote the distribution of λ_{cg} as

$$\lambda_{cg} \sim H_g(\cdot)$$

where the distribution H_g is modeled as an exponential family distribution with density:

$$h(x) = \exp\{Q(x)^T\alpha - \phi(\alpha)\} \quad [3]$$

where α is a vector of parameters and $\phi(\alpha)$ is the normalization factor. We do not use a simple parametric form for $Q(x)$, but set it as a 5-degree natural cubic spline function, thus allowing more flexibility to adapt to the unknown distribution of true gene expression. We discretize the distribution to simplify computation. As discussed in (19), we limit the possible values of true expression to $x \in \boldsymbol{x} = (x_1, \cdots, x_m)$. Then, for a random viariable X following the discretized form of H_g,

$$\mathbb{P}[X = \boldsymbol{x}] = \exp\{Q^T\alpha - \phi(\alpha)\} \quad [4]$$

where Q is the 5-degree natural cubic spline matrix at $\boldsymbol{x}$.

Due to transcriptional bursting and cell subpopulation heteogeneity, the true expression / concentration can have a point mass at 0, which we model separately. Similar to what is suggested by (19), to incorporate the point mass at 0, we define $Q_1 = \begin{pmatrix} 1 & 0 \\ 0 & Q \end{pmatrix}$. Let $\boldsymbol{x}_0 = (0, \boldsymbol{x})$ (the original discretization $\boldsymbol{x}$ should not include 0), the final discretized distribution H_g is

$$\mathbb{P}[X = \boldsymbol{x}_0] = \exp\{Q_1^T \begin{pmatrix} \alpha_0 \\ \alpha \end{pmatrix} - \phi_1(\alpha_0, \alpha)\} \quad [5]$$

where $\phi_1(\cdot)$ is the new normalization factor. The probability at 0 is simply $\exp(\alpha_0 - \phi_1(\alpha_0, \alpha))$, and thus can be an arbitrary value regardless of the shape of the positive part. In practice the points x_0 are chosen as equally spaced points between 0 and some quanitle (say 98%) of the scaled observed counts.

Now, consider the case where covariates are present, and the goal is to conduct inference on the effect of the covariates and, perhaps, also examine the covariate adjusted distribution, as defined in the Materials and Method section of the main manuscript. The covariates here are at the cell level, such as cell size and cell cycle. We allow the covariates U_c to affect both the zero fraction of the distribution and the mean of the positive part (the nonzero mean). We model these two effects separately, using the logit link for the former and the log link for the latter:

$$\text{logit}(\mathbb{P}[\lambda_{cg} = 0]) = U_c\widetilde{\beta}_g + \widetilde{\beta}_{0g}, \quad [6]$$

$$\log(\lambda_{cg})\big|_{\lambda_{cg}>0} = U_c\beta_g + \epsilon_{cg}, \quad [7]$$

where β_g, $\tilde{\beta}_g$ and $\tilde{\beta}_{0g}$ are parameters to be estimated from the counts for each gene across cells. The random variable $\exp(\epsilon_{cg})$ is assumed to have the distribution of Eq. (4). Interpretation of the deconvolution result under covariate adjustment is discussed in the Methods section of the main manuscript.

Model Estimation.

Estimating the technical noise. First, consider the Poisson-alpha model under the constant / linear form. With the presence of ERCC spike-ins, the cell specific efficiency constants can be estimated using ERCC spike-ins as

$$\widehat{\alpha}_{cg} \equiv \widehat{\alpha}_c = \sum_{g \text{ is spike-in}} Y_{cg} / \sum_{g \text{ is spike-in}} \lambda_g$$

where $\lambda_g = \mathbb{E}[\lambda_{cg}]$ is the known expected input molecule count of the spike-ins computed from dilution ratios. For the constant form, we have $\mathbb{E}[\widehat{\alpha}_{cg}] = \alpha_c = \alpha_{cg}$. For the linear form, we have

$$\mathbb{E}[\widehat{\alpha}_{cg}] = \alpha_c \frac{\sum_{g' \text{ is a spike-in}} \lambda_{g'}\gamma_{g'}}{\sum_{g' \text{ is a spike-in}} \lambda_{g'}} = \alpha_{cg} \frac{1}{\gamma_g} \frac{\sum_{g' \text{ is a spike-in}} \lambda_{g'}\gamma_{g'}}{\sum_{g' \text{ is a spike-in}} \lambda_{g'}}$$

which is proportional to the true α_{cg}. Subsequently, we simply plug-in these estimated $\hat{\alpha}_{cg}$ as true values for next steps in calculating the likelihood of observed counts. As mentioned, this approach leads to the interpretation of λ_{cg} being absolute gene expression counts.

When there are no ERCC spike-ins, we are unable to estimate cell-specific efficiency values. However, letting N_c be the total RNA count in each cell, we can still estimate the product of N_c and the efficiencies under the constant / linear form Poisson-alpha model to estimate our generalized definition of α_{cg} in the previous section. We estimate α_{cg} by simply using the library size

$$\widehat{\alpha}_{cg} \equiv M_c = \sum_g Y_{cg}.$$

This is enough for deconvolution of the distribution of relative expression levels since

$$Y_{cg} \sim \text{Poisson}(\text{efficiency of cell } c \text{ gene } g \times N_c \times \text{concentration of } g \text{ in cell } c).$$

following the noise model Eq. (2).

For the batch-correction form of the Poisson-alpha model where $\alpha_{cg} = \alpha_c \exp(U_c^b \beta_{bg})$, we can estimate the α_c part the same as described above. The ratio between $\mathbb{E}[\widehat{\alpha}_c]$ and the true α_c is then batch-specific. Both the effect of the difference of such ratio between batches the batch-specific coefficient β_{cg} can be corrected by adding U_c^b as a covariate in DESCEND.

For the "NB-alpha model", the cell-specific efficiencies can be estimated in the same way as in the Poisson-alpha model. The over-dispersion parameter θ can be estimated only with spike-ins, although the spike-ins do not need to be added to every cell. For example, in droplet-based sequencing protocols, a common approach is to create droplets containing only spike-ins, which can be used to estimate the over-dispersion associated with purely technical noise. The estimation of the dispersion parameter is as follows: First, the Poisson-alpha model is used to deconvolve the distribution of λ_{cg} for spike-ins; then, the coefficient of variation of the deconvolved distribution is computed. The parameter θ can be estimated by the squared limiting CV values of the spike-in genes whose λ_g are large. This is based on the fact that when

$$Y_{cg} \sim \text{Negative Binomial}(\alpha_{cg}\lambda_{cg}, \theta) \sim \text{Poisson}(\alpha_{cg} Z_{cg}),$$

$$Z_{cg} \sim \text{Gamma distributed}, \mathbb{E}[Z_{cg} \mid \lambda_{cg}] = \lambda_{cg}, \text{Var}[Z_{cg} \mid \lambda_{cg}] = \theta\lambda_{cg}^2,$$

using the "Poisson-alpha" model for deconvolution recovers the distribution of Z_{cg}, and the squared CV of Z_{cg} equals

$$\text{CV}^2 = \frac{\text{Var}[Z_{cg}]}{\lambda_g^2} = \frac{\lambda_g + \theta(\lambda_g^2 + \lambda_g)}{\lambda_g^2} \overset{\lambda_g \to \infty}{\longrightarrow} \theta.$$

Estimating the parameters of the underlying biological distribution. We only show here how the covariate-adjusted (including batch effect adjusted) DESCEND model can be fit. The no covariates scenario is easier and can be derived from the results here in a straightforward way. The estimation procedure for technical covariates (such as batches) and biological covariates (such as cell sizes) are the same. For notation simplicity, we use U_c to denote covariates, either technical or biological or both, and assume λ_{cg} follows Eq. (6) and Eq. (7). Note that λ_{cg} here may contain the technical covariates, thus is no longer the true absolute /relative expression levels. We still use λ_{cg} for notation simplification.

Let $T_{cg} = \lambda_{cg} / \exp(U_c \beta_g)$, then T_{cg} has density

$$\mathbb{P}[T_{cg} = \boldsymbol{x}_0] \triangleq \boldsymbol{h}_c(\tilde{\alpha}_g) = \exp\{Q_c^T \tilde{\alpha}_g - \tilde{\phi}_c(\tilde{\alpha}_g)\} \quad [8]$$

where each $Q_c = \begin{pmatrix} 1 & 0 \\ U_c^T & 0 \\ 0 & Q \end{pmatrix}$ is the cell specific covariate adjusted matrix. To see that Eq. (8) is consistent with Eq. (6), let $\tilde{\alpha} = \begin{pmatrix} \alpha_0 \\ \alpha_U \\ \alpha \end{pmatrix}$ then

$$\text{logit}(\mathbb{P}[T_{cg} = 0]) = \log\left(\frac{\mathbb{P}[T_{cg} = 0]}{\mathbb{P}[T_{cg} \neq 0]}\right) = \alpha_0 - \log\left[\left(\exp\left(Q^T\alpha\right)\right)^T 1\right] + U_c\alpha_U$$

which is equivalent to Eq. (6) with $\tilde{\beta}_g = \alpha_U$ and $\tilde{\beta}_0 = \alpha_0 - \log\left[\left(\exp\left(Q^T\alpha\right)\right)^T 1\right]$. For notation simplicity, we use α to denote the whole parameter vector $\tilde{\alpha}$ in later formulas.

Now we focus on a single gene g and omit the subscript g for notation simplification in ensuing formulas. Let the probability of observing read count Y_c be f_c. Denote $\boldsymbol{p}_c(\beta) = \mathbb{P}\left[Y_c \mid T_c = \boldsymbol{x}_0\right]$, then we have

$$f_c(\alpha, \beta) = \boldsymbol{p}_c(\beta)^T \boldsymbol{h}_c(\alpha)$$

The log-likelihood of the data is $l(\alpha, \beta) = \sum_c \log f_c(\alpha, \beta)$.

As suggested in (19), we maximize a penalized log-likelihood

$$\tilde{l}(\alpha, \beta) = l(\alpha, \beta) - s(\alpha), \quad s(\alpha) = c_0\|\alpha\|_2,$$

where the penalty term $s(\alpha)$ reduces the variance of the estimates by sacrificing some unbiasness. Let the Fisher information matrix of α be $I(\alpha)$, which can be estimated from the unpenalized likelihood. The regularization constant c_0 is selected iteratively and adaptively such that the approximate ratio of artificial to genuine information

$$R(\widehat{\alpha}) = \frac{\mathrm{tr}\{\ddot{s}(\widehat{\alpha})\}}{\mathrm{tr}\{\widehat{I}(\widehat{\alpha})\}}$$

is less than 1%, to avoid over shrinkage bias, but more than 0.05%, to reduce estimation variation. As shown in (19), there are relatively simple formulas for $\dot{\tilde{l}}(\alpha, \beta)$ and $\ddot{\tilde{l}}(\alpha, \beta)$ due to the exponential family framework. The optimization of $\tilde{l}(\alpha, \beta)$ is conducted by a Newton-type algorithm using the nlm function in R.

Statistical inference. (19) showed that second-order approximation formulas provide useful inference on $\widehat{\alpha}$ and $\widehat{\beta}$. By second-order approximation uses Taylor expansion around the true value α_* and β_*,

$$0 = \dot{\tilde{l}}(\widehat{\alpha}, \widehat{\beta}) \approx \dot{\tilde{l}}(\alpha_*, \beta_*) + \ddot{\tilde{l}}(\alpha_*, \beta_*)\left\{\begin{pmatrix}\widehat{\alpha}\\ \widehat{\beta}\end{pmatrix} - \begin{pmatrix}\alpha_*\\ \beta_*\end{pmatrix}\right\},$$

which yields

$$\begin{pmatrix}\widehat{\alpha}\\ \widehat{\beta}\end{pmatrix} - \begin{pmatrix}\alpha_*\\ \beta_*\end{pmatrix} = \ddot{\tilde{l}}(\alpha_*, \beta_*)^{-1}\dot{\tilde{l}}(\alpha_*, \beta_*).$$

Considering possible model mis-specification, we use the sandwich estimator. Define

$$A = -\ddot{\tilde{l}}(\widehat{\alpha}, \widehat{\beta}), \;\; B = \sum_c \dot{l}_c(\widehat{\alpha}, \widehat{\beta})\dot{l}_c(\widehat{\alpha}, \widehat{\beta})^T,$$

where $l_c(\alpha, \beta) = \log f_c(\alpha, \beta)$. Then, approximately,

$$\begin{pmatrix}\hat{\alpha} - \alpha_*\\ \hat{\beta} - \gamma_*\end{pmatrix} \dot{\sim} \left(-A^{-1}\dot{s}(\widehat{\alpha}), A^{-1}BA^{-1}\right).$$

Often we are interested in conducting inference on quantities that are functions of α, such as the population mean μ, the nonzero fraction $1 - p_0$, CV and Gini coefficients, we use second order approximations again, in that for any function $f(\cdot)$, Taylor expansion yields $f(\widehat{\alpha}) \approx f(\alpha) + \dot{f}(\alpha)(\widehat{\alpha} - \alpha)$. As a consequence, we get

$$\mathrm{Var}\left[f(\widehat{\alpha})\right] \approx \dot{f}(\widehat{\alpha})^2 \,\mathrm{Var}\left[\widehat{\alpha}\right], \;\; \mathrm{Bias}(f(\widehat{\alpha})) \approx \dot{f}(\widehat{\alpha})\mathrm{Bias}(\widehat{\alpha}).$$

The second order approximation may not be accurate because of regularization. In practice, this approximation tends to underestimate the bias of the estimates (see (19)). However, we find that ,empirically, the mean squared error: $\widehat{\mathrm{MSE}}(\hat{\theta}_i) = \widehat{\mathrm{Bias}}^2(\hat{\theta}_i) + \widehat{\mathrm{Sd}}^2(\hat{\theta}_i)$ obtained by combining the variance and bias gives meaningful quantification of the estimation uncertainty.

We further use the mean squared error for differential testing of p_0, CV and Gini Coefficients between two populations of cells. More specifically, to test $H_0 : \theta_1 = \theta_2$ where θ_i $(i = 1, 2)$ is some parameter expressed as a function $f(\alpha)$ as above, we compute the following Z-score:

$$Z = \frac{\hat{\theta}_1 - \hat{\theta}_2}{\sqrt{\widehat{\mathrm{MSE}}(\hat{\theta}_1) + \widehat{\mathrm{MSE}}(\hat{\theta}_2)}}$$

We run a permutation of the cell labels of the two populations to compute p-values for the Z-scores. Let Y_1 and Y_2 be two count matrices of two cell populations. We randomly re-assign the cells to each population to get new shuffled count matrices $Y_1^\star$ and $Y_2^\star$ which have the same dimensions as Y_1 and Y_2 respectively. We compute the Z-scores $Z^\star$ comparing the two permuted data matrices $Y_1^\star$ and $Y_2^\star$ for all the genes and use the distribution of $Z^\star$ across genes as the null distribution of the z-score. The p-values are then computed by calculating the quantiles of the z-scores with respect to this null distribution.

Again, as the second order approximations may not be accurate, we use likelihood ratio tests to test for specific values of the parameters in a single population, e.g. $H_{01g} : \mathbb{P}\left[\lambda_{cg} \neq 0\right] = 1$, the test for nonzero fraction in the true distribution of gene g. We calcuate the maximized unpenalized likelihoods $\hat{l}_{0g}$ and $\hat{l}_{1g}$ under the null and alternative hypotheses, respectively. The likelihood ratio test for gene g is $\hat{T}_g = 2[\hat{l}_{1g} - \hat{l}_{0g}]$. The same applies to the test of whether cell size has an effect on nonzero mean $(H_{02g} : \beta_g = 1)$, or on nonzero fraction $(H_{03g} : \tilde{\beta}_g = 0)$. Under the null hypothesis, these test statistics approximately follow the chi-squared distribution with one degree of freedom.

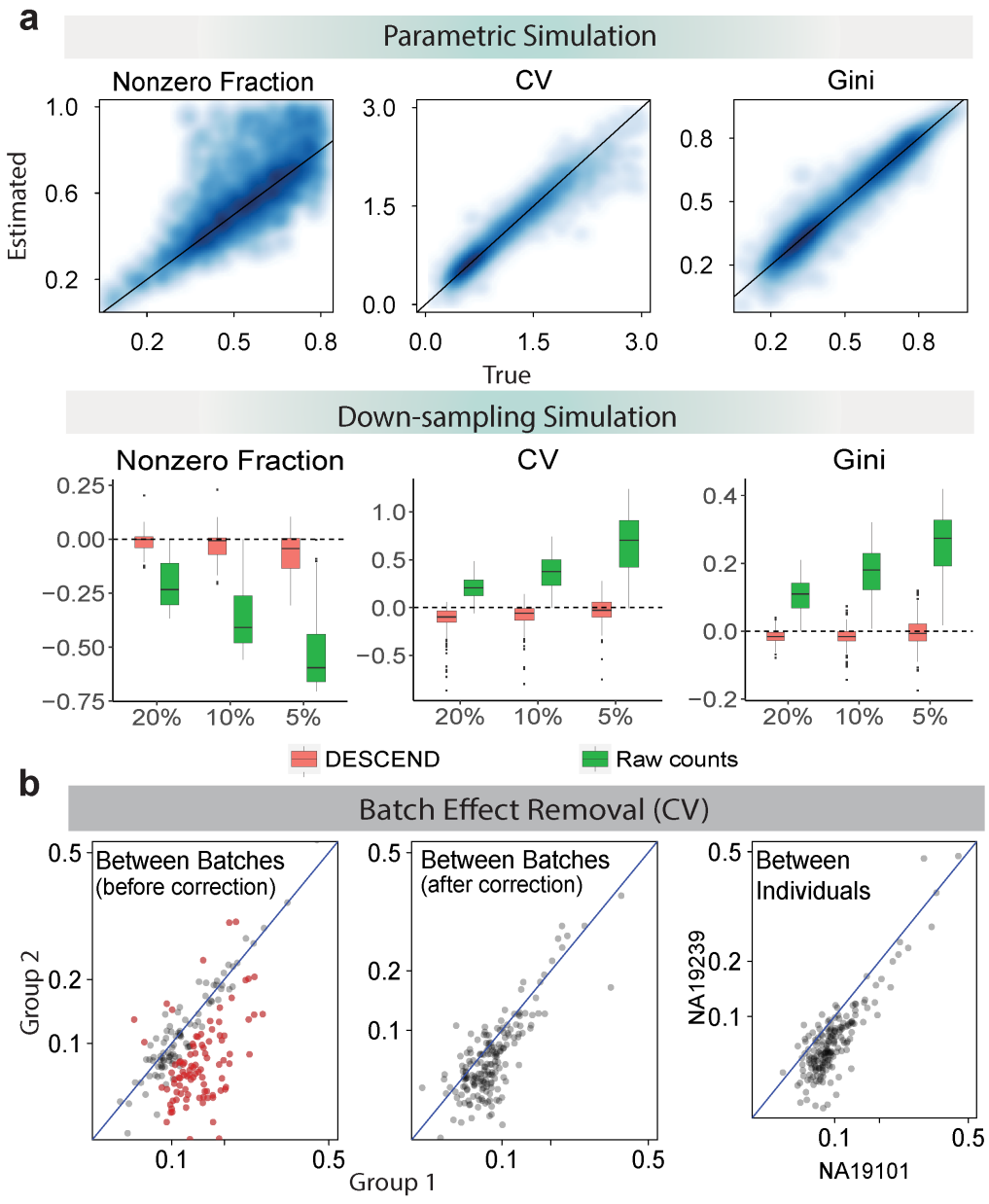

Fig. S1. More figures for validation of DESCEND. (**a**) For the parametric simulation: comparison between the estimated nonzero fraction, CV and Gini coefficients and the true values. For the down-sampling simulation, boxplots for values across genes where for each gene, the value is the difference between the estimated parameter value and the true value. (**b**)Batch effect removal: the DESCEND estimated CV coefficients are compared between the two artificial groups before (left) and after (middle) adding batches as covariates. Each dot is a gene. The estimated CV after adding batches as covariates are also compared between two individuals (right). The red dots are significantly differentially expressed genes (of CV) when FDR in controlled at level 5%.

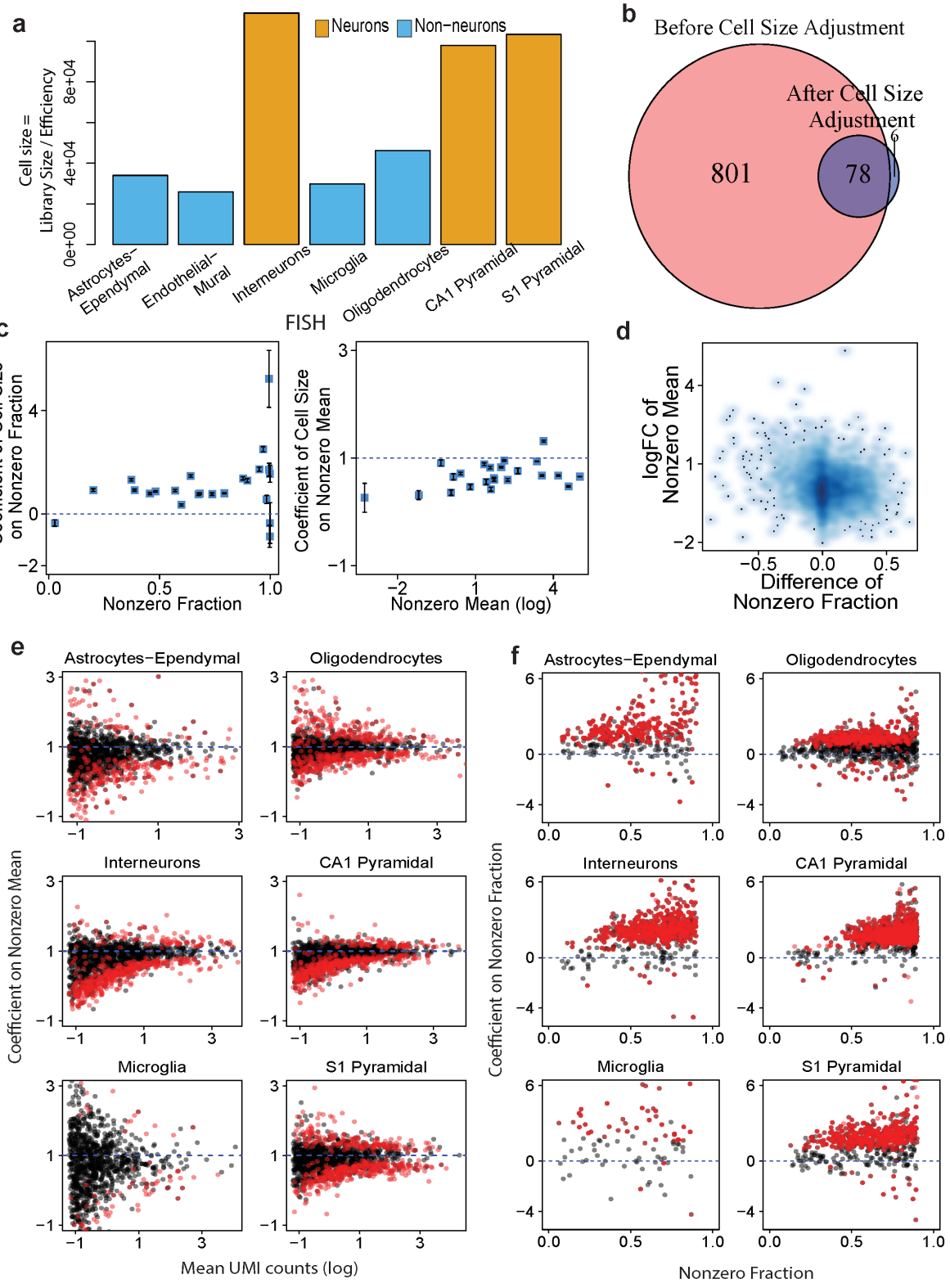

Fig. S2. Supplementary figures for the case study of Zeisel et al. (6). **(a)**Bar plot of the mean cell size (calculated as the ratio of library size and cell efficiency) of each cell type. Neuron cells are much larger than non-neuron cells. **(b)**Venn Diagram of the number of significantly differentially expressed genes based on nonzero fraction before and after cell size adjustment. **(c)**The coefficients of cell size on nonzero fraction and nonzero mean for the RNA FISH data. The black bars are one standard error bars. (**d**)Change of nonzero fraction versus the change of nonzero mean of genes between two cell types: Endothelial-Mural and CA1 Pyramidal. **(e)**A consistent sub-linear effect of cell size on nonzero mean across all cell types. **(f)**A consistent positive effect of cell size on nonzero fraction across all cell types.

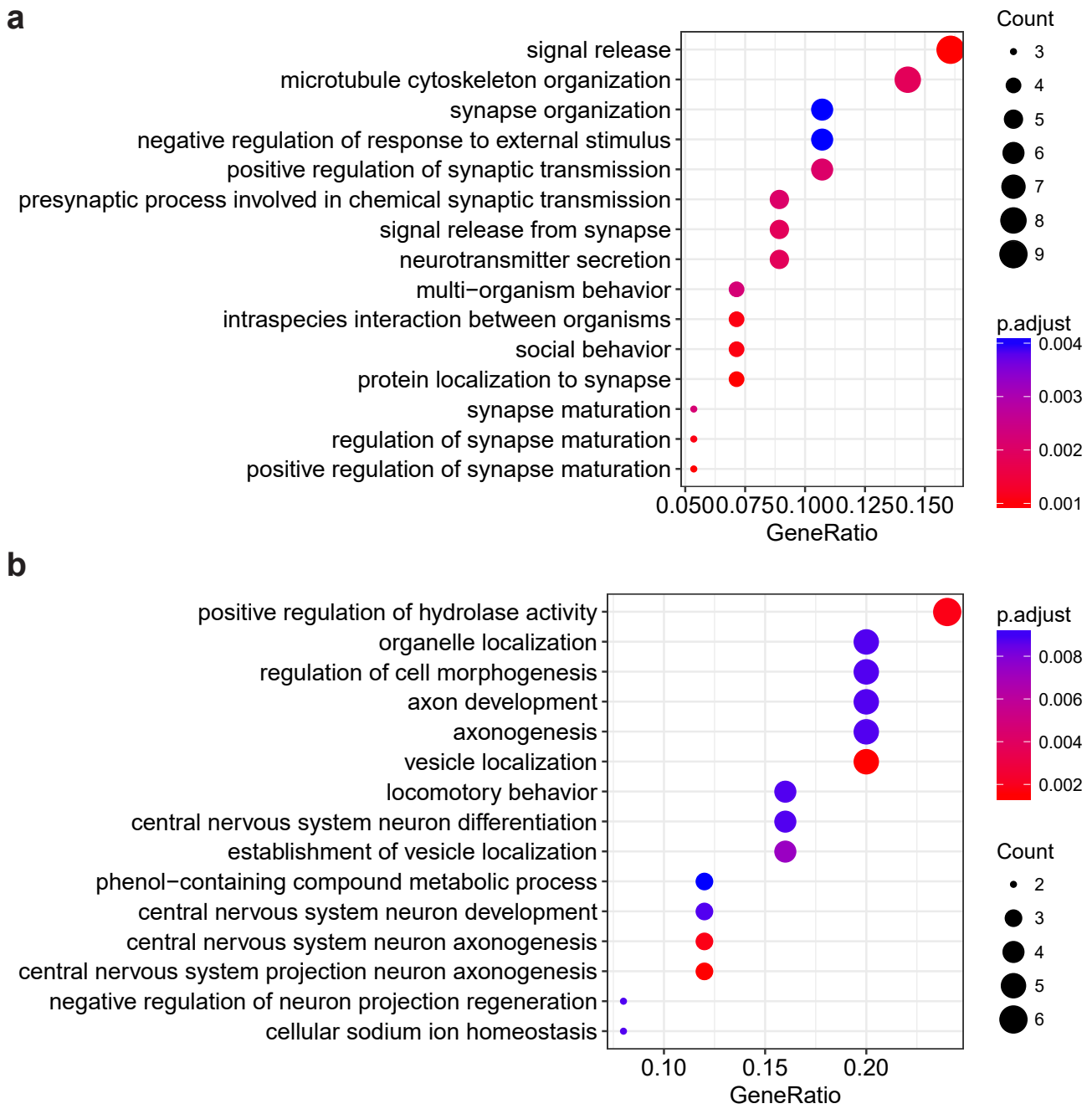

Fig. S3. Over-representation analysis for **(a)**genes whose nonzero fraction is significant differentially expressed but not the nonzero mean and **(b)**genes whose both nonzero fraction and nonzero mean are differentially expressed genes based on nonzero fraction between the non-neuron cell type Endothelial-Mural and the neuron cell type CA1 Pyramidal. The plot shows the 15 categories with the smallest p-values using Fisher's exact test.

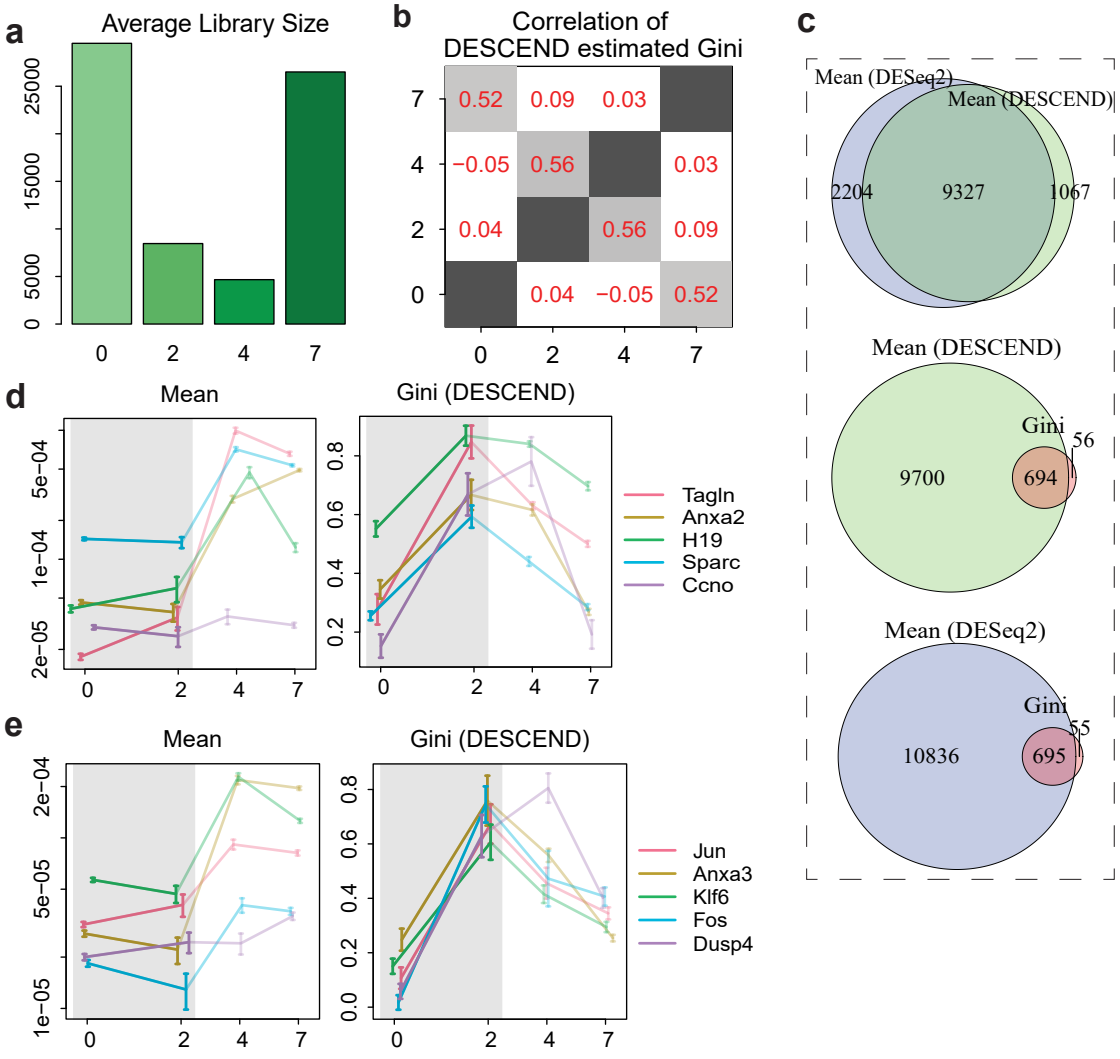

Fig. S4. Supplementary figures for the case study of Klein et al. (5). (**a**)Average library sizes for the cells at each day. (**b**)Correlation of the DESCEND estimated Gini coefficient between days. (**c**)Venn diagram of the number of differential expressed genes based on mean relative expression tested using DESeq2 and DESCEND; venn diagrams of the number of differentially expressed genes based on mean relative expresion and Gini coefficient, one for the case where the change are tested using DESeq2 and DESCEND, and one for the case where the change are tested both using DESCEND. FDR is controlled at 5%. (**d**)Change of the mean relative expression and Gini coefficients for 5 significant known marker genes for the various cell types that appear during differentiation. The colored bars are one standard error bars. (**e**)Change of the mean relative expression and Gini coefficients for 5 significant genes that are promising marker genes for the differentiated cell type.

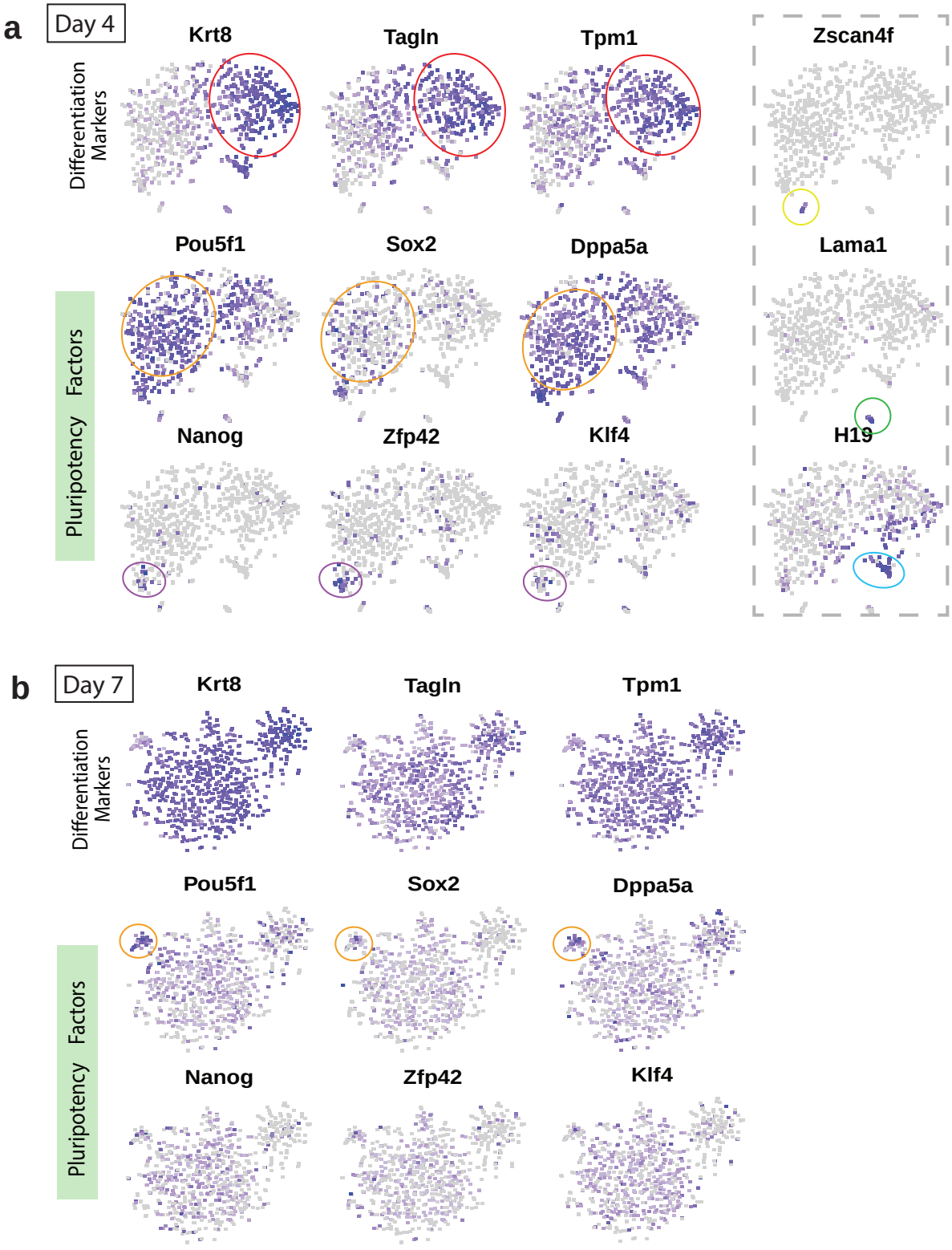

Fig. S5. Feature plots of the tSNE plots for the cells at day 4 and day 7. (**a**)The colored circles represents the differentiated cells (red), not fully differentiated cells (orange) and undifferentiated cells (purple). There are also three other cell clusters at Day 4 (yellow, green and blue circles). (**b**) The orange circled cells are those that have not fully differentiated at Day 7.

References

1. Jaitin DA, et al. (2014) Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* 343(6172):776–779.
2. Macosko EZ, et al. (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* 161(5):1202–1214.
3. Hashimshony T, et al. (2016) CEL-Seq2: sensitive highly-multiplexed single-cell rna-seq. *Genome biology* 17(1):77.
4. Svensson V, et al. (2017) Power analysis of single-cell RNA-sequencing experiments. *Nature methods*.
5. Klein AM, et al. (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* 161(5):1187–1201.
6. Zeisel A, et al. (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 347(6226):1138–1142.
7. Tung PY, et al. (2017) Batch effects and the effective design of single-cell gene expression studies. *Scientific reports* 7:39921.
8. Zheng GX, et al. (2017) Massively parallel digital transcriptional profiling of single cells. *Nature communications* 8:14049.
9. Torre EA, et al. (2017) A comparison between single cell RNA sequencing and single molecule RNA FISH for rare cell analysis. *bioRxiv* p. 138289.
10. Handcock MS (2016) *Relative Distribution Methods* (Los Angeles, CA). Version 1.6-6. Project home page at url-http://www.stat.ucla.edu/ handcock/RelDist.
11. Padovan-Merhar O, et al. (2015) Single mammalian cells compensate for differences in cellular volume and DNA copy number through independent global transcriptional mechanisms. *Molecular cell* 58(2):339–352.
12. Rigby RA, Stasinopoulos DM (2005) Generalized additive models for location, scale and shape,(with discussion). *Applied Statistics* 54:507–554.
13. Stasinopoulos M, Rigby B (2016) *gamlss.tr: Generating and Fitting Truncated 'gamlss.family' Distributions*. R package version 5.0-0.
14. Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology* 15(12):550.
15. Reimand J, Kolde R, Arak T (2016) *gProfileR: Interface to the 'g:Profiler' Toolkit*. R package version 0.6.1.
16. Satija R, Butler A, Hoffman P (2017) *Seurat: Tools for Single Cell Genomics*. R package version 2.1.0.
17. Yu G, Wang LG, Han Y, He QY (2012) clusterprofiler: an r package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology* 16(5):284–287.
18. Fraley C, Raftery AE, Murphy TB, Scrucca L (2012) *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*.
19. Efron B (2016) Empirical bayes deconvolution estimates. *Biometrika* 103(1):1–20.
20. Kim JK, Kolodziejczyk AA, Ilicic T, Teichmann SA, Marioni JC (2015) Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nature communications* 6:8687.
21. Bacher R, et al. (2017) Scnorm: robust normalization of single-cell rna-seq data. *Nature Methods* 14(6):584–586.