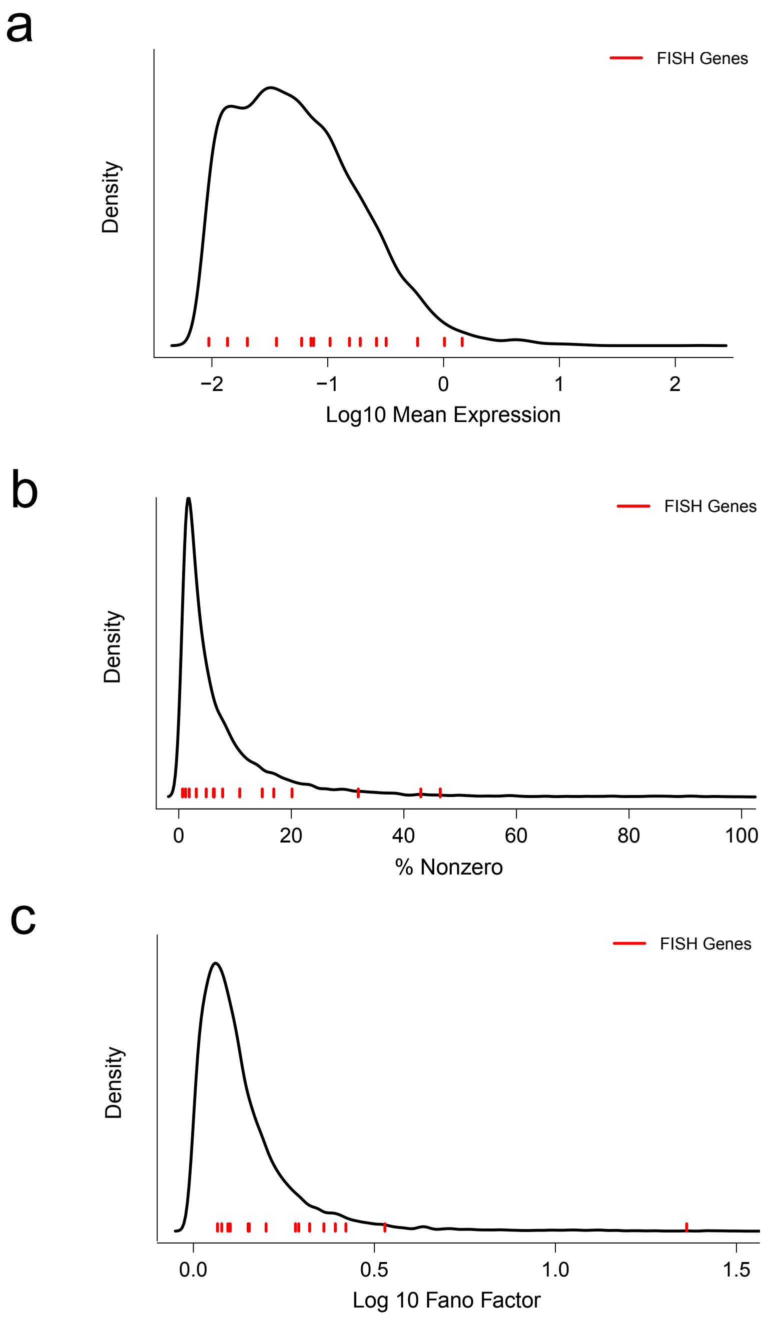
Contents

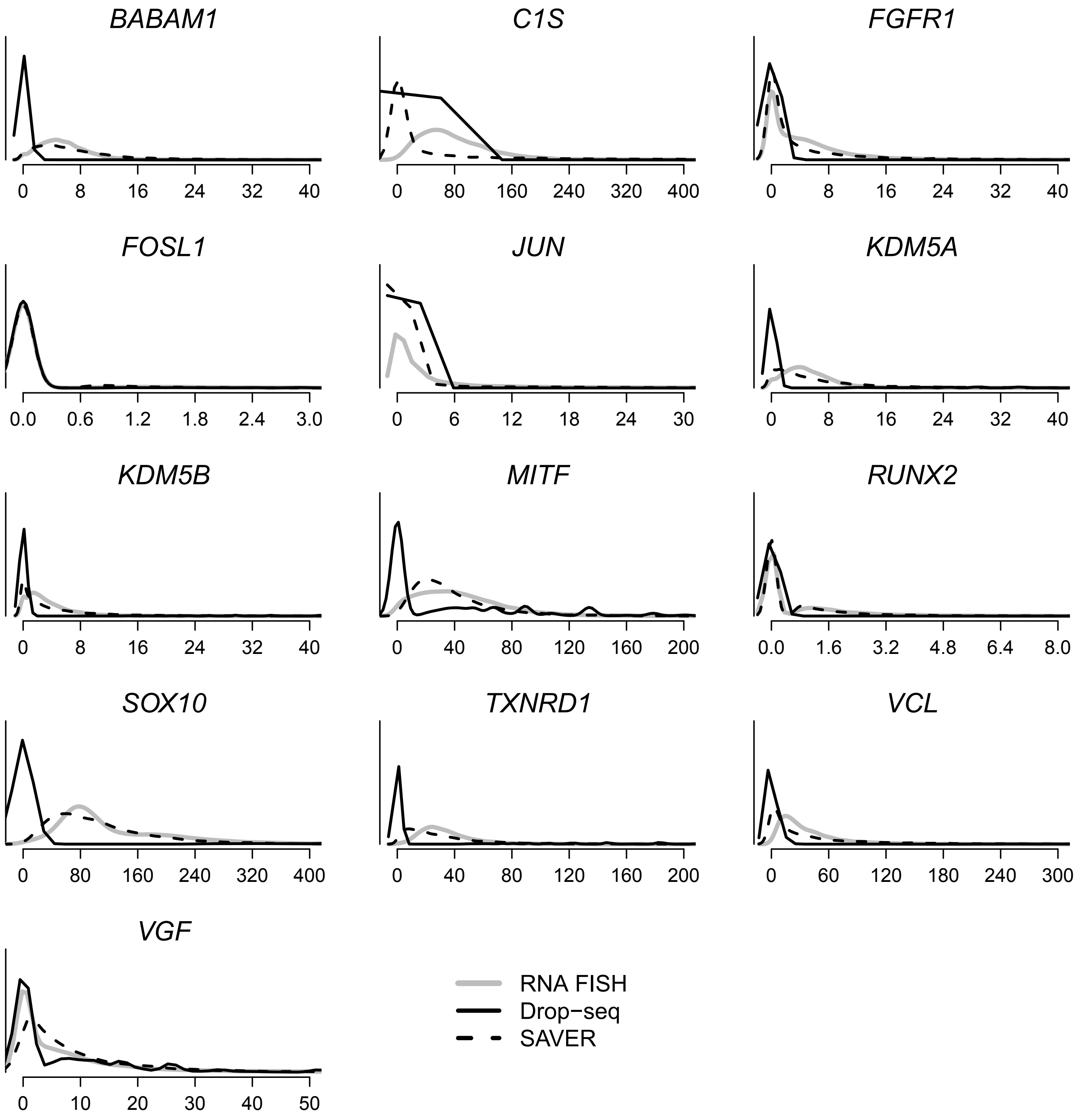
1 Supplementary Figures

2 Supplementary Tables

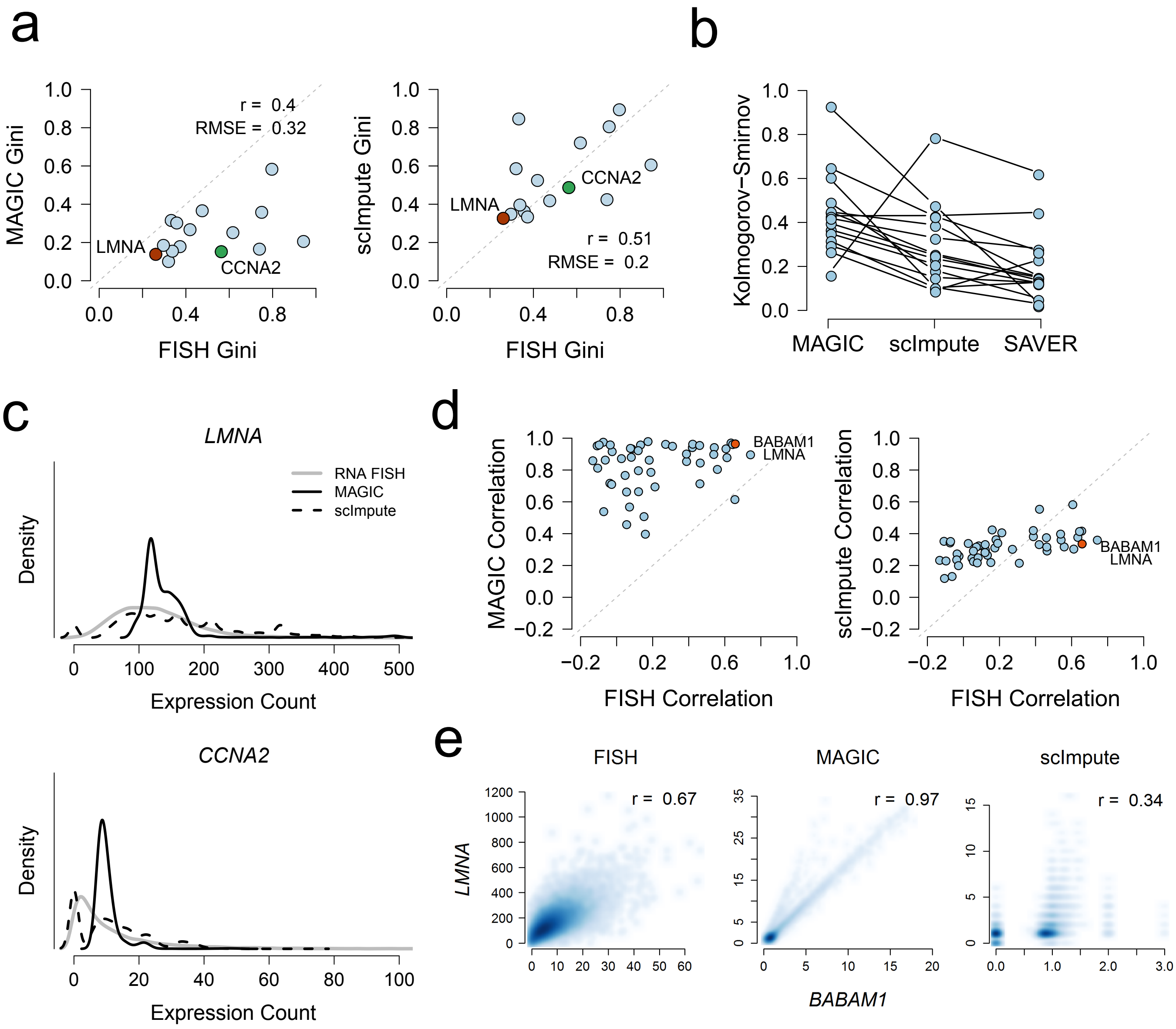
3 Supplementary Notes



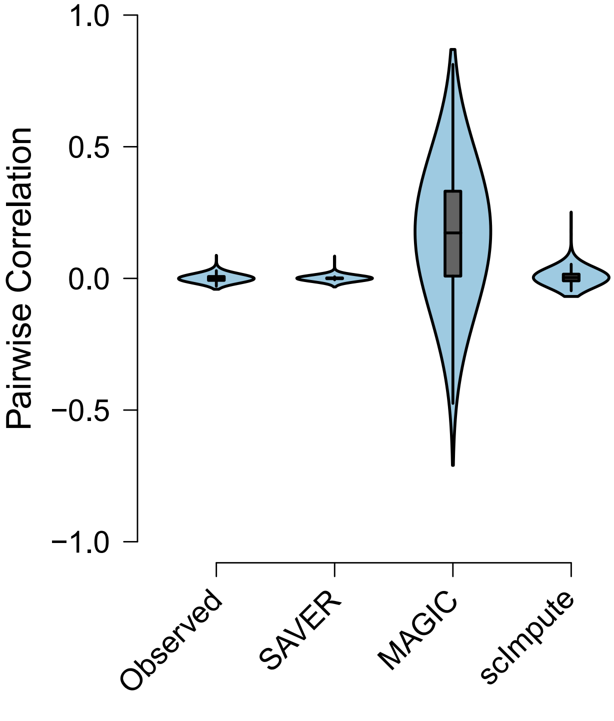
Supplementary Figure 1 Comparison of the 15 FISH genes with all genes in the Drop-seq data in terms of (**a**) mean expression, (**b**) percent of cells with non-zero expression, and (**c**) Fano factor, a measure of dispersion. The 15 FISH genes are representative of all genes for mean expression and percent non-zero, and have slightly higher dispersion than the population.



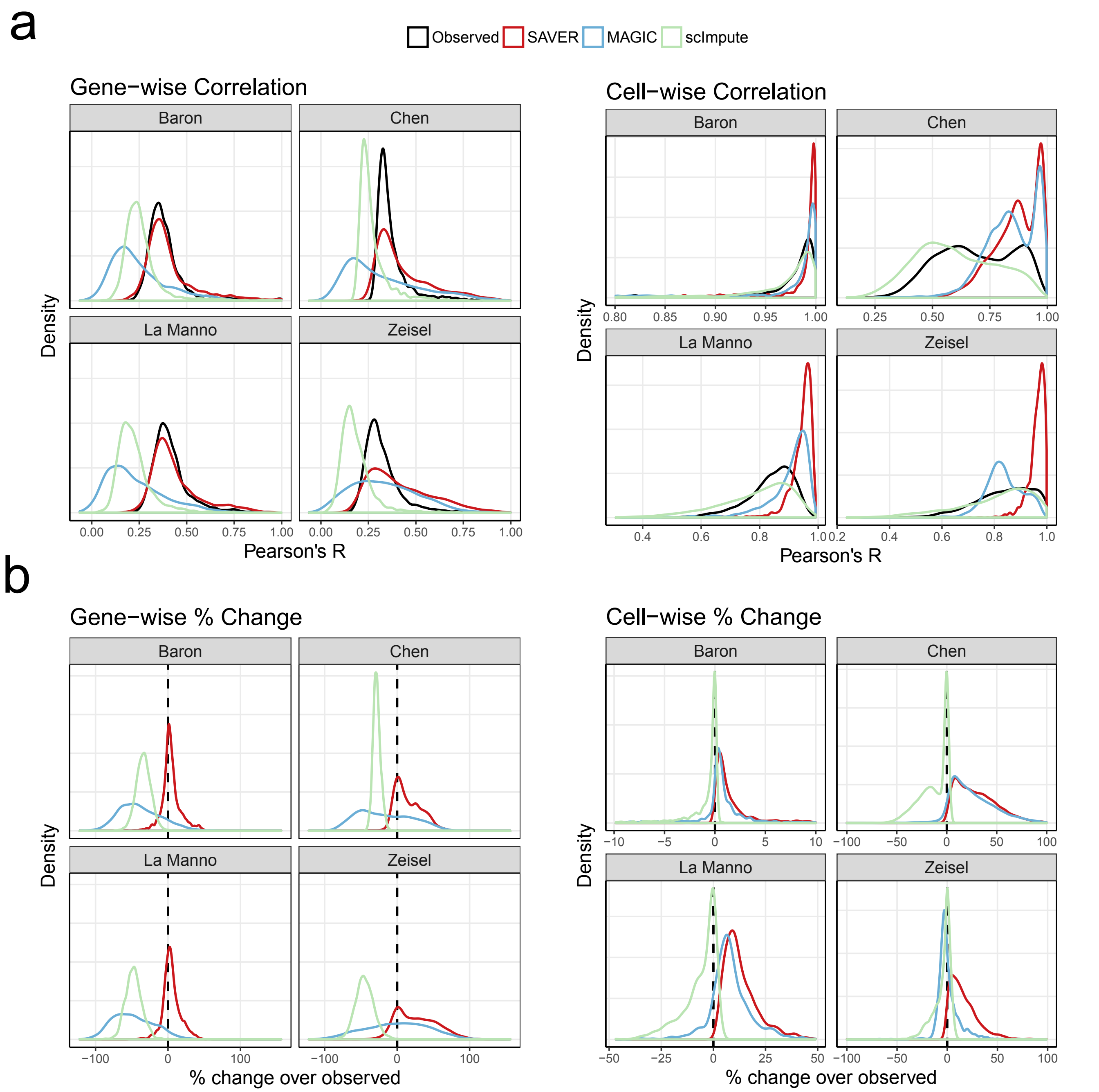
Supplementary Figure 2 Comparison of distributions of expression across cells between FISH, observed Drop-seq, and SAVER recovered expression for 13 genes. Densities were calculated using the *density* function in R with a Gaussian smoothing kernel. The bandwidth for each gene was set to be the default bandwidth selected for the FISH density. The distribution of the SAVER recovered expression matches more closely to the FISH distribution than the observed Drop-seq.



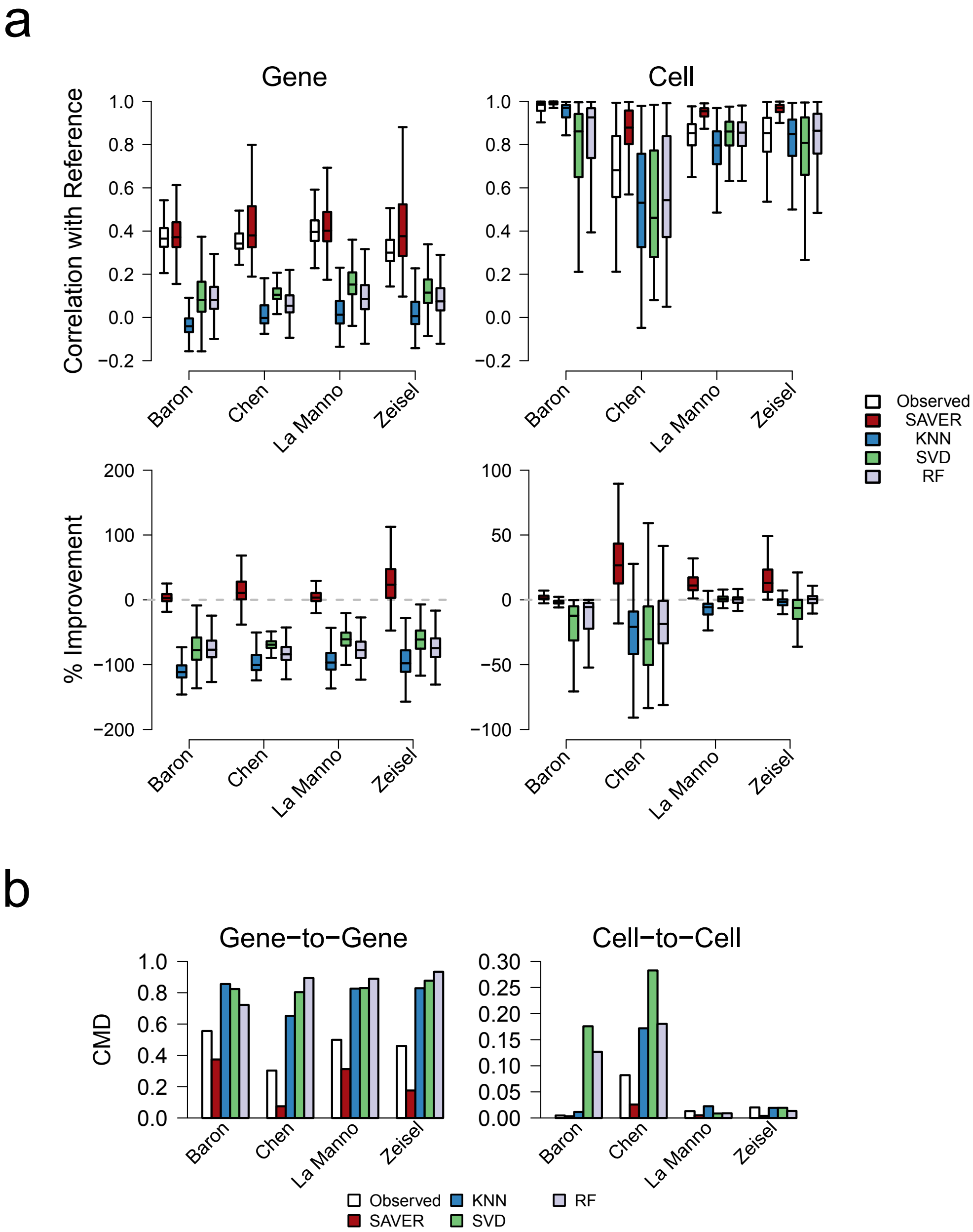
Supplementary Figure 3 RNA FISH validation of MAGIC and scImpute results for 15 genes. (**a**) Comparison of Gini coefficient for each gene between FISH and MAGIC (left) and between FISH and scImpute recovered values (right). SAVER outperforms MAGIC and scImpute in terms of correlation with FISH and root mean square error (RMSE). (**b**) Comparison of MAGIC, scImpute, and SAVER Kolmogorov-Smirnov (KS) distance to FISH distributions for the 15 genes. SAVER has lower KS distance with the FISH than MAGIC and scImpute for most genes. (**c**) Kernel density estimates of cross-cell expression distribution of *LMNA* (upper) and *CCNA2* (lower) for MAGIC and scImpute. (**d**) Comparison of pair-wise gene correlations computed from MAGIC (left) and from scImpute (right) with those computed from FISH counts. MAGIC introduces artificial correlations while scImpute inflates low correlations and shrinks high correlations. (**e**) Scatterplots of expression levels between *BABAM1* and *LMNA*. MAGIC creates unnatural relationships between the genes. scImpute contains discrete values which make gene relationships difficult to visualize.



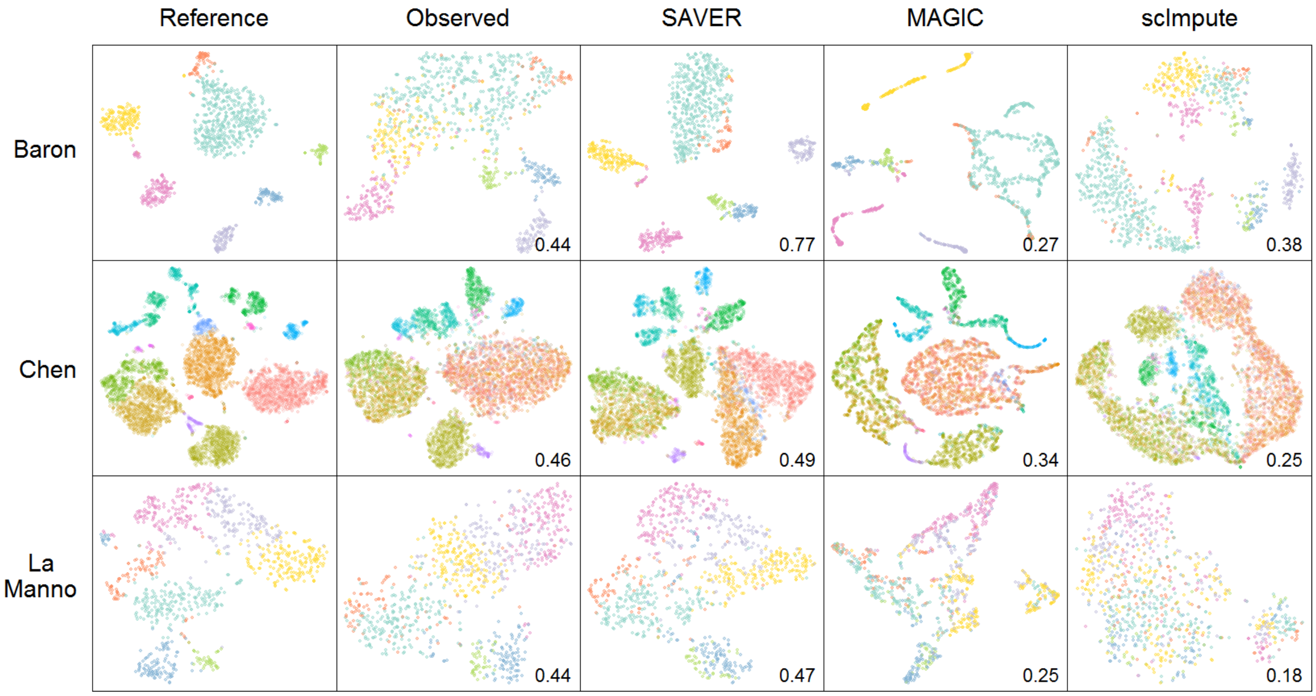
Supplementary Figure 4 Gene-to-gene correlations of 1,000 genes from a null dataset with no real gene relationships (Supplementary Note 4). SAVER is able to detect the absence of gene-to-gene correlations and shrinks the correlations to zero. MAGIC detects spurious correlation due to its tendency to oversmooth. scImpute also detects spurious correlation but to a lesser extent than MAGIC.



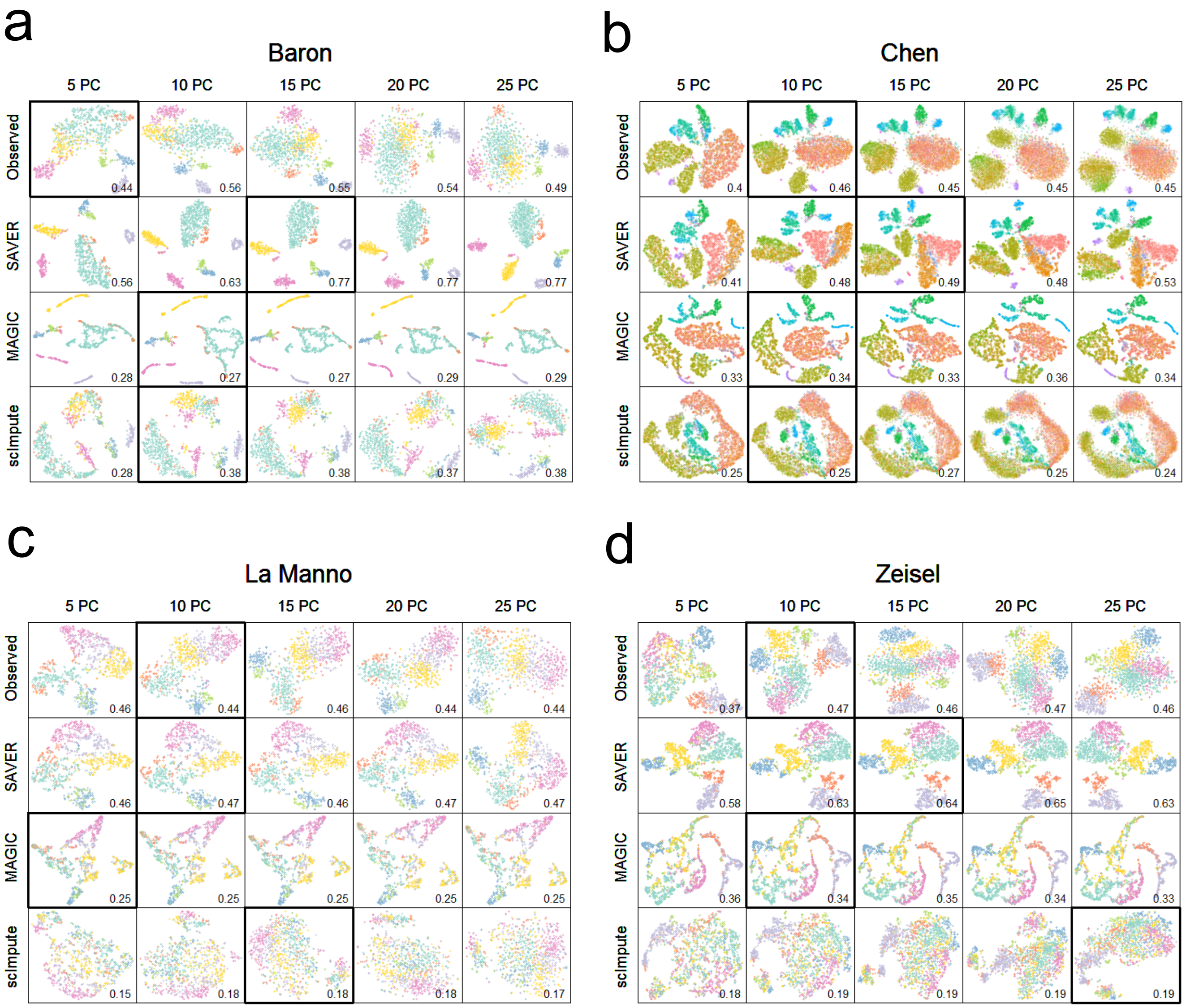
Supplementary Figure 5 Density plots of (**a**) gene-wise and cell-wise correlations with the reference and (**b**) % change in correlation compared to the observed data for SAVER, MAGIC, and scImpute. SAVER has the highest correlation with the reference for both gene-wise and cell-wise comparisions. SAVER is a large improvement over the observed for gene-wise correlations in the Chen and Zeisel datasets while performing similarly in the Baron and La Manno datasets. MAGIC and scImpute often perform worse. SAVER is a substantial improvement over the observed for cell-wise correlations in the Chen, La Manno, and Zeisel datasets, while performing slightly better than the observed for the Baron dataset due to the already high cell-wise correlations to the reference in the observed. MAGIC performs almost as well as SAVER for Baron, Chen, and La Manno, while scImpute performs worse than the observed.



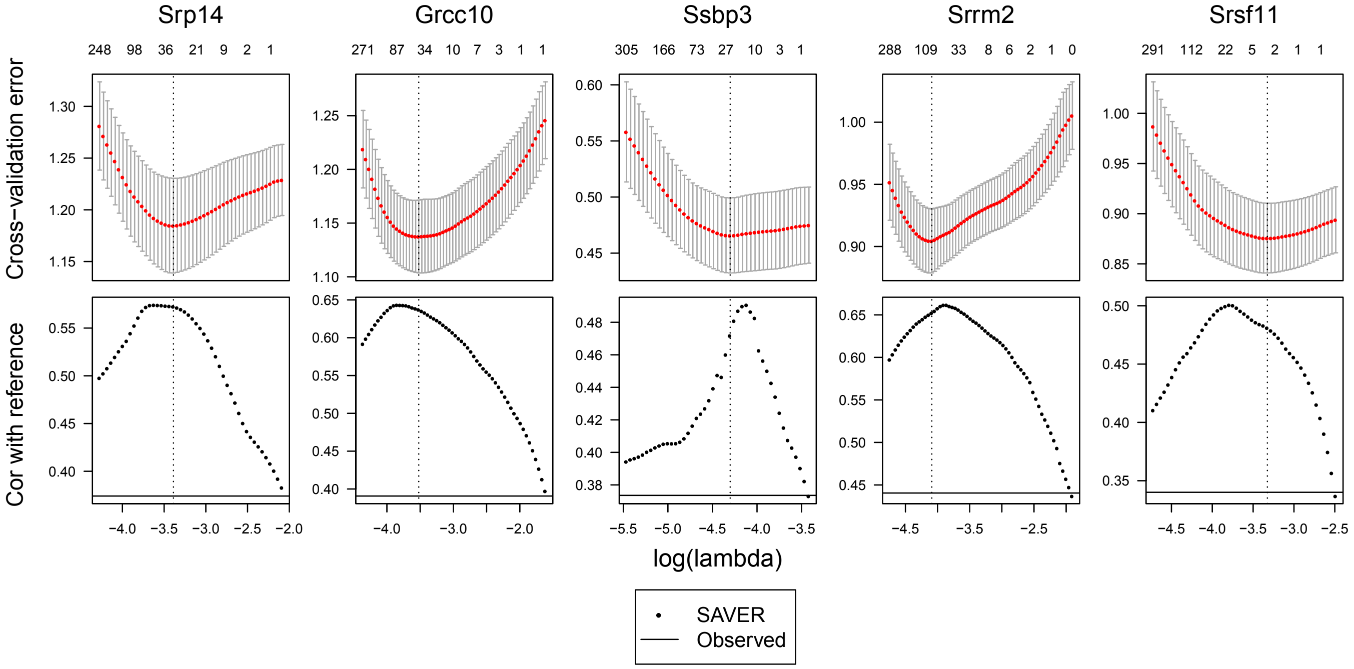
Supplementary Figure 6 Evaluation of SAVER against missing data imputation algorithms. The datasets are down-sampled from each reference dataset as described in Figure 2a. The algorithms are: SAVER, k-nearest neighbors (KNN) imputation, singular value decomposition (SVD) imputation, and random forest (RF) imputation. (**a**) Gene-wise and cell-wise correlations and improvements over the observed for each method. The imputation methods perform worse than the observed and SAVER, due to the fact that the zeros are not missing at random. (**b**) Comparison of gene-to-gene and cell-to-cell correlation matrices in terms of correlation matrix distance (CMD) from the reference. The imputation methods all have elevated CMD compared to the observed and SAVER.



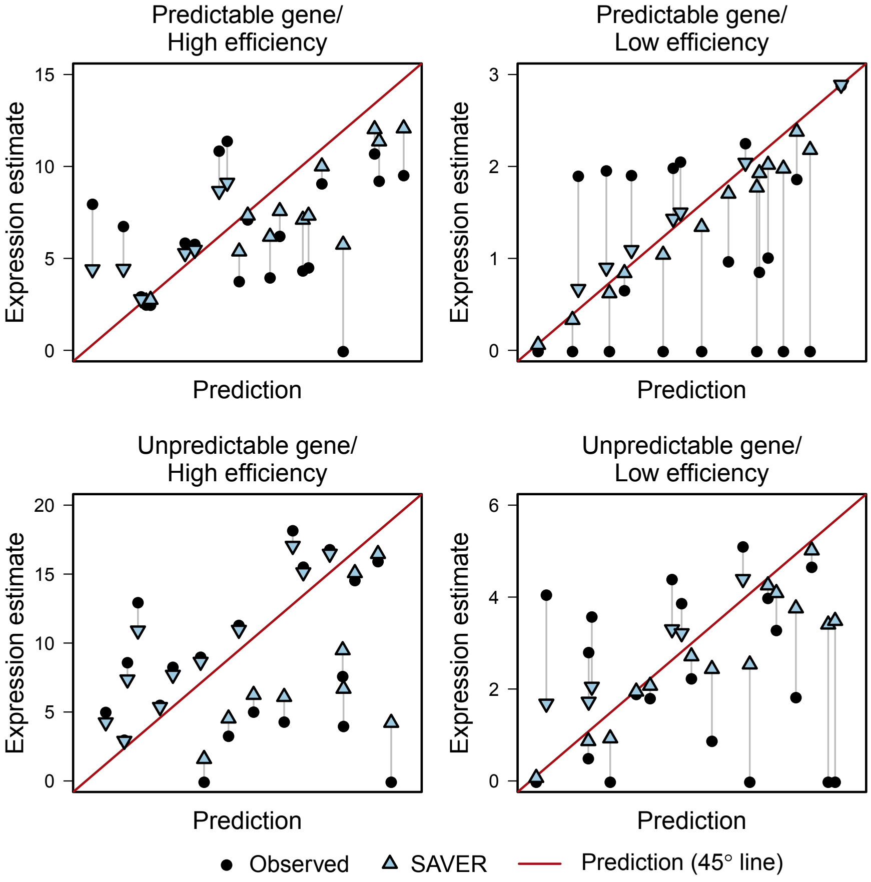
Supplementary Figure 7 Cell clustering and t-SNE visualization of the Baron, Chen, and La Manno reference, observed, and recovered down-sampled datasets. The colors represent the cell types identified by Seurat in the reference dataset. The Jaccard index measuring similarity between the observed/recovered clustering and reference clustering is displayed in the bottom right. SAVER t-SNE visualization and cell clustering are both more similar to the reference than the observed, MAGIC, and scImpute. Even though the Jaccard index for SAVER in the Chen and La Manno datasets are only slightly higher than the observed, the t-SNE visualization reveals that the cell types are more accurately represented in SAVER.



Supplementary Figure 8 Cell clustering and t-SNE visualization of the (**a**) Baron, (**b**) Chen, (**c**) La Manno, and (**d**) Zeisel observed and recovered down-sampled datasets. The number of principal components (PCs) used in the t-SNE visualization and clustering is varied from 5 PCs to 25 PCs. The number of PCs chosen by the jackStraw method is denoted by the bold outline. The colors represent the cell types identified by Seurat in the reference dataset using the number of PCs chosen by jackStraw. The Jaccard index measuring similarity between observed/recovered clustering and reference clustering is displayed in the bottom right. SAVER t-SNE visualization is robust to the number of PCs chosen, while other methods form different clusters depending on the number of PCs.



Supplementary Figure 9 Poisson Lasso regression cross-validation plots from Glmnet and correlation with Zeisel reference plots for five genes from the 5% efficiency dataset. The x-axis represents the size of the shrinkage penalty in the LASSO regression. The dotted vertical line represents the model with the lowest cross-validation error. SAVER correlation with the reference is approximately maximized when using the model with the lowest cross-validation error.



Supplementary Figure 10 Effect of predictability and efficiency on SAVER estimate. The SAVER estimate is a weighted average of the normalized observed expression and the predicted expression. The weight is dependent on the predictability of the gene and the cell-specific efficiency. Four scenarios are shown: Predictable (low ϕ_g) versus unpredictable (high ϕ_g) gene, in a high or low efficiency experiment. In each of the scatter plots, each point is a gene, and for each gene, the vertical lines connect the normalized observed expression with the gene's SAVER recovered value, which always lies between the normalized observed expression and the prediction (the 45 degree line).

			Mean exp			% zero			Average
Dataset	# Genes	# Cells	Orig	Ref	Down	Orig	Ref	Down	Efficiency
Baron	2284	1076	0.39	2.63	0.26	86.9%	46.5%	87.5%	10%
Chen	2159	7712	0.18	1.35	0.14	91.3%	54.0%	90.5%	10%
La Manno	2059	947	0.29	2.52	0.25	88.2%	39.7%	84.4%	10%
Zeisel	3529	1799	0.70	4.20	0.21	81.2%	27.3%	86.1%	5%

Supplementary Table 1 In the down-sampling experiment of four datasets, highly expressed genes and cells were chosen to form the reference dataset. Then, the expression levels for each reference dataset was set to be the mean of a Poisson random variable multiplied by a cell-specific efficiency parameter and the down-sampled data was created by sampling at each gene and cell. The reference dataset and efficiency was chosen such that the down-sampled data has approximately the same proportion of zeros as the original dataset.

	SAVER	MAST	scDD	SCDE
Reference	3049	2757	2829	2318
Down-sampled	1744	725	926	403

Supplementary Table 2 In the filtered Zeisel dataset, two subclasses of cells 351 CAPyr1 and 389 CA1Pyr2 cells were identified by BackSPIN, a biclustering algorithm. We compared the performance of the following differential expression methods under an FDR of 0.01: Wilcoxon rank sum test using SAVER, MAST, scDD, and SCDE. SAVER identifies the most differentially expressed genes for the down-sampled data.

Supplementary Note 1: Dispersion Estimation

Let Y_{gc} be the observed count of gene g in cell c. We can model Y_{gc} as a function of a cell-specific normalization constant s_c and a true expression λ_{gc}. We place a Gamma prior on λ_{gc}.

$$\begin{aligned} Y_{gc} &\sim \text{Poisson}(s_c \lambda_{gc}) \\ \lambda_{gc} &\sim \text{Gamma}(\alpha_{gc}, \beta_{gc}) \end{aligned}$$

Let μ be the mean and v be the variance of a Gamma-distributed random variable $X \sim \text{Gamma}(\alpha, \beta)$. Under the shape-rate parameterization,

$$\begin{aligned} f(x) &= \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \\ \mu &= \alpha/\beta \\ v &= \alpha/\beta^2 \end{aligned}$$

Instead of parameterizing by α and β, we can reparameterize in terms of the moments μ and v:

$$\lambda_{gc} \sim \text{Gamma}(\alpha_{gc}, \beta_{gc}) \Leftrightarrow \lambda_{gc} \sim \text{Gamma}\left(\frac{\mu_{gc}^2}{v_{gc}}, \frac{\mu_{gc}}{v_{gc}}\right)$$

μ_{gc} is obtained by fitting a Lasso Poisson regression as described in the Methods. Next, we want to estimate v_{gc}. We assume that for a given gene g, there is an underlying mean-variance or dispersion relationship common to that gene. The following are three scenarios which we consider:

1. Constant variance: $v_{gc} = v_g$
2. Constant Fano: $v_{gc} = F_g \mu_{gc}$
3. Constant CV^2: $v_{gc} = CV_g^2 \mu_{gc}^2$

Constant Variance

Under this scenario, we assume that all the cells for a particular gene share a variance v_g which is independent of the mean. This independence implies that the predicted values are homoscedastic. Thus, the prior Gamma distribution under the moment parametrization takes the form

$$\lambda_{gc} \sim \text{Gamma}\left(\frac{\mu_{gc}^2}{v_g}, \frac{\mu_{gc}}{v_g}\right)$$

To find v_g, we need to maximize the marginal likelihood of Y_{gc} given μ_{gc} and v_g. Here, $Y_{gc}|\mu_{gc}, v_g$ follows a negative binomial distribution with density function

$$f(y|\mu_{gc}, v_g) = \frac{s_c^y}{y!} \frac{(\frac{\mu_{gc}}{v_g})^{\frac{\mu_{gc}^2}{v_g}}}{\Gamma(\frac{\mu_{gc}^2}{v_g})} \frac{\Gamma(y + \frac{\mu_{gc}^2}{v_g})}{(s_c + \frac{\mu_{gc}}{v_g})^{y + \frac{\mu_{gc}^2}{v_g}}}$$

Assuming independence across cells, the likelihood is simply the product of the individual densities:

$$
\begin{aligned}
L(v_g|Y_{gc},\mu_{gc}) &= \prod_{c=1}^{C} \frac{s_c^{Y_{gc}}}{Y_{gc}!} \frac{(\frac{\mu_{gc}}{v_g})^{\frac{\mu_{gc}^2}{v_g}}}{\Gamma(\frac{\mu_{gc}^2}{v_g})} \frac{\Gamma(Y_{gc}+\frac{\mu_{gc}^2}{v_g})}{(s_c+\frac{\mu_{gc}}{v_g})^{Y_{gc}+\frac{\mu_{gc}^2}{v_g}}} \\
l(v_g|Y_{gc},\mu_{gc}) &= \sum_{c=1}^{C} Y_{gc}\log s_c - \log Y_{gc}! + \frac{\mu_{gc}^2}{v_g}\log\mu_{gc} - \frac{\mu_{gc}^2}{v_g}\log v_g - \log\Gamma\left(\frac{\mu_{gc}^2}{v_g}\right) \\
&\quad + \log\Gamma\left(Y_{gc}+\frac{\mu_{gc}^2}{v_g}\right) - \left(Y_{gc}+\frac{\mu_{gc}^2}{v_g}\right)\log\left(s_c+\frac{\mu_{gc}}{v_g}\right)
\end{aligned}
$$

We find the $\widehat{v}_g$ which maximizes this likelihood using the `optimize` function in R.

Constant Fano Factor

Under the constant Fano factor assumption, we assume that the variance scales linearly with the mean. This corresponds with assuming the distribution of a gene is Poisson-like in the mean-variance relationship. The Fano factor F_g can be expressed as

$$F_g = \frac{v_{gc}}{\mu_{gc}} = \frac{1}{\beta_g}$$

Thus, assuming a constant Fano factor is equivalent to assuming a constant rate β_g parameter in the usual Gamma distribution parametrization. Therefore, we have the following prior and want to find β_g.

$$\lambda_{gc} \sim \text{Gamma}(\mu_{gc}\beta_g, \beta_g)$$

The log-likelihood is calculated similarly as above.

$$
\begin{aligned}
l(\beta_g|Y_{gc},\mu_{gc}) &= \sum_{c=1}^{C} Y_{gc}\log s_c - \log Y_{gc}! + \mu_{gc}\beta_g\log\beta_g - \log\Gamma(\mu_{gc}\beta_g) \\
&\quad + \log\Gamma(Y_{gc}+\mu_{gc}\beta_g) - (Y_{gc}+\mu_{gc}\beta_g)\log(s_c+\beta_g)
\end{aligned}
$$

For numerical stability, we maximize with respect to $1/\beta_g$ to get $\widehat{\beta}_g$.

Constant Coefficient of Variation

Under the constant coefficient of variation assumption, we assume that the variance scales quadratically with the mean. This corresponds to a typical constant scaling of a Gamma distribution, since scaling by a constant c still gives a Gamma distribution with mean scaled by c and variance scaled by c^2. The coefficient of variation CV^2 can be expressed as

$$CV_g^2 = \frac{v_{gc}}{\mu_{gc}^2} = \frac{1}{\alpha_g}$$

Thus, assuming a constant coefficient of variation is equivalent to assuming a constant shape α_g parameter in the usual Gamma distribution parametrization. Therefore, we have the following prior and want to find $\widehat{\alpha}_g$.

$$\lambda_{gc} \sim \text{Gamma}\left(\alpha_g, \frac{\alpha_g}{\mu_{gc}}\right)$$

The log-likelihood is

$$l(\alpha_g | Y_{gc}, \mu_{gc}) = \sum_{c=1}^{C} Y_{gc} \log s_c - \log Y_{gc}! + \alpha_g \log \alpha_g - \alpha_g \log \mu_{gc} - \log \Gamma(\alpha_g) \\ + \log \Gamma(Y_{gc} + \alpha_g) - (Y_{gc} + \alpha_g) \log \left(s_c + \frac{\alpha_g}{\mu_{gc}} \right)$$

For numerical stability, we maximize with respect to $1/\alpha_g$ to get $\widehat{\alpha}_g$.

Estimating $\widehat{v}_{gc}$

For gene g, let $l^v(\widehat{v}_g)$, $l^F(\widehat{\beta}_g)$, and $l^{cv}(\widehat{\alpha}_g)$ be the maximized marginal likelihoods under constant variance, constant Fano factor, and constant coefficient of variation respectively. To find the noise model that corresponds to gene g, we take the maximum of $l^v(\widehat{v}_g)$, $l^F(\widehat{\beta}_g)$, and $l^{cv}(\widehat{\alpha}_g)$.

If the maximum is $l^v(\widehat{v}_g)$, then we assign a constant variance model for gene g and let $\widehat{v}_{gc} = \widehat{v}_g$.

If the maximum is $l^F(\widehat{\beta}_g)$, then we assign a constant Fano factor model for gene g and let $\widehat{v}_{gc} = \widehat{\mu}_{gc}/\widehat{\beta}_g$.

If the maximum is $l^{cv}(\widehat{\alpha}_g)$, then we assign a constant coefficient of variation model for gene g and let $\widehat{v}_{gc} = \widehat{\mu}_{gc}^2/\widehat{\alpha}_g$.

Supplementary Note 2: Correlation Calculation Adjustment

Recall the SAVER output for gene g and cell c is the posterior random variable λ_{gc}. Suppose we want to find the correlation between two genes g and g' across cells, namely $\text{Cor}(\boldsymbol{\lambda_g}, \boldsymbol{\lambda_{g'}})$, where $\boldsymbol{\lambda_g} = (\lambda_{g1}, \dots, \lambda_{gC})$. As the SAVER posterior mean $\widehat{\lambda}_{gc}$ is a point estimate, calculating the correlation between the SAVER estimates will be an overestimate of the magnitude of the correlation since the variance is not taken into account:

$$\left|\text{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})\right| \geq \left|\text{Cor}(\boldsymbol{\lambda_g}, \boldsymbol{\lambda_{g'}})\right|$$

Thus, if we want to calculate the correlation between gene g and g', we need to adjust $\text{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})$ by gene-specific correlation factors γ_g and $\gamma_{g'}$ such that

$$\text{Cor}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'}) = \gamma_g \gamma_{g'} \text{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})$$

Derivation of γ_g

Consider the definition of correlation:

$$\text{Cor}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'}) = \frac{\text{Cov}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'})}{\sqrt{\text{Var}(\boldsymbol{\lambda}_g)}\sqrt{\text{Var}(\boldsymbol{\lambda}_{g'})}}$$

Let $\boldsymbol{Z} = \{\boldsymbol{Y}_g, \boldsymbol{Y}_{g'}, \boldsymbol{\alpha}_g, \boldsymbol{\alpha}_{g'}, \boldsymbol{\beta}_g, \boldsymbol{\beta}_{g'}\}$. Given $\boldsymbol{Z}$, $\boldsymbol{\lambda}_g$ and $\boldsymbol{\lambda}_{g'}$ are independent. By the law of total covariance and independence,

$$\begin{aligned}\text{Cov}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'}) &= \text{E}[\text{Cov}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'}|\boldsymbol{Z})] + \text{Cov}[\text{E}(\boldsymbol{\lambda}_g|\boldsymbol{Z}), \text{E}(\boldsymbol{\lambda}_{g'}|\boldsymbol{Z})] \\ &= \text{Cov}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})\end{aligned}$$

In addition, by the law of total variance,

$$\begin{aligned}\text{Var}(\boldsymbol{\lambda}_g) &= \text{Var}[E(\boldsymbol{\lambda}_g|\boldsymbol{Z})] + \text{E}[\text{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})] \\ &= \text{Var}(\widehat{\boldsymbol{\lambda}}_g) + \text{E}[\text{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})]\end{aligned}$$

Next, we can solve for $\text{Cor}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'})$.

$$\begin{aligned}\text{Cor}(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'}) &= \frac{\text{Cov}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})}{\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_g) + \text{E}[\text{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})]}\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_{g'}) + \text{E}[\text{Var}(\boldsymbol{\lambda}_{g'}|\boldsymbol{Z})]}} \\ &= \frac{\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_g)}\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_{g'})}}{\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_g) + \text{E}[\text{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})]}\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_{g'}) + \text{E}[\text{Var}(\boldsymbol{\lambda}_{g'}|\boldsymbol{Z})]}} \frac{\text{Cov}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})}{\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_g)}\sqrt{\text{Var}(\widehat{\boldsymbol{\lambda}}_{g'})}} \\ &= \sqrt{\frac{\text{Var}(\widehat{\boldsymbol{\lambda}}_g)}{\text{Var}(\widehat{\boldsymbol{\lambda}}_g) + \text{E}[\text{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})]}}\sqrt{\frac{\text{Var}(\widehat{\boldsymbol{\lambda}}_{g'})}{\text{Var}(\widehat{\boldsymbol{\lambda}}_{g'}) + \text{E}[\text{Var}(\boldsymbol{\lambda}_{g'}|\boldsymbol{Z})]}}\text{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'}) \\ &= \gamma_g \gamma_{g'} \text{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'}),\end{aligned}$$

where

$$\gamma_g = \sqrt{\frac{\text{Var}(\widehat{\boldsymbol{\lambda}}_g)}{\text{Var}(\widehat{\boldsymbol{\lambda}}_g) + \text{E}[\text{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})]}}$$

γ_g takes into account the posterior variance through $\mathrm{E}[\mathrm{Var}(\boldsymbol{\lambda}_g|\boldsymbol{Z})]$. Thus, if the expected posterior variance across cells is high compared to the variance of the estimates, then γ_g will be small and $\left|\mathrm{Cor}(\boldsymbol{\lambda_g}, \boldsymbol{\lambda_{g'}})\right| \ll \left|\mathrm{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})\right|$. However, if the expected posterior variance across cells is small compared to the variance of the estimates, then $\gamma_g \approx 1$ and $\left|\mathrm{Cor}(\boldsymbol{\lambda_g}, \boldsymbol{\lambda_{g'}})\right| \approx \left|\mathrm{Cor}(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'})\right|$
In calculating the sample adjusted correlation, we have the expression

$$\mathrm{Cor}_s(\boldsymbol{\lambda}_g, \boldsymbol{\lambda}_{g'}) = \widehat{\gamma}_g\widehat{\gamma}_{g'}\mathrm{Cor}_s(\widehat{\boldsymbol{\lambda}}_g, \widehat{\boldsymbol{\lambda}}_{g'}),$$

where

$$\widehat{\gamma}_g = \sqrt{\frac{\mathrm{Var}_s(\widehat{\boldsymbol{\lambda}}_g)}{\mathrm{Var}_s(\widehat{\boldsymbol{\lambda}}_g) + \frac{1}{C}\sum_{c=1}^{C}[\mathrm{Var}(\lambda_{gc}|\boldsymbol{Z})]}}$$

and the subscript s represent sample estimates.

Example

As an example, we will take a look at the effect of taking into account the adjustment factor in calculating correlation between *BABAM1* and *LMNA*, the two genes analyzed in Figure 1e.

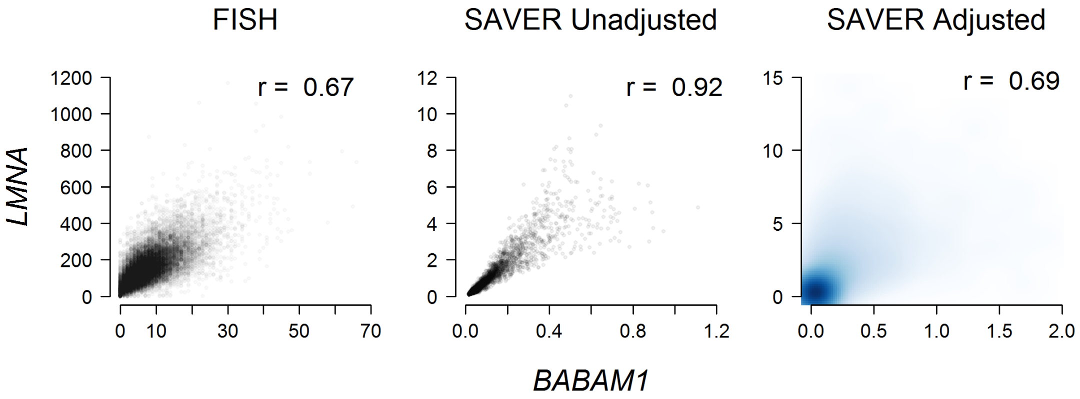


Figure 1 Scatterplot of *BABAM1* and *LMNA* for FISH (left), unadjusted SAVER (center), and adjusted SAVER (right). The smooth scatterplot for the adjusted SAVER was created by sampling from the posterior distribution for each cell.

The correlation between *BABAM1* and *LMNA* in FISH is 0.67. When we calculate the correlation in the SAVER estimates as $\mathrm{Cor}_s(\widehat{\boldsymbol{\lambda}}_{BABAM1}, \widehat{\boldsymbol{\lambda}}_{LMNA})$, we get a value of 0.92. Taking the posterior uncertainty into account, we calculate the adjustment factor as $\widehat{\gamma}_{BABAM1} = 0.81$ and $\widehat{\gamma}_{LMNA} = 0.93$, so the adjusted correlation is

$$r_{adj} = \widehat{\gamma}_{BABAM1}\widehat{\gamma}_{LMNA}\mathrm{Cor}_s(\widehat{\boldsymbol{\lambda}}_{BABAM1}, \widehat{\boldsymbol{\lambda}}_{LMNA}) = 0.69$$

Supplementary Note 3: Performance of SAVER under different scenarios

The performance of SAVER can vary depending on the sequencing depth, the number of cells, the amount of heterogeneity in the population, and gene-specific properties. We will address these factors one by one.

Sequencing depth

To examine the performance of SAVER under various sequencing depths, we down-sampled the Zeisel reference dataset at mean efficiencies of 25%, 10%, and 5% (Fig. 1, Table 1).

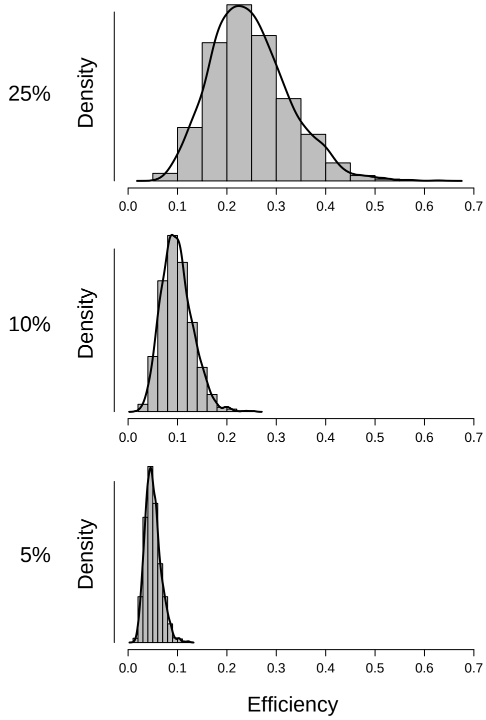


Figure 1 Histograms of cell-specific efficiencies at each mean efficiency level.

	Ref	25%	10%	5%
Mean exp	4.20	1.04	0.42	0.21
% zero	27.3%	61.9%	77.6%	86.1%

Table 1 Mean expression and percentage zero at each efficiency.

We then compared gene-wise correlations and cell-wise correlations with the reference for observed, SAVER, MAGIC, and scImpute (Fig. 2). Gene-wise and cell-wise correlations with the reference decreases as efficiency decreases, yet SAVER maintains the highest correlation with the reference across all efficiencies. The biggest improvement for SAVER over the observed occurs at the lowest efficiency of 5%.

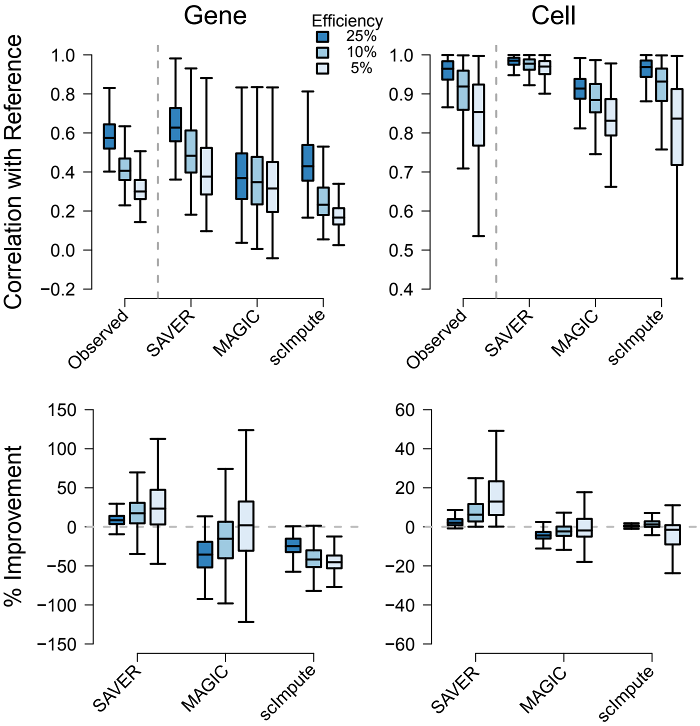


Figure 2 Performance of algorithms measured by correlation with reference, on the gene level (left) and on the cell level (right). Percentage improvement over using the observed data is shown in the lower two panels. SAVER is more closely correlated with the truth across both genes and cells.

Next, we calculated the correlation matrix distance (CMD) between the reference and the observed/recovered correlation matrices at each efficiency (Fig. 3). The CMD for both gene-to-gene and cell-to-cell correlation matrices increases as efficiency decreases, yet SAVER maintains the lowest CMD with the reference across all efficiencies. Similarly, the improvement of SAVER is most noticeable at the lowest efficiency of 5%.

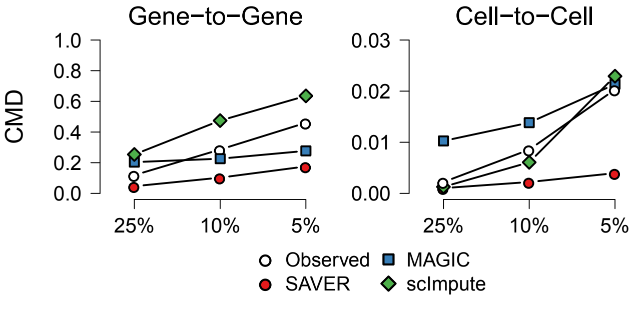


Figure 3 Comparison of gene-to-gene (left) and cell-to-cell (right) correlation matrices of recovered values with the true correlation matrices, as measured by correlation matrix distance (CMD).

To investigate the impact of sequencing depth on downstream analysis, we repeated the Zeisel differential expression analysis and cell clustering and visualization for each efficiency. In the differential expression analysis, the number of detected differentially expressed genes decreases as efficiency decreases, while SAVER maintains the highest number of differentially expressed genes across all efficiencies (Fig. 4).

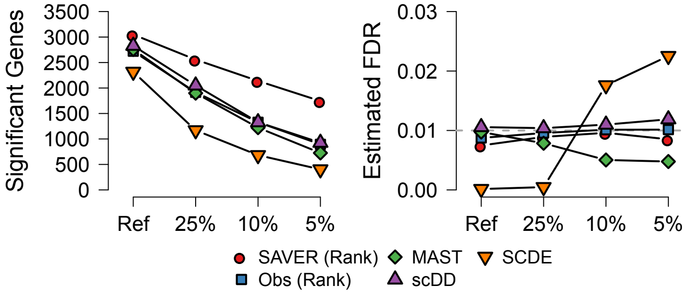


Figure 4 Differential expression (DE) analysis between CA1Pyr1 cells (n = 351) and CA1Py2 cells (n = 389). SAVER yields more significant genes across efficiencies (left), while still controlling false discovery rate at 0.01 (right).

In the cell clustering analysis, the Jaccard index decreases and the clusters become less defined in the t-SNE visualization as efficiency decreases (Fig. 5). SAVER maintains high cluster similarity across all efficiencies with the most improvement at the 5% efficiency.

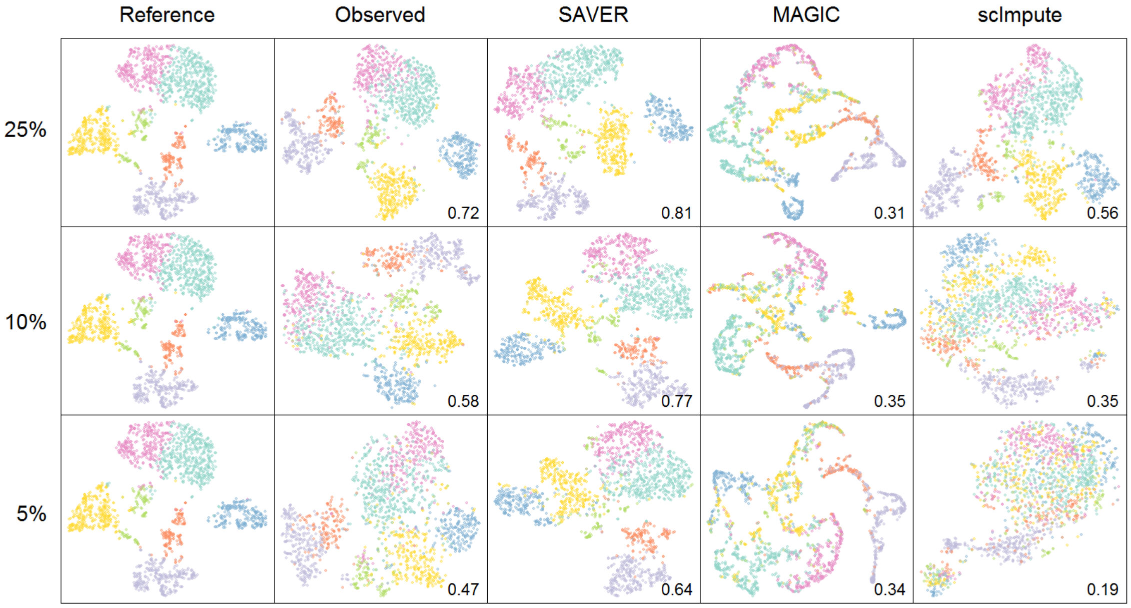


Figure 5 Cell clustering and t-SNE visualization of the Zeisel dataset across efficiencies.

By performing the down-sampling experiment at three different efficiencies for the Zeisel data, we discovered that as sequencing depth decreases, the observed data becomes less representative of the true expression data. However, SAVER is able to recover the original data, especially so for low efficiency setting.

Number of cells

Next, we wanted to see the effect of the number of cells on SAVER performance. Often, the number of cells and sequencing depth go hand-in-hand when choosing to design an experiment. An experiment with a large number of cells will usually have low sequencing depth and vice versa. The Zeisel down-sampled dataset contains 1,799 cells. We calculated the gene-wise and cell-wise correlations at subsamples of 250, 500, 1,000, and 1,500 cells across the three efficiencies (Fig. 6).

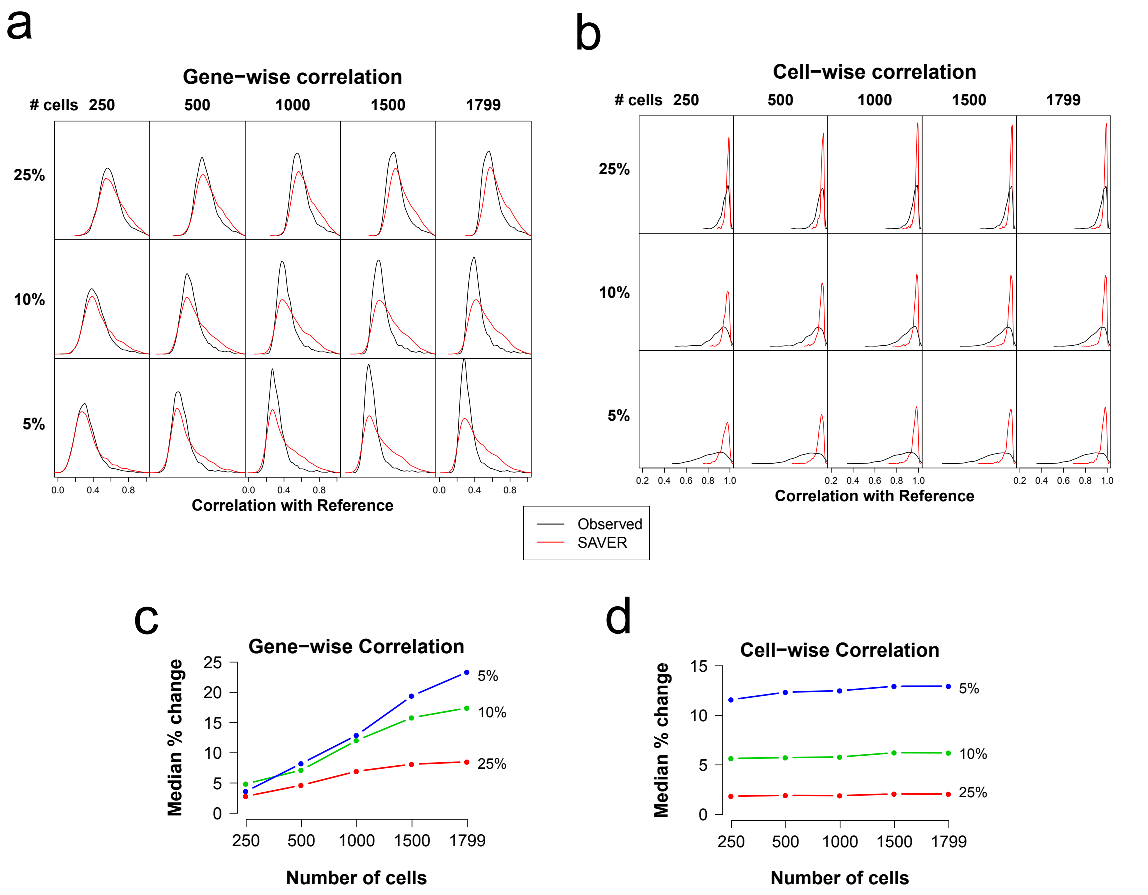


Figure 6 Effect of subsampling cells on SAVER correlation with the Zeisel reference dataset at 25%, 10%, and 5% efficiencies. Cells were randomly subsampled at each specified sample size. Density plots showing genewise (**a**) and cell-wise (**b**) correlations with reference across efficiencies reveal that SAVER performs at least as well as the observed even at small sample sizes. (**c**) Median percentage improvement over observed in terms of gene-wise correlations. (**d**) Median percentage improvement over observed in terms of cell-wise correlations.

As the number of cells increases, the gene-wise correlations increase as well across all efficiencies (Fig. 6a). At lower efficiencies, increasing the number of cells has a larger effect on the gene-wise correlation improvement than at higher efficiencies (Fig. 6c). It is important to note that when the number of cells is small, SAVER does not try to over-aggressively use information that is not there. As a result, the performance of SAVER will always improve on the observed data. Interestingly, the cell-wise correlation improvement depends on the efficiency but not on the number of cells (Fig. 6b,d). This is likely because cells behave independently so adding additional cells does not improve cell-wise correlations.

We also decided to look at the effect of subsampling on recovery of the Gini coefficient in the FISH Drop-seq experiment (Fig. 7). For most genes, the Gini coefficient is adequately recovered using only 250 out of the 8,498 cells and remains constant across all sample sizes. However, genes like *BABAM1*, *CCNA2*, and *SOX10* demonstrate improved estimation of the Gini coefficient by SAVER as the sample size increases.

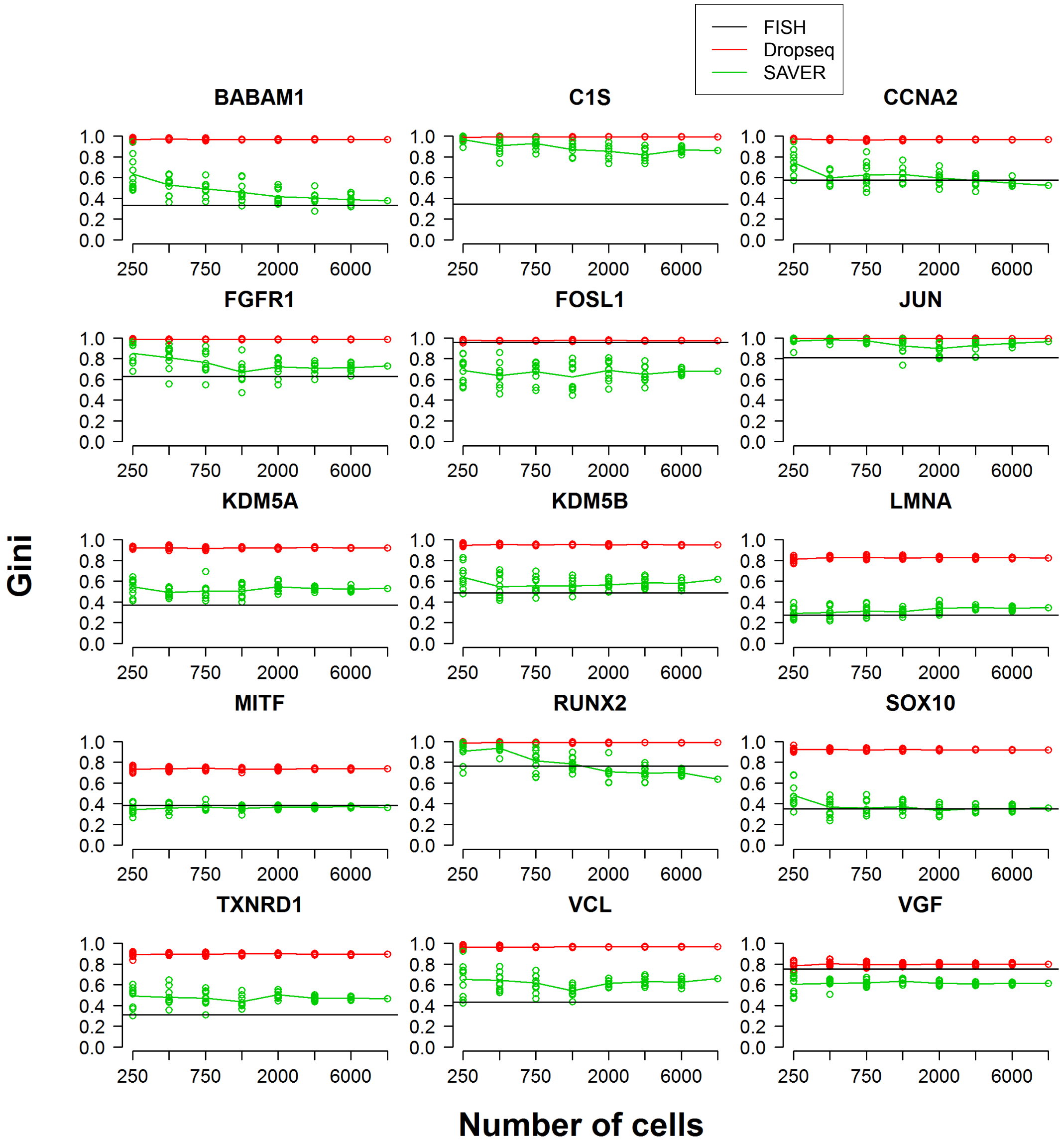


Figure 7 Effect of subsampling cells on SAVER recovery of Gini coefficient on the Torre and Dueck melanoma Drop-seq dataset. 10 datasets were randomly subsampled at each sample size, represented by the open points. The line represents the mean across the 10 datasets.

Cell Heterogeneity

Another factor that affects the performance of SAVER is the heterogeneity of the cell population in the study. To investigate this, we analyzed the down-sampled 5% Zeisel data, which contained 834 Pyramidal CA1 cells as identified by the authors. We selected these 834 cells and treated this as the homogeneous population. Then, we took a random sample of 834 other cells, which are a mix of different cell types, and treated this as the heterogeneous population. We applied SAVER to these two datasets separately and evaluated the performance by calculating the gene-wise and cell-wise correlations with the reference data.

The observed homogeneous dataset had lower gene-wise and cell-wise correlations with the reference than the observed heterogeneous dataset (Fig. 8). SAVER improves on the gene-wise correlation for both the homogeneous and heterogeneous datasets with greater improvement in the heterogeneous population. SAVER also substantially improves the cell-wise correlations with the reference, with greater improvement in the homogeneous population which in the observed data, had much lower correlation than the heterogeneous population.

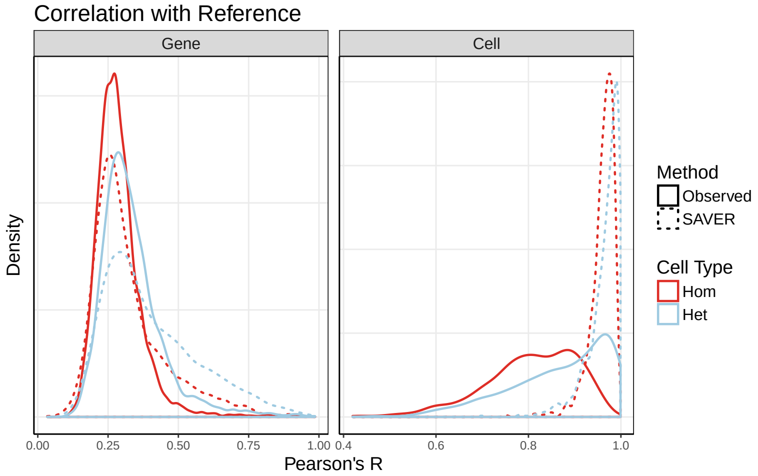


Figure 8 Comparison of the performance of SAVER on a homogeneous cell population versus a heterogeneous cell population.

Regardless of the composition of cell types in the data, SAVER improves on both gene-wise and cell-wise correlations. The biggest gains in gene-wise correlation are in heterogeneous datasets where there is more information to leverage gene relationships. The biggest gains in cell-wise correlation are in homogeneous datasets where the observed cell-wise correlations are low compared to heterogeneous datasets.

Gene properties

Properties of specific genes such as mean expression level, percentage of cells with non-zero expression, and variability of the expression across cells can affect the performance of SAVER in recovering the gene. To study these effects, we analyzed the 5% down-sampled Zeisel dataset and compared the percent improvement in gene-wise correlation with the reference over observed with mean expression, percent non-zero cells, and variability of expression (Fig. 9).

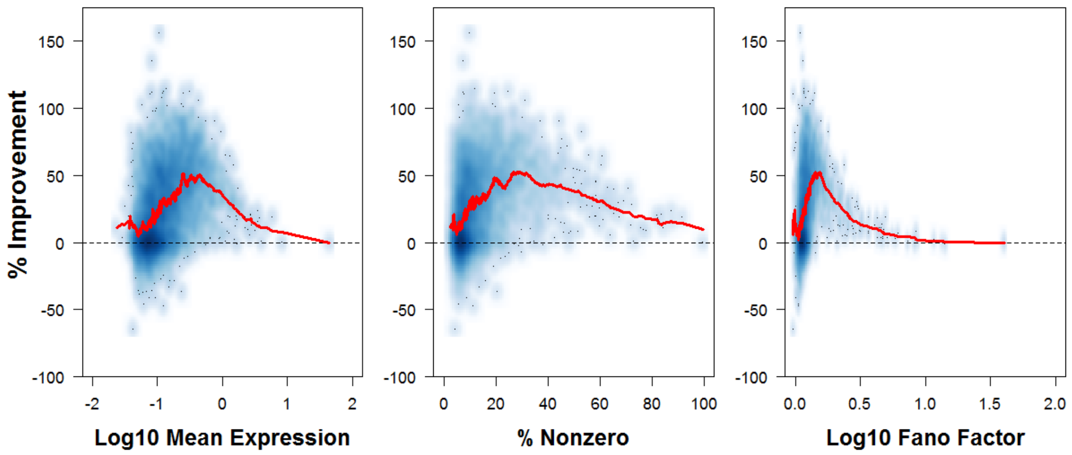


Figure 9 Comparison of gene-wise correlation percentage improvement over observed for 5% Zeisel SAVER recovered expression across gene metrics such as mean expression, % nonzero, and Fano factor. The red line represents the moving average.

For genes with high expression, expression in a high percentage of cells, or high dispersion, SAVER does not perform much better than the observed. The reason is that the true expression of these highly expressed, variable genes are less subject to noise so the observed values are good estimates for the true expression. SAVER recognizes this and puts more weight on the observed values. For genes with very low expression, expression in few cells, or low dispersion, SAVER also does not perform much better than the observed because of the lack of information available to generate reasonable predictions. Once again, SAVER can identify these genes and adaptively does not try to overfit. With the exception of these very highly expressed or lowly expressed genes, SAVER substantially improves on the gene-wise correlations with the reference.

Identification of true zeros

A difficult question to address in scRNA-seq expression analysis is whether an observed zero is a true zero, i.e. the true expression of the gene is zero, or an induced zero due to low sampling efficiency. Here, we wanted to see if SAVER could distinguish between true zeros and sampled zeros. Using the Zeisel reference dataset, we classified observed zero expression in the 25%, 10%, and 5% down-sampled datasets as either true zeros or sampled zeros. Then we compared the SAVER estimates for these two groups (Fig. 10).

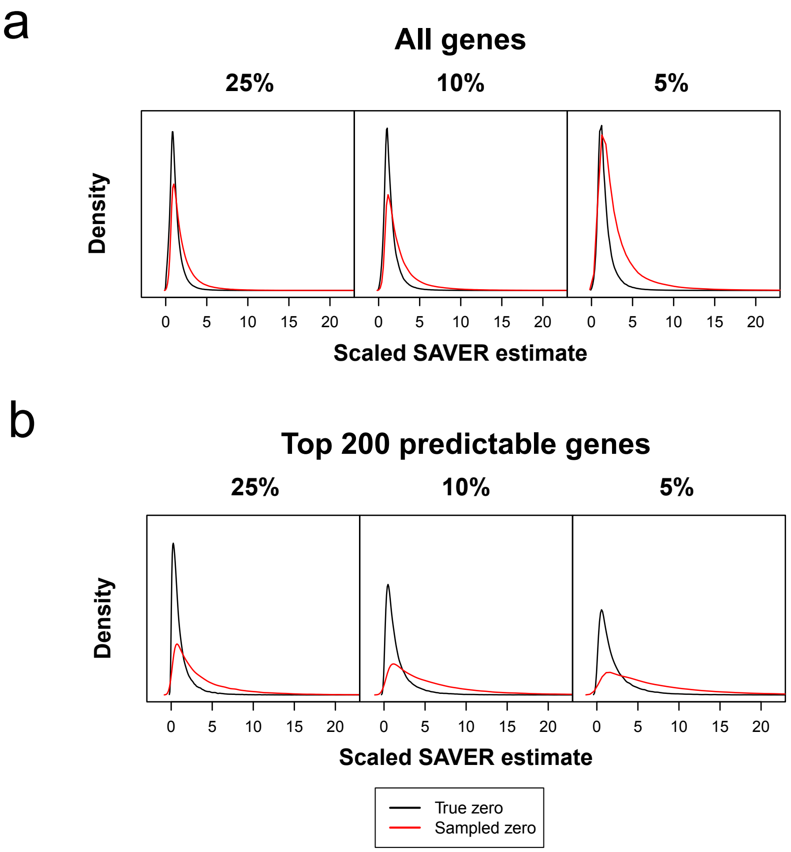


Figure 10 Density plots of the SAVER estimates for true zeros in the reference Zeisel dataset and down-sampled zeros which were non-zero in the reference dataset looking at (**a**) all genes and (**b**) the top 200 predictable genes as defined by the size of the correlation adjustment factor derived in Supplementary Note 2.

The SAVER estimates are slightly higher for the sampled zeros than the true zeros for all genes, but when restricted to the top 200 predictable genes, SAVER is able to more clearly distinguish the sampled zeros from the true zeros.

Supplementary Note 4: Permutation experiment to test for oversmoothing

A main danger in gene expression recovery is creating false correlations between genes. This could happen if the model overfits to the data, or if the final estimates rely too heavily on the prediction model and neglects the natural gene expression stochasticity between cells.

We investigated whether SAVER, MAGIC, and scImpute overfits by creating a null dataset where there are no real relationships between genes and cells. Instead of simulating from scratch, we take the existing Drop-seq dataset from the Torre and Dueck et al. and permute the cell labels independently for each gene. This maintains the marginal distributions of the original scRNA-seq dataset while destroying all relationships between genes. Between any pair of genes in the permuted data, the correlation should be zero.

SAVER, MAGIC, and scImpute are applied to this permuted data. We then calculated the gene-to-gene correlations for 1,000 randomly selected genes. Supplementary Figure 4 shows violin plots of these gene-to-gene correlations. Indeed, the correlations are mostly zero for the observed data. MAGIC grossly inflates the correlations, thus one would detect spurious correlation from MAGIC results. The correlations are also slightly inflated in the scImpute results. Reassuringly, SAVER does not inflate the correlations and even shrink them further towards zero; this is expected since SAVER reduces the noise as compared to the observed counts, and thus should more reliably estimate the true correlation of zero.