McImpute: Matrix completion based imputation for single cell RNA-seq data

Aanchal Mongia[1], Debarka Sengupta[1,3*], Angshul Majumdar[2]

1 Dept. of Computer Science and Engineering/IIIT-Delhi/Delhi-110020/India
2 Dept. of Electronics and Communications Engineering/IIIT-Delhi/Delhi-110020/India
3 Center for Computational Biology/IIIT-Delhi/Delhi-110020/India

* Corresponding Author
{aanchalm, debarka, angshul}@iiitd.ac.in

Abstract

Motivation: Single cell RNA sequencing has been proved to be revolutionary for its potential of zooming into complex biological systems. Genome wide expression analysis at single cell resolution, provides a window into dynamics of cellular phenotypes. This facilitates characterization of transcriptional heterogeneity in normal and diseased tissues under various conditions. It also sheds light on development or emergence of specific cell populations and phenotypes. However, owing to the paucity of input RNA, a typical single cell RNA sequencing data features a high number of dropout events where transcripts fail to get amplified.

Results: We introduce mcImpute, a low-rank matrix completion based technique to impute dropouts in single cell expression data. On a number of real datasets, application of mcImpute yields significant improvements in separation of true zeros from dropouts, cell-clustering, differential expression analysis, cell type separability, performance of dimensionality reduction techniques for cell visualization and gene distribution.

Availability and Implementation:
https://github.com/aanchalMongia/McImpute_scRNAseq

Introduction

In contrast to traditional bulk population based expression studies, single cell transcriptomics provides more precise insights into functioning of individual cells. Over the past few years this powerful tool has brought in transformative changes in the conduct of functional biology [39]. With single cell RNA sequencing (scRNA-seq) we are now able to discover subtypes within seemingly similar cells. This is particularly advantageous for characterizing cancer heterogeneity [27, 34], identification of new rare cell type and understanding the dynamics of transcriptional changes during development [1, 33, 40].

Despite all the goodness, scRNA-seq technologies suffer from a number of sources of technical noise. Most important of these is insufficient input RNA. Due to small quantities transcripts are frequently missed during the reverse transcription step. As a direct consequence, these transcripts are not detected during the sequencing step [12]. Often times the lowly expressed genes are the worst hit. Excluding these genes from analysis may not be the best solution as many of the transcription factors and cell

surface markers are sacrificed in this process [38]. Added to that, variability in dropout rate across individual cells or cell types, works as a confounding factor for a number of downstream analyses [18, 30]. Hicks and colleagues [8] showed, on a number of scRNA-seq datasets, that the first principal components highly correlate with proportion of dropouts across individual transcriptomes. In summary, there is a standing need for efficient methods to impute scRNA-seq datasets.

Very recently, efforts have been made to devise imputation techniques for scRNA-seq data. Most notable of among these are MAGIC [38], scImpute [19] and drImpute [15]. MAGIC uses a neighborhood based heuristic to infer the missing values based on the idea of heat diffusion, altering all gene expression levels including the ones not affected by dropouts. On the other hand, scImpute first estimates which values are affected by dropouts based on Gamma-Normal mixture model and then fills the dropout values in a cell by borrowing information of the same gene in other similar cells, which are selected based on the genes unlikely affected by dropout events. Overall performance of scImpute has been shown to to be superior to MAGIC. Parametric modeling of single cell expression is challenging due to our lack of knowledge about possible sources of technical noise and biases [30]. Moreover, there is clear lack of consensus about the choice of probability density function. Another method, Drimpute, repeatedly identifies similar cells based on clustering, and performs imputation multiple times by averaging the expression values from similar cells, followed by averaging multiple estimations for final imputation. We propose mcImpute, an imputation algorithm for scRNA-seq data which models gene expression as a low rank matrix and sprouts in values in place of dropouts in the process of recovering the full gene expression data from sparse single cell data. This is done by applying soft-thresholding iteratively on singular values of scRNA-seq data. One of the salient features of mcImpute is that it does not assume any distribution for gene expression.

We first evaluate the performance of mcImpute in separating "true zero" counts from dropouts on single cell data of myoblasts [35] (We call it Trapnell dataset). On the same dataset, we assess the impact of imputation on differential genes prediction. We further investigate mcImpute's ability to recover artificially planted missing values in a single cell expression matrix of mouse neurons [37].Accurate imputation should enhance cell type identity i.e., transcriptomic similarity between cells of identical type. We therefore quantify cell type separability as a metric and assess its improvement. In addition to these, we also test the impact of imputation on cell clustering. Four independent real datasets Zeisel, Jurkat-293T, Preimplantation and Usoskin ([43], [41], [40], [37]), for which cell type annotations are available and one dataset, Trapnell, ([35]) for which bulk RNA-seq data has been provided (required for validation of differential genes prediction and separation of "true zeros" from dropouts), are used for this purpose. mcImpute clearly serves as a crucial tool in scRNA-seq pipeline by significantly improving all the above mentioned metrics and outperforming the state-of-the-art imputation methods in majority of experimental conditions.

With the advent of droplet based, high-throughput technologies [23, 41], library depth is being compromised to curb the sequencing cost. As a result, scRNA-seq datasets are being produced with extremely high number of dropouts. We believe that great performance, will make mcImpute the method of choice for imputing scRNA-seq data.

Results/Discussion

We performed numerous experiments to evaluate the efficacy of our proposed imputation technique comparing mcImpute with a number of existing imputation methods for single cell RNA data: scImpute, drImpute and MAGIC.

Dropouts vs true zeros

The inflated number of zero counts in scRNA-seq data could either be biologically driven or due to lack of measurement sensitivity in sequencing. The transcript which is not detected because of failing to get amplified in sequencing step, essentially corresponds to a "false zero" in the finally observed count data and needs to be imputed. A reasonable imputation strategy which has this discriminating property should keep the "true zero" counts (where the genes are truly expressed and have no transcripts from the beginning) untouched, while at the same time attempt to recover the dropouts.

We investigate the performance of mcImpute in distinguishing "true zero" counts from dropouts on Trapnell data [35], for which the bulk-counterpart was available and hence, we could pull out low-to-medium expression genes from the corresponding bulk data for validation. The fraction of zero counts were observed for genes with expression ranging from zero to 500 for unimputed and imputed gene-expression data. It should be noted that an imputed count value ranging from 0-0.5 is taken as an imputed zero, rendering minor flexibility to all imputation techniques.

Given the nature of this analysis, gene filtering in single cell expressions has been skipped. DrImpute could not be taken into account since we could not programatically mute the gene filtering step in its pipeline.

We observe (**figure 2.(e)**) that with low expression genes, all imputation strategies successfully impute the "true zeros" while, as the gene expression amplifies, un-imputed matrix still exhibits large fraction of zeros, which essentially correspond to dropouts and only mcImpute and scImpute are able to curtail the fraction of zeros, thus recovering the dropouts back.

Matrix recovery

In this set of experiments, we study the choice of matrix completion algorithm – matrix factorization (MF) or nuclear norm minimization (NNM). Both the algorithms have been explained in section **Materials and Methods**.

The experiments are carried out on the processed Usoskin dataset [37]. We artificially removed some counts at random (sub-sampling) in the data to mimic dropout cases and used our algorithms (MF and NNM) to impute the missing values. **Figure 3.(a)-(c)** and **table S2** show the variation of Normalized Mean Squared Error (NMSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to compare our two methods for different sub-sampling ratios. This is the standard procedure to compare matrix completion algorithms [11, 25].

We are showing the results for Usoskin dataset, but we have carried out the same analysis for other datasets and the conclusion remained the same. We find that the nuclear norm minimization (NNM) method performs slightly better than the matrix factorization (MF) technique; so we have used NNM as the workhorse algorithm behind mcImpute.

Improvement in clustering accuracy

Correct interpretation of single cell expression data is contingent on accurate delineation of cell types. Bewildering level of dropouts in scRNA-seq data often introduces batch effect, which inevitably traps the clustering algorithm. A reasonable imputation strategy should fix these issues to a great extent. In a controlled setting we therefore examined if the proposed method enhanced clustering outcomes. For this, we ran K-means on first 2 principal component genes of log transformed expression profiles featured in each dataset. Since the prediction from this clustering algorithm tends to change with the choice of initial centroids, which are chosen at random, we analyze the

results on 100 runs of k-means to get reliable and robust results. We set the number of annotated cell types as the value of K for every data. Adjusted Rand Index (ARI) was used to measure the correspondence between the clusters and the prior annotations.

McImpute based re-estimation best separates the four groups of mouse neural single cells from Usoskin dataset and brain cells from Zeisel dataset, and clearly shows comparable improvement on other datasets too (**figure 2.(a)-(d), table S3**). Striking difference between Jurkat and 293T cells made them trivially separable through clustering, leading to same ARI across all 100 runs. Still, mcImpute was able to better maintain the ARI in comparison to other imputation methods.

Improved differential Genes prediction

Optimal imputation of expression data should improve accuracy of differential expression (DE) analysis. It is a standard practice to benchmark DE calls made on scRNA-Seq data against calls made on their matching bulk counterparts [12]. To this end we used a dataset of myoblasts, for which matching bulk RNA-Seq data were also available [35]. For simplicity this dataset has been referred to as the Trapnell dataset. DE and non-DE genes were identified using edgeR [42] package in R.

We used the standard Wilcoxon Rank-Sum test for identifying differentially expressed genes from matrices imputed by various methods. Congruence between bulk and single cell based DE calls were summarized using the Area Under the Curve (AUC) values yielded from the Receiver Operating Characteristic (ROC) curves (**figure 3.(d)**). Among all the methods mcImpute performed best with an AUC of 0.85.

For each method, the AUC value was computed on the identical set of ground truth genes. We had to make an exception only for drImpute as it applies the filter to prune genes in its pipeline. Hence AUC value for drImpute was computed based on a smaller set of ground truth genes.

Improvement in cell type separability

Downstream analysis becomes much easier if expression similarities between cells of identical type are considerably higher than that of cells coming from different subpopulations. To this end, we define cell-type separability score as follows:

For any two cell groups, we first find the median of Spearman correlation values computed for each possible pair of cells within their respective groups. We call the average of the median correlation values the intra-cell type scatter. On the other hand, inter-cell type scatter is defined as the median of Spearman correlation values computed for pairs such that in each pair, cells belong to two different groups. The difference between the intra-cell scatter and inter-cell type scatter is termed as the cell-type separability (CTS) score. We computed CTS scores for two sample cell-type pairs from each dataset. In more than 80 % (13 out of 16) of test cases, mcImpute yielded significantly better CS values (**figure 3.(e)-(h), Table S4**).

Cell visualization

Representing scRNA-seq data visually would involve reducing the gene-expression matrix to a lower dimensional space and then plotting each cell transcriptome in that reduced two or three dimensional space. Two well-known techniques for dimensionality reduction are PCA and t-SNE [9, 22]. It has been shown that t-Distributed Stochastic Neighbor Embedding (t-SNE) is particularly well suited and effective for the visualization of high-dimensional datasets [20]. So, we use t-SNE (**figure 4**) on Usoskin and Zeisel expression matrices to explore the performance of dimensionality reduction, both without and with imputation. The cells are visualized in 2-dimensional space,

coloring each subpopulation by its annotated group, both before and after imputation. To quantify the groupings of cell transcriptomes, we use an unsupervised clustering quality metric, silhouette index. The average silhouette values for each method have been shown in the plot titles (**figure 4**).

T-SNE analysis depicts that mcImpute brings all four groups of mouse neural cells from Usoskin data closest to each other in comparison to other methods and performs fairly well, competing with drImpute on Zeisel data too.

Improvement in distribution of genes

It has been shown that for single-cell gene expression data, in the ideal condition all genes should obey $CV = mean^{-1/2}$ [44] (CV: coefficient of variation), following a Poisson distribution as depicted by the green diagonal line **(figure 5)**. This is because individual transcripts are sampled from a pool of available transcripts for CEL-Seq. This accounts for technical noise component which obeys Poissonian statistics [45], and thus the CV is inversely proportional to the square root of the mean.

We model CV as a function of mean expression for all genes to analyze how various imputation methods affect the relationship between them. The results **(figure 5)** show that both mcImpute and drImpute succeeed to restore the relationship between CV and mean to a great extent (improving the dependency of the CV on the mean expression level to be more consistent with Poissonian sampling noise), while others do not.

Conclusion

As an inevitable consequence of steep decline in single cell library depth, dropout rates in scRNA-seq data have skyroketed. This works as a confounding factor [8], thereby hindering cell clustering and further downstream analyses. The proposed mcImpute algorithm shows remarkable performance on a number of measures including clustering accuracy, cell type separability, differential gene prediction, cell visualization, gene distribution, etc. McImpute can serve as a very crucial component in single-cell RNA seq pipeline.

Currently imputation and clustering are together a piecemeal two step process - imputation followed by clustering. In the future, we would like to incorporate both clustering and imputation as a joint optimization problem.

Materials and methods

Dataset description

We used five scRNA-seq datasets from four different studies for performing various experiments.

- **Jurkat-293T:** This dataset contains expression profiles of Jurkat and 293T cells, mixed *in vitro* at equal proportions (50:50). All $\sim$ 3,300 cells of this data are annotated based on the expressions of cell-type specific markers [41]. Cells expressing CD3D are assigned Jurkat, while those expressing XIST are assigned 293T.
  This dataset is also available at 10x Genomics website.

- **Preimplantation :** This is an scRNA-seq data of mouse preimplantation embryos. It contains expression profiles of $\sim$ 300 cells from zygote, early 2-cell stage, middle 2-cell stage, late 2-cell stage, 4-cell stage, 8-cell stage, 16-cell stage,

early blastocyst, middle blastocyst and late blastocyst stages. The first generation of mouse strain crosses were used for studying monoallelic expression.
We downloaded the count data from Gene Expression Omnibus (GSE45719) [40].

- **Zeisel:** Quantitative single-cell RNAseq has been used to classify cells in the mouse somatosensory cortex (S1) and hippocampal CA1 region based on 3005 single cell transcriptomes [43]. Individual RNA molecules were counted using unique molecular identifiers (UMIs) and confirmed by single-molecule RNA fluorescence in situ hybridization (FISH). A divisive biclustering method based on sorting points into neighborhoods (SPIN) was used to discover molecularly distinct, 9 major classes of cells. Raw data is available under the accession number GSE60361.

- **Usoskin:** This data of mouse neurons [37] was obtained by performing RNA-Seq on 799 dissociated single cells dissected from the mouse lumbar dorsal root ganglion (DRG) distributed over a total of nine 96-well plates. The cell labels (clusters of mouse lumbar DRG-NF, NP, TH, PEP populations) were computationally derived and assigned by performing PCA classification on single mouse neurons. RPM normalized counts are available under the accession number GSE59739.

- **Trapnell**: This is an scRNA-seq data of primary human myoblasts [35]. Differentiating myoblasts were cultured and cells were dissociated and individually captured at 24-hour intervals. 50–100 cells at each of four time points were captured following serum switch using the Fluidigm C1 microfluidic system. This data is available at Gene Expression Omnibus under the accession number GSE52529.

Data preprocessing

Steps involved in preprocessing of raw scRNA-seq data are enumerated below.

- **Data filtering:** It is ensured that data has no bad cells and if a gene was detected with ≥ 3 reads in at least 3 cells we considered it expressed. We ignored the remaining genes.

- **Library-size Normalization:** Expression matrices were normalized by first dividing each read count by the total counts in each cell, and then by multiplying with the median of the total read counts across cells.

- **Log Normalization:** A copy of the matrices were $\log_2$ transformed following addition of 1 as pseudocount.

- **Imputation:** Further, log transformed expression matrix was used as input to mcImpute. The algorithm returns imputed log transformed matrix, normalized matrix (after anti-log operation on imputed log-transformed expressions) and the count matrix after imputation.

Is a gene expression matrix low-rank?

Expression levels of genes at a particular instance is orchestrated by a complex regulatory network. This interdependency is best reflected by pairwise correlations between genes. It has previously been argued that a small number of interdependent biophysical functions trigger the functioning of transcription factors, which in turns influence the expression levels of genes, resulting in a highly correlated data matrix [10].

On the other hand, cells coming from same tissue source also lie on differential grades of variability of a limited number phenotypic characteristics. Therefore, it is just to assume that the gene expression values lie on a low-dimensional linear subspace and the data matrix thus formed may well be thought as a low-rank matrix.

Low-rank matrix completion: Definition

Our problem is to complete a partially observed gene expression matrix X where columns represent genes and rows, individual cells. The complete matrix is constituted by the known and the yet unknown values. We can assume that the single cell data that we have acquired, Y is a sampled version of the complete expression matrix X. Mathematically, this is expressed as,

$$Y = A(X) \tag{1}$$

Here A is the sub-sampling operator. It is a binary mask that has 0's where the counts of complete expression data X has not been observed and 1's where they have been. Our problem is to recover X, given the observations Y, and the sub-sampling mask A. It is known that X is of low-rank.

It should be noted that matrix completion is a well studied framework. In this work, we propose two algorithms for efficient imputation of scRNA-seq expression data- Matrix factorization and Nuclear norm minimization.

Matrix factorization

Matrix factorization is the most straightforward way to address the low-rank matrix completion problem; it has previously been used for finding lower dimensional decompositions of matrices [17]. Say X is of dimensions $m \times n$, but is known to have a rank r ($<m, n$). In that case, one can express $X_{m\times n}$ as a product of two matrices $U_{m\times r}$ and $V_{r\times n}$. Therefore the complete problem (1) can be formulated as,

$$Y = A(X) = A(UV) \tag{2}$$

Estimating U and V from (2) tantamount to recovering X. The two matrices U and V can be solved by minimizing the *Frobenius* norm of the following cost function.

$$\min_{U,V} ||Y - A(UV)||_F^2 \tag{3}$$

Since this is a bi-linear problem, one cannot guarantee global convergence. However it usually works in practice. It has been used for solving recommender systems problems [13], where (3) was solved using stochastic gradient descent (SGD). SGD is not an efficient techniques and requires tuning of several parameters. In this work, we will solve (3) in a more elegant fashion using majorization minimization [31]. The basic MM approach and its geometrical interpretation has been diagrammatically represented (**figure S1**). It depicts the solution path for a simple scalar problem but essentially captures the MM idea.

For our given problem, $J(X) = ||Y - A(X)||_F^2$ the majorization step basically decouples the problem (from A), so that we can solve the optimization problem by solving

$$\min_{U,V} ||B - UV||_F^2 \tag{4}$$

where $B = X_k + A^T(Y - A(X_k))$ at each iteration k.

This (4) is solved by alternating least squares, i.e. while updating U, V is assumed to be constant and while updating V, U is assumed to be constant.

$$U_k \leftarrow \min_U ||B - U_{k-1}V_{k-1}||_F^2 \tag{5}$$

$$V_k \leftarrow \min_V ||B - U_kV_{k-1}||_F^2 \tag{6}$$

Since the log-transformed input (with pseudo count added) expressions would never be negative, we have imposed non-negativity constraint on the recovered matrix X, so that it does not contain any negative values.

The matrix factorization algorithm has been summarized in **algorithm 1**. The initialization of factor V is done by keeping r right singular vectors of X in V, where r is the approximate rank of the expression matrix to be recovered.

Algorithm 1 Matrix completion using matrix factorization

procedure MATRIX-FACTORIZATION(Y, M, r)
  Initialize: $X = rand$, a, V (SVD initialization), *out* and *in*.
    For loop 1, iterate (k)
      $B_k = X_{k-1} + \frac{1}{a}M^T(Y - M \circ X_{k-1})$
      For loop 2, iterate (l)
        $U_l \leftarrow \min_U ||B_k - U_{l-1}V_{l-1}||_F^2$
        $V_l \leftarrow \min_V ||B_k - U_lV_{l-1}||_F^2$
      End loop 2
      $X_k = U_kV_k$
      $X_k \leftarrow X_k^+$
  End loop 1

Nuclear norm minimization

The problem depicted in (3) is non-convex. Hence, there is no guarantee for global convergence. Also one needs to know the approximate rank of the matrix X in order to solve it, which is unknown in this case. To combat this issues, researchers in applied mathematics and signal processing proposed an alternative solution. They would directly solve the original problem (1) with a constraint that the solution is of low-rank. This is mathematically expressed as,

$$\min_X rank(X) \text{ such that } Y{=}A(X) \tag{7}$$

However this turns out to be NP hard problem with doubly exponential complexity. Therefore, studies in matrix completion [6,7] proposed relaxing the NP hard rank minimization problem to its closest convex surrogate: nuclear norm minimization.

$$\min_X ||X||_* \text{ such that } Y{=}A(X) \tag{8}$$

Here the nuclear norm is defined as the sum of singular values of data matrix X. It is the l1 norm of the vector of singular values of X and is the tightest convex relaxation of the rank of matrix, and therefore its ideal replacement.

This is a semi-definite programming (SDP) problem. Usually its relaxed version (Quadratic Program) is solved [5].

$$\min_X ||Y - A(X)||_F^2 + \lambda||X||_* \tag{9}$$

The problem (9) does not have a closed form solution and needs to be solved iteratively.

To solve (9), we invoke MM once more. Here $J(X) = ||Y - A(X)||_F^2 + \lambda||X||_*$, we can express (9) in the following fashion in every iteration k

$$\min_X ||B - X||_F^2 + \lambda||X||_* \tag{10}$$

where $B = X_k + A^T(Y - A(X_k))$.

Using the inequality $||Z_1 - Z_2||_F \geq ||s_1 - s_2||_2$, where s_1 and s_2 are singular values of the matrices Z_1 and Z_2 respective, we can solve the following instead of solving the minimization problem (10).

$$\min_{s_x} ||s_B - s_X||_2^2 + \lambda||s_X||_1 \tag{11}$$

Here s_B and s_X are the singular values of B and X respectively. It has been shown that problem (10) is minimized by soft thresholding the singular values with threshold $\lambda/2$. The optimal update is given by

$$s_X = \begin{cases} s_B + \lambda/2 \text{ when } s_B^{\leq} - \lambda/2 \\ 0 \text{ when } |s_B| \leq \lambda/2 \\ s_B - \lambda/2 \text{ when} s_B \geq \lambda/2 \end{cases} \tag{12}$$

or more compactly by

$$s_X = soft(s_B, \lambda/2) = signum(s_B)max(0, |s_B| - \lambda/2) \tag{13}$$

Algorithm 2 Matrix completion via iterated soft thresholding

procedure MATRIX-IST(Y, M)
  Initialize: $X = rand$, a
    For loop , iterate (k)
      $B_k = X_{k-1} + \frac{1}{a}M^T(Y - M \circ X_{k-1})$
      Compute SVD of $B : B_k = USV^T$
      Soft threshold the singular values: $\Sigma = soft(S, \lambda/2)$ ▷ refer equation 13
      $X_k = U\Sigma V^T$
      $X_k \leftarrow X_k^+$
  End loop 1

We found that the algorithm is robust to values of λ as long as as it is reasonably small (< 0.01).

Here too, we have imposed non-negativity constraint on X since expressions cannot be smaller than zero.

Supporting information

S1 Fig. Schemematic diagram of Majorization-Minimization.

S1 Table. Separation of "true zeros" from dropouts. Fraction of zeros (values between 0 and 0.5) in single cell expression matrix against the median bulk expression. The genes are divided into 10 bins based on median bulk genes expression (first bin has only 0 expression gene

S2 Table. Matrix recovery error. Comparison of NMSE, RMSE and MAE between recovered matrices and unmasked Usoskin data using Nuclear Norm Minimization (NNM) and Matrix Factorization (MF) algorithms at hidden/masked positions

S3 Table. Average Adjusted Rand Index values . Average Adjusted Rand Index values (on 100 runs of PCA followed by k-means) measuring the correspondence between the k-means predicted clusters and the prior annotations.

S4 Table. CTS scores:. Cell type separability (CTS) for any 2 randomly chosen cell groups from each dataset.

Software

The source code of mcImpute is shared at
`https://github.com/aanchalMongia/McImpute_scRNAseq`

Acknowledgments

This work was supported by a grant by DST/INT/CANADA/IC-IMPACT/P-8/2016 awarded to AM and INSPIRE Faculty Award grant DST/INSPIRE/04/2015/003068 given to DS by DST, Govt. of India.

References

1. Biase, F. H. *et al.* (2014). Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell rna sequencing. *Genome research*, **24**(11), 1787–1796.

2. Blakeley, P. *et al.* (2015). Defining the three cell lineages of the human blastocyst by single-cell rna-seq. *Development*, **142**(18), 3151–3165.

3. Blumensath, T. *et al.* (2007). Iterative hard thresholding and l0 regularisation. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, **3**, III–877–III–880.

4. Cai, J.-F. *et al.* (2010). A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, **20**(4), 1956–1982.

5. Candès, E. J. and Plan, Y. (2009). Matrix completion with noise. *CoRR*, **abs/0903.3131**.

6. Candès, E. J. and Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theor.*, **56**(5), 2053–2080.

7. Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Found. Comput. Math.*, **9**(6), 717–772.

8. Hicks, S. C. *et al.* (2015). On the widespread and critical impact of systematic bias and batch effects in single-cell rna-seq data. *bioRxiv*, page 025528.

9. Holland, S. M. (2008). Principal components analysis (pca). *Department of Geology, University of Georgia, Athens, GA*, pages 30602–2501.

10. Kapur, A. *et al.* (2016). Gene expression prediction using low-rank matrix completion. *BMC bioinformatics*, **17**(1), 243.

11. Keshavan, R. H. *et al.* (2010). Matrix completion from a few entries. *IEEE Trans. Inf. Theor.*, **56**(6), 2980–2998.

12. Kharchenko, P. V. *et al.* (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods*, **11**(7), 740–742.

13. Koren, Y. *et al.* (2009). Matrix factorization techniques for recommender systems. *Computer*, **42**(8), 30–37.

14. Kuchaiev, O. and Ginsburg, B. (2017). Training deep autoencoders for collaborative filtering. *arXiv preprint arXiv:1708.01715*.

15. Kwak, I.-Y. *et al.* (2017). Drimpute: Imputing dropout events in single cell rna sequencing data. *bioRxiv*, page 181479.

16. Law, C. W. *et al.* (2014). voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology*, **15**(2), R29.

17. Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press.

18. Li, H. *et al.* (2017). Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nature Genetics*.

19. Li, W. V. and Li, J. J. (2017a). scimpute: accurate and robust imputation for single cell rna-seq data. *bioRxiv*, page 141598.

20. Liu, S. *et al.* (2017). Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, **23**(3), 1249–1268.

21. Love, M. I. *et al.* (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, **15**(12), 550.

22. Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, **9**(Nov), 2579–2605.

23. Macosko, E. Z. *et al.* (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, **161**(5), 1202–1214.

24. Majumdar, A. and Ward, R. (2011). Some empirical advances in matrix completion. *Signal Process.*, **91**(5), 1334–1338.

25. Marjanovic, G. and Solo, V. (2012). On lq optimization and matrix completion. **60**, 5714–5724.

26. Ouyang, Y. *et al.* (2014). *Autoencoder-Based Collaborative Filtering*, pages 284–291. Springer International Publishing, Cham.

27. Patel, A. P. *et al.* (2014). Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, **344**(6190), 1396–1401.

28. Pierson, E. and Yau, C. (2015). Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology*, **16**(1), 241.

29. Ritchie, M. E. *et al.* (2015). limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, **43**(7), e47–e47.

30. Sengupta, D. *et al.* (2016b). Fast, scalable and accurate differential expression analysis for single cells. *bioRxiv*, page 049734.

31. Sun, Y. *et al.* (2017). Majorization-minimization algorithms in signal processing, communications, and machine learning. *Trans. Sig. Proc.*, **65**(3), 794–816.

32. Suzuki, Y. and Ozaki, T. (2017). Stacked denoising autoencoder-based deep collaborative filtering using the change of similarity. *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 498–502.

33. Tang, F. *et al.* (2010). Tracing the derivation of embryonic stem cells from the inner cell mass by single-cell rna-seq analysis. *Cell stem cell*, **6**(5), 468–478.

34. Tirosh, I. *et al.* (2016). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, **352**(6282), 189–196.

35. Trapnell, C. *et al.* (2014). Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nature biotechnology*, **32**(4), 381.

36. Tung, P.-Y. *et al.* (2017). Batch effects and the effective design of single-cell gene expression studies. *Scientific reports*, **7**, 39921.

37. Usoskin, D. *et al.* (2015). Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing. *Nature neuroscience*, **18**(1), 145.

38. van Dijk, D. *et al.* (2017). Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. *BioRxiv*, page 111591.

39. Wagner, A. *et al.* (2016). Revealing the vectors of cellular identity with single-cell genomics. *Nature biotechnology*, **34**(11), 1145–1160.

40. Yan, L. *et al.* (2013). Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology*, **20**(9), 1131–1139.

41. Zheng, G. X. *et al.* (2017). Massively parallel digital transcriptional profiling of single cells. *Nature communications*, **8**, 14049.

42. Zhou, X. *et al.* (2014). Robustly detecting differential expression in rna sequencing data using observation weights. *Nucleic acids research*, **42**(11), e91–e91.

43. Zeisel, A. *et al.* (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, **347**(6226), 1138–1142.

44. Klein, A. M.*et al.* (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**(5), 1187-1201.

45. Grün, D.*et al.* (2014). Validation of noise models for single-cell transcriptomics. *Nature methods*, **11**(6), 637.

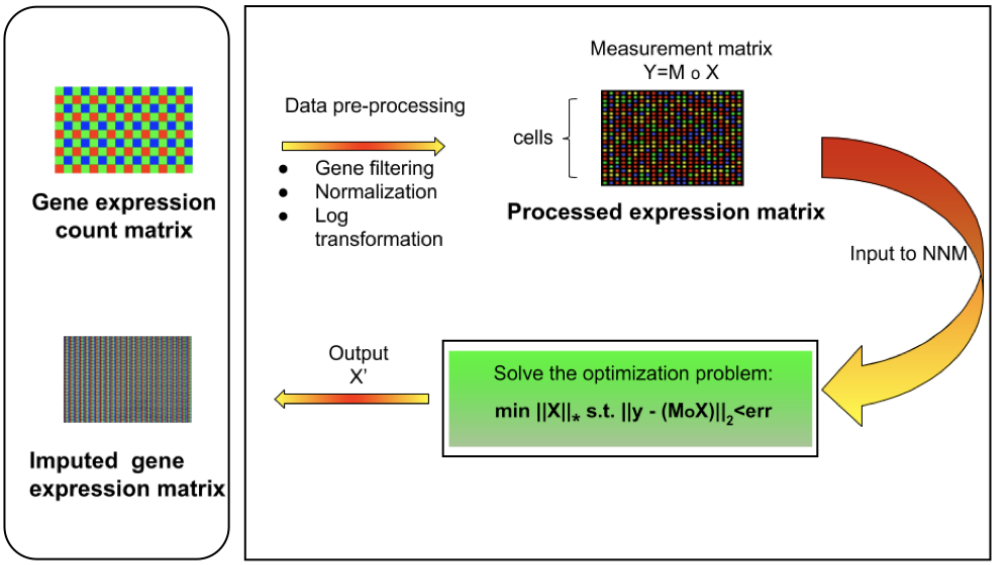


Fig 1. Overview of mcImpute framework for imputing single cell RNA sequencing data. Raw read counts were filtered for significantly expressed genes and then normalized by Library size. Then, the expression data was Log2 transformed (after adding a pseudo count of 1). This pre-processed expression matrix (Y) is treated as the measurement/observation matrix (and fed as input to mcImpute) from which the gene expressions of the complete matrix (X) need to be recovered by solving the non-convex optimization problem minimizing nuclear norm of expression matrix.

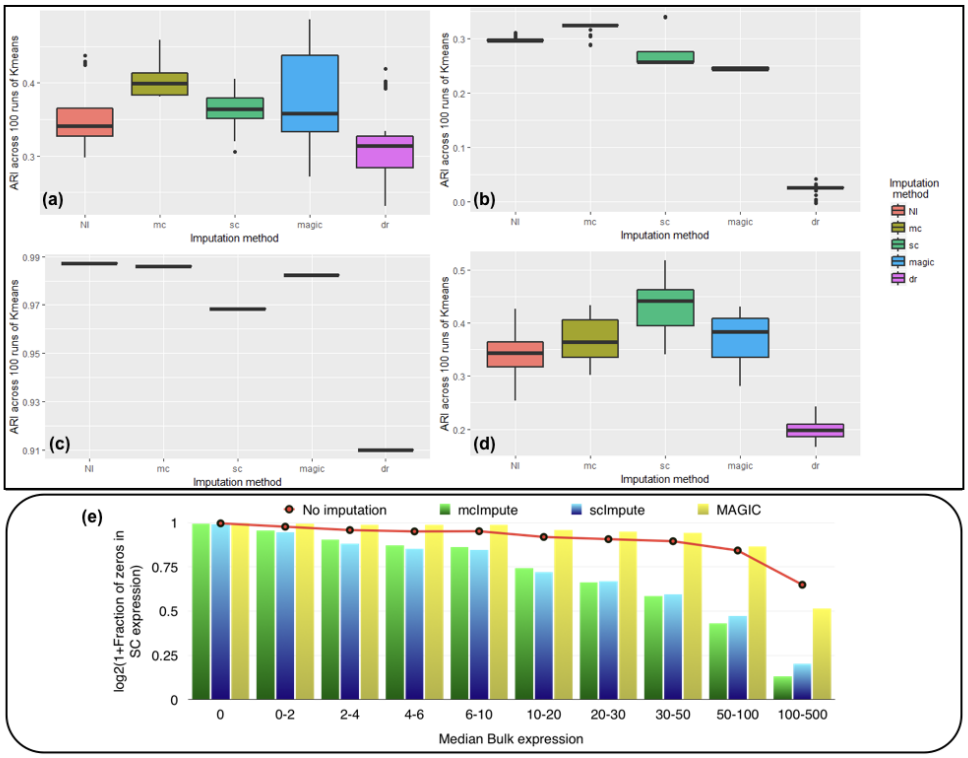


Fig 2. McImpute shows remarkable improvement in clustering of single cells and separation of "true zeros" from dropouts **(a)-(d) Boxplots showing the distribution of ARI** calculated on 100 runs of k-means clustering algorithm on first two principal components of single cell expression matrix for datasets **(a)** Zeisel **(b)** Usoskin **(c)** Jurkat-293T and **(d)** Preimplantation. **(e) Separation of "true zeros" from dropouts:** Plot showing fraction of zero counts (values between 0 and 0.5) in single cell expression matrix against the median bulk expression. The genes are divided into 10 bins based on median bulk genes expression (first bin corresponds to zero expression genes)

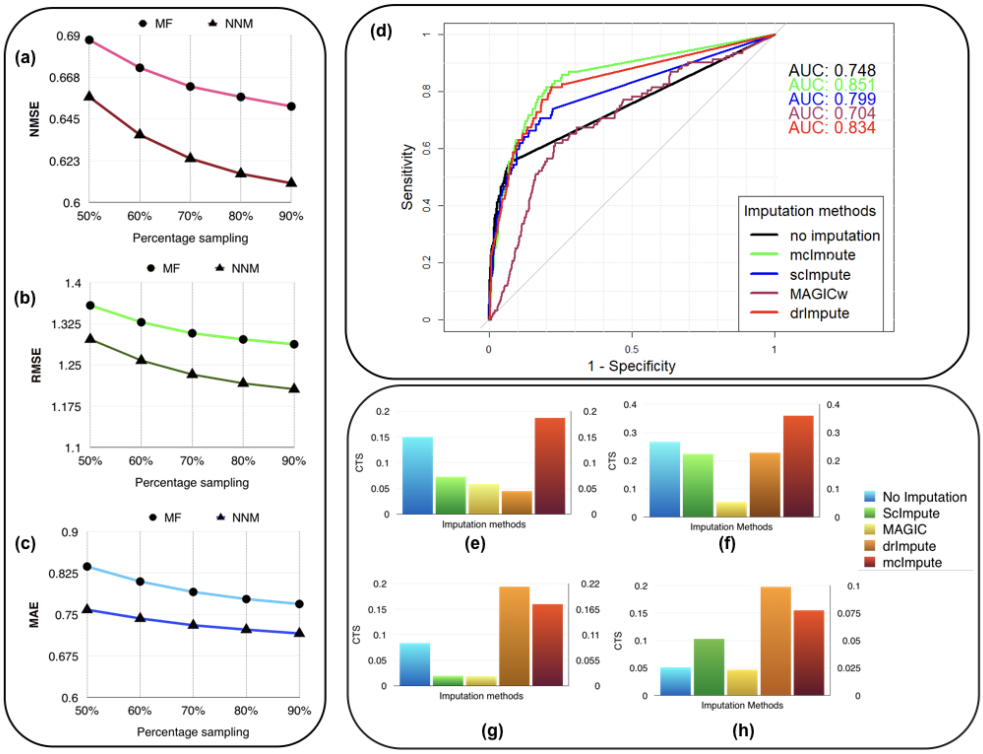
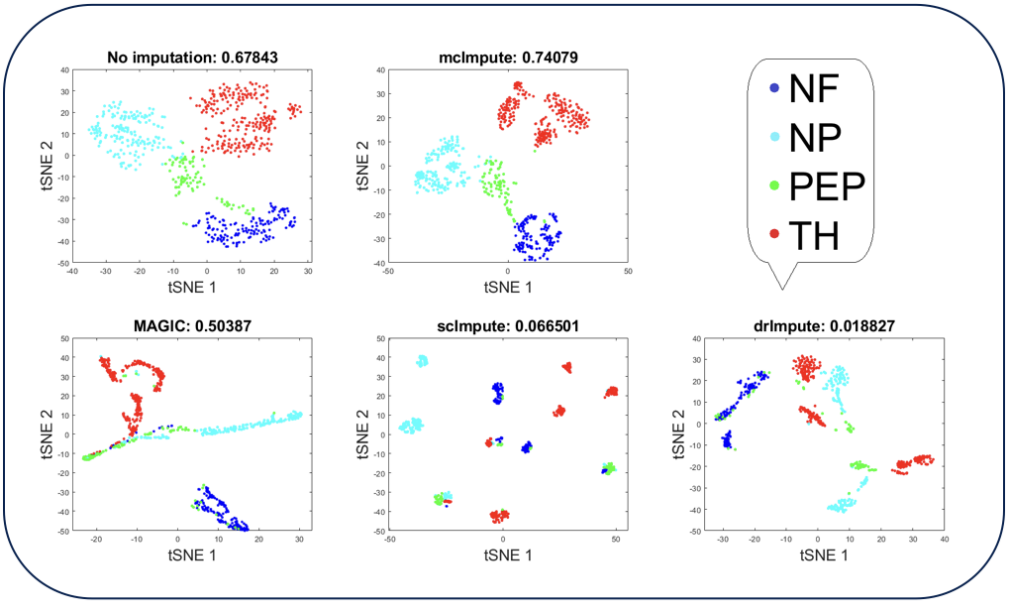
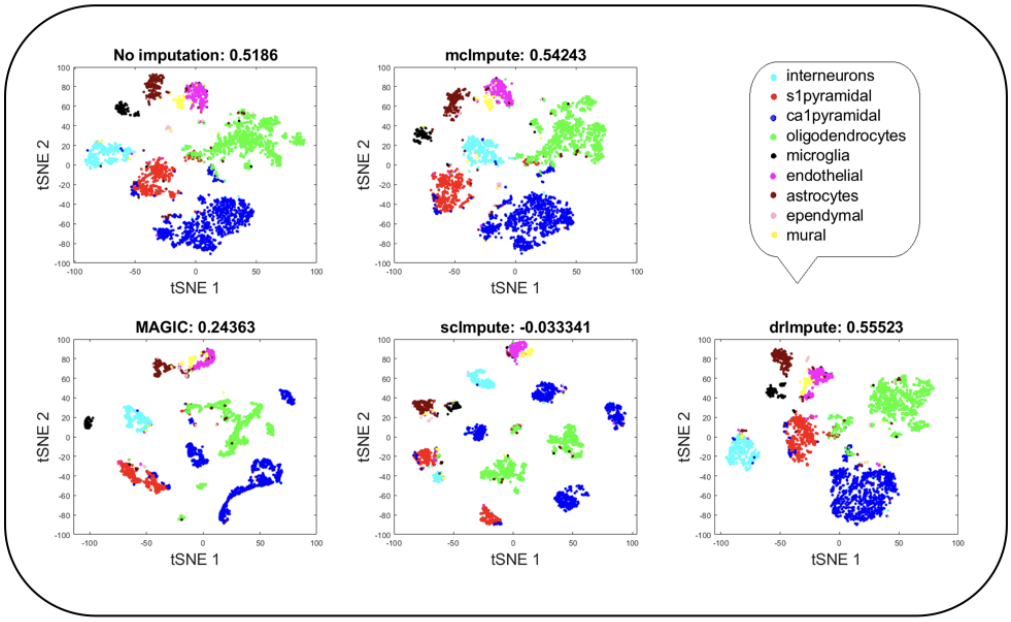


Fig 3. McImpute recovers the original data from their masked version with low error, performs best in prediction of differentially expressed genes and significantly improves CTS score. Variation of **(a)** NMSE, **(b)** RMSE and **(c)** MAE with sampling ration using ARM and MF on Usoskin dataset showing NNM performing better than MF algorithm. **(d)** ROC curve showing the agreement between DE genes predicted from scRNA and matching bulk RNA-Seq data [35]. DE calls were made on expression matrix imputed using edgeR. **(e)-(h)** 2D-Axis bar plot depicting improvement in Cell type separabilities between (e) Jurkat and 293T cells from Jurkat-293T dataset; (f) 8cell and BXC cell types from Preimplantation dataset; (g) S1pyramidal and Ependymal from Zeisel dataset and (h) NP and NF cells from Usoskin dataset. Refer **table S3** for absolute values.

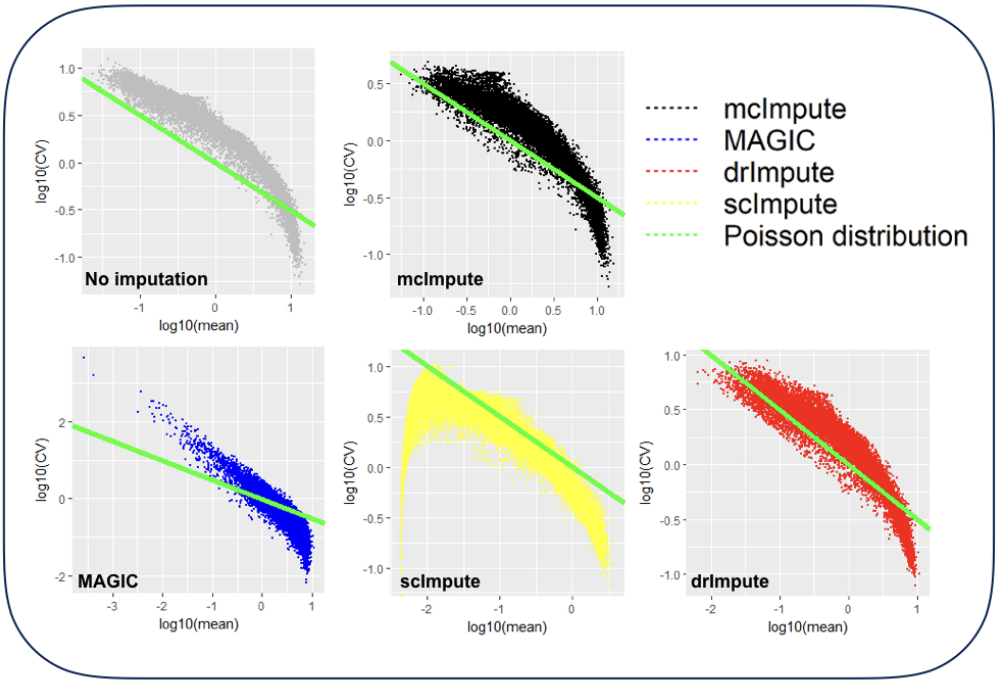


(a) T-SNE visualization of Usoskin dataset before and after imputation. McImpute improves the performance of t-SNE the most in visualizing all groups of mouse neural single cells amongst all imputation strategies.

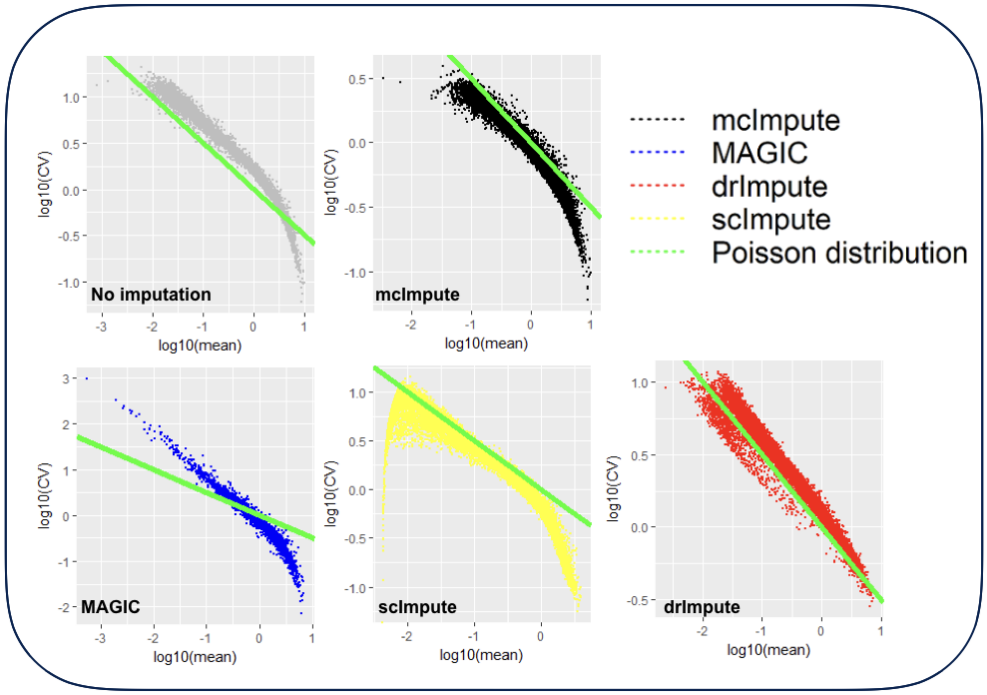


(b) T-SNE visualization of Zeisel dataset before and after imputation. Both mcImpute and drImpute bring brain cells closer, at the same time maintaining the structure of gene-expressions.

Fig 4. Plots showing [t-SNE visualization: average silhouette values] for (a)Usoskin and (b)Zeisel datasets before and after imputation. McImpute significantly improves the visual distinguishability of scRNA-seq datasets in 2-dimensional space in comparison to other methods.



(a) log10(CV) vs log10(mean) plot for Preimplantation dataset before and after imputation.



(b) log10(CV) vs log10(mean) plot for Usoskin dataset before and after imputation.

Fig 5. Plots showing log10(CV) vs log10(mean) relationship between genes for (a) Preimplantation and (b) Usoskin datasets before and after imputation. McImpute and drImpute improves the gene distribution, restoring the relationship between CV and mean to be more consistent with Poisson expression distribution, as expected.