ZoomDeflate: Imputing scRNAseq dropouts through matrix reconstruction

Jeremy P D'Silva, Jaeyoon Kim, Nikhil Shankar

May 1, 2020

1 Introduction

1.1 Single-cell RNA seq

Single-cell RNA sequencing (scRNAseq) experiments are used to measure the gene expression of many cells at single-cell resolution [16]. In the basic setup, single-cells are isolated, and the RNA is reverse-transcribed to cDNA, which is sequenced via next-generation sequencing. This method allows one to determine the relative quantity of RNA expression for each RNA species present in a cell; the data produced by a scRNAseq experiment is a set of vectors in gene expression space (each vector represents the expression of a cell).

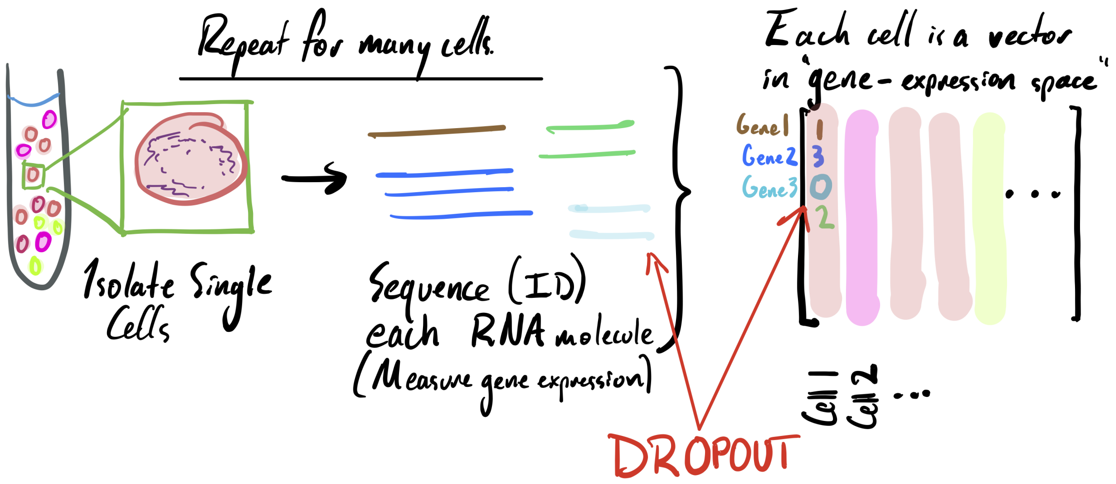

Repeat for many cells.
Each cell is a vector in "gene-expression space"
Gene1
Gene2
Gene3
1
3
0
2
...
Isolate Single Cells
Sequence (ID) each RNA molecule (Measure gene expression)
DROPOUT
Cell 1
Cell 2
...

Figure 1: A simplified schematic of scRNAseq data. After sequencing the transcripts present in cells, associating transcripts to single-cells, and associating the sequences to genes in the human genome, the output is a matrix whose entries are the number of sequencing reads of gene product i in cell j. A dropout event is indicated in the figure: 2 transcripts from "Gene 3" are present in the sample, but the value for gene 3 in the data is 0.

.

The data from scRNAseq experiments samples the underlying distribution of gene expression for the cells under study, and it is useful for answering a variety of biological questions.

For example, several recent studies use scRNAseq to identify novel cell types through their distinct signatures in gene expression space; this sometimes done via clustering algorithms (see [17, 19] for recent examples, or [2] for a review). In a similar vein, scRNAseq can be used to elucidate the gene expression signatures of known cell types or to identify novel marker genes for known cell types. Other research uses scRNAseq to demonstrate expression variability and elucidate its biological relevance, for example, in tumor cell populations [7, 9]. The scRNAseq data is often used to generate plots by dimension-reduction algorithms like tSNE or UMAP, colored by a clustering, in order to visualize cell populations and subpopulations in expression space; this can show the relative abundance of various cell types.

Plate-based scRNAseq methods are susceptible to a type of error called zero-inflation or dropouts [15], wherein transcripts that are present in the cell are not detected by scRNAseq, leading to artificial zeros (dropouts) in the expression data. These artifacts can hinder analyses by skewing the sampled distributions of gene expression. Hence, an active area of bioinformatics research is eliminating zero-inflation from scRNAseq datasets by inferring the true values in place of dropouts [11, 18, 21]. In this paper, we apply matrix reconstruction methods to the dropout problem. The goal of reconstruction is to improve the quality of the scRNAseq data for further analysis, such as the cell-type identification described above.

1.2 Matrix Reconstruction

Matrix reconstruction, or matrix completion, techniques aim to use observations of a small fraction of matrix entries to infer the values of all the remaining unobserved entries. More precisely, let $M \in \mathbb{R}^{m\times n}$ be a matrix, and suppose that we observe some entries of M corresponding to a subset $\Omega \subset [m] \times [n]$ sampled uniformly at random. Clearly, without further assumptions, there are a great number of potential reconstructions $X \in \mathbb{R}^{m\times n}$ which agree with M on the observation set Ω, and *a priori*, there is no reason to prefer one reconstruction over another. However, let us make the additional assumption that M is the unique minimal rank matrix that produces the data we observe. Then we can find M via the minimization

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(X) \\ \text{subject to} & P_\Omega(X) = P_\Omega(M) \end{array} \tag{1}$$

where P_Ω is the operator that projects onto Ω.

The above minimization is known to be NP-hard. A convex relaxation of 1 proposed by Fazel in [5] and by other groups is to minimize the nuclear norm, denoted $\|X\|_*$, which is defined as the sum of the singular values of the matrix (in contrast to rank, which is the number of nonzero singular values). Candès and Recht showed in [4] that the minimization

$$\begin{array}{ll} \text{minimize} & \|X\|_* \\ \text{subject to} & P_\Omega(X) = P_\Omega(M) \end{array} \tag{2}$$

recovers M with high probability given at least $O(n^{1.2} r \log(n))$ observations of M where $\operatorname{rank}(M) = r$. Candès and Tao improved this result to $O(\mu^2 nr \log(n)^6)$, where μ is a parameter describing the extent to which the singular vectors of M are "spread out."

In practice, one applies further relaxations to solve the minimization 2 approximately; a common idea is to solve the following:

$$\text{minimize} \quad \frac{1}{2}\|P_\Omega(X) - P_\Omega(M))\|_F^2 + \lambda\|X\|_* \tag{3}$$

where $\lambda > 0$ is a regularization parameter and $\|X\|_F$ is the Frobenius norm. Intuitively, the first term is a relaxation of the constraint $P_\Omega(X) = P_\Omega(M)$.

In this paper we will solve 3 with a modern heuristic approach due to Hastie et al [6]. Hastie and co-authors present an alternating-least-squares algorithm, called softImpute-ALS. Alternating-least-squares methods break up the expensive minimization into two coupled cheaper parts by factoring the matrix X. They then alternate between these two easier problems and iteratively converge to a solution. Although these algorithms are not as theoretically sound as nuclear norm minimization, they are fast and produce strong results in practice [14].

1.3 Literature on the dropout problem

Matrix reconstruction, and more broadly, low-rank approximation, have recently been applied to single-cell RNA seq data. A paper by Linderman and colleagues [12] uses a low-rank approximation obtained by singular-value decomposition; a paper by Mongia and colleagues applies low-rank reconstruction methods (in particular, using a heuristic to approximately solve the minimization 3), presenting an algorithm called McImpute. Our work on ZoomDeflate in this paper is an independent discovery of McImpute.

Other methods for scRNAseq reconstruction include Bayesian approaches to estimate underlying transcription rates [3] and estimation of the parameters for marginal distributions (expression distributions of each gene) using additional information about gene interaction networks [8]. Intuitively, these methods avoid an explicit assumption of low rank by making stronger assumptions about the probability distributions of expression and observation.

1.4 Summary of paper

In this paper, we present and test three methods to reconstruct zero-inflated single-cell data: ClusterMean (a novel algorithm), ZoomDeflate (independent rediscovery of [1]), and ALRA (the low-rank approximation method developed in [12]). We begin in section 2 by describing our synthetic data sets. In section 3 we explain each method in detail, and then in section 4 we test and compare the methods under the metrics of zero preservation, clustering, and RMSE. Finally, in sections 5 and 6 we discuss our results and conclude.

2 Synthetic Data

Given the preliminary nature of this exploration, we chose to test the matrix reconstruction methods on synthetic data. Our experimental results indicate that working with real biological data is a promising direction for further study.

We generated 6 synthetic data sets, using the Splatter framework [20]. In particular we used the Splat simulation, which models gene expression as a gamma-poisson distribution. Our data sets vary the parameters specifying number of different cell groups, cells, and genes:

$$\texttt{nGroups} \leftarrow \{2, 5, 10\}, \qquad \texttt{(nCells, nGenes)} \leftarrow \{(1000, 5000), (10000, 1000)\}.$$

We hold the remaining parameters constant across simulations whenever possible:

$$\texttt{group.prob} \leftarrow \frac{1}{\texttt{nGroups}}, \qquad \texttt{dropout.mid} \leftarrow 2, \qquad \texttt{dropout.shape} \leftarrow -1.$$

For convenience, we will refer to the data sets by their unique triples

$$\texttt{(nGroups, nCells, nGenes)}.$$

The data set (X, Y, Z) has X different types of cells (uniformly distributed), with Y cells each having Z genes. In the context of matrix reconstruction, X is the approximate rank of the $Y \times Z$ dimensional data.

This simulation produces 3 data sets of the form $(X, 1000, 5000)$ which are roughly 45% sparse and 3 data sets of the form $(X, 10000, 1000)$ which are roughly 70% sparse. The sparsity is due to both biological and technical zeros, we would like our algorithms to preserve the former and reconstruct the latter.

3 Methods

3.1 ClusterMean

ClusterMean is a straightforward algorithm with surprisingly competitive RMSE when compared with the more sophisticated ZoomDeflate and ALRA. It leverages the crucial idea that normalized gene expression vectors of cells from the same group should come from the same distribution.

More precisely, we expect that for each cell type c and gene expressed in that cell g there is some distribution $\mathcal{D}_{(c,g)}$ from which gene expression levels are drawn. Given some (c, g) let us momentarily assume we have *a priori* knowledge of $\mathcal{D}_{(c,g)}$. If a data entry corresponding to (c, g) is dropped out, how can we reconstruct it to minimize RMSE when compared to the ground truth value? Probability theory answers that we should replace all such dropouts with the expected expression level of a variable drawn from $\mathcal{D}_{(c,g)}$.

In practice, we do not have any knowledge of $D_{(c,g)}$. However, given single-cell data we can estimate cell types through clustering and then estimate the expected value of gene expression level by calculating the average observed expression of the gene within a cluster. ClusterMean implements these practicalities using spectral clustering on k-nearest neighbors.

There are a few significant drawbacks of ClusterMean which will be further discussed in section 5. Notably, ClusterMean requires a reasonable estimate of k to perform k-nearest neighbors and it depends on cell types being distinct rather than forming a continuum. Additionally, spectral clustering has an $O(n^3)$ run-time complexity where n is the number of cells. This is significantly slower than other methods studied in this paper.

3.2 ALRA

ALRA is a method for low-rank approximation and dropout imputation developed by Linderman and colleagues in [12]. Given log-normalized scRNAseq data X, ALRA first computes an approximate singular value decomposition:

$$X \approx U\Sigma V^*$$

using a randomized algorithm to determine the first 100 singular values. Then, the approximate SVD is reduced to an even lower-dimensional approximation by setting all singular values below a certain threshold to 0.

The threshold is determined as follows: first, calculate the differences between successive singular values for the last 21 singular values, as well as the mean and standard deviation of those differences (considered the "background"). Next calculate the successive differences between all singular values; the index of the last difference which is more than six standard deviations away from the "background mean" is the index of the last singular value used in the approximation; the smaller singular values are set to zero. If the last singular value used has index k, this gives a rank-k approximation of X, denoted $\hat{X}$:

$$\hat{X} = U\hat{\Sigma}V^*$$

(i.e. we have discarded all but the first k singular values of Σ).

This dimension reduction leads to some negative entries in $\hat{X}$ which need to be reinterprerted since the entries of X should always be within $(0, \infty)$. Linderman and co-authors assume that the biological zeros for each gene are distributed symmetrically around 0; hence, the absolute value of the lowest negative value of expression for a given gene (row of the matrix) is used to threshold the entries of that row (entries below that value are set to 0). The motivation for this step is to preserve biological zeros.

We also consider a modification of the ALRA algorithm, called ALRA-Alt, in which we use the values from ALRA to reconstruct zeros of the observed matrix, and assume the nonzero values have their original values. We feel that this may be a more direct comparison to the ZoomDeflate matrix reconstruction method, which includes an ℓ^2-norm term in the minimization to ensure that the nonzero values of the reconstructed matrix are forced to be close to the nonzero values of the observed matrix.

3.3 ZoomDeflate

Before presenting ZoomDeflate, we would like to give credit to Mongia et al for implementing and testing a nearly identical method last year [1]. We found Mongia's work late during our own exploration, and we are ultimately glad to have independently discovered and affirmed a method that is already part of the research literature.

Algorithm 1 ZoomDeflate

```
procedure ZoomDeflate($X$)    ▷ Reconstruct a gene expression counts matrix
    $X \leftarrow \text{logNormalize}(X)$    ▷ Normalize data for better handling
    $\Omega \leftarrow \{(i, j) \text{ s.t. } X_{ij} \neq 0\}$
    $Y \leftarrow \text{softImpute}(X, \Omega)$    ▷ Hastie et al reconstruction function [13]
    $Y \leftarrow \text{round}(\text{logNormalize}^{-1}(Y))$
    return $Y$
end procedure
```

There are a few points of discussion. First, the seemingly inconspicuous log-normalization step improves RMSE by a couple of orders of magnitude. This normalization is common in biology, however, it is somewhat unclear why it makes the data so much more agreeable. Perhaps it is an artefact of the synthetic data; however, given Mongia's results on biological data this seems unlikely. We are interested in possible answers to this mystery.

Next, note that ZoomDeflate assumes that all zero entries are unobserved. This is an unbiased method for picking Ω, and as we can see in the following section it preserves biological zeros at a frequency comparable to ALRA.

Finally, the main workhorse of ZoomDeflate is the matrix reconstruction algorithm softImpute. Hastie et al introduced softImpute in 2010; it solves the optimization in 3 using an alternating least squares iterative method.

4 Results

4.1 Preservation of Biological Zeros

An important goal of these reconstruction algorithms is to preserve biological zeros; that is, to correctly determine when a zero is due to a lack of gene expression rather than technical dropout. In ALRA, preservation improves as the number of cell groups increases. This may be because as the approximate rank of data increases it becomes easier for ALRA to identify the genes that have biological zeros in one cell group but are nonzero in other cell groups. On the other hand, ZoomDeflate consistently preserves roughly 40% of zeros across all data sets. It would be interesting to carefully test how # Groups, # Cells, and # Genes each impact these results.

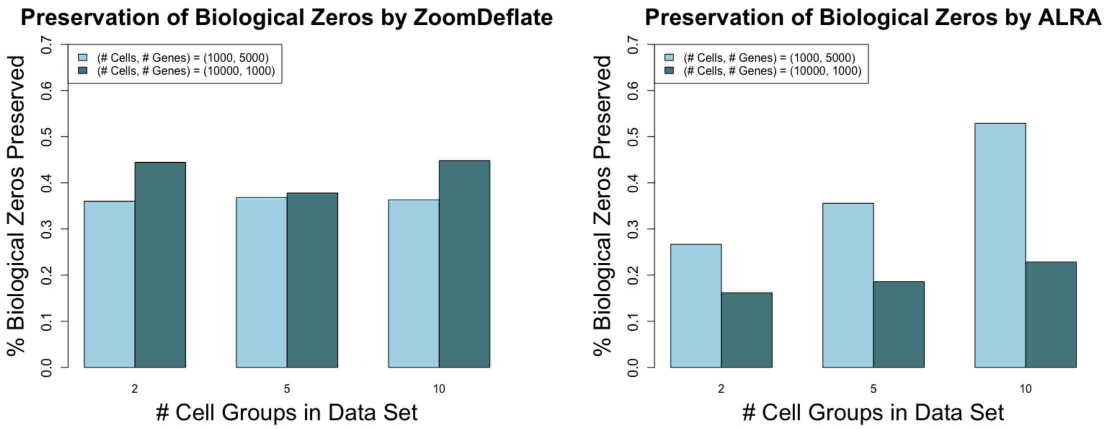


Figure 2: Under this measure ZoomDeflate is relatively agnostic to the shape and rank of the single-cell data. On the other hand, ALRA's performance improves as approximate rank increases and seems to strongly depend on the shape of the data.

4.2 Clustering Performance

Cell type classification is an important application of single-cell data reconstruction. Commonly, cells are classified via clustering algorithms which aim to partition data based on its internal structure. By measuring clustering quality, we gain insight into how well each reconstruction method recovers cell type related structure.

First, we will examine two-dimensional t-SNE plots on k principal components of the reconstructed data (k depends on the data set). This method is non-deterministic and usually involves subjective interpretation, but is valued in the biological community for its intuitive visualization of clustering.

In figure 3, in the t-SNE of the ground truth, almost all cells belong to a distinct cluster associated with a cell type. Unsurprisingly, the t-SNE for observed (zero-inflated) data is far more ambiguous. The gray cells have been split into two distinct clusters. Further, we observe many cells which deviate far from their correct clusters. Applying reconstruction methods to the observed data seems to significantly improve clustering. The t-SNE for each method results in 5 distinct clusters corresponding to one of 5 cell types.

Next, we compute the Adjusted Rand Index (ARI): a quantitative clustering metric to go along with the qualitative measure that the t-SNE plots provide. An ARI score is calculated by comparing true cell types to several k-means clusterings on reconstructed data with. We performed clustering with $k = (\texttt{nGroups} + 1)$, where `nGroups` is the number of cell types. The extra cluster was added to demonstrate that the performance of the clustering does not depend on knowing the exact number of cell types: in practice, the number of cells in the extra cluster was often very small. It takes on values between 0 and 1, scaled such that perfect classification results in ARI of 1 while random classification will achieve an ARI of 0 (on average).

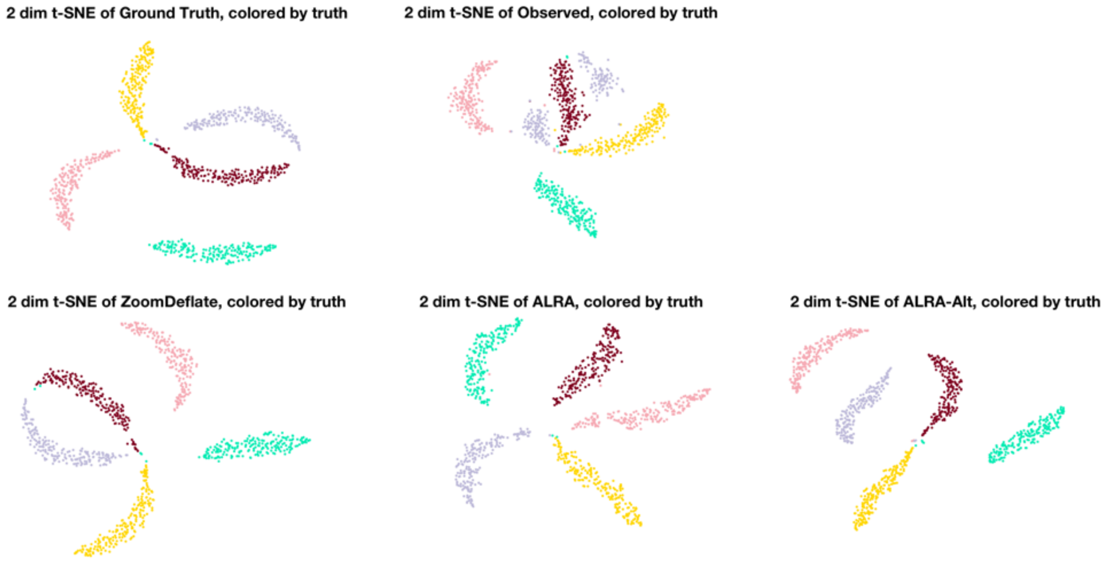
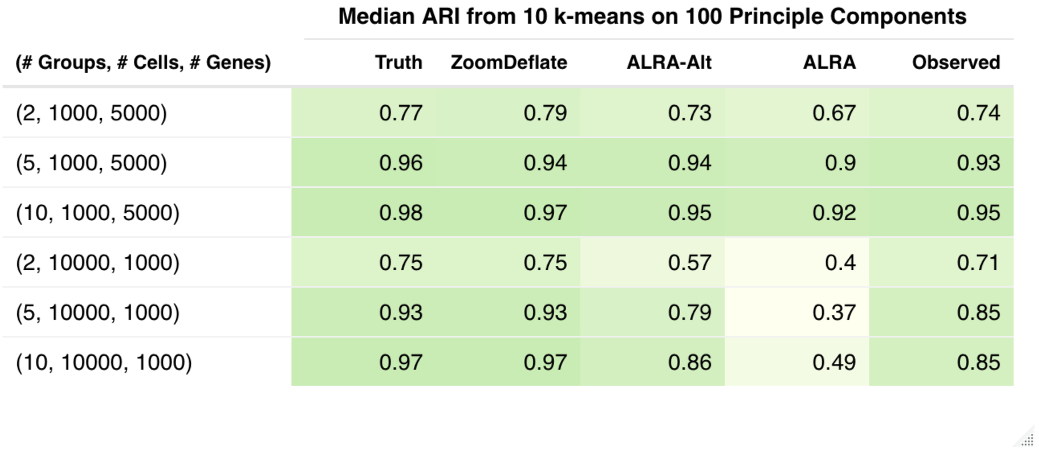


Figure 3: A comparison of t-SNE plots for various reconstruction methods on the data set (5, 1000, 5000). In each plot, cells are colored according to true cell type. The t-SNE algorithm was performed on a PCA of the column-centered data matrix using Rtsne with default settings [10].



	Median ARI from 10 k-means on 100 Principle Components				
(# Groups, # Cells, # Genes)	Truth	ZoomDeflate	ALRA-Alt	ALRA	Observed
(2, 1000, 5000)	0.77	0.79	0.73	0.67	0.74
(5, 1000, 5000)	0.96	0.94	0.94	0.9	0.93
(10, 1000, 5000)	0.98	0.97	0.95	0.92	0.95
(2, 10000, 1000)	0.75	0.75	0.57	0.4	0.71
(5, 10000, 1000)	0.93	0.93	0.79	0.37	0.85
(10, 10000, 1000)	0.97	0.97	0.86	0.49	0.85

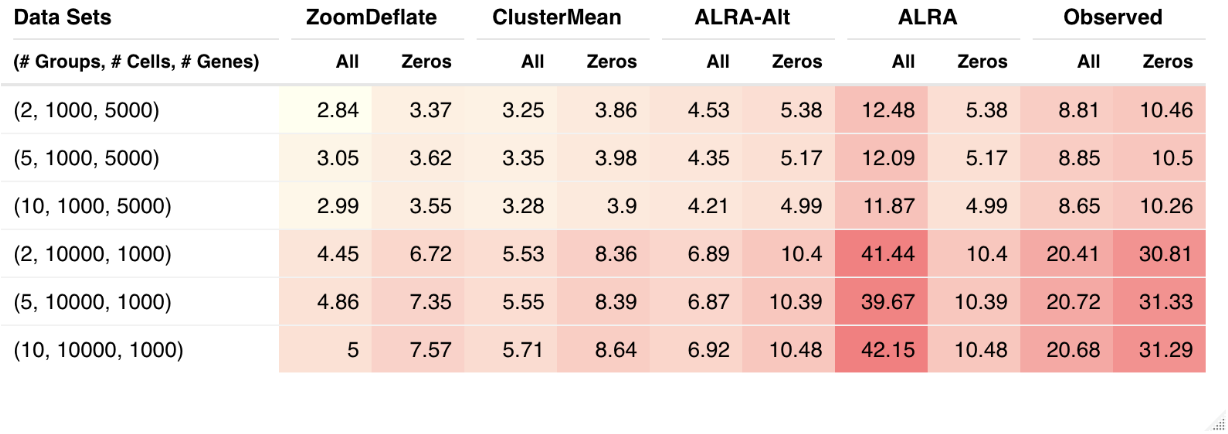
Figure 4: Adjusted Rand Index (ARI) of clustering on different data sets. The clustering was computed by k-means on the first 100 principle components of the normalized data. Because k-means is non-deterministic, we recorded median ARI out of 10 trials.

The clustering on the ground truth data generally achieves the highest median ARI. On the majority of data sets the clustering of the ground truth achieved an ARI of 0.93 or higher, which indicates near perfect classification of cell types. This indicates that the k-means clustering algorithm can successfully cluster the cells into their groups given perfect observation.

Surprisingly, the ZoomDeflate algorithm scores almost identically to the ground truth data. That is, the data reconstructed by ZoomDeflate is as amenable to clustering as the ground truth! Clustering on ALRA, ALRA-Alt, and the Observed data is generally weaker. Notably, it is generally better to cluster using the observed data than it is to cluster using either variant of ALRA.

4.3 Root Mean Squared Performance

Using reconstruction methods to better cluster cell types is helpful, but it does not tell us how well our reconstruction matrix approximates the ground truth. To measure the latter we compute both the RMSE on all data entries values and also the RMSE restricted to the 0 values of the observed data (i.e. value which have a high probability of being dropouts).



Data Sets	ZoomDeflate		ClusterMean		ALRA-Alt		ALRA		Observed	
(# Groups, # Cells, # Genes)	All	Zeros	All	Zeros	All	Zeros	All	Zeros	All	Zeros
(2, 1000, 5000)	2.84	3.37	3.25	3.86	4.53	5.38	12.48	5.38	8.81	10.46
(5, 1000, 5000)	3.05	3.62	3.35	3.98	4.35	5.17	12.09	5.17	8.85	10.5
(10, 1000, 5000)	2.99	3.55	3.28	3.9	4.21	4.99	11.87	4.99	8.65	10.26
(2, 10000, 1000)	4.45	6.72	5.53	8.36	6.89	10.4	41.44	10.4	20.41	30.81
(5, 10000, 1000)	4.86	7.35	5.55	8.39	6.87	10.39	39.67	10.39	20.72	31.33
(10, 10000, 1000)	5	7.57	5.71	8.64	6.92	10.48	42.15	10.48	20.68	31.29

Figure 5: Root mean squared error (RMSE) of different matrix reconstruction algorithms compared to the ground truth matrix. Columns labeled "All" measure the RMSE of the entire matrix. Columns labeled "Zeros" measure the RMSE of entries where observed gene expression was zero.

ZoomDeflate achieves the lowest RMSE on all data sets compared to all other methods. After ZoomDeflate, our naive algorithm ClusterMean comes in a close second. Shockingly, ALRA performs worse than the original observed matrix when considering all entries.

5 Discussions

In this paper, we present several algorithms to reconstruct spurious zeros in the observed data for scRNAseq, and evaluate their performance in preserving biological zeros, in subsequent clustering on reconstructed data, and in their distance from the ground truth. We feel that the theoretical justification for these methods requires further discussion. We also discuss limitations of this work and future directions to ameliorate the limitations and improve the applicability of the results.

In Tao and Candès' theorem on low-rank matrix reconstruction, they assume the observed entries are randomly chosen, either as a Bernoulli process or according to a uniform random distribution on the possible subsets of all matrix entries. However, our model for gene expression matrices does not satisfy either model. In our setup, all zeros of the gene expression matrix are considered to be unobserved entries. These zeros might arise either as dropouts or as biological zeros. In the Splatter model, the probability that a gene in a given cell is selected as a dropout is given by a logistic function depending on the mean expression of that gene; the biological zeros are chosen according to the gamma-poisson distribution for each row.

Even though our observation model does not satisfy the hypotheses of Tao and Candès, we still observe that our matrix reconstruction performs well. This may be due to the fact that a very large number of entries are observed, which could compensate for the fact that observed entries are not completely randomly chosen. We are also reminded of a remark by Professor Anna Gilbert: it is not unheard of that techniques in statistical learning theory are applied where the hypotheses do not hold, and nonetheless achieve some success.

A potential limitation of the work is that our Splatter generated data probably does not contain some structures present in real biological data, either because the data isn't pathological enough (insufficient amount of technical and experimental noise) or because the data fails to capture some underlying biological structure (such as a high level of correlation between genes in a regulatory network). In Splatter, different genes of a cell are uncorrelated (after being column normalized), which is not reflective of the underlying biology. Moreover, our assumption of discrete cell states is a reduction of the true complexity of biological populations. There may be cells in intermediate states, cells differentiating (transitioning between states), or other complicated phenomena. This is related to the low-rank assumption. We think of the rank as approximately corresponding to the number of cell types; a nonlinear transition between cell states would violate this assumption.

A natural next step is to evaluate our methods on some of the benchmark scRNAseq data sets commonly used to evaluate novel analysis methods. We remark that this will require new measurers of success since almost all of our metrics rely on accessing ground truth gene expression levels.

Through the ClusterMean algorithm, we would like to highlight the fact that the gene expression matrix may have a more rigid structure than being approximately low rank. Each column does not need to be expressed as a linear combination of r many vectors. Rather, each column is chosen from *one* of r distributions. Perhaps, it is possible to develop a better algorithm utilizing this rigid structure in single-cell RNA sequencing. The performance of the ClusterMean algorithm heavily depends on how accurately we can cluster our data. The algorithm is aimed for data sets with approximately equal number of cells for each cluster. Furthermore, spectral clustering has very poor run-time, and would not be applicable to very large data sets. In order to apply ClusterMean to more general data, we expect that a modification to our clustering algorithm will be needed.

6 Conclusions

In this project, we applied matrix reconstruction methods to scRNAseq data, and discussed applications for which matrix reconstruction is helpful. We note that although the hypotheses of matrix reconstruction may not be met, the results seem promising, and merit further exploration. Indeed, other groups have applied matrix reconstruction with success, which validates the premise of our study. There may be analyses of scRNAseq data for which matrix reconstruction is helpful or harmful; our perspective is that it is important to evaluate reconstruction methods in the context of the subsequent analyses. As the single-cell revolution continues, we predict that biologists will increasingly rely on tools from statistical learning theory to make meaning of data with sophisticated structure and complicated noise. Studies like these are vital to tease out the assumptions, advantages, and pitfalls of applying methods from statistical learning theory to single-cell-resolution datasets.

References

[1] Mongia Aanchal, Sengupta Debarka, and Majumdar Angshul, *Mcimpute: Matrix completion based imputation for single cell rna-seq data*, Frontiers in Genetics **10** (2019), 9.

[2] Tallulah S Andrews and Martin Hemberg, *Identifying cell populations with scrnaseq*, Molecular aspects of medicine **59** (2018), 114–122.

[3] Jeremie Breda, Mihaela Zavolan, and Erik J van Nimwegen, *Bayesian inference of the gene expression states of single cells from scrna-seq data*, bioRxiv (2019).

[4] Emmanuel J Candès and Benjamin Recht, *Exact matrix completion via convex optimization*, Foundations of Computational mathematics **9** (2009), no. 6, 717.

[5] Maryam Fazel, *Matrix rank minimization with applications* (2002).

[6] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh, *Matrix completion and low-rank svd via fast alternating least squares*, The Journal of Machine Learning Research **16** (2015), no. 1, 3367–3402.

[7] Aaron M Horning, Yao Wang, Che-Kuang Lin, Anna D Louie, Rohit R Jadhav, Chia-Nung Hung, Chiou-Miin Wang, Chun-Lin Lin, Nameer B Kirma, Michael A Liss, et al., *Single-cell rna-seq reveals a subpopulation of prostate cancer cells with enhanced cell-cycle–related transcription and attenuated androgen response*, Cancer research **78** (2018), no. 4, 853–864.

[8] Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang, *Saver: gene expression recovery for single-cell rna sequencing*, Nature methods **15** (2018), no. 7, 539–542.

[9] Charissa Kim, Ruli Gao, Emi Sei, Rachel Brandt, Johan Hartman, Thomas Hatschek, Nicola Crosetto, Theodoros Foukakis, and Nicholas E Navin, *Chemoresistance evolution in triple-negative breast cancer delineated by single-cell sequencing*, Cell **173** (2018), no. 4, 879–893.

[10] Jesse H. Krijthe, *Rtsne: T-distributed stochastic neighbor embedding using barnes-hut implementation*, 2015. R package version 0.15.

[11] Wei Vivian Li and Jingyi Jessica Li, *Issues arising from benchmarking single-cell rna sequencing imputation methods*, arXiv preprint arXiv:1908.07084 (2019).

[12] George C Linderman, Jun Zhao, and Yuval Kluger, *Zero-preserving imputation of scrna-seq data using low-rank approximation*, bioRxiv (2018), 397588.

[13] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani, *Spectral regularization algorithms for learning large incomplete matrices*, Journal of machine learning research **11** (2010), no. Aug, 2287–2322.

[14] Thomas Strohmer, *Measure what should be measured: progress and challenges in compressive sensing*, IEEE Signal Processing Letters **19** (2012), no. 12, 887–893.

[15] Valentine Svensson, *Droplet scrna-seq is not zero-inflated*, Nature Biotechnology (2020), 1–4.

[16] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann, *Exponential scaling of single-cell rna-seq in the past decade*, Nature protocols **13** (2018), no. 4, 599–604.

[17] Kyle J Travaglini, Ahmad N Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V Sit, Stephen Chang, Stephanie D Conley, Yasuo Mori, Jun Seita, et al., *A molecular cell atlas of the human lung from single cell rna sequencing*, bioRxiv (2019), 742320.

[18] Tianyu Wang and Sheida Nabavi, *Single-cell rnaseq imputation based on matrix completion with side information*, 2019 ieee international conference on bioinformatics and biomedicine (bibm), 2019, pp. 2763–2770.

[19] Haojia Wu, Yuhei Kirita, Erinn L Donnelly, and Benjamin D Humphreys, *Advantages of single-nucleus over single-cell rna sequencing of adult kidney: rare cell types and novel cell states revealed in fibrosis*, Journal of the American Society of Nephrology **30** (2019), no. 1, 23–32.

[20] Luke Zappia, Belinda Phipson, and Alicia Oshlack, *Splatter: simulation of single-cell rna sequencing data*, Genome biology **18** (2017), no. 1, 174.

[21] Lihua Zhang and Shihua Zhang, *Comparison of computational methods for imputing single-cell rna-sequencing data*, IEEE/ACM transactions on computational biology and bioinformatics (2018).