scIDS: Single-cell Imputation by combining Deep autoencoder neural networks and Subspace regression

Bang Tran
Computer Science & Engineering
University of Nevada, Reno
Reno, Nevada, USA
bang.t.s@nevada.unr.edu

Quyen Nguyen
Water Engineering and Management
Asian Institute of Technology
Pathum Thani, Thailand
st120032@ait.asia

Sangam Shrestha
Water Engineering and Management
Asian Institute of Technology
Pathum Thani, Thailand
sangam@ait.ac.th

Tin Nguyen*
Computer Science & Engineering
University of Nevada, Reno
Reno, Nevada, USA
tinn@unr.edu

2021 13th International Conference on Knowledge and Systems Engineering (KSE) | 978-1-6654-9975-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/KSE53942.2021.9648664

***Abstract*—Single-cell RNA-sequencing (scRNA-seq) has emerged as a powerful high throughput technique that enables the characterization of transcriptomic profiles at single-cell resolution. However, scRNA-seq data has an excess number of zeros in expressed genes due to a low amount of sequenced mRNA in each cell. This missing detection in a portion of mRNA molecules (dropout) presents a fundamental challenge for various types of data analyses. Here we introduce scIDS, a novel imputation method that is a combination of deep autoencoder neural networks and subspace regression to reliably recover the missing values in scRNA-seq data. We compare scIDS with two widely used methods using eight datasets. Extensive experiments demonstrate that scIDS outperforms existing approaches in improving the identification of cell populations while preserving the biological landscape.**

***Index Terms*—imputation, single-cell, neural networks**

I. Introduction

Single-cell RNA sequencing (scRNA-seq) was first known in 2009 when Tang et al. [1] monitored how individual cells respond to signals and other environmental cues at critical stages of cell-fate. Since then, next generation sequencing (NGS) platforms have revolutionized cell biology and clinical applications with the capability of sequencing millions of cells in parallel at reduced cost [2]. Despite these advances, a major problem for scRNA-seq is the sparsity of the expression matrix with a high number of zero values (dropouts) [3]. These dropout events usually occur due to the low RNA capture rate and failed amplification [4]. Since downstream analyses of scRNA-seq heavily rely on expression measurement's accuracy, it is important to impute the missing values introduced by dropout events. A number of imputation methods have been developed for this purpose. These methods can be classified into two categories: i) statistical modeling of the expression values, and ii) extrapolation of non-zero values using regression techniques.

Methods in the first category include scImpute [5], SAVER [6], BISCUIT [7], and scVI [8]. scImpute models the expression as a mixture of Gaussian (actual expression) and Gamma (dropout) distributions. It then estimates the mixture parameters and dropout values using the EM algorithm. SAVER [6] models read counts as a mixture of Poisson-Gamma and then uses a Bayesian approach to estimate the true expression values. BISCUIT [7] uses the Dirichlet process mixture model [9] to repeatedly perform the processing steps such as normalization, data imputation, and cells clustering by simultaneously inferring clustering parameters, estimating technical variations (e.g., library size), and learning co-expression structures of each cluster. Lastly, scVI uses a probabilistic approach for the normalization and analysis of scRNA-seq data using a hierarchical Bayesian model with conditional distributions specified by deep neural networks.

Methods in the second category include MAGIC [10], DrImpute [11], scScope [12], DCA [13], and DeepImpute [14]. MAGIC imputes zero values using heat diffusion [15] technique to compute the affinity matrix between cells. Then, MAGIC constructs the Markov transition matrix by normalizing and smoothing the computed affinity matrix. Finally, the method multiplies the exponentiated Markov matrix with the original data to obtain the imputed data. DrImpute [11] is based on the cluster ensemble strategy [16] and consensus clustering [17, 18]. It performs clustering to separate data into groups of similar cells and then imputes missing data by averaging expression values of similar cells. The other three methods (scScope, DeepImpute, DCA) rely on deep neural networks to denoise the data and to impute the missing values.

Despite initial success, the quality of the imputed data

978-1-6654-9975-0/21/$31.00 ©2021 IEEE

by statistical methods is determined by the validity of the assumption of the distribution models. Further, these methods usually require excessive computational power, which makes them slow on big datasets. For methods in the second category (regression approaches), the major drawback is that they rely on many parameters to fine-tune their models, which often leads to overfitting. Also, these methods can lead to over-smoothing and may remove the cell-to-cell stochasticity that represents biological meaningful variation in gene expression.

Here we propose a new approach, scIDS, that can reliably impute missing values from single-cell data. Our method consists of two modules. The first module performs data compression and clustering using deep neural networks. This compressed data is considered trustworthy information for imputation. The second module utilizes a z-test to detect genes that are highly impacted by dropouts. Then, the module imputes missing values affected by dropout events by learning the important features patterns in each cell group (identified in the first module). This strategy ensures that the true missing values are imputed by using only highly relevant information. In an extensive analysis using simulation and 8 real scRNA-seq datasets, we demonstrate that scISR improves the quality of single-cell data while preserving the transcriptome landscape.

II. Method

Figure 1 shows the overall analysis pipeline of scIDS. The input of scIDS is an expression matrix in which rows represent genes and columns represent cells. The first module (Figure 1A) filters the genes and compresses the input data into a low-dimensional representation using two autoencoders. Given the compressed data, this module segregates the cells that share similar characteristics into different groups. The second module (Figure 1B) performs a z-score parametric measure to identify which genes need to be imputed. For each group of cells identified from the first module, a generalized linear model will learn from the compressed data to impute the missing data in the high dropout genes set.

A. Compressing data using autoencoders

The input of Module 1 is an already-normalized expression matrix in which rows represent cells while columns represent genes. Given the input matrix, we rescale the data to a range of 0 to 1 as follows:

$$X_{ij} = \frac{M_{ij} - min(M_{i.})}{max(M_{i.}) - min(M_{i.})} \quad (1)$$

where M is the input matrix and X is the normalized matrix.

After the rescaling, we further process the data using an 1-layer autoencoder to filter out genes that do not significantly contribute to differentiate cells. Autoencoder is a self-learning neural network that consists of two core components: an encoder and a decoder. The encoder projects the input onto a lower-dimensional space (compressed data) while the decoder tries to reconstruct the original data from the compressed data. We also constrain the weights of the encoder to be non-negative. The non-negativity constraint ensures that each latent variable in the compressed space is a part-based, additive combination of the input. This technique shrinks the coefficients of less important features to zero while maintaining the non-negative coefficients of the significant features.

After the feature selection step, we obtain a denoised data matrix with the same number of cells that consists of important genes. Here, we further reduce the size of the data by conducting an additional step of dimensional reduction using a modified version of Variational Autoencoder (VAE) [19]. The VAE has the same structure as a standard autoencoder, which consists of an encoder and a decoder. The encoder (f_E) projects the input to a low-dimensional space while the decoder (f_D) reconstructs the original input from the compressed data. Given an expression profile of a cell x, we have $e = f_E(x)$, where e is the low-dimensional representation of x in the bottleneck layer. Instead of using e directly to reconstruct the data, VAE adds two transformations f_μ and f_σ to generate the parameters μ and σ. The new vector z is now sampled from the distribution $N(\mu, \sigma^2)$. The decoder uses z to reconstruct the data: $\bar{x} = f_D(z)$. Adding randomness to z will help the VAE model to avoid overfitting without losing the ability to learn a generalized representation of the input.

We call the second autoencoder a Stacked Variational Autoencoder because we modify the VAE model to generate multiple compressed spaces. Given a list of latent variables, we use a re-parameterization trick [19] to obtain multiple realizations of z as follows: $z = \mu + \sigma * N(0, 1)$. Given the list of latent variables, we use Weighted-based meta-clustering (wMetaC) to generate cells clusters and select the best latent variable as a compressed data M to be used for imputation.

B. Identifying dropouts and imputation

In this section, we aim to determine the set of genes that are likely to be impacted by dropouts. This is an important step to ensure that the missing data is correctly imputed without introducing false signals to the original data.

Our approach is based on the observation that for genes that are not impacted by dropouts, the log-transformed expression values are normally distributed [5]. Therefore, we use z-test to determine whether a zero value is observed by chance or by the impact of dropout events. For each gene g, we use the non-zero expression values to determine the parameters μ and σ of the Gaussian distribution. Next, we use z-test to estimate how likely a zero value occurs, given that the expression values follow the estimated Gaussian distribution. If the chance of observing a zero value is less than the significance threshold (0.05), we conclude that gene g is likely to be affected by dropout. By repeating this process for all genes, we can select a group of genes that are being affected by dropout and we call them as imputable set G.

After conducting the z-test, we obtain a new matrix with the same number of cells (rows), but the columns consist of genes that are highly impacted by dropout. Here, we perform imputation on imputable genes using the shared information within each cell group identified from the first module. For a gene $g_i \in G_i$ (imputable set) that belongs to the cell cluster

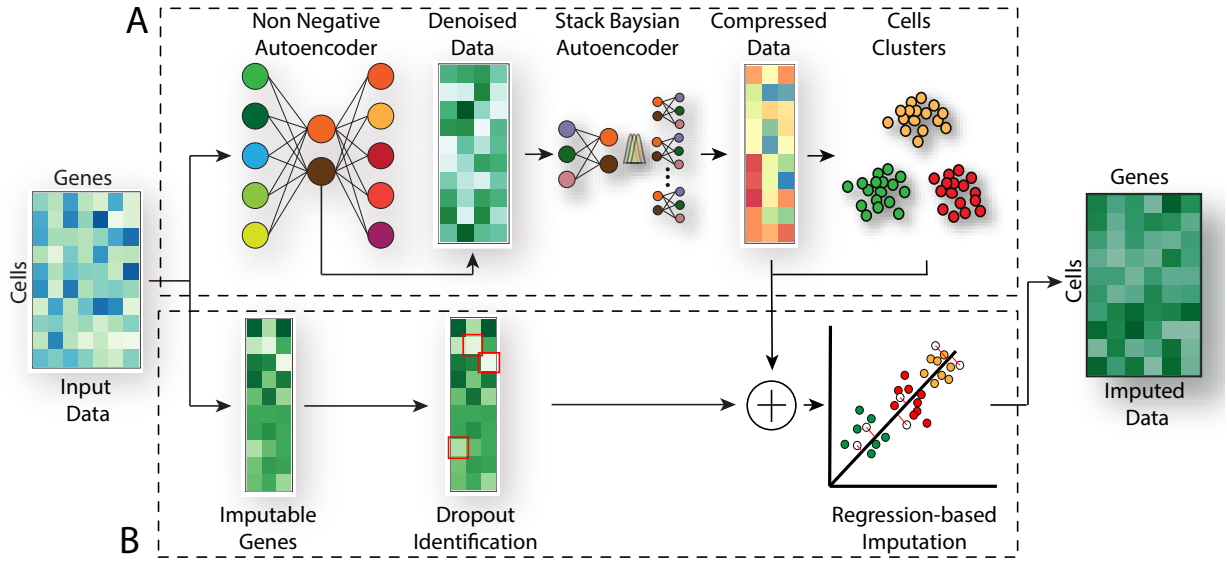

Fig. 1. The overall analysis pipeline of scIDS. The input is a matrix in which rows represent cells and columns represent genes. In the first module (A), we perform features selection, data embedding, and cells clustering using two autoencoders. In the second module (B), we apply the z-test hypothesis testing to determine which genes need to be imputed. From the obtained genes set, we segregate cells into different groups using cluster assignment obtained from module A. For each group of cells, we adopt the generalized linear model to estimate the missing values using embedding data in module A. and we perform. The algorithm outputs the imputed matrix that has the same number of rows and columns as of the input data.

i, let us denote y_i as the non-zero part of g_i. In the first step we calculate the Pearson correlation coefficient of y_i with the corresponding features in the compressed data M_i. We then select 5 features from M_i with the highest correlation coefficients. Denoting $\{m_{ij_1}, \ldots, m_{ij_5}\}$ as the selected features in M_i, we have $\{x_{ij_1}, \ldots, x_{ij_5}\}$ as the vectors obtained from $\{m_{ij_1}, \ldots, m_{ij_5}\}$ that are highly correlated with y_i. Note that each vector x_{ij_n} is a part of m_{ij_n}. We train the generalized linear model in which $\{x_{ij_1}, \ldots, x_{ij_5}\}$ are the predictor variables and y_i is the outcome variable. In our implementation, we adopt the lm function that is available in the *stats* package. Next, we use the trained linear model to estimate the missing values in g_i, using $\{m_{ij_1} \backslash x_{ij_1}, \ldots, m_{ij_5} \backslash x_{ij_5}\}$ as the predictors, where $m_{ij_n} \backslash x_{ij_n}$ is that part of m_{ij_n} that do not belong to x_{ij_n}.

We repeat this imputation process for all genes in each cells cluster generated by the first module. Given the already imputed genes, we merge them by the cells groups to obtain a new matrix that has the same size of the imputable set. Finally, we concatenate the set of good genes with the imputable set to obtain the final imputed data.

III. Result

In this section, we assess the performance of scIDS in the following capabilities: (1) improving the quality of cluster analysis, (2) preserving the cell transcriptome landscape. We compare scIDS with the raw data and two widely used scRNA-seq imputation methods, knn-smoothing [20], and MAGIC [10] using eight scRNA-seq datasets.

Table I shows the details of the eight single-cell datasets (accession ID, number of cells, number of cell types, organism, and single-cell protocol) used in our data analysis.

TABLE I
DESCRIPTION OF THE EIGHT SINGLE-CELL DATASETS USED TO ASSESS THE PERFORMANCE OF SCIDS.

Dataset	Accession ID	Size	K	Organism	Protocol
Pollen [22]	SRP041736	301	4	Human Tissues	SMARTer
Darmanis [23]	GSE67835	466	9	Human Brain	SMARTer
Usoskin [24]	E-MTAB-3321	124	3	Mouse Brain	STRT-Seq
Kolodziejczyk [25]	E-MTAB-2600	268	3	Mouse Embryo	SMARTer
Klein [26]	GSE65525	3,005	4	Mouse Embryo	inDrop
Baron [27]	GSE84133	3,005	14	Human Pancreas	inDrop
Hrvatin [28]	GSE102827	48,266	8	Mouse Visual Cortex	inDrop
Cao [29]	SCP454	90,579	7	Sea Squirt Embryos	10x Genomics

The Cao dataset was downloaded from Broad Institute Single Cell Portal (https://singlecell.broadinstitute.org/single_cell). The Hrvatin dataset was downloaded from NCBI [21]. The remaining six datasets were downloaded from Hemberg Group's website (https://hemberg-lab.github.io/scRNA.seq.datasets). In each of these datasets, the true cell type (labels) is known. This information will be used *a posteriori* to assess the performance of each clustering method. We apply a log transformation (base 2) to rescale the data if the maximum expression value of the data is larger than 100, and we remove the genes that do not express across in any cells.

A. scIDS improves the identification of cells population.

For each of the eight datasets, we use a raw matrix as the input of each imputation method. After imputation, we obtain four matrices: the raw data and three imputed matrices (from knn-smoothing, MAGIC, and scIDS). In order to assess the segregation of the cell types in each matrix, we use k-means to cluster each matrix and then compare the obtained cluster assignments with the known cell types. We use three different metrics to quantify the quality of the clustering result:

TABLE II
COMPARISONS USING ADJUSTED RAND INDEX (ARI).

Dataset	Adjusted Rand Index			
	Raw	knn-smooth	MAGIC	scIDS
Pollen	0.955	0.577	0.564	**0.959**
Darmanis	0.612	0.194	0.298	**0.702**
Usoskin	0.736	0.035	0.276	**0.741**
Kolodziejczyk	0.727	0.203	0.163	**0.996**
Klein	0.984	**0.991**	0.451	0.984
Baron	0.557	0.568	**0.578**	0.559
Hrvatin	0.713	0.822	0.821	**0.832**
Cao	0.376	N/A	0.378	**0.434**
Mean	0.708	0.484	0.441	**0.776**

TABLE III
COMPARISONS USING ADJUSTED MUTUAL INFORMATION (AMI).

Dataset	Adjusted Mutual Information			
	Raw	knn-smooth	MAGIC	scIDS
Pollen	0.95	0.788	0.79	**0.95**
Darmanis	0.722	0.411	0.532	**0.738**
Usoskin	0.716	0.069	0.404	**0.722**
Kolodziejczyk	0.774	0.304	0.296	**0.991**
Klein	0.97	**0.981**	0.579	0.97
Baron	0.681	0.65	**0.701**	0.683
Hrvatin	0.775	0.839	0.848	**0.862**
Cao	0.498	N/A	0.501	**0.551**
Mean	0.761	0.577	0.581	**0.808**

TABLE IV
COMPARISONS USING V-MEASURE.

Dataset	V-Measure Index			
	Raw	knn-smooth	MAGIC	scIDS
Pollen	0.953	0.802	0.799	**0.954**
Darmanis	0.723	0.469	0.563	**0.742**
Usoskin	0.721	0.1	0.473	**0.726**
Kolodziejczyk	0.784	0.364	0.354	**0.992**
Klein	0.973	**0.981**	0.625	0.973
Baron	0.767	0.715	**0.787**	0.769
Hrvatin	0.828	0.845	0.858	**0.877**
Cao	0.515	N/A	0.518	**0.567**
Mean	0.783	0.611	0.622	**0.825**

adjusted Rand index (ARI) [30], adjusted mutual information (AMI) [31] and V-measure [32] (see Appendix A for more details of the three metrics).

Table II shows the ARI values obtained for each method and for the raw data from eight datasets. For each row, the values highlighted in bold indicate the highest ARI value. The cell with "N/A" indicates out of memory or error. In this analysis, scIDS consistently outperforms all comparing methods by maintaining the highest average ARI value of 0.776. This is the highest value compared to 0.708, 0.484, and 441 of raw's, knn-smoothing's, and MAGIC's. More importantly, the ARI values obtained from scDIS are always higher than those obtained from raw data. This vast improvement demonstrates the ability of scIDS in imputing the dropouts without introducing false signals. Unlike scIDS, knn-smoothing and MAGIC have ARI values that are lower than the raw in 5 and 4 datasets. These methods rely on sophisticated models that might lead to overfitting. Moreover, knn-smoothing and MAGIC do not have an efficient mechanism to distinguish whether a low expression value is due to sequencing limitation (i.e., dropout) or indeed due to biological phenomena. Therefore, they are likely to add false signals to the imputed data.

Tables III and IV show the adjusted mutual information and V-measure values obtained for raw data and imputed data using knn-smoothing, MAGIC, and scIDS. Again, the result is similar to the analysis using ARI. scIDS has the highest AMI values in 6 out of 8 datasets with an average AMI value of 0.808 while the average AMI values of raw data, knn-smoothing, and MAGIC are 0.761, 0.577, and 0.581, respectively. The same trend can be seen for V-measure values in Table IV. scIDS has the highest average of V-measure value (0.825). All of the three benchmarking metrics show that scIDS consistently outperforms knn-smoothing and MAGIC in all analyses.

B. scIDS preserves the biological landscape.

In this section, we show that scIDS has a capability of correctly imputing missing values without making a change to the transcriptomics landscapes. Preferably, life scientists impute the data in order to improve the quality of downstream analyses. At the same time, imputation should not completely change the data because of falsely introduced signals, leading to wrong or compromised findings. Since single-cell data are high-dimensional, the common practice is to project the high-dimensional data into a low dimensional space with two or three dimensions. The visualization in 2-D or 3-D helps researchers to interpret the single-cell data more efficiently. To reduce the running time, we first use a fast partial singular value decomposition method [33] to quickly reduce the number of features to 20. Then, we use t-SNE [34], and UMAP [35] to project the compact data into two-dimensional space for visualization.

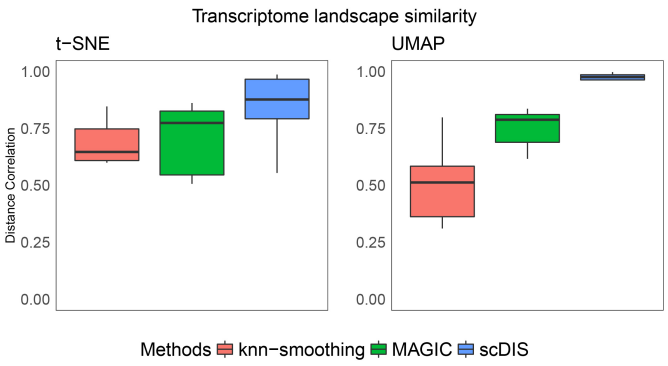

Fig. 2. The similarity between the imputed and original landscapes.

To quantify the similarity between the imputed and original landscapes, we calculate the distance correlation index ($dCor$) [36] for each imputed landscape generated by t-SNE and UMAP. Given X and Y as the 2D representation of the raw and imputed data, $dCor$ is calculated as $dCor = \frac{dCov(X,Y)}{\sqrt{dVar(X)dVar(Y)}}$ where $dCov(X,Y)$ is the distance

covariance between X and Y while $dVar(X)$ and $dVar(Y)$ are distance variances of X and Y. The $dCor$ coefficient takes value between 0 and 1, with the $dCor$ is expected to be 1 for a perfect similarity. Unlike Pearson correlation, $dCor$ measures both the linear and nonlinear associations between X and Y [36]. Especially, $dCor$ remains constant when we rotate the transcriptome landscape. Figure 2 shows the distribution of $dCor$ values for all eight analyzed datasets. In this figure, the left panel shows the values obtained from t-SNE while the right panel shows the values obtained from UMAP representations. The bar plot shows that scIDS has the highest $dCor$ values. A Wilcoxon test also confirms that the correlation $dCor$ obtained from scIDS are significantly higher than the rest ($p = 1.1 \times 10^{-2}$ and 6.11×10^{-5} for t-SNE and UMAP, respectively).

Figure 3 shows the visualization of the raw data and the imputed data for the Baron dataset. Among three comparing methods, the transcriptomics landscape of scIDS is similar to that of the original data (raw), demonstrating that scIDS did not alter the transcriptomics landscape. On the contrary, the transcriptomics landscapes obtained from knn-smoothing and MAGIC are very different from the original data.

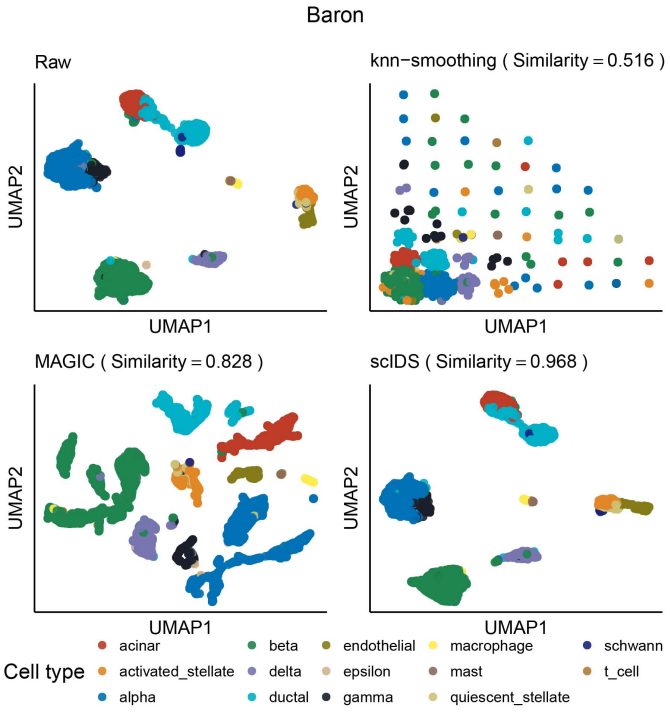

Fig. 3. The visualization of Baron dataset.

IV. Conclusion

In this work, we introduced a new method to reduce the effects of dropout events that frequently happen during the sequencing process of individual cells. The contribution is two-fold. First, we use deep autoencoder neural networks to get the low-dimensional representation of the input data and cell cluster assignment. Second, we impute missing values by using highly correlated genes from the group of cells that share similar biological characteristics. We compared our approach with two widely used methods using eight scRNA-seq datasets. We demonstrated that scIDS outperforms other imputation methods in improving the quality of downstream analyses, including cluster analysis and transcriptome landscape preservation. Generally, scIDS is easy to use and it does not require a sophisticated parameters tuning. The tool can be seamlessly embedded into other single-cell analysis pipelines. Finally, we plan to extend this work to improve the data for network analysis [37–45], meta-analysis [46–49], and the analysis of omics data other than mRNA [50–55].

Acknowledgment

This work was partially supported by NIH NIGMS under grant number GM103440, and by NSF under grant numbers 2001385 and 2019609. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the funding agencies.

Appendix

Rand index (RI) evaluates the similarity between predicted clusters and true cell types. Given P as a set of clusters and Q is a set of true cell types then RI is calculated as:

$$RI = \frac{t+u}{t+u+v+s} = \frac{t+u}{\binom{N}{2}} \tag{2}$$

where t is the number of pairs belonging to the same cell type in Q and are grouped together in the same cluster in P, u is the number of pairs of different cell types in Q and are grouped to different clusters in P, v is the number of pairs of same cell types in Q and are grouped to different clusters in P, s is the number of pairs in different cell types in Q and are grouped together in the same cluster in P, N is the total number of cells, and $\binom{N}{2}$ is the number of possible pairs. The Adjusted Rand Index (ARI) [30] is the corrected-for-chance version of the Rand Index. The ARI values ranged from -1 to 1 in which 0 indicates for a random grouping. The ARI score is calculated as :

$$ARI = \frac{RI - exptected\_RI}{max(RI) - expected\_RI} \tag{3}$$

Adjusted mutual information (AMI) is an adjustment of the mutual information (MI) score to account for random partitioning. It accounts for the fact that the MI is generally higher for two clusters with a larger number of clusters, regardless of whether there is actually more information shared. Given a dataset of n cells with true partition $X = \{X_1, X_2, ..., X_R\}$ of R clusters and predicted partition $Y = \{Y_1, Y_2, ..., Y_C\}$ of C clusters. The mutual information of cluster overlap between X and Y can be summarized as a contingency table $M_{R\times C} = [n_{ij}]$, where $i = 1...R, j = 1...C$, and n_{ij} represents the number of common data point falls into cluster X_i is $p(i) = \frac{|x_i|}{n}$. The entropy associated with the clustering X is calculated as $H(X) = \sum_{i=1}^{R} P(i)logP(i)$. The mutual

information (MI) between two clusters X and Y is calculated as $MI(X,Y) = \sum_{i=1}^{R}\sum_{j=1}^{C} P(i,j)log\frac{P(i,j)}{P(i)P(j)}\frac{n_{ij}}{n}$ where $P(i,j)$ is the cell that is classified to both clusters X_i in X and Y_j in Y. $P(i,j)$ is calculated as $P(i,j) = \frac{|X_i \cap Y_j|}{n}$. MI gives outputs as non-negative values bounded by the entropies $H(X)$ and $H(Y)$ and 0 indicates that there is no cell classified to the same cluster. AMI is defined as follows:

$$AMI(X,Y) = \frac{MI(X,Y) - E\{MI(X,Y)\}}{max\{H(X),H(Y)\} - E\{MI(X,Y)\}} \quad (4)$$

where $E\{MI(X,Y)\}$ is the expected mutual information between two random clusterings. The AMI takes values between 0 and 1 where 0 stands for random clustering and 1 represents a perfect partition.

V-Measure is the harmonic mean between two measures: homogeneity and completeness. Homogeneous clustering is when each cluster has data points belonging to the same class. Complete clustering is when all dat a points belonging to the same class are clustered into the same cluster. Given a set of classes $C = \{C_1, C_2, ..., C_l\}$, a set of cluster $K = \{K_1, K_2, ..., K_m\}$ and the conditional entropy of the class distribution given the identified clustering is computed as $H(C|K)$. The homogeneity is defined as follows:

$$h = \begin{cases} 1 & if H(C|K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & otherwise \end{cases} \quad (5)$$

The completeness is symmetrical to homogeneity. To measure the completeness, the distribution of cluster assignments within each class is assessed. In a perfect clustering, each of these distributions will be completely skewed to a single cluster. Given the homogeneity h and completeness c, the V-measure is computed as the weighted harmonic mean β between homogeneity and completeness:

$$V - measure = \frac{1 + \beta * h * c)}{(\beta * h) + c} \quad (6)$$

if β is greater than 1, completeness is weighted more strongly in the calculation. If β is less than 1, homogeneity is weighted more strongly. Since the computations of homogeneity, completeness and V-measure are completely independent of the number of classes, the number of clusters, the size of the dataset and the clustering algorithm, these measures can be employed for evaluating any clustering solution.

REFERENCES

[1] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, K. Lao, and A. Surani, "mRNA-Seq whole-transcriptome analysis of a single cell," *Nature Methods*, vol. 6, no. 5, pp. 377–382, 2009.

[2] C. Ziegenhain, B. Vieth, S. Parekh, B. Reinius, A. Guillaumet-Adkins, M. Smets, H. Leonhardt, H. Heyn, I. Hellmann, and W. Enard, "Comparative analysis of single-cell RNA sequencing methods," *Molecular Cell*, vol. 65, no. 4, pp. 631–643, 2017.

[3] F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle, "Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells," *Nature Biotechnology*, vol. 33, no. 2, pp. 155–160, 2015.

[4] A. Haque, J. Engel, S. A. Teichmann, and T. Lönnberg, "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications," *Genome Medicine*, vol. 9, no. 1, p. 75, 2017.

[5] W. V. Li and J. J. Li, "An accurate and robust imputation method scImpute for single-cell RNA-seq data," *Nature Communications*, vol. 9, p. 997, 2018.

[6] M. Huang, J. Wang, E. Torre, H. Dueck, S. Shaffer, R. Bonasio, J. I. Murray, A. Raj, M. Li, and N. R. Zhang, "SAVER: gene expression recovery for single-cell RNA sequencing," *Nature Methods*, vol. 15, no. 7, pp. 539–542, 2018.

[7] E. Azizi, S. Prabhakaran, A. Carr, and D. Pe'er, "Bayesian inference for single-cell clustering and imputing," *Genomics and Computational Biology*, vol. 3, no. 1, pp. e46–e46, 2017.

[8] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature Methods*, vol. 15, no. 12, pp. 1053–1058, 2018.

[9] D. Görür and C. E. Rasmussen, "Dirichlet process gaussian mixture models: Choice of the base distribution," *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 653–664, 2010.

[10] D. Van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, B. Bierie, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er, "Recovering gene interactions from single-cell data using data diffusion," *Cell*, vol. 174, no. 3, pp. 716–729, 2018.

[11] W. Gong, I.-Y. Kwak, P. Pota, N. Koyano-Nakagawa, and D. J. Garry, "DrImpute: imputing dropout events in single cell RNA sequencing data," *BMC Bioinformatics*, vol. 19, p. 220, 2018.

[12] Y. Deng, F. Bao, Q. Dai, L. F. Wu, and S. J. Altschuler, "Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning," *Nature Methods*, vol. 16, no. 4, pp. 311–314, 2019.

[13] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell rna-seq denoising using a deep count autoencoder," *Nature Communications*, vol. 10, p. 390, 2019.

[14] C. Arisdakessian, O. Poirion, B. Yunits, X. Zhu, and L. X. Garmire, "DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data," *Genome Biology*, vol. 20, no. 1, pp. 1–14, 2019.

[15] Z. I. Botev, J. F. Grotowski, D. P. Kroese *et al.*, "Kernel density estimation via diffusion," *The Annals of Statistics*, vol. 38, no. 5, pp. 2916–2957, 2010.

[16] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[17] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1-2, pp. 91–118, 2003.

[18] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hamberg, "SC3: consensus clustering of single-cell RNA-seq data," *Nature Methods*, vol. 14, no. 5, pp. 483–486, 2017.

[19] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114*, 2013.

[20] F. Wagner, Y. Yan, and I. Yanai, "K-nearest neighbor smoothing for high-throughput single-cell rna-seq data," *BioRxiv*, p. 217737, 2017.

[21] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko, A. Yefanov, H. Lee, N. Zhang, C. L. Robertson, N. Serova, S. Davis, and A. Soboleva, "NCBI GEO: archive for functional genomics data sets–update," *Nucleic Acids Research*, vol. 41, no. D1, pp. D991–D995, 2013.

[22] A. A. Pollen, T. J. Nowakowski, J. Shuga, X. Wang, A. A. Leyrat, J. H. Lui, N. Li, L. Szpankowski, B. Fowler, P. Chen, N. Ramalingam, G. Sun, M. Thu, M. Norris, R. Lebofsky, D. Toppani, D. W. Kemp Ii, M. Wong, B. Clerkson, B. N. Jones, S. Wu, L. Knutsson, B. Alvarado, J. Wang, L. S. Weaver, A. P. May, R. C. Jones, M. A. Unger, A. R. Kriegstein, and J. A. A. West, "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex," *Nature Biotechnology*, vol. 32, no. 10, pp. 1053–1058, 2014.

[23] S. Darmanis, S. A. Sloan, Y. Zhang, M. Enge, C. Caneda, L. M. Shuer, M. G. H. Gephart, B. A. Barres, and S. R. Quake, "A survey of human brain transcriptome diversity at the single cell level," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 112, no. 23, pp. 7285–7290, 2015.

[24] D. Usoskin, A. Furlan, S. Islam, H. Abdo, P. Lönnerberg, D. Lou, J. Hjerling-Leffler, J. Haeggström, O. Kharchenko, P. V. Kharchenko, S. Linnarson, and P. Ernfors, "Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing," *Nature Neuroscience*, vol. 18, no. 1, pp. 145–153, 2015.

[25] P. Brennecke, S. Anders, J. K. Kim, A. A. Kolodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. A. Teichmann, J. C. Marioni, and M. G. Heisler, "Accounting for technical noise in single-cell RNA-seq experiments," *Nature Methods*, vol. 10, no. 11, pp. 1093–1095, 2013.

[26] A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner, "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells," *Cell*, vol. 161, no. 5, pp. 1187–1201, 2015.

[27] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, and I. Yanai, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure," *Cell Systems*, vol. 3, no. 4, pp. 346–360, 2016.

[28] S. Hrvatin, D. R. Hochbaum, M. A. Nagy, M. Cicconet, K. Robertson, L. Cheadle, R. Zilionis, A. Ratner, R. Borges-Monroy, A. M. Klein, B. L. Sabatini, and M. E. Greenberg, "Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex," *Nature Neuroscience*, vol. 21, no. 1, pp. 120–129, 2018.

[29] C. Cao, L. A. Lemaire, W. Wang, P. H. Yoon, Y. A. Choi, L. R. Parsons, J. C. Matese, M. Levine, and K. Chen, "Comprehensive single-cell transcriptome lineages of a proto-vertebrate," *Nature*, vol. 571, no. 7765, pp. 349–354, 2019.

[30] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[31] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

[32] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 410–420.

[33] J. Baglama, L. Reichel, and B. W. Lewis, *irlba: Fast Truncated Singular Value Decomposition and Principal Components Analysis for Large Dense and Sparse Matrices*, 2018, r package version 2.3.2. [Online]. Available: https://CRAN.R-project.org/package=irlba

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[35] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell, "Dimensionality reduction for visualizing single-cell data using UMAP," *Nature Biotechnology*, vol. 37, no. 1, pp. 38–44, 2019.

[36] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[37] H. Nguyen, D. Tran, B. Tran, B. Pehlivan, and T. Nguyen, "A comprehensive survey of regulatory network inference methods using single-cell RNA sequencing data," *Briefings in Bioinformatics*, vol. 22, no. 3, pp. 1–15, 2021.

[38] T.-M. Nguyen, A. Shafi, T. Nguyen, and S. Draghici, "Identifying significantly impacted pathways: a comprehensive review and assessment," *Genome Biology*, vol. 20, p. 203, 2019.

[39] A. Shafi, T. Nguyen, A. Peyvandipour, H. Nguyen, and S. Draghici, "A multi-cohort and multi-omics meta-analysis framework to identify network-based gene signatures," *Frontiers in Genetics*, vol. 10, p. 159, 2019.

[40] A. Shafi, T. Nguyen, A. Peyvandipour, and S. Draghici, "GSMA: an approach to identify robust global and test Gene Signatures using Meta-Analysis," *Bioinformatics*, vol. 36, no. 2, pp. 487–495, 2019.

[41] H. Nguyen, S. Shrestha, D. Tran, A. Shafi, S. Draghici, and T. Nguyen, "A comprehensive survey of tools and software for active subnetwork identification," *Frontiers in Genetics*, vol. 10, p. 155, 2019.

[42] H. Nguyen, D. Tran, J. M. Galazka, S. V. Costes, A. Beheshti, S. Draghici, and T. Nguyen, "CPA: A web-based platform for consensus pathway analysis and interactive visualization," *Nucleic Acids Research*, vol. 49, no. W1, pp. W114–W124, 2021.

[43] T. Nguyen, C. Mitrea, and S. Draghici, "Network-based approaches for pathway level analysis," *Current Protocols in Bioinformatics*, vol. 61, no. 1, pp. 8–25, 2018.

[44] J. Tanevski, T. Nguyen, B. Truong, N. Karaiskos, M. E. Ahsen, X. Zhang, C. Shu, K. Xu, X. Liang, Y. Hu, H. V. Pham, L. Xiaomei, T. D. Le, A. L. Tarca, G. Bhatti, R. Romero, N. Karathanasis, P. Loher, Y. Chen, Z. Ouyang, D. Mao, Y. Zhang, M. Zand, J. Ruan, C. Hafemeister, P. Qiu, D. Tran, T. Nguyen, A. Gabor, T. Yu, J. Guinney, E. Glaab, R. Krause, P. Banda, DREAM SCTC Consortium, G. Stolovitzky, N. Rajewsky, J. Saez-Rodriguez, and P. Meyer, "Gene selection for optimal prediction of cell position in tissues from single-cell transcriptomics data," *Life Science Alliance*, vol. 3, no. 11, 2020.

[45] E. Cruz, H. Nguyen, T. Nguyen, and I. Wallace, "Functional analysis tools for post-translational modification: a post-translational modification database for analysis of proteins and metabolic pathways," *The Plant Journal*, vol. 99, no. 5, pp. 1003–1013, 2019.

[46] T. Nguyen, R. Tagett, M. Donato, C. Mitrea, and S. Draghici, "A novel bi-level meta-analysis approach-applied to biological pathway analysis," *Bioinformatics*, vol. 32, no. 3, pp. 409–416, 2016.

[47] T. Nguyen, C. Mitrea, R. Tagett, and S. Draghici, "DANUBE: Data-driven meta-ANalysis using UnBiased Empirical distributions - applied to biological pathway analysis," *Proceedings of the IEEE*, vol. 105, no. 3, pp. 496–515, 2017.

[48] T. Nguyen, D. Diaz, R. Tagett, and S. Draghici, "Overcoming the matched-sample bottleneck: an orthogonal approach to integrate omic data," *Scientific Reports*, vol. 6, p. 29251, 2016.

[49] T. Nguyen, A. Shafi, T.-M. Nguyen, A. G. Schissler, and S. Draghici, "NBIA: a network-based integrative analysis framework–applied to pathway analysis," *Scientific Reports*, vol. 10, p. 4188, 2020.

[50] T. Nguyen, R. Tagett, D. Diaz, and S. Draghici, "A novel approach for data integration and disease subtyping," *Genome Research*, vol. 27, no. 12, pp. 2025–2039, 2017.

[51] H. Nguyen, S. Shrestha, S. Draghici, and T. Nguyen, "PINSPlus: A tool for tumor subtype discovery in integrated genomic data," *Bioinformatics*, vol. 35, no. 16, pp. 2843–2846, 2019.

[52] D. Tran, H. Nguyen, U. Le, G. Bebis, H. N. Luu, and T. Nguyen, "A novel method for cancer subtyping and risk prediction using consensus factor analysis," *Frontiers in Oncology*, vol. 10, p. 1052, 2020.

[53] M. P. Menden, D. Wang, M. J. Mason, B. Szalai, K. C. Bulusu, Y. Guan, T. Yu, J. Kang, M. Jeon, R. Wolfinger, T. Nguyen, M. Zaslavskiy, AstraZeneca-Sanger Drug Combination DREAM Consortium, I. S. Jang, Z. Ghazoui, M. E. Ahsen, R. Vogel, E. C. Neto, T. Norman, E. K. Y. Tang, M. J. Garnett, G. Y. Di Veroli, C. Zwaan, S. Fawell, G. Stolovitzky, J. Guinney, J. R. Dry, and J. Saez-Rodriguez, "Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen," *Nature Communications*, vol. 10, p. 2674, 2019.

[54] J. C. Stansfield, D. Tran, T. Nguyen, and M. G. Dozmorov, "R tutorial: Detection of differentially interacting chromatin regions from multiple Hi-C datasets." *Current Protocols in Bioinformatics*, vol. 66, no. 1, pp. e76–e76, 2019.

[55] A. Shafi, C. Mitrea, T. Nguyen, and S. Draghici, "A survey of the approaches for identifying differential methylation using bisulfite sequencing data," *Briefings in Bioinformatics*, vol. 19, no. 5, pp. 737–753, 2018.