Single-cell RNA-seq data imputation using Feature Propagation

Sukwon Yun [*1] **Junseok Lee** [*1] **Chanyoung Park** [1]

Abstract

While single-cell RNA sequencing provides an understanding of the transcriptome of individual cells, its high sparsity, often termed dropout, hampers the capture of significant cell-cell relationships. Here, we propose scFP (single-cell Feature Propagation), which directly propagates features, i.e., gene expression, especially in raw feature space, via cell-cell graph. Specifically, it first obtains a warmed-up cell-gene matrix via Hard Feature Propagation which fully utilizes known gene transcripts. Then, we refine the k-Nearest Neighbor (kNN) of the cell-cell graph with a warmed-up cell-gene matrix, followed by Soft Feature Propagation which now allows known gene transcripts to be further denoised through their neighbors. Through extensive experiments on imputation with cell clustering tasks, we demonstrate our proposed model, scFP, outperforms various recent imputation and clustering methods. The source code of scFP can be found at `https://github.com/Junseok0207/scFP`.

1. Introduction

Single-cell RNA-sequencing (scRNA-seq) analysis has attracted significant attention due to its property to profile transcriptome-wide gene expression at single-cell resolution. It allows researchers to perform various analyses, including identifying cell types (Tian et al., 2019; Lee et al., 2023), and inferring cell trajectories (Trapnell et al., 2014; Zhang et al., 2023). However, analyzing scRNA-seq data is challenging due to the noisy nature of the gene expression. Specifically, scRNA-seq data often suffer from low transcript capture, referred to as dropout phenomena (Hicks et al., 2018), which causes the occurrence of false zero values. Furthermore, the observed non-zero expression values also suffer from biologically irrelevant signals, such as batch effects (Shaham et al., 2017) and transcriptional noise (Wagner et al., 2016).

[*]Equal contribution [1]KAIST. Correspondence to: Chanyoung Park <cy.park@kaist.ac.kr>.

The 2023 ICML Workshop on Computational Biology. Baltimore, Maryland, USA, 2023.

Many works have been proposed to denoise the scRNA-seq data by imputing the dropout values. Among them, smoothing-based methods assume cells with similar expression profiles will likely share similar underlying biological characteristics. By leveraging this assumption, these methods impute gene expression values by forcing the expression values of neighboring cells to be more similar. Specifically, DrImpute (Gong et al., 2018) averages the expression values based on a pre-calculated cluster, and MAGIC (van Dijk et al., 2018) performs a diffusing process on the calculated Markov affinity-based graph. Despite its effectiveness in reducing noise from various factors, there are some limitations to be considered: 1) it can decrease the meaningful cell variability by smoothing expression values across incorrect cell groups when neighboring cells are misdefined, and 2) noise expression values could be spread during smoothing when the observed expression values are noisy, which can be severe if it contains many false zero values due to the dropout phenomena.

Recently, most researchers are focused on the imputation methods that utilize deep neural networks (DNNs) to reconstruct gene expressions. These lines of methods utilize an autoencoder architecture, where the cell representation is learned through an encoder, and then impute values by passing them through the decoder layer. Specifically, DCA (Eraslan et al., 2019) reconstructs the gene expression by assuming zero-inflated negative binomial (ZINB) distribution and AutoClass (Li et al., 2022) further learns the classifier by providing pseudo-labels generated by pre-clustering. Furthermore, scGCL (Xiong et al., 2023), inspired by AFGRL (Lee et al., 2022), leverages relationship information between cells using graph neural networks (GNNs) on a cell-cell graph and learns cell representations in a self-supervised manner. However, despite their complex modeling, the output of these methods often shows poor performance on the subsequent downstream tasks compared to that of raw expression values. We argue that this is because effectively optimizing complex DNN models on a given dataset requires appropriate hyper-parameter choices, which can be challenging in the context of imputation tasks due to the unsupervised nature.

In this paper, we propose scFP, a simple yet effective imputation method for scRNA-seq data with a bi-level feature propagation scheme. Specifically, we first impute the zero

values using the feature propagation on the initially defined kNN graph while preserving the non-zero values that contain relatively lower noise to prevent the contamination of biological signals from prevalent false zero values. Using this warmed-up data, the kNN graph is further refined to capture the correct neighbors of each cell. We then apply another feature propagation module to denoise the bias or noise on non-zero values. Through experiments on both real and simulated scRNA-seq datasets, we demonstrate the effectiveness of scFP.

2. Methods

Notation. Given a cell-gene feature matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, where N and M are the number of cells and genes, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a cell-cell graph. $\mathcal{V} = \{v_1, ..., v_N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the set of nodes and edges, respectively. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix with $\mathbf{A}_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$ and $\mathbf{A}_{ij} = 0$ otherwise. We denote $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$ as a row-stochastic adjacency matrix, and the graph Laplacian matrix as $\mathbf{\Delta} = \mathbf{I} - \tilde{\mathbf{A}}$, where $\mathbf{I}$ is the identity matrix.

2.1. Preliminary: Feature Propagation

Recently, Feature Propagation (FP) (Rossi et al., 2022) has been introduced to mitigate missing features in the graph domain. The core idea of FP is to diffuse the features that we know, i.e., provided, to the features that we do not know, i.e., missing, while maintaining the initial state of known features. Formally, given a node feature matrix consisting of k known features and u unknown features in each feature dimension f, the gradient flow of Dirichlet energy, i.e., $\ell(\mathbf{X}_{\cdot,f}, \mathcal{G}) = \frac{1}{2}\mathbf{X}_{\cdot,f}^{\top}\mathbf{\Delta}\mathbf{X}_{\cdot,f}$, at time step t can be expressed as the heat diffusion equation:

$$\dot{\mathbf{X}}(t) = -\nabla\ell(\mathbf{X}(t)) = -\mathbf{\Delta}\mathbf{X}(t),$$
$$(\text{IC})\mathbf{X}_{\cdot,f}(0) = \begin{bmatrix} \mathbf{X}_{k,f} \\ \mathbf{X}_{u,f}(0) \end{bmatrix}, (\text{BC})\mathbf{X}_{k,f}(t) = \mathbf{X}_{k,f}, \quad (1)$$
$$\forall k \in \mathcal{V}_{k,f}, \forall u \in \mathcal{V}_{u,f}, \forall f \leq F$$

where $\mathbf{X}_{\cdot,f} \in \mathbb{R}^N$ denotes feature vector at dimension f bounded by F, (IC) and (BC) denotes initial and boundary conditions, respectively. $\mathcal{V}_{k,f}$ and $\mathcal{V}_{u,f}$ denotes a set of *known* nodes and *unknown* nodes at feature dimension, f, respectively. Here, solving Equation 1 by linear equation, we obtain closed-from solution, $\mathbf{X}_u = -\mathbf{\Delta}_{uu}^{-1}\mathbf{\Delta}_{ku}^{\top}\mathbf{X}_k$. However, during calculation, it induces complexity of $\mathcal{O}(|\mathcal{V}_u|^3)$, which is not desirable in large graphs. Thus, we resort to an iterative Euler scheme and derive a formula as follows:

$$\begin{aligned}\mathbf{X}^{(i+1)} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{\Delta}_{uk} & \mathbf{I} - \mathbf{\Delta}_{uu} \end{bmatrix} \mathbf{X}^{(i)} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{A}}_{uk} & \tilde{\mathbf{A}}_{uu} \end{bmatrix} \mathbf{X}^{(i)}\end{aligned} \quad (2)$$

where $\mathbf{X}^{(i)}$ represents imputed feature matrix at i step. It is important to note that this formula is basically equivalent to multiplying the feature matrix with normalized adjacency while maintaining known feature values as its initial state. It shows the robust performance in the graph domain even when the data contains more than 90% missing features (Rossi et al., 2022). However, it cannot be naively applied to scRNA-seq data due to its following inherent nature: 1) The information regarding which features are missing or noisy is not provided. 2) As the graph structure is not provided, it is crucial to construct a graph that connects biologically relevant cells.

2.2. Proposed Methodology: scFP

Here, we propose the bi-level feature propagation method that extends FP in a manner that is well-suited to the scRNA-seq domain. Our overall architecture can be found in Figure 1. We first define a cell-cell graph using initial scRNA-seq data and perform a diffusion process using a Hard Feature Propagation scheme (Sec 2.2.1) that primarily pays attention to imputing zero values in the cell-gene count matrix while preserving non-zero values. After that, we refine the cell-cell graph by calculating k-nearest neighbors for each cell (Sec 2.2.2) using the previously smoothed outputs (i.e., warmed-up data). Then, we pass through Soft Feature Propagation (Sec 2.2.3) which allows the denoising of observed gene transcripts. Detail about each component can be found in the following sections.

2.2.1. Hard Feature Propagation

We start with imputing zero values, i.e., dropouts in gene expression with the aid of similar cells. To do so, we first need to define an adjacency matrix which first brings us a challenge compared to the graph domain, where adjacency information is provided. Here, the intuitive and cheap way that facilitates message-passing between similar cells is to introduce a kNN graph by calculating cosine similarities. However, considering the sparsity of cell-gene matrix (van Dijk et al., 2018; Yang et al., 2018), which is a direct resource for building kNN graph, we argue that obtained kNN graph should merely serve as an initialized graph for warming-up sparse and noisy gene transcripts. In this regard, as our primal goal is to impute zero-valued gene expressions (indexed by z) via non-zero-valued gene expressions (indexed by n), we apply Hard Feature Propagation as follows:

$$\mathbf{X}^{(i+1)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{A}}_{zn}^{\text{initial}} & \tilde{\mathbf{A}}_{zz}^{\text{initial}} \end{bmatrix} \mathbf{X}^{(i)} \quad (3)$$

where $\mathbf{X}^{(i)} \in \mathbb{R}^{N \times M}$ represents imputed cell-gene feature matrix at step i. After that, we obtain a converged warm-up matrix, $\mathbf{X}^{(K)}$, which is now denser and smoothed via neighbors. Note that in this step, we employ a hard clamping strategy where the non-zero values, which correspond to

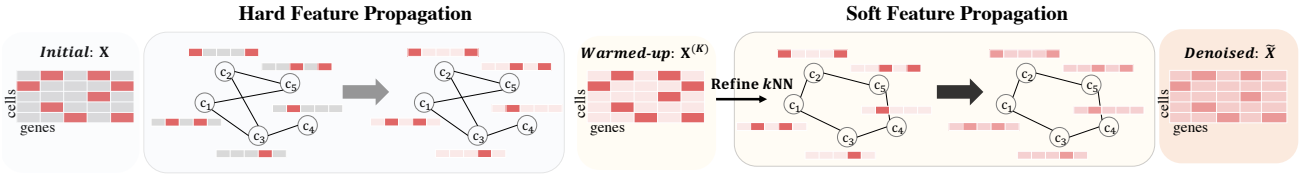


Figure 1. The overall architecture of scFP.

the observed expression values, are retained at their original values. This is because, in this step, more emphasis is placed on the imputation of zero-values, which are more prevalent in the cell-gene count matrix, compared to denoising non-zero transcript values. This is because non-zero values generally contain less noise than zero-values, i.e., dropouts.

2.2.2. Refining Cell-Cell Graph

With a warmed-up cell-gene matrix, $\mathbf{X}^{(K)}$ originating from Hard Feature Propagation, we make use of this matrix to refine an initial cell-cell graph which was built when cell representation was sparse. More precisely, a warmed-up cell-gene matrix is used as input for kNN graph as follows:

$$\tilde{\mathbf{A}}^{\text{refined}} = k\text{NN}(\mathbf{X}^{(K)}) \tag{4}$$

where $\tilde{\mathbf{A}}^{\text{refined}} \in \mathbb{R}^{N \times N}$ is refined normalized adjacency for a cell-cell graph which is built considering the denoised zero-values, which was not feasible to capture in the initial stage of kNN graph generation. In other words, this process could potentially reveal hidden or implicit graph structures that were not initially detectable due to sparse and noisy gene expression. We argue that this refined adjacency matrix may provide more accurate or robust graph representations, which could enhance subsequent analyses or learning tasks.

2.2.3. Soft Feature Propagation

Now, equipped with warmed-up cell-gene matrix, $\mathbf{X}^{(K)}$ and refined adjacency matrix for cell-cell graph, $\tilde{\mathbf{A}}^{\text{refined}}$, we hereby run Soft Feature Propagation. Specifically, compared to Hard Feature Propagation which is based on hard clamping (Zhu, 2005; Zhang & Lee, 2006; Raghavan et al., 2007), at this moment, we rather adopt soft clamping (Zhou et al., 2003; Wang & Zhang, 2006), which basically leaves room for updating the originally transcripted gene value of its neighbors endowed with implicit graph structures thanks to the warmed-up procedure. This aligns with our motivation to take into account noise from not only low transcript capture but also from the observed non-zero expression values that might possess biologically irrelevant signals, e.g., batch effects and transcriptional noise (Shaham et al., 2017; Wagner et al., 2016). Formally, Soft Feature Propagation is applied as follows:

$$\mathbf{X}^{(K+j+1)} = \alpha \tilde{\mathbf{A}}^{\text{refined}} \mathbf{X}^{(K+j)} + (1-\alpha)\mathbf{X}^{(K)} \tag{5}$$

where $\mathbf{X}^{(K+j)}$ is the updated cell-gene matrix at step j, $0 < \alpha < 1$ is the constant[1] that controls amount of information that $\mathbf{X}^{(K+j)}$ receives from its neighbors. After another K iteration, we finally obtain a converged and denoised cell-gene matrix $\tilde{\mathbf{X}} = \mathbf{X}^{(2K)}$. This denoised matrix considers both the noise from zero-values (dropouts) and the noise from non-zero-values (transcriptional noise), and it will serve as the main resource for subsequent downstream tasks. Detailed algorithm for scFP can be found in Appendix C.

3. Experiments

Experimental Settings. To evaluate the effectiveness of scFP, we evaluated scFP with 5 widely used real-world scRNA-seq data, Baron Mouse, Mouse ES, Mouse Bladder, Zeisel, and Baron Human. The detailed statistics of these datasets can be found in Appendix A.

Performance on Imputation. Table 1 shows the overall performance in imputation tasks with dropout rates ranging from the low case, e.g., 20%, to the severe case, e.g., 80%. We observe scFP shows robust performance regardless of dropout rates outperforming other baselines designed for denoising cell-gene matrix. Among baselines, it is worth noting that more complex models, e.g., AutoClass, and scGCL, exhibit lower performance than DCA. This tells us that the complexity of the model does not always guarantee imputation performance in the scRNA-seq domain. In this regard, the proposed method, scFP, does not require any learnable parameters and demonstrates its effectiveness by simply propagating features on the raw feature space. This highlights the importance of carefully handling observed values in order to obtain a well-imputed cell-gene matrix.

Performance on Cell Clustering. With obtained denoised cell-gene matrix, we further evaluate whether the denoised matrix performs well on the representative downstream task, cell clustering. As shown in Table 2, we observe scFP achieve promising results in terms of Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Clustering Accuracy (CA). Interestingly, while DCA performed well on imputation tasks compared to other baselines, its robustness could not be maintained in clustering tasks. It is also worth noting that scFP consistently outperforms the performance of MAGIC, demonstrating that

[1] As we aim to denoise known values, i.e., not to maintain its original state, we fixed α as 0.99 and used as constant.

Table 1. Overall performance in imputation task measured by RMSE.

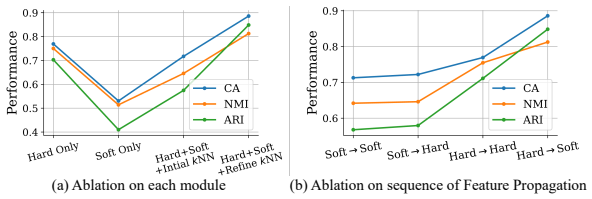
	Baron Mouse			Mouse ES			Mouse Bladder			Zeisel			Baron Human		
	Dropout Rates			Dropout Rates			Dropout Rates			Dropout Rates			Dropout Rates		
	20%	40%	80%	20%	40%	80%	20%	40%	80%	20%	40%	80%	20%	40%	80%
MAGIC	0.61	0.73	0.99	0.53	0.73	1.21	0.50	0.60	0.82	0.60	0.82	1.31	0.58	0.74	1.06
DCA	0.42	0.43	0.49	0.35	0.35	0.36	0.37	0.38	0.41	0.39	0.42	0.44	0.41	0.43	0.47
AutoClass	0.63	0.76	0.98	0.53	0.75	1.23	0.52	0.64	0.82	0.60	0.84	1.32	0.59	0.76	1.08
scGCL	0.64	0.74	0.97	0.59	0.75	1.16	0.51	0.62	0.81	0.66	0.82	1.29	0.63	0.77	1.08
scFP (Ours)	**0.36**	**0.37**	**0.43**	**0.32**	**0.32**	**0.36**	**0.26**	**0.26**	**0.31**	**0.39**	**0.40**	**0.44**	**0.33**	**0.34**	**0.39**

Table 2. Overall performance of cell clustering task measured by ARI, NMI, CA.

	Baron Mouse			Mouse ES			Mouse Bladder			Zeisel			Baron Human		
	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA
Raw	0.44	0.71	0.56	0.74	0.75	0.79	0.59	0.75	0.68	0.70	0.75	0.77	0.44	0.71	0.56
MAGIC	0.42	0.72	0.57	0.80	**0.85**	0.83	0.55	0.75	0.64	0.70	0.75	0.76	0.56	0.78	0.59
DCA	0.46	0.69	0.59	0.76	0.78	0.81	0.39	0.59	0.54	0.67	0.72	0.75	0.53	0.74	0.55
AutoClass	0.44	0.71	0.52	0.74	0.75	0.81	0.51	0.75	0.64	0.71	0.75	0.77	0.44	0.71	0.52
scGCL	0.43	0.72	0.54	0.73	0.75	0.79	0.53	0.75	0.64	0.65	0.70	0.73	0.50	0.78	0.62
kNN-smoothing	0.43	0.72	0.55	0.72	0.74	0.79	0.59	0.76	0.68	0.68	0.73	0.76	0.56	0.78	0.56
scFP (Ours)	**0.61**	**0.82**	**0.76**	**0.82**	0.83	**0.85**	**0.65**	**0.77**	**0.73**	**0.85**	**0.81**	**0.89**	**0.68**	**0.83**	**0.73**

the bi-level feature propagation and structure refinement are important to effectively denoise the scRNA-seq data. Furthermore, scFP demonstrates its robustness by consistently exhibiting superior performance compared to raw data across all five datasets.

Ablation studies. Figure 2 shows the ablation studies from two perspectives. First, by incremental ablation on each module, we observe scFP fully benefits when feature propagation has been made in both Hard and Soft ways. Here, it is important to note that utilizing a refined kNN graph, where a graph is reconstructed by a warmed-up matrix, is crucial before processing Soft Feature Propagation. Also, with ablation on the sequence of Feature Propagation, we verify the usage of Hard Feature Propagation, which maintains the observed value with its initial state at the early stage of imputation is significant. However, solely resorting to Hard Feature Propagation outputs sub-optimal results since it does not leave room for the observed values, which also possess noise, to be denoised via their neighbors.



(a) Ablation on each module (b) Ablation on sequence of Feature Propagation

Figure 2. Ablation studies of scFP. (a) "Hard+Soft+Refine kNN" corresponds to scFP. (b) "Hard→Soft" corresponds to scFP. Zeisel dataset is used.

Simulation studies. To demonstrate our claim that the diffusion of noise from false zeros can have a negative effect, we conducted experiments using a simulation dataset generated by Splatter Package (Zappia et al., 2017), where we can control the dropout rate, indicating the proportion of false zeros. In Figure 3, MAGIC, which diffuses both zero and non-zero values, successfully separates cell types by preserving biologically relevant signals when the dropout

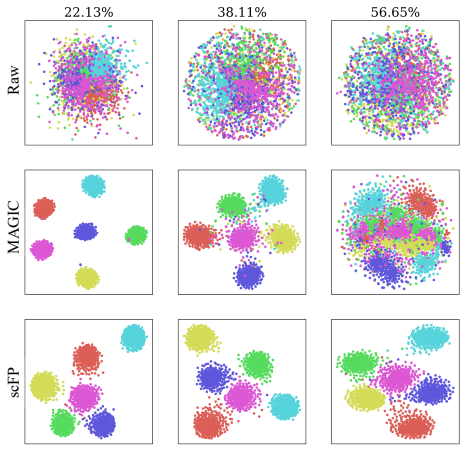


Figure 3. t-SNE visualization result on simulated dataset over various dropout rates. The rates at the top represent the dropout rate, which equals to the proportion of false zero values.

rate is relatively low (i.e., 22.13%). However, in cases with a high dropout rate (i.e., 56.65%), where a significant number of false zeros are present, it fails to separate cell types due to the contamination from false zeros. On the other hand, scFP effectively separates cell types even in situations with a high dropout rate, thanks to the careful diffusion process of the Hard FP step, which preserves non-zero values.

4. Conclusion

In this paper, we proposed scFP which imputes and denoises the observed scRNA-seq data that is inherently sparse and noisy. Specifically, we first aimed to impute zero-values of transcripts in Hard Feature Propagation with hard clamping of observed values. With a warmed-up matrix, we then refined the kNN graph and proceeded with Soft Feature Propagation in order to denoise known values with its neighbors, taking into account the potential of transcriptional noise. With a simple and lightweight design, its imputation and cell clustering performance under various datasets verifies the effectiveness of scFP.

References

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390, 2019.

Gong, W., Kwak, I.-Y., Pota, P., Koyano-Nakagawa, N., and Garry, D. J. Drimpute: imputing dropout events in single cell rna sequencing data. *BMC bioinformatics*, 19:1–10, 2018.

Hicks, S. C., Townes, F. W., Teng, M., and Irizarry, R. A. Missing data and technical variability in single-cell rna-sequencing experiments. *Biostatistics*, 19(4):562–578, 2018.

Lee, J., Kim, S., Hyun, D., Lee, N., Kim, Y., and Park, C. Deep single-cell RNA-seq data clustering with graph prototypical contrastive learning. *Bioinformatics*, 39(6):btad342, 05 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btad342.

Lee, N., Lee, J., and Park, C. Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7372–7380, 2022.

Li, H., Brouwer, C. R., and Luo, W. A universal deep neural network for in-depth cleaning of single-cell rna-seq data. *Nature Communications*, 13(1):1901, 2022.

Raghavan, U. N., Albert, R., and Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

Rossi, E., Kenlay, H., Gorinova, M. I., Chamberlain, B. P., Dong, X., and Bronstein, M. On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features, 2022.

Shaham, U., Stanton, K. P., Zhao, J., Li, H., Raddassi, K., Montgomery, R., and Kluger, Y. Removal of batch effects using distribution-matching residual networks. *Bioinformatics*, 33(16):2539–2546, 2017.

Tian, T., Wan, J., Song, Q., and Wei, Z. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4):191–198, 2019.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nature biotechnology*, 32(4):381, 2014.

van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., Bierie, B., Mazutis, L., Wolf, G., Krishnaswamy, S., and Pe'er, D. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.e27, 2018. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2018.05.061.

Wagner, A., Regev, A., and Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nature biotechnology*, 34(11):1145–1160, 2016.

Wang, F. and Zhang, C. Label propagation through linear neighborhoods. In *Proceedings of the 23rd international conference on Machine learning*, pp. 985–992, 2006.

Xiong, Z., Luo, J., Shi, W., Liu, Y., Xu, Z., and Wang, B. scgcl: an imputation method for scrna-seq data based on graph contrastive learning. *Bioinformatics*, 39(3): btad098, 2023.

Yang, M. Q., Weissman, S. M., Yang, W., Zhang, J., Canaann, A., and Guan, R. Misc: Missing imputation for single-cell rna sequencing data. *BMC Systems Biology*, 12(S7), 2018. doi: 10.1186/s12918-018-0638-y.

Zappia, L., Phipson, B., and Oshlack, A. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):174, 2017.

Zhang, X. and Lee, W. Hyperparameter learning for graph based semi-supervised learning algorithms. *Advances in neural information processing systems*, 19, 2006.

Zhang, Y., Tran, D., Nguyen, T., Dascalu, S. M., and Harris, F. C. A robust and accurate single-cell data trajectory inference method using ensemble pseudotime. *BMC bioinformatics*, 24(1):1–21, 2023.

Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

Zhu, X. *Semi-supervised learning with graphs*. Carnegie Mellon University, 2005.

A. Data statistics

Table 3. Statistics for real datasets used for experiments.

Data	# of Cells	# of Genes	# of Subgroups
Baron Mouse	1,886	14,861	13
Mouse ES cells	2,717	24,047	4
Mouse Bladder cells	2,746	19,771	16
Zeisel	3,005	19,972	7
Baron Human	8,569	20,125	14
Shekhar Mouse Retina cells	27,499	13,166	19

B. Extension to large dataset

Table 4. Performance of cell clustering in Shekhar Mouse Retina dataset.

	Shekhar Mouse Retina		
	ARI	NMI	CA
Raw	0.54	0.76	0.61
MAGIC	0.64	0.82	0.73
DCA	0.34	0.37	0.42
AutoClass	0.74	0.75	0.81
scGCL	0.54	0.74	0.58
kNN-smoothing	0.47	0.75	0.55
scFP (Ours)	**0.91**	**0.83**	**0.82**

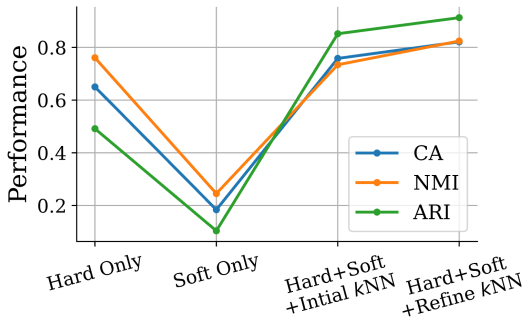


Figure 4. Ablation of each module in scFP in Shekhar Mouse Retina dataset. "Hard+Soft+Refine kNN" denotes scFP.

Here, we further extended our experiment on a relatively large dataset, Shekhar mouse retina cells, and compared the performance of cell clustering in Table 4 with its ablation study (Figure 4) on each module on scFP. Despite the absence of trainable parameters, our proposed method still demonstrates promising clustering performance on a large dataset. This suggests that, sometimes, rather than focusing on the complexity of the model in a trainable sense, giving careful consideration to the raw feature space and leveraging given resources, e.g., observed transcripts can be crucial.

C. Pseudocode of the proposed method

Algorithm 1 single-cell Feature Propagation (scFP)

Input: Cell-Gene Matrix $\mathbf{X}$, Initial kNN $\tilde{\mathbf{A}}^{\text{initial}}$
Output: Denoised Cell-Gene Matrix $\tilde{\mathbf{X}}$
$\mathbf{Y} \leftarrow \mathbf{X}$
while $\mathbf{X}$ has not converged **do**
  $\mathbf{X} \leftarrow \tilde{\mathbf{A}}^{\text{initial}}\mathbf{X}$
  $\mathbf{X}_{k,d} \leftarrow \mathbf{Y}_{k,d} \forall k \in \mathcal{V}_{k,d}, \forall d \leq M$ ▷ Hard Clamping
end while
$\tilde{\mathbf{A}}^{\text{refined}} = k\text{NN}(\mathbf{X}^{(K)})$ ▷ Refine kNN
while $\mathbf{X}^{(K)}$ has not converged **do**
  $\mathbf{X}^{(K)} \leftarrow \alpha\tilde{\mathbf{A}}^{\text{refined}}\mathbf{X}^{(K)} + (1-\alpha)\mathbf{X}^{(K)}$ ▷ Soft Clamping
end while

Observing that Equation 3 in Hard Feature Propagation essentially propagates features (i.e., gene expressions) via neighbors and resets the originally expressed gene values, we formulate the whole process in Algorithm 1. For the iteration until convergence, we used 40, which is enough to converge, as mentioned in FP (Rossi et al., 2022). It is important to note that we did not use any trainable parameters during the whole process and obtained a denoised matrix solely by raw feature space. Overall, in this work, we aim to emphasize simple and straightforward ways to enhance performance in imputation on subsequent downstream tasks, e.g., cell clustering.