Structural Inductive Bias in Supervised Learning for Single-cell Data

Elyas Heidari[*1]

[1]Department of Biological Systems Sciences and Engineering, ETH Zürich, Switzerland

[*]eheidari@student.ethz.ch

January 12, 2021

Abstract

Recent advancements in single-cell RNA sequencing (scRNAseq) have unraveled the transcriptional diversities underlying heterogeneous batches of cells. Current single-cell classification methods take genes as independent features to assign cell types disregarding their interactome which is of great importance in cellular processes. As a result, existing methods are prone to dataset-specific classification artifacts which prevents generalization of the model for cell-type classification in new datasets. Here, we introduce scPotter which takes feature interactions (e.g., gene-gene interactions) into account to classify single-cells and interpret the marker genes. Using two different peripheral blood mononuclear cell (PBMC) datasets as train and test sets, we demonstrate the potentials of scPotter for transfer learning on single-cell RNA sequencing (scRNAseq) data.

Contents

1 Introduction and motivation . . . 2

2 Materials and Methods . . . 3

2.1 PBMC dataset . . . 3

2.2 Models to incorporate gene-gene interactions . . . 3

2.3 Baseline model . . . 4

2.4 Interpreting Neural Network models . . . 4

3 Experiments and results . . . 4

3.1 Preprocessing . . . 4

3.2 Classification based on gene interaction networks . . . 5

4 Discussion . . . 8

4.1 Gene network reconstruction and transferability . . . 8

4.2 Software developed . . . 9

4.3 Potential applications and future steps . . . 9

1 Introduction and motivation

Single-cell transcriptomics has enabled the resolution of transcriptional diversities of individual cells, hence the possibility to classify cells' identity and function via their transcriptional state. Noting that such identity classifications can elucidate the organization and function of tissues in health and disease states, recently more investments have been devoted to this task, for example, in the Human Cell Atlas project (Rozenblatt-Rosen et al. 2017).

Current single-cell RNA sequencing (scRNA-seq) classification methods use gene expression as independent features to assign cell types, disregarding their interactome which is of great importance in the consistency of different cell states. Cell type classification based on a few marker genes, for example, yields better results in intra data set settings where both train and test datasets are sampled from the same dataset (Abdelaal et al. 2019) and may fail to correctly classify cell types under perturbations when the expression of the predefined marker genes varies concerning the control reference data set. Moreover, using gene expression as independent features can delude cell lineage relationships for cell states and cell types with a subtle difference in their gene expression profiles. In Fig. 1 we schematically illustrate how utilizing gene interactions may provide improvements in cell population deconvolution.

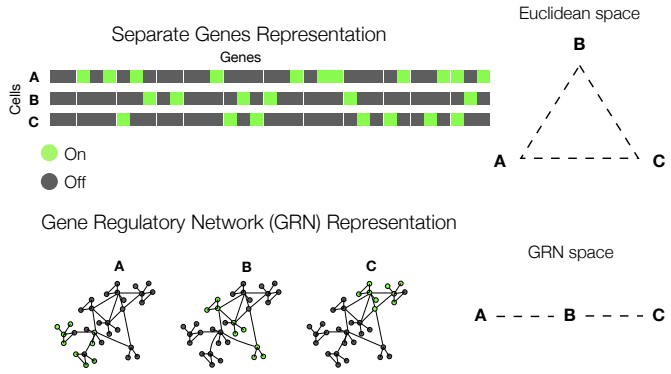

Figure 1: Incorporating feature (gene) Interactions
While using the contemporaneity Euclidean metric disregarding gene interactions on the single cell space could lead to spurious results, incorporating gene interactions can be used to decouple cell populations.

Structures are ubiquitous in biology. In reality, components of a biological system, interact with each other, forming functional modules (e.g., pathways), and in turn, modules interact to form a complex network creating the biological system (Alm and Arkin 2003). Gene regulatory networks, namely, perfectly embody such a phenomenon. While in many applications, as mentioned above, biological states (e.g., cell type), can be described by a few components (e.g., marker genes), incorporating the underlying structure can provide many useful inputs both for discriminative tasks (e.g., cell type classification) and interpretation tasks (e.g., finding marker genes/pathways). This is the goal of this project. That is to leverage structures on which input features function as an inductive bias for supervised learning on single-cell data. More precisely, here we show how one can exploit such underlying structures to perform cell-type classification based on single-cell RNA seq data.

In this project, we show how including such structures in the neural network classifiers facilitates better accuracy, robustness, and interpretability. We also provide an end-to-end pipeline to reconstruct and feed in such structures to the neural network classifiers. We decompose the problem of supervised learning into two parts. The first part is to train a model to perform annotation, which hereafter we call the "forward" problem. Once the model is trained, we interpret the model, that is, we aim to capture features that are the classes' characteristics In other words, we seek a few features which can discriminate classes. We refer to the latter as the "backward" problem.

2 Materials and Methods

2.1 PBMC dataset

A pivotal application of cell-type annotation is transferring knowledge gained from one dataset to new unseen datasets. Therefore, we sought fro two datasets with the same cell type profile but acquired in separate experiments to inspect how well the method can transfer between various environments and overcome unwanted variations (such as extrinsic noise and batch effects). We chose two human peripheral blood mononuclear cell (PBMC) scRNA-seq experiments (Ding et al. 2019). The dataset contains two separate experiments, hereafter called PBMC 1 and PBMC 2, generated in a single center, but employing seven different scRNA-seq methods. Both datasets are annotated by cell types which can be used to train and validate our classifiers. This suits our goal to verify our framework on a simulated task of real transfer learning. As in potential applications we expect our methods to learn annotation priorly on the complete (training) dataset and transfer the learned model to annotate the unseen (test) one.

2.2 Models to incorporate gene-gene interactions

It is known that in each cell proteins interact with each other through several regulatory mechanisms, forming an interaction network. In the single-cell field, however, we lack such an information at the protein level, yet, we can use rich RNA-seq datasets. One can use prior information, such as protein-protein interaction databases (Mering et al. 2003), and replace genes with their protein products. Also, one can reconstruct the gene-gene interaction network from scratch, by inferring it from the input dataset, e.g., using graphical models. The idea is to provide the model with a structure, on which features (here genes) interact with each other. There is a multitude of approaches to do so, but we do not take all of them into account here. Instead, we introduce a class of models, which apply to all graphical structures.

Recent advancements of deep learning models to capture modular structures in other applications such as image, audio, and speech analysis as well as the emergence of high-throughput scRNA-seq technologies that provide the large amounts of data required for such models motivated us to employ deep learning models applicable to non-regular graph data. Amongst such models, which are generally referred to as "geometric deep learning" (Bronstein et al. (2017), Monti et al. (2017)), Graph Convolutional Networks (GCNs) have significant power in extracting informative feature maps from signals defined over nodes of a graph while considering meaningful local structures and connectivities among them. This makes GCNs suitable for our purpose of analyzing non-regular gene networks (Zhou et al. 2018).

2.3 Baseline model

We investigate to which extent inferring the graph structure from the data and using it as an inductive bias to a GCN improves robustness, generalization, and interpretation in comparison to the standard fully connected neural networks (FCNNs) without any prior structural information. We use Multi-Layer Perceptrons (MLPs) (Gardner and Dorling 1998) as the FCNN model.

2.4 Interpreting Neural Network models

While Neural Networks are widely known as a black box, for they are difficult to interpret, several methods have been recently put forward to interpret and explain Neural Networks (Montavon, Samek, and Müller 2018). In particular, captum python library (Kokhlikyan et al. 2020) has been recently introduced in python to facilitate the interperetibility of neural networks. We use captum to find the characteristic features or marker genes of cell types. All Neural Networks are implemented in python, using pytroch (Paszke et al. 2019) and pytorch-geometric (Fey and Lenssen 2019). For preprocessing, visualization, and exploratory data analysis we used R (Ihaka and Gentleman 1996).

3 Experiments and results

3.1 Preprocessing

We take PBMC 1 as our training dataset and PBMC 2 as our test dataset. The task is to learn the model on PBMC 1, and validate it on PBMC 2, by predicting cell type annotations, and comparing them to true labels. Preprocessing includes three steps, first, cell type selection and cell-wise data preprocessing, second, gene subset selection, and lastly, structure reconstruction:

- **Cell type selection & preprocessing:** We select a subset of cell types in the intersection of cell types present in both training and test datasets. Afterward, data for each cell should be preprocessed. Cells with less than 100 counts and cells with a high percentage of mitochondrial gene counts (>0.01) are filtered out due to low sequencing/experimental quality. Finally, we used the scran package (Lun et al. 2017) in R to normalize counts. We ended up with 400 cells per class for the training dataset. We do not subsample the test dataset. Cell counts for training and test datasets are shown in Fig. 2.
- **Gene subset selection:** Following the widely-used procedure in the field, we select highly variable genes, using the scran package. We ended up with 260 genes in total.
- **Gene network reconstruction:** We use Gaussian graphical models implemented by R package glasso (Friedman, Hastie, and Tibshirani 2008) to reconstruct a sparse gene interaction network for the training dataset. The reconstructed networks utilizing glasso are shown in Fig. 3-A&B. In Fig. 3-A nodes are colored based on their graph communities detected by the Louvain algorithm (Blondel et al. 2008), same colors are used for the second network in Fig. 3-B, reconstructed on the test dataset. For classifiers, we just use the graph reconstructed on the training dataset as we want to leave the test dataset, unseen.

Cell type	PBMC 1	PBMC 2
B cell (B)	400	2450
CD14+ monocyte (CD14+)	400	1582
CD16+ monocyte (CD16+)	400	230
CD4+ T cell (CD4+)	400	2832
Cytotoxic T cell (Cyto.)	400	2114
Megakaryocyte (MK)	400	61
Natural killer cell (NK)	400	691

Figure 2: Cell type count profiles of training and test datasets

A 2-dimensional UMAP embedding of the training and test datasets after preprocessing is shown in Fig. **??**-A. As one can see, the classes are not separated and such a heterogeneity affects the classification accuracy.

3.2 Classification based on gene interaction networks

We use a recently-introduced graph neural network, called TransformerConv Net (Shi et al. 2020). In TransformerConv Net, each node generates a message based on its features and sends it to its neighbors. Then each node aggregates the messages of its neighbors and uses this aggregate to updates its features. The aggregation is done with a permutation-invariant function. In TransformerConv Net, the updated feature vector of node i after one round of message passing is $x_i' = W_1 x_i + W_2 \text{mean}_{j \in \mathcal{N}(i)}(x_j)$ where W_1 and W_2 are learnt weight matrices, shared for all of the nodes, and $\mathcal{N}(i)$ is the set of neighbors of i. In our setting, each node starts with only one feature (which is the expression level of one gene), but we can have a message-passing function that creates hidden feature vectors of higher dimensions.

As mentioned before, after training the graph neural network, we use captum to find the most important features, based on a well-known interpretation model, the Integrated Gradients method (Sundararajan, Taly, and Yan 2017). We also compared the results to an MLP which does not impose any structure on the input features. Both MLP and GCN are optimized naively, by selecting a proper learning rate, dropout, and the number of hidden layers and hidden nodes. The results are shown in Fig. 5. The GCN outperforms the MLP on both training and test datasets. By comparison to the Human Protein Atlas (Pontén, Jirström, and Uhlen 2008), our results suggest how using a graphical structure as an inductive bias leads to improvement in interpretation. While the MLP can not fully capture marker genes of the PBMC cell types, GCN perfectly recovers the characteristic genes, the results are coherent with previous results found by relevant toolkits SCANPY (Wolf, Angerer, and Theis (2018), notebook) and Seurat (Stuart et al. (2019), notebook) which essentially use statistical methods to find marker genes. Fig. (fig:umaps)-B&C indicate how precise the GCN could

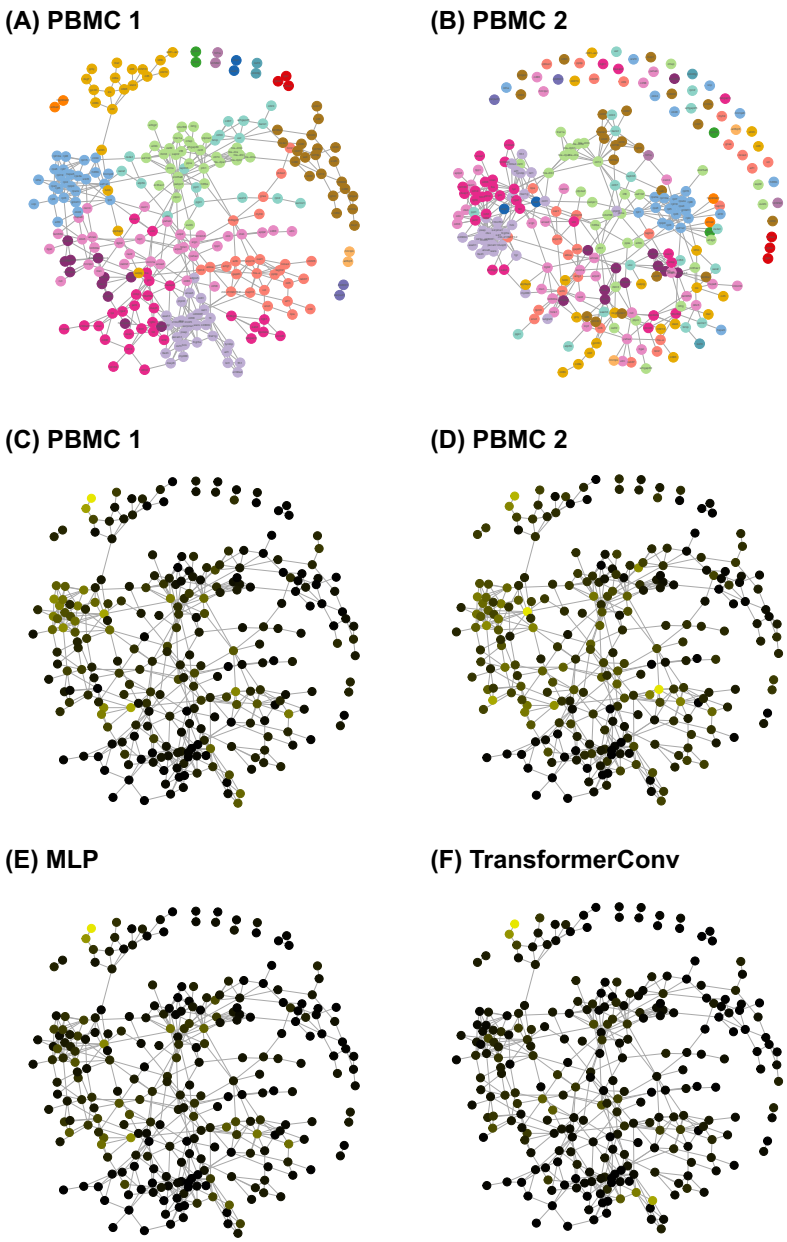

Figure 3: The reconstructed gene-interaction networks
Each node represents a gene and each edge represents an interaction, estimated by the glasso algorithm. (A) The interaction network reconstructed on the training dataset, PBMC 1. (B) The interaction network reconstructed on the training dataset, PBMC 2. (C-D) Each node is colored based on the average expression of the gene in Natural Killer Cells, in (C) training and (D) test datasets (the brighter the higher). (E-F) Each node is colored based on the average importance (based on captum) of the gene in Natural Killer Cells, from (E) MLP and (F) TransformerConv classifiers (the brighter the higher).

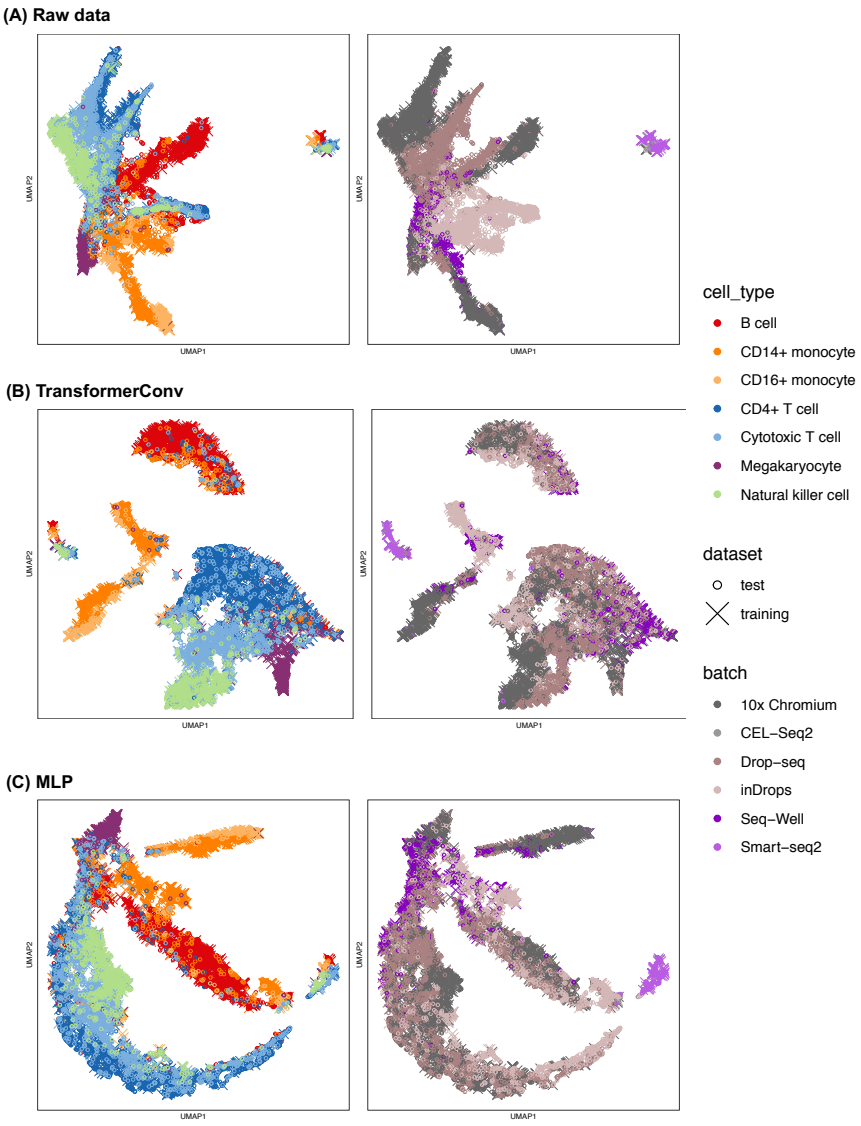

Figure 4: UMAP embedding
Left: colored by cell types, Right: colored by scRNAseq technology. (A) Raw data with all 260 genes. (B-C) Just using top 2 gene markers per class (14 total) captured by (B) TransformerConv and (C) MLP classifiers.

deconvolve the cell types (at least at the broad level), and MLP fails to do so. While the MLP simply recovers the genes with the higher mean in each class, GCN recovers the true genes or gene modules that are known to contribute in cell identities (Fig. 3-E&F).

(A) Prediction Accuracy

Method	train. Acc. ± s.d.	test Acc. ± s.d.
TransformerConv	**0.943 ± 0.001**	**0.846 ± 0.011**
MLP	0.902 ± 0.015	0.813 ± 0.018

(B) TransformerConv

B	CD14+	CD16+	CD4+	Cyto.	MK	NK
cd74	s100a9	lst1	actb	ccl5	ppbp	gnly
hla-dra	lyz	psap	rpl4	nkg7	ftl	tyrobp
ms4a1	s100a8	fcgr3a	pabpc1	mtrnr2l1	fth1	nkg7
ighm	ftl	cst3	rps3	gzmh	actb	fcer1g
hla-drb1	vcan	aif1	rps6	eef1a1	tagln2	itgb2

(C) MLP

B	CD14+	CD16+	CD4+	Cyto.	MK	NK
cd74	lyz	ftl	eef1a1	eef1a1	actb	gnly
hla-dra	hla-dra	sat1	tpt1	ccl5	ppbp	nkg7
igkc	ftl	actb	rps6	hla-c	hla-e	hla-b
actb	s100a9	ptma	rpl15	rps3	ccl5	eef1a1
eef1a1	fos	psap	rps4x	nkg7	fth1	hla-e

Figure 5: Classification and interpretation results
(A) Prediction accuracies on training and test datasets. (B) Marker genes captured by the GCN model. (C) Marker genes captured by the MLP model.

4 Discussion

We developed new tools for leveraging gene interactions in single-cell data analysis. We showed how such a piece of structural information could be employed as an inductive bias for better generalization and interpretation. This project was simply a proof of concept but it sheds light on the importance and utility of interaction networks in the field of single-cell and even more, in other biological applications. The advantage of this approach over similar approaches in contemporary single-cell field is that it can perform both forward and backward tasks of supervised learning in one pipeline, also it is a generic approach that can be well generalized to any biological application, where the task is annotation.

4.1 Gene network reconstruction and transferability

We reconstructed gene interaction networks, without any prior knowledge, using the graphical lasso. The graphical lasso leads to sparse yet modular solutions on scRNA-seq data, as shown in Fig. 3. This is in line with the fact that the true underlying gene regulatory networks are essentially sparse. Moreover, sparsity reduces the estimation noise and ensures the accuracy of the graph representation. Therefore, imposing such an inductive bias in the GCN framework through the graph structure improves the performance through regularization of the parameter space. One can also compare Fig. 3-A and Fig. 3-B, where the gene-interaction networks

are reconstructed on two separate PBMC datasets. Higher compatibility will lead to better transferability of the model along environments. We observe that many gene communities from the PBMC 1 are conversed on the PBMC 2 dataset; suggesting that graphical lasso is a reliable method for reconstructing gene networks with scRNA-seq data, in that, it preserves the global network structures.

4.2 Software developed

We developed scPotter GitHub.com/EliHei2/scPotter as an end-to-end pipeline for 1) annotation 2) finding the most important features for each annotation category. The backbone is adopted from two of the packages, previously developed by us, GNNProject GitHub.com/e-sollier/DL2020 in python and scGCNUtiles GitHub.com/EliHei2/scGCN in R. Many models from synthetic data generation to multiple neural network classifiers are implemented along with tools for visualization and exploratory data analysis. The modular structure of the pipeline enables extensions to new models and analyses. The pipeline is designed to meet state of the art reproducibility standards and criteria of open science, such as coherence, integrity, documentation, and readability.

4.3 Potential applications and future steps

Although here we just focused on gene-gene interaction networks and reconstructed them, the pipeline can be also used on other data modalities (e.g., chromatin structure), where the interaction network is already given. For instance, Hi-C data (Nagano et al. 2013) can be used to reconstruct genomic interaction networks, and the rest of the pipeline, that is, classification and interpretation can be used as-is, to find marker genomic locations.

Genomic locations and gene proximity on chromosomes can be used as additional information for cell annotation. The approach is inspired by the fact that genes in proximity on the chromosomes are more likely to undergo the same epigenetic events, such as chromatin openness and genomic interactions (e.g., in topologically associating domains, TADs). One can sort genes based on genomic locations. By doing so, each cell can be represented as either a 1-dimensional signal in which each (time) point represents the expression/activation level of the corresponding gene or a linear graph in which each node is assigned by the expression level of the corresponding gene. By such modeling, one can use both 1-dimensional Convolutional Neural Networks and Graph Neural Networks. Both of these models are implemented in our pipeline, to perform supervised learning.

Acknowledgments

Hereby, I want to appreciate the extremely helpful comments of Will Macnair, Izaskun Mallona, and Stephany Orjuela. I thank my collaborators in the related previous projects, Laleh Haghverdi and Etienne Sollier. Finally, I express my regards to the STA426 instructors, Mark Robinson, Hubert Rehrauer, and Ahmadreza Yousefkhani for motivating me to conduct this piece of research.

Data availability

The raw dataset is available on the Gene Expression Omnibus with accession number GSE132044. The preprocessed dataset is available here.

Supplementary information

The intermediate results and the supplementary information are available on the project's GitHub repository.

References

Abdelaal, Tamim, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel JT Reinders, and Ahmed Mahfouz. 2019. "A Comparison of Automatic Cell Identification Methods for Single-Cell Rna Sequencing Data." *Genome Biology* 20 (1). Springer: 194.

Alm, Eric, and Adam P Arkin. 2003. "Biological Networks." *Current Opinion in Structural Biology* 13 (2). Elsevier: 193–202.

Blondel, Vincent D, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. "Fast Unfolding of Communities in Large Networks." *Journal of Statistical Mechanics: Theory and Experiment* 2008 (10). IOP Publishing: P10008.

Bronstein, Michael M, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. "Geometric Deep Learning: Going Beyond Euclidean Data." *IEEE Signal Processing Magazine* 34 (4). IEEE: 18–42.

Ding, Jiarui, Xian Adiconis, Sean K Simmons, Monika S Kowalczyk, Cynthia C Hession, Nemanja D Marjanovic, Travis K Hughes, et al. 2019. "Systematic Comparative Analysis of Single Cell Rna-Sequencing Methods." *BioRxiv*. Cold Spring Harbor Laboratory, 632216.

Fey, Matthias, and Jan Eric Lenssen. 2019. "Fast Graph Representation Learning with Pytorch Geometric." *arXiv Preprint arXiv:1903.02428*.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2008. "Sparse Inverse Covariance Estimation with the Graphical Lasso." *Biostatistics* 9 (3). Oxford University Press: 432–41.

Gardner, Matt W, and SR Dorling. 1998. "Artificial Neural Networks (the Multilayer Perceptron)—a Review of Applications in the Atmospheric Sciences." *Atmospheric Environment* 32 (14-15). Elsevier: 2627–36.

Ihaka, Ross, and Robert Gentleman. 1996. "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics* 5 (3). Taylor & Francis Group: 299–314.

Kokhlikyan, Narine, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, et al. 2020. "Captum: A Unified and Generic Model Interpretability Library for Pytorch." *arXiv Preprint arXiv:2009.07896*.

Lun, Aaron, Karsten Bach, Jong Kyoung Kim, Antonio Scialdone, and Laleh Haghverdi. 2017. "Package 'Scran'."

Mering, Christian von, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. 2003. "STRING: A Database of Predicted Functional Associations Between Proteins." *Nucleic Acids Research* 31 (1). Oxford University Press: 258–61.

Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller. 2018. "Methods for Interpreting and Understanding Deep Neural Networks." *Digital Signal Processing* 73. Elsevier: 1–15.

Monti, Federico, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. "Geometric Deep Learning on Graphs and Manifolds Using Mixture Model Cnns." In *Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition*, 5115–24.

Nagano, Takashi, Yaniv Lubling, Tim J Stevens, Stefan Schoenfelder, Eitan Yaffe, Wendy Dean, Ernest D Laue, Amos Tanay, and Peter Fraser. 2013. "Single-Cell Hi-c Reveals Cell-to-Cell Variability in Chromosome Structure." *Nature* 502 (7469). Nature Publishing Group: 59–64.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "Pytorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems*, 8026–37.

Pontén, Fredrik, Karin Jirström, and Matthias Uhlen. 2008. "The Human Protein Atlas—a Tool for Pathology." *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland* 216 (4). Wiley Online Library: 387–93.

Rozenblatt-Rosen, Orit, Michael JT Stubbington, Aviv Regev, and Sarah A Teichmann. 2017. "The Human Cell Atlas: From Vision to Reality." *Nature News* 550 (7677): 451.

Shi, Yunsheng, Zhengjie Huang, Shikun Feng, and Yu Sun. 2020. "Masked Label Prediction: Unified Massage Passing Model for Semi-Supervised Classification." *arXiv Preprint arXiv:2009.03509*.

Stuart, Tim, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. 2019. "Comprehensive Integration of Single-Cell Data." *Cell* 177 (7). Elsevier: 1888–1902.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. 2017. "Axiomatic Attribution for Deep Networks." *arXiv Preprint arXiv:1703.01365*.

Wolf, F Alexander, Philipp Angerer, and Fabian J Theis. 2018. "SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis." *Genome Biology* 19 (1). BioMed Central: 15.

Zhou, Jie, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. "Graph Neural Networks: A Review of Methods and Applications." *arXiv Preprint arXiv:1812.08434*.