# Introduction

## Instructions

This is an introduction to your report, you should edit this text/markdown section to compose. In this text/markdown, you should introduce:

* Background of the problem
  + what type of problem: disease/readmission/mortality prediction, feature engineering, data processing, etc
  + what is the importance/meaning of solving the problem
  + what is the difficulty of the problem
  + the state of the art methods and effectiveness.
* Paper explanation
  + what did the paper propose
  + what is the innovations of the method
  + how well the proposed method work (in its own metrics)
  + what is the contribution to the research regime (referring to the Background above, how important the paper is to the problem).

## Our report

**scGCL: an imputation method for scRNA-seq data based on Graph Contrastive Learning**

**Background**

The sparse nature of single-cell RNA sequencing (scRNA-seq) data arises from the biological and technical challenges encountered when measuring gene expression at the individual cell level. This sparsity indicates that numerous genes go undetected in the examined cells within the dataset. Several factors contribute to this phenomenon, including the complexities of capturing and analyzing the minute amounts of RNA present in single cells, such as:

1. Low RNA content- Low content makes it challenging to detect all mRNA molecules present; some cells naturally contain more mRNA molecules than others, which have more chance to detect the genes.
2. Low gene expression or gene expression variance- Natural differences in gene expression levels between cells, even within the same cell type, can contribute to the observed sparsity. Some genes are only expressed in specific cell states or conditions, leading to a large number of zeros in the data matrix where a particular gene is not active in most cells.
3. Technical variance- The process of isolating single cells, reverse transcribing RNA into cDNA, and amplifying the cDNA before sequencing introduces technical variability. Some mRNA molecules may be lost or degraded during these steps, resulting in incomplete detection of the transcriptome.
4. Dropout events- The most important factor contributing to sparsity, dropout events occur when mRNA molecules present in the cell are not detected, leading to a zero count for genes that are actually expressed but missed during sequencing. This is often due to inefficiencies in reverse transcription, amplification, or sequencing steps. Addressing dropout events through imputation in scRNA-seq data can significantly streamline the analysis process by filling in gaps where gene expression values are missing or undetected.

The importance of addressing this sparsity problem lies in several key areas:

Enhanced Data Quality: Sparsity in scRNA-seq data means that many gene expression readings are missing or near zero, leading to a dataset filled with a lot of noise and few signals. By effectively imputing these missing values, the quality of the data can be significantly enhanced, leading to more reliable and interpretable results.

Improved Biological Insights: The main goal of scRNA-seq analysis is to uncover the complex mechanisms of cellular processes and heterogeneity among cells in different states or environments. Addressing the sparsity issue allows for a more accurate identification of gene expression patterns, cell types, and developmental states, thereby facilitating deeper biological insights and discoveries.

Enabling Advanced Analysis: Many downstream analyses, such as clustering, trajectory inference, and differential expression analysis, require robust datasets without extensive missing values. By solving the sparsity problem, these analyses can be performed more effectively, leading to more nuanced understanding of cellular behavior and interactions.

Increased Comparability and Integration: As the field moves towards large-scale studies involving multiple datasets, the ability to accurately impute missing data becomes crucial for integrating and comparing datasets from different sources. This comparability is essential for drawing broader conclusions and for the reproducibility of findings across studies.

The difficulty of addressing the sparsity in scRNA-seq data include:

1. Lack of Ground Truth: In many cases, there is no "ground truth" for what the imputed values should be. This makes it difficult to train models and evaluate their performance objectively. The best that can be done is to rely on biological validation or downstream analysis outcomes, which can be time-consuming and not always definitive.
2. Data complexity: The high dimensionality of the data, coupled with the sparsity of gene expression (many genes are not expressed in many cells), makes it difficult to accurately impute missing values without introducing bias or losing critical information.
3. Noise: scRNA-seq data is also plagued by technical noise and dropout events, where genes are expressed but not detected due to limitations in sequencing depth or efficiency. Differentiating between true biological zeros and technical dropouts is a significant challenge.

The ideal approach we desired:

1. The approach not only diminishes noise in the data but also enhances the precision of clustering and classification efforts.
2. It bolsters the accuracy of differential expression analyses, and aids in the integration of data from various sources, thereby simplifying the overall complexity of handling scRNA-seq datasets.

**Paper proposed**

The paper proposes a novel method, scGCL (single-cell Graph Contrastive Learning), specifically designed for imputing missing values in single-cell RNA sequencing (scRNA-seq) data. This method innovatively combines graph contrastive learning with the Zero-inflated Negative Binomial (ZINB) distribution model to estimate dropout events accurately. By employing contrastive learning within a graph theory framework, scGCL is adept at capturing the complex relationships between cells, thereby enhancing the prediction and reconstruction of missing gene expression values.

The innovation of the scGCL method lies in its unique approach to leveraging the strengths of graph contrastive learning, tailored specifically to the graph domain of scRNA-seq data. This allows for a sophisticated encapsulation of both global and local semantic information within the data. Furthermore, scGCL introduces a strategic selection of positive samples that significantly improve the representation of target nodes. This is complemented by the utilization of an autoencoder framework based on the ZINB distribution, specifically designed to model the global probability distribution of gene expression data effectively. This nuanced approach provides a robust solution to dropout imputation challenges, setting it apart from existing methods.

The contribution of scGCL to the research regime is substantial, addressing critical challenges outlined in the background of sparse data, technical noise, and the need for advanced computational strategies in scRNA-seq data analysis. By providing a more accurate and efficient method for imputing missing values, scGCL enhances the quality of scRNA-seq data analysis, leading to deeper biological insights and facilitating advancements in personalized medicine and genomics. This approach not only solves a significant problem in the field but also pushes the boundaries of what is possible with self-supervised learning in genomics, highlighting the importance of this paper to the ongoing research and development within the domain.

**scGCL: an imputation method for scRNA-seq data based on Graph Contrastive Learning**

The sparse nature of single-cell RNA sequencing (scRNA-seq) data arises from the biological and technical challenges encountered when measuring gene expression at the individual cell level. This sparsity indicates that numerous genes go undetected in the examined cells within the dataset. Several factors contribute to this phenomenon, including the complexities of capturing and analyzing the minute amounts of RNA present in single cells, such as:

1. Low RNA content- Low content makes it challenging to detect all mRNA molecules present; some cells naturally contain more mRNA molecules than others, which have more chance to detect the genes.
2. Low gene expression or gene expression variance- Natural differences in gene expression levels between cells, even within the same cell type, can contribute to the observed sparsity. Some genes are only expressed in specific cell states or conditions, leading to a large number of zeros in the data matrix where a particular gene is not active in most cells.
3. Technical variance- The process of isolating single cells, reverse transcribing RNA into cDNA, and amplifying the cDNA before sequencing introduces technical variability. Some mRNA molecules may be lost or degraded during these steps, resulting in incomplete detection of the transcriptome.
4. Dropout events- The most important factor contributing to sparsity, dropout events occur when mRNA molecules present in the cell are not detected, leading to a zero count for genes that are actually expressed but missed during sequencing. This is often due to inefficiencies in reverse transcription, amplification, or sequencing steps. Addressing dropout events through imputation in scRNA-seq data can significantly streamline the analysis process by filling in gaps where gene expression values are missing or undetected.

This approach not only diminishes noise in the data but also enhances the precision of clustering and classification efforts, bolsters the accuracy of differential expression analyses, and aids in the integration of data from various sources, thereby simplifying the overall complexity of handling scRNA-seq datasets.

Specific approach: Given the scarcity of cell type labels, contrastive learning has emerged as a powerful tool in self-supervised learning, achieving enhanced feature representation by amplifying the similarity between positive samples and diminishing that between negative ones. When applied within the context of graph theory, this approach evolves into graph contrastive learning, which is theoretically capable of discerning the intricate relationships between cells and subsequently reconstructing missing gene expression values. This integration harnesses the strength of contrastive learning, tailored to the graph domain, to effectively capture the cellular interactions and dynamics inherent in scRNA-seq data.

This paper introduces a method called scGCL, a single-cell Graph Contrastive Learning approach for imputing missing values in scRNA-seq data, which combines graph contrastive learning with the Zero-inflated Negative Binomial (ZINB) distribution for estimating dropout events. scGCL leverages contrastive learning to encapsulate both global and local semantic information, and it employs a strategy for selecting positive samples that enhance the representations of target nodes. To effectively model the global probability distribution of gene expression data, scGCL utilizes an autoencoder framework based on the ZINB distribution. This framework is designed to reconstruct scRNA-seq data by leveraging a prior distribution, offering a nuanced approach to addressing dropout imputation challenges.

# Scope of Reproducibility

## Instructions

List hypotheses from the paper you will test and the corresponding experiments you will run.

* Hypothesis 1: xxxxxxx
* Hypothesis 2: xxxxxxx

You can insert images in this notebook text, [see this link](https://stackoverflow.com/questions/50670920/how-to-insert-an-inline-image-in-google-colaboratory-from-google-drive) and example below:

## Our report

Hypotheses to be tested:

Hypothesis 1: We hypothesize that scGCL will exhibit better clustering performance and imputation performance than other existing methods.

~~Hypothesis 2: We hypothesize that scGCL will best perform with a combination of ZINB encoder~~

# Methodology

### Instructions

This methodology is the core of your project. It consists of run-able codes with necessary annotations to show the experiment you executed for testing the hypotheses.

The methodology at least contains two subsections data and model in your experiment.

## Data

### Instructions

Data includes raw data (MIMIC III tables), descriptive statistics (our homework questions), and data processing (feature engineering).

* Source of the data: where the data is collected from; if data is synthetic or self-generated, explain how. If possible, please provide a link to the raw datasets.
* Statistics: include basic descriptive statistics of the dataset like size, cross validation split, label distribution, etc.
* Data process: how do you munipulate the data, e.g., change the class labels, split the dataset to train/valid/test, refining the dataset.
* Illustration: printing results, plotting figures for illustration.

You can upload your raw dataset to Google Drive and mount this Colab to the same directory. If your raw dataset is too large, you can upload the processed dataset and have a code to load the processed dataset.

### Our report

* Source of the data: The paper actually runs the experiment on 14 different datasets. However, we will run our project on a single dataset which the paper has called "Adam" because it comes from this paper ( Adam et al. (2017)). The dataset is a set of RNA sequences that is taken from kidney cells of mice. This dataset contains data for 3,660 cells, 23,797 genes and 8 cell types. The dataset is available to download from the github page of the original paper.
* Data Process: The expression matrix of the scRNA-seq data is taken as the input raw data. To reduce noise in the scRNA-seq data, the paper pre-process the raw gene expression profiles using the following pre-processing methods:

1. Data filtering and quality control are the first steps in scRNA-seq data pre-processing. Therefore, we only keep genes with non-zero expression in more than 1% of cells and cells with non-zero expression in more than 1% of genes.
2. Since the data in the count matrix are discrete and affected by the size factor, we normalize it by the size factor then transform discrete values through the log function. Finally, we select the top *‘t’* highly variable genes based on the normalized discrete values computed by the scanpy package.

Generally, we select t=2048. These selected genes are the most variable genes for training. When using multiple datasets, the paper uses a consistent pre-processing method before running all baseline methods.

Those genes with the highest variation are then randomly shuffled and split into train/validation/test sets with the following split: 10/10/80 (respectively). This process of shuffling and splitting the dataset is done 20 times and each iteration of the generated test/validation/train split is then concatenated with the overall set from the previous iteration.

* Statistics: Run cell below to see statistics
* Illustration: See below for plots of our data

## Model

### Instructions

The model includes the model definition which usually is a class, model training, and other necessary parts.

Model architecture: layer number/size/type, activation function, etc

Training objectives: loss function, optimizer, weight of each loss term, etc

Others: whether the model is pretrained, Monte Carlo simulation for uncertainty analysis, etc

The code of model should have classes of the model, functions of model training, model validation, etc.

If your model training is done outside of this notebook, please upload the trained model here and develop a function to load and test it.

## 

### Our report

Model Comparison:

From paper:

To validate the performance of scGCL on scRNA-seq data imputation, we evaluate scGCL and other state-of-the-art baseline methods on multiple downstream analysis tasks such as clustering performance, recovering gene expression levels and pseudo-time analysis. scGCL demonstrates its effectiveness on scRNA-seq data imputation through extensive experiments. Furthermore, we verify the influence of different hyperparameters on the clustering results of scGCL, and the importance of each part on scGCL through ablation experiments.

Clustering analysis of scRNA-seq data is an essential analysis task, which affects the discrimination of cell types and subtypes. To evaluate the Clustering analysis of scGCL, we compare it with four state-of-the-art baseline methods. Among these four baseline methods, GraphSCI and scTAG are based on graph convolution, AutoClass is based on pre-clustering, and MAGIC is the traditional scRNA-seq imputation method. Two traditional Clustering analysis criteria, adjusted rand index (ARI) and normalized mutual information (NMI) are used to evaluate the Clustering performance of scGCL.

The Clustering results of scGCL slightly outperforms on the average all baseline methods on most datasets. The average ARI and NMI of scGCL across all 14 datasets are 0.82 and 0.80 with the second best values of 0.80 and 0.75. At the same time, we find that state-of-the-art graph convolutional embedding methods have certain drawbacks through UMAP visualization. scTAG measures the similarity between cells and cluster centers, making cells with ambiguous clusters move closer to the wrong cell clusters. GraphSCI applies association information between genes, which destroys the original properties of scRNA-seq data. Therefore, although GraphSCI outperforms the raw data in terms of clustering effect, its UMAP visualization is affected by the gene graph, and it is difficult to show the original biological characteristics. scGCL not only effectively distinguishes different types of cells but preserves the biological properties of scRNA-seq data (Fig. 3c). The number of cells detected by scGCL on different cell clusters is the closest to the number of cells of the real cell type, which indicates that scGCL maintains the characteristics of the cell type. In summary, scGCL can effectively eliminate the dropout events of scRNA-seq data and facilitate downstream analysis.

# Results

## Instructions

In this section, you should finish training your model training or loading your trained model. That is a great experiment! You should share the results with others with necessary metrics and figures.

Please test and report results for all experiments that you run with:

* specific numbers (accuracy, AUC, RMSE, etc)
* figures (loss shrinkage, outputs from GAN, annotation or label of sample pictures, etc)

Model Comparison:

* compare you model with others
* you don't need to re-run all other experiments, instead, you can directly refer the metrics/numbers in the paper

Clustering Analysis

The two metrics used to evaluate the performance of scGCL are adjusted rand index (ARI) and normalized mutual information (NMI.) These numbers are compared against four “state-of-the-art” baseline imputation methods: GraphSCI, scTAG, AutoClass, and MAGIC. The clustering results of scGCL on the Adam dataset outperforms all four of these methods, at 0.91 ARI and 0.89 NMI, respectively. From the paper itself including metrics from the other 13 datasets, the average ARI and NMI of scGCL across all datasets are 0.82 and 0.80 with the second best values of 0.80 and 0.75.

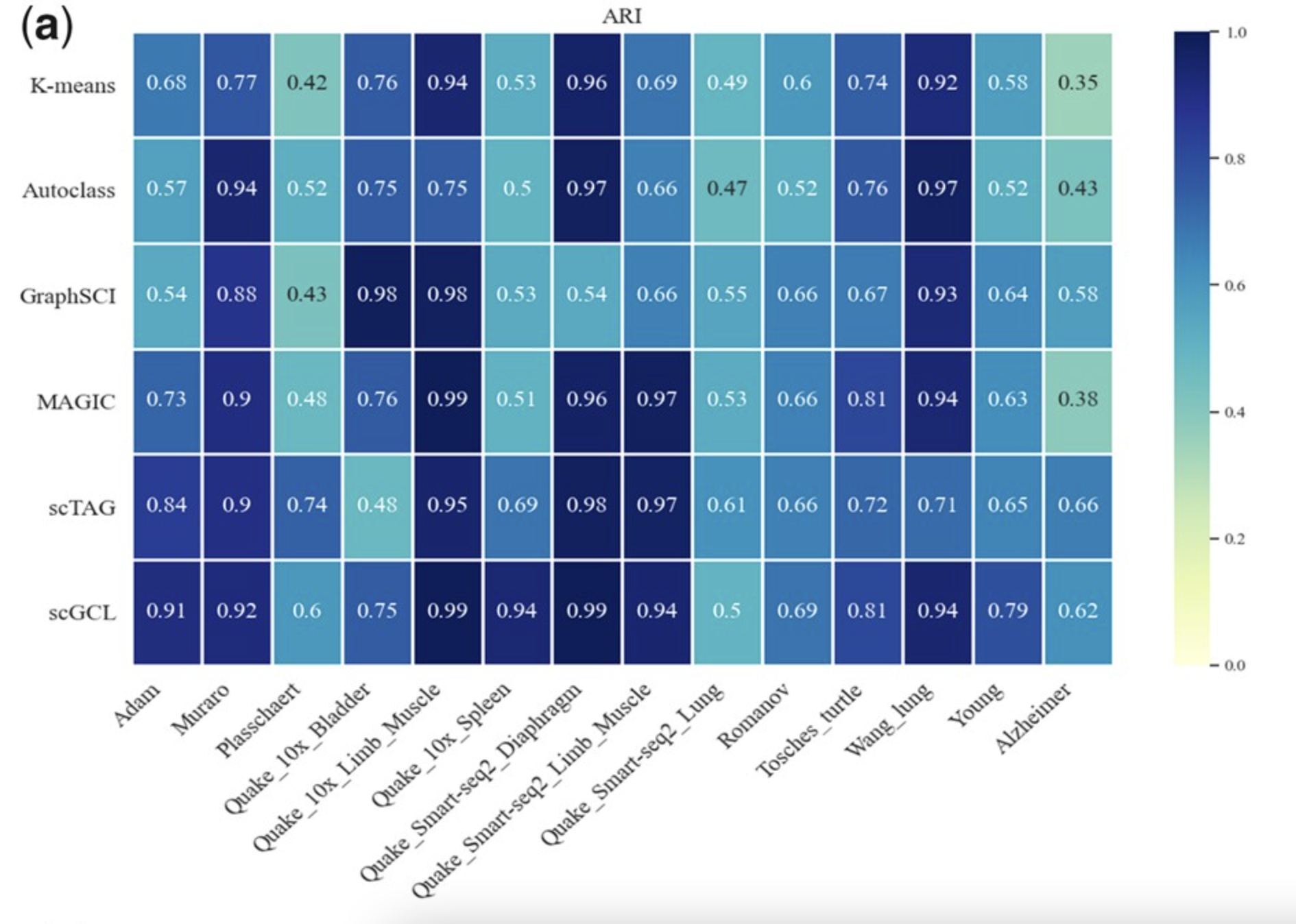


Figure a. scGCL ARI results compared to other baseline methods

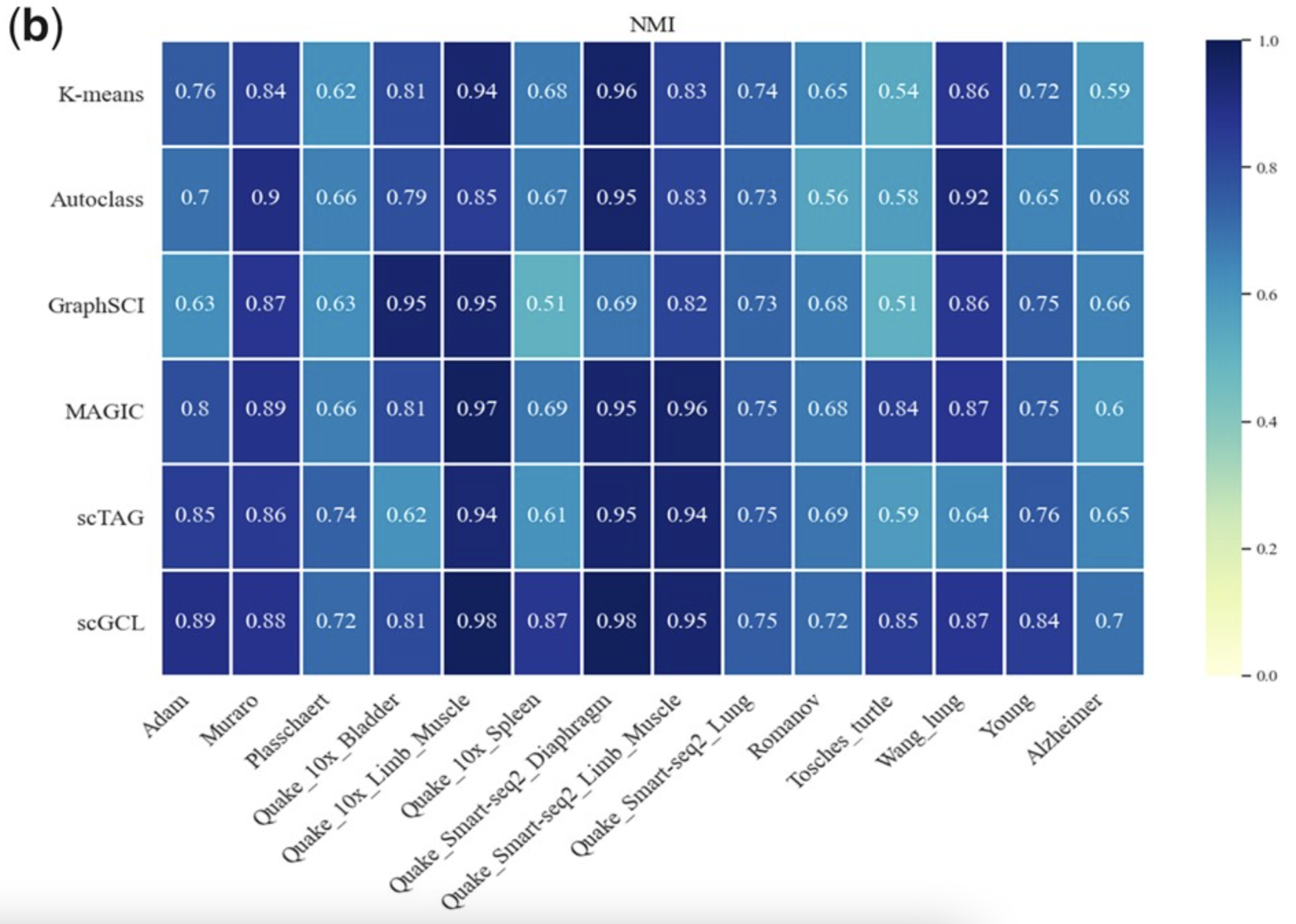
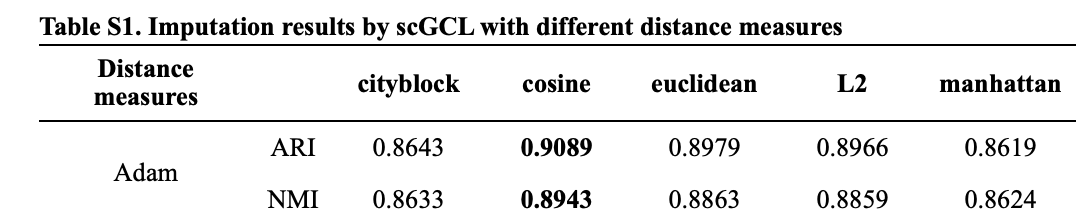


Figure b. scGCLl NMI compared to other baseline methods

Hyperparameter Analysis

For hyperparameter analysis, the *k* and distances measures are used as metrics for comparison. In this experiment, only the baseline method scTAG uses the K-NN method to construct cell graphs. The distance measures were set to ‘cityblock’, ‘cosine’, ‘euclidean’, ‘L2’, and ‘manhattan’ while the *k* values were set to 5, 10, 15, 20, and 25.

From the results, the ‘cosine’ distance measure had both the highest ARI and NMI for the Adam dataset. This means it captures the distance information/relationships between cells most effectively, which is explained due to the high sparsity of scRNA-seq data. Out of the *k* values, the ARI and NMI values are highest when *k* is set to 15. Additionally, scGCL always performs better than the baseline method scTAG across all distance measures and *k* values.



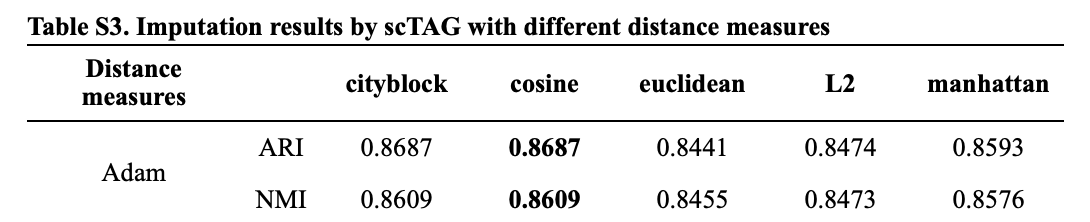
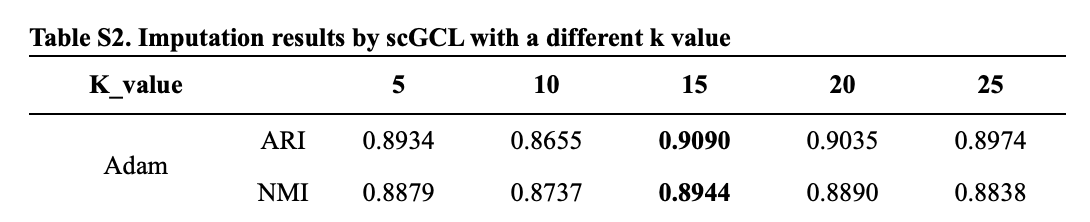
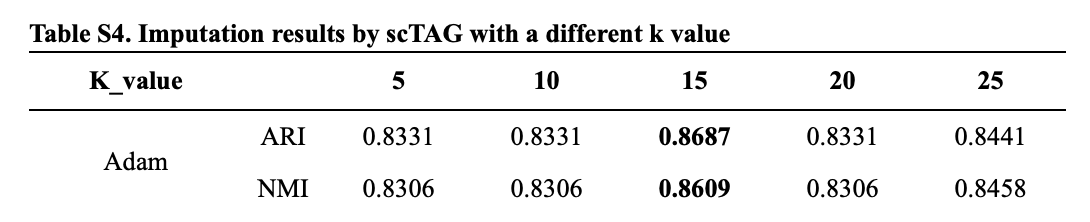
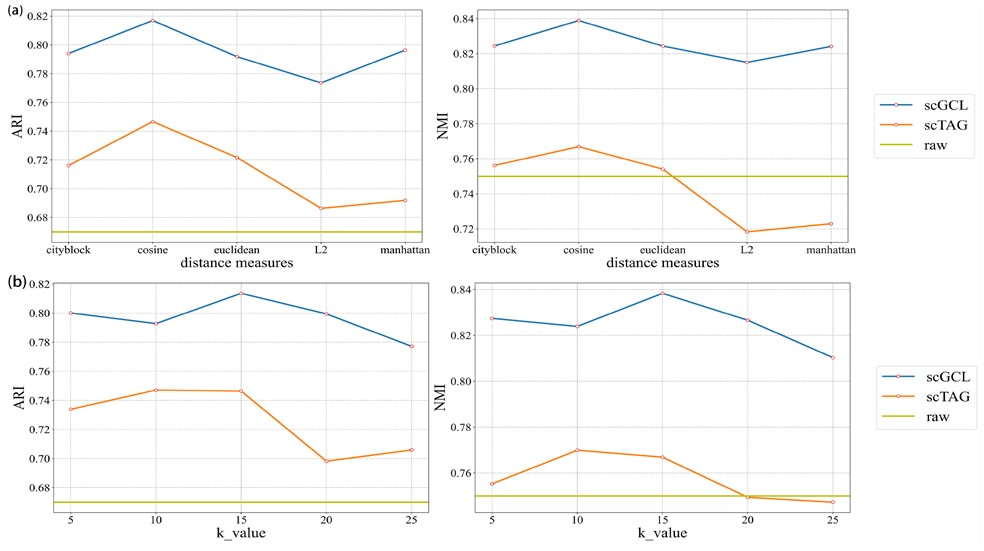


Table S1. Imputation results for Adam dataset



Average imputation results

Ablation Study

Although not implemented in our project, the paper also conducted ablation studies against the Adam dataset. Two key components were systematically removed to assess their impact on the system's performance:

The first scenario involved the removal of the ZINB-based encoder while maintaining the AFGRL and graph convolutional models. This allowed us to observe how the system performs without the contribution of the ZINB encoder, focusing solely on the effects of the AFGRL framework and the graph convolutional approach.

In the second scenario, the ZINB-based encoder was retained, but the graph convolution and AFGRL models were excluded. This setup aimed to isolate and evaluate the significance of the ZINB-based encoder by removing the influence of the other two components, offering insight into its standalone contribution to handling the datasets.

As shown by the values in Table \_\_, the method performs better with the ZINB distribution included, which captures the global probability distribution of scRNA-seq data. For the Adam dataset, having both the ZINB-based encoder and AFGRL model produced the best results.

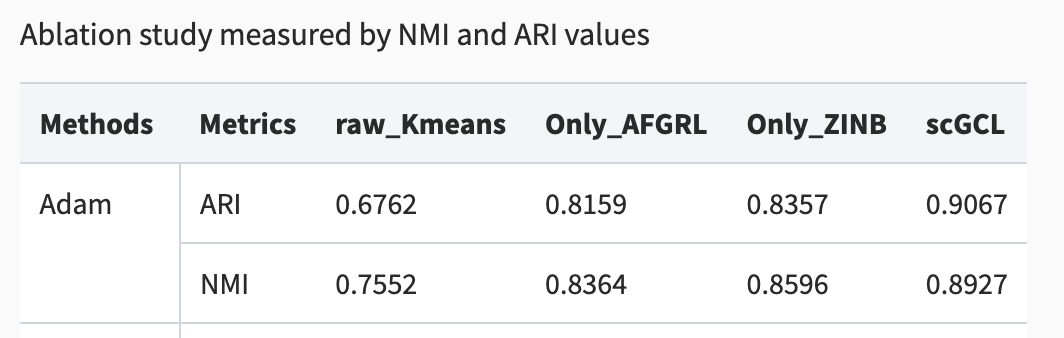


Table. Ablation Study Measured by ARI and NMI values

**DISCUSSION**

In this section, you should discuss your work and make future plan. The discussion should address the following questions:

* Make an assessment that the paper is reproducible or not.
  + The paper is reproducible, although revision is required.
* Describe “What was easy” and “What was difficult” during the reproduction.
  + Easy: Tasks, including data preprocessing, clustering, training, etc. were intuitively split up and easy to dissect
  + Difficult: Parts of code were deprecated and many portions of code were not necessary, not all datasets are readily or easily accessible
* Make suggestions to the author or other reproducers on how to improve the reproducibility.
  + Better annotations
  + More readily available data
* What will you do in the next phase.
  + Add more detailed annotations
  + Clean up any unnecessary code
  + Transfer code to github repo
  + Create data download instructions
  + Video report

# Discussion

## Instructions

In this section,you should discuss your work and make future plan. The discussion should address the following questions:

* Make assessment that the paper is reproducible or not.
* Explain why it is not reproducible if your results are kind negative.
* Describe “What was easy” and “What was difficult” during the reproduction.
* Make suggestions to the author or other reproducers on how to improve the reproducibility.
* What will you do in next phase.

# Instructions

Draft (20 Points)

Jupyter Notebook (both .PDF and .ipynb files)

You need to use the report template and fill out the following sections, each of which we will score based on the clarity and appropriateness of your writing (percentage of total grade for each component shown). All the information must be in the Jupyter notebook.

Introduction (2)

A clear, high-level description of what the original paper is about and what is the contribution of it

Scope of reproducibility (2)

Methodology (8)

Data

Data descriptions

Implementation code

Model

Model descriptions

Implementation code

Training

Computational requirements

Implementation code

Evaluation

Metrics descriptions

Implementation code

Results (8)

Results

Analyses

Plans

Before you use this template

This template is just a recommended template for project Report. It only considers the general type of research in our paper pool. Feel free to edit it to better fit your project. You will iteratively update the same notebook submission for your draft and the final submission. Please check the project rubriks to get a sense of what is expected in the template.

FAQ and Attentions

Copy and move this template to your Google Drive. Name your notebook by your team ID (upper-left corner). Don't eidt this original file.

This template covers most questions we want to ask about your reproduction experiment. You don't need to exactly follow the template, however, you should address the questions. Please feel free to customize your report accordingly.

any report must have run-able codes and necessary annotations (in text and code comments).

The notebook is like a demo and only uses small-size data (a subset of original data or processed data), the entire runtime of the notebook including data reading, data process, model training, printing, figure plotting, etc, must be within 8 min, otherwise, you may get penalty on the grade.

If the raw dataset is too large to be loaded you can select a subset of data and pre-process the data, then, upload the subset or processed data to Google Drive and load them in this notebook.

If the whole training is too long to run, you can only set the number of training epoch to a small number, e.g., 3, just show that the training is runable.

For results model validation, you can train the model outside this notebook in advance, then, load pretrained model and use it for validation (display the figures, print the metrics).

The post-process is important! For post-process of the results,please use plots/figures. The code to summarize results and plot figures may be tedious, however, it won't be waste of time since these figures can be used for presentation. While plotting in code, the figures should have titles or captions if necessary (e.g., title your figure with "Figure 1. xxxx")

There is not page limit to your notebook report, you can also use separate notebooks for the report, just make sure your grader can access and run/test them.

If you use outside resources, please refer them (in any formats). Include the links to the resources if necessary.