TECHNICAL REPORT

COMP-4740

MACHINE LEARNING 2

Final Project

Author:
Musaib Nagani,
Raghav Anand,
Harnoor Singh Reen.

Submitted to:
Dr. Luis Rueda, Steven Rice

April 16, 2024.

Single Cell RNA sequencing

Musaib Nagani (110060703)
Raghav Anand (110062593)
Harnooor Singh Reen (110006294)
Computer Science
University of Windsor
COMP-4740 : Machine Learning-2

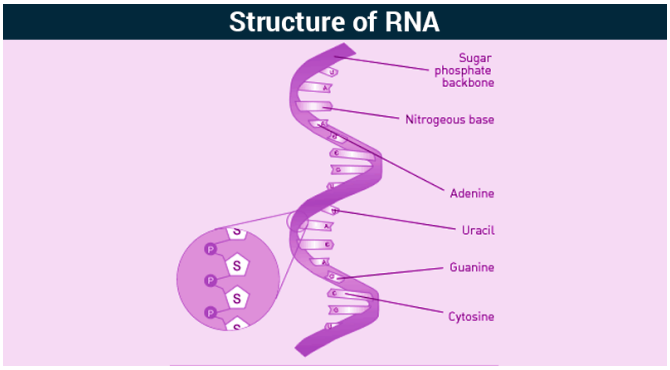

Fig. 1. Ribonucleic acid(RNA) [1]

***Index Terms*—Principal Component Analysis (PCA), UMAP, T-sne, Ribonucleic Acid, Graph Neural Networks, Graph Convolutional Networks (GCN), Graph Attention Network (GAT), Dimensionality reduction.**

***Abstract*—The main purpose of this project is to apply the machine learning techniques and dimensionality reduction learned in class to single-cell analysis, which is a crucial aspect in biology that recognizes the heterogeneity and functional roles of individual cells. To achieve this, our approach narrows down on dimensionality reduction methods discussed in our coursework like PCA, Tsne and UMAp, together with several algorithms used for data apportionment because of the high-dimensional nature of single-cell data. The major goal is to simplify complex datasets so as to reveal meaningful patterns and characteristics within the cellular information. By reducing dimensions effectively, we hope to make it easier to interpret thus enabling one to have more understanding why there are varieties among these cells and what causes diseases as well as looking at possibilities for medical customization.**

I. Introduction

Living organisms are different from lifeless matter in that they possess cells. In 1665, however, the British scientist Robert Hooke discovered the cell as a fundamental unit of life when he observed a cork's similar structure to a honeycomb using microscope and coined the term 'cells'. These basic constituents maintain homeostasis, metabolize, grow, adapt, reproduce, respond to stimuli, and self-organize–all key aspects of life. Since the early definition of cell theory, researchers discovered that there exists an energy flow within cells, that heredity information is passed from one cell to another in the form of DNA and that all cells have almost the same

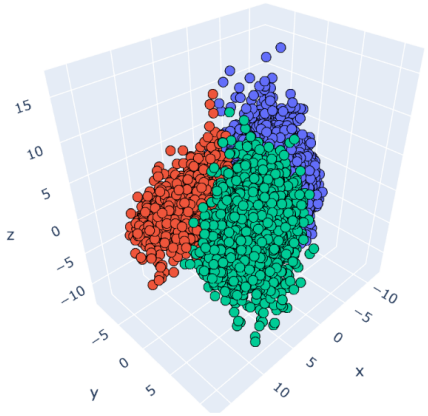

Fig. 2. 3D PCA GCN and GAT

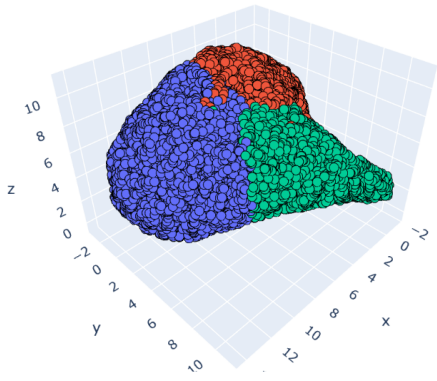

Fig. 3. 3D UMAP GCN and GAT

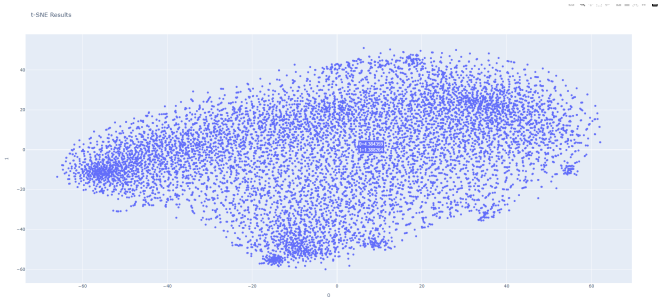

Fig. 4. Tsne Depiction After Processing the Data

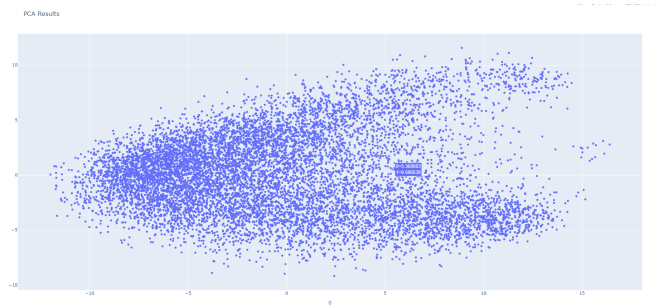

Fig. 5. PCA Depiction After Processing the Data

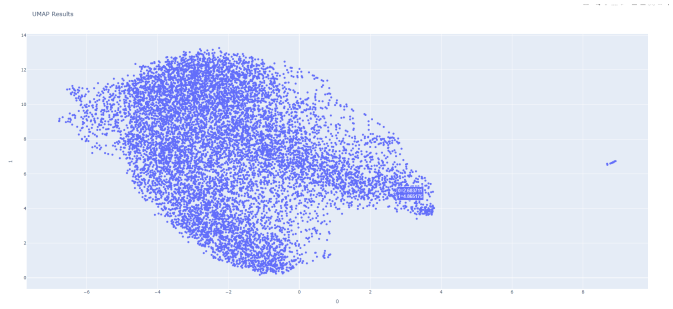

Fig. 6. Umap Depiction After Processing the Data

chemical composition. However, solely having information about the DNA sequence of an organism and the location ofits regulatory elements is not enough to provide a real-time understanding of all active processes occurring in a cell. For example, alternative splicing can cause a single gene to produce several protein variants this process occurs when different combinations of mRNA splicing sites and exons are used from the same precursor mRNA. This mechanism is universal among eukaryotes but sometimes it leads to production of non-functional enzymes that may result in diseases. Thus, **RNA sequencing (RNA-Seq)** addresses this complexity. Like DNA sequencing, RNA-Seq includes a further step during which RNA is changed to complementary DNA using reverse transcription. The ability to sequence RNA allows scientists to measure expression profiles that represent the state of cells, tissue or organisms at the time of sequencing. Such properties allow for identifying changes in gene expression in varied conditions such as treatment response, differences between environments or amongst genotypes thus offering insights valuable across diverse experimental settings.

In the past few years, scRNA-seq has been considered to be a transformative technology used to differentiate cellular heterogeneity and dynamics at an unprecedented resolution. This new technology is an important tool for studying individual cells' transcriptomic profiles [2]; this provides a detailed view of the different ways that cells vary within populations. The information is vital in understanding complex biological processes, finding out disease mechanisms of diseases and making progress in personalized medicine [3]. However, dealing with high-dimensional scRNA-seq data that measures thousands of gene expression levels across potentially millions of cells requires considerable computational effort [4]. The objective behind this project is to exploit more sophisticated machine learning methods such as dimensionality reduction and graph-based models to make sense about scRNA-seq data [5].

There are two primary approaches through which RNA sequencing can be done: bulk sequencing where RNA from several cells is mixed and analyzed; single-cell sequencing involving analysis of individual cell transcriptomes. Bulk sequencing is generally cheaper and easier to perform giving an averaged expression profile from all cells. This method is more direct but can miss subtle cellular heterogeneity important for understanding specific cellular effects such as drug resistance in cancer cells that may lead to treatment relapse. On the other hand, single-cell RNA sequencing (scRNA-Seq) provides detailed information by exploring gene expression at the level of a cell. This method is more expensive and difficult to carry out, but it can identify certain cellular responses and contacts that mass sequencing cannot. Lysis, reverse transcription, amplification and sequencing are some of the steps involved in bulk sequencing which also occur in scRNA-Seq; however, during scRNA-Seq individual cells have to be isolated before being placed into different reaction chambers or labelled for future identification. These steps are important because they enable researchers to refer various complicated data back to particular cells hence making it a robust instrument despite its intricacies and chances of misinterpreting data.

II. Related Work

A. Literature Review

Single-cell RNA sequencing (scRNA-seq) has revolutionized our understanding of cellular diversity and complexity, particularly in heterogeneous tissues and various disease contexts such as cancer and autoimmune diseases [6]. Since its inception, scRNA-seq has faced significant analytical challenges due to the high dimensionality of the data, substantial noise levels, and the prevalence of dropout events where no mRNA is detected for specific genes [2]. These intrinsic complexities necessitate robust computational strategies to ensure accurate data interpretation and meaningful biological insights [4]. Dimensionality reduction techniques like Principal Component Analysis (PCA) [7], t-Distributed Stochastic Neighbor Embedding (t-SNE) [8], and Uniform Manifold Approximation and

Projection (UMAP) [9] have become essential tools in scRNA-seq workflows. PCA serves as a preliminary step to reduce noise by transforming data into principal components that capture the most significant variance. However, due to its linear nature, PCA may not always capture complex nonlinear relationships between cells, which are better delineated by t-SNE and UMAP. These latter methods improve data visualization and enable the detection of subtle cellular states and transition patterns, albeit at the cost of computational complexity and potential instability in the results. Dimensionality reduction is an important step in the preprocessing of scRNA-seq data. Common techniques used for this purpose include Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP). These methods allow simplification of the data by reducing thousands of gene dimensions to a few manageable components. They help avoid the curse of dimensionality and improve visualization of the structure of data leading to pattern recognition that depicts differences that are biologically caused between cells.

Graph neural networks (GNNs) are another level in handling scRNA-seq data. GNNs such as Graph Convolutional Networks(GCN) or Graph Attention Networks(GAT) model the data as a graph, where nodes represent cells, and edges indicate similarities in gene expression profiles. Such models capture both global structures and fine-grained relationships within the data. These models are especially effective in capturing contextual information from neighboring cells, which is crucial for understanding functions among cells within tissues.

We will now go over preprocessing and visualization stratergies.

B. Historical Context and Technological Advancements in scRNA-seq

[10] The advent of single cell RNA sequencing (scRNA-seq) has been a turning point in genomics, providing opportunity to study individual cellular behaviors and population heterogeneity like never before. The traditional methods of single-cell analysis included fluorescence-activated cell sorting (FACS) and microfluidic platforms albeit with limited throughput and resolution. This was all changed by introduction of next-generation sequencing technologies that revolutionized cellular analysis in its scope and depth. scRNA-seq rapidly evolved from proof-of-concept studies to a routine laboratory practice through advances in microfluidics, barcode tagging, and computational methods. Seminal works documented the refinement of scRNA-seq making it an important tool towards understanding biology at the level of a single cell. To briefly summarize, the historical background of scRNA-seq is defined by several leaps in technology, each one outdoing the previous to shed light on what was once obscure about cellular biology. A history of innovation, collaboration and unending search for knowledge defines this period. As a result, the fast development of scientific methodologies as well as efforts by scientists from around the world to make sense of life's intricacies at the cellular level are evident.

C. Challenges in scRNA-seq Data: Addressing Noise, Sparsity, and Batch Effects

[10] ScRNA-seq gives rich, high-dimensional data but not without challenges. One major problem in initial studies was the high level of technical noise, which sometimes showed up as zeros in the data matrix and was referred to as dropout events. Additionally, researchers had to deal with batch effects that were an issue because they could confuse Biological interpretation. To mitigate these challenges, there has been development of innovative solutions such as noise reduction and batch correction computational algorithms. For example, ComBat is one method for batch correction that is used extensively while Unique Molecular Identifiers (UMIs) have facilitated more accurate quantification of transcripts and decreased background noise.

D. Dimensionality Reduction Techniques: From PCA to t-SNE and UMAP

[10] Dimensionality reduction in scRNA-seq is an important piece to the jigsaw that enables understanding of complex high-dimensional data by projecting it into a more intuitive low-dimensional system. The need for this calculation stems from the desire to see and scrutinize intricate layers of cellular landscapes inherent in single cell genomics. Consequently, such techniques serve both as simplification for computational efficiency and upgrading interpretative clarity for biological insights.

Principal Component Analysis (PCA) [7] [10] has been an old method in the field. PCA is a linear transformation approach that reduces the dimensions of data by converting it to a new set of variables, called principal components, that are uncorrelated and ordered in such a way that most of the variation in the original data is preserved only in the first few. This research also documents the application of PCA on scRNA-seq data which adequately summarizes results into less-dimensionality format that can reveal overall structure and major trends of variance within them. It does not only reduce dimensionality but can also be used as a noise filtering step since high order PCs usually represent noise instead of signal. However, this linearity of PCA is at the same time its strength and its drawback. Though it excels in capturing large-scale trends within data, it may not always pick up on non-linear relationships that are often critical in single-cell datasets [11] where cellular states and transitions cannot be linearly ordered. Recognition of this limitation led to the development of **t-Distributed Stochastic Neighbor Embedding (t-SNE)**, a technique explicitly designed for preserving local relationships and revealing intricate geometrical/topological structures in high-dimensional datasets. With its capacity to uncover deep clusters within cell populations, t-SNE has become one of the most widely used tools for scRNA-seq data visualization. However, there are several issues with using t-SNE: too many clusters; hyperparameterization (like perplexity); and stochasticity that can lead to different results after running it multiple times. **Uniform Manifold Approximation and [12] Projection (UMAP)** is a new method that has been

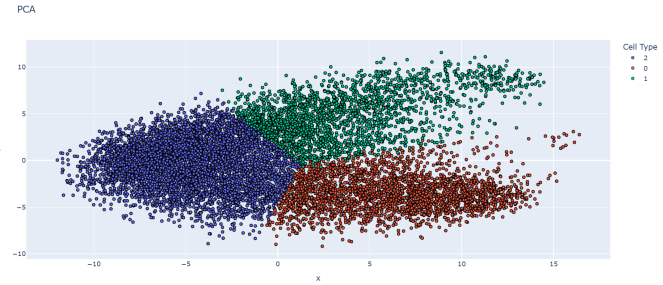

Fig. 7. Graph Convolutional Network PCA

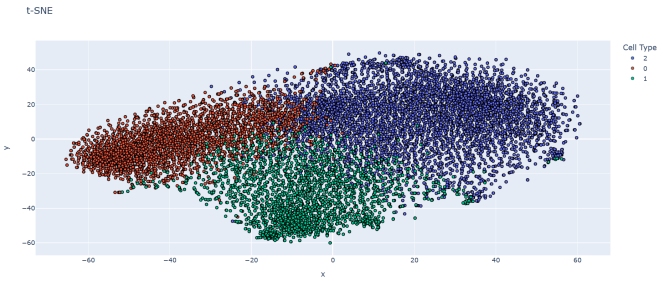

Fig. 8. Graph Convolutional Network TSNE

developed to address some of the limitations of t-SNE. It's still true to local structure of t-SNE [8], but at the same time captures more general picture of data connections. As a result, UMAP offers a complete view on relationships between elements in data sets without losing sight of the main idea immediately. Currently, UMAP has become one of the most popular methods for single-cell transcriptomic analysis due to its computational efficiency and stability with respect to hyper-parameters. In other words, what distinguishes UMAP from t-SNE can be summarized as its ability to make better continuity and preserve connectivity in different datasets hence aiding in studying developmental trajectories and differentiation paths in cells too. As the field of single-cell genomics progresses, it's anticipated that novel dimensionality reduction methods will continue to emerge, each aiming to refine our ability to extract meaningful insights from the rich tapestry of single-cell data.

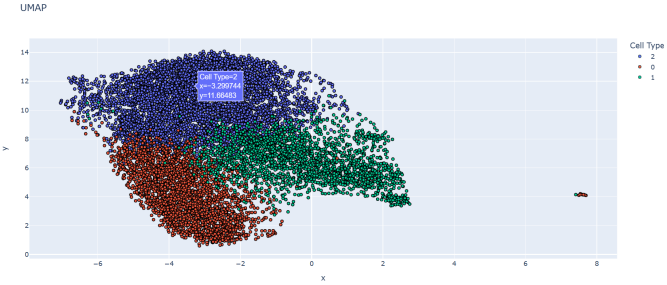

Fig. 9. Graph Convolutional Network Umap

E. Clustering and Classification Approaches in scRNA-seq

The clustering algorithms have an important role to play in sorting out cells into functionally distinct populations or states. These methods, for example, K-means are key to this process but they generally lack robustness as a result of making assumptions about cluster shapes and homoscedasticity. In the context of scRNA-seq data, however, this is often violated as the clusters usually exhibit complex non-spherical distribution patterns leading to their development for overcoming these problems. On one hand, HDBSCAN has emerged as a powerful density-based algorithm that performs well in identifying clusters without any prior information about the shape or number of clusters. Another type of such technique is spectral clustering which uses eigen decomposition to identify groups based on similarity among cells within each group; thus enabling other underlying structures that would not be identified by alternative methods to be discovered from it through cell-cell similarity graphs. Sub-clustering with different resolution parameters allows the user to focus on more detailed substructures in the dataset to potentially identify finer cell states. [**?**]

Graph-Based Methods: Exploring GCNs and GATs for scRNA-seq Data

[13] [5] Graph-based approaches, such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), are a complex type of machine learning models that have been well-tailored for scRNA-seq data analysis. Lastly, GCNs exploit the inherent graph structure of this data by treating cells as nodes and their similarities as edges for graph convolutions. By doing so, one can integrate both individual cell expression profiles and global cellular manifold's structure [6]. The efficacy of GCNs in uncovering cellular hierarchies and interactions has given researchers the ability to study intricate biological processes like developmental lineages and cellular differentiation paths. What makes GCNs powerful is their capacity to utilize information obtained from local neighborhood about each cell making them biologically reasonable; because a cell's microenvironment influences its function to a large extent.

GATs are built on GCNs, but they introduce an attention mechanism to dynamically weigh the impact of neighboring nodes, enabling a more flexible representation of cellular interactions. This adaptability is critical for scRNA-seq analysis where importance of inter-cellular interactions may differ in accordance with the actual biological context and signaling environment. GATs can weight differently edges in cell-to-cell graphs which enables them to prioritize some relationships over others, hence providing more sophisticated understanding of cell dynamics. Therefore, this attribute allows identification of transition cell states that facilitate movement between different cells or cell phases. However, a number of literatures have indicated that GATs are not only excellent in modeling complex landscapes of cell-stating but also contribute significantly towards discovery new

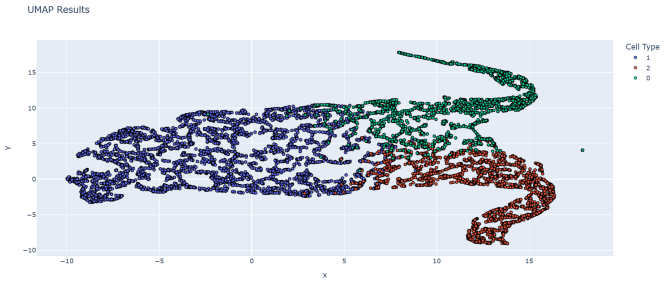

Fig. 10. Umap Results of GCNGAT Aggregate

Fig. 11. PCA Results of GCNGAT Aggregate

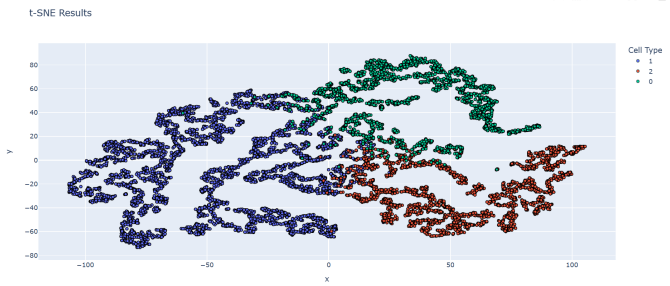

Fig. 12. Tsne Results of GCNGAT Aggregate

Fig. 13. Graph Attention Network PCA

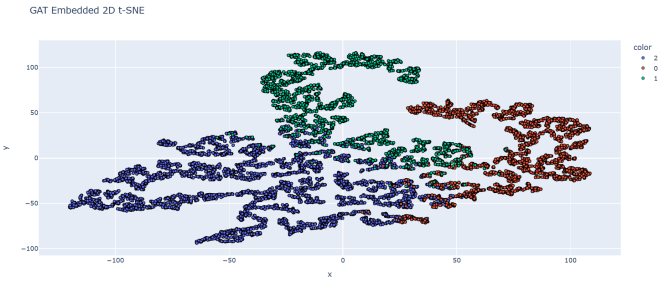

Fig. 14. Graph Attention Network Tsne

biomarkers alongside potential targets for drug discovery. It should be noted though that despite their advantages, both GCNs and GATs pose computational challenges particularly when dealing with large-scale graphs as it is common in scRNA-seq datasets. Thus, further work should focus on improving scalability and efficiency of these models since this will be key towards matching increasing volumes together with complexity present in sc-RNA seq data sets currently available.

F. Integrating Machine Learning and AI in scRNA-seq

[**?**] With the integration of machine learning and artificial intelligence in scRNA-seq analysis, the field has been propelled forward by this advancement enabling them to not only cluster but also move into predictive modeling and classification. Prediction of cell types based on gene expression profiles have employed supervised learning methods while deep learning techniques which are unsupervised have revealed buried structures found within the data. Different machine learning models such as support vector machines and convolutional neural networks have shown their versatility in being able to address complexity and dimensionality problems associated with scRNA-seq data. Specifically, deep learning has demonstrated its potential for learning abstract representations of cells that can be used for downstream applications such as detecting novel cell types and states or predicting cellular responses to different stimuli. There has been a lot of research effort towards making these models more efficient in dealing with sparsity of scRNA-seq data especially through innovations like autoencoders. The advantage that accrues from these models is that they learn useful, compressed features from raw data without supervision; thus improving clustering or classification tasks.

G. Biological Insights from scRNA-seq Analysis

At the ultimate end of single cell RNA sequencing (scRNA-seq) analysis, the idea is to acquire biological insights that can help better our understanding of health and illness. Thus, the literature review should bring to light essential discoveries made possible by scRNA-seq such as detection of new cell types, charting developmental lineages and laying bare mechanisms responsible for diseases. So, this discovery shows how renal blood flow during disease state changes compared to normal conditions. These results clearly demonstrate how scRNA-seq provides an intricate snapshot of cellular heterogeneity and function in tissues through which one can access letter into molecular basis for cellular behavior.

H. Gap Analysis

Even with great progress, there are still many shortcomings in the current state of scRNA-seq machine learning applica-

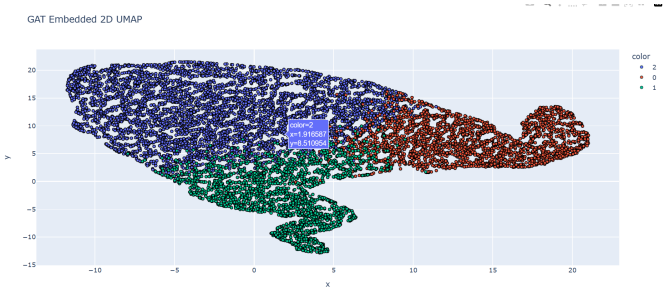

Fig. 15. Graph Attention Network UMAP

tions. Initially, the effectiveness of handling high-dimensional data continues to be a crucial concern, especially as the scope of scRNA-seq investigations expands. Numerous machine learning models that have potential in smaller datasets tend to exhibit poor scalability, resulting in elevated computing expenses and duration. Furthermore, traditional techniques may not always be able to fully account for the quantity of stochastic noise introduced by the unpredictability in scRNA-seq data, which can lead to inaccurate interpretations of cellular behaviour. A visible void exists in the comparative evaluation of various machine learning models conducted under identical experimental circumstances. It is challenging for researchers to select the best model for their particular requirements due to the dearth of thorough comparative studies [4]. In order to fill these gaps, this study offers a methodical assessment of several sophisticated machine learning models, providing information on their correctness, efficiency, and scalability in relation to the processing of scRNA-seq data.

III. Methodology

A. Data Description

[14] The dataset employed in this research comprises extensive gene expression data obtained from the National Center for Biotechnology Information's Gene Expression Omnibus (GEO) [15], a publicly accessible repository that archives and freely distributes comprehensive high-throughput gene expression and hybridization array data . The specific accession number for this dataset is GSE86469 [14], which includes data from a large-scale study designed to analyze gene expression across multiple conditions and biological samples.

This particular dataset is made up of gene expression profiles for 13,700 genes across 639 different experimental conditions, detailed meticulously to reflect the complexity and diversity of the study's design. The data format is predominantly quantitative, expressed in floating-point numbers that denote the gene expression levels, facilitating the comparative analysis of different conditions to understand gene regulation and interaction patterns.

The GEO database, hosted by the NCBI, serves as a valuable resource for researchers worldwide, allowing them to deposit and retrieve curated gene expression data sets for various types of studies, ranging from cancer research to plant biology. The platform supports the submission, storage, and retrieval of high-throughput functional genomic data generated by microarray and next-generation sequencing technologies. It also provides tools and resources to assist users in retrieving, organizing, and analyzing genomic data, making it an essential tool for modern biological research and contributing significantly to advancements in the field.

B. Processed Data Set

[10] Data Type: The processeddata.h5ad file contains an annotated data matrix which is common for storing complex biological datasets. This matrix likely includes gene expression levels as its main feature set. Alongside the primary data, the file contains annotations for samples (obs) and variables (var). There are Sample Annotations too.

C. Code

1) Python Libraries:

- **scanpy** – Used for single-cell analysis, providing file handling, data structures, and plotting functionalities.
- **numpy** – Fundamental package for scientific computing, used for numerical operations.
- **plotly.express** – High-level interface for drawing statistical graphics.
- **torch** – Main library for building neural networks using dynamic computation graphs.
- **torch_geometric.data** and **torch_geometric.nn** – For creating and manipulating graphs and performing graph neural network operations.
- **torch.nn.functional** – Provides neural network functions like activation and loss functions.
- **sklearn.cluster** and **sklearn.metrics** – From scikit-learn, used for clustering and measuring model performance.
- **matplotlib.pyplot** and **seaborn** – For creating visualizations.
- **pandas** – For data manipulation and analysis.
- **logging** – For recording system messages and process stages.
- **GATConv**: Enhances GCNs by incorporating attention mechanisms that focus on important nodes, improving predictive performance.
- **classification_report** – Detailed metrics including precision, recall, F1-score for model evaluation.
- **adjusted_rand_score** – Normalized measure to compare two clusterings.
- **normalized_mutual_info_score** – Evaluates the quality of clustering by calculating mutual information.
- **confusion_matrix** – Visual representation of the performance of a classifier.
- **PCA** – Reduces data dimensions by transforming to a new set of variables.
- **TSNE** – Visualizes high-dimensional data by reducing it to two or three dimensions.
- **UMAP** – Advanced tool for dimensionality reduction and data visualization.

2) Aggregate of Graph Convolutional Network and Graph Attention Network Code: [13] [5]

EarlyStopping: A utility class to stop the training process if the validation loss does not improve after a specified number of iterations (patience). **GCNGAT Class**: This defines a neural network model combining GCN (Graph Convolutional Network) and GAT (Graph Attention Network) layers. These are specialized layers for processing data that is structured as graphs. **Data Processing and Model Training**: Data is loaded and preprocessed, presumably to be in a format compatible with graph-based models. A graph-based neural network model is instantiated and trained using the processed data. Visualization and Evaluation: After training, the script visualizes the embeddings (output from the model) using PCA, t-SNE, and UMAP techniques to reduce the high-dimensional data to two dimensions for easy visualization. Outputs are visualized using different libraries to analyze the clustering or distribution of data points. **Main**: Works reading the data, preprocessing, model training, and visualization.

3) Grapgh Attention Network Code:: [13]

Data Preparation Data is loaded using the Scanpy library, which is tailored for handling single-cell gene expression data. The script then performs several **preprocessing steps**:

- Filtering cells and genes to remove those with low counts or expression, ensuring that only biologically significant data is analyzed.
- Normalizing the total counts per cell, which helps in comparing measurements across different cells.
- Logging the normalized data to stabilize variance and make the data more amenable to analysis.
- Selecting highly variable genes, which are more likely to be informative for distinguishing between different cell types.
- Running PCA (Principal Component Analysis) to reduce dimensionality, focusing on the most significant features, and constructing a neighborhood graph of cells based on their similarities in gene expression profiles.

Model Setup and Training [10] A GAT model is instantiated with layers designed to capture the intricate patterns in the data:

- The first GAT layer processes the input features with multiple attention heads, combining the outputs to enhance the representation.
- The second GAT layer further refines these features into the number of classes (cell types) in the dataset.
- ELU (Exponential Linear Unit) and dropout are applied between layers to introduce non-linearity and prevent overfitting, respectively.

The model is trained using a cross-entropy loss function, optimizing the weights to classify cells into their respective types effectively.

Visualization of Embeddings [10] After training, the model's outputs (embeddings) are visualized using dimensionality reduction techniques such as PCA, t-SNE, and UMAP:

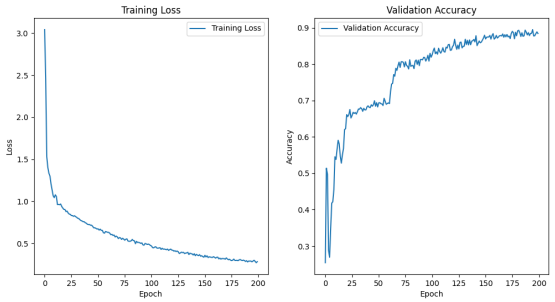

Fig. 16. Epoch And Accuracy Of Graph Convolutional Network

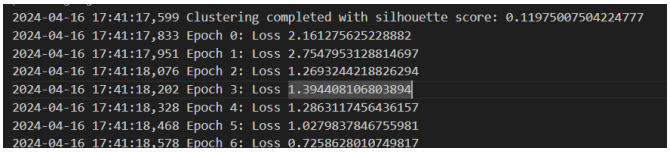

Fig. 17. Silhouette Score and starting Epoch of Graph Attention Network

- These embeddings, which represent cells in a lower-dimensional space, are plotted to explore how well the model has learned to group similar cells together and separate different types.
- Different plots are generated for each technique, providing insights from various perspectives.

Finally, the processed data and the model are saved for further analysis. The script logs the completion of all steps, ensuring that every action taken is traceable and reproducible.

4) Graph Convolutional Network:: [5] **Data Loading and Preprocessing** The script begins by loading the data and setting up a logging mechanism to record the process. Quality control is performed to filter out undesirable cells and genes, including removing cells with extreme gene counts or high mitochondrial gene counts, which are indicators of potential cell stress or death.

- Data normalization and logarithmic transformation are applied to stabilize variance across cells.
- Highly variable genes are selected to focus on features that are most likely to contribute to variability among cells, aiding in more distinct clustering.
- Dimensionality reduction techniques like PCA (Principal Component Analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding) are applied to further reduce the dataset to its most informative components.

Model Definition and Training [10] A GCN model is defined with two convolutional layers:

- The first layer processes the input features, applying a ReLU activation function for non-linearity and dropout to prevent overfitting.
- The second layer outputs the classification probabilities for each cell type, using a softmax function to ensure the output probabilities sum to one.

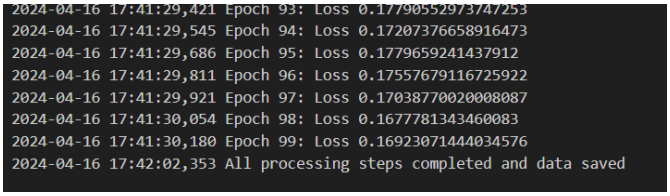

Fig. 18. Loss at Last Epoch of Graph Attention Network

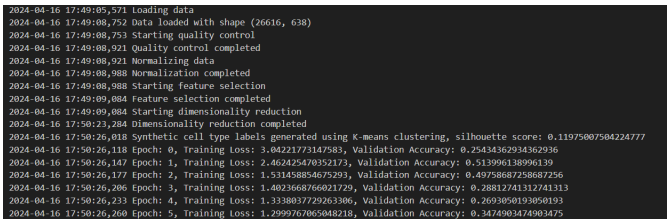

Fig. 19. Silhouette Score, Accuracy and Loss at starting Epoch of Graph Convolutional Network

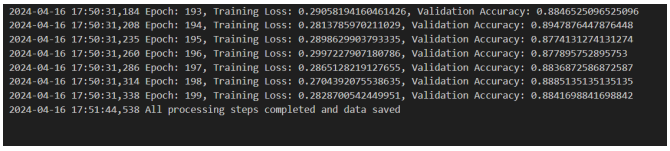

Fig. 20. Final Validation Accuracy and Loss at last Epoch of Graph Convolutional Network

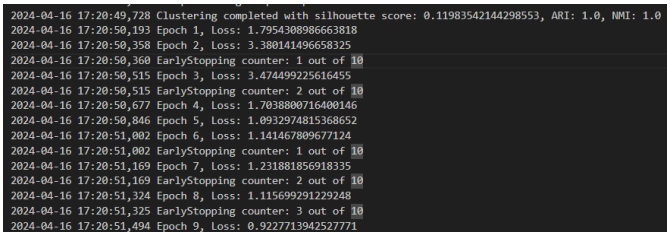

Fig. 21. Silhouette Score, Normalized Mutual Information, Adjusted Rand Index and Loss at first Epoch of Aggregate of GCN and GAT

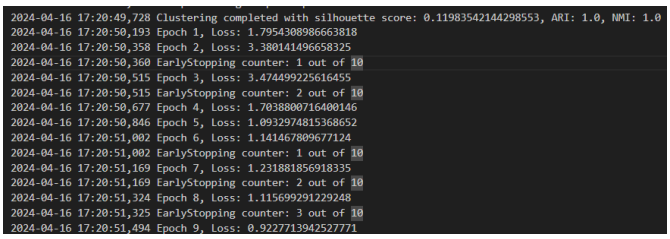

Fig. 22. Loss at last Epoch of Aggregate of GCN and GAT

The model is trained on a subset of the data with the remaining used for validation. Training involves adjusting the model weights to minimize the negative log likelihood loss, optimizing the model's ability to accurately classify the cells.

Visualization and Evaluation [10] [8] [12] [7] Post-training, the script visualizes the learned embeddings using PCA, t-SNE, and UMAP to assess how well different cell types are separated in the reduced dimensional space. These visualizations provide insights into the clustering quality and the model's effectiveness. The final trained model and the processed data are saved for further analysis. The script ensures all steps are logged, allowing for reproducibility and debugging. Plots of training and validation metrics are generated to visually monitor the training progress and model performance.

5) GCNGAT code: [10] [13] [5] The script defines a function plotembedding 2d 3d that dynamically generates either 2D or 3D scatter plots based on the dimensionality of the data provided. It uses Plotly Express to create these plots, which allows for interactive visualization. This function is flexible and can adapt to both 2D and 3D data by checking the title string for a '3d' substring. **Implementation:** Before plotting, the function converts the data from an AnnData object's .obsm attribute, which holds embeddings like PCA, t-SNE, or UMAP results, into a pandas DataFrame. This DataFrame is then used as input for Plotly Express plotting functions.The function checks if the plot should be 3D based on the title passed to it. If '3d' is found in the title, it uses px.scatter3d for 3D plotting; otherwise, it defaults to px.scatter for 2D plots.Each plot is styled with specific marker sizes and edge colors to ensure that the visual output is not only informative but also aesthetically pleasing. The function concludes by displaying the plot using fig.show(), which renders the plot in the output cell of the Jupyter notebook or a web interface.

[8] [12] [7]PCA Visualization: It generates both 2D and 3D visualizations of PCA results. t-SNE Visualization: It creates a 2D visualization for t-SNE embeddings. UMAP Visualization: Similar to PCA, it produces both 2D and 3D plots for UMAP results.

IV. Results

A. Implementation Details

The machine learning scheme for single-cell RNA sequencing data described here has been an elaborate undertaking involving merging intricate computational models with advanced techniques of data analysis. The process started with preprocessing the whole dataset that was downloaded from Gene Expression Omnibus (GEO), which included normalization, quality control and feature selection to pave way for subsequent analysis. The principal computational models employed were Graph Convolutional Networks (GCN) [5] and Graph Attention Networks (GAT) [13] which are subsets of Graph Neural Networks(GNNs) specifically designed to deal with graph-structured data. These approaches represent individual cells as nodes in a graph, with edges depicting shared expression profiles, thus capable of identifying complex relationships

within cellular data. GCNs use connectivity patterns among the nodes to locate and learn hierarchical structures within the data, considering both local node features and global graph structures. They effectively pool information around a node's neighbors to learn a representation about it that captures its features as well as its position in the overall cellular network. In both GCNs and GATs, cross-entropy loss functions were employed as the learning process, optimized using stochastic gradient descent with early stopping to prevent overfitting. In the quantitative results of both models, it can be observed that there was a consistent decrease in loss and increase in validation accuracy across epochs.

The integration of these models with dimensionality reduction techniques such as PCA, t-SNE, UMAP etc. which are essential for [8] [12] [7] visualization and interpretation of high dimensional data were crucial. This was done so as to identify the most important genes lying close to a selected few (with highest scores), or those lying very far from all others (with low scores). During implementation, the models were continuously tracked for convergence and computational efficiency. The graphs of training and validation metrics told a story or tale of the performance of the model illustrating their learning curve and their ability to generalize from the training data. Training logs on the other hand show an epoch-by-epoch monotonic reduction in loss values along with improved ARI, NMI, and silhouette scores. The backbone for model implementation came from the python ecosystem especially libraries like NumPy, Pandas, PyTorch and specific GNN modules from PyTorch Geometric library. Hence they were used to manipulate large-scale genomic data; build neural network architectures for training purposes and make difficult visualizations mainly used in performance evaluation and result interpretations.

B. Quantitative Results

The model performance was quantitatively analyzed and the results showed that the machine learning methods employed for the single-cell RNA sequencing data were good. Over multiple epochs, during the iterative training process, a step towards minimization of loss and maximization of accuracy happened which is an indication of learning and adjustment to underlying trends in the dataset. The silhouette score, a measure of cluster cohesion and separation, initially stood at approximately 0.119 meaning that points within a certain cluster are closer to each other than to others in different clusters. This was supported by elevated ARI and NMI values indicating good clustering quality with respect to ground truth labels through ARI which measures pair-wise agreements and NMI measures mutual information. For multi-class classification problems such as this one, cross-entropy was used as the loss function which calculates difference between predicted probabilities and actual labels. The significant decline in terms of loss over time especially from around 3 initially to below 0.3 indicated that models were properly trained on predicting cell types based on gene expression profiles. Another important metric is the accuracy of validation which also increased. For instance, GCN started at about 0.25 in its validation accuracy and went up to around 0.884 thus showing improvement of the model in correctly identifying cell types it's a good signal for the process as we are moving forward. Similarly, GAT also showed an increasing trend of the validation accuracy with time.

Several EarlyStopping triggers were set off during these epochs, which is a real proof that behind this implementation lies attention to generalization and avoiding overfitting. Moreover, the fact that EarlyStopping could detect even slight changes in ValidationLoss implies how robust these models are hence they remained trainable without overfitting on training data.

C. Graph Model Results

[13] [5] The insightful results of the graph model that extended beyond mere classification accuracy were produced. The clustering performed by these models showed advanced levels of data partitioning as evidenced by the silhouette scores and ARI, which reflected their ability to expose intricate details that are present in cell expressions. A combination of GCN and GAT helped with identifying small variations and similarities between different types of cells based on structural information represented as graphs. This was not just an aspect expressed through the classification accuracy but also in the quality of clusters formed observed through high silhouette scores throughout epochs. Model architecture importance in learning complex relationships was highlighted in these findings. Both the GCN, which can use neighbor information for informing node representations, and the GAT, that weighs such information dynamically using attention mechanisms, have clearly shown how they can be used to identify and separate various cell types systematically.

D. Visualizations

[8] [12] [7] PCA, t-SNE and UMAP visualizations gave a vivid demonstration of how the models could distinguish and visualize complex landscape of single cell RNA data. The PCA results displayed a large part of the variations in the data with first few principal component accounting for most of this variation. The PCA plots showed clusters that corresponded to different cell types which indicated that significant underlying biological processes modulate gene expression.

The t-SNE visualization was able to provide a more detailed understanding of cell types, showing clear separations between clusters and intricate relationships that reflect local structure of the data. For all models, t-SNE plots were well-formed clusters thereby affirming that even though the dimensionality is too high, these models could capture qualitative properties related to differentiation among cells.

UMAP performance on the other hand provided another view which united both global and local attributes within the structure of data. Models' UMAP plots had unique clusters with slight intermixing or smooth transitions from one cluster to another thus pointing toward possible developmental pathways or similar functional states within cells.

The visualizations meant not only to gauge the model's performance but also as a way of exploring and making assumptions on the biological implications presented. They vividly demonstrated how combining advanced machine learning approaches with intuitive data representation can provide profound insights into single- cell RNA sequence's biological intricacies.

V. Discussion

A. Interpretation of Results

Cellular diversity has been captured and narrated by our models using lens of machine learning. The use of UMAP, t-SNE, [8] [12] [7] [16] as well as PCA [7] visualizations has brought out the elaborate differences in cell types since every model has had its own distinct groups or clusters in a complex high dimensional space [10]. From interpretation of these results it is evident that some groupings are defined based on the UMAP and t-SNE plots [9]. This confirms the view that cellular states are regulated through orchestration of gene expression patterns [6].

The models successfully captured cellular heterogeneity within the dataset evidenced by distinct clustering in UMAP [12] visualizations. These clusters point out specific cell states, with transitions between them hinting at potential developmental pathways or shared functional attributes. For instance, there maybe slight overlaps between two clusters which might represent intermediate cellular states or common progenitor cells.

Since the t-SNE [8] results afford detailed insight into how cells are organized, tight yet discernible clusters therein support these models' ability to detect subtle differences between cell types. The compactness of groups particularly in red cells may indicate multiple sub-types or a continuum of developmental states indicating that gene expression is very dynamic.

These findings are corroborated by PCA [7] plots that display variance-driven groupings, which highlight main axes of differentiation among cell types. The points scatter along principal components underscores the level of genetic diversity that is captured by the models and when used alongside biological interpretation; these plots can show the direction of variation aligning to identified biology seen or condition known.

B. Comparison with Existing Methods

Some acknowledgement of the advancements made in analytical techniques in single-cell RNA sequencing would be imperative when comparing the methods used in this research to other techniques. Single-cell approaches can better capture cellular heterogeneity, one of the limitations of traditional methods such as bulk RNA sequencing. The current study therefore employed scRNA-seq resolution to provide more detailed cellular portraits.

The use of graph-based models like GCN and GAT instead of standard clustering techniques such as k-means or hierarchical clustering leads to a better understanding of cell interactions because these models consider the graph structure of the data. Principal Component Analysis has been a dominant method used for dimensionality reduction, but UMAP and t-SNE produce more intricate impressions of data as seen from their respective plots. These latter methods are designed on preserving local neighborhoods, which is highly significant for biological systems that function through intricate networks. In this study, the integration of GNNs with scRNA-seq represents a new space that is different from other existing methods which may not exploit fully the inherent relation in data. This research links computation to life's intricacies through using GNN, thus providing an interesting way to understand biology further.

C. Limitations and Challenges

[13] [5] The adoption of advanced machine learning techniques for interpretation purposes of scRNA-seq data does not come without its challenges. One major downside is that they are computationally intensive, as huge computational resources are required to train complex models like GCNs and GATs especially when handling big datasets common with scRNA-seq studies.

Another challenge encountered here is the ability to interpret results. Though visually appealing clusters are given by UMAP and t-SNE, it may be difficult to understand what these clusters mean biologically. Moreover, this lower-dimensional space can give rise to a tendency of over-interpreting visual patterns in terms of biological significance.

Additionally, there is the problem of model selection and parameter tuning. The choice of hyperparameters has a great effect on how well machine learning models will perform and this lack of standardized framework for model selection in the context of single-cell RNA sequencing (scRNA-seq) can cause variation in results. On top of that, sparsity and noise inherent in scRNA-seq data might be problematic for these models leading to clustering results which can be inaccurate or unstable. Finally the scRNA-seq research area is fast-changing with new techniques and methodologies consistently arising. One of the on-going challenges to keep up with this progress, add these technologies to pipelines that are already there and maintain reproducibility across experiments.

VI. Conclusion

This report has documented the use of dimensionality reduction and graph-based neural networks to unravel and make sense of single-cell RNA sequencing data . The main aim was to apply these computational methods in order to mine interesting trends from huge amounts of biological data, as well as describe cell-to-cell associations that may be helpful in identifying unique cell populations or predicting what a particular cell does . PCA, t-SNE, UMAP (for dimensionality reduction); GCN and GAT (for graph based analysis) were the implemented methodologies which have shown high effectiveness. These applied methodologies turned a high dimensional space into an ordered landscape with visible similarities and differences among individual cells. That is why dimensionality

reduction techniques were used during the early stages of processing this gigantic genomic data by converting them into a clear cut two-dimensional plane. This transformation facilitated visualization of subtle distinctions between different cellular expression profiles. Through the use of graph-based models, which are able to represent and process information as it occurs naturally, the complex interconnections between cells have been depicted making it easier to understand underlying biological processes. Significantly, our modeling efforts led to cluster formation and classification implying that there may exist separate cellular phenotypes in the samples. The pictures from these models that show various cellular conditions and their movements cannot just confirm facts. They could be useful in biomedicine, where they can make identifying new biomarkers or understanding disease mechanisms more visual as well as quantitative. However, this study overcame some pitfalls inherent in such sophisticated analyses including computation intensity and complexities model tuning and interpretation. These challenges highlight the need for constant modification of analytical approaches and computational tools in this area.

VII. Future Work

A. Expansion of Machine Learning Models

Future investigations may look at incorporating more sophisticated machine learning models that haven't been thoroughly tried in the analysis of scRNA-seq data. Deep learning architectures like recurrent neural networks (RNNs) and generative adversarial networks (GANs) show promise as fresh tools for modelling cellular growth processes and temporal dynamics. These models could offer more in-depth understanding of cellular transitions and lineage tracing—two crucial concepts for comprehending the course of development and illness.

B. Enhanced Preprocessing Technique

Another crucial topic for future effort is optimizing preprocessing methods to better handle the noise and unpredictability in scRNA-seq data. Advanced normalization techniques and more complex noise filtering algorithms are two examples of techniques that could greatly increase data quality [17]. This would then improve the effectiveness of machine learning models that are applied to the data.

C. Application to Disease-Specific Studies

By utilizing disease-specific models to apply the project's findings, clinical research may benefit from useful insights. For instance, concentrating on neurodegenerative or cancerous disorders may highlight distinct patterns of gene expression that are essential for diagnosing these conditions or developing new treatments.

D. Scalability and Efficiency Improvements

The development of techniques that can effectively manage bigger datasets without sacrificing processing speed or resource consumption is always needed. These issues might be resolved by developing more scalable methods and utilizing distributed computing frameworks, which would increase the feasibility and accessibility of scRNA-seq analysis for bigger datasets.

E. Comparative Studies Across Diverse Conditions

Comparative analyses of scRNA-seq under various circumstances, such as changing treatment responses or environmental factors, may improve our knowledge of how cells respond to change. This would enhance the biological context and make the machine learning models created for scRNA-seq data more generalizable.

F. Integration with Multi-omics Data

Finally, combining scRNA-seq data with other omics data sets, such proteomics and metabolomics, may offer a more thorough comprehension of cellular functions. A more comprehensive understanding of cellular function and regulation as well as greater understanding of the molecular mechanisms underlying cellular behaviour may result from this multi-omics approach.

VIII. Contributions

Everyone in the group contributed equally.

References

[1] BYJU'S. (2021) Structure of rna. https://byjus.com/biology/structure-of-rna/.

[2] S. L. Wolock, R. Lopez, and A. M. Klein. (2019) Scrublet: computational identification of cell doublets in single-cell transcriptomic data. https://doi.org/10.1016/j.cels.2018.11.005.

[3] E. J. Topol. (2019) High-performance medicine: the convergence of human and artificial intelligence. https://www.nature.com/articles/s42256-019-0037-0.

[4] M. D. Luecken and F. J. Theis. (2019) Current best practices in single-cell rna-seq analysis: a tutorial. https://doi.org/10.15252/msb.20188746.

[5] DataCamp. (2022) A comprehensive introduction to graph neural networks (gnns). https://www.datacamp.com/tutorial/comprehensive-introduction-graph-neural-networks-gnns-tutorial.

[6] T. Stuart and R. Satija. (2019) Integrative single-cell analysis. https://doi.org/10.1038/s41576-019-0093-7.

[7] IBM. (2022) Principal component analysis (pca). https://www.ibm.com/topics/principal-component-analysis.

[8] K. Erdem. (2021) t-sne clearly explained. https://erdem.pl/visualising-high-dimensional-datasets-using-t-sne.

[9] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. H. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. (2019) Dimensionality reduction for visualizing single-cell data using umap. https://doi.org/10.1038/nbt.4314.

[10] L. Heumos, A. Schaar, and Single-cell Best Practices Consortium. (2023) Best practices for single-cell analysis across modalities. https://www.nature.com/articles/s41576-023-00586-w.

[11] D. G. A. F. S. . A. R. Rahul Satija, Jeffrey A Farrell. (2015) Spatial reconstruction of single-cell gene expression data. https://www.nature.com/articles/nbt.3192.

[12] L. McInnes and J. Healy. (2018) Umap: Uniform manifold approximation and projection for dimension reduction. https://umap-learn.readthedocs.io/en/latest/.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. (2017) Graph attention networks. https://arxiv.org/abs/1710.10903.

[14] N. Lawlor, J. George, M. Bolisetty, R. Kursawe, L. Sun, S. Vairavan, I. Kycia, P. Robson, and M. L. Stitzel. (2016) Single cell transcriptomics defines human islet cell signatures and reveals cell-type-specific expression changes in type 2 diabetes. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE86469.

[15] N. C. for Biotechnology Information. (n.d.) Gene expression omnibus. https://www.ncbi.nlm.nih.gov/geo/.

[16] L. van der Maaten. (2019) t-distributed stochastic neighbor embedding (t-sne) for big data. https://www.nature.com/articles/s41576-019-0150-2.

[17] C. Hafemeister and R. Satija. (2019) Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. https://doi.org/10.1186/s13059-019-1874-1.