A Pipeline for Analysis of scRNA-Sequence Data

Atul Kumar
110143931
atulkum@uwindsor.ca

Nawaf Nazeer
110158601
nazeer@uwindsor.ca

Prithvika Babu
110127176
babu91@uwindsor.ca

Tasmia Islam
110058801
islam76@uwindsor.ca

ABSTRACT

Single-cell RNA sequencing (scRNA-seq) analysis, although it can provide valuable cellular and genomic insights, is challenging due to the high complexity of the data as well as its high dimensionality. In an effort to reduce this complexity, dimensionality reduction techniques are used to extract features that have significance from this large dataset. After dimensionality reduction, differentiating cell populations and visualizing cellular heterogeneity are achieved through the use of clustering techniques. These clustering techniques will also help in visualizing the dataset and identifying any correlation of the features extracted. Following this, classification approaches evaluate how well clustering results are achieved using various dimensionality reduction strategies. This project incorporates a comprehensive investigation of different dimensionality reduction techniques and clustering techniques to investigate and identify which technique will be optimal for this dataset and similar ones.

KEYWORDS

scRNA-seq Analysis, Dimensionality Reduction, Clustering Algorithms, Classification Models

ACM Reference Format:
Atul Kumar, Nawaf Nazeer, Prithvika Babu, and Tasmia Islam. 2024. A Pipeline for Analysis of scRNA-Sequence Data. In *Proceedings of COMP8720 - Representation Learning (Project Report).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Cells are the basic blocks of all living organisms, and numerous aspects of cells are being researched to produce medicines, antibiotics, etc. One such method that offers good insights into cellular heterogeneity and understanding gene expression dynamics at a cellular level, is the single-cell RNA sequencing (scRNA-seq). This is used since bulk RNA sequencing results in a cell-averaged expression that is simpler but may mask significant cellular heterogeneity, whereas scRNA-seq which provides the resolution needed to identify any issues with different cell types or the interactions within intricate tissues, especially in situations like drug resistance. However, there are drawbacks to scRNA-seq research, which is the high cost and experimental complexity, as well as the possibility of drawing incorrect conclusions from analysis because of improved data resolution. Despite these difficulties, bulk RNA-seq and scRNA-seq methods are similar, with modifications for single-cell isolation, amplification, and sequencing. One crucial stage in scRNA-seq research is transcript quantification, which can be accomplished using either full-length or tag-based techniques, each with different consequences for recording gene expression data. Thus, the sequencing protocols are chosen as per the goal of the research being conducted.

2 DESCRIPTION OF PROBLEM

The scRNA-sequence datasets basically contain essential biological factors that allow researchers to explore cellular diversity, gene regulation, and biological processes at the single-cell level. However, the scRNA-sequence datasets have massive amounts of information, and this high-dimensionality results in challenges, such as data sparsity and high computational complexity. Many genes are either not expressed or expressed at very low levels in individual cells, leading to high levels of data sparsity. Unless these features are overlooked, it will result in overfitting as well as high complexity while processing. In addition, the sparsity of data will complicate the analysis and interpretation, thus, it will require specialized algorithms and techniques to accurately capture meaningful biological signals. Furthermore, scRNA-seq datasets typically involve measurements for thousands of genes across hundreds or thousands of individual cells, resulting in high-dimensional data. Managing and analyzing such high-dimensional data can be computationally intensive and may require dimensionality reduction techniques for feature extraction, visualization and interpretation.

In addition, the cellular heterogeneity present in the scRNA-seq dataset may remain unexplored which will limit insights into the composition, diversity, and functional states of the cellular extractions. Apart from that, distinguishing between different cell types or identifying rare cell populations is challenging due to the volatile nature of the dataset which hinders the research of understanding the biological significance of individual cell populations. These populations, in addition, may have relationships or correlations that may remain unidentified due to the high complexity of the data, as well as their natures i.e. if they are similar or different. In order to overcome these challenges, we propose implementing dimensionality reduction as well as clustering to not only be able to extract the useful features, but also to identify any existing relationships

Project Report, April 21, 2024, University of Windsor

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

on the extracted features. Lastly, we used the cluster labels to train the ML models to predict the unseen data.

3 DESCRIPTION OF SOLUTION

As stated in the problem statement, using scRNA-sequence datasets may encounter challenges such as data sparsity and high computational complexity. The purpose of this research is to build a pipeline that could overcome the challenges mentioned in the problem statement.

Firstly, we are using a dataset [3] from the textbook Single cell best-practices [4], which was already normalized and went through the process of feature selection. This dataset has 14814 observations, which represents cells' barcodes and 20171 variables, which corresponds to gene identifiers. Each entry in the feature matrix corresponds to the expression level of those genes in each cell. However, working with these many dimensions may not always be a good choice as it can easily lead to the curse of dimensionality, as data in the higher dimensional spaces may contain the information that may not be useful for the visualization, clustering, or cells classification.

After importing the dataset, we have applied several dimensionality reduction techniques to reduce the original dimensional space to 2D space for passing it further to the pipeline. The aim is to find the reduction technique which captures the maximum information, while mapping it to 2D space. Along with that, it would help visualize this higher dimensional dataset effectively and reduce the overall complexity of the dataset making further processes computationally fast and efficient. The implementation includes several dimensionality reduction techniques such as PCA, KPCA, Standard LLE, Hessian LLE, MLLE, Spectral Embedding, t-SNE, and UMAP. We will be discussing the results and how effective each of them is in the next section.

Once we have the reduced dimensional space, we are passing this to the several clustering algorithms to identify the clusters sharing the same characteristics. The aim is to find the algorithm that well separates the cluster from other clusters. But, the next challenge is to find the optimal number of clusters. However, using the indexes like Dunn Index, Davies-Bouldin index, etc., to find the optimal number of clusters wasn't an efficient approach for a limited RAM system as it was taking more than available computational space. So, after trial and error, it has been found that forming seven clusters is a good approach, as it separates the clusters well and reduces the overlapping between them to some extent for some of the reduced spaces. Also, it's not including the cells that are quite dissimilar. Several clustering techniques were implemented such as K-Means, GMM, Spectral Clustering, Louvain, and Leiden.

After clustering, we found out the cluster labels to which each cell belongs. That cluster labels were for every clustering algorithm and for each of the dimensionality reduction techniques, generating different clusters labels by each of them. However, we considered each of them as their true labels and performed multi-class classification tasks to classify the unseen cells based on the cells we fed while training.

For the classification task, we aim to classify the cells according to the cluster labels generated from the last step and implement two most popular models: Stochastic Gradient Descent and Random Forest Classifier. The reason behind implementing them is that the computation cost was quite high to train the models, which were overcome by these algorithms, due to their faster operations execution [2].

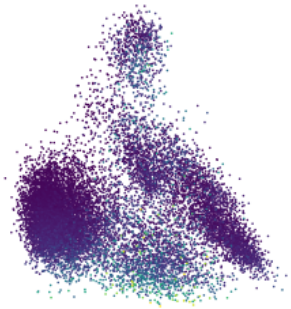

Figure 1: PCA Reduction

(1) **Stochastic Gradient Descent:** Since it picks a random sample at every step and computes the gradient based on only that sample. This makes it quite faster than algorithms like SVC.
(2) **Random Forest Classifier:** It generates multiple decision trees simultaneously, which contributes to predicting the cell label quite faster than decision trees as it splits the data into subsets once at each step. Once we classified the labels, we have tested the models on the unseen data to see how effectively they are learned.

4 RESULTS AND DISCUSSIONS

Here we will first see the visualization of our dimensions-reduced data. Following that, we will see the clustering results for each of the dimensionality-reduced data. Lastly, we will see the accuracy of the ML models on the test set using the labels we got through clustering.

4.1 Dimensionality Reduction

4.1.1 PCA:. PCA also known as principal component analysis is one of the oldest and most simplest dimensionality reduction techniques. It works by reducing the dimensions of the datasets by identifying the directions of maximum variance and projecting the data onto a lower dimensional space spanned by these principal components. The scRNA-seq dataset is made up of vectors in numerous dimensions, and because of that, not every feature is necessary. These vectors are converted into a new set of uncorrelated variables via Principal Component Analysis (PCA), which ranks the variables according to variance and reduces dimensions by minimizing loss of information. In the PCA of the dataset used, the features are extracted and the scatterplot shows the relationships of the features, here, areas of dense and sparse data points, and the total count of the features based on their occurrence in the dataset.

Although PCA is efficient and interpretable, but it may not be the best option for visualizing non-linear, sparse single-cell RNA sequencing (scRNA-seq) data as shown in Figure 1. However, it can be used to choose a portion of the best PCs for further examination.

Figure 2: KPCA Reduction

Figure 3: Standard LLE Reduction

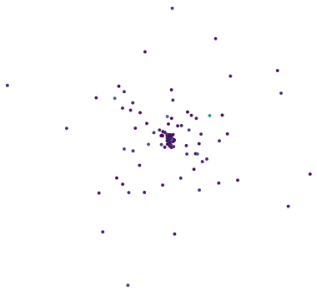

Figure 4: Hessian LLE Reduction

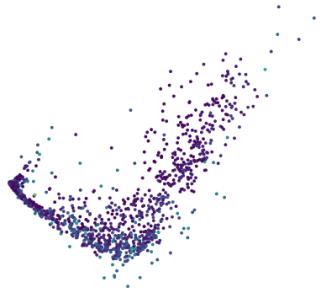

Figure 5: Modified LLE Reduction

4.1.2 KPCA:. In Kernel Principal Component Analysis (KPCA), the kernel serves a similar purpose as the covariance matrix does in linear PCA. The kernel selected depends on the scenario, and in this, a rbf (Radial Basis Function) kernel is used due to the non-linearity and high amounts of sparse data present in the scRNA-seq dataset. [3] After reducing dimensions, the result is plotted on a scatter plot in Figure 2.

As observed in the scatter plot, the data points are more dense than sparse using KPCA in comparison with PCA. There does exist some sparsity of data, but the accuracy is more than that of PCA. While KPCA can capture non-linear relationships and high dimensionality data, it does have the limitation of being sensitive to outliers or anomalies in the data, which can distort and affect the quality of the dimensionality reduction.

4.1.3 LLE:.

(1) **Standard:** The scatter plot shows the distribution of data points in the reduced-dimensional space obtained by applying LLE with the standard method. Points that are close to each other in the plot are considered to be similar in the original high-dimensional space. The color bar indicates the total counts associated with each data point, providing additional information about the density or importance of the points. After reducing dimensions, the result is plotted on a scatter plot in Figure 3.
By visualizing the embedded data, we can observe the local structure and relationships between data points. LLE aims to preserve local relationships, hence the points that are close to each other in the plot are likely to have similar local structures in the original high-dimensional space. This visualization helps in understanding the intrinsic structure of the data and identifying any clusters or patterns that may exist.

(2) **Hessian LLE and Modified LLE:** Hessian Locally Linear Embedding (HLLE) and Modified Locally Linear Embedding (MLLE), which differ in the method used to retain the structure of the data. HLLE puts more emphasis on capturing the original shape and curvature of the original high-dimensional space by using a mathematical tool called the Hessian that captures the curvature at a specific point. MLLE, on the other hand, puts more emphasis on the relationship between neighboring points in the original space.
From Figures 4 and 5, we can see that the two models generate very varied results, with HLLE showing multiple points superimposed while retaining the original structure, while MLLE shows the data points closer to each other along a uniform pattern.
It should be noted that although the various reduction algorithms were implemented using the "auto" eigen-solver, which uses the ARPACK solver, HLLE, in particular, was implemented using the "dense" eigen-solver. This is mainly because the ARPACK solver struggles with singular matrices, while the implementation of HLLE in sklearn primarily uses a singular weight matrix.

4.1.4 Spectral Embedding: The plot shows the data points in a 2D space after being embedded by Spectral Embedding. Spectral Embedding aims to preserve local neighborhood relationships in the high-dimensional space, so nearby points in this plot are expected

Figure 6: Spectral Embedding Reduction

to be similar in the original space. After reducing dimensions, the result is plotted on a scatter plot as shown in Figure 6.

The scatter plot gives an overview of the data's distribution in the reduced space, which could be useful for further analysis or visualization. The code employs Spectral Embedding for dimensionality reduction, condensing high-dimensional data into a 2D space. Through parameter settings like the number of neighbors and components, it captures local relationships while preserving essential information. The resulting scatter plot visualizes data points in the reduced space, with colors. This approach facilitates understanding complex data structures and can guide subsequent analyses or visualizations effectively.

4.1.5 t-SNE:. As we know, t-SNE focuses on preserving the local structure of the cells. This means the cells that are neighbors in the higher dimensional space are likely to remain close to each other in the new 2D space. We set the parameter to X_pca for representation of the cells, which means it uses PCA as a preprocessing step to reduce the dimensionality while preserving the most of its variance and computes the t-SNE embedding. t-SNE computes the conditional probabilities between the neighbors in the original space and try to keep the same pattern in the reduced dimensions. However, unlike PCA, it tries retaining the same closeness for only the cells that are neighbors w.r.t. to a particular cell (local structure), but doesn't apply to the cells which were not its neighbors in the original space (global structure)

As our dataset was in quite a higher dimension containing non-linear relationships, which we reduced only to two dimensions that had the maximum variance (information). After projecting it to 2D space, it can be visualized that t-SNE captures the local structure very effectively. As we can see in the Figure 7, t-SNE generated clusters of similar cells that could be easily separated using clustering algorithms.The ones that are close in the following figure were also close in the original space, and may have some similar characteristics or may share common features or attributes.

4.1.6 UMAP:. Just like t-SNE, it also calculates the distance between every pair of cells in original dimensional space and tries to preserve the distance between them the same when it maps to low dimensional space. However, the advantage of UMAP is that it captures not only the local structure like t-SNE, but also captures global structure in original space, which means it's not limited to retaining the same closeness between cells that neighbors, but also retains the same distance between the cells that are not neighbors.

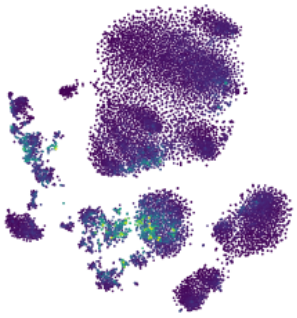

Figure 7: t-SNE Reduction

Figure 8: UMAP Reduction

The cells closest to each other in original space will be clustered together in the lower 2D space, and ones far apart will be in different clusters.

For our dataset, the results seem pretty good, in Figure 8, as it was able to well separate the cells that are closer to each other from the ones which were far apart in the original space, enabling identification of cells exactly as it would have appeared in higher dimensions. This means, instead of maintaining just the locality, it further preserved the global structures as shown in the plot below.

Since they are well separated we can apply the clustering algorithms to identify the cells having similar characteristics for further processing.

4.2 Clustering

4.2.1 PCA:. The results of PCA are clustered using different clustering techniques as shown in Figure 9. In K-Means clustering, the clusters have smaller inter-cluster distances and bigger intra-cluster distances, which makes it not as efficient as a clustering technique for this dataset. In addition, the clustering doesn't acknowledge the sparsity of the data, which tends to be a disadvantage here. However with GMM clustering and spectral clustering, we can see that the clusters are efficiently captured and wrapped in different shapes, with GMM clustering performing significantly better than the spectral cluster due to the 5th and 6th clusters being shaped properly, wherein in spectral clustering, clusters 6 and 5 are somewhat mixed together. In the Leiden and Lovuain clusters, the Louvian cluster has a mixture of cluster 2 in two locations, which is inefficient, however,

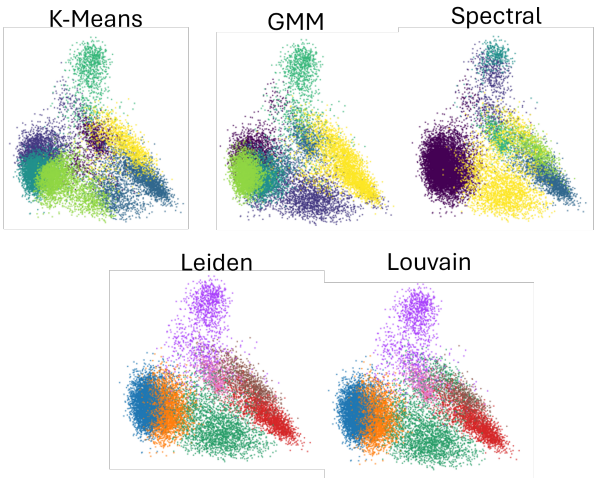

Figure 9: Clustering on PCA Reduced data

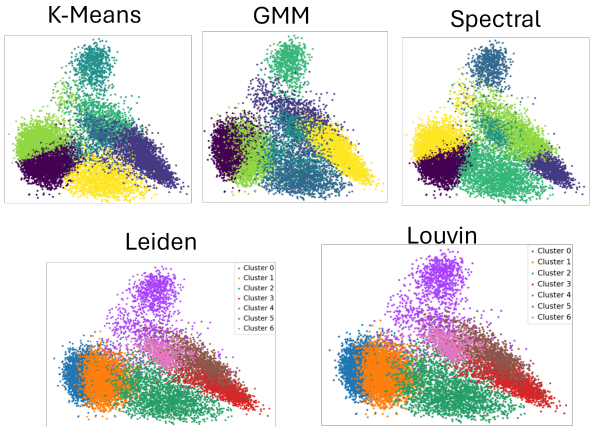

Figure 10: Clustering on KPCA Reduced data

the Leiden cluster can accurately cluster the brown cluster (cluster 5), rather than two green clusters. This is because the Leiden cluster enhances the quality of community detection by refinement [5].

4.2.2 KPCA:. Figure 10 shows the result of clustering techniques for the dimension reduced dataset of scRNA-seq by KPCA. As observed, we can see that in GMM, the clustering is done in an inefficient way due to the clusters having high sparsity. In Kmeans, cluster 2 and 3 are getting mixed together, which is not the best way to cluster, however, it captures the 3 clusters 1, 4 and 6 efficiently. Spectral clustering, like K-Means, also efficiently captures the bottom clusters, but also segregates the other clusters well. In addition, the Leiden and Louvain clustering techniques were also implemented, and it this, its observed that both these clustering techniques has resulted in a similar outcome, since both clustering techniques are closely related, but the Leidan clustering seems to outperform the Louvain clustering [1] by a small fraction since it is able to accurately cluster the Cluster 5, whereas Louvain was clustering inaccurately clustered Cluster 5 as Cluster 2.

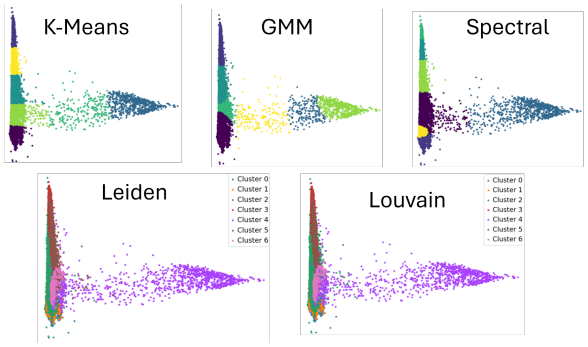

Figure 11: Clustering on Standard LLE Reduced data

4.2.3 LLE:.

(1) **Standard:** The Figure 11 shows

- **K-Means Clustering:** In the plot for K-Means clustering, we observe distinct clusters where each group is formed around a central point or centroid. K-Means clustering aims to minimize the variance within each cluster, ensuring that the data points are as close as possible to their respective centroids. The effectiveness of K-Means in this plot is evident from the clear boundaries between clusters, which suggests a successful separation of data into coherent groups post-LLE dimensionality reduction. This method is highly efficient for spherical clusters, as shown in our results.
- **Gaussian Mixture Model Clustering:** The Gaussian Mixture Model (GMM) clustering plot shows a probabilistic model approach, where each cluster can be considered to have emerged from a different Gaussian distribution. Unlike K-Means, GMM accommodates varied cluster shapes and can include covariance among the data points within each cluster, resulting in more flexible cluster assignments. This flexibility is noticeable in the smooth transitions and overlap areas between clusters in the plot, indicating that GMM considers the probability of membership to multiple clusters.
- **Spectral Clustering:** Spectral clustering uses the eigenvalues of a similarity matrix to reduce dimensionality before clustering, which can capture complex cluster structures that are not necessarily compact or separated by linear boundaries. The plot from spectral clustering post-LLE shows well-differentiated clusters even in data that may have intricate shapes. This method is particularly useful for identifying clusters that are connected through a graph structure.
- **Louvain Clustering:** Louvain clustering is a community detection method used primarily on graph data, optimized here through LLE. In the plot, we observe that Louvain clustering has grouped the nodes into communities based on modularity optimization. This method efficiently discovers hierarchically structured communities, which is evident from the dense interconnections within clusters

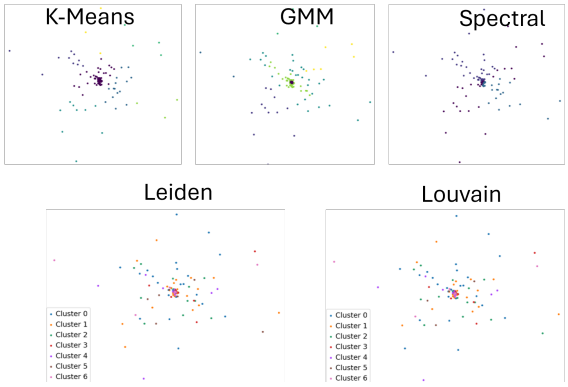

Figure 12: Clustering on Hessian LLE Reduced data

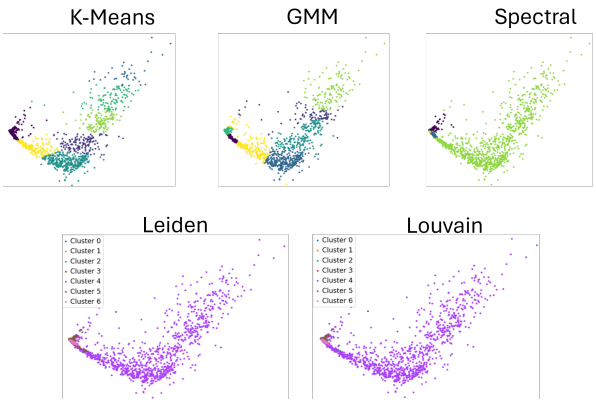

Figure 13: Clustering on Modified LLE Reduced data

compared to between clusters. It is ideal for large datasets due to its scalability and speed.

- **Leiden Clustering:** Finally, the plot for Leiden clustering, which we approximated using the Louvain method due to limitations in our tools, also shows community detection based on graph structures. Leiden algorithm improves upon Louvain by offering a finer level of refinement and potentially higher quality of community detection, although our plot uses Louvain as a stand-in. In practical use, Leiden would typically show even more defined community structures compared to Louvain, with a stronger focus on local optimization.

(2) **Hessian and Modified LLE:**

- **K-Means Clustering:** As we can see from Figures 12 and 13, K-Means Clustering works well for both HLLE and MLLE, with reasonable and distinct clusters. However, due to the poor reduction using HLLE, the algorithm could only create five clusters as opposed to the pre-defined parameter of 7 clusters. On the other hand, the reduction from MLLE resulted in clear boundaries between the clusters formed using K-Means, with data points being equally distributed among the clusters due to the distance-based working of the algorithm.

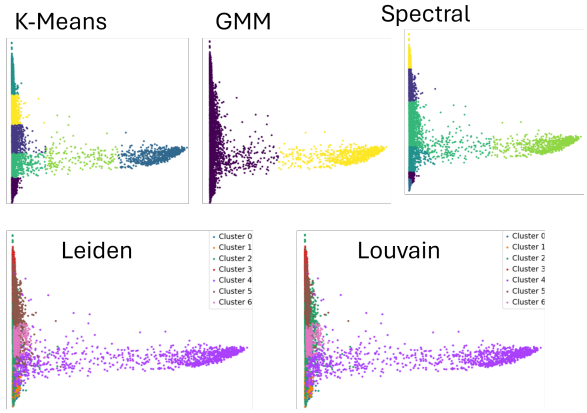

Figure 14: Clustering on Spectral Embedding Reduced data

- **Gaussian Mixtures Model Clustering:** Similarly, we can notice that in the case of GMM, the poor reduction from HLLE results in poor clustering with unclear boundaries between the different clusters. For MLLE however, although the cluster boundaries are much more defined, the probabilistic nature of GMM resulted in data points not being equally distributed among the various clusters.
- **Spectral Clustering:** Spectral Clustering produces poor separation between the various clusters for both reduction algorithms despite its ability to capture complex cluster structures. Although the clusters formed with HLLE show some separation vertically, a majority of the clusters are concentrated in the middle, while for MLLE, the clusters are mainly concentrated at the tip of the 'L' shape. This can be attributed to the fact that the implementation of Spectral Clustering was done using the "nearest neighbors" affinity, essentially working as a community detection algorithm, this can lead to identifying tightly knit communities in the data but a poor performance on globally well-separated clusters.
- **Leiden and Louvain Clustering:** Leiden and Louvain, both being community detection methods, had similar performances with MLLE with clusters being concentrated at the tip of the 'L' shape, although Leiden provided better separation and distinction between the clusters as compared to Louvain. This can be attributed to Leiden being an enhancement of Louvain with more robust partitions between the clusters. However, due to the poor reduction of HLLE, both approaches performed extremely poorly with no clear separation between the data clusters.

4.2.4 Spectral Embedding: The Figure 14 shows

(1) **K-Means Clustering:** There are four different clusters marked by various colors. After applying Spectral Embedding, the data points are transformed into a space where clusters are more discernible. This transformation aids K-Means in effectively segregating the clusters even if the original data structure was complex and intertwined.

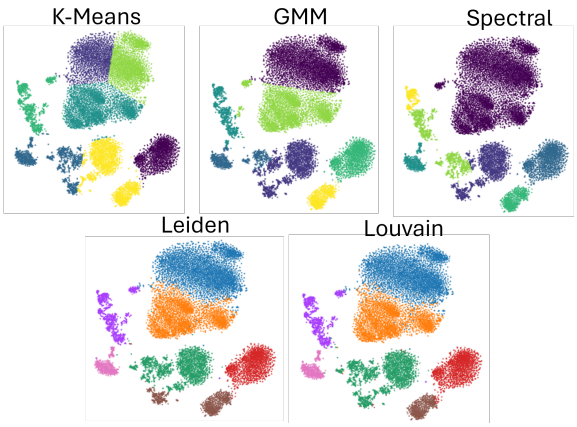

Figure 15: Clustering on t-SNE Reduced data

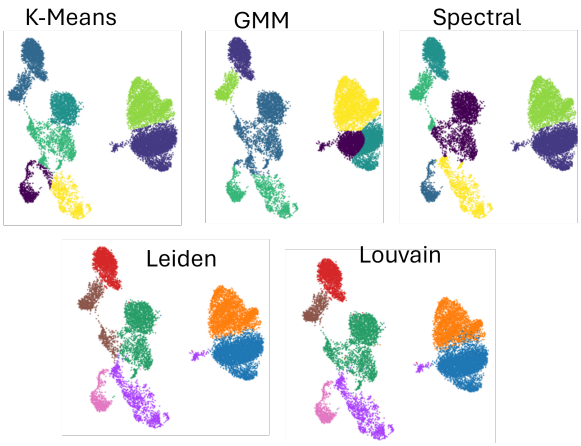

Figure 16: Clustering on UMAP Reduced data

(2) **Gaussian Mixture Model Clustering:** Some colors differentiate the most probable cluster assignments for each point based on the underlying Gaussian distributions. The effectiveness of GMM combined with Spectral Embedding is evident, as the method handles overlapping clusters well, where each data point can share membership in multiple clusters.
(3) **Spectral Clustering:** It showcases clusters identified through the spectral clustering method. These clusters may represent complex structures that traditional methods like K-Means could not resolve. The spectral embedding beforehand helps in emphasizing the separation by mapping the high-dimensional data into a lower-dimensional space that preserves the neighborhood structures.
(4) **Louvain Clustering:** Louvain clustering has grouped the data into distinct communities. The spectral embedding helps by positioning similar data points closer together in the reduced space, thereby enhancing the effectiveness of the Louvain method in detecting more coherent and distinct communities.
(5) **Leiden Clustering:** Results of Louvain clustering as a stand-in for Leiden (due to limitations in the environment), the interpretation would be similar. The plot reflects robust community detection where spectral embedding has facilitated the clustering process by enhancing the local neighborhood structures within the data.

4.2.5 t-SNE:. As we can see in Figure 15, K-Means separates the clusters in a linear fashion. Depending on the cell to centroid distances, it allocates the cells to that cluster. Linear boundary can also be seen between two clusters, which is dividing them. In the case of EM, however, it may struggle where clusters have complex shapes. Since, in this scenario, almost all the clusters are "blob-like", where EM works well, it found a clear separation between the clusters.

With regards to Spectral Clustering, which involves computing the eigenvectors. These eigenvectors are used to identify the clusters of the cells. It seemed to perform really well, since all the clusters formed are well separated either by linear or non-linear boundaries depending on similarity between two cells.

Leiden and Louvain clustering [5], which are also used for community detection, were able to cluster the items almost in a similar way, with a bit improvement in the case of the Leiden algorithm. This difference is because the Leiden only yields that the clusters are strongly connected, unlike Louvain, which may sometimes also be considered badly connected clusters (as we can see the brown color cluster formed at the center bottom of Leiden).

4.2.6 UMAP:. As discussed, UMAP generates the reduction even better in terms that it preserves the global structure along with the local, leading to generating two big groups of cells as can be seen in Figure 16.

As discussed in t-SNE, K-Means clustering divides the items in a linear fashion. Similarly here, it was able to generate well separated clusters separated by the linear boundaries. However, it may not be feasible at all times, since this is calculated using euclidean distance between cells and centroid. In the case of GMM, it again formed the "blob-like" clusters as it involves calculating the probability of each cell belonging to each of the distributions.

Just like t-SNE, spectral clustering was able to form well separated clusters depending on the eigenvectors. Also, both linear and non-linear boundaries can be seen, which itself is an advantage over k-means clustering.

Again, Leiden and Lovain separated the clusters almost in a similar fashion with the certain improvement in the case of Leiden, as it only generates the communities that are strongly connected (as we can see the brown color cluster formed at the center left of Leiden)

4.3 Multi-class Classification

The results from the classification models are tabulated in Tables 1 and 2. As we can see, both the RandomForest Classifier and SGD Classifier models had similar performances on each combination of reduction and clustering algorithms. An interesting observation is that Leiden and Louvain produced the most consistent results across all combinations for both classifier models, with Leiden performing slightly better for each since it is an enhancement of the Louvain

Table 1: Random Forest Classifiers

Classifiers	Cell Classification: Test Accuracies				
	K-Means	GMM	Spectral	Leiden	Louvain
PCA	88%	87%	97%	95%	94%
KPCA	89%	89%	88%	94%	94%
LLE-Standard	74%	87%	65%	94%	94%
LLE-Hessian	100%	99%	69%	94%	94%
LLE-Modified	96%	96%	62%	94%	94%
SE	77%	99%	64%	94%	94%
UMAP	92%	90%	95%	94%	94%
T-SNE	89%	93%	95%	95%	94%

Table 2: Stochastic Gradient Descent Classifier

Classifiers	Cell Classification: Test Accuracies				
	K-Means	GMM	Spectral	Leiden	Louvain
PCA	91%	88%	97%	96%	95%
KPCA	90%	91%	88%	95%	95%
LLE-Standard	74%	88%	62%	95%	95%
LLE-Hessian	100%	99%	55%	95%	95%
LLE-Modified	97%	97%	56%	95%	95%
SE	76%	100%	59%	95%	95%
UMAP	95%	90%	95%	96%	95%
T-SNE	89%	93%	96%	96%	95%

model. Additionally, both classifier models produced accuracies above 95% with MLLE using K-Means and GMM.

Apart from Leiden and Louvain, Spectral Embedding produced poor results for the remaining clustering algorithms, while presenting perfect results with GMM. As we can infer from Figure 14, although GMM created poor clusters from the Spectral Embedding reduction, the clear distinction between the only two clusters that were formed explains the performance of the classifier models in this aspect. Similarly, although the HLLE algorithm produced a poor reduction of the data, the clusters produced show some distinct separation with a lower number of clusters than defined, leading to the higher accuracies of the classifier models.

5 CONCLUSION

- This research successfully addressed the challenges in analyzing the scRNA-sequence datasets, primarily focusing on data sparsity and computational complexity.
- Developed a pipeline starting with dimensionality reduction on data, followed by clustering to get the cluster labels.
- Performed multi-class classification task to predict the cell labels, resulting in some near-to-perfect accuracy scores on the test set.
- This whole pipeline's outcome showcases the effectiveness of approach in handling high dimensional scRNA-sequence datasets for further insights.

6 STATEMENT OF CONTRIBUTION

All authors have contributed equally to this project in terms of both time and effort, however, various members took charge of different aspects of the project:

Atul: Implemented **Leiden**, **Louvain**, and **Hessian LLE**, and worked on the classification table for analyzing the accuracy of the clustering methods. Furthermore, worked on the **UMAP** and **TSNE** in the report, providing a detailed description of the algorithms and their results for each clustering algorithm.

Nawaf: Implemented all the different clustering techniques which the dimensionality reduced data feeds into. In addition, worked on the **Hessian** and **Modified LLE**'s description and their outcomes for each of their clustering techniques, as well as the classification description.

Prithvika: Implemented **PCA** and **KPCA** algorithms as well as assisted the team in writing the abstract, introduction, problem description, and the discussion of **PCA** and **KPCA**, as well as their results for each of the clustering technique.

Tasmia: Implemented **Isomap** and **Standard and Modified LLE** as well as focussed on writing a detailed observation on the **Standard LLE** as well as **Spectral Embedding**, as well as the outcome of each of their clustering techniques.

ACKNOWLEDGMENTS

We would like to express our gratitude to Dr. Luis Rueda for his guidance and support throughout this course. His knowledge and lectures through the duration of the semester were instrumental in shaping our understanding of the field and the various technologies implemented in this project.

REFERENCES

[1] 2008. Louvain Clustering. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/louvainclustering.html.
[2] Aurélien Géron. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly, Canada.
[3] Lukas Heumos and Anna Schaar. 2023. Dataset: Single-cell best practices. https://figshare.com/ndownloader/files/40016014.
[4] Lukas Heumos and Anna Schaar. 2023. *Single-cell best practices.* Theislab.
[5] V.A. Traag, L. Waltman, and N.J. van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities.