CS690: Computational Genomics

Mosaic single-cell data integration using deep generative models

Lecturer: Hamim Zafar
Group #2
Final Project
November 23, 2022

1 INTRODUCTION

Integration of different molecular tests is referred to as "mosaic data integration," and it has arisen as a crucial topic for the purpose of collecting and benefiting from vast amounts of single-cell information. At the moment, the bulk of methods for integrating mosaic data is limited to making use of the set of features that are shared by many modalities. Despite this, there is a rising requirement for the development of mosaic data integration algorithms as the number and complexity of single-cell datasets continue to grow. StabMap projects all cells into supervised or unsupervised reference coordinates using all available features regardless of overlap with other datasets, as opposed to relying on the overlap between datasets like UINMF and MultiMAP do. MultiMAP is a graph-based technique that assumes a uniform distribution of cells throughout a latent manifold structure fitted by an optimization approach. UINMF adds a latent metagene matrix to the factorization problem. The purpose of this study is to develop a deep generative model for mosaic data integration that has the potential to surpass methods that are currently being used. Methods such as StabMap distinguish between integration and batch correction in their calculations. It may be possible to achieve both of these objectives with a single plan by making use of a deep generative model (for example, an architecture based on VAE).

2 CURRENT SOTA

The current state-of-the-art model that performs "Mosiac data integration" is the Stabmap.
StabMap is a flexible approach that first infers a mosaic data topology and then projects all cells onto supervised or unsupervised reference coordinates by traversing the shortest paths along the topology
Performs well in various simulation contexts, facilitates **disjoint** mosaic data integration, and enables the use of novel spatial gene expression features for mapping dissociated single-cell data onto a spatial transcriptomic reference.
StabMap projects all cells onto supervised or unsupervised reference coordinates utilizing all available features regardless of overlap with other datasets, instead relying on traversal along the mosaic data topology
The input to StabMap is a set of single-cell data matrices and an optional set of discrete-cell labels. From this data structure, StabMap extracts the mosaic data topology (MDT)

"A network with nodes corresponding to each given dataset, and edges between nodes, weighted by the absolute number of shared features between the datasets"

3 DATASETS

The dataset for the algorithm had to be two datasets from the different tests performed on the same cell clusters. Looking for 2 unintegrated datasets, We found one at GEO accession number- GSM5160432 (parent GEO accession number- GSE168732). This is a typical patient's PBMC (peripheral blood mononuclear cells) sample. Since the two datasets are composed of the same cell type, their expression values should also indicate the same. They should be composed of similar cell clusters, and that is what we aim to achieve without integration, also utilizing the non-shared features. Another caveat is that we initially removed 5000 different genes from both datasets, which allowed us to validate our data filling later. This validation made us know if our methods were improving or not, and we could make corrections accordingly.

3.1 Dataset Details - I

PBMCs isolated and cryopreserved by AllCells. PBMCs are primary cells with relatively small amounts of RNA (1pg RNA/cell).

- 5,000 cells targeted
- 3,363 cells detected

- Sequenced on Illumina NovaSeq with approximately 61,935 reads per cell 28bp read1 (16bp Chromium barcode and 12bp UMI), 91bp read2 (transcript), and 8bp I7 sample barcode

3.2 Dataset Details - II

Peripheral blood mononuclear cells (PBMCs) from a healthy donor (the same cells were used to generate pbmc-1k-v2, pbmc-10k-v3). PBMCs are primary cells with relatively small amounts of RNA (1pg RNA/cell).

- 11,769 cells detected
- Sequenced on Illumina NovaSeq with approximately 54,000 reads per cell 28bp read1 (16bp Chromium barcode and 12bp UMI), 91bp read2 (transcript), and 8bp I7 sample barcode

4 OUR INITIAL APPROACH

Initially, we tried to use a VAE model to fill in the data missing from the datasets. It was composed of converting the high dimensional data composed of shared features into lower dimensional data. Then, we used this low-dimensional data to get back the high-dimensional missing data. This approach was dependent on the assumption that even the missing data from the datasets would yield the same low-dimensional data. This method didn't work well, and we obtained poor integration, thus we sought for newer approaches.

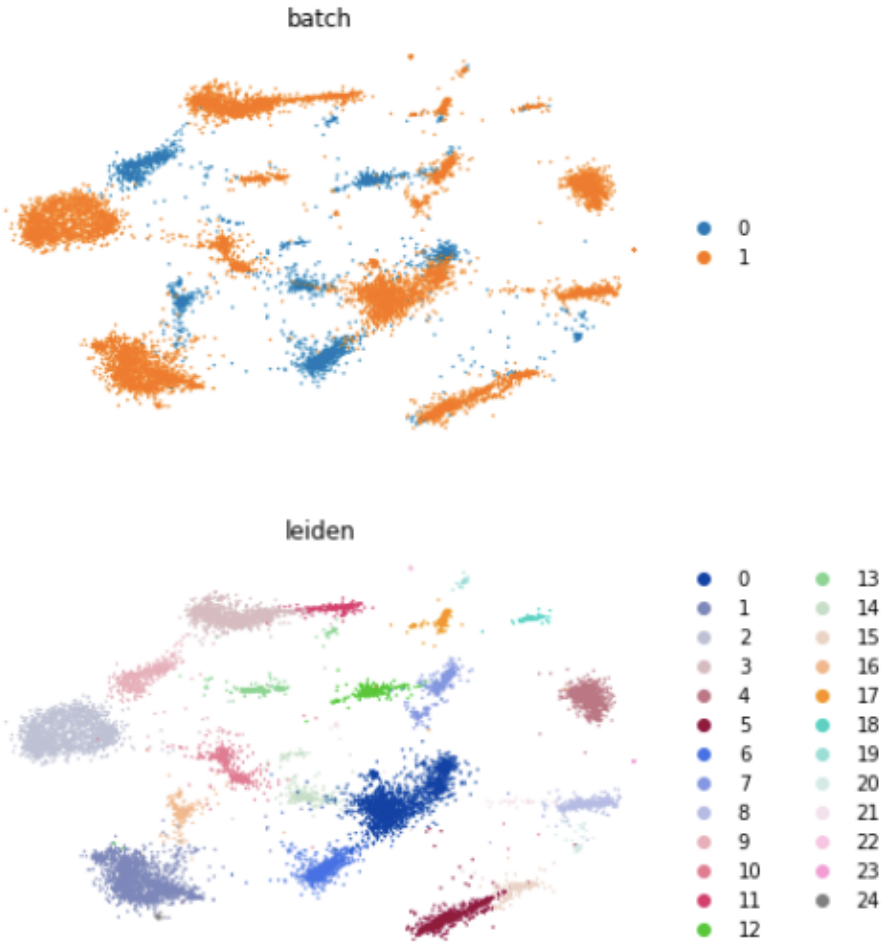


Fig. 1. Performance of our initial method

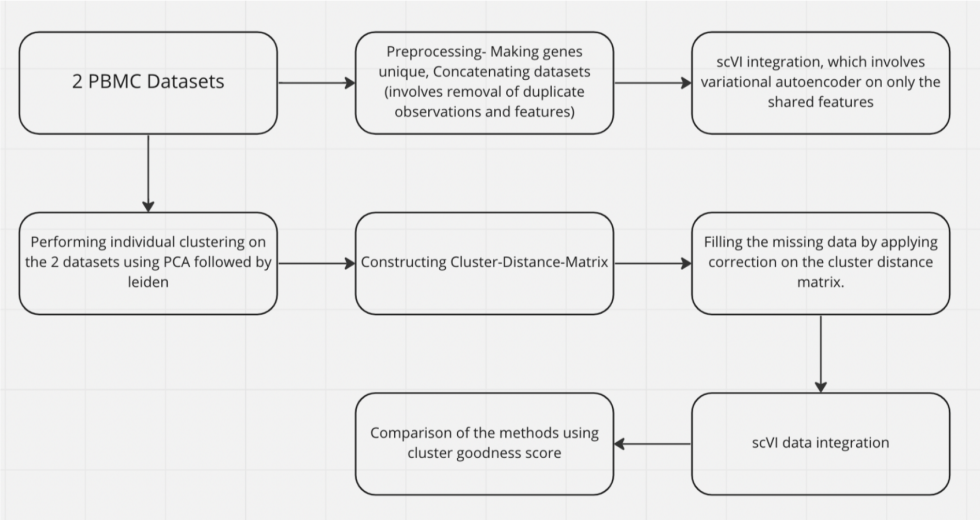


Fig. 2. Overall Method

5.1 Individual clustering of the two datasets

First, we individually cluster the two datasets. This is done using simple Principle Component Analysis, followed by Leiden clustering. Some tweaking of the **n pca** and **n neighbors** parameters was done to get good clusters. The purpose was to obtain as discrete clusters as possible because we will be finding the analog of a cluster from one dataset in the other dataset.

We had also tried another method of clustering, involving dimension reduction using VAE. It gave us more number of clusters, and the correspond heatmap for the cluster distance matrix is shown. However, there was no significant difference in the integration results, so we switched to using PCA only, as it took less time to run.

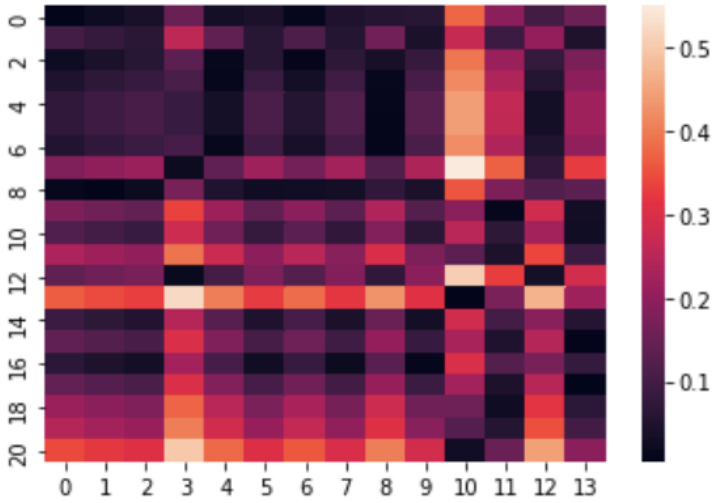


Fig. 3. Cluster Distance Matrix

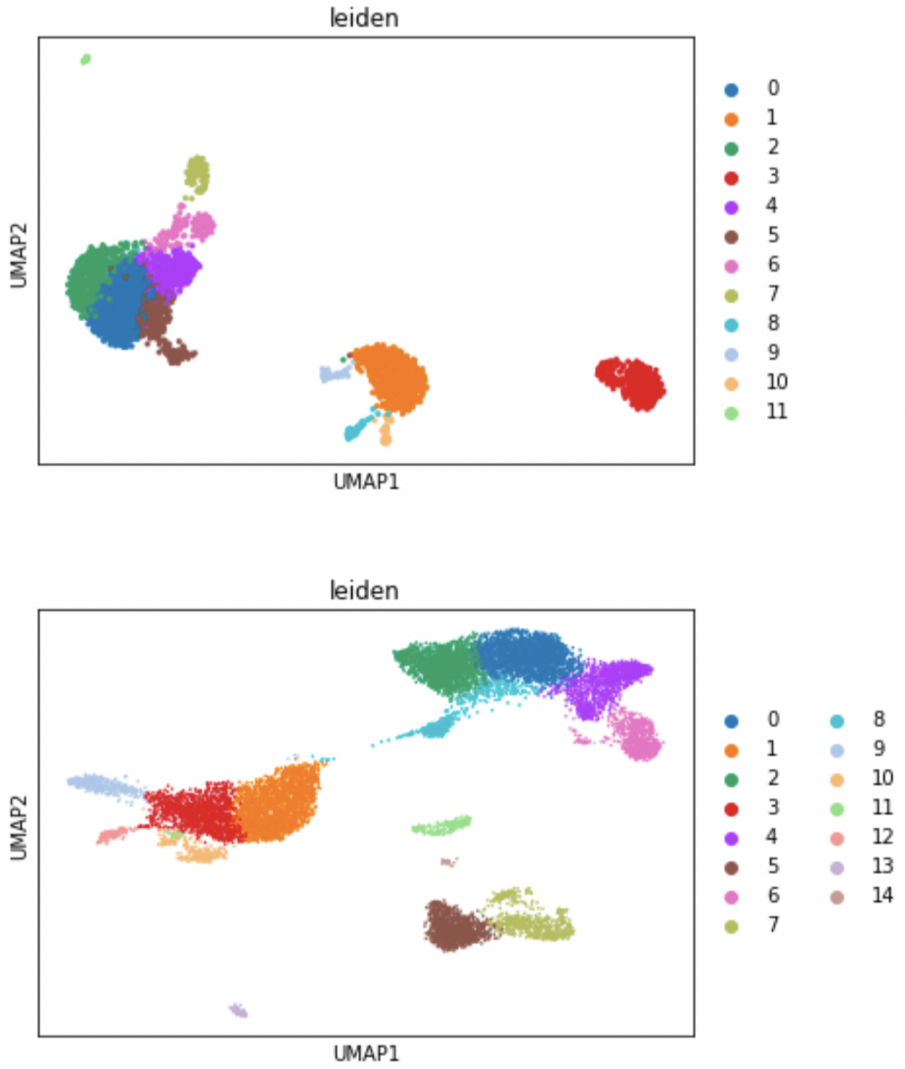


Fig. 4. Clusters obtained in the two datasets

5.2 Constructing cluster distance matrix

Then, the distance between any two clusters in the two datasets is computed. This approach for determining the distance between two clusters is prone to be altered and has scope for a lot of improvement.

To calculate the cluster distance between any two clusters, the average gene expression (read counts) is calculated for all the observations in each of them, and their difference is calculated. This is based on the assumption that the difference between the two datasets is simply due to the difference in the method to obtain it, and is a quantity independent of the gene.

Then, generate a clustered matrix in which each cell of the matrix contains the distance between any two clusters. The given heatmap shows the obtained cluster matrix. Ideally, it should have all values in any row/column high, and only one very low. This is seen in some cases.

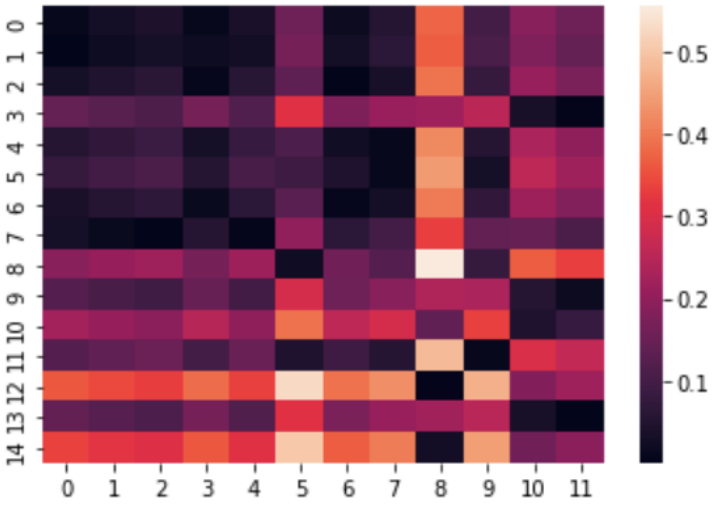


Fig. 5. Cluster Distance Matrix

5.3 Filling in the missing gene expression information

Now, using the cluster matrix, complete the missing data (the nonshared features). For this, we simply take all the observations from one dataset for a given cluster and find the cluster from the other dataset with the lowest distance from it.

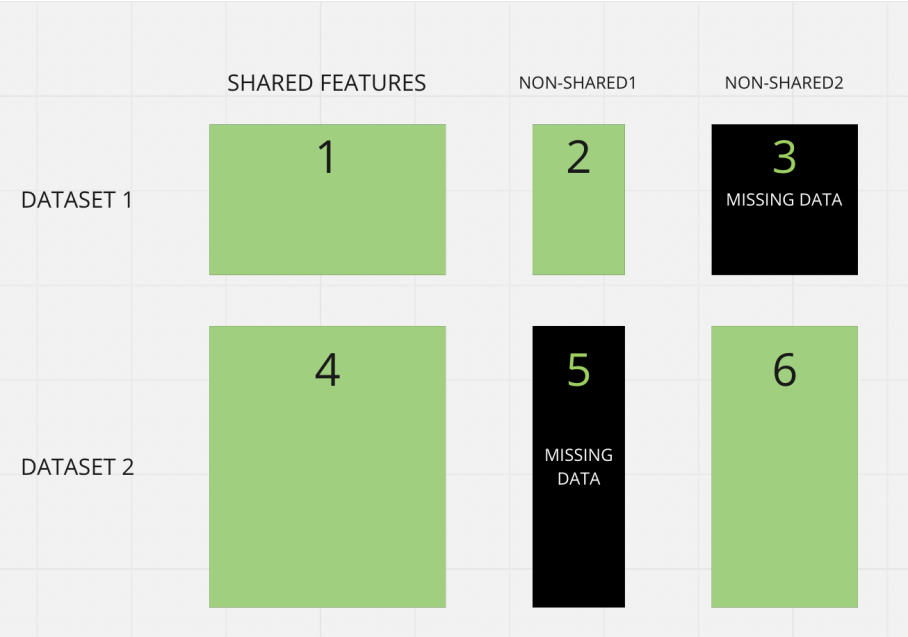


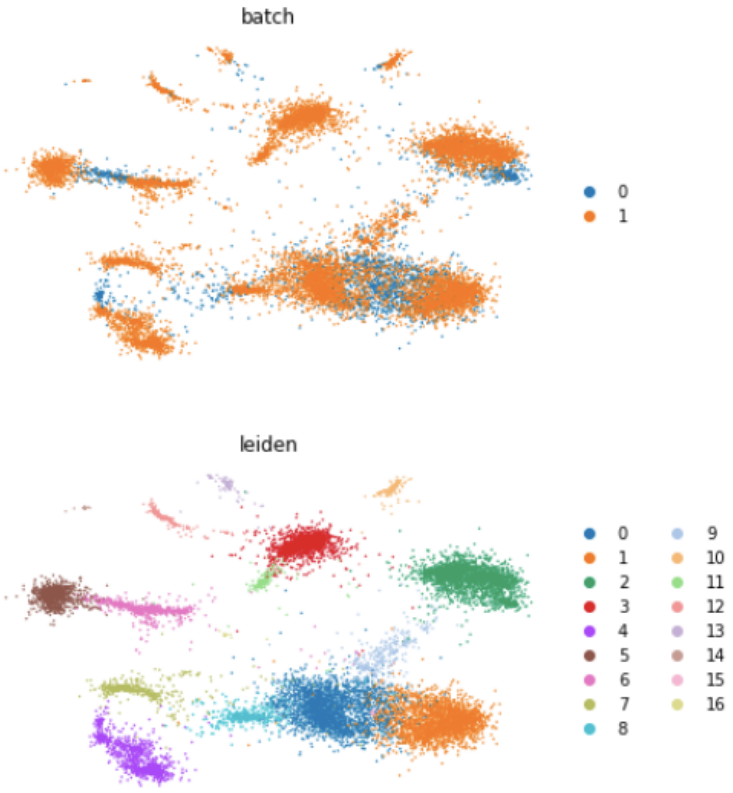
Fig. 6. Filling Missing Information

For example, in the above figure, suppose we want to fill section 5 of the whole data. For that, we will find a cluster from dataset 2, and then using the cluster distance matrix, find its analogous cluster in dataset 1. Then, we will see what distance they are apart according to the matrix, and we will simply correct the values from section 2 of the cells belonging to that analogous cluster by the difference amount from the matrix, and fill in the values in section 5 for the particular cluster. This we will do for every cluster in dataset 2, to fill section 5. The same process will be employed to fill the whole section 3 too.

Note that we have not yet normalized or taken a log of any data; we are just concerned with the read counts now, as it is better to apply error correction directly to read counts.
Then, using the filled data, we build a complete data object, normalize it, and log it.

5.4 Final SCVI integration with the whole data

Then, we use this filled dataset and submit it to a regular SCVI integration process. (we also notice how many non-shared characteristics were added in this stage after selecting highly variable genes). There were usually around 500 genes from the data filled by us in the filtered highly variable genes, out of a total of 2000 highly variable genes.
The clustering obtained as a result is shown.



5.5 Benchmarking- Cluster Goodness Score

Then, we attempt to calculate the success of the integration. This is accomplished by examining each Leiden cluster of the anndata and comparing the proportion of cells from the two datasets (ideally, this ratio should be close to the ratio of total cells in both datasets). This ratio is computed for each cluster, and its difference from the ideal ratio is determined and the overall mean for all cluster values is the cluster goodness score.

$$\text{Cluster Goodness Score} = \frac{1}{n}\sum_{i=1}^{n}(G - C_i)\frac{C_{cells}}{T_{cells}} \tag{1}$$

where C_{cells} is the number of cells in a given cluster and T_{cells} is the total number of cells combined across the dataset, G is defined as the ratio of cells in dataset 1 by dataset 2, and C_i is the number of cells in dataset i.

Finally, filled data was fed to scVI tools integrator, which first filters out the highly variable genes, we kept track of how many of our filled genes were used by it. On running multiple iterations, the number of genes filled by us used after the highly variable gene filtering was around 500 out of a total of 2000.

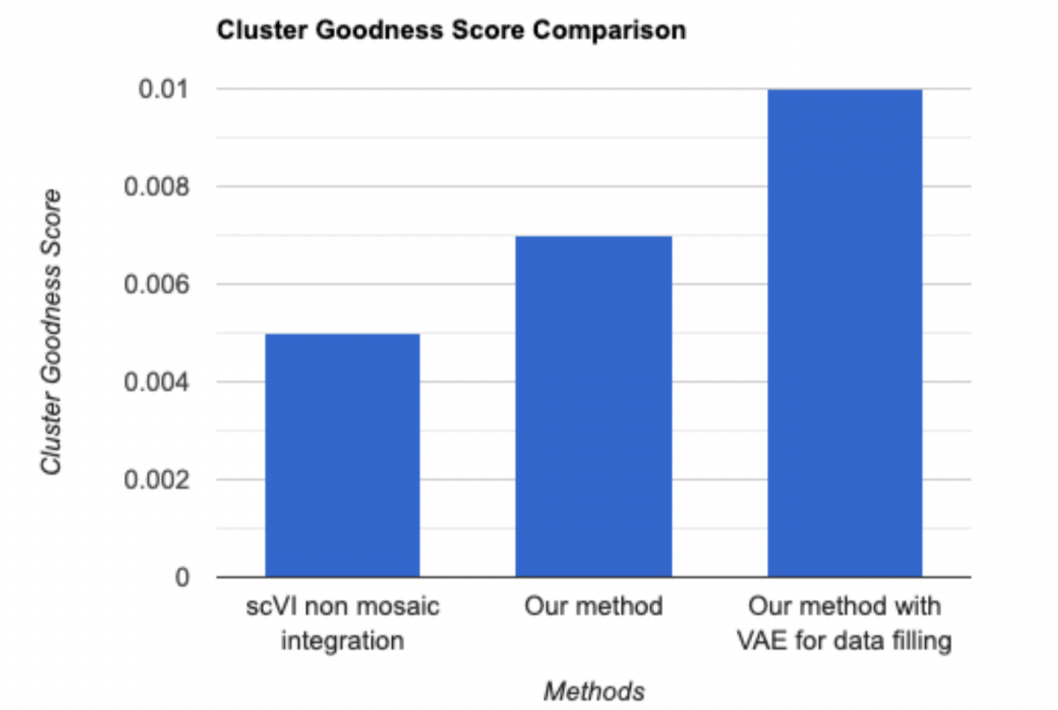


Fig. 7. Comparison of the methods

The clusters goodness average for all clusters in SCVI integration was **0.0046** and in our case was **0.0070**. Our first approach, using VAE for data filling performed worse, with a score of 0.0095. We also tried tweaking the number of highly variable genes selected from the dataset (changed from 2000 to 5000), but that gave worse results. SCVI now permits non-normalized/logged data, thus a raw backup was created prior to normalization/logging, highly variable genes were identified, and the readings were returned to non-normalized/logged. This result was obtained: cluster goodness score - **0.0078**. Next, SCVI was used to cluster individual datasets, yielding the *clust_dist_mat* heatmap shown below.
In addition, we compared the filled data with the data that was removed from the original datasets. Cluster Goodness **0.094**.

7 FURTHER IMPROVEMENTS

When we first attempted to fill in the missing data, we used a totally deep generative model; however, this model did not produce very excellent results and also required a very significant amount of time to execute (around 3 hours). After that, we attempted to complete the data by utilizing a different approach. To complete the task of filling in the missing data, we relied on a straightforward procedure that involved cluster mapping and read count correction. This approach not only outperformed the one that came before it but also required less time to complete (around 30 minutes). There is a significant amount of room for development in the approach. To begin, the clustering that was done on each separate dataset wasn't very good. Ideally, there would be no overlap between the many groups that were created. This can be made better by employing more effective clustering strategies. We attempted to accomplish the same thing by utilizing Variational Autoencoder, however it did not result in any gains. Second, the method for estimating the distance between clusters consisted mostly of determining the average read count of a single cluster. This is a fairly basic approach that can be made more sophisticated by additional development. In the final

step of filling in the data, it was necessary to deduct the minimum cluster distance from each and every observation that belonged to that cluster. It's possible that this won't be the optimal solution for all of the observations in that cluster; instead, each one will need to be rectified separately so that the process can be refined in the future.

8 CONCLUSIONS

Mosaic data integration is a method that doesn't only have applications in absent gene features. It also is useful in cases like integrating datasets of different species and using non-orthologous genes in the analysis. Also, Assay for Transposase Accessible Chromatin (ATAC) data, which has the information of accessible chromosome regions of the DNA, can be integrated using Mosaic data integration. Thus, tackling with this problem is of utmost importance in the field of genomics.

9 CONTRIBUTION

- Shashank (190794) - Primary VAE approach, Cluster Goodness metric, Cluster Distance Approach and Report
- Antreev (190163) - Algorithm Development, Benchmarking results and Report
- Gurbaaz (190349) - Developed codebase for experiments, Literature Review and Presentation

10 REFERENCES

(1) Ghazanfar, Shila, Carolina Guibentif, and John C. Marioni. "StabMap: Mosaic single cell data integration using non-overlapping features." bioRxiv (2022).
(2) Kriebel, April R., and Joshua D. Welch. "UINMF performs mosaic integration of singlecell multi-omic datasets using nonnegative matrix factorization." Nature communications 13.1 (2022): 1-17.
(3) Jain, Mika Sarkin, et al. "MultiMAP: dimensionality reduction and integration of multimodal data." Genome biology 22.1 (2021): 1-26.
(4) Lopez, Romain, et al. "Deep generative modeling for single-cell transcriptomics." Nature methods 15.12 (2018): 1053-1058.
(5) Ternes, Luke, et al. "A multi-encoder variational autoencoder controls multiple transformational features in single-cell image analysis." Communications biology 5.1 (2022): 1-10.
(6) Dataset 1 link: https://www.10xgenomics.com/resources/datasets/peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-chromium-connect-channel-5-3-1-standard-3-1-0
(7) Dataset 2 link: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3?