Comparative Analysis of Machine Learning Techniques to classify Cell Types in PBMCs from scRNA-Seq Data

Introduction

Transcriptomics has greatly enhanced our understanding of cellular functions and the mechanisms by which alterations in gene activity lead to diseases. Single-cell RNA sequencing (scRNA-seq) offers a detailed perspective by capturing gene expression at the single-cell level, identifying rare cell types, delineating cell lineages, and observing gene expression dynamics during development, differentiation, or disease progression (Asada et al., 2021). Single-cell data is growing exponentially due to improvements in sequencing technology and cost reductions and traditional classification techniques are no longer sufficient to analyze scRNA-seq data. Therefore machine learning techniques have become instrumental in analyzing scRNA-seq datasets. These methods facilitate the classification of cells based on their origin or type, leveraging unsupervised clustering techniques such as k-means and hierarchical clustering to group cells by gene expression similarities before cell type assignment using known marker genes (Abdelaal et al. 2019).

Peripheral blood mononuclear cells (PBMCs) are a diverse group of circulating immune cells, widely utilized in immunological and clinical research to model the human immune response in vitro. Comprising primarily lymphocytes (T, B, and NK cells), monocytes, and infrequently, dendritic cells, PBMCs present a snapshot of the immune system's complexity. In addition to cellular identification via surface markers, specific genes serve as molecular signatures that uniquely define each cell type within PBMCs. Mature B cells, identifiable as CD19+ within PBMCs, can be further subclassified into various subsets including memory B cells differentiated by CD27+ and CD21+ expression markers, while T cells are commonly marked by the expression of CD3, CD8. Such markers are especially useful in technologies like single-cell RNA sequencing (scRNA-seq), which allows for a detailed examination of cellular heterogeneity at the gene expression level (Abdelaal et al., 2019).

Identifying the distinct cell composition present in a given sample is an important aspect of analyzing the single cell data. This study aimed to cluster and annotate PBMCs using pre-labeled datasets from the 68k PBMC dataset, to discern the constituent immune cell types. Two machine learning models- Random Forest and Support Vector Machines (SVM) were employed to classify these cell types. The data required to train the model is subjected to a series of quality control, feature filtering and standardization to make it fit the training requirements. Through this project we not only aim to assess the performance of these computational models but also explore their capacity to refine our understanding of cell type characteristics based on gene expression profiles.

Methodology

The cell expression data stored in .h5ad format, was accessed and loaded as training (pbmc) and test (unknown_pbmc) using the *sc.read_h5ad* function. In the initial preprocessing steps, mitochondrial (MT) genes were identified by selecting genes with names that started with 'MT-', while ribosomal genes were identified with names starting with ('RPS', 'RPL'), respectively.

Quality Control:
To access the quality of the dataset, the *sc.pp.calculate_qc_metrics* function was applied to calculate quality control metrics for both datasets, focusing on the percentage of reads mapping to MT and ribosomal gene content.Cells with a high proportion of mitochondrial reads are often stressed or dying, leading to non-representative transcriptomes. Similarly, cells with high reads mapping to ribosomal genes often consist of lower complexity transcriptomes, resulting in a lower variety of genes. To ensure good data quality, cells with over 50% ribosomal and over 5% MT content were identified and removed from the training dataset using boolean indexing.

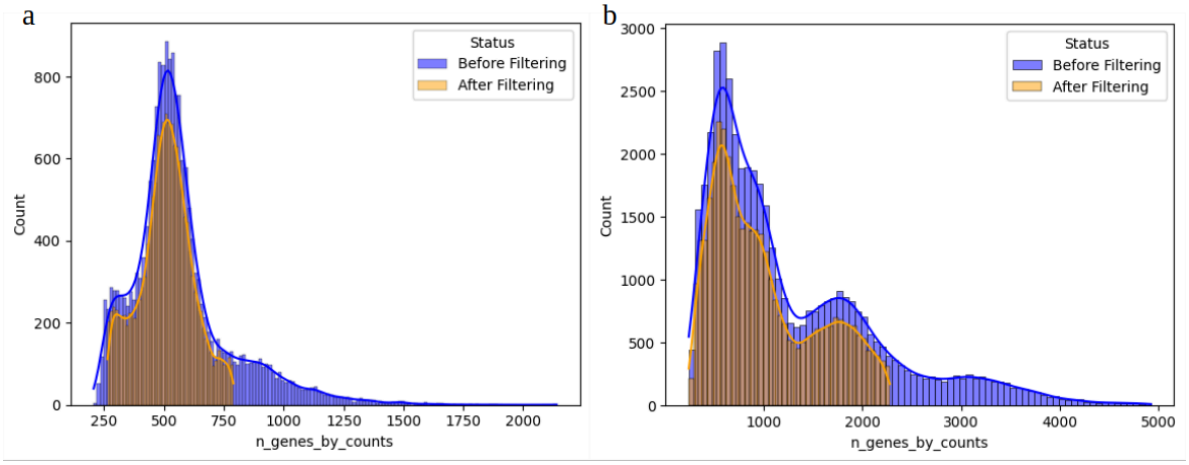

Figure 1: Distribution of gene counts/cell before and after outlier removed from the training (a) and the testing (b) dataset. In (a-b) we illustrate the effect of excluding outliers (orange) compared to the original distribution (blue)

Cells having a high gene count suggest over amplified libraries during capture are termed Outliers often representing noise rather than meaningful biological information. In this study, outlier cells were removed using the median gene count and median absolute deviation (MAD). Cells with gene counts outside three times the MAD from the median were flagged as outliers and discarded. A total of 2834 and 5649 cells were excluded from the training and testing dataset, respectively. Following this, genes with low expression were filtered out using *sc.pp.filter_genes(new_pbmc, min_cells = 3),* reducing noise in the dataset (*Fig 1*). The filtered datasets were normalized using total counts per cell (*sc.pp.normalize_total*) and then log-transformed for stabilization of variances (*sc.pp.log1p*).

Feature Selection :
Highly variable genes were selected using *sc.pp.highly_variable_genes(new_pbmc, min_mean=0.0125, max_mean=3, min_disp=0.5)*, to ensure only the most informative genes will be included in the downstream analysis. The selected genes were then scaled to ensure all features were on the common scale. Dimensionality reduction was performed using principal Component Analysis (PCA), followed by visualization with uniform manifold approximation and projection (UMAP) *sc.tl.umap(new_pbmc) (Fig 2)*.

Machine Learning Training and Testing
The analysis then proceeded with machine learning classification using Random Forest and Support Vector Machine (SVM) models. The extracted features from the UMAP-reduced data (*X = new_pbmc.obsm["X_umap"]*) and the corresponding cell type labels were prepared for both of the machine learning classification modelsOn the unknown dataset, the cell_type were renamed

to match the existing cell types from the training dataset. The features of cells from unknown data were extracted from the UMAP-reduced data and corresponding cell type labels as the training data.

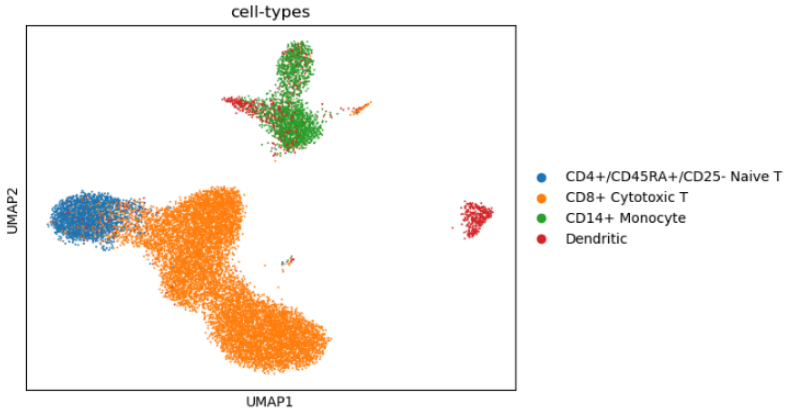

Figure 2: UMAP visualization of cell clusters based on highly variable gene expression profiles. The UMAP shows distinct clusters representing different cell types: Naive T cells (blue), CD8+ Cytotoxic T cells (orange), CD14+ Monocytes (green), and Dendritic cells (red). Distinct T lymphocytes clusters are separated from the monocytes and dendritic cells. Naive T cells are clearly clustered apart from cytotoxic T cells with minimal overlap, while monocytes cluster shows partial overlap with dendritic cell clusters.

RandomForestClassifier was then trained, where its hyperparameter optimization was performed using *GridSearchCV* across various key parameters, such as number of estimators, maximum depth of the trees, minimum samples per split and leaf, and the maximum number of features. After performing 5-fold cross-validation, the best hyperparameters were identified as 200 estimators, depth of 10, minimum 4 features per leaf, *sqrt* for maximum features, and a minimum split size of 10. The final *RandomForestClassifier* was retrained under these parameters and evaluated with the testing dataset. Similarly, SVM classifiers were also trained, where given the UMAP-reduced dimensionality of the dataset, a linear kernel is considered as one of the hyperparameters. This kernel supports the linear separation of data points in the feature space, which is effective for many biological classification tasks. Random state was fixed to ensure reproducibility of results

After training the both classifiers, the model was used to predict the cell type for the unknown dataset. To evaluate the model's performance, a confusion matrix was generated using *confusion_matrix(unknown_y, y_pred)*, with the true and predicted cell types plotted as a heatmap. Additionally, the accuracy of the prediction and the F1 score was calculated with the accuracy_score function and f1_score function with macro average to address for class imbalance.

Results:

The performance of the Random Forest classifier on the unknown dataset was evaluated, and the overall accuracy was 0.3728 indicating that the model did not generalize effectively to this dataset. The ARI (Adjusted Rand Index) and NMI scores for the Random Forest classifier on the unknown dataset were 0.62, suggesting moderate agreement between the predicted clusters and

the true labels, but also highlighting the model's limitations in accurately distinguishing cell types.

One key observation was the misclassification of CD14+ Monocytes, where a substantial number of these cells were classified as CD8+ Cytotoxic T cells and Dendritic cells highlighting a crucial limitation in the model's ability to capture the unique gene expression profiles (Fig 3a). Similarly, there was significant overlap between CD4+ Naive T cells and CD8+ Cytotoxic T cells, suggesting the model's inability in distinguishing between the two closely related lymphocytes. Dendritic cells were also poorly classified, as the model often confused them with both CD14+ Monocytes, further indicating the classifier's difficulties in distinguishing between certain populations. While the classifier still separates some cell types well, it shows signs of weak predictions, specifically with dendritic cells and monocytes.

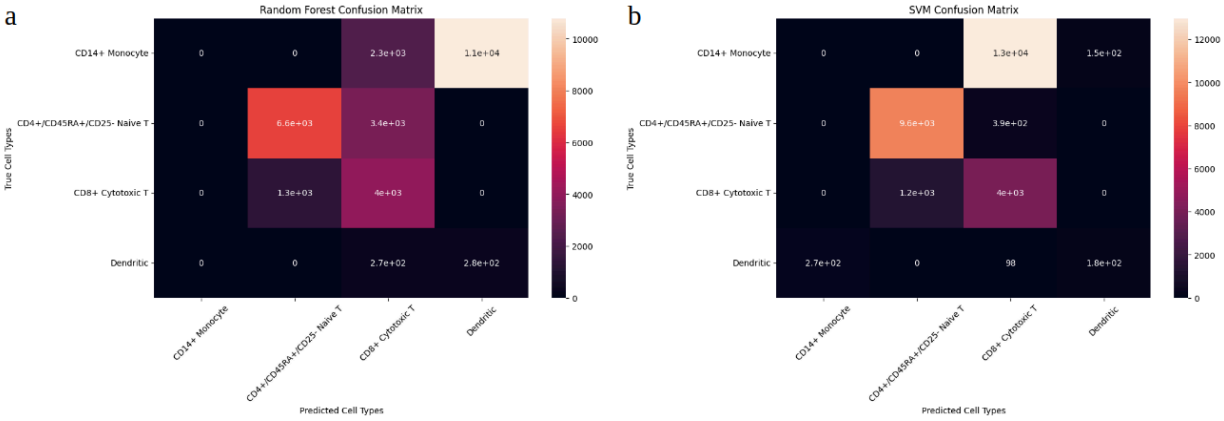

Figure 3: Random Forest Confusion Matrix for Cell Type Classification. (a) shows the confusion matrix generated from the prediction of unknown dataset with Random Forest Classifiers, and (b) shows the confusion matrix generated from the prediction of unknown dataset with the SVM Classifier. The color intensity of each cell reflects the number of predictions, with lighter colors indicating misclassification rate.

Similarly, the SVM classifier performed below expectations on the unknown dataset, although having an higher overall accuracy of 0.4782, compared to the lower accuracy predicted by the Random Forest Classifier. The SVM model exhibited similar misclassifications, particularly with the misclassification of CD14+ Monocytes with CD8+ Cytotoxic T cells and Dendritic cells. The ARI and NMI scores for the SVM model on the unknown dataset were 0.60, reflecting comparable performance to the Random Forest model.

The performance of the Random Forest and SVM classifiers was also quantified using the F1 score, a metric that balances the precision and recall of the classifier. The Random Forest F1 achieved a score of 0.1974 and SVM F1 was at 0.2773. The observed F1 scores for both classifiers were relatively low, indicating challenges in model performance. Such outcomes may stem from several factors inherent to the analysis of single-cell RNA sequencing (scRNA-seq) data like High Dimensionality, Imbalanced Classes, Overfitting or Underfitting

Discussion

After conducting comprehensive data quality control, we used a subset of the assigned 68k PBMC dataset to train two machine learning models, Random Forest(RF) and Support Vector

Machine (SVM), respectively, to classify the four distinct cell types. Cells with abnormal genes count and abnormal mitochondrial/ribosomal gene counts were filtered to ensure data quality for training and testing. The highly variable genes were then selected to emphasize the intracellular differences. These filtered data and features were chosen as they would be be able to effectively distinguish the various cell types when used to train the model

The accuracy of the prediction was used as the primary metrics to assess the quality of our machine learning model, as the models are trained to predict the cell types based on the gene expression level. By training the RF and SVM models on a labeled dataset and testing on unseen data, we aimed to evaluate the robustness of these models while exploring the importance of various genetic markers, and advance the automation of cell type classification, crucial for scaling up scRNA-seq applications in biomedical research. These models work well to provide a valuable framework for validating such labels and potentially uncover deeper insights into cell type differentiation (Wang et al., 2020).

In our trained model, the SVM classifier was able to achieve a higher accuracy (47.82%) compared to that of Random Forest Classifier (37.28%) on the same unknown dataset. Initially, our prediction was that random forest would outperform SVM, as it is better suited at capturing complex nonlinear relationships between features. On the other hand, SVM with a linear kernel assumes that the classes are linearly differentiable, but the dataset used for training may not be.

The reason for the discrepancy between the experimental findings and the predictions is likely due to overfitting during training. This is evidenced by the strong performance on the training dataset but poor generalization to the testing dataset. SVM, known for its robustness in high-dimensional spaces, likely avoided overfitting, especially when the feature size is large relative to its sample size (Hu et al., 2016).

Another key factor for affecting the classification performance could be the complexity of the dataset at hand. As the complexity or the number of cell populations increases, the performance of the classifier generally decreases. The performance of all classifiers is relatively low on the Zheng 68K dataset (Abdelaal et al. 2019), where the high pairwise correlations between the mean expression profiles of different cell populations contributes to the lower model performance.

In addition, the suboptimal performance of both classifiers might also stem from the relatively small data size from the filtered data, which may not have been sufficient to train the model effectively. Although SVM performs better relative to Random Forest, its predictive ability was still far from ideal.

A more rigorous data screening and feature selection is a feasible way to solve the current possible RF overfitting problem. However, in the dataset used in this report, only 2834 valid cell data were obtained after screening. If more stringent data screening is performed, the accuracy of the model may be further reduced due to insufficient training samples. Therefore the ideal improvement would be to use a larger scale of data for more stringent data screening to train the model. Increasing the amount of quality training data can also improve the performance of SVM.

Disclosure:
This project and report were written as a collaborative effort of our team members. Hiu On was primarily responsible for code development while Peter and Justy provided review and advice on the code. Justy contributed to writing the Introduction section, while Peter authored the Methodology section, and Han was responsible for compiling and writing the discussion section. Hiu On confirmed and revised the technical details of the report, and Han and Justy provided comments on the improvement of the report. Justy and Hiu On made the final improvements to the article. Each team member's work was informed by the suggestions of the other team members.

Code Availability:

The code for this project is available in :
(https://github.com/OHM314159/Binf_7000_assignment-3.git).

Reference List

Abdelaal, T. *et al.* (2019) 'A comparison of automatic cell identification methods for single-cell RNA sequencing data', *Genome Biology*, 20(1). doi:10.1186/s13059-019-1795-z.

Asada, K. *et al.* (2021) 'Single-cell analysis using machine learning techniques and its application to medical research', *Biomedicines*, 9(11), p. 1513. doi:10.3390/biomedicines9111513.

Hu, Y. *et al.* (2016) 'A machine learning approach for the identification of key markers involved in brain development from single-cell transcriptomic data', *BMC Genomics*, 17(S13). doi:10.1186/s12864-016-3317-7.

Wang, H. *et al.* (2020) 'Pathway-based single-cell RNA-seq classification, clustering, and construction of gene-gene interactions networks using random forests', *IEEE Journal of Biomedical and Health Informatics*, 24(6), pp. 1814–1822. doi:10.1109/jbhi.2019.2944865.