Active Learning for Cell Type Annotation

Amanda Qi
acqi@andrew.cmu.edu

Peneeta Wojcik
pawojcik@andrew.cmu.edu

Zhaoyi You
zhaoyiy@andrew.cmu.edu

Abstract

Automatic annotation of cell types from single cell RNA sequencing (scRNA-seq) data is highly nontrivial due to complex relationships in gene expression data. In this work, we train a deep neural network classifier using active learning. We implement four batch selection methods and evaluate their performance on a highly complex dataset. Our results demonstrate that a simple two hidden layer neural network trained within is comparable to established tools such as scmap and Seurat. Despite implementing active learning, we observe that model performance converges across sampling strategies after a few iterations, and that batch size has minimal long-term impact on classification accuracy. The PBMC dataset contains cells that are difficult to distinguish, and future work would entail hyperparameter tuning and using more complex data preprocessing methods.

1 Introduction

Tissues are comprised of heterogeneous cell populations with distinct gene expression patterns. The advent of single cell sequencing methods has made it possible to discover nuanced changes in gene expression within cell types as a result of treatments or disease states. A common task in single cell analysis is cell type annotation, where unlabeled cells are annotated as a particular cell type given a reference dataset from the same tissue type. The most common approach is to identify key marker genes for each cell type in the reference data and utilize these to annotate new samples as cell types, although there exist many other methods including deep learning and supervised classification methods [1].

In cases where data is costly to annotate, it is inefficient to utilize the full data set or label samples at random. Active learning involves intelligently selecting the most informative samples from the data to train a base learner with some criterion, maximizing its performance while minimizing number of samples used for training. One sampling method within active learning is pool-based sampling, where samples are selected from an unlabeled pool, annotated, and incorporated into the training data to update the model. Samples can be selected one at a time or in a batch.

Batch learning is preferable in practice since updating a model with a batch of data is less computationally expensive than updating it with a single sample at each iteration. Batches should be diverse enough to capture different portions of the underlying sample distribution. Batch selection strategies tend to balance exploration and exploitation. Exploration involves selecting samples in underexplored regions in the data, whereas exploitation involves selecting samples that the model is most uncertain in predicting. An ideal batch should strike a balance between the two, ensuring that the sampled points capture the underlying distribution of the data.

Kimmel et al. utilize a semi-supervised adversarial approach to train a two-layer multilayer perceptron (MLP) for cell type annotation, which they name scNym [2]. scNym trained using the adversarial approach achieved extremely high classification accuracy of above 0.9 for all datasets. The purpose of this work is to train a deep learning classifier similar to scNym using active learning approaches. We assess the performance of different pool-based batch selection strategies on model performance for a supervised cell type annotation task.

2 Methods

2.1 Dataset and Preprocessing

The data used for this project is a scRNA-seq dataset of peripheral blood mononuclear cells (PBMC) obtained from the Broad Institute Single Cell Portal [3]. This dataset was chosen because it is highly complex and contains nine total cell types, which provides a robust set of classes for prediction (Fig. 1). The expression data in the portal had been log-normalized and batch corrected, therefore no additional normalization or correction was performed. The Scanpy single cell analysis python library was used to filter cells and genes with low read count [4]. Mitochondrial, ribosomal, and hemoglobin genes were removed from the dataset, then genes that were present in less than 10 cells and cells with less than 500 genes were filtered out. The final dimensionality of the data is 31,021 samples by 33,694 genes. The top 2000 highly variable genes were selected as features for downstream analysis.

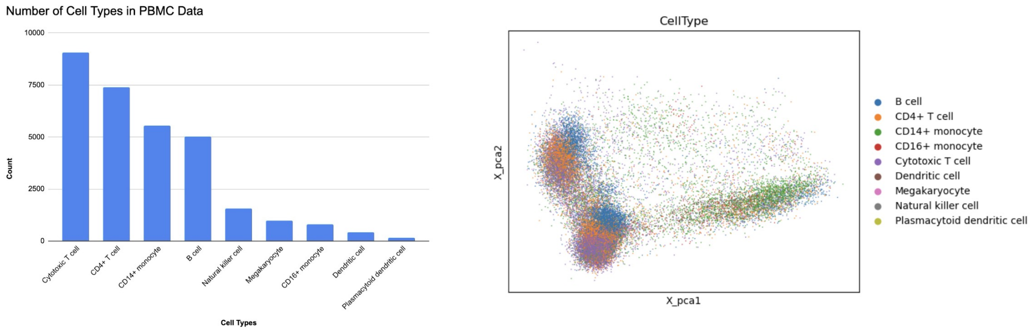

Figure 1: Frequency of cell types in PBMC dataset and PCA plot of the data

2.2 Base Learner

2.2.1 Neural Network Model

We selected highly variant 2000 genes expressions as the input for neural network. After mapping the input to the two low-dimension hidden layers, the output is the probability of each cell type of the input sample. For neural network model, linear transformation is used for mapping between the input and hidden layer, the two hidden layers, and the hidden layer and output as shown in Eq. (1) (3) (5). After each linear transformation, ReLU is used as activation function for the layer output except for the last layer as shown in Eq. (2) (4). Finally, SoftMax function is applied to the output vector to calculate the probability of each of the 9 cell types as shown in Eq. (6). CrossEntropy is chosen as the loss function for optimization as shown in Eq. (7). Our training used mini-batch SGD algorithm.

$$h_i = \gamma_{i,j} + \sum_{j=1}^{2000} \gamma_{i,j} \cdot x_j \tag{1}$$

$$c_i = ReLU(h_i) \tag{2}$$

$$a_j = \alpha_{j,i} + \sum_{i=1}^{256} \alpha_{j,i} \cdot c_i \tag{3}$$

$$z_j = ReLU(a_j) \tag{4}$$

$$b_k = \beta_{k,b} + \sum_{j=1}^{256} \beta_{k,j} \cdot z_j \tag{5}$$

$$\hat{y}_k = \frac{\exp(b_k)}{\sum\limits_{l=1}^{9} \exp(b_l)}, \ \forall k \in \{1, \ldots, 9\} \tag{6}$$

$$l(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{i=1}^{C} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) \tag{7}$$

2.2.2 Hyperparameters

A 2-hidden-layer fully connected neural network is designed as base learner for our cell annotation task. The parameters are chosen as the table shows according to a previous research [2]. Besides 2-hidden-layer network, we also design a 3-hidden-layer network with layer normalization to avoid vanishing gradient.

Table 1: Hyperparameters of Neural Network

Names	Values
First Hidden Layer Units	256
Second Hidden Layer Units	256
Third Hidden Layer Units	64
Learning Rate	0.001
Dropout Rate	0.3

2.3 Batch Selection Methods

We implemented four batch selection strategies: uncertainty sampling, diversity sampling, weighted k-means clustering (WKMC), and query-by-committee (QBC), with random sampling as a baseline. To evaluate the impact of batch size on active learning performance, we tested three batch sizes: 9, 27, and 81. These sizes correspond to sampling approximately 1, 3, or 9 examples per cell type given an ideal proportional batch, as our dataset includes 9 distinct cell types. This setup allows us to assess how the number of queried samples per round influences model performance and the biological diversity captured during the active learning process.

2.3.1 Uncertainty Sampling

For uncertainty sampling, a single forward pass of the model was used to obtain the predicted class probabilities for each sample in the unlabeled pool. Entropy was calculated for each of these values, and the points with the highest entropy values were greedily sampled as a batch.

2.3.2 Query by Committee

Monte Carlo (MC) dropout was used to simulate a committee of models. Dropout layers were introduced between each layer and uncertainty was quantified by activating these dropout layers during inference. Ten predictions were generated for each sample in the pool, each with a random dropout mask. The entropy of all predictions was calculated as an estimate of disagreement between committee models for a particular sample. The top most uncertain points were greedily selected from the unlabeled pool.

2.3.3 Diversity Sampling

K++ initialization was utilized in diversity sampling to ensure that cluster centroids are spread across the data. In this method, the squared distance of each point to its respective centroid is calculated. A probability distribution is created proportional to these distances and new centroids are sampled from this distribution. The probability that point i is selected as the next centroid is given by Eq. 8, where D_i^2 represents the squared distance of point i to its centroid and $\sum_j D_j^2$ is the sum of squared distances of all points to their corresponding centroids.

$$P_i = \frac{D_i^2}{\sum_j D_j^2} \tag{8}$$

This method improves clustering performance and expedites convergence. The number of clusters was set as the current batch size, and points were sampled randomly from each of the clusters.

2.3.4 Weighted K-Means Clustering

For the WKMC method, a set of the top most uncertain points from the unlabeled pool was selected. This sampled subset was the size of the current batch multiplied by 5 in our implementation. The diversity sampling method described in the previous section was then used to cluster points with K++ initialized K-means and select points randomly from each cluster.

3 Results

3.1 Comparison of Different Active Learning Strategies

Different network design has impacts on which active learning method performs the best. In our 2-hidden-layer neural network setting (batch size = 81), WKMC performs the best in the first few rounds and uncertainty sampling is the best method in the latter rounds (Fig. 2). While in 3-hidden-layer neural network (batch size = 81), the best performance is achieved in WKMC and QBC (Fig. 3). The 3-hidden-layer neural network has similar results with or without layer normalization (Fig. 4).

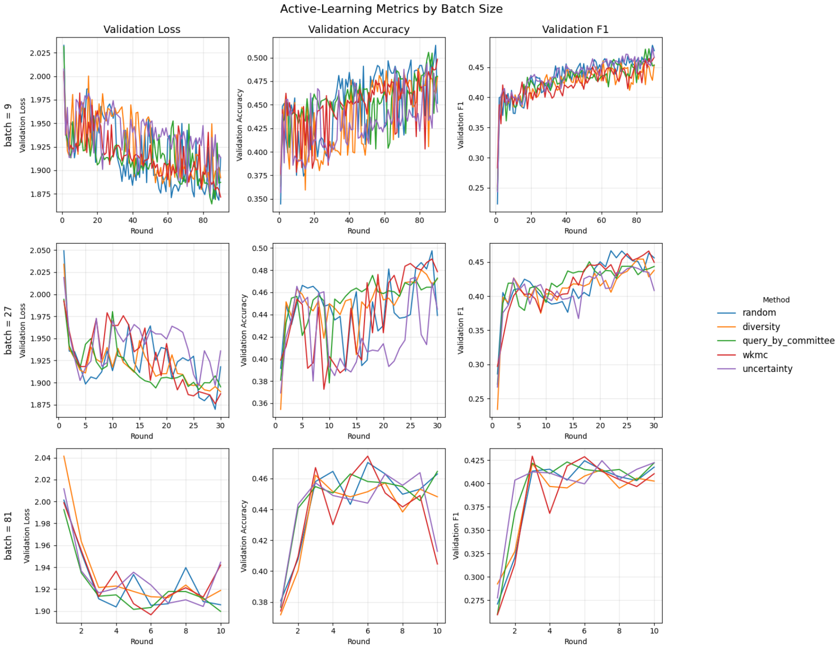

Figure 2: Accuracy, F1 Score, and Loss for 4 Different Sampling Strategies based on a 2-Hidden-Layer Neural Network

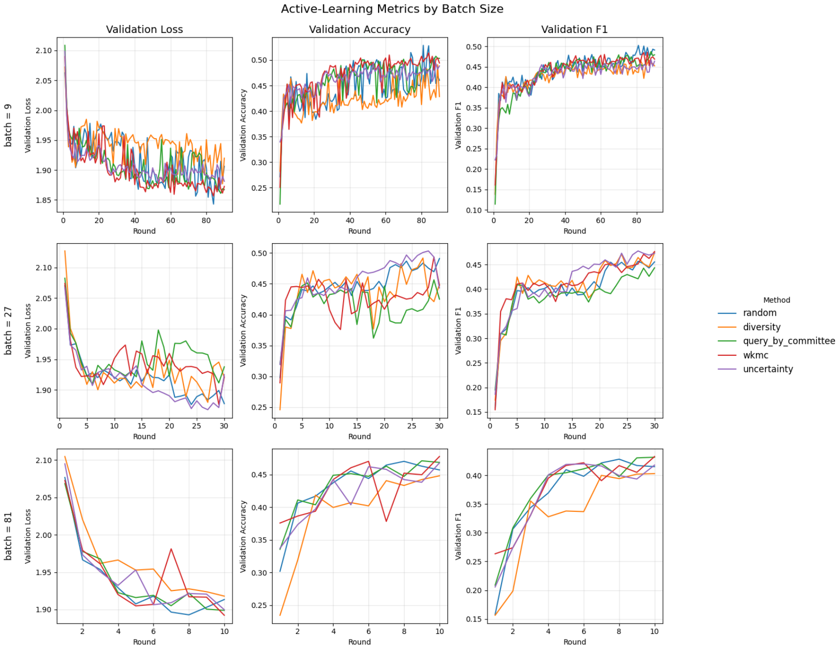

Figure 3: Accuracy, F1 Score, and Loss for 4 Different Sampling Strategies based on a 3-Hidden-Layer Neural Network

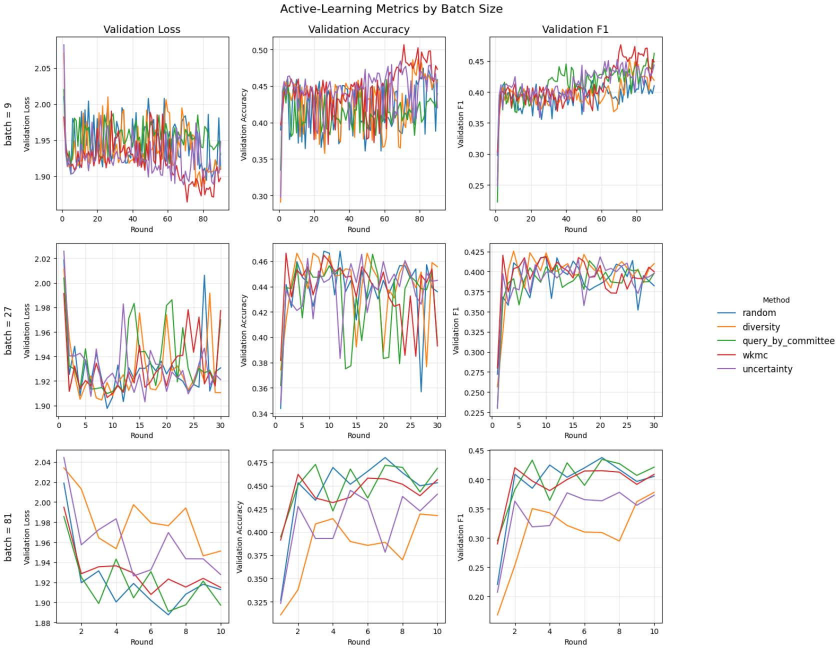

Figure 4: Accuracy, F1 Score, and Loss for 4 Different Sampling Strategies based on a 3-Hidden-Layer Neural Network with Layer Normalization

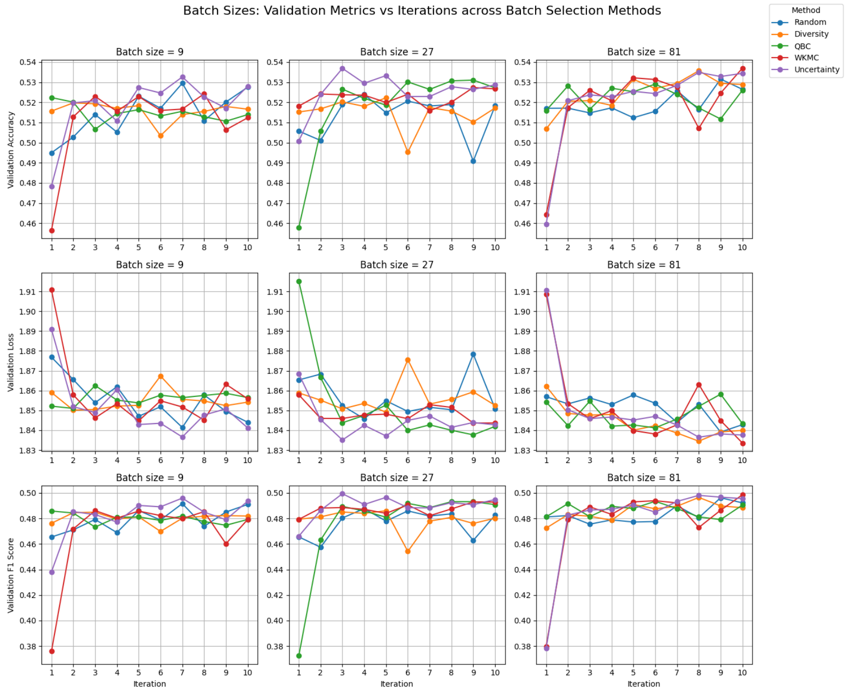

Figure 5: Comparison of batch selection strategies across different batch sizes and active learning iterations. Validation accuracy (top row), loss (middle row), and F1 score (bottom row) are shown for batch sizes of 9 (left), 27 (center), and 81 (right) over 10 acquisition iterations. The five selection methods evaluated are random sampling, diversity sampling, QBC, WKMC, and uncertainty sampling.

Figure 5 compares the performance of five batch selection strategies (random, diversity, QBC, WKMC, and uncertainty sampling) across three batch sizes (9, 27, and 81) over 10 active learning iterations. Validation metrics include accuracy, loss, and F1 score. Across all batch sizes and metrics, model performance across the five sampling methods appears largely similar, with no method showing consistently superior performance. Notably, after approximately two iterations, the models converge to comparable levels of accuracy and F1 score, and the validation loss stabilizes. This convergence suggests that the choice of batch selection method has limited impact on model performance in this setting, especially beyond the initial rounds of sampling. Moreover, larger batch sizes naturally lead to faster sample accumulation across iterations, but do not result in markedly better validation performance, indicating diminishing returns with increasing batch size. Overall, the results imply that simpler selection strategies such as random sampling may be sufficient in this context, as more complex methods do not offer substantial gains in performance.

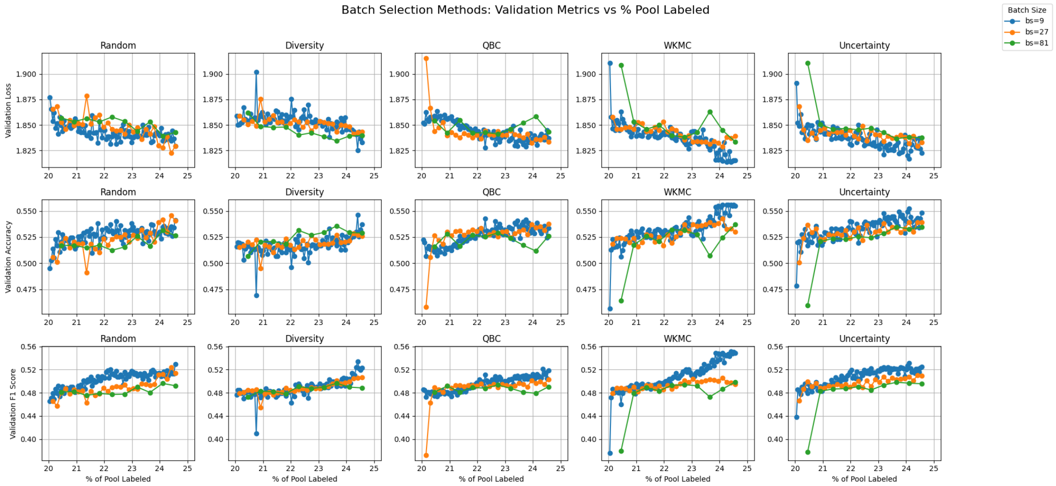

Figure 6: Validation loss (top), accuracy (middle), and F1 score (bottom) for five batch selection methods across batch sizes of 9, 27, and 81, plotted against the percentage of the pool labeled. All runs began with 20% of the pool labeled and proceeded until a common labeled percentage was reached.

Figure 6 presents a comparison of validation performance across batch sizes (9, 27, and 81) for five sampling strategies: random, diversity, QBC, WKMC, and uncertainty sampling. All experiments began with 20% of the pool labeled and continued until each batch size reached the same overall labeling percentage. Performance is reported across three validation metrics (loss, accuracy, and F1 score) as a function of the percentage of the pool labeled. Across all sampling strategies, model performance trends were largely consistent regardless of batch size. While larger batches consume the pool more quickly, performance metrics generally converge once approximately 23% of the pool is labeled. Notably, smaller batch sizes (particularly batch size 9) showed a slight but consistent advantage in F1 score compared to larger batches, especially for WKMC. However, these differences were modest and did not significantly affect overall model quality. These results suggest that when the total number of labeled samples is fixed, batch size has minimal long-term impact on performance. Instead, it is the proportion of labeled data that primarily determines model improvement, reinforcing the idea that smaller batches or more complex sampling strategies provide limited benefit after a sufficient amount of data has been acquired.

3.2 Batch Analysis

To evaluate the diversity of each sampled batch, the proportion of each cell type in each batch was calculated (Fig. 7). All batches showed identical proportions despite differing batch sizes and selection methods. These proportions matched the cell type proportions in the original data (Fig. 1). The distribution of sampled points for each batch selection method showed slight variation across batch sizes (Fig. 8). Random sampling selected points evenly throughout the distribution, which was expected. All selection methods sampled primarily from the leftmost two clusters, however diversity sampling selected a larger number of points from the rightmost cluster.

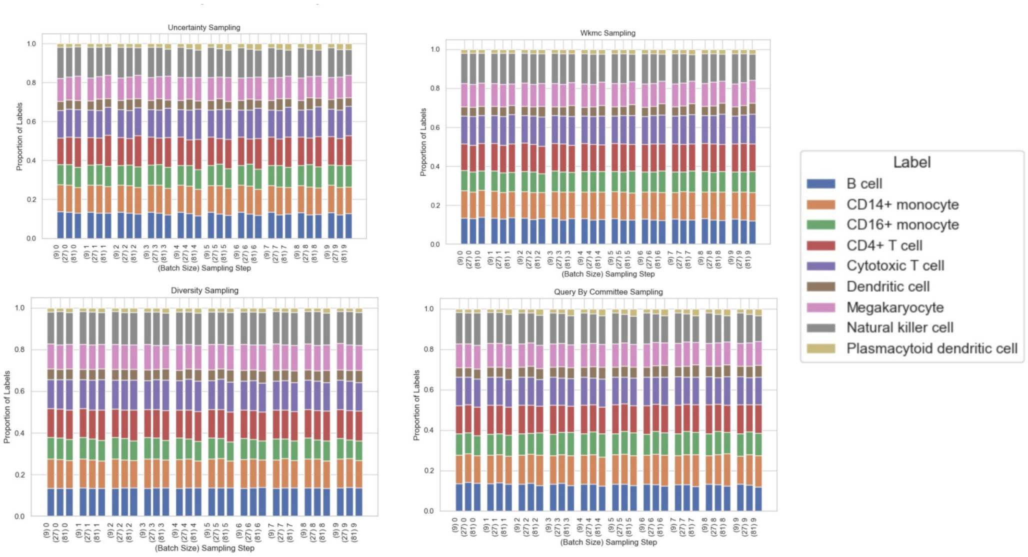

Figure 7: Proportion of cell types per batch for uncertainty sampling, WKMC, Diversity, and QBC.

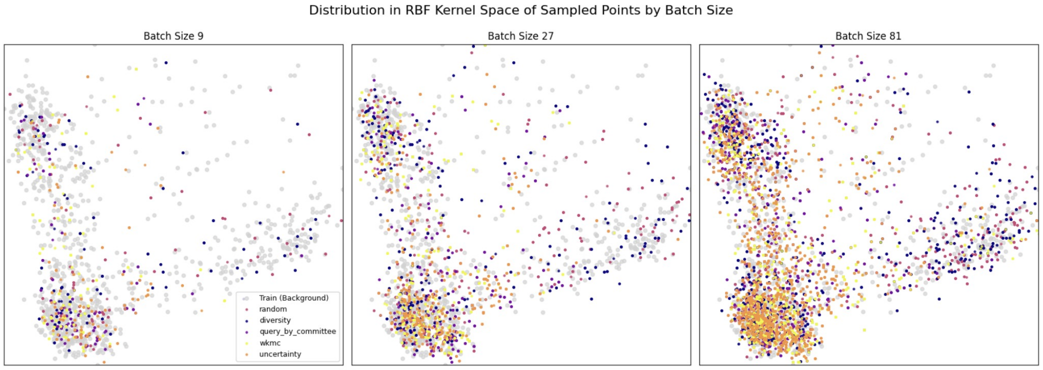

Figure 8: PCA plot of distribution of sampled points for each batch size

4 Discussion

Our results demonstrate that a simple two-hidden-layer neural network can achieve cell type annotation performance on the PBMC dataset that is comparable to established tools, such as scmap [5], Seurat [6] and CHETAH [7]. Our results also found an additional hidden layer will not help to enhance the model's performance in PBMC dataset. These findings indicate that neural networks alone have intrinsic limitations for cell type annotation and may need to be complemented by data preprocessing based on prior knowledge. Researchers find that hierarchical clustering before the neural network classification would enhance the model performance in PBMC dataset [8].

In terms of batch analysis, the sampled batches showed minimal variation in cell type proportions. This is likely be due to the high level of noise in the data and substantial overlap between the different cell types. Most of the points sampled for all selection methods in regions where B cell and T cell clusters are located, and there is a high level of overlap between the different subtypes in these regions (Fig 1, 8). The base learner may not be complex enough to capture nuanced differences between the different subtypes within this dataset due to the presence of closely-related variants such as cytotoxic and CD4+ T cells and two types of monocytes (CD14+ and CD16+). One potential approach may be to group the subtypes together and evaluate the model's ability to distinguish between broader cell subtypes. This would improve model performance by reducing classification complexity, but at the cost of decreased specificity. Additional layers or incorporation of additional multi-modal features such as gene ontology annotations for genes may improve subtype prediction.

These findings are consistent with our batch-level analysis, which showed that after approximately two batch selection iterations, model performance converged across all selection strategies regardless of batch size. Furthermore, once a certain proportion of the pool was labeled, increasing the batch size had minimal additional effect on performance. Several factors may explain why active learning strategies did not consistently outperform random sampling in this context. First, the high dimensionality of the gene expression data ($\sim 2{,}000$ features) likely dilutes informative signals and introduces noise, which can reduce the effectiveness of sample selection. Second, the uncertainty estimates used to guide sampling–based on only 10 Monte Carlo dropout passes–may be too weak to reliably capture model uncertainty, leading to suboptimal query selections. Third, the two-layer MLP architecture with dropout may be insufficiently expressive to model subtle distinctions between overlapping or closely related cell types, limiting the benefits of targeted acquisition. Together, these limitations may reduce the potential gains of active learning and suggest that improvements in model capacity, uncertainty quantification, or feature dimensionality reduction may be necessary to realize more substantial benefits.

References

[1] Giovanni Pasquini, Jesus Eduardo Rojo Arias, Patrick Schäfer, and Volker Busskamp. Automated methods for cell type annotation on scrna-seq data. *Computational and Structural Biotechnology Journal*, 19:961–969, 2021.

[2] Jacob C Kimmel and David R Kelley. Semisupervised adversarial neural networks for single-cell classification. *Genome Res.*, 31(10):1781–1793, October 2021.

[3] Jiarui Ding, Xian Adiconis, Sean K. Simmons, Monika S. Kowalczyk, Cynthia C. Hession, Nemanja D. Marjanovic, Travis K. Hughes, Marc H. Wadsworth, Tyler Burks, Lan T. Nguyen, John Y. H. Kwon, Boaz Barak, William Ge, Amanda J. Kedaigle, Shaina Carroll, Shuqiang Li, Nir Hacohen, Orit Rozenblatt-Rosen, Alex K. Shalek, Alexandra-Chloé Villani, Aviv Regev, and Joshua Z. Levin. Systematic comparative analysis of single cell rna-sequencing methods. *bioRxiv*, 2019.

[4] Fabian A. Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:15, 2018.

[5] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods*, 15(5):359–362, May 2018.

[6] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, 33(5):495–502, May 2015.

[7] Jurrian K de Kanter, Philip Lijnzaad, Tito Candelli, Thanasis Margaritis, and Frank C P Holstege. CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing. *Nucleic Acids Res.*, 47(16):e95, September 2019.

[8] Ziyi Li and Hao Feng. A neural network-based method for exhaustive cell label assignment using single cell RNA-seq data. *Sci. Rep.*, 12(1):910, January 2022.