Coreset-based logistic regression for atlas-scale cell type annotation

Danrong Li[1], Young Kun Ko[*1], and Stefan Canzar[*2]

[1]Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, United States
[2]Faculty of Informatics and Data Science, University of Regensburg, Regensburg, Germany

Abstract

Logistic regression provides a powerful tool for the predictive analysis of single cell and spatial omics data. It has been shown to outperform more complex models on fundamental tasks such as the selection of marker genes and the annotation of cell types. Training logistic regression models on increasingly large, publicly available single-cell datasets opens up opportunities to build accurate and robust cell type prediction models but poses major computational challenges. Here, we adapt and extend the theory of coresets for logistic regression. In particular, we compute a previously introduced classification complexity measure using linear programming to identify single-cell datasets which admit coresets, i.e. random samples of cells, that preserve logistic loss. We show that applying this complexity measure requires a prior dimensionality reduction by, e.g., principle component analysis, and prove that logistic loss is preserved under this linear transformation. In experiments on several cell atlases, we compared our method to CellTypist, a logistic regression-based immune cell atlas and scMulan, a pre-trained language model, to demonstrate that theoretical guarantees translate to an accurate cell type label transfer method that scales to atlas-scale datasets.

1 Introduction

A routine step in the computational analysis of single-cell omics data is the annotation of cell types. Machine learning and deep learning models trained on publicly available single-cell data sets are increasingly being used for this purpose. In particular, foundation models such as scFoundation [1] and scGPT [2] which are pre-trained on vast amounts of unlabeled single-cell RNA-seq data and then fine-tuned on a smaller, task-specific dataset have demonstrated enormous success in the predictive analysis of single cell data.

*Corresponding authors

However, experiments have shown that classic machine learning methods such as logistic regression have an advantage over large-scale pre-trained language models when it comes to relatively simple tasks such as cell type annotation [3]. In fact CellTypist [4], an immune cell atlas of myeloid and lymphoid lineages in human tissue was developed based on a logistic regression model trained on more than 700,000 cells from 19 different studies covering 20 different tissue types. For convergence and storage reasons, the training data was divided into many small batches on which training by stochastic gradient descent was carried out one at a time. While this strategy achieved a substantial learning speed-up over traditional logistic regression, it does not provide any guarantees on the quality of the solution.

In this work we propose an alternative approach based on the theory of coresets to train a logistic regression model on very large datasets efficiently and with theoretically guaranteed performance. In particular, we sketch a very large input matrix, that is, we randomly sample a small subset of cells (rows of the matrix), a so-called coreset [5, 6], so that the optimization of the much smaller problem instance yields a solution that is close to the optimal solution of the full problem.

This method however does not work for all large datasets. Some datasets are known to be resilient to sampling [7]. This is captured by a previously introduced classification complexity measure μ which identifies single-cell datasets which admit coresets that preserve logistic loss and regression coefficients. In particular, if a dataset has a small μ value, it admits a small coreset depending on how small μ is [5, 7]. Note that this is a sufficient condition for coresets. There are datasets with bad μ value that empirically admit a good coreset, alas without theoretical guarantee. We implement a linear program (LP) [8] to compute classification complexity for various single-cell datasets. We show that applying this complexity measure requires a prior dimensionality reduction using, e.g., principal component analysis. We prove that logistic loss is preserved under this linear transformation.

In experiments on several cell atlases from different species and tissues, we compared our method to CellTypist and scMulan [9], a pre-trained language model, to demonstrate that theoretical guarantees translate to an accurate cell type label transfer method that scales to atlas-scale datasets.

2 Methods

Preliminaries and Problem Setting Define a data matrix $H \in \mathbb{R}^{n \times d}$ and a label vector $y \in \{\pm 1\}^n$. The logistic regression problem can be constructed as an optimization problem: Minimizing over $\beta \in \mathbb{R}^d$, we want to reduce the logistic loss

$$\mathcal{L}(\beta; H) = \sum_{i=1}^{n} f(\langle h_i, \beta \rangle \cdot y_i) = \sum_{i=1}^{n} \ln(1 + \exp(-\langle h_i, \beta \rangle \cdot y_i)) + \frac{\lambda}{2} \|\beta\|_2^2 \tag{1}$$

where $\forall i \in [n]$, h_i represents the i-th row of H and $\lambda > 0$ is the ℓ_2 regularization parameter. Define the optimal logistic regression coefficient as $\beta_{opt} = \arg\min_{\beta \in \mathbb{R}^d} \mathcal{L}(\beta; H)$. Finding such β_{opt} is costly, especially when both n and d can be really large. For biological data,

in order to mitigate this issue, popular methods like CellTypist [4] use stochastic gradient descent (SGD) with randomly selected mini-batches to reduce the computation time per iteration. In this paper, we discuss the option of reducing the dimension of H via uniform sampling the rows to form a coreset $\tilde{H} \in \mathbb{R}^{m \times d}$ where $m \leq n$. By reducing the sample size from n to m, we lessen the computational effort. As will be shown in our Theorem 1, the logistic loss is preserved for all β, and this suggests that $\beta_{\tilde{H}}$ is a good approximation for β_{opt}, where $\beta_{\tilde{H}} = \arg\min_{\beta \in \mathbb{R}^d} \mathcal{L}(\beta; \tilde{H})$. In addition, for matrix $H \in \mathbb{R}^{n \times d}$, we define its rank k approximation as $H' \in \mathbb{R}^{n \times d}$ with $\beta_{H'} = \arg\min_{\beta \in \mathbb{R}^d} \mathcal{L}(\beta; H')$.

Theorem 1 (Main)**.** *Define $H \in \mathbb{R}^{n \times d}$ and its rank k approximation $H' \in \mathbb{R}^{n \times k}$. For small classification complexity measure $\mu(H')$ (see [8] and Supplementary Section 1), we can sample $m = \tilde{O}(\frac{k \cdot \mu(H')^2}{\epsilon^2})$ rows from H to form its coreset $\tilde{H} \in \mathbb{R}^{m \times d}$. After running logistic regression on $\tilde{H}$ to get $\beta_{\tilde{H}}$, we can use this parameter to predict the labels for H. The logistic loss $\mathcal{L}(\beta_{\tilde{H}}; H)$ will not deviate too much from $\mathcal{L}(\beta_{\tilde{H}}; \tilde{H})$, which is a minimized value. More formally, with high probability,*

$$\left| \mathcal{L}(\beta_{\tilde{H}}; \tilde{H}) - \mathcal{L}(\beta_{\tilde{H}}; H) \right| \leq 2 \cdot \sqrt{n} \cdot \sigma_{k+1}(H) \cdot \|\beta_{\tilde{H}}\|_2 + \epsilon \cdot \mathcal{L}(\beta_{\tilde{H}}; H'), \ \epsilon \in [0, 1] \tag{2}$$

Intuitively, μ captures the maximum imbalance between correctly classified vs. incorrectly classified points, that is how separable the data set is. And it is known that there are data sets with high μ value that do not admit a small coreset [7]. Furthermore, there exists a simple linear program to exactly compute μ [8]. We remark that the original LP formulation in [8] contained what we believe to be a typo. We correct the error in Supplementary Section 1.

Proof Outline for Theorem 1 The goal is to bound the absolute logistic loss between original matrix H and its coreset H' over the same β using triangle inequality. Our proof consists of the following steps:

- In Section 2.1, we show the logistic loss is preserved for rank k approximation H' within some bounded error term.
- In Supplementary Section 1 and Section 2.2, we introduce complexity measure $\mu(H)$ and state that if H is μ-complex (H has a good μ value), then we can construct a coreset matrix $\tilde{H}$ where the logistic loss is preserved within some bounded error term.
- The absolute difference of the logistic loss between $\tilde{H}$ and $\tilde{H}'$ is also bounded analogously as in Section 2.1. In Section 2.3, we combine these error terms using triangle inequality. The summation of aforementioned three error terms bounds the absolute difference of the logistic loss resulting in the right-hand-side of (2).

We assume that the single cell dataset H has finite samples n and $\beta_{H'}$ has bounded ℓ_2 norm due to the regularization term. In addition, we assume the $(k+1)th$ largest singular value for H is very small. This assumption is valid since we normalized and log transformed the dataset during the pre-processing step.

2.1 Rank k Approximation Logistic Loss Preservation

The following theorem shows that a low-rank approximation of the original data preserves the logistic loss with the same regression coefficients.

Theorem 2 ([8, Theorem 3])**.** *If $X, X' \in \mathbb{R}^{n\times d}$, for all $\beta \in \mathbb{R}^d$,*

$$|\mathcal{L}(\beta; X) - \mathcal{L}(\beta; X')| \leq \sqrt{n}\|X - X'\|_2\|\beta\|_2 \tag{3}$$

The right-hand-side of (3) can be bounded by the $(k+1)th$ largest singular value of the original matrix with the following standard theorem:

Theorem 3 ([10, Eckart Young Mirsky Theorem])**.** *Suppose matrix A' is the rank k approximation for matrix $A \in \mathbb{R}^{n\times d}$. Then we have*

$$\|A - A'\|_2 = \sigma_{k+1}(A) \tag{4}$$

where $\sigma_{k+1}(A)$ represents the $(k+1)\,th$ largest singular value for matrix A.

With Theorem 2 and Theorem 3, we can show that under the condition that H has a good rank k approximation H' (i.e. $\sigma_{k+1}(H)$ is small), implying that the logistic loss between H' and H differs by a small value.

Theorem 4 (Our Logistic Loss Preservation Theorem after Rank k Approximation)**.** *Assume the same notations as described previously. Recall $H' \in \mathbb{R}^{n\times d}$ represents the rank k approximation for H. Then for all $\beta \in \mathbb{R}^d$,*

$$|\mathcal{L}(\beta; H) - \mathcal{L}(\beta; H')| \leq \sqrt{n} \cdot \sigma_{k+1}(H) \cdot \|\beta\|_2 \tag{5}$$

Proof. From Theorem 3, we get

$$\|H - H'\|_2 = \sigma_{k+1}(H) \tag{6}$$

while Theorem 2 gives

$$|\mathcal{L}(\beta; H) - \mathcal{L}(\beta; H')| \leq \sqrt{n}\|H - H'\|_2\|\beta\|_2 \tag{7}$$

By combining (6) and (7), we conclude the proof. □

2.2 Coreset Logistic Loss Preservation

We will now state Theorem 5 which argues that the coreset $\tilde{H}$ generated under the Lewis weight sampling (see [11]) preserves the logistic loss.

Theorem 5 ([5] Corollary 9 restated)**.** *Assume the same annotations as described before. Using the ℓ_1 Lewis weight, suppose we sample $m = \tilde{O}(\frac{k\cdot\mu(H)^2}{\epsilon^2})$ rows from matrix $H \in \mathbb{R}^{n\times k}$ to construct coreset $\tilde{H} \in \mathbb{R}^{m\times k}$, where $\epsilon \in [0, 1]$ and $\mu(H)$ is the complexity measure (defined in Supplementary Section 1). Then with high probability, for all $\beta \in \mathbb{R}^k$, we have:*

$$\left|\mathcal{L}(\beta; \tilde{H}) - \mathcal{L}(\beta; H)\right| \leq \epsilon \cdot \mathcal{L}(\beta; H) \tag{8}$$

Theorem 5 suggests that after creating the coreset $\tilde{H}$, we can calculate its logistic regression coefficient $\beta_{\tilde{H}}$. Then with high probability:

$$\left|\mathcal{L}(\beta_{\tilde{H}}; \tilde{H}) - \mathcal{L}(\beta_{\tilde{H}}; H)\right| \leq \epsilon \cdot \mathcal{L}(\beta_{\tilde{H}}; H), \quad \epsilon \in [0, 1] \tag{9}$$

2.3 Combine Rank k Approximation and Coreset

Proof for Theorem 1. Recall that Theorem 4 and Theorem 5 both have the guarantee on all $\beta \in \mathbb{R}^d$. By substituting $\beta_{\tilde{H}}$ into β from (5) in Theorem 4, we have

$$|\mathcal{L}(\beta_{\tilde{H}}; H) - \mathcal{L}(\beta_{\tilde{H}}; H')| \leq \sqrt{n} \cdot \sigma_{k+1}(H) \cdot \|\beta_{\tilde{H}}\|_2 \tag{10}$$

By substituting $\beta_{\tilde{H}}$ into β from (8) and using the rank k approximation matrix H', with high probability, we have

$$\left|\mathcal{L}(\beta_{\tilde{H}}; \tilde{H}') - \mathcal{L}(\beta_{\tilde{H}}; H')\right| \leq \epsilon \cdot \mathcal{L}(\beta_{\tilde{H}}; H'), \quad \epsilon \in [0, 1] \tag{11}$$

By substituting $\beta_{\tilde{H}}$ and $\tilde{H}$ into (7), we get

$$\begin{aligned} |\mathcal{L}(\beta_{\tilde{H}}; \tilde{H}) - \mathcal{L}(\beta_{\tilde{H}}; \tilde{H}')| \leq \sqrt{n} \cdot \|\tilde{H} - \tilde{H}'\|_2 \cdot \|\beta_{\tilde{H}}\|_2 &\leq \sqrt{n} \cdot \|H - H'\|_2 \cdot \|\beta_{\tilde{H}}\|_2 \\ &\leq \sqrt{n} \cdot \sigma_{k+1}(H) \cdot \|\beta_{\tilde{H}}\|_2 \end{aligned} \tag{12}$$

After combining (10), (11) and (12) with triangle inequality, we get with high probability,

$$|\mathcal{L}(\beta_{\tilde{H}}; H) - \mathcal{L}(\beta_{\tilde{H}}; \tilde{H})| \leq 2 \cdot \sqrt{n} \cdot \sigma_{k+1}(H) \cdot \|\beta_{\tilde{H}}\|_2 + \epsilon \cdot \mathcal{L}(\beta_{\tilde{H}}; H'), \ \epsilon \in [0, 1] \tag{13}$$

which completes the proof of Theorem 1.

□

3 Results

3.1 Datasets

We evaluated methods on five datasets. We included two datasets that were used in [9] to benchmark scMulan. One contained 60,345 human cardiac nuclei from the left ventricle (LV) in 8 healthy controls [12] and one in which the authors in [9] sampled 200,000 cells from the human Ensemble Cell Atlas (hECA), which contains labeled human cells from 116 published datasets, covering 38 organs and 11 systems. Moreover, we benchmarked methods on a human skin atlas of 710,704 cells [13], on 504,278 nuclei from the cardiovascular (CV) system of the healthy Wistar rat [14], and on single-nuclei RNA-seq of 111,907 cells from the dorsal vagal complex (DVC) of mouse and rat [15]. Following the strategy in [4], all datasets were preprocessed by normalizing counts to 10,000 and log transformed after adding a pseudocount of 1. Gene expression was standardized to zero mean and unit variance. The hECA dataset contained 2,000 highly variable genes that the authors selected when benchmarking scMulan in [9].

3.2 Compressibility μ

We implemented the linear program introduced in Theorem 1 in Supplementary Section 1 to compute the classification complexity measure μ using Gurobi. We used this implementation to compute the complexity measure μ for the five datasets using varying numbers of principal components. As Figure 1 shows, omitting PCA yields infinitely large μ values. Under PCA transformation, μ decreases rapidly, rendering these PCA-transformed datasets as good candidates for coreset sampling in logistic regression. This inspired the formulation of Theorem 1 where we construct a coreset given that the PCA-transformed matrix has a good μ value. As remarked in Supplementary Section 1, having a small μ value is a sufficient condition for the coreset to be a good candidate for logistic regression problems, but not a necessary nor equivalent one. As experimental results will show in Section 3.5, although the five datasets have $\mu = \infty$, they indeed generate reasonable numerical outputs.

#PCs	10	100	1000	no PCA
LV	3.75	3.89	4.54	∞
DVC	3.21	3.37	3.80	∞
hECA	1.57	3.76	7.57	∞
CV	3.06	6.26	15.63	∞
Skin	4.21	5.60	6.15	∞

Figure 1: Complexity measure μ for varying number of principal components (#PCs) across all five datasets.

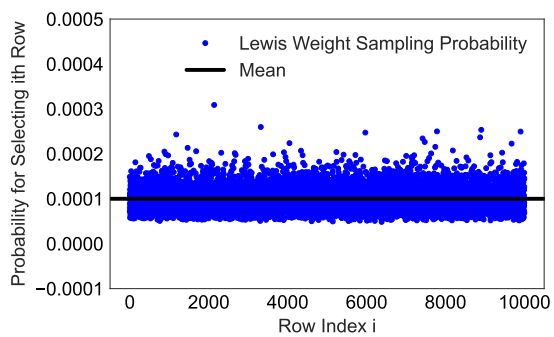


Figure 2: Lewis weights for 10k randomly selected cells from the hECA dataset.

Note that in Figure 1, the μ values under PCA are all considered small constants and can be represented as $O(1)$. According to [7] as well as in our Supplementary Section 1, the parameter μ roughly corresponds to the ratio between the log of correctly estimated odds and the log of incorrectly estimated odds. A μ value of ∞ indicates that the dataset is exactly separable, or in other words, there exists a hyperplane that perfectly divides all points in one class from all points in the other class. As the model assigns close to 1 probability to correctly classified points, this leads to the odds of correct estimates going to ∞. Thus we say this $\mu = \infty$ is not a good μ value and this data is not a good candidate for logistic regression. Although we currently do not have an exact numerical value for the threshold of good or bad μ, the μ values listed in Figure 1 after PCA are observed to be relatively small, and can be considered good. Because of this, in our Theorem 1, we specifically stress that if matrix $H' \in \mathbb{R}^{n \times k}$ has a good μ value, then we can construct a good coreset from matrix $H \in \mathbb{R}^{n \times d}$, where H is the original matrix, and H' is its rank k approximation.

3.3 Coreset Construction Technique

We construct coresets using uniform balanced sampling, where we uniform-randomly select a fixed number of cells per label to form coresets.

Uniform Sampling We used uniform sampling to select rows into coresets instead of Lewis weights as stated in Theorem 5. In fact, after normalizing Lewis weights to lie between zero and one, the probabilities of selecting rows (i.e. cells) vary tightly around $1/n$ which is used in uniform sampling, see Figure 2. In our experiments, we observed nearly uniform Lewis weights for all five datasets (data not shown).

Balanced Sampling Recall that our Theorem 1 implies that sampling $m = \tilde{O}(\frac{k \cdot \mu(H')^2}{\epsilon^2})$ rows from the original matrix H is sufficient to construct a coreset $\tilde{H} \in \mathbb{R}^{m \times d}$. Sampling size m can be assumed to be a constant value. This is because both k and $\mu(H')$ are constants, as k is chosen arbitrarily to a fixed value as seen in Figure 1 and $\mu(H')$ does not depend on n or d as summarized in Figure 1. Consequently, for a small constant error ϵ, the required coreset size m reduces to $\tilde{O}(1)$, where tilde hides a log factor in d and $\mu(H')$, and a $\log\log$ factor in n [5]. In our application, rather than solving the multi-class classification problem directly, we address it by decomposing it into binary classification problems—one for each cell type—using the one-versus-rest (OvR) strategy available in the scikit-learn logistic regression implementation. Therefore, the theoretical $\tilde{O}(1)$ coreset size represents the number of rows required *per label (cell type)*. To leverage this theoretical guarantee, we employ balanced sampling, selecting an equal number of cells for each cell type, thus ensuring a balanced representation in the coreset.

While we aim for balanced sampling by selecting a fixed number of cells per label for a given coreset ratio, some cell types might inherently have fewer total cells than this intended number. In such cases, we include all available cells for that label. This practice can lead to an observed coreset size (represented on the x-axis of Figure 3 and Figure 4) that is smaller than the intended size calculated by multiplying the intended coreset ratio with the number of rows.

3.4 Dimension Alignment after PCA Transformation

In our following experiments, we used PCA to construct the rank k approximation matrix $H' \in \mathbb{R}^{n \times k}$ for $H \in \mathbb{R}^{n \times d}$ where $k = 100 \leq d$ refers to the number of principal components. The rank k approximation matrix $H'' \in \mathbb{R}^{n \times d}$ in d-dimensional space can be calculated by multiplying $H' \in \mathbb{R}^{n \times k}$ with the principal component loadings, a matrix of size $(k \times d)$ with PCs as rows.

For the PCA version of our coreset-based logistic regression method (discussed in Section 3.5), although we trained the model on $H' \in \mathbb{R}^{n \times k}$, we can expand the k-dimensional regression coefficients back to its original d-dimensional space using the same principal components loadings as introduced previously.

3.5 Accuracy of Coreset-based Logistic Regression

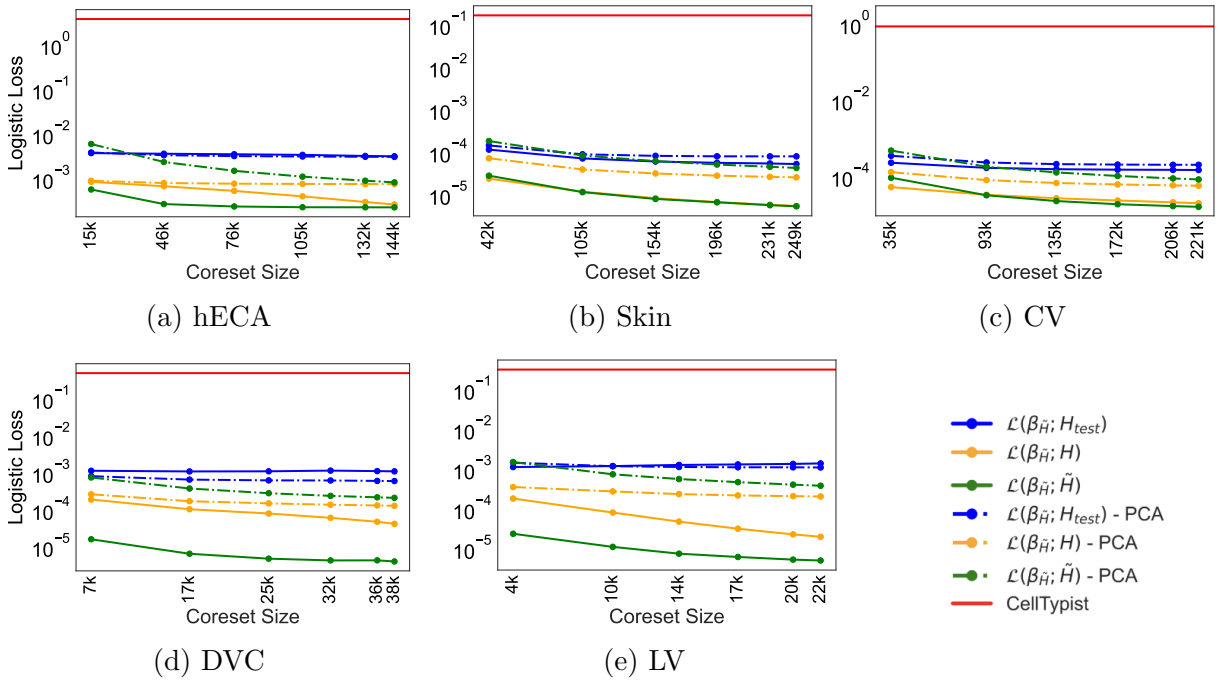
We evaluated logistic loss and balanced classification accuracy (sklearn's function balanced_accuracy_score) when constructing coresets with and without prior PCA transformation (randomized algorithm in sklearn's TruncatedSVD function) and selected 100 PCs on all datasets. Balanced accuracy accounts for imbalanced cell type compositions and is more sensitive to rare cell types. We compared our coreset-based approach to CellTypist, which follows an alternative strategy to speeding-up training a (ℓ_2-regularized) logistic regression model. It splits training data into mini-batches (1000 by default) and trains the model by stochastic gradient descent which led to an enormous speed-up compared to traditional logistic regression in [16]. Ground truth labels were obtained from the original publications. For the PCA version of our coreset based logistic regression method, although we trained the model on data with a reduced dimensionality of 100, we still tested on the same 20% data which has the original column size, by expanding the reduced coefficients back to its original feature space as described in Section 3.4. By fitting the model in this reduced subspace with additional back transforming, we gained computational speed while still complying with Theorem 1.

We split the data into 80% for training and 20% for testing. For our coreset-based logistic regression approach we trained a ℓ_2-regularized logistic regression model using function LogisticRegression of the scikit-learn package. We performed 3-fold cross validation on a coreset of size 10% instead of the entire dataset to further speed-up our approach. For LogisticRegression's parameter C (which is the inverse of regularization coefficient λ in Equation 1), cross validation based on the negative logistic loss yielded values 0.001 for datasets CV and Skin and 0.1 on all remaining datasets. Using the same 3-fold cross validation on the entire training set, the corresponding parameter C was set by CellTypist to 0.00001 for datasets CV and Skin and to 0.0001 for the remaining three datasets.

The theoretic guarantee on the coreset-based logistic loss provided by Theorem 1 is supported by experiments on all five datasets. In Figure 3, we observe near-identical loss on the entire matrix (orange) and its coreset (green) after rank k approximation (here PCA; dotted lines) for all five datasets. Even when omitting the PCA transformation, orange and green solid lines show only a small gap on datassets hECA, Skin and CV which further decreases with increasing coreset size. The gaps between the solid green and orange lines are slightly larger for the DVC and LV data. One possible reason lies in the coreset construction technique mentioned in Section 3.3. As the observed coreset size is less than the intended size based on coreset ratio, rare cell types are likely to be under-represented, and this will affect logistic loss as well as balanced accuracy (discussed later). Despite the larger discrepancies, the gap-minimizing trend as coreset sizes increase still holds true in Figure 3. Across all datasets and across all coreset sizes, the theoretic guarantees on the training loss yielded a more accurate solution, that is, a smaller logistic loss on the test data (blue lines) compared to the model trained by CellTypist (red line).

Although classification accuracy is not directly controlled by our theory and depends on specific properties of the dataset–such as the relative abundance of cell types, gene expression similarity between cell (sub-)types, and heterogeneity of cell states–Figure 4

demonstrates that a theoretically guaranteed small logistic loss results in accurate cell type classification in practice. As observed in Figure 3 and Figure 4, our coreset-based logistic regression method outperforms a competing cell type annotation approach that utilizes the entire dataset, which suggests that the number of sampled cells per cell type significantly exceeds the theoretical $\tilde{O}(1)$ threshold across all five datasets.



(a) hECA (b) Skin (c) CV

(d) DVC (e) LV

Figure 3: Logistic loss (without regularization term) in log scale as a function of observed coreset size. The actual coreset sizes shown on the x-axis are typically smaller than the intended coreset ratios (here 0.1, 0.3, 0.5, 0.7, 0.9, and 1.0) and are listed for all datasets in Table 1 in Supplementary Section 2. Logistic loss is shown for the original matrix H (orange), its coreset (green), and on the test data (blue), with (dotted line) and without (solid line) prior PCA transformation, and compared to CellTypist's loss (red) on the same test data.

The slight decreasing trend observed in Figure 4 (b)-(e) as the coreset size increases might be attributed to two factors: (1) potential overfitting with larger coresets, and (2) poorer per-class accuracy for less represented cell types negatively impacting the overall balanced accuracy. This latter point can arise due to our coreset construct technique addressed in Section 3.3. For the experiments shown in Figure 4, we also conducted an additional CellTypist experiment where the "top genes" parameter (a built-in CellTypist setting) was set to 100. In this configuration, CellTypist selects the top 100 genes for each label based on their absolute regression coefficients; the final feature set used for classification is the union of these selected genes across all labels. This additional feature selection step had a negligible effect on the balanced accuracy of CellTypist (Figure 4).

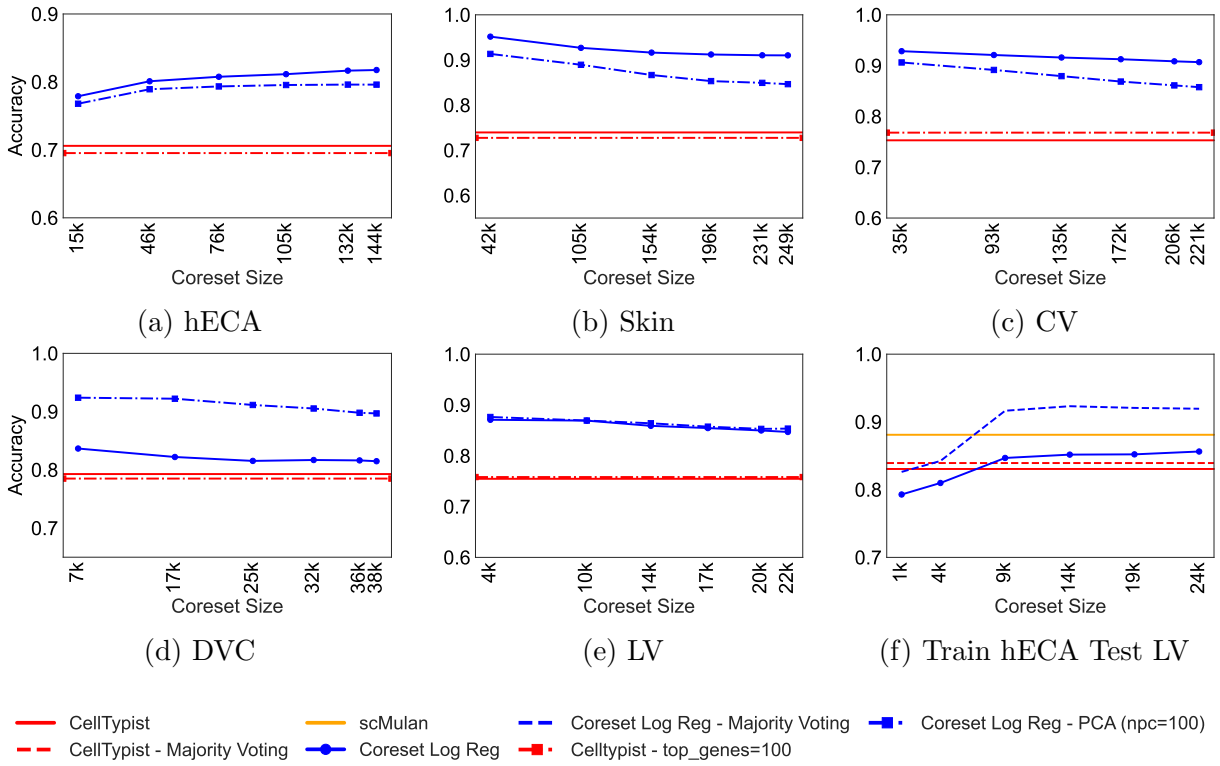


Figure 4: Accuracy as a function of observed coreset size. (a)-(e): Balanced accuracy of coreset-based logistic regression (blue) with and without PCA (dotted and solid lines, respectively) and of CellTypist (red) with and without selecting the top 100 genes (dotted and solid lines, respectively) achieved on 20% test data. The actual coreset sizes shown on the x-axis are typically smaller than the intended coreset ratios (here 0.1, 0.3, 0.5, 0.7, 0.9, and 1.0) and are listed for all datasets in Table 1 in Supplementary Section 2. (f): Accuracy of coreset-based logistic regression (blue), CellTypist (red), and scMulan (orange) when training on hECA and testing on LV. Dashed blue and dashed red lines indicate additional majority voting for coreset-based logistic regression and CellTypist, respectively. The x-axis represents observed coreset ratios corresponding to 0.005, 0.025, 0.05, 0.075, 0.1, and 0.125 of in total 200,000 cells, shown from left to right.

Finally, we evaluated the generalization ability of the different models by training them on the hECA dataset and testing them on the LV data. This is the zero-shot setting without fine-tuning in which the cell type annotation performance of scMulan was benchmarked in [9]. To harmonize cell type labels between datasets, we used the label mapping provided by the authors in [9]. In addition, we followed the evaluation scheme in [9] by not considering the identification of a true cell type label as one of its subtypes as a false positive prediction. For testing on LV data, we selected the same 2,000 genes that were used when training on the hECA dataset.

Even in this zero-shot setting, coreset-based logistic regression achieved results comparable to scMulan (Figure 4f). With a simple majority voting scheme similar to that applied by CellTypist in [4], our coreset-based logistic regression approach predicted cell

types with an overall accuracy of 91.6%—achieved using only 5% of the data. In contrast, scMulan's complex language model reached 88.1% accuracy when trained on the entire set of 10 million cells, while CellTypist achieved 83.9% overall accuracy after majority voting.

3.6 Running Times

Besides theoretical guarantees on the logistic loss, the main advantage of the coreset-based approach is its efficiency. The experiments in this work were run on Intel Xeon Gold 6226R CPU with 8 cores.

Cross Validation Figure 5 compares the running times of CellTypist and our coreset-based approach for performing cross-validation. On the hECA dataset, CellTypist required about one order of magnitude more computation time than our coreset-based logistic regression method for 3-fold cross-validation.

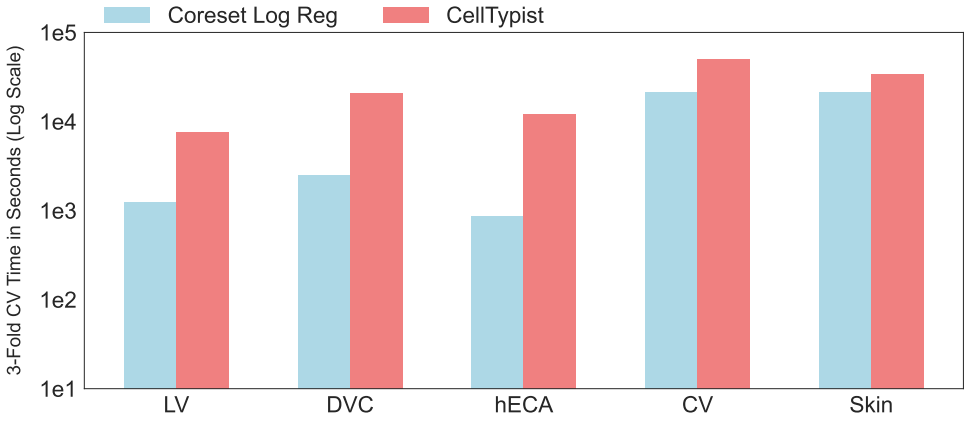


Figure 5: Total time in seconds (log scale) needed for 3-fold cross validation on the scoring metric of negative logistic loss with intended coreset ratio 0.1 (observed coreset size might be smaller) for the two logistic regression based models.

Training Step Figure 6 shows the computation times for all five datasets, using the optimal regularization parameter C selected during cross-validation. As expected, the coreset-based logistic regression with PCA achieves the lowest overall computation time, even when including both the PCA transformation and model training. In contrast, CellTypist, with or without the selection of the top 100 genes, consistently incurs the highest computational cost across all datasets. On the LV dataset, for example, CellTypist requires approximately 1.5 orders of magnitude more time than our coreset-based logistic regression with PCA.

Remarkably, to reach a level of accuracy that is comparable to that of scMulan (Figure 4f), our approach required only 56 seconds to train on just 5% of cells, compared to 192 hours scMulan needed for training on 10 million cells on four Nvidia-A800 GPUs.

For comparison, the CellTypist approach needed 633 seconds for training. Furthermore, annotating cell types in the LV dataset based on the pre-trained scMulan model exceeded 9 hours of computation time, which the authors suggest can be reduced substantially using GPUs [9]. In contrast, predicting cell types took logistic regression based methods around one second.

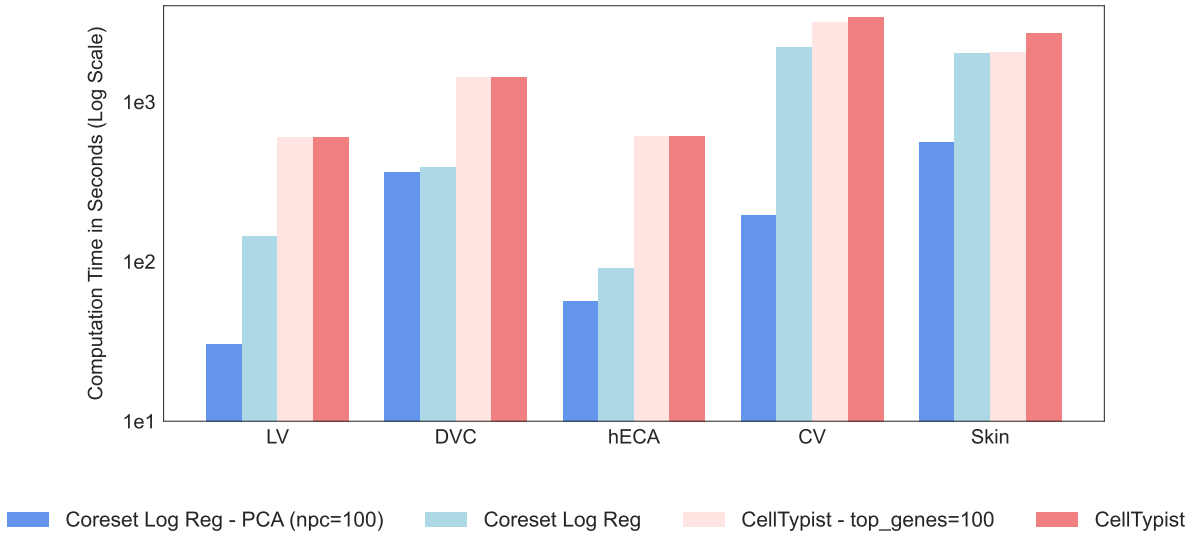


Figure 6: Total time in seconds (log scale) needed to train our coreset-based logistic regression with intended coreset ratio 0.1 (observed coreset size might be smaller) compared to CellTypist with and without selecting 100 top genes. Running times are shown with and without prior PCA transformation and include PCA time if applied.

4 Conclusion

Here we have proposed a simple cell type annotation approach based on the theory of coresets to train a logistic regression model on large datasets efficiently and with theoretically guaranteed performance. In our experiments, we have demonstrated that these guarantees translate to an accurate labeling of cell types. In particular, we showed that the single-cell datasets used in this study yield a low complexity measure after transforming the data using PCA which is commonly applied in scRNA-seq analysis. We prove that logistic loss is preserved under this transformation. Training a logistic regression model on a coreset is rapid enough to facilitate the transfer of cell type labels from sources such as immune cell atlases and other tissue specific atlases to individual single-cell studies.

Even though we sample rows (cells) uniformly at random, our theoretical guarantees are stronger than previous results based on uniform sampling [17]. As noted in our Lewis weights calculation, the weights are near uniform. Our conjecture is that if rows are i.i.d. samples (like cells in single-cell omics experiments), Lewis weights must be within some constant factor of the uniform distribution with high probability.

We note that, despite the proposed subsampling of cells for annotating cell types, single-cell studies can greatly benefit from profiling a large number of cells. For instance, a prior clustering of cells based on their overall transcriptional similarity can uncover rare cell types or previously unknown states, provided that a sufficiently large sample of those cells is captured and sequenced. Indeed, more sophisticated sketching approaches, such as Sphetcher [18], directly utilize large datasets by sampling cells that accurately represent the transcriptional heterogeneity of the entire dataset. In the supervised setting studied in this work, cell types are assumed to be known during training, allowing for a balanced sampling strategy that ensures the inclusion of rare cell types in the coreset.

In our benchmark, scMulan–pre-trained on 10 million cells–achieved slightly inferior results compared to logistic regression combined with a simple majority voting scheme, even in the more challenging zero-shot setting. This finding aligns with previous research [3]. Nevertheless, through fine-tuning, foundation models provide powerful tools for various other types of downstream analyses of single-cell datasets, and their capabilities will continue to improve as the size of the training datasets increases.

Here our focus was on cell type annotation, but logistic regression is used widely in the predictive analysis of single cell omics data, and has shown particularly strong performance in the selection of marker genes [19]. We will study the suitability of similar coreset-based approaches for other types of analyses in future work.

The code for our coreset-based logistic regression, the linear program for computing μ, and the code for reproducing our results obtained for method CellTypist is available at `https://github.com/danrongLi/coreset_based_log_reg`. In addition, the five datasets used in this study can be downloaded from the same repository.

5 Acknowledgements

We thank the authors of scMulan [9] for providing us with the label mapping between the hECA and the LV dataset, and the cell type hierarchy used in the evaluation of accuracy.

References

1. Hao, M. *et al.* Large-scale foundation model on single-cell transcriptomics. en. *Nature Methods* **21,** 1481–1491. ISSN: 1548-7091, 1548-7105. (2024) (Aug. 2024).

2. Cui, H. *et al.* scGPT: toward building a foundation model for single-cell multi-omics using generative AI. en. *Nature Methods* **21,** 1470–1480. ISSN: 1548-7091, 1548-7105. (2024) (Aug. 2024).

3. Boiarsky, R., Singh, N., Buendia, A., Getz, G. & Sontag, D. *A Deep Dive into Single-Cell RNA Sequencing Foundation Models* en. Oct. 2023. (2025).

4. Domínguez Conde, C. *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. en. *Science* **376,** eabl5197. ISSN: 0036-8075, 1095-9203. (2024) (May 2022).

5. Mai, T., Musco, C. N. & Rao, A. *Coresets for Classification – Simplified and Strengthened* en. in *Advances in Neural Information Processing Systems* (Nov. 2021). (2024).

6. Daliri, M., Freire, J., Li, D. & Musco, C. *Matrix Product Sketching via Coordinated Sampling* arXiv:2501.17836 [cs]. Jan. 2025. (2025).

7. Munteanu, A., Schwiegelshohn, C., Sohler, C. & Woodruff, D. *On Coresets for Logistic Regression* in *Advances in Neural Information Processing Systems* **31** (Curran Associates, Inc., 2018). (2024).

8. Dexter, G., Khanna, R., Raheel, J. & Drineas, P. *Feature Space Sketching for Logistic Regression* arXiv:2303.14284 [cs, stat]. Mar. 2023. (2024).

9. Bian, H. *et al. scMulan: a multitask generative pre-trained language model for single-cell analysis* en. Jan. 2024. (2024).

10. Eckart, C. & Young, G. The approximation of one matrix by another of lower rank. en. *Psychometrika* **1,** 211–218. ISSN: 0033-3123, 1860-0980. (2024) (Sept. 1936).

11. Cohen, M. B. & Peng, R. *ℓ_p Row Sampling by Lewis Weights* arXiv:1412.0588 [cs, math]. Dec. 2014. (2024).

12. Simonson, B. *et al.* Single-nucleus RNA sequencing in ischemic cardiomyopathy reveals common transcriptional profile underlying end-stage heart failure. en. *Cell Reports* **42,** 112086. ISSN: 22111247. (2024) (Feb. 2023).

13. Fiskin, E. *et al. Multi-modal skin atlas identifies a multicellular immune-stromal community associated with altered cornification and specific T cell expansion in atopic dermatitis* en. Oct. 2023. (2025).

14. Portal, B. I. S. C. *Transcriptional profile of the rat cardiovascular system at single cell resolution*

15. Hes, C. *et al. A unified rodent atlas reveals the cellular complexity and evolutionary divergence of the dorsal vagal complex* en. Sept. 2024. (2025).

16. Cox, D. R. The Regression Analysis of Binary Sequences. en. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **20,** 215–232. ISSN: 1369-7412, 1467-9868. (2024) (July 1958).

17. Curtin, R. R., Im, S., Moseley, B., Pruhs, K. & Samadian, A. *On Coresets for Regularized Loss Minimization* Version Number: 2. 2019. (2024).

18. Do, V. H., Elbassioni, K. & Canzar, S. Sphetcher: Spherical Thresholding Improves Sketching of Single-Cell Transcriptomic Heterogeneity. en. *iScience* **23,** 101126. ISSN: 25890042. (2024) (June 2020).

19. Pullin, J. M. & McCarthy, D. J. A comparison of marker gene selection methods for single-cell RNA sequencing data. en. *Genome Biology* **25,** 56. ISSN: 1474-760X. (2024) (Feb. 2024).