Biorxiv, 2024, pp. 1–10

doi: -
Paper

PAPER

Tissue-Specific Cell Type Annotation with Supervised Representation Learning using Split Vector Quantization and Its Comparisons with Single-cell Foundation Models

Yusri Dwi Heryanto,[1] Yao-zhong Zhang[1,*] and Seiya Imoto[1,*]

[1]Division of Health Medical Intelligence, Human Genome Center, The Institute of Medical Science, The University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, 108-8639, Tokyo, Japan
[*]Corresponding author: yaozhong@ims.u-tokyo.ac.jp
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Cell-type annotation in single-cell data involves identifying and labeling the cell types in a dataset based on their gene expression profiles or molecular features. Recently, with advances in foundation models (FMs) trained on single-cell data, unsupervised annotation and transfer learning with FMs have been explored for cell-type annotation tasks. However, because FMs are usually pre-trained in an unsupervised manner on data spanning a wide variety of tissues and cell types, their representations for specific tissues may lack specificity and become overly generalized. In this work, we propose a novel supervised representation learning method using split-vector-quantization, single-cell Vector-Quantization Classifier (scVQC). We evaluated scVQC against both supervised and unsupervised representation learning approaches, with a focus on foundation models (FMs) pre-trained on large-scale single-cell datasets, such as scBERT and scGPT. The experimental results highlight the importance of label supervision in cell-type annotation tasks and demonstrate that the learned codebook effectively profiles and distinguishes different cell types.

Key words: single cell RNASeq, vector quantization, cell annotations, foundation models

Introduction

Cell annotation is a crucial step in single-cell analysis. Two commonly used methods for this are unsupervised clustering with marker-based annotation and label transfer using reference datasets [1]. In the first method, cells are grouped into clusters based on their expression profiles using unsupervised clustering algorithms. These clusters are then annotated by identifying differentially expressed genes and matching them to known cell type markers from literature or databases. The unsupervised clustering methods can be time-intensive and requires expertise in selecting reliable marker genes. To overcome these limitations, label transfer methods were developed. This approach utilizes annotated reference datasets to automatically assign cell type labels to new data through computational tools, offering a faster, consistent solution that leverages prior knowledge from reference datasets.

Recent studies demonstrated the effectiveness of foundation models (FMs) in single-cell analysis [2, 3, 4]. These models are pretrained on extensive single-cell datasets spanning various human tissues, organs, and even data from other species. After pretraining, the models can be fine-tuned for a range of downstream tasks. Among the leading LLMs in single-cell analysis are scGPT [3] and scBERT [2].

While FMs hold significant promise for single-cell genomics, the field also faces notable challenges and criticisms. Since FMs are pre-trained in a self-supervised or unsupervised manner on diverse single-cell datasets from various tissues, the representations they learn may lack specificity. This issue arises because the definition of a cell type often depends on the tissue or organ in which it resides (e.g., epithelial cells in the skin differ from epithelial cells in the pharynx). As a result, FMs may generalize too broadly, potentially compromising their ability to capture tissue-specific distinctions [5]. Additionally, the computational, time, and memory demands of FMs are considerable, even for fine-tuning only [6, 7]. Some FM architectures also require specific GPU configurations; for instance, scGPT relies on FlashAttention [8], which is unavailable on older or smaller GPUs.

In cell type annotation, foundation models (FMs) map input features into an embedding space learned from pretrained data, where a classifier is trained to categorize cells. Cell representations in this embedding space is learned in an unsupervised manner. A common approach involves using a variational autoencoder (VAE) to learn cell representations [9].

© The Author 2024. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

However, standard VAEs have limitations in single-cell type classification, as they assume a multidimensional normal prior for the low-dimensional latent variables. This assumption often leads to clustering cell representations near the center of the latent space, even when distinct cell types are present [10]. Additionally, VAE models are prone to degenerate solutions, a problem known as posterior collapse [11]. To address these issues, the Vector Quantised-Variational AutoEncoder (VQ-VAE) was introduced [12]. Unlike VAEs, VQ-VAE encodes discrete rather than continuous codes. Discrete representations are generally more suitable for cell type prediction, providing clear boundaries between cell types, reduced overlap in latent space, and avoiding posterior collapse [13].

However, representations learned during self-supervised pretraining or unsupervised learning may not be optimal for cell classification. A supervised neural network can be trained to refine these representations for classification tasks. For instance, SCLSC [14] uses supervised contrastive learning to enhance cell type representations, aligning each sample closely with its representative cell in the new embedding space while maintaining separation from representatives of other labels. SCLSC assumes that the gene profile of a representative cell for each type is the average of gene profiles for samples within that cell type.

Inspired by the previous works, we developed the single-cell Vector-Quantization Classifier (scVQC) model for annotating cell types in scRNA-seq data. The scVQC model is designed to both predict cell types accurately and extract cell latent embeddings, achieving performance comparable to foundational models in cell annotation tasks. This model leverages split-vector-quantization to create discrete representations, which helps in distinguishing between cell types with greater clarity.

In this study, we conducted a comprehensive comparison of foundation model (FM)-based methods, specifically scGPT and scBERT, with supervised neural network approaches, including SCLSC and scVQC. Utilizing four benchmark single-cell datasets, we evaluated the performance of these methods across multiple dimensions: classification accuracy, clustering quality, representation learning capabilities, and scalability 1. Our analysis aimed to provide insights into how well each approach captures the underlying structure of single-cell data, as well as their ability to handle large-scale datasets efficiently.

Materials and methods

Data Preprocessing

Before analysis, we split each dataset into two parts: 60% for the training set and 40% for the test set using a stratified split. We then applied a preprocessing pipeline on these datasets. In this pipeline, cells with high mitochondrial gene expression (greater than 5% of the total cell count) were excluded, along with cells showing minimal gene expression (fewer than 200 genes per cell) and genes detected in only a small number of cells (fewer than 3 cells expressing the gene). Afterward, we normalized each cell's count to 10,000 counts and applied a log(x+1) transformation. We then selected the top 3,000 highly variable genes (HVGs) for further analysis.

The scVQC Framework

The scVQC takes as input a preprocessed gene-profile matrix X of dimensions $n \times d$, where n represents the number of cells and d denotes the number of genes. The model is composed of three key components: an encoder that maps the feature space to a continuous latent embedding space, a split quantizer that converts this continuous space into a discrete latent embedding space, and a classifier head that predicts cell types based on these discrete embeddings (Figure 2).

Encoder structure

The encoder accepts $x_i \in \mathbb{R}^d$ as the input where d denotes the number of genes. The gene profile x_i represents the i-th row of the matrix X, which is a vector of gene expression values for cell i. The encoder in scVQC consists of two Linear-dropout-Relu-batch normalization block layers and followed by dense linear layer. The output of the encoder for each x_i is $z_i \in \mathbb{R}^{d'}$ where d' denotes the dimension of embedding space.

In our study, the linear layers contain 1024, 256, and 100 nodes in the first, second, and third layers, respectively. The dropout rate for both dropout layers is set to 0.1.

Split quantizer

The latent embedding z_i is then split into M vectors $z_i^m, m \in \{1, ...M\}$. Using Vector Quantization [12], each split z_i^m is assigned to the nearest code $e_k \in \mathbb{R}^{d^*}, k \in \{1, ...K\}$ in the codebook. This assignment is represented by as Q_i^m. Here $d^* = \frac{d'}{M}$ is the dimension of the codes and K is a hyperparameter defining the number of codes within the codebook. This assignment is formulated as follows:

$$q_i^m = e_j \text{ where } j = \underset{k \in \{1,\ldots,K\}}{\arg\min} \|z_i^m - e_k\|_2 \tag{1}$$

Next, we concatenate back each q_i^m into $q_i \in \mathbb{R}^{d'}$ that represents the quantized z_i. The

The split vector quantization [15] is a modified approach to vector quantization designed to improve codebook utilization by addressing the issue of underused codes. Instead of quantizing all dimensions of a vector at once, the vector is first divided into smaller, equally sized subvectors. Each subvector is then quantized individually using a shared codebook. From previous studies, this method enhanced codebook utilization compared to models that perform quantization on the entire vector simultaneously [13].

In our study, we set the codebook to have 200 codes, an embedding dimension of 100, and 20 splits (i.e. resulting in each split having a dimensionality of $100 \div 20 = 5$).

Classifier head

The classifier head accepts q_i, the output from the split quantizer, as its input. It consists of a linear layer followed by a log-softmax activation layer for classification.

Training Process

For each cell, a gene expression profile vector x_i and its associated cell type label y_i from the training dataset are provided as input to scVQC. The encoder transforms the input data into a continuous latent embedding, which is then converted into a discrete latent embedding by the split quantizer. This discrete latent embedding is subsequently used by the classifier head to predict the cell type. The training objective of scVQC comprises three components. The first component is the negative log likelihood (NLL) loss for measuring classification accuracy, defined as:

$$\text{NLL Loss} = -\frac{1}{N}\sum_{i=1}^{N} \log(p(y_i|x_i)) \tag{2}$$

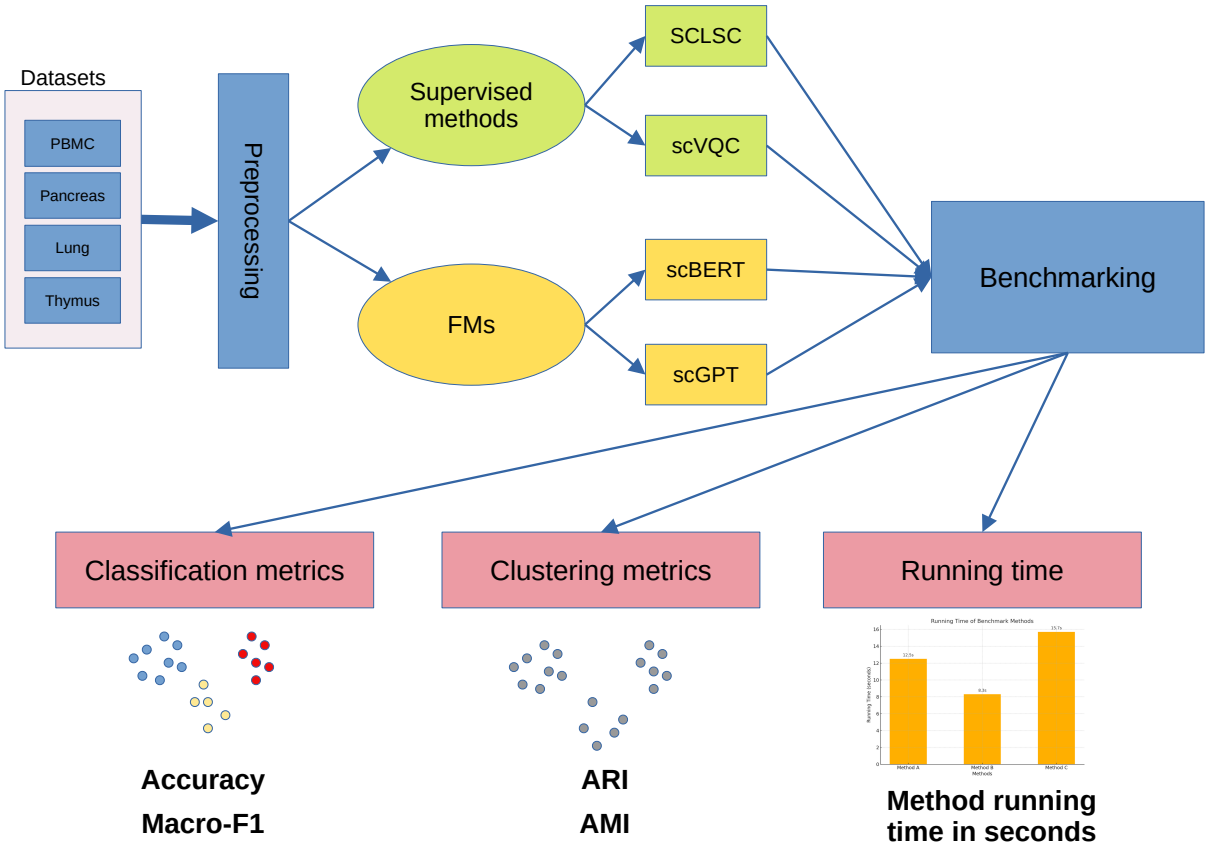


Fig. 1. The workflow of our study. The analysis begins with preprocessing four datasets (PBMC, Pancreas, Lung, and Thymus), followed by comparisons between supervised methods (SCLSC and scVQC) and foundation models (scBERT and scGPT). Benchmarking includes evaluations of classification metrics (Accuracy, Macro-F1), clustering metrics (ARI, AMI), and method running times.

where N is the number of samples, and $p(y_i|x_i)$ is the predicted probability for the correct class y_i, given the inputx_i. This classification loss updates the encoder and classification head.

The second component is the commitment loss. This loss ensures the encoder output is "committed" to a specific code in the codebook. The commitment loss defined as follows:

$$\text{commitment loss} = \beta \|z_i - sg(e_k)\|_2 \tag{3}$$

where sg represents the stop-gradient operator, which acts as an identity function during forward computation but has zero partial derivatives. This operator prevent the operands get updated during backpropagation. The commitment loss only update the encoder. The β is a factor that balances the trade-off between the commitment loss and other losses. We set $\beta = 0.25$, the same default value used in the original paper [12].

The last component is the codebook update, designed to update the codebook by moving selected code vectors closer to the encoder's output. We used the Exponential Moving Average (EMA) from Oord et al. [12] as the function for updating the codebook. Specifically, for each code vector e_k in the codebook, the EMA update rule is:

$${e_k}^{new} = \frac{(1-\gamma) \cdot e_k + \gamma \cdot z_i}{L_k} \tag{4}$$

where γ is the decay factor of EMA set at 0.99, ${e_k}^{new}$ and e_k is the updated and old codes, z_i the encoder's output for an input that has been assigned to e_k, and L_k is the usage count for e_k. This count helps normalize updates and prevent inactive codebook vectors from drifting randomly.

Thus, the total training loss:

$$\text{loss} = \text{Classification loss} + \text{Commitment loss} + \text{Codebook loss} \tag{5}$$

We trained the scVQC for 5 epochs in our study with learning rate 5e-4 using Adam optimizer. For datasets containing more than 100,000 cells, the batch size is set to 128. For datasets with 100,000 cells or fewer, the batch size is set to 32.

Competing Methods

SCLSC

SCLSC[14] is a deep learning method developed to apply supervised contrastive learning principles for transferring annotations from a single-cell reference dataset to a query dataset. By contrasting each cell with its cell type representation, SCLSC effectively learns a low-dimensional representation that is well-suited for cell type prediction. We used the default hyperparameters provided in the SCLSC

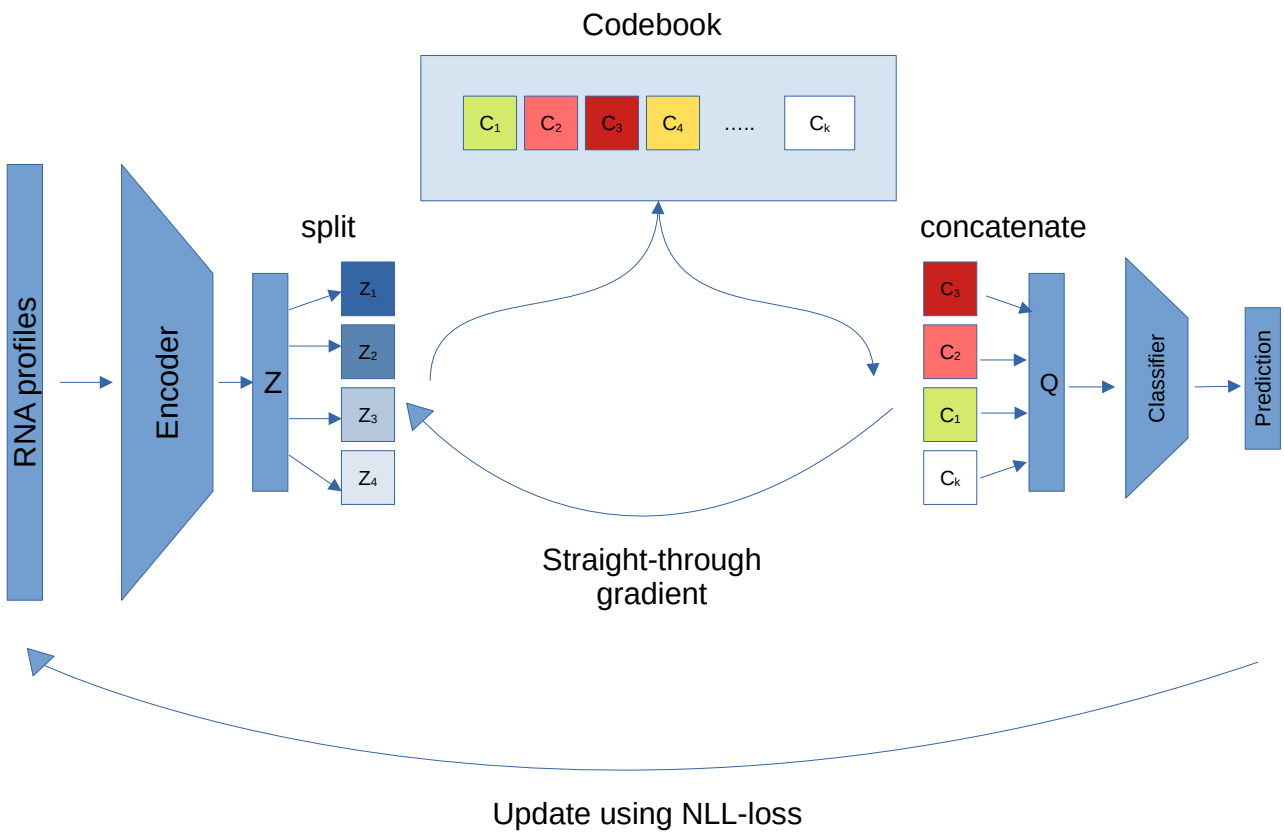


Fig. 2. The scVQC architecture.The scVQC consists of three main components: a 3-layer MLP encoder, a split-vector quantizer, and a dense layer with a log-softmax activation serving as the classifier head.

GitHub repository (`https://github.com/yaozhong/SCLSC`), with a maximum training epoch of 30.

scGPT

scGPT[3] is a generative pre-trained model designed for analyzing single-cell RNA data. Its core architecture consists of stacked transformer layers with multi-head attention, enabling simultaneous generation of cell and gene embeddings. Pretrained on large-scale, diverse datasets with over 33 million cells from the CELLxGENE collection, scGPT produces rich cell representations. These embeddings are then used as input to a multi-layer perceptron for cell-type classification. We used the hyperparameters and workflow from the scGPT tutorial in the GitHub repository (`https://github.com/bowang-lab/scGPT/blob/main/tutorials/Tutorial_Reference_Mapping.ipynb`) to learn cell embbedings. We then employed a simple two-layer Multi-Layer Perceptron (MLP) classifier trained in the embedding space to transfer labels from the reference dataset to the query dataset. The first layer takes an input with the dimension same as the dimension of embedding space and projects it to a hidden layer of 64 neurons. This hidden layer is followed by a ReLU activation function. The second linear layer then maps the hidden layer's output to number of class outputs. Finally, a log softmax function is applied to the output, which normalizes the values into log-probabilities across the classes.

scBERT

scBERT[2] is a large language model built on the architecture and pretraining strategies of BERT (Bidirectional Encoder Representations from Transformers) [16]. It was pretrained on the PanglaoDB dataset, which includes 1,126,580 cells from 74 different tissues, gathered from a range of experimental sources and platforms. After pretraining, scBERT is fine-tuned specifically for cell-type classification. The query dataset annotation was carried out using the default hyperparameters and workflow provided in the scBERT repository (`https://github.com/TencentAILabHealthcare/scBERT`). To obtain the cell embeddings, we used the output from the PerformerLM module and passed it through a convolutional layer.

Comparison with Standard-VQ

In this experiment, we assessed the impact of the split-vector quantizer on the performance of scVQC. For the Conventional Standard-VQ, we set the number of splits to one, equivalent to performing vector quantization without any splitting. For the Split-VQ procedure, we used the default number of splits set to 20, ensuring that the dimensionality of each split embedding remained at 5. All hyperparameters were kept at their default values for both Split-VQ and Standard-VQ.

Feature Spectrum

The feature spectrum was obtained using the Term Frequency Inverse Document Frequency (TF-IDF) transformation applied to the utilization frequencies of codes within the codebooks. In this approach, each code in the codebook was treated as a word, each cell type as a document, and the entire single-cell dataset as a corpus or collection of documents. To calculate the Term Frequency (TF), we measured the relative frequency of each code within a cell type by dividing the raw count of a code's occurrences in the cell type by the total number of codes in that cell type. To evaluate how common or rare a code was across all cell types, we computed the Inverse Document Frequency (IDF) as the logarithm of the inverse of the fraction of cell types (documents) containing the code. The TF-IDF for each code in each cell type was then calculated as $TF \times IDF$. This process was applied to all codes across all cell types. It will produce a TF-IDF matrix $T \in R^{C \times Q}$ where C is the number of cell types and Q the number of codes. Each row of the matrix T is the feature spectrum of the corresponding cell type.

To aid the visualization of the matrix T, we rearranged its columns as follows. First, for each column j we computed two measures:

$$t_j = \max(T_{i,j}), i \in (1, ..., C) \tag{6}$$

$$n_j = \arg\max(T_i, j), i \in (1, ..., C) \tag{7}$$

where C is the number of cell types. The columns of the T were then reordered in non-descending order based on the n_j values, grouping codes that are highly utilized within each cell type. The last step, for each row, we selected the columns that contained t_j. The selected columns were then further reordered in non-ascending order of the t_j values. This final step was repeated for each row.

Evaluation Metrics

We assessed the performance of our method relative to other approaches using Accuracy and Macro-Weighted F1 Score metrics. To compute these scores, we calculated the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Accuracy was defined as the ratio of correct classifications, both positive and negative, to the total number of classifications. The accuracy was calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

For the macro-weighted F1-score, we calculated precision P_i and recall R_i for each cell type $i \in C$ to obtain the F1 score for each cell type. We then computed the unweighted average of these F1 scores. The formula is as follows:

$$\text{F1}_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i},$$

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{i=1}^{|C|} \text{F1}_i \tag{9}$$

where $|C|$ is the number of cell types. We used the sklearn implementations sklearn.metrics.accuracy_score() and sklearn.metrics.f1_score(average='macro') for calculating accuracy and macro-F1 score, respectively.

We further evaluated our method's ability to learn low-dimensional representations using clustering metrics, including the Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI). Both ARI and AMI are adjusted versions of the Rand Index and Mutual Information scores, accounting for chance and correcting for agreements that may occur solely by chance between clusterings. To compute these metrics, we used the implementations provided by sklearn.metrics: adjusted_rand_score() for ARI and adjusted_mutual_info_score() for AMI.

Results

Classification performance comparison

We conducted a comparative analysis of cell type prediction performance across four human datasets: PBMC 68k [17], lung [18], thymus [19], and pancreas [18]. We compared scVQC with a supervised model and two foundational models. The supervised model, SCLSC [14], uses supervised contrastive learning to learn cell representations in a low-dimensional embedding space optimized for classification. The foundational models are scBERT, a BERT-based large language model [16] pre-trained on 1 million human cells for cell type prediction, and scGPT [3], a foundation model for single-cell biology pre-trained on over 33 million single-cell sequencing data, supporting diverse downstream tasks such as cell-type annotation, multi-omic integration, and gene network inference.

The supervised approaches (scVQC and SCLSC), particularly scVQC, consistently outperform the foundational models in key metrics across most datasets (Figure 3). For accuracy, scVQC achieves the highest scores in the lung, pancreas, and thymus datasets, while SCLSC also performs competitively, maintaining scores close to scVQC. In contrast, scGPT lags significantly behind, with notably lower accuracy across all datasets. scBERT performs well in accuracy, achieving the highest score for PBMC, but its performance is less consistent across other datasets compared to scVQC.

For the Macro-F1 metric, which evaluates the balance between precision and recall, the supervised approaches again show clear advantages (Figure 3). scVQC excels with the highest Macro-F1 scores in the pancreas and thymus datasets, demonstrating its ability to classify both common and rare cell types effectively. SCLSC follows closely in performance but lags behind scVQC, particularly in the pancreas dataset. Foundational models perform less effectively in this metric, with scGPT showing particularly poor scores, and scBERT falling behind on thymus and pancreas. While scBERT is ranked first in Macro-F1 for PBMC, it does not consistently maintain this performance across datasets.

Overall, supervised approaches, especially scVQC, exhibit superior performance in both accuracy and Macro-F1 compared to foundational models. scVQC's ability to learn task-specific cell representations through supervision gives it a distinct edge over foundational models, which, despite their generalizability, underperform in some tasks.

Evaluation of clustering metrics

We further examined scVQC and other methods ability to cluster cell representations in the lower-dimensional embedding space. We used ARI and AMI metrics to assess clustering quality, with higher scores indicating a closer alignment with true cell type labels.

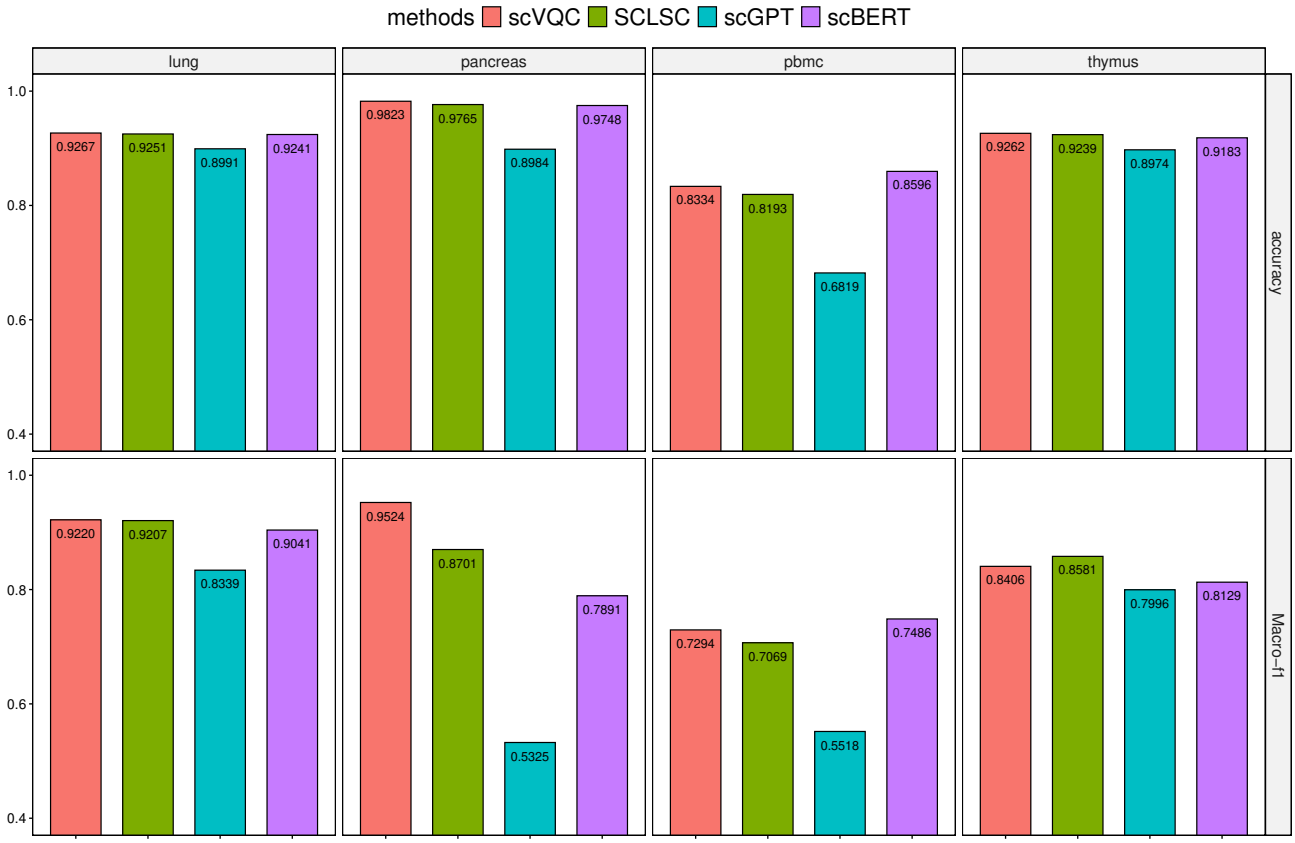


Fig. 3. Comparison of Cell Classification Performance. The supervised approaches (scVQC and SCLSC) have better classification performance overall compared to foundational models (scBERT and scGPT) for cell classification tasks using accuracy and Macro-F1 metrics.

Our results reveals a significant advantage for supervised methods (Figure 4a). scVQC excelled in ARI, achieving the highest scores in almost all datasets, while SCLSC demonstrated competitive or superior performance in AMI, particularly in the PBMC and thymus datasets. In contrast, foundational models scGPT and scBERT lagged behind in both ARI and AMI across all datasets. scGPT generally performed better than scBERT, particularly in ARI, but still fell short compared to supervised approaches. scBERT exhibited inconsistent performance, with relatively better AMI in the pancreas dataset but significantly lower scores in other datasets. These results show that supervised approaches are more effective than foundational models, such as scGPT and scBERT, in clustering cells of the same type more tightly while maintaining clearer separation between different cell types. (Figure 4b).

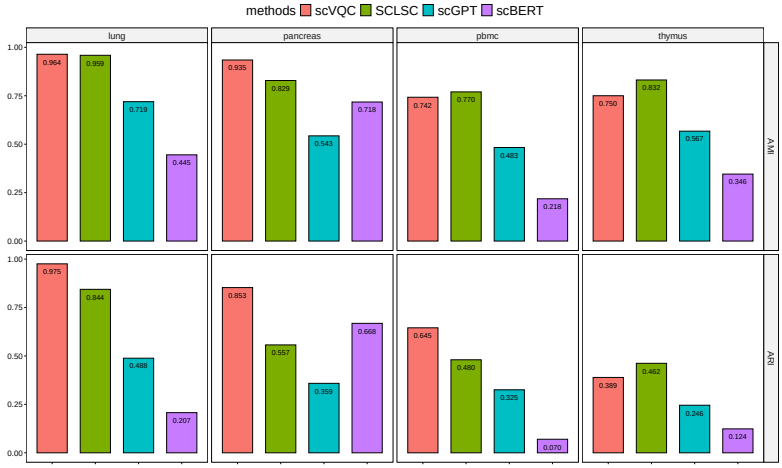
The clustering performance of supervised approaches, particularly scVQC, consistently surpasses FMs across all datasets, as evidenced by higher scores in both Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI)(Figure 4a). scVQC achieves the highest ARI scores in PBMC, lung, and pancreas, while SCLSC performs reasonably well but generally lags behind scVQC. In contrast, scGPT struggles with significantly lower ARI scores, reflecting weaker clustering accuracy. Similarly, for AMI, scVQC excels with the highest scores in PBMC, lung, and pancreas, while SCLSC slightly outperforms scVQC in the thymus dataset. Foundational models like scGPT perform poorly in AMI across all datasets. These results demonstrate that supervised approaches, particularly scVQC, are far more effective than foundational models scGPT for clustering cells and aligning with true cell type structures (Figure 4b).

Evaluation of running time

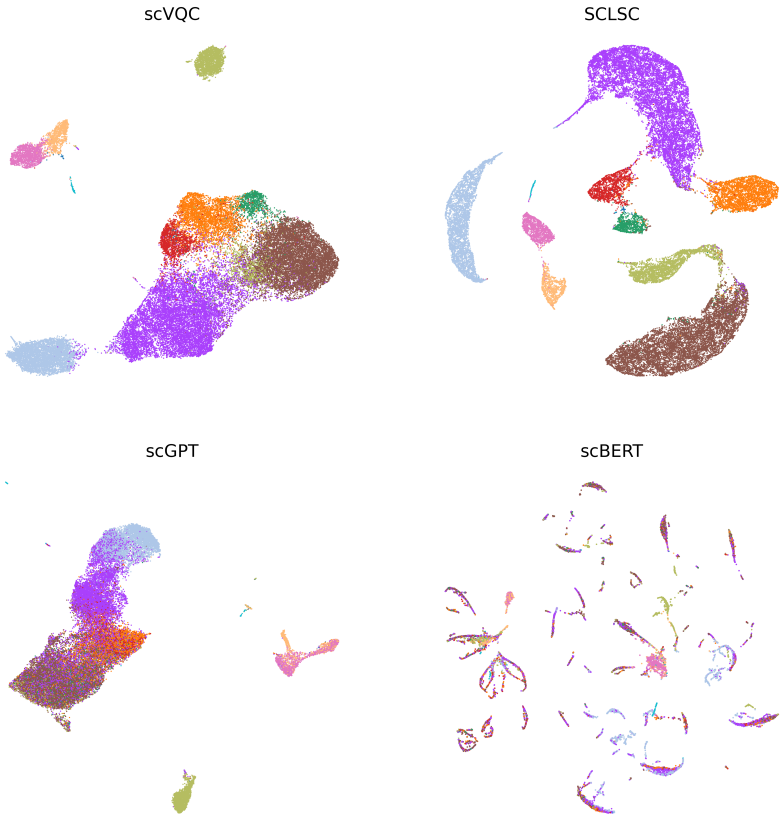
In this experiment, we measured the running time of scVQC and other methods on an NVIDIA V100 GPU. We used the thymus dataset, which contains over 200,000 cells, making it suitable for benchmarking. Running time is defined as the total time required for training the model and predicting cell types. In our study, scVQC is the fastest, completing the task in just 48.06 seconds. scGPT and SCLSC take significantly longer, with running times of 297.35 seconds and 377.51 seconds, respectively. scBERT has by far the longest running time, taking 70,493.27 seconds, making it considerably slower than the other models.

Split-VQ vs Standard-VQ

Split quantization has previously been used to address code perplexity problem in studies on protein subcellular localization [15] and chromatin accessibility [13]. In this work, we explored the effects of split quantization on cell classification performance, as well as its impact on learning cell representation and clustering (Table 1). In terms of accuracy, Split-VQ generally performs better than Standard-VQ, achieving the highest accuracy across all datasets, although the Macro-F1 score results are more mixed. However,



(a)



(b)

Fig. 4. (a) The clustering performance of the scVQC, SCLSC, and scGPT. Supervised approaches, consistently outperform FMs on every dataset, as reflected by higher Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI) scores. **(b)** The UMAP visualization of PBMC cells in the embedding space learned by the benchmarking methods, with colors representing different cell types. The supervised approaches, SCLSC and scVQC, generate embeddings where cells of the same type are more tightly clustered and distinctly separated from other cell types, compared to the foundational models, scGPT and scBERT.

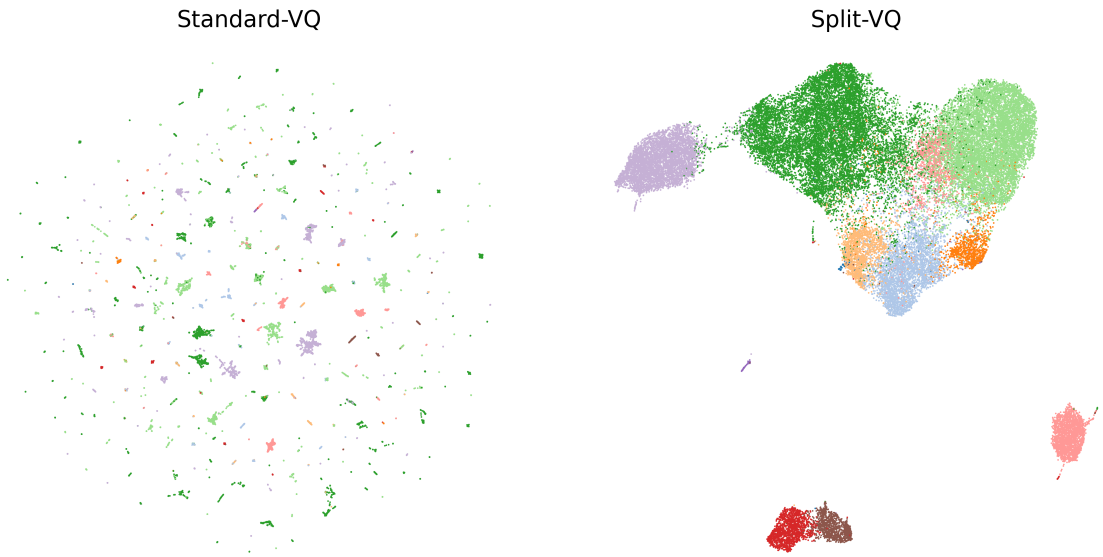


Fig. 5. The UMAP representation of Split-VQ and Standard-VQ. The UMAP representation of Split-VQ and Standard-VQ shows that Split-VQ has a greater ability to cluster cells compared to Standard-VQ. In contrast, the cell representation in Standard-VQ appears more dispersed than that of Split-VQ.

Table 1. Comparison of Split-VQ vs Standard-VQ

Methods	metrics	pbmc	lung	pancreas	thymus
Split-VQ	accuracy	**0.8250**	**0.9248**	**0.9831**	**0.9260**
Standard-VQ	accuracy	0.8118	0.9230	0.9760	0.9220
Split-VQ	Macro-f1	**0.7132**	0.9199	0.9181	**0.8360**
Standard-VQ	Macro-f1	0.6977	**0.9222**	**0.9341**	0.8354
Split-VQ	ARI	**0.7856**	**0.9823**	**0.6970**	**0.3950**
Standard-VQ	ARI	0.0862	0.2585	0.2441	0.0966
Split-VQ	AMI	**0.7846**	**0.9692**	**0.8772**	**0.7580**
Standard-VQ	AMI	0.4968	0.7165	0.6487	0.5699

when it comes to learning cell representations, as indicated by the clustering metrics ARI and AMI, Split-VQ consistently outperforms Standard-VQ across all datasets. These findings suggest that Split-VQ is notably more effective at clustering cells in alignment with true cell types (Figure 5).

Feature spectrum

The feature spectrum is a novel representation of cell types, created by applying the TF-IDF transformation to the utilization of codes within each cell type. Drawing an analogy to natural language processing, the codes in the codebooks are analogous to words, cell types correspond to documents, and the dataset represents a corpus or collection of documents. In this work, we applied the feature spectrum to identify characteristics of cell types in the PBMC dataset.

Figure 6 illustrates a distinct set of specific features that are either enriched or depleted within the feature spectrum of each cell type in the pancreas dataset. The cell-type-specific feature spectrum, derived from discrete cell embeddings of the single-cell RNAseq profiles, offers an interpretable method for visualizing the overarching and detailed patterns of cells within a specific cell type, providing a quantitative framework to better understand cell heterogeneity.

Discussion

Large foundation models (FMs), pre-trained on extensive single-cell datasets, have recently achieved significant progress in the field of single-cell analysis [2, 3, 4]. Prominent examples of these cutting-edge models include scGPT [3] and scBERT [2], which are widely utilized and studied in single-cell research and comparative analyses. The pre-training of these FMs, followed by fine-tuning on specific downstream tasks, enables the use of much smaller, application-specific datasets compared to training models from scratch. This sample efficiency is particularly advantageous in the biological domain, where clinical data or experimentally labeled datasets are often challenging and expensive to obtain, resulting in limited datasets for downstream applications.

Despite the significant advancements of foundation models (FMs) in single-cell analysis, several limitations and criticisms remain. For instance, Boiarsky et al. demonstrated that a simple regression model can outperform FMs in fine-tuning tasks for cell-type annotation, depending on the specific dataset [20]. Similarly, Kedzierska et al. showed that FMs can underperform in clustering tasks compared to simpler neural network methods [7]. These findings are consistent with our

results, where supervised neural network approaches often achieved competitive or even superior performance compared to FMs.

A key factor contributing to these observations is the potential overfitting or inefficiency in fine-tuning FMs when the data distributions of pre-training and downstream tasks are not closely aligned [21]. FMs typically have a large number of trainable parameters. For example in our study, the total number of learnable parameters is approximately 6.5 million for scBERT and 51 million for scGPT. In comparison, SCLSC and scVQC have 1 million and 3 million parameters, respectively. Overfitting occurs when these models become excessively complex and large, causing them to memorize the training data rather than learning meaningful patterns. This results in poor generalization performance on unseen data. Supervised methods, specifically designed for labeled datasets, excel at capturing dataset-specific features and relationships, offering a tailored approach to the analysis.

Another challenge for FMs is their reliance on large-scale, high-quality pre-training datasets. Any biases or noise in these datasets can propagate through the models, reducing their generalizability to niche or highly specific single-cell datasets [22]. In contrast, supervised methods are task- and tissue-specific, enabling fine-tuned optimization that minimizes the impact of pre-training biases. This targeted learning approach allows supervised models to more effectively capture biologically meaningful structures and distinctions between cell types.

On the other hand, while FMs are designed to generalize across diverse datasets and tasks, this broad generalization can dilute their ability to fine-tune embeddings for specific datasets or tissues.

Additionally, supervised models generally demand less computational effort compared to foundation models, which often involve a vast number of parameters. This reduced computational overhead not only makes supervised models competitive in terms of performance but also more accessible to researchers with limited computational resources. Our results further demonstrate that supervised approaches can outperform foundation models in terms of speed, sometimes by a significant margin.

In this study, we introduced scVQC, a cell type annotation framework that employs a split-Vector-Quantizer (split-VQ) to learn discrete embeddings of cells in a lower-dimensional space. Using codebooks generated from split-VQ, we can construct a novel representation of cells called the feature spectrum. The feature spectrum for each cell type is derived by applying the TF-IDF transformation to the utilization of codes within that cell type, analogous to identifying key words in documents. This method highlights features that are enriched or depleted for specific cell types, offering a powerful way to visualize overarching and detailed patterns within cell populations. Moreover, it provides a systematic approach to summarize complex single-cell RNA-seq data, potentially enabling the discovery of unique molecular signatures, rare cell types, and variations within heterogeneous populations, such as immune cells in PBMC datasets. Additionally, the feature spectrum serves as a bridge between computational analysis and biological interpretation, supporting hypothesis generation and deepening biological insights. Previous studies have applied the feature spectrum to characterize protein subcellular localization [15] and the epigenomic profiles of single cells [13]. However, to the best of the authors' knowledge, this study is the first to utilize the feature spectrum in the analysis of single-cell transcriptomic datasets.

Conclusion

In this study, we compared the effectiveness of foundation models (FMs) and supervised neural network approaches for the task of cell type prediction. Our findings showed that supervised approaches achieved performance comparable to FMs while requiring significantly less time and computational resources. Additionally, we introduced scVQC, a classifier based on split-vector quantization, which effectively learns cell representations for classification tasks. Moreover, scVQC can generate the cell type feature spectrum, a novel characterization of cell types derived from TF-IDF transformations of cell gene profiles quantized using the split-VQ method.

Competing interests

No competing interest is declared.

Code availability

The source code for scVQC is accessible at `https://github.com/yusri-dh/scVQC`

Data availability

We used 4 human single cell datasets in our study which can be accessed as follows:

- PBMC 68k [17]: `https://www.10xgenomics.com/resources/datasets/fresh-68-k-pbm-cs-donor-a-1-standard-1-1-0`
- pancreas [18]: `https://figshare.com/ndownloader/files/22891151`
- lung [18]: `https://figshare.com/ndownloader/files/24539942`
- thymus [19]: `https://zenodo.org/record/5500511`

All datasets were accessed and downloaded in June 2024.

Author contributions statement

Y.D.H. was responsible for the the data curation, analyses, and visualization, and writing the original draft of the manuscript. Y.Z. contributed to study conceptualization, supervision, and manuscript editing. S.I. managed funding acquisition, project administration, supervision, and manuscript editing. All authors reviewed the manuscript.

Funding

This study is supported by internal laboratory funding.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions.

References

1. Zoe A. Clarke, Tallulah S. Andrews, Jawairia Atif, Delaram Pouyabahar, Brendan T. Innes, Sonya A. MacParland, and

Gary D. Bader. Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. *Nature Protocols*, 16(6):2749–2764, May 2021.

2. Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, September 2022.
3. Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, February 2024.
4. Artur Szałata, Karin Hrovatin, Sören Becker, Alejandro Tejada-Lapuerta, Haotian Cui, Bo Wang, and Fabian J. Theis. Transformers in single-cell omics: a review and new perspectives. *Nature Methods*, 21(8):1430–1443, August 2024.
5. Hongkui Zeng. What is a cell type and how to define it? *Cell*, 185(15):2739–2755, July 2022.
6. Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2023.
7. Kasia Z. Kedzierska, Lorin Crawford, Ava P. Amini, and Alex X. Lu. Assessing the limits of zero-shot foundation models in single-cell biology, October 2023.
8. Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022.
9. Daniel P. Gomari, Annalise Schweickart, Leandro Cerchietti, Elisabeth Paietta, Hugo Fernandez, Hassen Al-Amin, Karsten Suhre, and Jan Krumsiek. Variational autoencoders learn transferrable representations of metabolomics data. *Communications Biology*, 5(1), June 2022.
10. Jiarui Ding and Aviv Regev. Deep generative model embedding of single-cell rna-seq profiles on hyperspheres and hyperbolic spaces. *Nature Communications*, 12(1), May 2021.
11. Bin Dai, Ziyu Wang, and David Wipf. The usual suspects? reassessing blame for vae posterior collapse, 2019.
12. Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2017.
13. Xuejian Cui, Xiaoyang Chen, Zhen Li, Zijing Gao, Shengquan Chen, and Rui Jiang. Discrete latent embedding of single-cell chromatin accessibility sequencing data for uncovering cell heterogeneity. *Nature Computational Science*, 4(5):346–359, May 2024.
14. Yusri Dwi Heryanto, Yao-zhong Zhang, and Seiya Imoto. Predicting cell types with supervised contrastive learning on cells and their types. *Scientific Reports*, 14(1), January 2024.
15. Hirofumi Kobayashi, Keith C. Cheveralls, Manuel D. Leonetti, and Loic A. Royer. Self-supervised deep learning encodes high-resolution features of protein subcellular localization. *Nature Methods*, 19(8):995–1003, July 2022.
16. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.
17. Grace X. Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1), January 2017.
18. Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, December 2021.
19. Jong-Eun Park, Rachel A. Botting, Cecilia Domínguez Conde, Dorin-Mirel Popescu, Marieke Lavaert, Daniel J. Kunz, Issac Goh, Emily Stephenson, Roberta Ragazzini, Elizabeth Tuck, Anna Wilbrey-Clark, Kenny Roberts, Veronika R. Kedlian, John R. Ferdinand, Xiaoling He, Simone Webb, Daniel Maunder, Niels Vandamme, Krishnaa T. Mahbubani, Krzysztof Polanski, Lira Mamanova, Liam Bolt, David Crossland, Fabrizio de Rita, Andrew Fuller, Andrew Filby, Gary Reynolds, David Dixon, Kourosh Saeb-Parsy, Steven Lisgo, Deborah Henderson, Roser Vento-Tormo, Omer A. Bayraktar, Roger A. Barker, Kerstin B. Meyer, Yvan Saeys, Paola Bonfanti, Sam Behjati, Menna R. Clatworthy, Tom Taghon, Muzlifah Haniffa, and Sarah A. Teichmann. A cell atlas of human thymic development defines t cell repertoire formation. *Science*, 367(6480), February 2020.
20. Rebecca Boiarsky, Nalini Singh, Alejandro Buendia, Gad Getz, and David Sontag. A deep dive into single-cell rna sequencing foundation models, October 2023.
21. Dusan Varis and Ondřej Bojar. Sequence length is a domain: Length-based overfitting in transformer models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.
22. Angelina Wang and Olga Russakovsky. Overwriting pretrained bias with finetuning data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3957–3968, October 2023.

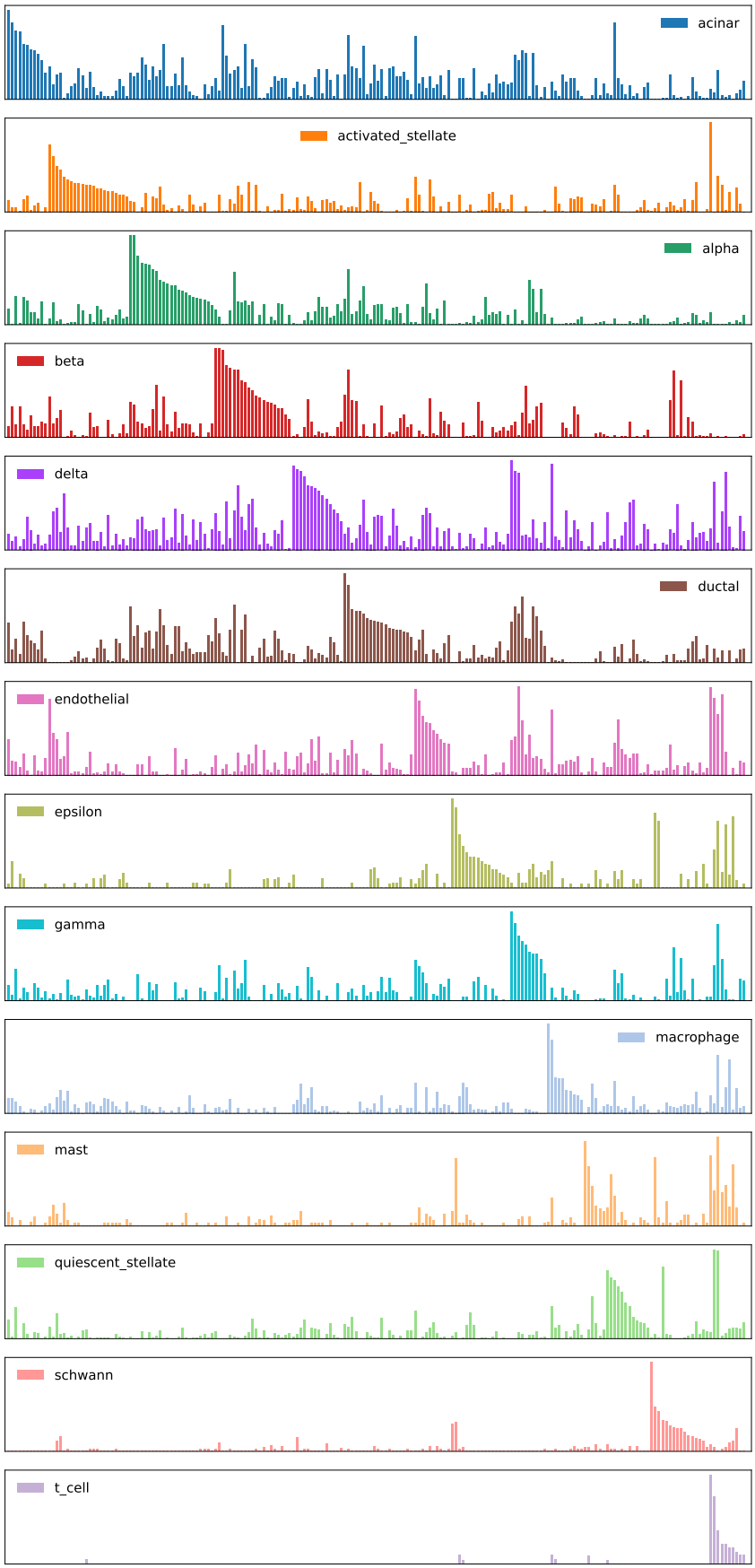


Fig. 6. The cell type feature spectrum. The feature spectrum of cell types in the pancreas dataset, derived from the TF-IDF transformation of code frequency utilization, revealed distinct characteristics for each cell type.