scMulan: a multitask generative pre-trained language model for single-cell analysis

Haiyang Bian[1,#], Yixin Chen[1,#], Xiaomin Dong[1,#], Chen Li[1], Minsheng Hao[1], Sijie Chen[1], Jinyi Hu[2], Maosong Sun[2], Lei Wei[1,*], and Xuegong Zhang[1,3,*]

[1] MOE Key Laboratory of Bioinformatics and Bioinformatics Division of BNRIST, Department of Automation, Tsinghua University, Beijing, China

[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China

[3] Center for Synthetic and Systems Biology, School of Life Sciences and School of Medicine, Tsinghua University, Beijing, China

[#] These authors contribute equally.

[*] To whom correspondence should be addressed. Email: zhangxg@tsinghua.edu.cn, weilei92@tsinghua.edu.cn

Abstract

Gene expression could be perceived as a form of cell language, with underlying regulatory mechanisms akin to biological grammar. Decoding this "language" is critical in understanding cellular functions and behaviors, but presents significant challenges. Several works have attempted to learn the biological language by pre-training large foundation models based on single-cell transcriptomic data, inspired by the success of large language models in natural language processing. In this study, we further enrich the pre-training paradigm by integrating an abundance of metadata and a multiplicity of pre-training tasks, and obtain scMulan, a multitask generative pre-trained language model tailored for single-cell analysis. We represent a cell as a structured cell sentence (c-sentence) by encoding its gene expression, metadata terms, and target tasks as words of tuples, each consisting of entities and their corresponding values. We construct a unified generative framework to model the cell language on c-sentence and design three pre-training tasks to bridge the microscopic and macroscopic information within the c-sentences. We pre-train scMulan on 10 million single-cell transcriptomic data and their corresponding metadata, with 368 million parameters. As a single model, scMulan can accomplish tasks zero-shot for cell type annotation, batch integration, and conditional cell generation, guided by different task prompts. Also, scMulan is ready to be expanded for novel tasks through fine-tuning. We have evaluated the effectiveness of scMulan on multiple downstream tasks. As a foundation model, scMulan is pre-trained to capture both the microscopic regulations and macroscopic patterns of gene expression, positioning it as a multifunctional and easily expandable tool for comprehensive single-cell analysis.

1 Introduction

With the advancement of high-throughput sequencing technologies, there has been an exponential increase in the volume of available single-cell transcriptomics data. This wealth of data is pivotal for biological analyses, such as the identification of cell types, deduction of gene relations, discernment of regulatory factors, and exploration of developmental processes. Based on these data, numerous deep-learning methods have been developed specifically for various downstream tasks. However, the task-specific design hinders the models from harnessing a comprehensive understanding of biological mechanisms from massive data and thus limits the performance of these methods.

Recently, a trend has emerged that well-designed task-specific models are being surpassed by foundation models with transformer architectures [1–4], also known as large language models (LLMs), in various tasks within the natural language processing (NLP) domain [4–7]. These models benefit from extensive pre-training on vast corpus, showcasing their prowess when fine-tuned for specific tasks or applied in a zero-shot manner. Drawing inspiration from this trend, the single-cell field has seen the advent of pre-trained foundational models including scBERT [8], GeneFormer [9], scGPT [10], and our early work scFoundation [11]. These models demonstrate the adaptability of foundational models, improving performance on downstream tasks such as cell type annotation and batch correction through fine-tuning. Yet, their zero-shot capabilities remain relatively underdeveloped [12]. These models learn gene expression profiles through masked language modeling, i.e. predicting masked gene expressions, and then work as an encoder to obtain embeddings of cells for downstream analysis with fine-tuning. Moreover, they primarily focused on learning the microscopic information of gene regulations but underleveraged or even ignored the vast macroscopic information available in single-cell data in the pre-training stage. There needs a foundation model that can take full advantage of both microscopic and macroscopic information and be able to deal with various tasks zero-shot.

In light of this, we introduce scMulan, a multitask generative pre-trained language model for single-cell analysis, aiming to fully exploit single-cell transcriptomic data and abundant metadata. We propose a formulation of cell language that transforms gene expressions and metadata terms into cell sentences (c-sentences), and we model the cell language through a generative training process. We design multiple pre-training tasks to bridge the microscopic and macroscopic information within the data, and we integrate these tasks within the unified generative framework based on cell language modeling, which is more convenient for data usage and further application. What's more, we define several natural-language-based terms as prompts for the cell language to guide the model aware of which task it is expected to perform. In this way, scMulan can generate not only genes with expressions but also metadata and other terms according to the tasks. After pre-training, scMulan can accomplish tasks like cell type annotation, metadata prediction, and multi-conditional data generation, and is also ready to be expanded and adapted for novel tasks through fine-tuning. We trained scMulan with more than 10 million single-cell data and evaluated its effectiveness on multiple downstream tasks zero-shot and through fine-tuning. scMulan is expected to serve as a unified model for various single-cell analysis tasks to empower biological research.

2 Methods

The success of generative models like the GPT series [6,7], T5 [8], and the Llama series [3,4] — which either implicitly or explicitly incorporate multi-tasks into their pre-training — suggests that this multi-task approach can enhance the model's contextual understanding and bring zero-shot capability. Besides, they modeled natural language through conditional generation, which is also employed for accomplishing all the tasks.

We design scMulan as a multitask generative pre-trained language model, aimed at achieving zero-shot capability for single-cell biology. scMulan is pre-trained on multiple tasks derived from a large transcriptomics dataset, leveraging the abundance of metadata but without the need for additional labeling. Furthermore, scMulan reframes tasks in single-cell analysis as generative tasks through cell language modeling, ensuring efficiency and scalability in pre-training and fine-tuning. We pioneer the use of task prompts to perform corresponding tasks in

single-cell analysis with scMulan, enabling scMulan to benefit from all the pre-training tasks together and perform multiple zero-shot tasks as a foundation model (Figure 1. a).

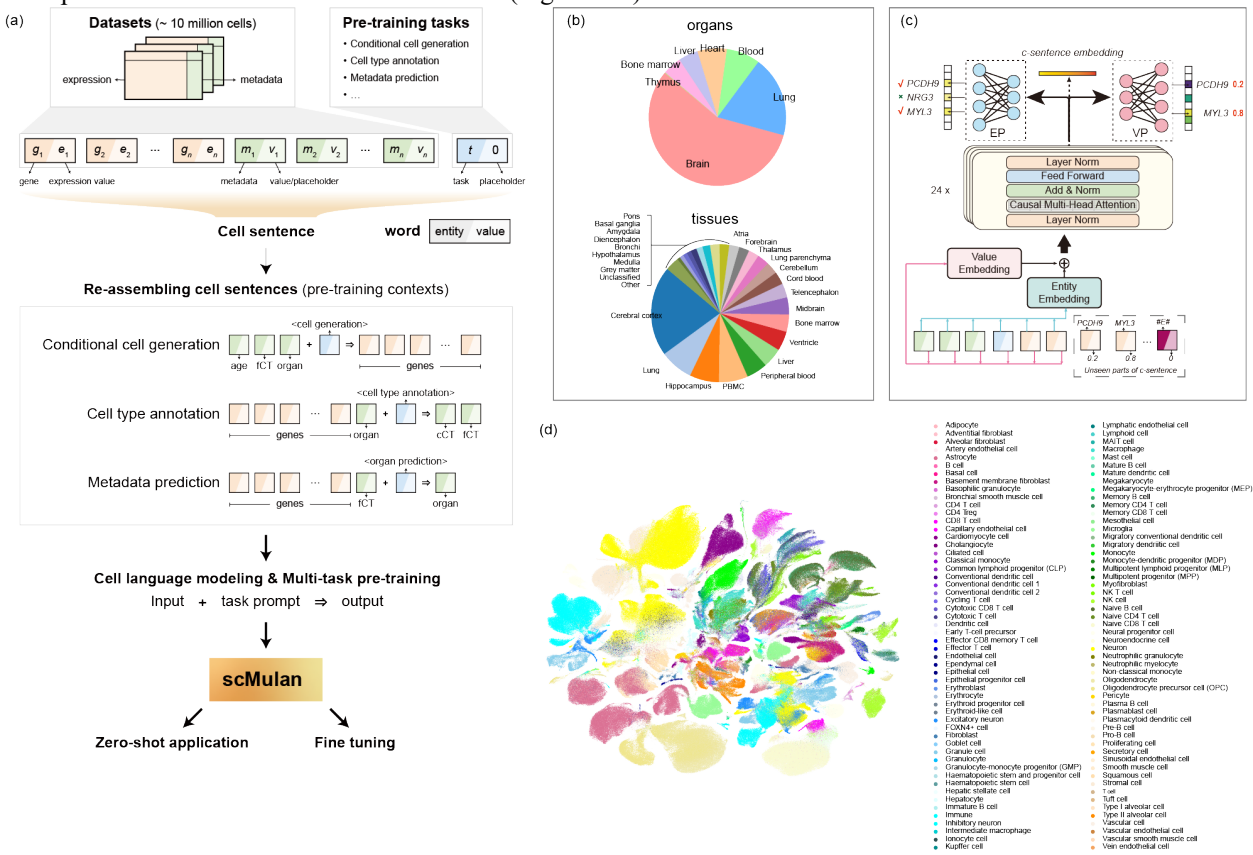

Figure 1. The overview of scMulan. (a) The workflow of scMulan. We use cell sentences to represent scRNA-seq data into dense sentences and assemble the expressed genes and metadata to form multiple pre-training tasks. scMulan could conduct multiple tasks guided by predefined prompts by fine-tuning or in a zero-shot manner. (b) The proportion of organs and tissues in the pre-training dataset hECA-10M. (c) The detailed architecture of scMulan. (d) UMAP visualization of the cell embeddings from scMulan on a 10% validation set of hECA-10M, colored by all cell types. cCT=coarse-grained cell type, fCT=fine-grained cell type, EP=Entity prediction head, and VP=Value prediction head.

2.1 hECA-10M dataset for pretraining

We selected a subset from hECA [13] for training scMulan, named hECA-10M. The hECA-10M data encompass more than 10 million high-quality single-cell transcriptome data from vital human organs or tissues including heart, brain, lung, liver, bone marrow, blood, and thymus (Figure 1. b). Each cell in hECA-10M is profiled with 42,117 gene expressions, and the accompanying metadata record attributes including organ source, specific region within the organ, donor age, donor gender, sequencing technology, and cell type. This comprehensive dataset allows us to design and compile metadata and gene expression profiles to establish multiple pretraining tasks, without any supervision and extra labeling.

Data preprocessing details for this dataset and other used datasets are described in supplementary methods.

2.2 c-sentence transferring

We conceptualized the information profile of each cell as a 'cell sentence' (c-sentence), comprising both the symbol and expression value of all expressed genes as well as the cell's metadata terms (Figure 1. a). Here, the gene content articulates the cell's microscopic characteristics, while the metadata reflects, to some extent, the macroscopic context or environment.

A notable distinction between c-sentences in single-cell biology and sentences in natural language is the

explicit association of expression values with genes. Within this framework, we define a 'word-of-gene' (W_G) in the c-sentence as a pairing of a gene and its corresponding expression level, which we term 'entity' and 'value', respectively. For instance, a gene CD3D, with an expression level of 0.2, would be represented as a W_G (*CD3D*, 0.2). We also introduce additional terminologies for other categories:

'word-of-metadata' ($\boldsymbol{W_M}$) , with entities being terms from the metadata information;

'word-of-task' ($\boldsymbol{W_T}$) , indicating the specific task associated with the c-sentence;

'word-of-special-tokens' ($\boldsymbol{W_S}$), where entities are special terms, like '#E#' signaling the end of a c-sentence.

For these categories, we assign a value of zero, given that they do not possess an expression level as genes do. We emphasize that the entities across all these word categories derive from natural language terms, such as 'Heart', 'T cell', or 'CD3D'. Together, these are collectively referred to as cell words in our study.

Here we provide an example of a c-sentence of a cardiomyocyte from the heart,

$$(Heart, 0), (Cardiomyocyte\ cell, 0), (A2M, 0.2), (ZNF385B, 0.6), \dots, (CD83, 0.1), (E, 0) \quad (1)$$

2.3 Cell language modeling

Cell language modeling is the core of our approach. It draws inspiration from the natural language modeling of Large Language Models (LLMs) but diverges from the sequential dependency that characterizes natural language. Traditional natural language models optimize for the joint probability distribution by multiplying conditional probabilities in sequences:

$$P(x) = \prod_{m=1}^{n} p(s_m | s_1, s_2, \dots, s_{m-1}) \quad (2)$$

for a set of examples$(x_1, x_2, \dots, x_n)$, with each x comprising variable-length sequences of symbols $(s_1, s_2, \dots, s_m)$. These models aim to maximize log-likelihood by estimating $p(w_i | w_1, w_2, \dots, w_{i-1})$, thereby predicting the next symbol given the previous ones.

In contrast, in the context of single-cell c-sentences, the order of genes is nonsequential and bears no intrinsic significance. Our objective is not to predict the next gene but to predict the set of all potential subsequent genes (entities). Therefore, we reformulate our goal to estimate $P(\{E - \{e_1, e_2, \dots, e_i\}\} | e_1, e_2, \dots, e_i)$, where E represents the set of all entities in a c-sentence and $\{E - \{e_1, e_2, \dots, e_i\}\}$ is the set of entities not yet conditioned upon. Moreover, we take values, especially gene expression values, into account by conditioning on both the entities and their values. The objective thus becomes to minimize the negative log-likelihood of $P(\{E - \{e_1, e_2, \dots, e_i\}\} | e_1, e_2, \dots, e_i; v_1, v_2, \dots, v_i)$, where v_i is the value of entity e_i. To make efficient use of the self-attention computation, we further minimize the summation of negative log-likelihood at each position of c-sentence, $\sum_i -logP(\{E - \{e_1, e_2, \dots, e_i\}\} | e_1, e_2, \dots, e_i; v_1, v_2, \dots, v_i)$. It differs from the masked language modeling approaches that models expressions for given genes, which is used in all the previous studies including scGPT [10], GeneFormer [9], and scFoundation [11]. Additionally, we employ regression to predict the expression values of the unobserved genes given the observed entities and their values.

To ensure the model's insensitivity to the order of input and to increase its robustness, gene entities within c-sentences are randomly shuffled during training.

2.4 Pre-training tasks

Based on the cell language modeling schema, we can formulate different tasks as a unified generation task with the help of task words in c-sentences. We designed a set of tasks for the pre-training phase to equip scMulan with a comprehension of the relational dynamics of gene expressions and patterns, particularly concerning diverse types of metadata such as cell type, spatial, temporal, and sexual information. To extensively leverage both metadata and gene expression information, we established three primary tasks: conditional cell generation, hierarchical cell type annotation, and metadata prediction.

Conditional cell generation Positioned as a foundational task, conditional cell generation involves the generation of genes and their corresponding expression values based on given conditions. These conditions may encompass various factors including the tissue name, cell type, donor age, sequence technology, given gene entities and their expression values, or any joint combination thereof. This task necessitates that our model grasps the intricate relationships interlinking genes and the metadata conditions. In the training process, scMulan is set to condition on metadata terms only.

Hierarchical cell type annotation This task is the generation of dual hierarchical levels of cell type annotations for a given cell, which includes both fine-grained and coarse-grained cell types. The aim is to steer the model towards understanding the hierarchy inherent to cell types, and their connectivity to gene expressions, and thereby gain an enhanced perspective on the concept of homogeneity and heterogeneity across cell types.

Other metadata prediction In this study, we demonstrate this task with organ region prediction. This classification task requires the model to determine the originating organ/tissue and specific region of a given input cell. It's a critical function that instills the spatial-related expression variance of cells in the model. As the information in metadata encompasses important sources of cell diversity, any further covariant of the cell diversity could be added in the same way.

2.5 c-sentence formatting for input and output

In pursuit of pre-training a single model across a diverse spectrum of tasks described above, we've adapted these tasks into a generation format-that is, a task where the model is fed part of a c-sentence for context or conditioning and is then asked to produce some output cell words including entities and values of genes or metadata terms. This approach ensures a uniform objective for both pre-training and future fine-tuning. To specify which task the model should perform, we add pre-defined task prompts as $W_T s$ in the input c-sentence, where their entity indicates the targeted tasks.

For instance, if the model is instructed to generate a 'Cardiomyocyte cell' of the heart, with expression values of genes g_1 and g_2 set at 0.8 and 0.6 respectively, the input would be a cell sentence like "(Cardiomyocyte cell, 0.0), (Heart, 0.0), (g_1, 0.8), (g_2, 0.6), (<cell generation>,0.0)". The model would be trained to sequentially output "(g_3, 0.3), (g_9, 0.5), …, (g_n, 0.2), (#E#, 0.0)", a cell sentence with the rest W_Gs of this ''Cardiomyocyte cell'. Herein, the W_T '<cell generation>' operates as a prompt signifying the cell generation task, whereas '#E#' is employed as a distinctive entity symbolizing the conclusion of the generated cell sentence.

For the cell type annotation task, the model provides both coarse-grained and fine-grained cell type annotations, instructed by a cell sentence that includes the task prompt '<cell type annotation>'. For example, when the input cell sentence is "(CD3D, 0.7), (CD4, 0.6), …, (CCR7,0.3), (<cell type annotation>,0.0)", the model is expected to output "(T cell, 0.0), (CD4 T cell, 0.0), (#E#, 0.0)".

By formatting into c-sentences, the profiles of cells could be equally used for different tasks in the same framework.

2.6 scMulan Model

2.6.1 Architecture

We build scMulan upon a Transformer decoder-only architecture [1], inspired by the OpenAI GPT models, with crucial adaptations to cater to the distinct features of single-cell data. These modifications span three primary areas including dual embedding layers, value prediction head, and flexible generation prediction (Figure 1).

The dual embedding layers are used to project input entities and values as high-dimensional embeddings with the same dimensions, through an entity embedding (E_e) layer and a value embedding (E_v) layer, respectively. (Figure 1. c) Then the entity embedding and value embedding are summed, combining the gene's identity and its expression level, which then serves as input to the Transformer layers.

To generate outputs with values, especially gene expression values, we feature a unique value prediction head

(VP) for scMulan besides the original entity prediction head (EP). (Figure 1. c) The value prediction head interprets the logits from the Transformer's final layer to predict the expression levels for genes in the W_Gs. The dual embedding layers and value prediction head enable scMulan to handle gene expression levels at both the input and the output.

What's more, recognizing that genes in single-cell data intrinsically lack a definitive sequence, we remove the traditional position-aware design in LLMs. For input entities, we remove positional encoding to keep the position-insensitivity mechanism of self-attention. For generating outputs, we set scMulan to predict all possible entities and values of a c-sentence, conditioned on the given input words at each time step in training. When inferencing, it generates one entity for each step sampled from the distributions of logits from the entity prediction head. Though the input sequence still relies on the position of a c-sentence, we could introduce more variability to reduce the impact of input order and enhance the robustness by random shuffling genes in c-sentences. We define the generation strategy as flexible generation.

The detailed model settings are described in the supplementary.

2.6.2 Pre-training

Given a single-cell dataset formulated as a c-sentence corpus mixed genes, metadata terms, and tasks, we could define a vocabulary of entities $E = \{e_1, \dots, e_n\}$. Our objective is to optimize both the cell language modeling and the value prediction objectives.

The cell language modeling objective is expressed as the negative log-likelihood as defined in section 2.3:

$$L_1 = -\sum_i logP(\{E - \{e_1, e_2, \dots, e_i\}\}|e_1, e_2, \dots, e_i; v_1, v_2, \dots, v_i;\ \theta) \tag{3}$$

where the conditional probability P is modeled by scMulan's entity prediction head parameterized by θ.

For value prediction, we employ a mean square error (MSE) objective:

$$L_2 = -\frac{1}{|E_{\text{unobs}}|}\sum_{e_j \in E_{\text{unobs}}} \left(v_j^{\text{true}} - v_j^{\text{pred}}(E_{obs}; V_{obs}; \varphi)\right)^2 \tag{4}$$

where v_j^{true} denotes the true value of the entity e_j, v_j^{pred} represents the predicted value from scMulan's value prediction head given the observed entities E_{obs} and their values V_{obs} with parameter φ, and $E_{\text{unobs}} = \{E - E_{obs}\}$ represents the set of unobserved entities.

The entity and value prediction heads, $head_e$ and $head_v$, respectively, leverage shared hidden embeddings derived from a multi-layer Transformer decoder:

$$\begin{gathered} h_0 = E_e(E_{obs}) + E_v(V_{obs}) \\ h_i = \text{transformer_block}(h_{i-1}) \quad \forall i \in [1, n] \\ P(E_{\text{unobs}}) = \text{head}_e(h_n) \\ V^{\text{pred}}(V_{\text{unobs}}) = \text{head}_v(h_n) \end{gathered} \tag{5}$$

where E_e and E_v are the entity and value embedding layers, respectively, n is the number of layers, and $head_e$ and $head_v$ are implemented as multi-layer perceptrons (MLPs).

We thus optimize the compound objective with a weighting factor λ:

$$L = L_1 + \lambda L_2 \tag{6}$$

The parameters of scMulan are trained via adaptive gradient algorithms [14].

2.7 Downstream experiments

Within the realm of NLP, zero-shot performance serves as a critical benchmark for assessing the fundamental language comprehension of large language models. Contrastingly, in the single-cell research arena, zero-shot

evaluations are seldom employed to evaluate the efficacy of large single-cell models. We designed experiments to evaluate the zero-shot capability of scMulan on cell type annotation and batch integration. Herein, the zero-shot setting is defined as directly using the foundation model on unseen datasets without further fine-tuning. We also examined the fine-tuning performance of scMulan for cell type identification in novel tissues. We evaluated the generated cells from scMulan given metadata conditions. Finally, we hope to study the decision-making mechanism of scMulan by applying explanation methods for LLMs.

The detailed description of the experiments and used datasets is listed in supplementary methods.

3 Results

scMulan could conduct several tasks as a foundation model prompted by different task entities. For cell type annotation, we use scMulan's generated fine-grained cell types as prediction, taking the input cell's gene expression (W_gs) followed by a task entity '<cell type annotation>' as the prompt. In cell generation, we convert organ-cell type pairs into c-sentences, appended with the task entity '<cell generation>' to direct scMulan to sequentially generate gene expressions. To generate cell embeddings, we extract cell representations from scMulan's final transformer layer. This process involved inputting a c-sentence composed of a cell's gene expression profile followed by the task entity '<cell type annotation>'. We additionally showcased the UMAP of cell embeddings for the validation set of hECA-10M (Figure 1. d), illustrating that these embeddings retain information on organ and cell type, hinting at scMulan's adeptness in learning biological insights.

3.1 scMulan achieved precise zero-shot cell type annotation across multiple organs

To assess the foundational capabilities of scMulan, we subjected it to a series of zero-shot downstream tasks and benchmarked its performance against existing methods. The initial downstream task was automatic cell type annotation, a pivotal task in scRNA-seq data analysis that could reflect a model's understanding of cell identities and their gene expression signatures.

We benchmarked scMulan against Celltypist [15], Geneformer [9] and scGPT [10]. Celltypist is a logistic regression-based method designed for cross-organ annotation with demonstrated excellence. Geneformer and scGPT are pretrained models for single cell data which enables cell type annotation through additional fine-tuning, and fine-tuned scGPT achieved SOTA performance. In this study, a Celltypist model was trained from scratch on the hECA-10M-sampled dataset which is a randomly sampled subset of hECA-10M (Supplementary methods). A Geneformer model (pre-trained on the Genecorpus-30M [9]), and a scGPT-whole-human model (pre-trained on the whole-human 33M dataset [10]) were fine-tuned on the hECA-10M-sampled with cell-type labels for the annotation task. scMulan is used as a pre-trained model with a task-specific prompt '<cell type annotation>' to directly annotate novel datasets

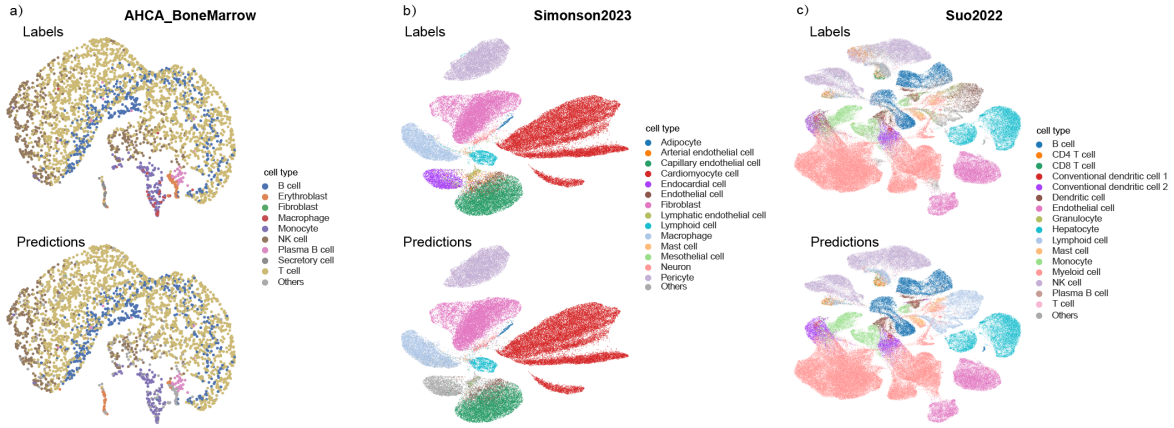

Figure 2. Cell type annotation on three datasets without fine-tuning. UMAP visualization depicting gene expression in cells from the (a) AHCA_BoneMarrow, (b) Simonson2023, and (c) Su2022 datasets, annotated with manually assigned cell types based on the original study and predicted labels from scMulan.

(Table S1).

We evaluated these models on cell type annotation across three novel datasets - AHCA_BoneMarrow [16], Simonson2023 [17], and Suo2022 [18] – that ranged from straightforward to complex, and are not included in hECA-10M. AHCA_BoneMarrow (Figure 2. a) has approximately 3,000 bone marrow cells from a single adult donor, which posed the least complexity. The Simonson2023 dataset (Figure 2. b) included 60,345 cells from 8 human left ventricle samples, presenting a more challenging scenario due to inter-individual variation. Su2022 (Figure 2. c) was a more diverse dataset with roughly 140,000 liver cells from 14 fetal donors at various developmental stages. Details of the datasets are in the supplementary methods. These datasets underwent meticulous label alignment where all annotations from the original studies were manually mapped onto the cell types in the hECA-10M(see the mapping results in Supplementary file 1). The evaluated metrics include accuracy, weighted precision, and weighted F1 score (Supplementary methods).

In all scenarios, scMulan delivered the most precise annotations, achieving a weighted F1 score of about 0.9, and consistently outperformed the other methods, particularly in the more challenging contexts (Table 1). A closer inspection of scMulan's predictions versus the original annotations revealed that scMulan accurately identified most cell types (Figure S1-S3.). These results underscore scMulan's generalization ability to efficiently annotate cell types on multiple organs in intricate scenarios, without the necessity for additional fine-tuning.

Table 1. Annotation metrics. Accuracy, weighted Precision, and weighted-F1 score for Celltypist, scGPT, Geneformer and scMulan applied on AHCA_BoneMarrow, Simonson2023, and Su2022 datasets. Acc=Accuracy, Prec=weighted-Precision, F1=weighted-F1 score.

	AHCA_BoneMarrow			Simonson2023			Su2022		
	Acc.	Prec.	F1.	Acc.	Prec.	F1.	Acc.	Prec.	F1.
scMulan	**0.917**	**0.961**	**0.937**	**0.927**	**0.947**	**0.934**	**0.873**	**0.963**	**0.894**
CellTypist	0.600	0.859	0.665	0.865	0.942	0.895	0.344	0.909	0.420
scGPT	0.686	0.928	0.740	0.877	0.938	0.898	0.792	0.878	0.830
Geneformer	0.793	0.798	0.796	0.588	0.939	0.635	0.742	0.911	0.760

We also observed that misannotations mainly occurred in cell types absent in the training corpus, such as the secretory cell in the bone marrow and the endocardial cell in the heart (Figure S1, S2.). It suggests potential for model enhancement through fine-tuning novel data, especially from noval organs not included in the pretraining corpus.

3.2 scMulan achieved enhanced performance in cell type annotation with fine-tuning

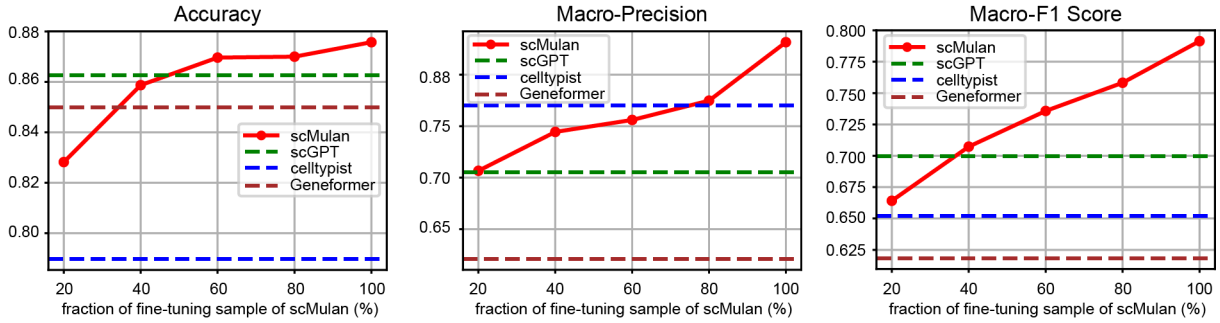

Figure 3. Cell type annotation on the intestine dataset with fine-tuning. We show the variation in annotation metrics (accuracy, macro-precision, and macro-F1 score) on the test set as the fraction of fine-tuning samples increases. The red solid line represents the metrics of scMulan, while the green, brown and blue dashed line corresponds to the scGPT, Geneformer and Celltypist model fine-tuned on all samples.

To evaluate how pre-trained scMulan could be enhanced through further fine-tuning, we conducted an experiment

focusing on the cell type annotation task for a new tissue, the intestine. We used 55,214 intestinal cells across 24 cell types of the Intestine_HCL_55k [19] dataset and randomly split them into training and testing sets with a ratio of 9:1. We incrementally fine-tuned scMulan with varying fractions of cells (ranging from 20% to 100%) from the training set. For comparison, we trained a Celltypist model and fine-tuned the scGPT-whole-human model on the entire training set as a baseline (Table S2). We evaluated the models on the testing set.

The fine-tuned scMulan model on just 40% of samples achieved competitive performance to the other models on 100% of training samples and surpassed it upon fine-tuning with 60% or more samples (Figure 3). Besides, the results indicated a consistent enhancement in performance corresponding to the increased fraction of cells used for fine-tuning the scMulan pre-trained model (Figure 3). The macro-F1 score escalated from 0.66 to nearly 0.80, whereas scMulan without fine-tuning attained only 0.25, demonstrating a significant boost from fine-tuning on new tissues (Figure S4). These intriguing findings underscore that scMulan benefits from an increased fine-tuning dataset and exhibits a better transferring ability than scGPT in finetuning efficiency.

3.3 scMulan enables fast and accurate integration across multiple batches without fine-tuning

In single-cell analysis, unwanted variances such as technical biases or individual differences within the research group are termed batch effects. Data integration aims to minimize these effects while preserving biological variability in data. To address this, we evaluated scMulan against established methods including BBKNN [20], Harmony [21], Seurat v3 [22], FASTMNN [23], scVI [24], scANVI [24], Geneformer without fine-tuning (Geneformer_zeroshot), fine-tuned scGPT (scGPT_finetune), and scGPT without fine-tuning (scGPT_zeroshot). Geneformer_zeroshot was trained on the Genecorpus-30M. Both scGPT_finetune and scGPT_zeroshot were pre-trained on the whole-human-33M dataset, with scGPT_finetune undergoing additional fine-tuning on the evaluated datasets for this task. Conversely, scMulan was applied zero-shot to obtain cell embeddings directly from the hidden layer. The datasets in the pre-training and fine-tuning process of the models are shown in Table S3.

We constructed two assessment scenarios. The first one, same as that in the scGPT paper, involved an immune cell dataset with 274,346 cells spanning 18 batches, including healthy donors and those with varying COVID-19 severity [25]. The second and less complex one, was selected from the benchmark of single-cell integration scIB [26] and involved the Lung dataset with 32,472 cells from 16 donors, various sequencing protocols and spatial locations. We utilized the benchmarking method in scIB [26] and calculated the average biological conservation

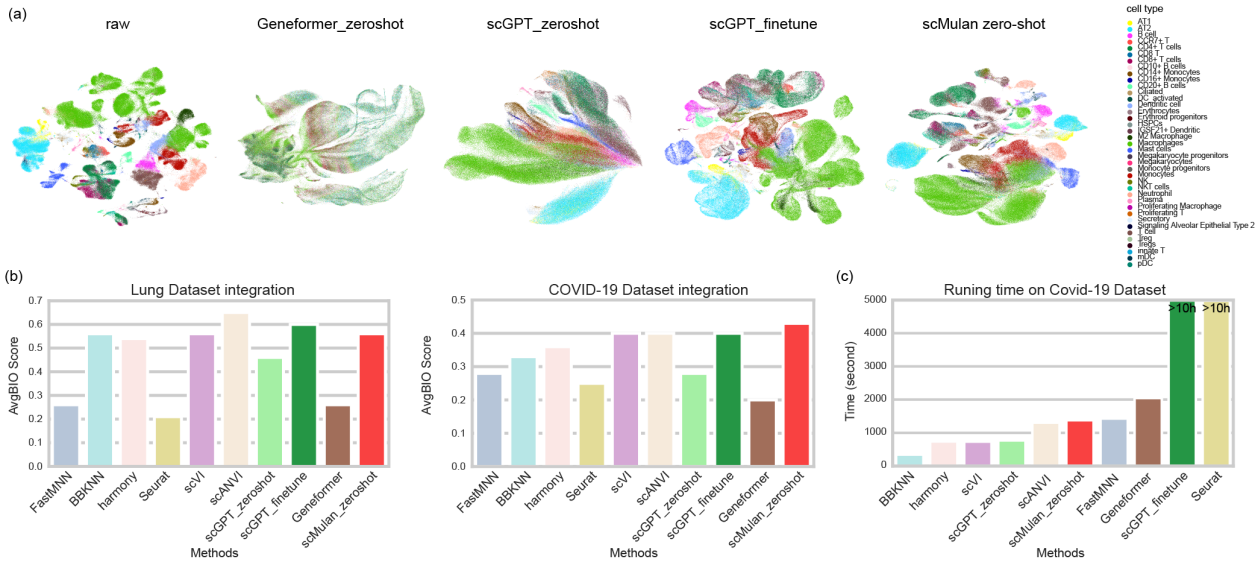

Figure 4. Data integration experiments in two scenarios. (a) UMAP visualization depicting gene expression in cells from the COVID dataset before (left) and after (right) different integration methods, annotated with cell types from the original study. (b) the integration metric (average biological preserving score) of the heart dataset integration task (left) and COVID dataset integration task (right). c) the execution time of different integration methods on the Covid-19 dataset.

score (AvgBIO) and the average batch correction score (AvgBATCH) used in scGPT paper for method comparison. AvgBIO assessed the consistency between the derived clusters and the ground truth cell type labels and AvgBATCH assessed the removal of the batch effects (Supplementary methods).

Integration results for the COVID-19 dataset showed that scMulan and the scGPT_finetune model succeeded in preserving biological variance, as cells of the same type clustered closely together, while Geneformer_zeroshot and scGPT_zeroshot showed less efficacy (Figure 4. a). And scMulan outperformed all the other methods in AvgBIO scores, followed by scGPT_finetune, scANVI and scVI (Figure 4. b). In AvgBATCH scores, scMulan achieved comparative performance with scGPT_finetuned,scANVI and scVI, and outperformance the other methods (Figure S5). In the Lung dataset analysis, scMulan_zeroshot, while surpassed by scGPT_finetune and scANVI in AvgBIO scores, maintained a performance advantage over other methods.

In both evaluating datasets, scMulan achieved competitive performance with scGPT_finetune and notably faster, while eliminating the extensive fine-tuning period necessary for scGPT, eliminating the need for the additional 10 hours of fine-tuning required by scGPT (Figure 4. c). Notably, scMulan significantly outperformed scGPT_zeroshot and Geneformer_zeroshot, showcasing a superior zero-shot representation of learning the biological contexts. These results highlight scMulan's capability for quick and precise batch integration across diverse batches, negating the need for fine-tuning.

3.4 scMulan enables conditional generation of single-cell expression profiles

As a generative model, scMulan is capable of generating single-cell expression profiles conditioned on any combination of metadata terms and genes. In this study, we showcase its ability to generate cells based on specific organ-cell type pair conditions. For each of six organs within the hECA-10M dataset, we created 3,000 conditions, each assigned to an organ-cell type pair as a prompt for scMulan, yielding 140 unique organ-cell type pairs. We then generated a corresponding set of 18,000 cells (hereafter 'generated data') based on these prompts. We sampled an equivalent number of real cells (hereafter 'real data') from hECA-10M, maintaining the same proportion of cell types, for comparative analysis. Details of the experiments are in supplementary methods.

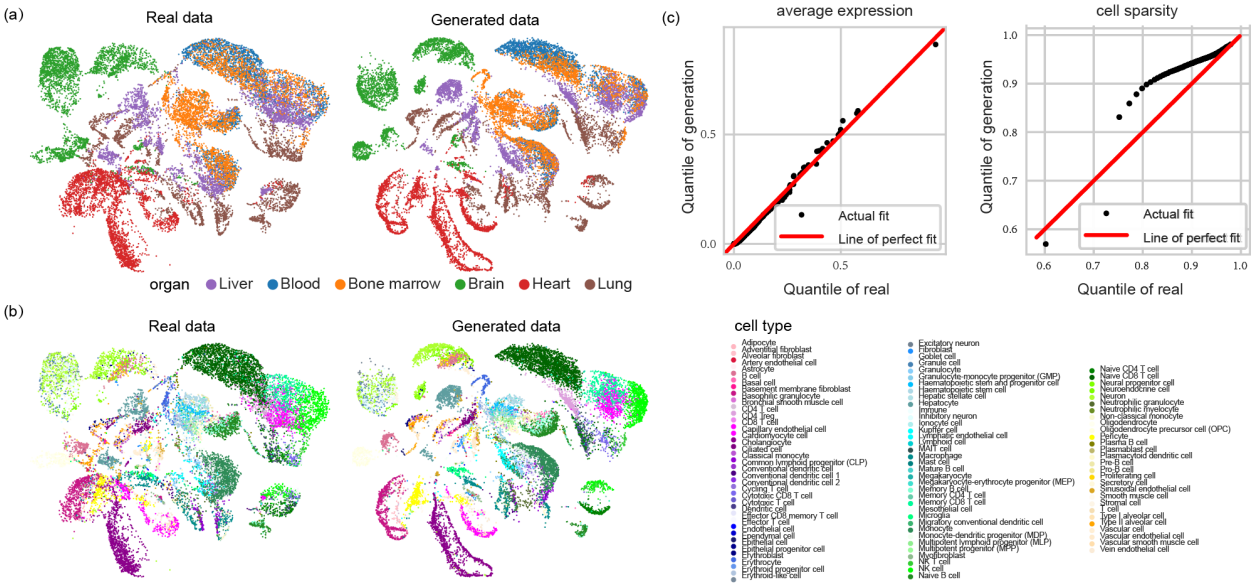

Figure 5. Conditional generation of cellular expression profiles and evaluation. (a) UMAP visualization depicting gene expression of the real cells (left) and generated cells (right), annotated with the tissue labels of the real data and the set tissue labels of the generated data. (b) UMAP visualization depicting gene expression of the real cells (left) and generated cells (right), annotated with the cell type label of the real data and the set cell type labels of the generated data. (c) The Q-Q plots compare the distributions of three key metrics: average gene expression level (left) and sparsity of the expression profiles (right), between the real data and the generated data.

By merging the generated data with the real data and mapping them into the same embedding space (without batch correction), the UMAP visualization (Figure 5. a, b) demonstrated that the generated cells closely mirrored their real-cell counterparts in terms of both organ and cell type specificity. This suggests a substantial similarity between the generated and real data while still preserving biological diversity. To quantitatively assess the comparison between the generated and real data sets, we analyzed the distribution of gene expression and cell sparsity (the ratio of non-expressed genes per cell) (Figure 5. c). The findings showed remarkable consistency in the average expression levels of all genes between the datasets. However, the generated data exhibited greater cell sparsity compared to the real data, a difference we attribute to scMulan's inclination to terminate the c-sentence prematurely, potentially influenced by the incorporation of noisy or irrelevant genes.

The conditional generation ability of scMulan can be further extended to support any included metadata such as donor information including age and gender. Such a feature has the potential to enrich datasets with rare cell types and create cells under novel condition combinations.

3.5 Explanaton on scMulan identified saliency genes for cell type annotation

Explaining LLM models is challenging and crucial [27], we attempted to examine the gene attributions for decision-making of scMulan when predicting cell types. We explored the interpretability of scMulan by examining the genes associated with cell type annotations, referred to as "saliency genes" (Supplementary methods). By analyzing the saliency genes (Figure S5), we identified several well-established marker genes, such as ACTA2 and MYH11 in smooth muscle cells, DCN in fibroblasts, NRXN3 in neuron cells, and CD79A in B cells. We also discovered numerous additional genes beyond the known markers. This result suggested that scMulan has, to some extent, learned biologically plausible expression patterns and may also have captured more complex expression patterns or uncovered new gene relationships.

4 Discussion

In summary, we developed scMulan, a foundational model tailored to facilitate single-cell analysis. scMulan stands out as the first foundational model capable of performing multiple tasks simultaneously without the need for additional fine-tuning or external layers. Our experiments validate scMulan's zero-shot capabilities in cell type annotation and batch integration, where it even surpasses models that have undergone fine-tuning. Moreover, fine-tuning experiments have showcased its excellent potential as a pre-trained starting point for downstream tasks.

scMulan currently is the largest foundation model in single-cell biology with over 350 million parameters. Its successful training and performance can be attributed to two key innovations. First, the c-sentence formulation, which effectively condenses the structured gene expression matrix into memory-efficient sentences, capitalizes on the sparsity of gene expressions. Second, the cell language modeling framework drastically enhances data utilization and computational efficiency by exploiting the parallel processing power inherent in self-attention mechanisms. The frameworks for c-sentence transferring and cell language modeling lay the groundwork for future efforts to train larger models within the domain of single-cell biology.

Through its multitask pre-training strategy, scMulan has learned to navigate both micro-level details and macro-level insights, which in turn enhances each of the tasks it performs. Shared metadata and gene expression patterns have proven to be crucial links in bolstering the model's foundational skills. For instance, in the conditional generation task, scMulan's ability to generate genes based on cell types contributes to the cell type annotation task by discerning gene regulation patterns and co-expression dynamics across varied cell types. The model's understanding of the interplay between cell type and gene expression is further evidenced by its use of learned cell representations for zero-shot batch integration.

Despite being pre-trained on the relatively modest hECA-10M dataset, scMulan has demonstrated superior performance. This success likely stems from the model’s multitask pre-training approach and its efficient data utilization, bolstering our confidence in scMulan's potential for enhanced zero-shot learning on even larger datasets.

Furthermore, due to its adeptness at learning efficient cell representations, scMulan has a high potential to perform in silico gene perturbation and simulate drug responses. These capabilities open up exciting avenues for future research and applications.

Avaliablity

The pretrained model of scMulan can be found at https://github.com/SuperBianC/scMulan. The hECA-10M dataset used for pre-traing scMulan is available through cloud storage link (https://cloud.tsinghua.edu.cn/d/0858c961251e4be2b4f0/).

Acknowledgments

The work is supported in part of National Natural Science Foundation of China (grants 62373210, 62250005, 61721003), and National Key R&D Program of China (grant 2021YFF1200900).

Reference

1. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30.

2. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, et al. Llama: Open and efficient foundation language models. ArXiv Prepr ArXiv230213971. 2023;

3. Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: Open foundation and fine-tuned chat models. ArXiv Prepr ArXiv230709288. 2023;

4. Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, et al. On the opportunities and risks of foundation models. ArXiv Prepr ArXiv210807258. 2021;

5. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI Blog. 2019;1:9.

6. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.

7. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21:5485–551.

8. Yang F, Wang W, Wang F, Fang Y, Tang D, Huang J, et al. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. Nat Mach Intell. 2022;4:852–66.

9. Theodoris CV, Xiao L, Chopra A, Chaffin MD, Al Sayed ZR, Hill MC, et al. Transfer learning enables predictions in network biology. Nature. 2023;618:616–24.

10. Cui H, Wang C, Maan H, Pang K, Luo F, Wang B. scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI [Internet]. bioRxiv; 2023 [cited 2023 Sep 22]. p. 2023.04.30.538439. Available from: https://www.biorxiv.org/content/10.1101/2023.04.30.538439v2

11. Hao M, Gong J, Zeng X, Liu C, Guo Y, Cheng X, et al. Large scale foundation model on single-cell transcriptomics. BioRxiv Prepr Serv Biol. 2023;2023–05.

12. Kedzierska KZ, Crawford L, Amini AP, Lu AX. Assessing the limits of zero-shot foundation models in single-cell biology [Internet]. Bioinformatics; 2023 Oct. Available from: http://biorxiv.org/lookup/doi/10.1101/2023.10.16.561085

13. Chen S, Luo Y, Gao H, Li F, Chen Y, Li J, et al. hECA: The cell-centric assembly of a cell atlas. iScience. 2022;25:104318.

14. Loshchilov I, Hutter F. Decoupled weight decay regularization. ArXiv Prepr ArXiv171105101. 2017;

15. Domínguez Conde C, Xu C, Jarvis L, Rainbow D, Wells S, Gomes T, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. Science. 2022;376:eabl5197.

16. He S, Wang L-H, Liu Y, Li Y-Q, Chen H-T, Xu J-H, et al. Single-cell transcriptome profiling of an adult human cell atlas of 15 major organs. Genome Biol. 2020;21:294.

17. Simonson B, Chaffin M, Hill MC, Atwa O, Guedira Y, Bhasin H, et al. Single-nucleus RNA sequencing in ischemic cardiomyopathy reveals common transcriptional profile underlying end-stage heart failure. Cell Rep. 2023;42:112086.

18. Suo C, Dann E, Goh I, Jardine L, Kleshchevnikov V, Park J-E, et al. Mapping the developing human immune system across organs. Science. 2022;eabo0510.

19. Construction of a human cell landscape at single-cell level | Nature [Internet]. [cited 2023 Nov 8]. Available from: https://www.nature.com/articles/s41586-020-2157-4

20. Polański K, Young MD, Miao Z, Meyer KB, Teichmann SA, Park J-E. BBKNN: fast batch alignment of single cell transcriptomes. Bioinforma Oxf Engl. 2020;36:964–5.

21. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, et al. Fast, sensitive and accurate integration of single-cell data with Harmony. Nat Methods. 2019;16:1289–96.

22. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, et al. Comprehensive integration of single-cell data. Cell. 2019;177:1888–902.

23. Zhang F, Wu Y, Tian W. A novel approach to remove the batch effect of single-cell data. Cell Discov. 2019;5:46.

24. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15:1053–8.

25. Lotfollahi M, Naghipourfar M, Luecken MD, Khajavi M, Büttner M, Wagenstetter M, et al. Mapping single-cell data to reference atlases by transfer learning. Nat Biotechnol. 2022;40:121–30.

26. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods. 2022;19:41–50.

27. Zhao H, Chen H, Yang F, Liu N, Deng H, Cai H, et al. Explainability for Large Language Models: A Survey [Internet]. arXiv; 2023 [cited 2023 Oct 14]. Available from: http://arxiv.org/abs/2309.01029