scELMo: **E**mbeddings from **L**anguage **Mo**dels are Good Learners for **S**ingle-**c**ell Data Analysis

Tianyu Liu[1,2], Tianqi Chen[2†], Wangjie Zheng[2†], Xiao Luo[3], Hongyu Zhao[1,2*]

[1]Interdepartmental Program in Computational Biology & Bioinformatics, Yale University, New Haven, 06511, CT, USA.
[2]Department of Biostatistics, Yale University, New Haven, 06511, CT, USA.
[3]Department of Computer Science, University of California, Los Angeles, Los Angeles, 90095, CA, USA.

*Corresponding author(s). E-mail(s): hongyu.zhao@yale.edu;
Contributing authors: tianyu.liu@yale.edu; tianqi.chen@yale.edu; wangjie.zheng@yale.edu; xiaoluo@cs.ucla.edu;
†These authors contributed equally to this work.

Abstract

Different from a recent approach to building the Foundation Models (FMs) for analyzing single-cell data based on the pre-training and fine-tuning framework, in this manuscript, we extend the concept from GenePT [1] and propose a novel approach to leverage the advantages from Large Language Models (LLMs) to formalize a foundation model for single-cell data analysis, known as scELMo. We utilize LLMs like GPT 3.5 as a generator for both the description of metadata information and the embeddings for such description. Moreover, we also combine the embeddings from LLMs with the raw data under both zero-shot learning framework and fine-tuning framework to handle different tasks. We proved that scELMo is capable of cell clustering, batch effect correction, and cell-type annotation without training a new model. Moreover, the fine-tuning framework of scELMo can help with more challenging tasks including in-silico treatment analysis or modeling the perturbation information. scELMo has a lighter structure and lower requirement for resources but impressive performance, which poses a promising path for developing domain-specific FMs.

Keywords: Single-cell data analysis, Foundation Model, Large Language Model, In-silico Treatment Analysis, Perturbation Analysis.

1 Introduction

Developing large language models (LLMs) or foundation models (FMs) is becoming more and more important for both engineering and science [2–4]. In the biology area, these models are widely used to analyze information of DNA sequence [5, 6], representation of cells and genes [7–10], etc. Therefore, FMs can facilitate various downstream analysis for biological or medical research. Here we investigate a specific type of biomedical tabular data, known as single-cell sequencing data [11, 12], for its intersection with FMs. Single-cell sequencing data can help us describe the biological activity and information at the cell level. Generally, the observations of single-cell data are cells, while the features can be many candidates, including gene expression information [12], protein expression information [13, 14], methylation information [15], etc.

Several pre-training-based methods were developed to analyze the single-cell data based on the large-scale sequencing results from different databases. However, as a recent work, GenePT [1] questions the contribution of only using the sequencing information for biological analysis. GenePT believes that we can either utilize the information from the National Center for Biotechnology Information (NCBI) [16] as prompts and get the embeddings of such prompts from LLMs to describe the gene (GenePT-w), or transfer the cell to sentence by gene ranking and use the ranking results as prompt to extract the cell embeddings from LLMs (GenePT-s). Based on the results from GenePT, incorporating pre-existing knowledge or literature of the features in single-cell datasets is also meaningful to explore. However, GenePT limits the ability of LLMs and the information from the NCBI database may not be enough to summarize the function of genes in a well-structured approach. Moreover, the idea of using GenePT-s to generate cell embeddings is not very practical. The first reason is the high sparsity of single-cell data, so the genes with zero expression are hard to rank. GenePT-s is also not capable of handling large-scale datasets because of the usage and time limitation from OpenAI API [17]. Recently, the access of LLMs including GPT 3.5 [18], GPT 4 [19], and LLaMa [20] offers us opportunities to explore the information of these features with the help of LLMs. These models have been widely used for knowledge summarizing [21], network searching/model design [22], and other approaches to enhance scientific research [23, 24].

Therefore, we follow the hypothesis of GenePT and explore the ability of using LLMs to generate meaningful text descriptions of cell-level or feature-level metadata as well as embeddings of such descriptions. We assume the embeddings from LLMs carry biological properties and can be utilized in various downstream applications. Hence, we introduce scELMo, as a pipeline for analyzing single-cell multi-omic data based on the text description and embeddings directly from LLMs [25]. Using gene as one example, we use LLMs to summarize the functional information of the given gene with a suitable prompt and also use the same LLMs to extract the embeddings of such description. We then can either incorporate the embeddings directly into the sequenced data by matrix manipulation or combine the embeddings with other models with fine-tuning targets for various challenging tasks.

2 Results

Overview of scELMo. The high-level idea of scELMo is to transfer the information of each cell from the sequencing data space to the LLM embedded space. We can finish this transformation by incorporating information from feature space (for example, genes and proteins.) or cell space (for example, cell types or cell states). To transfer the feature information into LLM embedded space, we can either use scientific information from a known database like NCBI or use the summary given by LLMs as a prompt, and then generate the embeddings for the prompt based on LLMs' embedding layers. Here we choose GPT 3.5 as the tool to summarize the function of features and to generate embeddings based on our evaluation in the Results section.

After having embeddings for features or cells, we can use them based on the zero-shot learning framework for clustering or batch effect correction. Moreover, we can also combine the embeddings with known models to improve their performance for various downstream tasks. Figure 1 shows the flowchart of scELMo under two different frameworks. Details of scELMo under different frameworks are summarized in the Methods section.

Evaluation of Hallucinations in LLMs. The first step of our work is to define a suitable LLM for generating text descriptions and embeddings. The suitable LLM should not suffer from Hallucinations [26], that is, generating fake or incorrect information about our given feature information or cell information. Here we consider GPT 2 [27], GPT 3.5, GPT 4, LLaMa-2 (70B), Mistral [28], bioGPT [29], Claude 2 [30], and Bard (PaLM 2) [31] as candidates to generate text description. Considering the diversity of tokens as well as the access to embedding layers, we utilized GPT 3.5 to generate embeddings. We sampled 20 proteins and 20 genes from the known proteins and genes, and used these models to generate the text description of these features. We verified the correctness of the outputs by comparing them with known databases (GeneCards [32] and NCBI). The proportion of correct outputs is shown in Figure 2 (a). Moreover, we also consider the time usage for one query for each LLM, shown in Figure 2 (b).

Based on our results, GPT 3.5 and GPT 4 gave us the most accurate description. However, GPT 4 required longer query time and some of the outputs from GPT 4 did not follow our restriction of format mentioned in the prompt. By considering the trade-off between query time and the proportion of correct outputs, we selected GPT 3.5 as our tool to generate text descriptions of features. Moreover, we also evaluated the stability of the embeddings of LLM's outputs based on the same prompt for the same gene, as well as the similarity of embeddings from different genes. Our results are summarized in Extended Data Figures 1 (a) and (b). According to Extended Data Figure 1 (a), we found that the correlation between arbitrary embeddings from different LLM outputs of the same gene is higher than 0.9, which implied that the embeddings from GPT 3.5 have high stability and they can capture the intrinsic functions for each gene. Moreover, based on Extended Data Figure 1 (b), we figured out that embeddings from different genes had lower correlation comparing with the embeddings from the same gene. Therefore, embeddings from GPT 3.5 can also capture the functional heterogeneity of different genes. Furthermore, we loaded the gene functional annotation information provided by Geneformer [9] and used the known information to predict

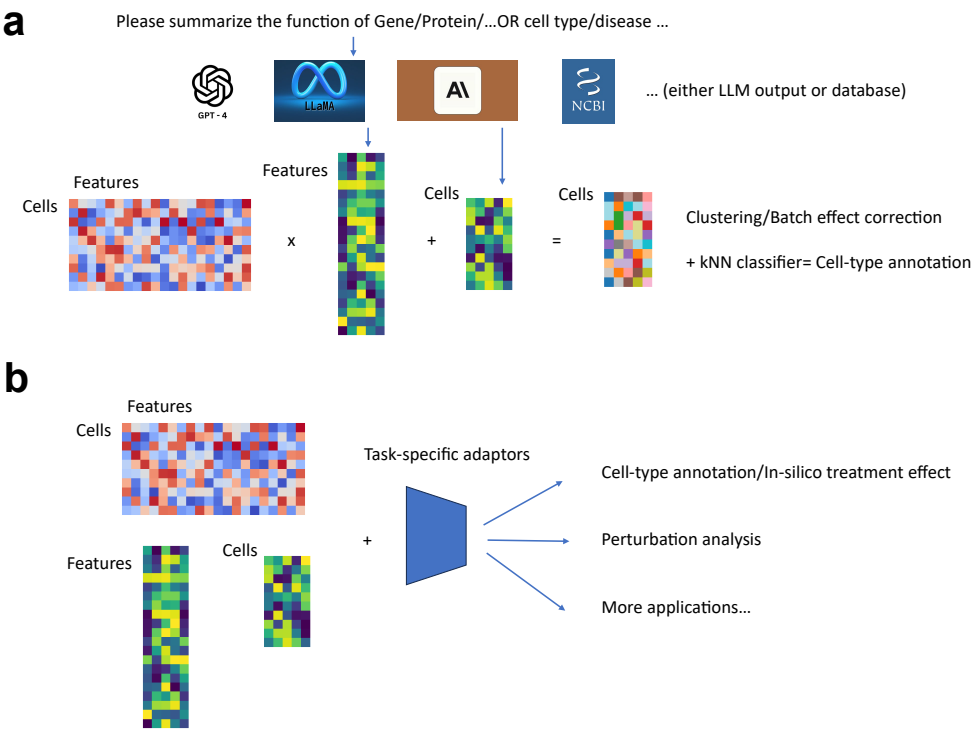


Fig. 1 Overview of scELMo. (a) Zero-shot learning framework of scELMo. We extract the text description of metadata by using either databases or LLMs. Then we use GPT 3.5 to generate the embeddings of the text description as the embeddings of features or cell states. We then aggregate these embeddings with single-cell profiles to generate cell embeddings. (b) Fine-tuning framework of scELMo. We combine embeddings of metadata and single-cell profiles with task-specific adaptors and train the adaptors to address downstream applications.

the functional information for the rest of genes based on a k-nearest-neighbor (kNN) classifier. Extended Data Figures 2 (a) and (b) display the functional clustering for these genes. We found that genes with similar functions were embedded into a joint cluster. We also analyzed text descriptions from NCBI and LLMs in Appendix A. The outputs of different LLMs are summarized in Supplementary File 1.

scELMo for clustering and batch effect correction. In this section, we investigated the contribution of feature embeddings from scELMo for cell-level tasks including clustering and batch effect correction.

Clustering is a straightforward approach to check whether feature embeddings can also carry biological information. We incorporated gene embeddings or protein embeddings into the sequenced data and analyzed the performance of clustering based on cell embeddings. The metrics we chose here include NMI, ARI, and ASW [34]. These metrics are widely used in the evaluation of clustering for single-cell data. In the following analysis, If not otherwise specified, GenePT means GenePT-w. Since we generated gene embeddings for gene names from both NCBI and Ensemble [35], we had more genes comparing with the version provided by GenePT. Moreover, when

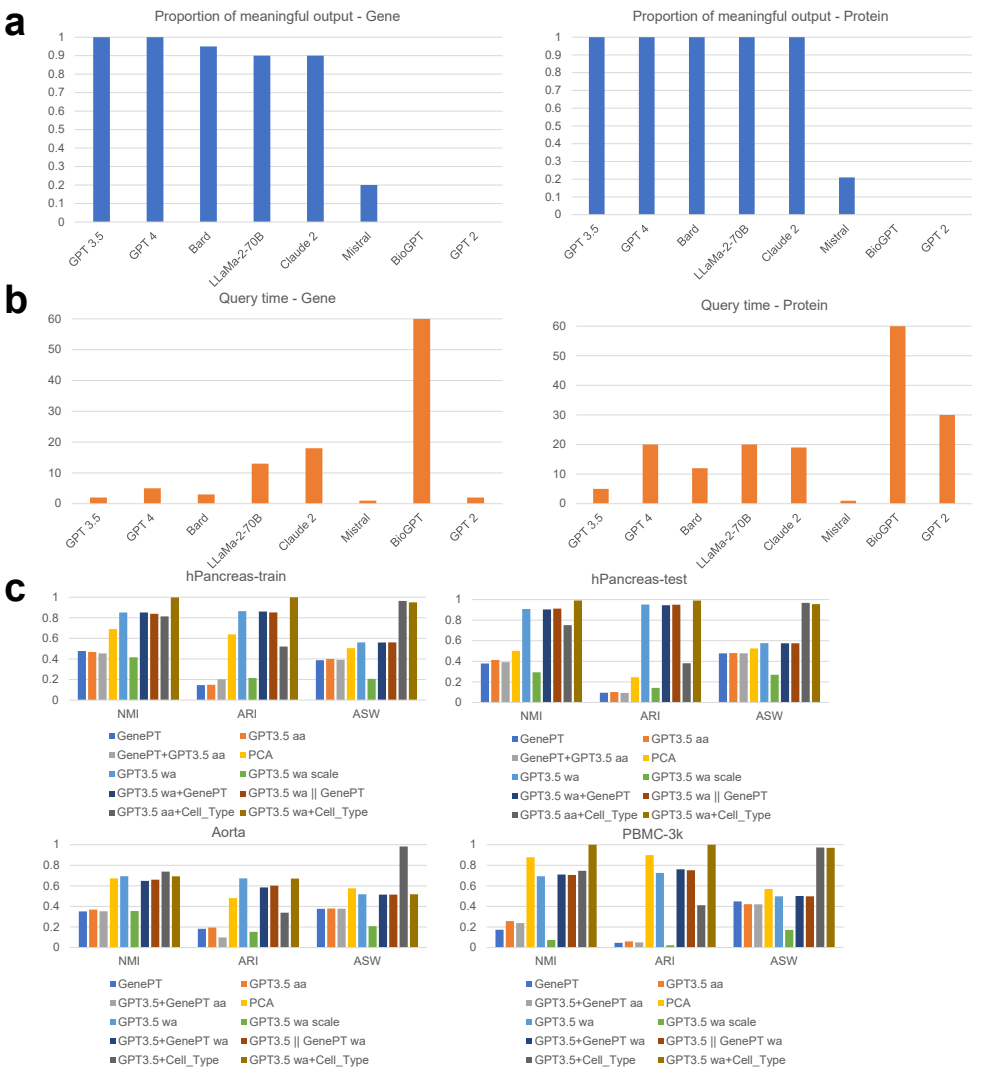


Fig. 2 Evaluations of the outputs of LLMs and the clustering performance. (a) Proportion of meaningful outputs of biological features across different LLMs. The left panel represents the proportion of genes, while the right panel represents the proportion of protein. (b) Average query time for each LLM. The left panel represents the query time of genes, and the right panel represents the query time of proteins. (c) Evaluations of the clustering performance based on different methods. Different panels represent the results of different datasets.

incorporating the feature embeddings, GenePT utilized a naive arithmetic average (aa) method, which meant GenePT treated that the gene expression with different non-zero values had the contribution independent of their scale. However, the level of gene expression is a key factor affecting cellular function [36, 37]. We questioned the hypothesis from GenePT and treated the gene expression levels as weight and computed the weighted average (wa) value for each cell as an alternative averaging

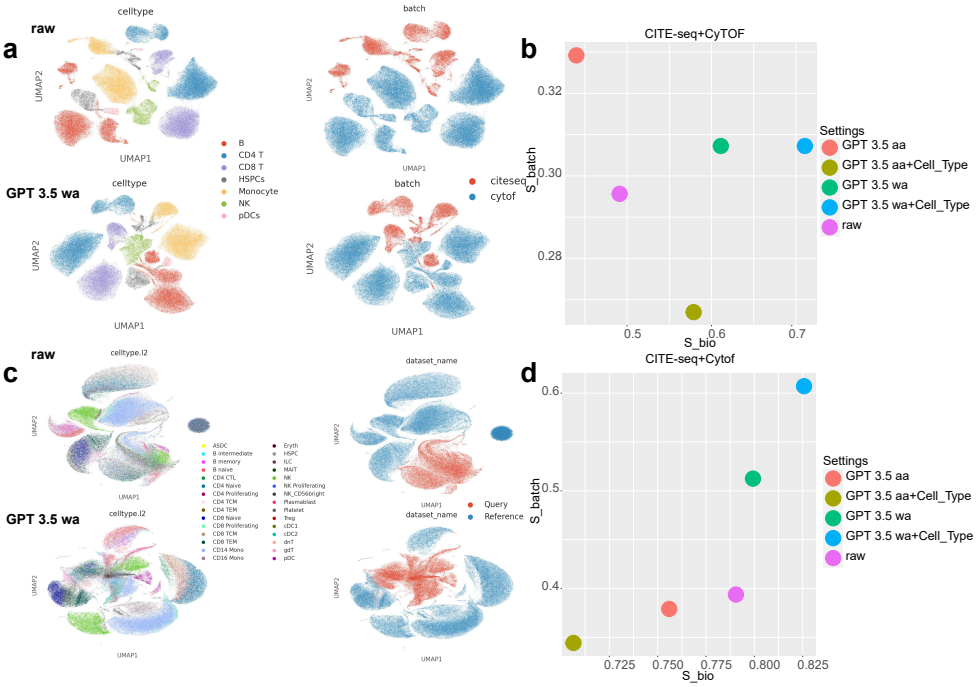


Fig. 3 Results of batch effect correction for single-cell proteomic data. (a) UMAPs [33] for the cell-type information (left) and batch information (right) of cite-seq-based datasets. The upper panel represents the raw data. The bottom panel represents the cell embeddings from GPT 3.5 *wa* mode. (b) Evaluations of the batch effect correction for cite-seq-based datasets across different methods. (c) UMAPs for the cell-type information (left) and batch information (right) of cite-seq-based dataset and CyTOF-based dataset. The upper panel represents the raw data. The bottom panel represents the cell embeddings from GPT 3.5 *wa* mode. (d) Evaluations of the batch effect correction for cite-seq-based dataset and CyTOF-based dataset across different methods.

approach. Weighted average approach is widely used in building statistical models, for example, Best Linear Unbiased Estimator (BLUE) [38] and Generalized Least Squares (GLS) [39]. We unify these approaches for generating cell embeddings and details are included in the Methods section.

Figure 2 (c) shows that using *wa* mode is better than using *aa* mode for clustering. Moreover, if we considered combining cell embeddings with the embeddings of cell-type information from GPT 3.5, we could get scores close to one, which implied that the cell-type embeddings contained meaningful cell-type information. Moreover, using embeddings from scELMo also improves the clustering performance compared with the embeddings from GenePT, advocating the choice of using LLMs as a tool for summarizing scientific concepts. However, combining the embeddings from GenePT and GPT 3.5 under different mechanisms (GPT 3.5 + GenePT *wa* means sum by genes, and GPT 3.5 || GenePT *wa* means concatenation by genes.) did not improve the scores comparing with the individual setting. The average ranks of all methods across different datasets are summarized in Extended Data Figure 3 (a), and GPT 3.5

wa mode has the lowest rank. We also analyzed the potential factors that could affect the clustering performance in the Appendix B.

For the batch effect correction task, since GenePT has already analyzed the performance of such embeddings for single-cell RNA sequencing (scRNA-seq) data, we focused on single-cell proteomic data, such as sequencing data based on CITE-seq [14] and CyTOF [13]. Here we considered two scenarios. We tried to use the embeddings from scELMo and GenePT to reduce the batch effect for datasets from the same protocols and different protocols. To evaluate the performance of batch effect correction, we used metrics from scIB [34] and considered the contribution of our embeddings for reducing batch effect (known as S_{batch}) and for preserving the biological variation (known as S_{bio}) separately. Figures 3 (a) and (b) display the results for batch effect correction from two datasets based on CITE-seq and Figures 3 (c) and (d) display the results for batch effect correction based on one CITE-seq-based dataset and one CyTOF-based dataset. From these four figures, we found that using the *aa* approach could not help on this task and could make the case become worse by reducing these two scores simultaneously. However, if we used the *wa* approach, we could reduce the batch effect proved by the improvement of these two scores. Moreover, incorporating the cell-type embeddings with *aa* mode also could not improve S_{batch} and S_{bio} jointly. However, combining the cell-type embeddings with cell embeddings under *wa* mode can also improve the S_{bio} score for both cases. Therefore, finding a good base embedding space is important for utilizing cell-state embeddings in a proper approach. As an extension, we summarized our analysis of multi-omic data integration [40] in Appendix C.

scELMo for cell-type annotation. Cell-type annotation is a foundational problem in the single-cell analysis [41]. Here we investigated the ability of scELMo to address this task under both zero-shot learning framework and fine-tuning framework. For the zero-shot learning framework, we incorporated the gene embeddings into training datasets and testing datasets to access the cell embeddings and used a kNN classifier implemented by GenePT for the cell-type annotation in the testing dataset. However, this approach failed when training datasets were combined from multiple resources or with large batch effects (shown in the PBMC section of Table 1). Therefore, we also constructed a simple classifier based on neural networks [42] and contrastive learning [43]. We then trained the model based on gene embeddings and expression profiles. Such model is also known as an adaptor in Natural Language Processing (NLP) tasks [25]. We tested the ability of cell embeddings from this model for annotating cell types.

From Table 1, we found that the zero-shot learning ability of embeddings from GPT 3.5 is impressive for the cell-type annotation in the hPancreas dataset and the Aorta dataset. Moreover, representing cells by using the rank of genes and using the representation as prompts cannot help GPT 2 or GPT 4 to give us the correct answers for the cell types of these cells. Therefore, zero-shot learning based on gene embeddings is capable for datasets with no obvious batch effect for cell-type annotation. However, for the PBMC dataset, which is constructed based on datasets from different sources, the gap between zero-shot learning results and fine-tuning results is obvious. Moreover, the fine-tuning results by combining the gene embeddings from

Table 1 Scores of cell-type annotation task under different settings. Parts of the results are directly extracted from GenePT. Here PCA represents principal component analysis, and scELMo+random emb represents fine-tuning scELMo with random numbers as meaningless gene embeddings. Average ranks of all methods across datasets are summarized in Extended Data Figure 3 (b).

	zero-shot settings	Accuracy	Precision	Recall	F1		Accuracy	Precision	Recall	F1		Accuracy	Precision	Recall	F1
hPancreas	GPT 2 query	0.000	0.000	0.000	0.000	Aorta	0.000	0.000	0.000	0.000	PBMC	0.000	0.000	0.000	0.000
	GPT 4 query	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.000
	scGPT (z)	0.770	0.610	0.560	0.550		**0.938**	0.921	0.894	0.901		**0.915**	**0.781**	**0.803**	**0.789**
	Geneformer (z)	0.500	0.250	0.340	0.270		0.860	0.700	0.600	0.620		0.126	0.128	0.130	0.095
	GenePT-w	**0.940**	**0.720**	**0.630**	**0.650**		0.870	0.910	0.680	0.720		0.286	0.607	0.270	0.315
	GenePT-s	0.890	0.650	0.530	0.560		0.860	0.700	0.600	0.620		-	-	-	-
	PCA	0.633	0.369	0.382	0.357		0.929	**0.923**	**0.911**	**0.910**		0.436	0.392	0.278	0.285
	GPT3.5 aa	0.933	0.702	0.614	0.629		0.877	0.867	0.678	0.719		0.190	0.564	0.181	0.230
	GPT3.5 wa	0.933	0.702	0.613	0.629		0.877	0.867	0.677	0.718		0.190	0.562	0.181	0.231
	fine-tuning settings														
	scGPT	**0.970**	0.742	**0.758**	**0.741**		**0.962**	0.938	0.939	0.937		**0.933**	0.798	0.822	0.807
	Geneformer	0.321	0.105	0.094	0.086		0.340	0.087	0.092	0.080		0.235	0.146	0.147	0.131
	scELMo+GenePT	0.963	0.790	0.662	0.680		0.958	**0.952**	**0.940**	**0.946**		0.919	0.785	0.823	0.801
	scELMo+random emb	0.968	0.683	0.693	0.680		0.951	0.913	0.874	0.882		0.903	**0.854**	**0.826**	**0.835**
	scELMo+GPT 3.5	0.966	**0.788**	0.712	0.731		0.956	0.946	0.932	0.938		0.924	0.801	**0.826**	0.811

GPT 3.5 or GenePT and our adaptor are comparable with FMs like scGPT [8] and Geneformer [9]. These two models need more resources for fine-tuning [44]. For all of the datasets we tested, scELMo with a fine-tuning framework performed better than the same settings but under the zero-short learning framework. Moreover, scELMo could improve the annotation score for embeddings from different sources. Therefore, the cell-type annotation based on fine-tuning the adaptor with gene embeddings from LLMs is accurate and user-friendly.

scELMo for in-silico treatment analysis. Using computational simulation methods to facilitate the discovery of novel targeted therapies and drugs is an important task for computer-aided healthcare [45–47]. In this section, we examined the ability of scELMo to use the adaptor for cell-type annotation and the gene embeddings we extracted from GPT 3.5 for modeling human disease and further revealing the potential therapeutic targets. Inspired by Geneformer [9], we finetuned our adaptor based on the classification task of cell conditions and used the cell embeddings from the adaptor for therapeutic target discovery. We ensured that our adaptor reached its best performance and tested the change of embeddings by removing the differenetially expressed genes (DEGs) in the expression levels across different conditions. We discussed the importance of fine-tuning a model in Appendix D. If we remove a suitable candidate for targeted therapies, the cell embeddings under disease will be more similar to the cell embeddings under the control case. The metric we used to evaluate similarity is the cosine similarity between the average of cell embeddings under the diseased case and the average of cell embeddings under the control case. To verify the targets we discovered, we searched the possible research about the target therapies of certain diseases by literature review, which is summarized in Supplementary File 2.

The first group of diseases we considered are hypertrophic or dilated cardiomyopathy states [48]. The scRNA-seq dataset we used here is known as the Heart dataset. We determined the genes whose in-silico deletion in diseased conditions could significantly shift the cell embeddings from disease conditions to the non-failing (NF) or control condition. The change of cosine similarity before and after the deletion for cardiomyopathy states is summarized in Figure 4 (a). Based on this figure, our adaptor identified two novel potential therapeutic targets for cells under the DCM case and four novel targets for cells under the HCM case. The silence of genes identified

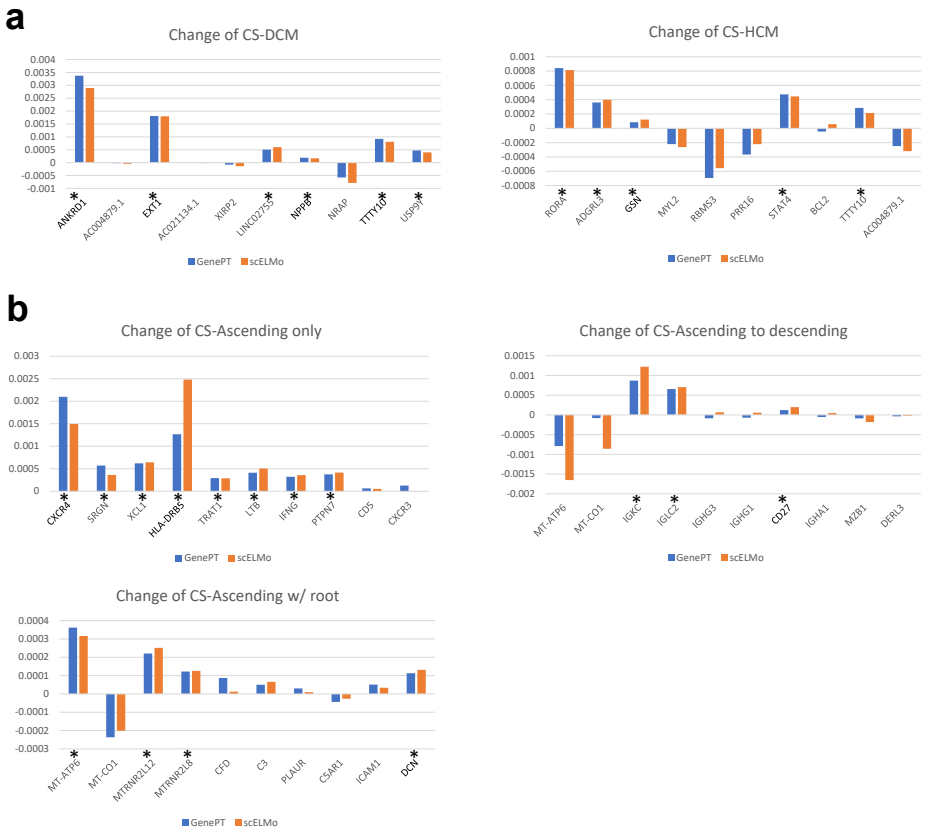


Fig. 4 Results of in-silico treatment analysis. A gene is treated as a potential therapeutic target if the change of cosine similarity (CS) by removing such gene is larger than 1e-4. We chose the top 10 DEGs as candidates. (a) The change of CS by removing different genes in the expression space for hypertrophic (HCM) or dilated cardiomyopathy (DCM) states. (b) The change of CS by removing different genes in the expression space for ascending aortic aneurysm (Ascending only, Ascending to descending, and Ascending w/ root). We highlighted the genes detected by both GenePT and scELMo using stars (*) and marked the genes that were discovered by previous research as therapeutic targets using **bold** type.

by scELMo may help the shift from cells under diseased conditions to cells under the control condition. Moreover, GSN [9] was verified by CRISPR-based technology [49] as a therapeutic target, which strongly supported the reliability of genes selected by scELMo.

The second group of disease we considered is ascending aortic aneurysm [50]. This disease has three different disease states. The scRNA-seq dataset we used here is known as the Aorta dataset. We used the similar approach we mentioned for analyzing cardiomyopathy states and summarized the change of cosine similarity before and after deletion in Figure 4 (b). Based on this figure, we found that scELMo identified six novel genes as potential therapeutic targets for cells under the Ascending only state, two novel genes as potential therapeutic targets for cells under the Ascending to

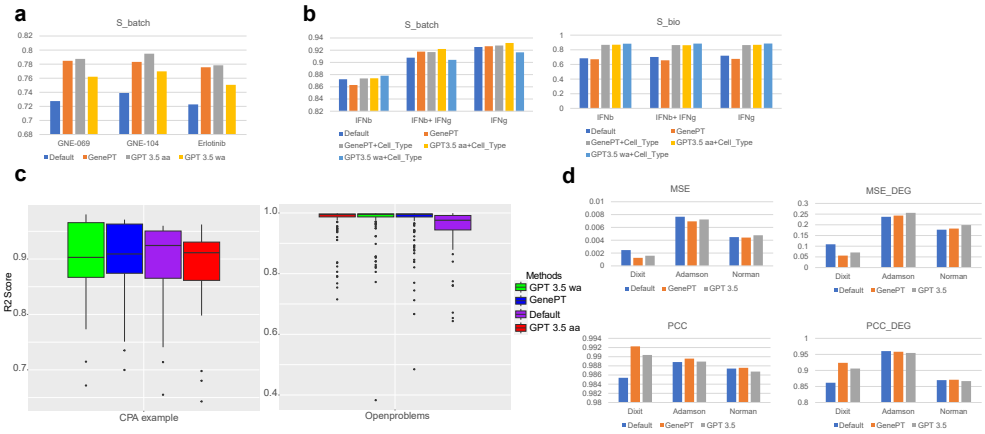


Fig. 5 Results of perturbation analysis. (a) Scores of casual factor analysis for ChangYe2021 dataset [53, 54] based on CINEMA-OT across different input data. We considered four different methods. (b) Scores of causal factor analysis for perturbed PBMC dataset [55] based on CINEMA-OT across different input data. (c) Scores of gene expression prediction under different perturbation cases based on CPA. We considered four different methods and two datasets. (d) Scores of gene expression prediction using perturb-seq datasets based on GEARS. We considered three different methods and three datasets.

descending state, and three genes as potential therapeutic targets for cells under the Ascending w/ root state. Interestingly, MT-ATP6 gene was selected by scELMo as a potential target, which implied that cells under different states might have different speeds for cell death [51] or neurodegeneration [52]. Such results can inspire researchers to explore the correlation between cellular function and clinical manifestations of this disease.

scELMo for perturbation analysis. Analyzing the perturbation effect on cell level is also an important task. Here we focused on three tasks across different perturbation types, for example, cell-level perturbation [56] or gene-level perturbation [57]. Our idea is to use the cell embeddings or gene embeddings generated based on the zero-shot learning framework of scELMo to replace or combine with the original input of different models to enhance their performance on the three downstream tasks. We evaluated the contribution of scELMo for the causal factor analysis task based on CINEMA-OT [55], the gene expression prediction task for cell-level perturbation data based on CPA [58], and the gene expression prediction task for gene-level perturbation data based on GEARS [59]. All of the metrics we considered in the three tasks are from these models.

CINEMA-OT is a tool based on a causal learning framework and optimal transport that helps us distinguish between perturbation effects and intrinsic cell-state effects for scRNA-seq datasets. We replaced the input of CINEMA-OT with cell embeddings from scELMo and verified the contribution of our cell embeddings by evaluating the reduction of batch effect in the space of intrinsic cell-state effects (also known as the confounder space). By removing the perturbation effect, the perturbation case in the intrinsic cell-state space became the batch label and we should not observe the difference of the embeddings from the same cell types in this space. Based on Figure 5

(a), using cell embeddings rather than default PCA results as input can improve the performance of CINEMA-OT for cells from the same cell type. Moreover, embeddings from GenePT here are a better choice compared with the embeddings from GPT 3.5. Moreover, based on Figure 5 (a), only incorporating gene embeddings may not contribute to the casual factor analysis for cells from different cell types. However, if we consider incorporating the cell-type embeddings with cell embeddings and use the updated embeddings as new input, we can improve the performance of CINEMA-OT for both reducing the batch effect and preserving the biological variation across different perturbation cases. Moreover, Extended Data Figures 4 (a) and (b) display that the visualization could also be improved by using the updated embeddings as model input. Extended Data Figures 4 (c) and (d) show that incorporating the cell embeddings into CINEMA-OT did not affect the analysis of gene synergy effect, and the difference between synergy of Monocyte and synergy of other cell types was also obvious. For the causal factor analysis task, using *wa* mode did not contribute much to the score improvement. One possible reason is that CINEMA-OT can learn the best representation of cell embeddings with a good start for optimization, so the *aa* mode is enough. Therefore, scELMo can improve the performance of CINEMA-OT on the causal factor analysis task by offering a new candidate of input data.

CPA is a tool based on Conditional Variational Auto-encoder (CVAE) that helps us predict the gene expression levels for the out-of-distribution (OOD) samples of scRNA-seq data under certain perturbation cases. Here we combined the gene embeddings from scELMo with the original input dataset and learned a new latent space for gene expression prediction. We verified the contribution of gene embeddings by comparing the R2 score between these two different settings. We computed the R2 score based on predicted gene expression levels and ground truth gene expression levels. Figure 5 (c) shows the performance of CPA under different methods across two different datasets. For the analysis of the CPA example dataset shown in the left panel, we found that using the cell embeddings from GPT 3.5 *wa* mode in the training process could slightly improve the average R2 score, while its median value was still lower than the default mode. Therefore, the results were comparable across different methods. However, for the analysis of the Openproblems dataset [60] shown in the right panel, we found that using cell embeddings from both GenePT and GPT 3.5 could improve the performance of CPA. R2 score based on combining cell embeddings with CPA had a higher average value and lower variance compared with the default mode. Therefore, scELMo could improve the performance of CPA on the prediction task by introducing the cell embeddings into the training process.

GEARS is a tool based on Graph Neural Networks (GNN) [61] that helps us predict the gene expression levels for perturb-seq-based datasets. Here we combined the gene embeddings from scELMo with the original gene embeddings of GEARS to learn the predicted value of target genes. We verified the contribution of gene embeddings by comparing the Pearson Correlation Coefficient (PCC) and Mean Squared Errors (MSE) between the default settings and updated settings for both all genes case and DEGs case. All results were summarized in Figure 5 (d). For the Dixit dataset, using gene embeddings from either GPT 3.5 or GenePT could improve the gene expression prediction made by GEARS for both all genes case and DEGs case. However, for the

Adamson and Norman datasets, the contribution of using gene embeddings was not significant. Moreover, gene embeddings from GenePT were always better than the gene embeddings from GPT 3.5 across these three datasets, which implied that information from NCBI or real databases might be more important than the functional information for the expression levels prediction task related to perturb-seq-based datasets.

3 Discussion

Developing methods to model genetic and cellular functionalities is an important task in computational biology research. One idea is to propose a biological question and figure out the specific model to handle this question, while another approach is to have a foundation model to address all the known and unknown biological questions. In the single-cell data analysis area, recently many researchers tended to pre-train a large-scale model and declared a foundation model by showing their comparable performance with different task-specific models in certain downstream tasks. However, it is hard to find tasks that could only be resolved by such pre-training and fine-tuning framework [44], which questioned the contribution of consuming large-scale resources to develop such models. Therefore, inspired by GenePT, we figured out another approach to generalize a foundation model in the single-cell research area, that is, utilizing the contribution of LLMs to generate meaningful feature embeddings and cell embeddings. We could either directly utilize these embeddings for clustering or reducing batch effect, or combine these embeddings with task-specific models to improve their performance. These two approaches are summarized as scELMo. Accessing the outputs of known LLMs like GPT 3.5 did not require many resources, and our results did show the superiority of scELMo in answering multiple biological questions.

For scELMo under the zero-shot learning framework, we could utilize cell embeddings to perform clustering and batch effect correction. These contributions are based on the fact that embeddings from the text description of features in single-cell datasets are good for representing biological concepts or functions. We also discussed the factors that could affect the performance of generating meaningful cell embeddings, including the approach to compute the average embeddings, the number of cells in one dataset, the number of features we need, etc. We also proved that such embeddings could be used for multi-omic data analysis, which posed the power of using LLMs as tools for incorporating prior information to enhance task-specific analysis.

Considering the limitation of the zero-shot learning framework, we also proposed a fine-tuning framework for scELMo. By combining the feature embeddings from GPT 3.5 with a light-structured neural network, we could use the embeddings to annotate cell types and reach a level similar to those of FMs that require a lot of resources for pre-training and fine-tuning. Moreover, scELMo can also be used for detecting novel therapeutic targets by observing the change of embeddings corresponding to the removal of certain genes. We could also directly incorporate the cell embeddings or feature embeddings with task-specific models for better performance in modeling the data with perturbation. In-silico treatment analysis and perturbation analysis are two challenging and important tasks with cell-level knowledge, which further supported the potential of scELMo and related work.

However, both GenePT and scELMo had the following limitations. Firstly, the fast developing speed of LLMs questions the contribution of embeddings from GPT 3.5. With a more powerful model, we will have a better representation of features and cells. Secondly, LLMs could not generate meaningful information for genes that were recently discovered or analyzed. Although GPT 3.5 does not makeup concepts for genes, the lack of knowledge still presents a question for the applications of scELMo. Thirdly, without enough resources, fine-tuning LLMs to generate better domain-specific embeddings is hard to deploy. Moreover, the training datasets for fine-tuning domain-specific LLMs are also hard to collect. Finally, extracting features from other biomedical data like GWAS [62] or scATAC-seq [63] data will be difficult since the number of features for these data is quite large.

Therefore, in the future, we plan to generate a database containing the text description as well as the embeddings of such descriptions of features from different LLMs. Models like CPA can utilize these embeddings to perform prediction under the OOD cases. Moreover, we will keep on figuring out a more practical approach to offer a gene-specific LLM. In addition, since the idea from scELMo is capable of modeling arbitrary biomedical data with tabular format, we believe that extending the usage of scELMo to other tasks or areas is also very promising.

4 Methods

Problem definition. For a typical single-cell dataset $X^{n\times m}$ after normalization with n cells and m features, our target is to utilize the text description from a mapping function $\mathcal{M}()$ for feature-level metadata information $f^{m\times 1}$ and cell-level metadata information $c^{n\times 1}$ to learn the embeddings of cells. If we define the embedding generation layer of $\mathcal{M}()$ as $\mathcal{M}_{\mathrm{e}}()$, our cell embeddings can be represented as:

$$\begin{aligned} e_f &= \mathcal{M}_{\mathrm{e}}(\mathcal{M}(Prompt(f))) \\ e_c &= \mathcal{M}_{\mathrm{e}}(\mathcal{M}(Prompt(c))) \\ e_{cells} &= AVG(X)e_f + e_c, \end{aligned}$$

where $Prompt$ is a mapping function that can transfer the name of input data to the prompt space. The prompts can be used as the input of language models. The function $AVG()$ represents the method we used to average the embeddings of all genes for each cell. If the mode is aa, we divide X by m. If the mode is wa, we divide each row of X by the sum of this row. Considering the cell with index i, we can define these two processes as:

$$\begin{aligned} AVG_{aa}(X_i) &= \frac{X_i}{m}, \\ AVG_{wa}(X_i) &= \frac{X_i}{sum(X_i)}. \end{aligned}$$

Then we use matrix multiplication to combine the feature embeddings and the expression profile. Our default setting of the mapping function is a LLM. GenePT can be treated as a special case of scELMo, that is, replacing the LLM as a known database and using aa mode. Incorporating the embeddings of cell-level metadata is an optional

choice. We intend to investigate if the cell embeddings can offer a better representation compared with the raw data.

Moreover, with the embeddings of feature-level metadata information and cell-level metadata information, we also consider if incorporating our embeddings with task-specific model $\mathcal{T}$ can improve the performance of $\mathcal{T}$, that is:

$$Score(\mathcal{T}(X)) < Score(\mathcal{T}(X, e_f, e_c)),$$

where $Score()$ is a metric to evaluate the output of the given model, higher value represents better output. We also may not need to have e_f and e_c for every model.

scELMo under the zero-shot learning framework. To evaluate the performance of e_{cells} under the zero-shot learning framework, we consider three tasks: clustering, batch effect correction, and cell-type annotation based on a kNN classifier. In this section, $\mathcal{T}$ is a kNN classifier.

For clustering and batch effect correction, we directly use e_{cells} as a new representation for X and evaluate e_{cells} for these two tasks. For cell-type annotation based on a kNN classifier, we consider training dataset X_{train} and testing dataset X_{test}, and their corresponding cell embeddings e_{cells}^{train} and e_{cells}^{test}. We use e_{cells}^{train} with its cell types to train a kNN classifier and perform cell-type annotation based on e_{cells}^{test}. Since kNN is based on similarity searching, we treat this method as an ability of zero-shot learning. We follow the settings from GenePT for this classifier and set k=10.

scELMo under the fine-tuning framework. To evaluate the performance of e_f and e_c under the fine-tuning framework, we considered three tasks: Cell-type annotation with an adaptor, in-silico treatment analysis with an adaptor, perturbation analysis with task-specific models as adaptors.

For the cell-type annotation and in-silico treatment analysis tasks, we propose a light-structured neural network with a contrastive learning [43, 64] design. Here $\mathcal{T}$ is a neural network with ReLU [65] as the activation function. Our intuitions come from the requirement for a good representation of cells with different labels and conditions. Therefore, we formalize the loss function of our model as:

$$\mathcal{L}_{total} = \mathcal{L}_{classifier} + \lambda \mathcal{L}_{contrastive},$$

where $\mathcal{L}_{classifier}$ represents the classification loss of the model output as we use cell-type labels for model training. $\mathcal{L}_{contrastive}$ represents the contrastive learning loss we use to distinguish the representations of cells under different conditions in the latent space. λ is a hyper-parameter and we set $\lambda = 100$ in this manuscript to assign a larger weight for label-aware clustering. We utilize the embeddings after fine-tuning as training and testing datasets and use a kNN classifier to annotate the cell types to evaluate the representation we learn based on scELMo.

To analyze the target of in-silico treatment, we first compute the cosine similarity (CS_{old}) between the average cell embeddings based on the diseased case and control case. Then we delete the target gene by setting its expression profile as zero and compute the new embeddings and cosine similarity (CS_{new}). We define the score of our targeted gene g as:

$$Score(g) = CS_{new} - CS_{old}.$$

If such a score is larger than 1e-4, we treat the gene we analyze as a candidate for therapeutic targets.

For the perturbation analysis task, we consider three different models for three different tasks. Here $\mathcal{T}$ represents different models corresponding to different perturbation analysis tasks. For the causal factor analysis task and CINEMA-OT, we replace the original input of CINEMA-OT with e_{cells}. For the gene expression prediction task and CPA, we add a new neural network component to make e_{cells} learnable and combined the output of this component with the latent space from the original CPA. We do not modify the training process of CPA. For the gene expression prediction task based on perturb-seq-based datasets and GEARS, we add the e_f to the original gene embeddings of GEARS. We do not modify the training process of GEARS.

Data pre-processing. We follow the data pre-processing steps from Scanpy [66] for scRNA-seq datasets. For single-cell proteomic datasets, we follow the pre-processing steps from TotalVI [67] and MARIO [68] and do not change the distribution of the original data because of its density.

Metrics. For the evaluations of clustering and batch effect correction, we utilize the metrics described and implemented by scIB [34]. We compute all the metrics we could compute in the evaluation process. All of the scores of the metrics in scIB are in [0,1] and a higher value means better performance.

For clustering, we use NMI, ARI, and ASW_{label} for evaluation.

For batch effect correction, we compute ASW_{batch}, PCR, Graph Connectivity, and iLISI and averaged the scores from these metrics to generate S_{batch}. We compute ASW_{label}, NMI, ARI, and cLISI and averaged the scores from these metrics to generate S_{bio}.

For the evaluations of cell-type annotation, we use Scikit-learn [69] to calculate Accuracy, Precision, Recall, and F1 score by comparing the predicted cell-type labels and ground-truth cell-type labels. All of the metrics are in [0,1] and a higher value means better performance.

For the evaluations of in-silico treatment analysis, we use Scipy to compute the cosine similarity between the mean cell embeddings from the control case and the mean cell embeddings from the diseased case. The definition of the score here is described in the Methods section.

For the evaluations of perturbation analysis, we have three different tasks with different metrics. For the causal factor analysis task, the metrics are the same as what we use in the batch effect correction task. For the gene expression prediction task based on CPA, we use the R2 score as a metric to evaluate the performance for regression. R2 score is defined as:

$$R2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where y_i represents the ground-truth gene expression level, f_i represents the predicted gene expression level, and $\bar{y}$ represents the average expression levels of the given gene. A higher averaged R2 score and a lower variance mean better performance.

For the evaluation of gene expression prediction tasks based on GEARS, we use PCC and MSE as metrics. We can define PCC as:

$$PCC = \frac{1}{m}\sum_{i}^{m}\frac{cov(y_i, f_i)}{\sigma_{y_i}\sigma_{f_i}},$$

where m represents the number of genes we used, cov represents the covariance, σ represents the standard deviation. ρ is in [0,1] and a higher value means better performance. We can also define MSE as:

$$MSE = \frac{1}{mn}\sum_{i}^{m}\sum_{j}^{n}(y_{ij} - f_{ij})^2,$$

where y_{ij} represents the ground truth gene expression level of gene i in cell j, and f_{ij} represents the predicted gene expression level of gene i in cell j. Lower MSE score means better performance.

We consider computing these two metrics for both all genes case and top 20 DEGs case.

Explanation baseline models. For tasks related to clustering and batch effect correction, we consider PCA (raw) and GenePT as baseline models. PCA is widely used for the dimension reduction of single-cell data. We explain the principles of GenePT in the Introduction section.

For tasks related to cell-type annotation, we consider GPT 2, GPT 4, scGPT, Geneformer, and PCA for evaluation. To evaluate GPT 2 and GPT 4, we transfer the cell information into a sentence based on the rank of genes for each cell and treat this task as a Question-Answer task. scGPT is a pre-training-based FM for multiple tasks in single-cell research. The authors utilized multi-layer transformers to construct the model architecture and pre-trained scGPT based on large-scale scRNA-seq datasets. They finetuned scGPT to address problems in downstream applications like cell-type annotation. Geneformer is also a model based on pre-training. The authors utilized BERT [70] as a based model and pre-trained BERT from the sketch using scRNA-seq data transferred into sentences. They also finetuned Geneformer to address problems in downstream applications including cell-type annotation, in silico-treatment analysis, etc. We extract the embeddings from these two models based on both zero-shot learning framework and fine-tuning framework, and utilize kNN to perform classification based on the embeddings.

For tasks related to perturbation analysis, we consider CINEMA-OT, CPA, and GEARS for modification and evaluation. CINEMA-OT is a method for separating confounding signals or causal factors from perturbation information at single-cell level. The first step is to initialize the expected matrix rank based on biwhitening [71]. The second step is to separate confounder signals and treatment-associated signals based on Independent Component Analysis (ICA) [69]. The last step is to match the cells based on entropy regularized optimal transport [72]. We analyze the confounder embeddings for CINEMA-OT and follow the benchmarking pipeline mentioned in the

original paper. CPA is a method for modeling the gene expression levels of scRNA-seq data with perturbation information. CPA treats perturbation information of cells as covariates and encodes these covariates as embeddings into the training process of a CVAE. CPA can also be used to predict the gene expression levels of OOD samples. GEARS is a method for predicting gene expression levels of perturb-seq-based datasets. It models the perturbation information of genes based on knowledge graph and gene embeddings, thus utilizing GNN and Multi-layer Perceptrons (MLP) to predict the gene expression levels under gene-level perturbation.

Datasets information. We did not generate new sequencing datasets in this project. The download links and data statistics are summarized in the Supplementary File 3.

Reproductivities. To generate the text description and embeddings, we rely on the API of OpenAI. To run scELMo, we rely on Yale High-performance Computing Center (YCRC) and utilize one NVIDIA A5000 GPU with up to 30 GB RAM for fine-tuning. We utilize the bigmem node with up to 1000 GB RAM for analysis.

The codes of scELMo can be found in https://github.com/HelloWorldLTY/scELMo. The licence is MIT license.

Acknowledgements. We appreciate information of datasets from Gefei Wang and Chen Lin for the usage of scELMo. We also appreciate the discussion with Yiqun Chen.

Author contributions. T.L. designed this study. T.L. and X.L. designed the model. T.L., T.C. and W.Z. run all the experiments. T.L., T.C. and W.Z. wrote the manuscript. H.Z. supervised this work.

Competing interests. The authors declare no competing interests.

References

[1] Chen, Y.T., Zou, J.: Genept: A simple but hard-to-beat foundation model for genes and cells built from chatgpt. bioRxiv, 2023–10 (2023)

[2] Wu, J., Yang, S., Zhan, R., Yuan, Y., Wong, D.F., Chao, L.S.: A survey on llm-gernerated text detection: Necessity, methods, and future directions. arXiv preprint arXiv:2310.14724 (2023)

[3] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)

[4] Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al.: A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. arXiv preprint arXiv:2302.09419 (2023)

[5] Nguyen, E., Poli, M., Faizi, M., Thomas, A., Birch-Sykes, C., Wornow, M., Patel, A., Rabideau, C., Massaroli, S., Bengio, Y., et al.: Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. arXiv preprint arXiv:2306.15794 (2023)

[6] Marin, F.I., Teufel, F., Horrender, M., Madsen, D., Pultz, D., Winther, O., Boomsma, W.: Bend: Benchmarking dna language models on biologically meaningful tasks. arXiv preprint arXiv:2311.12570 (2023)

[7] Fan, Y., Li, Y., Ding, J., Li, Y.: Gfetm: Genome foundation-based embedded topic model for scatac-seq modeling. bioRxiv, 2023–11 (2023)

[8] 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. bioRxiv, 2023–04 (2023)

[9] Theodoris, C.V., Xiao, L., Chopra, A., Chaffin, M.D., Al Sayed, Z.R., Hill, M.C., Mantineo, H., Brydon, E.M., Zeng, Z., Liu, X.S., et al.: Transfer learning enables predictions in network biology. Nature, 1–9 (2023)

[10] Yang, F., Wang, W., Wang, F., Fang, Y., Tang, D., Huang, J., Lu, H., Yao, J.: 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 (2022)

[11] Han, X., Zhou, Z., Fei, L., Sun, H., Wang, R., Chen, Y., Chen, H., Wang, J., Tang, H., Ge, W., *et al.*: Construction of a human cell landscape at single-cell level. Nature **581**(7808), 303–309 (2020)

[12] Saliba, A.-E., Westermann, A.J., Gorski, S.A., Vogel, J.: Single-cell rna-seq: advances and future challenges. Nucleic acids research **42**(14), 8845–8860 (2014)

[13] Cheung, R.K., Utz, P.J.: Cytof—the next generation of cell detection. Nature Reviews Rheumatology **7**(9), 502–503 (2011)

[14] Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P.K., Swerdlow, H., Satija, R., Smibert, P.: Simultaneous epitope and transcriptome measurement in single cells. Nature methods **14**(9), 865–868 (2017)

[15] Karemaker, I.D., Vermeulen, M.: Single-cell dna methylation profiling: technologies and biological applications. Trends in biotechnology **36**(9), 952–965 (2018)

[16] Wheeler, D.L., Barrett, T., Benson, D.A., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Edgar, R., Federhen, S., *et al.*: Database resources of the national center for biotechnology information. Nucleic acids research **35**(suppl_1), 5–12 (2007)

[17] Shevlane, T.: Structured access: an emerging paradigm for safe ai deployment. arXiv preprint arXiv:2201.05159 (2022)

[18] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P.,

Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.*: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)

[19] OpenAI: GPT-4 Technical Report (2023)

[20] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)

[21] Xiao, L., Chen, X.: Enhancing llm with evolutionary fine tuning for news summary generation. arXiv preprint arXiv:2307.02839 (2023)

[22] Jawahar, G., Abdul-Mageed, M., Lakshmanan, L.V., Ding, D.: Llm performance predictors are good initializers for architecture search. arXiv preprint arXiv:2310.16712 (2023)

[23] Kumar, V., Gleyzer, L., Kahana, A., Shukla, K., Karniadakis, G.E.: Mycrunchgpt: A llm assisted framework for scientific machine learning. Journal of Machine Learning for Modeling and Computing **4**(4) (2023)

[24] Varghese, J., Chapiro, J.: Chatgpt: The transformative influence of generative ai on science and healthcare. Journal of Hepatology (2023)

[25] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Walker, M., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://doi.org/10.18653/v1/N18-1202 . https://aclanthology.org/N18-1202

[26] Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al.: Siren's song in the ai ocean: A survey on hallucination in large language models. arXiv preprint arXiv:2309.01219 (2023)

[27] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.*: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[28] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.-A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7B (2023)

[29] Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., Liu, T.-Y.: Biogpt: generative pre-trained transformer for biomedical text generation and mining.

Briefings in Bioinformatics **23**(6), 409 (2022)

[30] Models, C.: Model card and evaluations for claude models (2023). https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf

[31] Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J.H., Shafey, L.E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G.H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C.A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A.C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D.R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., Wu, Y.: PaLM 2 Technical Report (2023)

[32] Safran, M., Dalah, I., Alexander, J., Rosen, N., Iny Stein, T., Shmoish, M., Nativ, N., Bahir, I., Doniger, T., Krug, H., et al.: Genecards version 3: the human gene integrator. Database **2010** (2010)

[33] McInnes, L., Healy, J., Saul, N., Großberger, L.: Umap: Uniform manifold approximation and projection. Journal of Open Source Software **3**(29), 861 (2018) https://doi.org/10.21105/joss.00861

[34] Luecken, M.D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Müller, M.F., Strobl, D.C., Zappia, L., Dugas, M., Colomé-Tatché, M., *et al.*: Benchmarking atlas-level data integration in single-cell genomics. Nature methods **19**(1), 41–50 (2022)

[35] Martin, F.J., Amode, M.R., Aneja, A., Austine-Orimoloye, O., Azov, A.G., Barnes, I., Becker, A., Bennett, R., Berry, A., Bhai, J., *et al.*: Ensembl 2023. Nucleic acids research **51**(D1), 933–941 (2023)

[36] Soneson, C., Robinson, M.D.: Bias, robustness and scalability in single-cell differential expression analysis. Nature methods **15**(4), 255–261 (2018)

[37] Armingol, E., Officer, A., Harismendy, O., Lewis, N.E.: Deciphering cell–cell interactions and communication from gene expression. Nature Reviews Genetics **22**(2),

71–88 (2021)

[38] Henderson, C.R.: Best linear unbiased estimation and prediction under a selection model. Biometrics, 423–447 (1975)

[39] Kariya, T., Kurata, H.: Generalized Least Squares. John Wiley & Sons, ??? (2004)

[40] Miao, Z., Humphreys, B.D., McMahon, A.P., Kim, J.: Multi-omics integration in the age of million single-cell data. Nature Reviews Nephrology **17**(11), 710–724 (2021)

[41] Zeng, H.: What is a cell type and how to define it? Cell **185**(15), 2739–2755 (2022)

[42] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[43] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020). PMLR

[44] Liu, T., Li, K., Wang, Y., Li, H., Zhao, H.: Evaluating the utilities of large language models in single-cell data analysis. bioRxiv, 2023–09 (2023)

[45] Li, Y.Y., An, J., Jones, S.J.: A computational approach to finding novel targets for existing drugs. PLoS computational biology **7**(9), 1002139 (2011)

[46] Kumar, R., Saha, P.: A review on artificial intelligence and machine learning to improve cancer management and drug discovery. International Journal for Research in Applied Sciences and Biotechnology **9**(3), 149–156 (2022)

[47] Abdelazim, M.A., Nasr, M.M., Ead, W.M.: A survey on classification analysis for cancer genomics: Limitations and novel opportunity in the era of cancer classification and target therapies. Annals of Tropical Medicine and Public Health **23**, 24 (2020)

[48] Spudich, J.A.: Hypertrophic and dilated cardiomyopathy: four decades of basic research on muscle lead to potential therapeutic approaches to these devastating genetic diseases. Biophysical journal **106**(6), 1236–1249 (2014)

[49] Barrangou, R., Doudna, J.A.: Applications of crispr technologies in research and beyond. Nature biotechnology **34**(9), 933–941 (2016)

[50] Davies, R.R., Kaple, R.K., Mandapati, D., Gallo, A., Botta Jr, D.M., Elefteriades, J.A., Coady, M.A.: Natural history of ascending aortic aneurysms in the setting of an unreplaced bicuspid aortic valve. The Annals of thoracic surgery **83**(4), 1338–1344 (2007)

[51] Moreno-Loshuertos, R., Movilla, N., Marco-Brualla, J., Soler-Agesta, R., Ferreira, P., Enríquez, J.A., Fernández-Silva, P.: A mutation in mouse mt-atp6 gene induces respiration defects and opposed effects on the cell tumorigenic phenotype. International Journal of Molecular Sciences **24**(2), 1300 (2023)

[52] Stendel, C., Neuhofer, C., Floride, E., Yuqing, S., Ganetzky, R.D., Park, J., Freisinger, P., Kornblum, C., Kleinle, S., Schöls, L., et al.: Delineating mt-atp6-associated disease: From isolated neuropathy to early onset neurodegeneration. Neurology Genetics **6**(1) (2020)

[53] Chang, M.T., Shanahan, F., Nguyen, T.T.T., Staben, S.T., Gazzard, L., Yamazoe, S., Wertz, I.E., Piskol, R., Yang, Y.A., Modrusan, Z., *et al.*: Identifying transcriptional programs underlying cancer drug response with trace-seq. Nature Biotechnology **40**(1), 86–93 (2022)

[54] Green, T.D., Peidli, S., Shen, C., Gross, T., Min, J., Garda, S., Taylor-King, J.P., Marks, D.S., Luna, A., Blüthgen, N., *et al.*: scperturb: Information resource for harmonized single-cell perturbation data. In: NeurIPS 2022 Workshop on Learning Meaningful Representations of Life (2022)

[55] Dong, M., Wang, B., Wei, J., O. Fonseca, A.H., Perry, C.J., Frey, A., Ouerghi, F., Foxman, E.F., Ishizuka, J.J., Dhodapkar, R.M., et al.: Causal identification of single-cell experimental perturbation effects with cinema-ot. Nature Methods, 1–11 (2023)

[56] Lotfollahi, M., Wolf, F.A., Theis, F.J.: scgen predicts single-cell perturbation responses. Nature methods **16**(8), 715–721 (2019)

[57] Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C.P., Jerby-Arnon, L., Marjanovic, N.D., Dionne, D., Burks, T., Raychowdhury, R., *et al.*: Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. cell **167**(7), 1853–1866 (2016)

[58] Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I.L., Srivatsan, S.R., Naghipourfar, M., Daza, R.M., Martin, B., et al.: Predicting cellular responses to complex perturbations in high-throughput screens. Molecular Systems Biology, 11517 (2023)

[59] Roohani, Y., Huang, K., Leskovec, J.: Gears: Predicting transcriptional outcomes of novel multi-gene perturbations. BioRxiv, 2022–07 (2022)

[60] Burkhardt, D., Benz, A., Lieberman, R., Gigante, S., Chow, A., Holbrook, R., Cannoodt, R., Luecken, M.: Open problems – single-cell perturbations. Kaggle, (2023)

[61] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017). https:

//openreview.net/forum?id=SJU4ayYgl

[62] Visscher, P.M., Brown, M.A., McCarthy, M.I., Yang, J.: Five years of gwas discovery. The American Journal of Human Genetics **90**(1), 7–24 (2012)

[63] Mimitou, E.P., Lareau, C.A., Chen, K.Y., Zorzetto-Fernandes, A.L., Hao, Y., Takeshima, Y., Luo, W., Huang, T.-S., Yeung, B.Z., Papalexi, E., *et al.*: Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. Nature biotechnology **39**(10), 1246–1258 (2021)

[64] Musgrave, K., Belongie, S., Lim, S.-N.: Pytorch metric learning. arXiv preprint arXiv:2008.09164 (2020)

[65] Agarap, A.F.: Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375 (2018)

[66] Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. Genome biology **19**, 1–5 (2018)

[67] Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K.L., Streets, A., Yosef, N.: Joint probabilistic modeling of single-cell multi-omic data with totalvi. Nature methods **18**(3), 272–282 (2021)

[68] Zhu, B., Chen, S., Bai, Y., Chen, H., Liao, G., Mukherjee, N., Vazquez, G., McIlwain, D.R., Tzankov, A., Lee, I.T., *et al.*: Robust single-cell matching and multimodal analysis using shared and distinct features. Nature Methods **20**(2), 304–315 (2023)

[69] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)

[70] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423 . https://aclanthology.org/N19-1423

[71] Landa, B., Zhang, T.T., Kluger, Y.: Biwhitening reveals the rank of a count matrix. SIAM Journal on Mathematics of Data Science **4**(4), 1420–1446 (2022)

[72] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)

[73] Cordero, P., Erbilgin, A., Shang, C., Morley, M.P., Wheeler, M., Dewey, F., Smith, K.S., Hu, R., Brandimarto, J., Liu, Y., et al.: A community overlap strategy reveals central genes and networks in heart failure. bioRxiv, 038174 (2016)

A Examples of the text description outputs of different LLMs.

In this section, we present the difference between the text description from NCBI and the text description from GPT 3.5 for the same gene. We highlight the problematic output information of each text using red color, highlight the description of functional information of each text using blue color, and highlight the specific information contained in each text using **bold** text.

Here is an example for the text description of gene COL1A1 from NCBI:

Official Symbol COL1A1provided by HGNC Official Full Name collagen type I alpha 1 chainprovided by HGNC Primary source HGNC:HGNC:2197 See related ***Ensembl:ENSG00000108821 MIM:120150; AllianceGenome:HGNC:2197 Gene type protein coding RefSeq status REVIEWED Organism Homo sapiens Lineage Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae; Homo Also known as OI1; OI2; OI3; OI4; EDSC; CAFYD; EDSARTH1*** *Summary This gene encodes the pro-alpha1 chains of type I collagen whose triple helix comprises two alpha1 chains and one alpha2 chain. Type I is a fibril-forming collagen found in most connective tissues and is abundant in bone, cornea, dermis and tendon. Mutations in this gene are associated with osteogenesis imperfecta types I-IV, Ehlers-Danlos syndrome type VIIA, Ehlers-Danlos syndrome Classical type, Caffey Disease and idiopathic osteoporosis. Reciprocal translocations between chromosomes 17 and 22, where this gene and the gene for platelet-derived growth factor beta are located, are associated with a particular type of skin tumor called dermatofibrosarcoma protuberans, resulting from unregulated expression of the growth factor. Two transcripts, resulting from the use of alternate polyadenylation signals, have been identified for this gene. [provided by R. Dalgleish, Feb 2008] Expression Biased expression in gall bladder (RPKM 850.7), urinary bladder (RPKM 497.1) and 11 other tissues See more Orthologs mouse all.*

Here is an example for the text description of gene COLA1 from GPT 3.5 using the prompt: *Please summarize the major function of a gene: COL1A1. Use academic language in one paragraph and include pathway information.*:

COL1A1 is a gene that encodes the alpha1 chain of type I collagen, ***which is the most abundant extracellular matrix protein and a major component of the bone, skin, tendon, and other connective tissues. The gene plays a crucial role in the synthesis and assembly of type I collagen fibers, contributing to the structural integrity and mechanical properties of these tissues. Additionally, type I collagen is involved in the regulation of signaling pathways, such as the TGF-β signaling pathway, which plays a key role in the regulation of cell growth, differentiation, and extracellular matrix synthesis. Mutations in COL1A1 are associated with various connective***

tissue disorders, including osteogenesis imperfecta and Ehlers-Danlos syndrome. Therefore, understanding the function of COL1A1 is important in elucidating the pathogenesis of these disorders and developing potential therapeutic strategies.

For the text description from NCBI, we need to clean the format of the original data and transfer them into text representations. Therefore, the sentences and words in this text are incoherent and their format does not strictly follow the grammar. Such difference poses the problem of model alignment with human value. Moreover, text descriptions of NCBI focus specifically on the variety of gene names, symbols, refseq status, and other properties. Moreover, it contained detailed functional information of the given gene. Some of the information may be redundant. One advantage of the text representation from NCBI is its authority, which means the reliability of text descriptions from NCBI is generally greater than the outputs from LLMs for researchers.

For the text description from GPT 3.5, we can see that the sentences and words are coherent and formalized under the correct grammar framework. This text focuses more on the overview of functions for this given gene, including its major functional tissues and cell types (for cell-type marker genes like CD79). Moreover, this text also includes the relation between COLA1A1 and certain diseases, highlighting the potential of this gene as a therapeutic target. Such prior information will be helpful in the in-silico treatment analysis research. Moreover, different prompts can generate different types of descriptions for the same gene, exploring the diversity of the outputs from LLMs is also an interesting research track.

Therefore, these two different types of text descriptions have their own advantages and disadvantages, which might be the explanation for their different suitable scenarios. One interesting research topic will be how to combine the advantages of these two kinds of text descriptions to enhance their downstream applications.

Here is an example for the text description of B cells from GPT 3.5 using the prompt: *Please summarize the major function of cell type: B. Use academic language in one paragraph.*:

B cells, a crucial component of the adaptive immune system, play a pivotal role in ***humoral immunity by generating antibodies (immunoglobulins) in response to the recognition of specific antigens. Following activation, B cells undergo clonal expansion and differentiation into plasma cells, which are specialized antibody-producing cells. The antibodies secreted by plasma cells serve as effectors in immune responses, neutralizing pathogens, facilitating their removal by phagocytic cells, and promoting various immune processes such as opsonization and complement activation. Additionally, B cells can differentiate into memory B cells, which confer long-term immunity by enabling a rapid and robust antibody response upon re-exposure to the same pathogen. Overall, B cells are central to the adaptive immune system's ability to recognize and respond to a wide array of infectious agents and contribute significantly to immune memory and protection.***

For the text description from GPT 3.5, we find that the text description is also coherent and follows the standard grammar. It summarizes the major functions of B cells as well as the cell-cell communication that B cells involve.

B Understanding the effect of the number of cells and the number of genes towards GenePT and scELMo.

In this section, we further investigated the difference between the two averaging modes and the application scenarios and analyzed the relationship between the attributes of raw data and clustering effects.

First, we directly plotted the visualization results of three different methods for the hPancreas-train dataset in Extended Data Figure 5. From this figure, we found that using neither gene embeddings from GenePT nor GPT 3.5 with *aa* mode could preserve the cell-type-specific clusters in the space of UMAPs. However, using gene embeddings from GPT 3.5 with *wa* mode could preserve the major cell-type-specific clustering information. Based on this interesting observation, we further compared the weights used for these two averaging modes based on KL-divergence (KL-Div) [69], which could reflect the difference for the distribution of gene expression levels in each cell and a uniform distribution with $P = \frac{1}{m}$. For each dataset, the row sum is one for both these two cases. Based on Extended Data Figure 7 (a), we found that for all the four datasets we compared in the cell clustering task, the KL-div was larger than two and none of them had cells with zero divergence. Therefore, the distribution of gene expression levels carried more information compared with the weights based on uniform distribution. Moreover, we also showed that using *wa* mode was better for batch effect correction. Therefore, the *wa* mode is more suitable to handle tasks under the zero-shot learning framework.

Second, we analyzed the relation between the clustering performance and the number of genes. The first scenario we intended to investigate is the relation between the number of recorded genes and the clustering performance. In Extended Data Figure 7 (b), we display the change of clustering metrics with respect to the change of recorded genes based on the hPancreas-train dataset using *aa* mode. We noticed that the *wa* mode was not suitable for this research because we might have cells with zero expression by filtering some genes. There is no obvious correlation between the number of recorded genes and the clustering performance. Moreover, since the sources of GenePT or scELMo do not match all of the genes for every scRNA-seq dataset, sometimes we need to fill the gene embeddings of missing genes as zero. Therefore, we also investigated the relation between the number of matched genes and the clustering performance, shown in Extended Data Figure 7 (c). From this figure, we still did not observe a strong correlation between the number of matched genes and the clustering performance under the *aa* mode. However, for the *wa* mode, we found an obvious correlation between these two values. Therefore, having more matched genes can contribute to cell clustering under the *wa* mode of scELMo. Such conclusion also demonstrates the importance of extending our databases of feature embeddings.

Third, we analyzed the relation between the clustering performance and the number of cells. We subsampled different proportions of cells from the large-scale Onek1k PBMC dataset and computed the clustering results under different numbers of cells. Based on Extended Data Figure 7 (d), we found that there was no obvious correlation between the number of cells and the clustering performance for Onek1k PBMC

dataset. Therefore, cell number may not be a factor that can affect the performance of gene embeddings in this task. Moreover, scELMo is also capable of the analysis of large-scale scRNA-seq datasets.

C Analysis of multi-omic data integration.

In this section, we analyzed the possibility of utilizing gene embeddings from GPT 3.5 to resolve multi-omic data integration task. Here we consider datasets from scRNA-seq and scATAC-seq without paired information. To reduce the dimensions of the scATAC-seq dataset, we transfer the feature information of such dataset from the space of peaks to the space of gene activity scores. The visualization results are summarized in Extended Data Figures 6 (a) and (b). According to these figures, we can still observe significant batch effect or the difference of cell embeddings from the cells with same cell types. Therefore, the function of scELMo for multi-omic data integration under the zero-shot learning framework is not good. Moreover, based on Extended Data Figure 6 (c), neither *wa* mode nor *aa* mode can improve the S_{batch} score and the s_{bio} score significantly. Incorporating the cell-type information into the cell embeddings space can significantly improve the averaged scores, but for metrics like iLISI to evaluate the mixture of batch information, such embeddings still had zero score. Therefore, scELMo is not capable of multi-omic data integration under the zero-shot learning framework.

D The contribution of finetuned model in the in-silico treatment analysis.

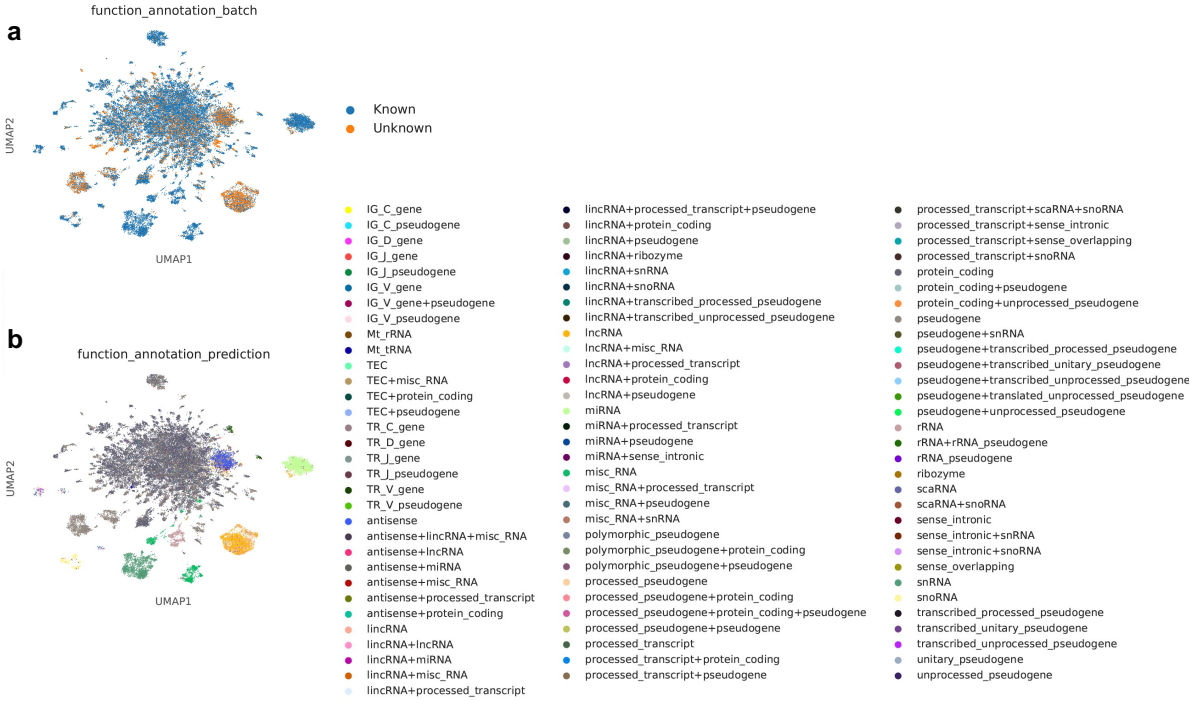
In this section, we demonstrated the necessity of using a finetuned model rather than zero-shot learning for in-silico treatment analysis. In Extended Data Figure 8 (a), we display the change of CS for the same group of DEGs under the ascending aortic aneurysm disease and all genes were not significant for the Ascending only state. Moreover, the change of CS under PCA was nearly constant by varying different genes for removal, and the change of CS was very small for cell embeddings based on GenePT across all three states. Therefore, we concluded that the cell embeddings from either GenePT or PCA were not capable of modeling this disease without the knowledge of the intercellular variability due to diseases. Moreover, based on Extended Data Figures 8 (b) and (c), such variability was covered by the noise in the original expression space. Therefore, we need to learn a model that can distinguish the cells under different conditions as well as generate representative cell embeddings for the inference of novel therapeutic targets. Our ideas aligned with the strategy adopted by Geneformer.

Moreover, we compared the classification metrics for the cell-level disease condition under different gene embeddings for these two datasets and displayed the results in Extended Data Figure 9. From this figure, we found using gene embeddings containing information from genes generally had better performance than using embeddings from random numbers. Moreover, the classification results of scELMo based on gene embeddings from GPT 3.5 were slightly better than the results based on gene embeddings from GenePT. Therefore, using gene embeddings from GenePT and GPT 3.5 all contributed to generating representative latent space for different diseases.

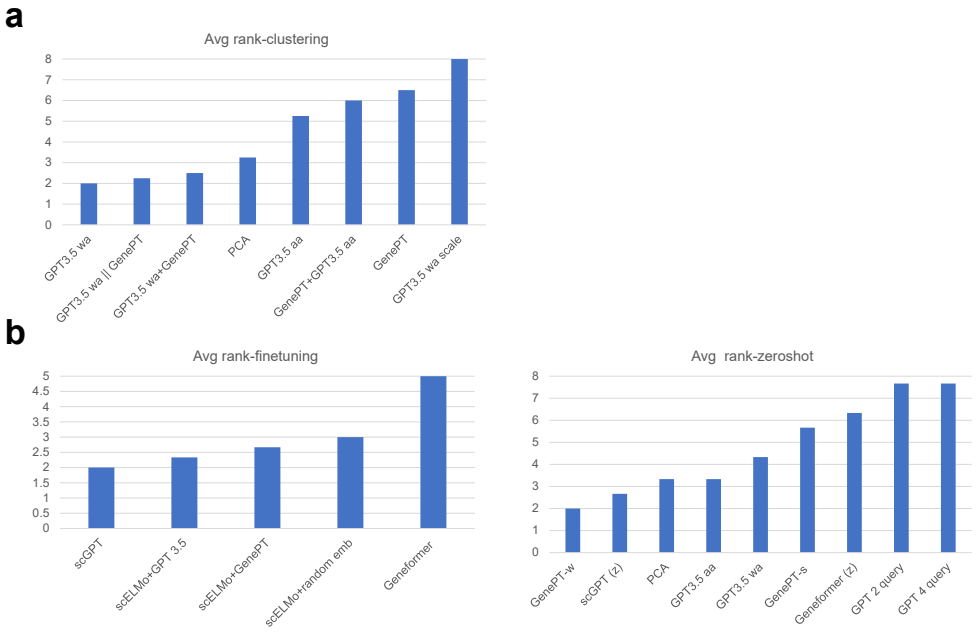
We also considered genes whose silence might shift the cell embeddings from the control condition to diseased conditions. Therefore, our candidate genes became DEGs for the control case and we reversed our score to keep its direction (higher score means that the removal of this gene contributes to the change of cells from the healthy condition to diseased conditions). Our results are summarized in Extended Data Figures 10 (a) and (b). Based on our results, we identified different number of genes for different diseases. Moreover, there existed gene overlap across three states of the given disease, which implied that the removal or silence of such gene might have different contributions for different disease. Because the function of genes is closely related to the pathway [73], our findings can help analyze the pathogenesis of some diseases.

E Supplementary figures

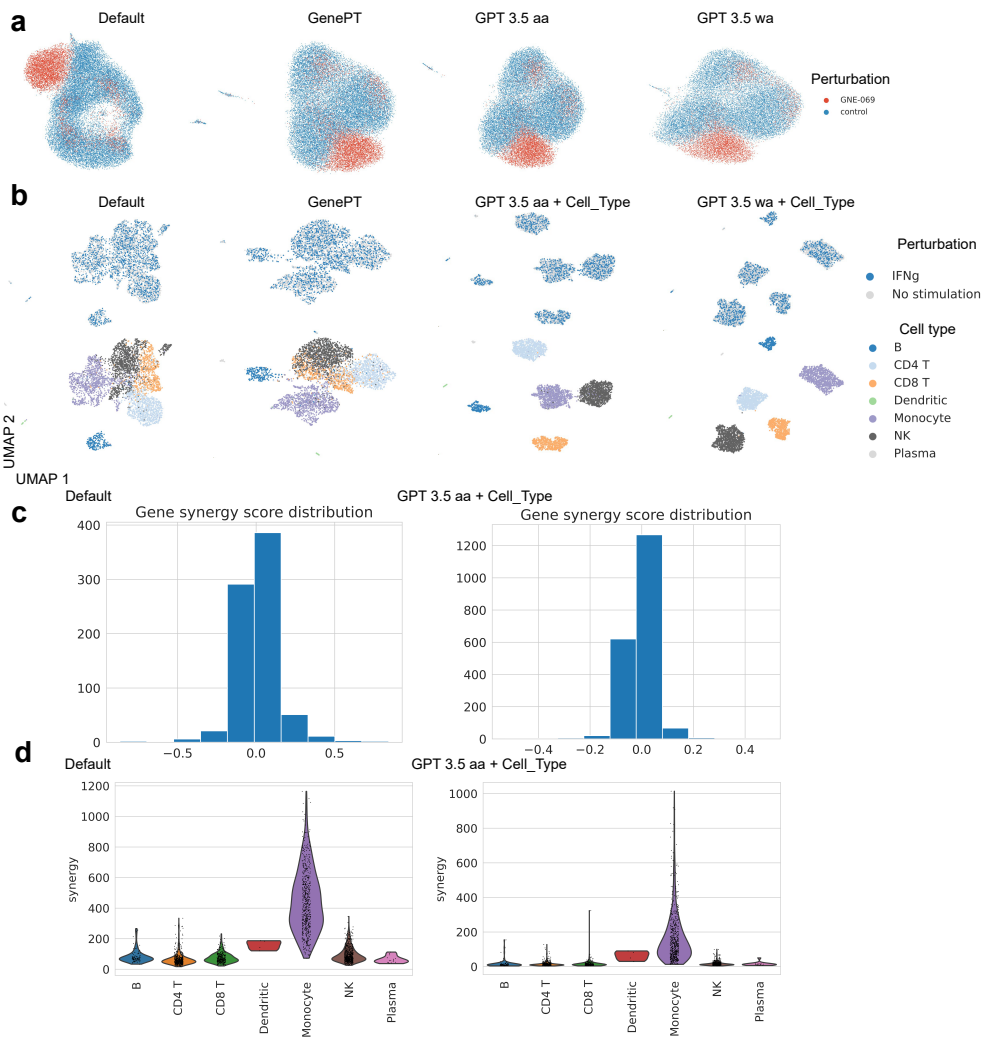
Extended Data Fig. 1 Evaluations of the similarity for embeddings under different conditions. (a) Heatmaps of correlation of embeddings from the same gene under 10 different LLM outputs. We randomly selected 10 genes and generated the LLMs' outputs for these 10 genes based on the same set of prompts. We then computed the embeddings of these outputs and calculated the Pearson correlation for these embeddings, hence we have 10 different heatmaps to represent the results for 10 different genes. (b) Heatmap of the correlaiton of embeddings from 10 different genes. We computed the Pearson correlation for the embeddings of different genes. The number $_0$ represents the index of embeddings we computed based on 10 different LLM outputs.



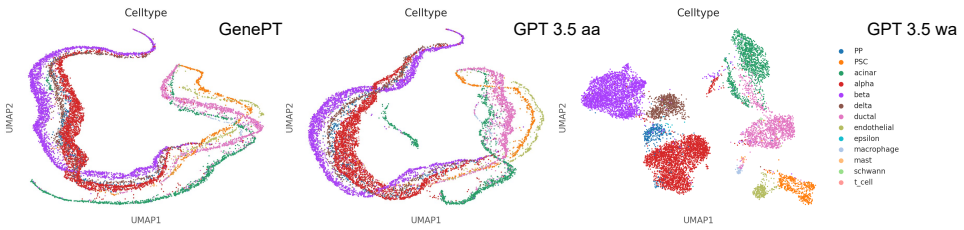
Extended Data Fig. 2 UMAPs for the visualization of gene functional information. (a) UMAPs for the genes with known functional information and unknown functional information. (b) UMAPs for the genes with annotated functional information based on a kNN classifier. For genes with multiple functional annotation, we combined the functions as a new label.



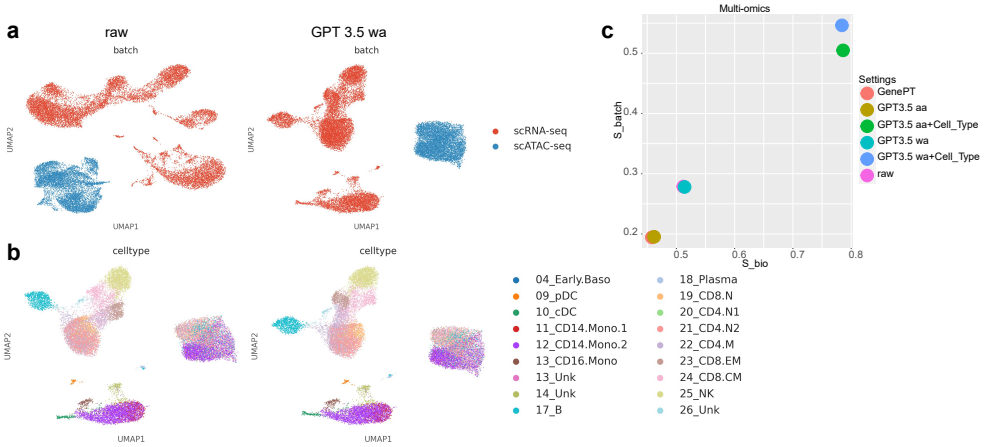
Extended Data Fig. 3 Average ranks for clustering and cell-type annotation. (a) Average-rank information for different methods across datasets. (b) The left panel represents the average-rank information of methods based on fine-tuning for cell-type annotation. The right panel represents the average-rank information of methods based on zero-shot learning for cell-type annotation.



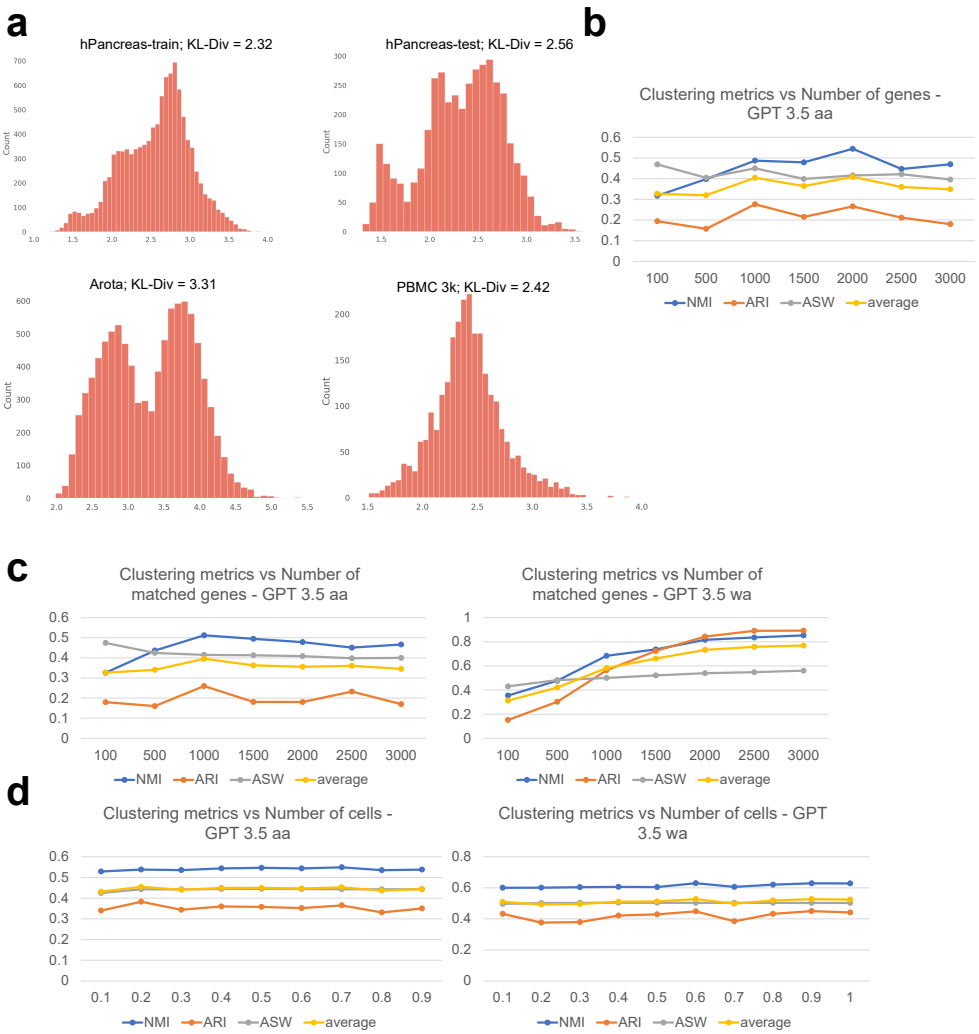
Extended Data Fig. 4 UMAPs for the results of CINEMA-OT under different input settings and datasets. (a) UMAPs visualization for the confounder space of CINEMA-OT under different methods based on the ChangYe2021 dataset. All cells in this dataset have the same cell type. (b) UMAPs visualization for the confounder space of CINEMA-OT under different methods based on perturbed PBMC dataset. The labels for the UMAPs include perturbation conditions (upper panel) and cell types (lower panel). (c) Plots for the gene synergy score distribution, labelled by different methods. (d) Plots for the gene synergy distribution across cell types, labelled by different methods.



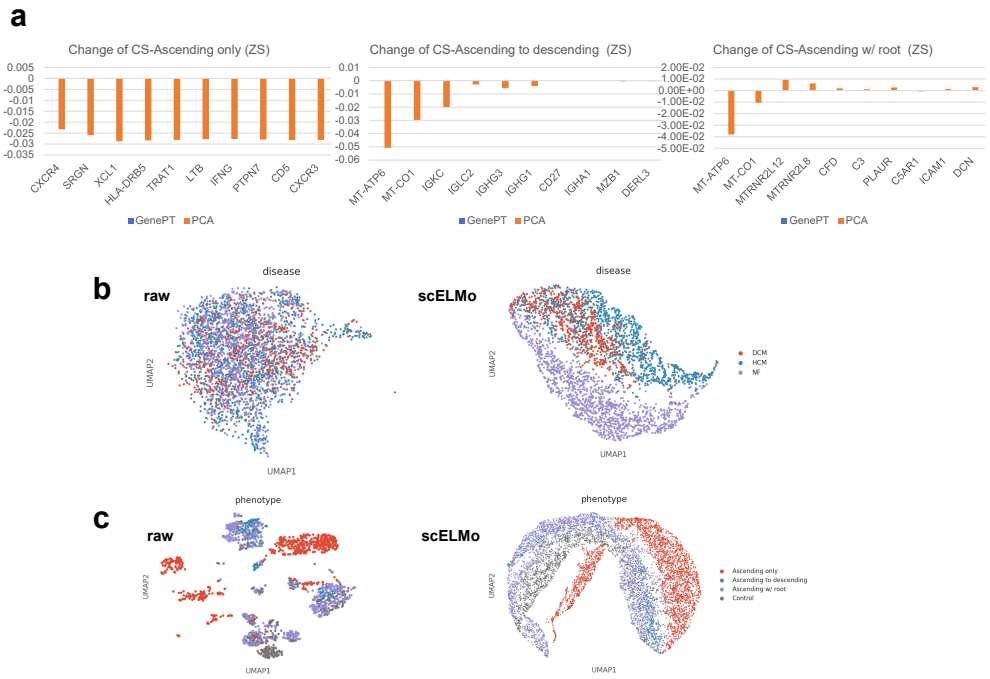
Extended Data Fig. 5 UMAPs for cell embeddings with different sources based on the hPancreas-train dataset. Each panel is colored by the cell types. (a) UMAPs for the cell embeddings based on GenePT. (b) UMAPS for the cell embeddings based on *aa* mode. (c) UMAPs for the cell embeddings based on *wa* mode.



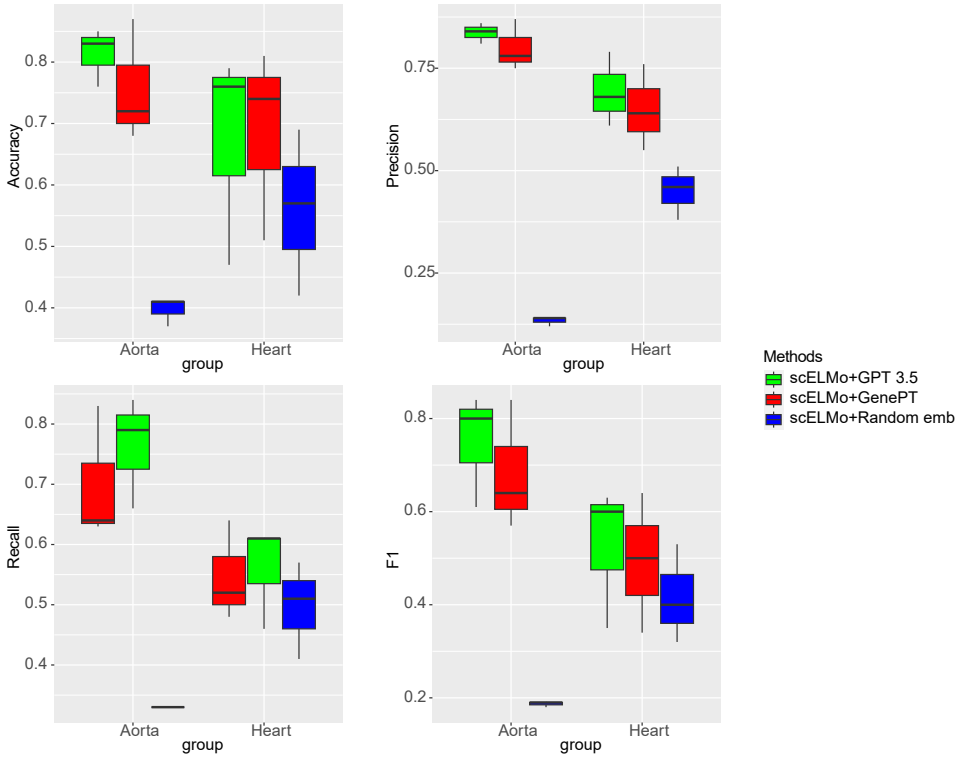
Extended Data Fig. 6 Results of batch effect correction for multi-omic data (scATAC-seq, scRNA-seq). (a) UMAPs of batch information for the raw data and cell embeddings from GPT 3.5 *wa* mode. (b) UMAPs of cell-type information for the raw data and cell embeddings from GPT 3.5 *wa* mode. (c) Evaluations of the batch effect correction for multi-omic datasets across different methods.



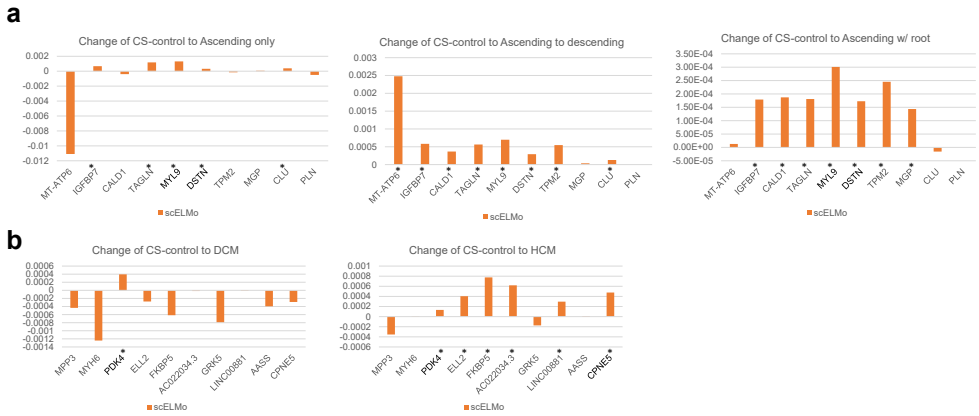
Extended Data Fig. 7 Results of our exploration for the cell clustering task. (a) KL-Div for the two distributions across different datasets. (b) The relation between the number of recorded genes and clustering metrics is based on the GPT 3.5 *aa* mode. (c) The relation between the number of matched genes and clustering metrics is based on the GPT 3.5 *aa* mode (left panel) and the GPT 3.5 *wa* mode (right panel). (d) The relation between the proportion of cells and clustering metrics is based on the GPT 3.5 *aa* mode (left panel) and the GPT 3.5 *wa* mode (right panel).



Extended Data Fig. 8 Change of CS under the zero-shot (ZS) learning framework and UMAPs for visualization. (a) The change of CS based on cell embeddings from GenePT or PCA for the Aorta dataset. We considered all three different disease states. (b) UMAPs visualization for the original gene expression space (left panel) and cell embeddings from finetuned scELMo (right panel) based on the Heart dataset. Figures are colored by cell conditions. (c) UMAPs visualization for the original gene expression space (left panel) and cell embeddings from finetuned scELMo (right panel) based on the Aorta dataset. Figures are colored by cell conditions.



Extended Data Fig. 9 Metrics for disease classification under different gene embeddings across the two datasets. Different panels represent values from different metrics, and we have four metrics in this task.



Extended Data Fig. 10 Change of CS for silencing DEGs in the control case. (a) The change of CS based on cell embeddings from scELMo for the Aorta dataset. (b) The change of CS based on the cell embeddings from scELMo for the Heart dataset. We highlighted the genes detected by both GenePT and scELMo using stars (*) and marked the genes that were discovered by previous research as genes related to disease pathway using **bold** type.