Large Scale Foundation Model on Single-cell Transcriptomics

Minsheng Hao[1,+], Jing Gong[2], Xin Zeng[2], Chiming Liu[2], Yucheng Guo[2], Xingyi Cheng[2], Taifeng Wang[2], Jianzhu Ma[3,4,*], Le Song[2,5,*], Xuegong Zhang[1,6,*]

1. MOE Key Laboratory of Bioinformatics and Bioinformatics Division, BNRIST, Department of Automation, Tsinghua University, Beijing, China
2. BioMap, Beijing, China
3. Department of Electrical Engineering, Tsinghua University, Beijing, China
4. Institute for AI Industry Research, Tsinghua University, Beijing, China
5. Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
6. School of Life Sciences and School of Medicine, Center for Synthetic and Systems Biology, Tsinghua University, Beijing, China

* Corresponding author: (Xuegong Zhang) zhangxg@tsinghua.edu.cn, (Le Song) songle@biomap.com, (Jianzhu Ma) majianzhu@tsinghua.edu.cn.

\+ Work was done while interning at BioMap.

Abstract

Large-scale pretrained models have become foundation models, leading to breakthroughs in natural language processing and related fields. Developing foundation models in life science, aimed at deciphering the "languages" of cells and facilitating biomedical research, is challenging yet promising. We developed a large-scale pretrained model, scFoundation, for this purpose. scFoundation was trained on over 50 million human single-cell transcriptomics data, which contain high-throughput observations on the complex molecular features in all known types of cells. scFoundation is currently the largest model in terms of the size of trainable parameters, dimensionality of genes and the number of cells used in the pre-training. Experiments showed that scFoundation can serve as a foundation model for single-cell transcriptomics and can achieve state-of-the-art performances in a diverse array of downstream tasks, such as gene expression enhancement, tissue drug response prediction, single-cell drug response classification, and single-cell perturbation prediction.

Introduction

Large-scale pretrained models are revolutionizing research in natural language processing related fields and becoming a new paradigm toward general artificial intelligence. These models trained on huge corpora becomes foundation models due to their fundamental importance in leading of breakthroughs in downstream tasks and their ability in discerning patterns and entity relationships within language. While in life science, the organism has different "languages". Cells, the basic structural and functional units of the human body, constitute these "sentences" composed of a myriad of "words" such as DNA, RNA, proteins, and gene expression values. The intriguing question is: Can we develop foundation models based on these biological "languages" for modeling complex cellular activities?

Single-cell RNA sequencing (scRNA-seq) data, also known as single-cell transcriptomics, offers high-throughput observations into cellular systems[1–3], making the first groundwork for developing foundation models of cells. In transcriptomic data, gene expression profiles depict a complex system of gene-gene co-

expression and interaction within cells. With the efforts of the Human Cell Atlas[4] and other studies, the data scale is exponentially growing[5]. Given the nearly 20,000 protein-coding genes across millions of cells, the observed gene expression values scale to magnitude of trillions, which is comparable to the volume of natural language text used to train models such as GPT. This paves the way for us to pre-train a large-scale model to extract complex, multifaceted internal patterns of cells, similar to how large language models (LLMs) learn human knowledge from extensive natural language texts.

In the LLMs pre-training[6,7], the growth in both model and data scale is critical for constructing a foundation model that can effectively mine intricate multi-level internal relationships. Despite pre-training models on single-cell gene expression data are receiving increased attention[8–10], the development of a large-scale foundation model still presents unique challenges. First, the gene expression pre-training data needs to encompass a landscape of cells across different statuses and types. Currently, scRNA-seq data are loosely organized, and a comprehensive database is lacking. Second, when modeling each cell as a sentence and each gene expression value as a word, the nearly 20,000 protein-coding genes make the "sentence" exceptionally long, a scenario that traditional transformers struggle to handle[11,12]. Third, scRNA-seq data across different sequencing techniques and laboratories exhibit high variance in sequencing read depth. This technical noise prohibits the model from learning uniform and meaningful representations for cells.

In this study, we address these challenges and pre-train the first large-scale foundational model, scFoundation. We collect the largest scRNA-seq data set with over 50 million gene expression profiles for pre-training, covering cells from different statuses and various tissues. We develop an asymmetric architecture designed for single-cell RNA-seq data to accelerate the training process and improve model scalability. We develop a read-depth-aware pre-training task that enables scFoundation not only to model the gene co-expression patterns within a cell but also to link the cells with different read-depth.

To verify the ability of scFoundation for learning the characteristics of both cells and genes, we conduct experiments on multiple downstream tasks, including gene expression enhancement, drug response prediction on bulk data, single-cell drug response classification, and single-cell perturbation prediction. scFoundation achieved state-of-the-art performance by adapting context embeddings to the corresponding downstream models. Our work reveals the efficacy and value of large-scale pre-trained models for transcriptomics data and demonstrates its foundation function in facilitating both biology and medical tasks learning. We explore and push the boundaries of the foundation model in the single-cell field.

Results

The scFoundation pre-training framework

We pre-trained a large-scale model, scFoundation, modeling 19,264 genes with 100 million parameters on over 50 million scRNA-seq data. To our knowledge, this is the largest model parameter size, gene coverage, and data scale in the single-cell field. The ability to efficiently train such a model benefited from our pre-training framework, which consists of three parts: model design, pre-training tasks, and data collection (Fig. 1A).

We developed xTrimoGene[13], a scalable transformer-based model with both algorithmically efficient and engineering acceleration strategies. It included an embedding module and an asymmetric encoder-decoder. The embedding module converted continuous gene expression scalars into learnable high-dimensional vectors, which were then used as the input of the encoder and decoder. This module fully retained information from the raw expression values, a notable improvement over the discretized values used in previous models[9,14]. The asymmetric encoder-decoder architecture was specifically designed to accommodate the high sparsity

characteristics of single-cell gene expression data. The encoder only accepted embeddings of the non-zero and non-masked expressed genes' as input, and the decoder accepted all genes' embedding (i.e. the encoder processed embeddings and zero and mask embeddings) as input. This architecture gave differential attention and computational resources to zero and non-zero values, thereby achieving the efficient learning of all genes relationships without any selection (e.g., highly variable gene selection). Moreover, the model deployment incorporated a variety of large-scale model training optimization techniques to ensure efficient training (Methods).

We developed a new pre-training task called the read-depth-aware (RDA) modeling, which was an extension of masked language modeling[15], considering single-cell gene expression data had a high variance in read depth, especially in the large-scale pre-training data. In RDA, we trained the model to predict the masked gene expression of a cell based on other genes' context. The context was from a counterpart or a low read-depth variant of that cell's gene expression profile. Specifically, we processed a raw gene expression training sample with a hierarchical Bayesian downsampling strategy to generate an input sample with an unchanged or altered total count (Methods). We treated the total count of all gene expressions as the measure of one cell's read depth and defined two total count indicators: T (representing 'target') and S (representing 'source'), corresponding to the total counts of the raw and input samples respectively. We randomly masked the expression values of genes in the input sample and record the index of masked genes. Then the model took the masked input sample and two indicators to predict the expression value of the raw sample at the masked index (Fig. 1B). A regression loss was conducted on the predicted and raw gene expression values. This pre-training process enabled the pre-trained model not only to capture the gene-gene relationship within the cell but also to harmonize the cell with different read depths. When used for inference, the number T could control the total count of the output gene expression profiles. Thus, we could feed the cell's raw gene expression to the pre-training model and set the T higher than its total count S to generate a gene expression value with enhanced sequencing depth.

We constructed a comprehensive single-cell gene expression data set by collecting data from various publicly available single-cell resources, including GEO[16], Single Cell Portal, HCA[4], EMBL-EBI[17], etc. We aligned all data to a consolidated gene list that comprised 19,264 protein-coding and common mitochondrial genes, as identified by the HGNC[18]. After preliminary data quality control (Methods), we got over 50 million human scRNA-seq data for pre-training. The abundant data sources made our pre-training dataset rich in biological patterns. From an anatomical perspective, the data covered more than 100 tissue types in different diseases, tumors, and normal states (Fig. 1A); From a cell ontology perspective, the data encompassed almost all known human cell types and cell states.

After pre-training, we applied the scFoundation model to several downstream tasks (Fig. 1C). The outputs of the scFoundation encoder were pooled into cellular embeddings, which were used for cell clustering and drug response prediction. The outputs of the scFoundation decoder were gene-level context embeddings, which were used for perturbation prediction.

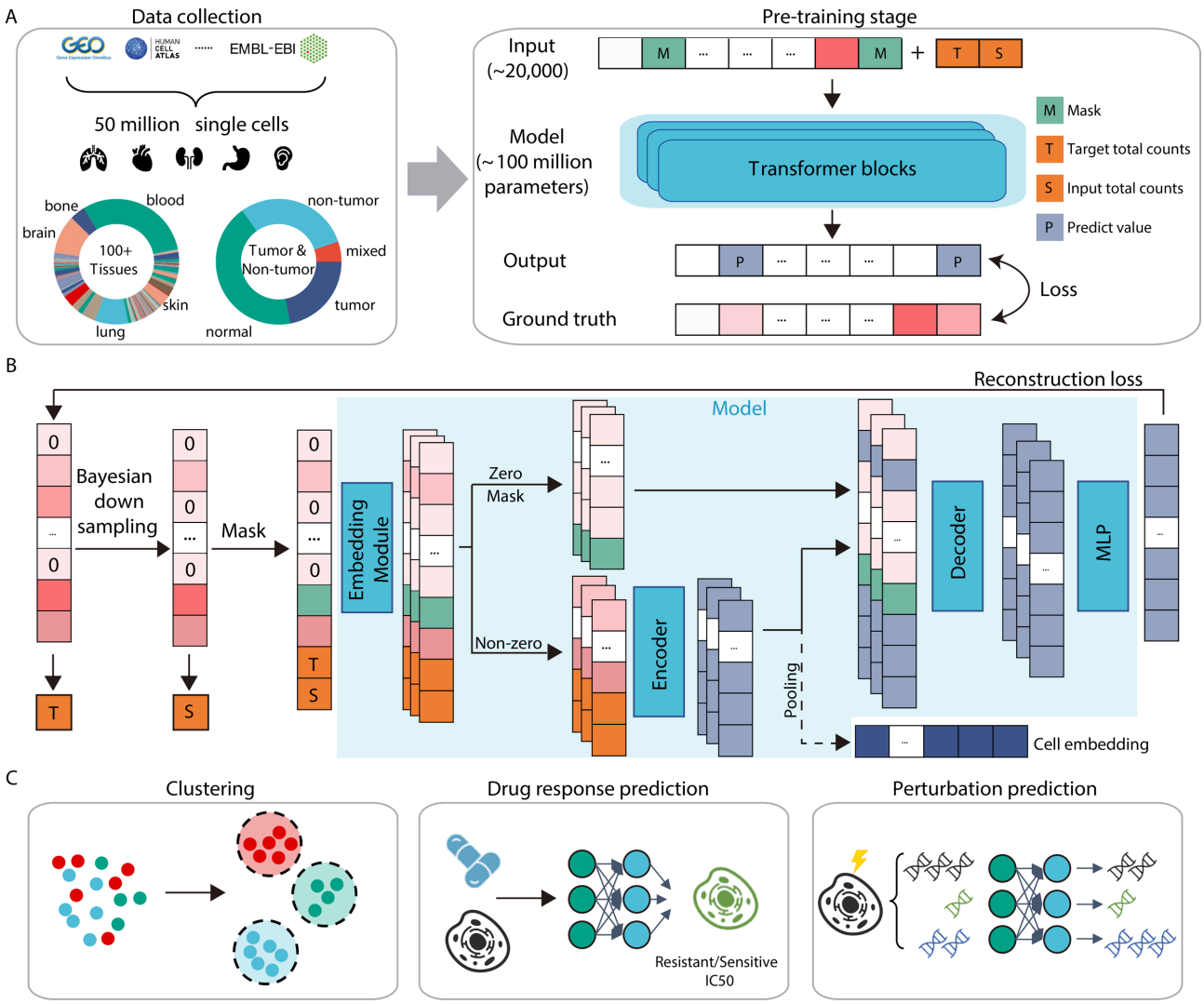

Figure 1. The schematic overview of the pre-training framework. A) 50 million single-cell gene expression profiles were collected, covering tumor and non-tumor cells from various tissues. These data were used for the read-depth-aware (RDA) task to pre-train the model. In the RDA task, the input consists of the masked gene expression vector and two total count indicators (T and S). The output is the predicted expression value for all genes, and the loss is computed at the masked positions. B) Outline of the pre-training process. A raw gene expression vector serves as a training sample. A hierarchical Bayesian downsampling strategy generates the input sample. The gene expression total count values (T and S) of the raw and input samples are computed. Values in the input sample are randomly masked. The scalar values are converted into embeddings. Only embeddings corresponding to non-zero and non-masked values (including T and S) are fed into the model encoder. The output embeddings of the encoder are then combined with mask and zero embeddings and fed into the decoder. Also, the encoder output can be pooled to generate a cell embedding for downstream usage. The decoder output embeddings are projected to the gene expression value via a shared multilayer perceptron (MLP) layer. The regression loss between the predicted and raw sample's gene expression values is computed. C) The pre-training embeddings can be leveraged as substitutes for the gene expression profiles, facilitating read depth enhanced cell clustering, drug response prediction, and perturbation prediction.

scFoundation is a scalable model for read-depth enhancement without fine-tuning

In large-scale language models, researchers observed a power-law relationship between the loss and various factors, including model size, the amount of computation utilized during training, etc[6,19]. This relationship, commonly referred to as the scaling law, was also confirmed in our framework. Fig. 2A illustrated the training of three models with parameter sizes of 3 million, 10 million, and 100 million. It was found that as the model parameters and the total number of floating point operations (FLOPs) increased, the loss on the validation dataset exhibited a power-law decline. Our scFoundation model, with 100 million parameters, surpassed all

previous models[8–10,14] in terms of parameter size and achieved the lowest reconstruction loss.

Our RDA modeling enables scFoundation to enhance the read depth of the input cell by setting T as a higher number than S. We first assessed this ability on an independent test dataset. We downsampled the total count into 1%, 5%, 10%, and 20% of the original profiles, generating four corresponding datasets with varying total count fold changes. For each dataset, we didn't finetune the scFoundation but directly utilized it to enhance the cells with a low total count by setting the desired total count T as the reciprocal of the sample rate. We computed the mean absolute error (MAE) and mean relative error (MRE) metric between the predicted and ground-truth values of non-zero expressed genes (refer to Methods). Additionally, we evaluated the Pearson Correlation Coefficient (PCC) of gene expression between the enhanced and ground-truth data. We employed the downsampled gene expression values as the baseline for comparison. As depicted in Fig. 2B and Fig. S1, scFoundation demonstrated a significant reduction of half the MAE and MRE compared to the non-enhanced baseline. Particularly, when the downsampling rate was below 10%, the baseline results exhibited a 97.5% MRE, indicating that most of the non-zero values were sampled as zeros. However, scFoundation consistently maintained an MRE of 60%. As the downsampling rates increased, the baseline results showed a lower error and a higher PCC, yet scFoundation continued to outperform the baseline. These findings highlight the ability of scFoundation to enhance gene expressions in scenarios even with extremely low total counts, and robust to be generalized to new datasets.

Clustering is another common analysis to validate the performance of read-depth enhancement methods. We compared scFoundation with imputation methods including MAGIC[20], SAVER[21], scImpute[22], and scVI[23] on a human pancreatic islet dataset[24] processed by SAVER. This dataset comprised manually generated downsampled gene expression profiles and their corresponding reference data. For scFoundation, we set T as the different folds of S (ranging from 1 to 5), fed the downsampled data, and obtained the five sets of cell embeddings. For other methods, we first used the downsampled data to train the methods. Then we got cell embeddings from scVI and got the imputed gene expression from other methods. The ground truth cluster labels were obtained from the reference data. For evaluating clustering accuracy, we employed metrics including Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Silhouette Coefficient (SIL). The clustering performance obtained from the downsampled data was used as the baseline.

We found that CPT already outperformed both the baseline and scImpute in all metrics when T was equal to S (fold change=1, Fig. 2C), indicating that the scFoundation embeddings provided more informative representations compared to the raw gene expression. As the fold change increased, we observed an initial "jump" in scFoundation's performance, surpassing the other methods, followed by a continuous improvement in all metrics. We generated Uniform Manifold Approximation and Projection (UMAP) plots for visualizing the scFoundation embedding results at a fold change of 5 and the results obtained from other methods (Fig. 2D). Notably, scFoundation's cell embeddings exhibited distinct cluster boundaries compared to the baselines and other methods. Furthermore, we performed clustering on all methods' results and mapped the cluster labels onto the reference UMAP. The labels produced by the other methods were mixed in the reference, particularly within the ground truth cluster 0. scFoundation stood out as the only method that visually demonstrated aligned cell cluster assignments consistent with the reference results.

We then applied scFoundation to the Zheng68K dataset from human peripheral blood mononuclear cells (PBMC)[25]. This dataset had about 600,000 cells, comprised of cell types that exhibited high similarity and were sequenced using the first version of the 10X Chromium platform. Each cell in this dataset only had non-zero expression values for approximately 500 genes, and the total count was less than 2000 (Fig. S2). We fed these cells into scFoundation to obtain the read depth enhanced cell embeddings by setting T as 10,000.

Comparing the UMAP plot of the scFoundation embeddings with that of the raw gene expression profiles, we observed that the scFoundation embeddings made it easier to distinguish Memory T cells from other T cell populations and exhibited improved discrimination between CD14 Monocytes and CD34 cells. To further evaluate the clustering performance with other methods, we trained the only scalable model scVI on this dataset and compared it with the scFoundation embeddings (Fig. 2F). Both clustering results outperformed the raw gene expression profiles and achieved comparable performance in terms of NMI and ARI metrics, and the scFoundation embeddings had a higher SIL score.

These findings demonstrated that scFoundation possessed the capability to enhance the read depth of cells. Notably, an important distinction between scFoundation and other imputation methods was that scFoundation could achieve the best performance without the need for dataset-dependent training procedures.

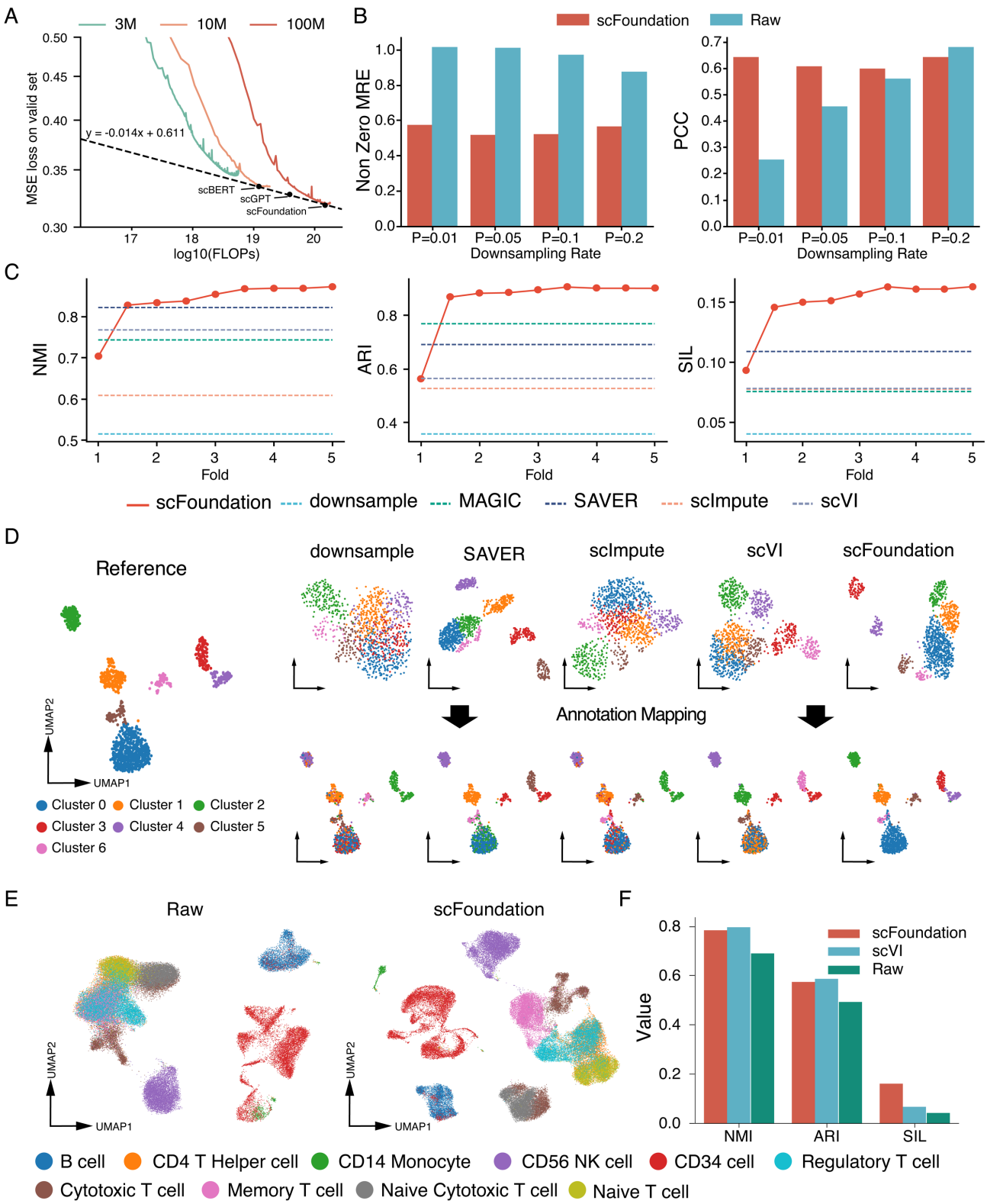

Figure 2 A) Training loss under different parameter sizes and FLOPs. B) Evaluation of read depth enhancement performance on an unseen dataset. Mean relative error among non-zero expressed genes and Pearson correlation among all genes were used as metrics to evaluate the recovered gene expression performance. Lower error values indicate better performance, while higher Pearson correlation values indicate better performance. C) Comparison of the scFoundation model with other imputation methods based on cell clustering metrics. The x-axis represents the fold change between the desired total count value and the input total count value, while the y-axis represents the score. D) UMAP plots of cell embeddings generated by different methods. The left plot shows the reference UMAP plot obtained using raw gene expression, with colors indicating cell clusters. The upper-right plots display clustering results obtained by different methods (Sample, MAGIC, scImpute, scVI, and scFoundation). "Sample" refers to no imputation process. The number of clusters is aligned. The lower-right plots depict the clustering results of each method mapped onto the reference UMAP

plot. E) UMAP plot comparing raw gene expression and scFoundation-imputed cell embeddings on the Zheng68K dataset. F) Comparison of clustering performance among scFoundation, scVI, and raw data on the Zheng68K dataset.

scFoundation improves cancer drug response prediction

Cancer Drug Responses (CDRs) study the tumor cell response upon drug intervention. Computationally predicting CDR is critical to guiding anticancer drug design and understanding cancer biology[26]. We combined scFoundation with a CDR prediction method, DeepCDR[27], to predict the half-maximal inhibitory concentrations IC50 values of drugs across several cell line data. This experiment served as a validation of whether scFoundation could provide informative embeddings for bulk-level gene expression data, despite being trained on single cells.

The original DeepCDR model used drug structural information and multi-omics data as input and outputted the predicted IC50. Here, we focused on gene expression data and replaced the transcriptome MLP subnetwork in DeepCDR with scFoundation. In other words, we used scFoundation to extract transcriptome features and fed them into the subsequent prediction module (Fig. 3A). We integrated the Cancer Cell Line Encyclopedia (CCLE)[28] and Genomics of Cancer Drug Sensitivity (GDSC)[29] datasets to obtain the input cell line gene expression data, the input drugs and IC50 labels.

We evaluated the performance of scFoundation-based results and gene expression-based results across multiple drugs and cell lines (Fig. 3B). Most drugs and all cancer types achieved a higher PCC by using scFoundation embeddings. We further visualized the best prediction case of drug and cancer types (Fig. 3C). Regardless of the high or low lC50, the scFoundation embedding-based DeepCDR model could predict accurate values and achieved a PCC above 0.93.

The CDR prediction models were often used to predict the IC50 of the new unseen compound across the cell line. For testing the generalization ability to new drugs, we then left one drug-related dataset at a time as the test set and used the remaining data as the training set. Under this drug-blind test, we found that models based on scFoundation embeddings consistently outperformed the original model. The top 1 PCC-gaining drug, PHA-793887, was a potent ATP-competitive CDK inhibitor, and its PCC improved from 0.07 to 0.73. Even for the 200th-ranked drug, Zobotentan, which was used for blocking Endothelin A receptor activity, its PCC improved from 0.49 to 0.64.

We further grouped drugs into different therapy types to examine whether the IC50 prediction performance was related to their intrinsic mechanisms. We observed that, based on scFoundation-predicted results, drugs belonging to chemotherapy, such as anti-tumor antibiotics and topoisomerase inhibitors, had a higher PCC than drugs belonging to targeted therapy, such as ATM and PARP inhibitors (Fig. 3C right). This may be due to the fact that specific gene mutations often have an important impact on targeted therapy[26], but mutation information is difficult to reveal from gene expression data; while chemotherapy drugs were widely reported to be related to gene expression[30,31] so their IC50 is easier to predict. Gene expression-based results had an overall lower PCC, and we did not observe a significant difference between drug types.

Then we used our model to predict unknown CDR in the data, and thus each drug would have a predicted IC50 on all cell lines. To validate these new predictions, we performed a Gene Set Enrichment Analysis (GSEA)[32] on the new predictions with relatively low IC50, which indicated the sensitivity of the cell line to the drug. For instance, the sphingolipid signaling pathway was enriched in Doxorubicin-sensitive cell lines. According to the KEGG database[33], this pathway was related to sphingomyelin (SM) and its metabolism. And SM was reported to interact synergistically with Doxorubicin by altering cell membrane permeability resulting in a lower IC50 of the drug in these cell lines[34]. Additionally, the mTOR signaling pathway was enriched in

Vorinostat-sensitive cell lines. Previous studies have shown that Vorinostat inhibits carcinoma growth by dampening the mTOR signaling pathway[35]. And other clinical studies have also shown that mTOR inhibitors were often used in conjunction with Vorinostat[36,37], suggesting the relationship between Vorinostat and the mTOR pathway. These examples support the validity of our predictions.

Our results demonstrated that although the CPT model was pre-trained on single-cell transcriptomics data, the learnt gene relationships were transferable to bulk-level expression data to produce condensed embeddings, facilitating more accurate IC50 prediction. These findings illustrated the potential of scFoundation in expanding our understanding of drug responses in cancer biology and possibly guiding the design of more effective anticancer treatments.

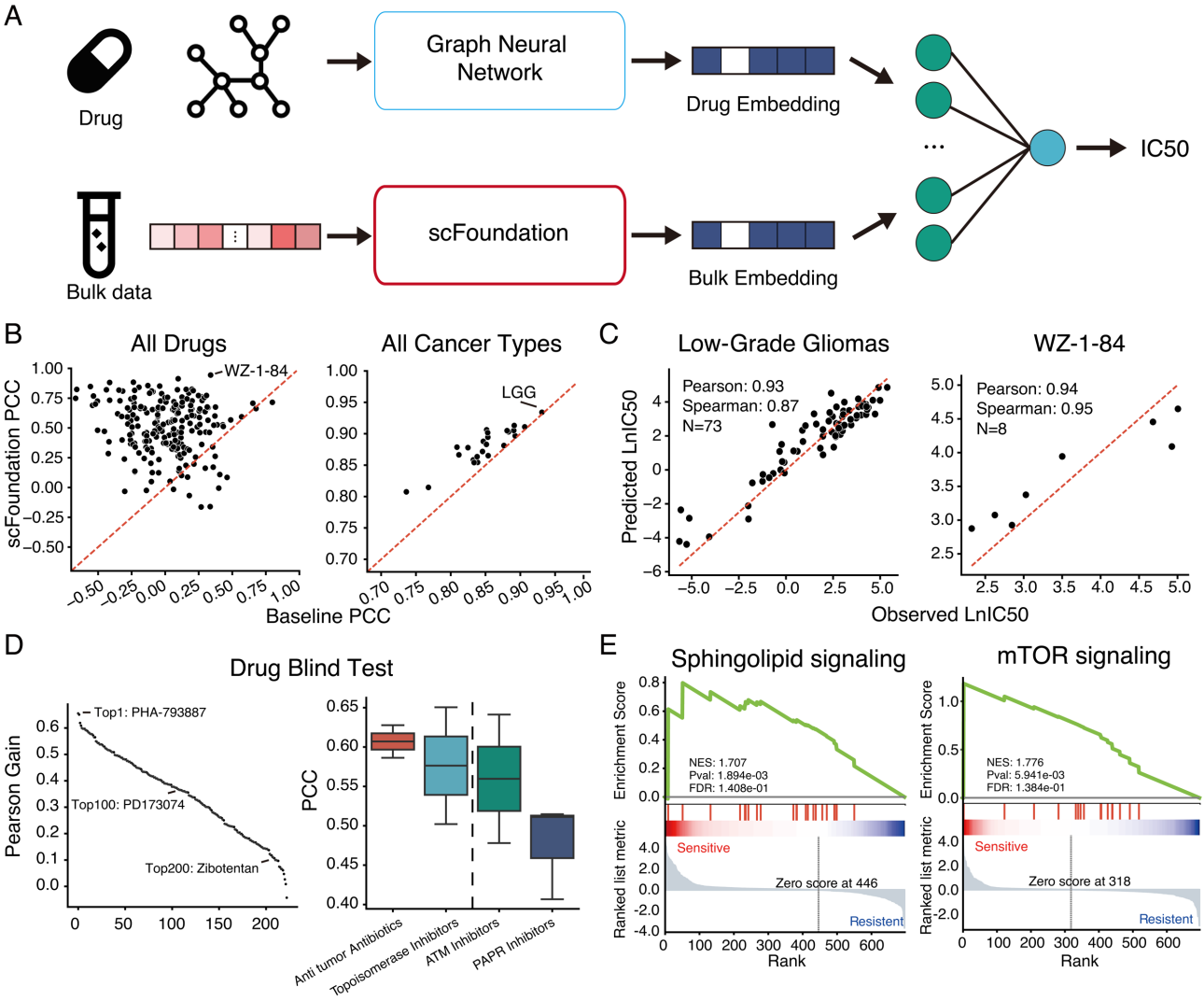

Figure 3. Performance of drug response prediction using scFoundation embeddings. A) Illustration of the embedding-based CDR prediction model. B) Pearson correlation coefficient (PCC) between all drugs and cancer types in the test set. Each dot represents a drug or cancer type, with the x-axis and y-axis showing the PCC obtained by DeepCDR using gene expression and scFoundation embeddings, respectively. C) Comparison of predicted and observed IC50 values for the drug WZ-1-84 on the cancer-type low-grade gliomas. Each dot represents a drug and cell line combination. D) Leave-one-drug-out blind test performance. The left plot shows the PCC gain obtained by replacing gene expression with cellular embeddings. Each dot represents a drug, with the y-axis indicating the gained PCC values and the x-axis representing the rank. Higher-ranked drugs have a higher PCC gain. The right plot groups the drugs into different types. E) Gene set enrichment analysis (GSEA) results on cell line data with lower predicted IC50 values. The Sphingolipid signaling pathway was enriched in Doxorubicin-sensitive cell lines, while the mTOR signaling pathway was enriched in Vorinostat-sensitive cell lines.

scFoundation transfers bulk-level drug sensitivity prediction model to single-cell data

Intratumor heterogeneity is a well-known issue that often leads to treatment failure[38,39]. Inference of drug sensitivities at the single-cell level can help identify specific cell subtypes that exhibit different drug resistance characteristics, offering valuable insights into underlying mechanisms and potential new therapies[40–42]. We applied scFoundation to tackle this crucial task of single-cell level drug response classification based on the SCAD[43] downstream model. Due to the fact that the limited availability of cellular-level drug response data for only a small subset of drugs and cancer types, SCAD was trained on the bulk-level datasets to learn pharmacogenomic information and then transferred this knowledge to infer drug efficacy at the single-cell level. In other words, it took both bulk and single-cell data, affected by the same drug, as input and provided sensitive or non-sensitive labels for each cell. In our study, we used scFoundation to obtain unified embeddings of bulk and single-cell data (Fig. 4A). Subsequently, we used these embeddings to train SCAD models and assessed the classification performance.

We focused on four drugs (Sorafenib, NVP-TAE684, PLX4720, and Etoposide) that exhibited lower AUROC values as reported in the original study. These drugs had drug-sensitive labels of bulk data in the GDSC[29] database and the cell-level drug-sensitive labels were obtained in different ways. For the drug PLX4720 and Etoposide-affected single cells, cells from drug-untreated cell lines were considered sensitive, while cells that survived after drug exposure were considered resistant[44]. For drug Sorafenib and NVP-TAE684 affected cells, the cells' sensitive labels were determined by the value of senescence-related (EpiSen) program scores which were proved to have a relation with drug responses previously[45].

We compared our results with the baseline SCAD model, which took all genes' expression values as input. We found that the scFoundation embedding-based model achieved higher AUROC values for all four drugs (Fig. 4B). Particularly for drugs NVP-TAE684 and Sorafenib, the scFoundation embedding-based model exhibited an improvement of above 0.2 in AUROC. We then employed the Spearman correlation as a metric to assess the agreement between the predicted probability of drug sensitivity provided by the model and the EpiSen score. As depicted in Fig. 4C, drug sensitivity for NVP-TAE684 and Sorafenib was positively and negatively correlated with EpiSen scores, respectively. The scFoundation embedding-based model yielded a Spearman correlation of 0.56 and -0.55 for the two drugs, while the baseline model achieved a Spearman correlation of only 0.24 and -0.06. These findings indicated that the embedding obtained from scFoundation facilitated drug sensitivity prediction in the original single-cell dataset with poor performance, and the embedding-based SCAD model had the potential to capture the signal of drug sensitivity biomarkers.

These results further motivated us to investigate whether the embedding itself was more informative than gene expression without the necessity for the SCAD model to extract the signal. We conducted principal component analysis (PCA) on the single-cell dataset SSC47, corresponding to drugs NVP-TAE684 and Sorafenib. By visualizing cells with different EpiSen scores on the first two principal components (PC) plots (Fig. 4D), we observed that compared to gene expression data, the first two PCs of the embeddings displayed less linear correlation, indicating the provision of richer information. Additionally, we used drug sensitivity as the label and computed the clustering performance of the embeddings and gene expression for both bulk and single-cell data. Calinski-Harabasz (CH) and SIL scores were used as metrics. The results for both single-cell and bulk data (Fig. 4E and Fig. S3) demonstrated that the scFoundation embedding better-grouped cells or bulk cell lines with the same drug response, compared to the gene expression baseline.

Overall, these findings highlighted that the unified embedding obtained from scFoundation aligned bulk and single-cell data into a unified representation space, and this condensed representation effectively facilitated the

transfer of pharmacogenomics information from bulk cell lines to single-cell data.

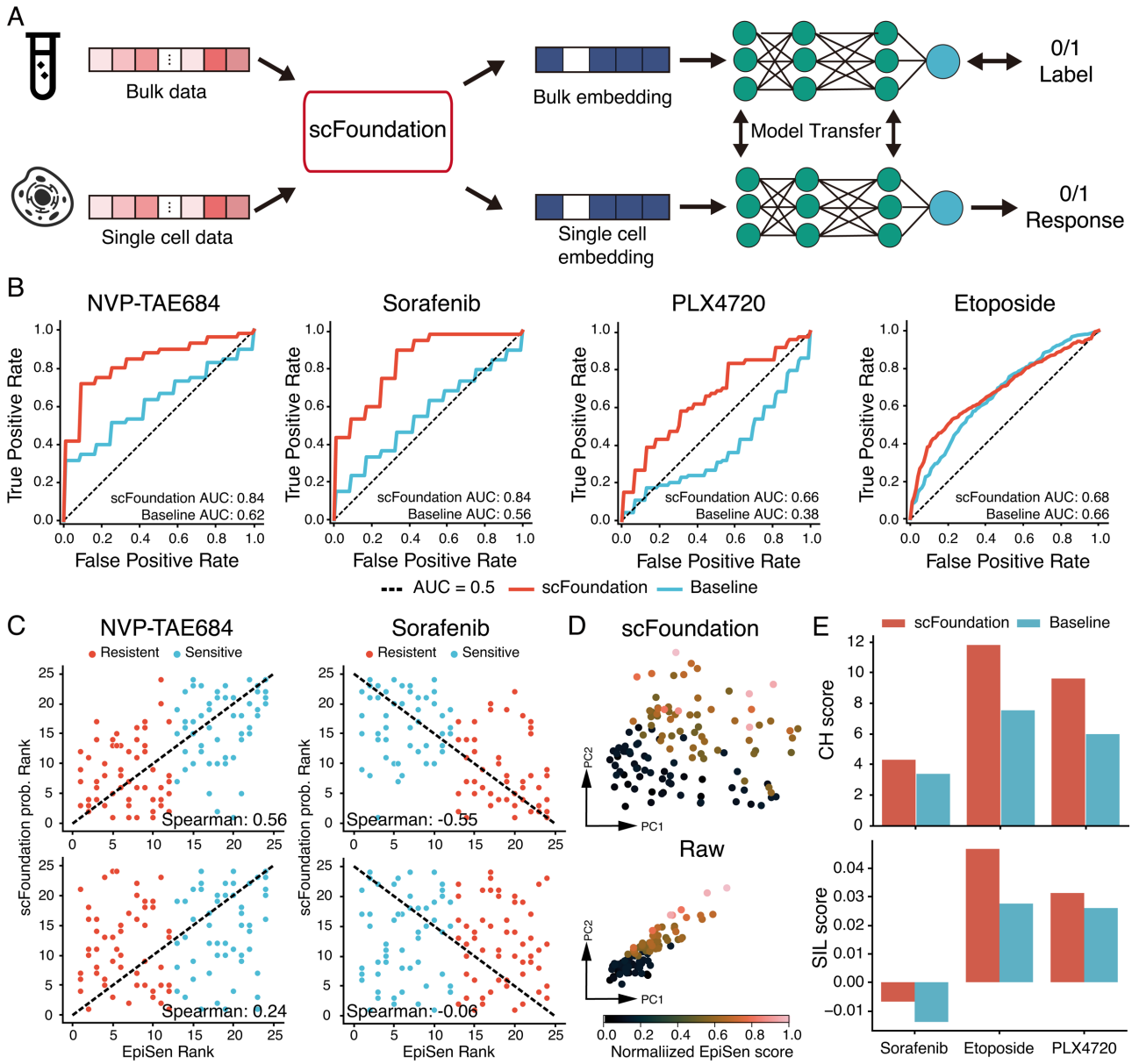

Figure 4. Single-cell drug response classification tasks based on scFoundation cell embeddings. A) Illustration of the embedding-based single-cell response classification model. B) Receiver operating characteristic (ROC) curves for the four drugs. The red and blue lines represent the performance of SCAD using scFoundation embeddings and gene expressions, respectively. C) Correlation between drug sensitivity probability and normalized EpiSen score. Each row corresponds to a model, and each column represents a drug. D) PCA plot of the SSC47 single-cell dataset from cell lines. Color denotes the reference epithelial senescence-related program (EpiSen) score. Cells with different EpiSen scores exhibit distinct responses to drugs. E) Clustering performance on all three drugs-related bulk datasets. Each bulk dataset has two types of labels: sensitive and resistant. CH score: Calinski-Harabasz score. SIL score: Silhouette Coefficient score.

scFoundation predicts more accurate perturbation responses

Cellular response to perturbation is crucial for biomedical applications and drug design, as it helps identify gene-gene interactions across different cell types and potential drug targets[46,47]. Perturb-seq[48,49] was a recently developed method for screening single-cell gene expression responses to several perturbations. Using these perturbation data resources to train models for modeling cellular response to perturbations is a central goal of computational biology[50–52].

We combined the scFoundation with an advanced model GEARS[50] for the perturbation prediction task. In the original GEARS model, a gene co-expression graph was combined with perturbation information to predict the post-perturbation gene expression. Each node in the co-expression graph represented a gene, with initially randomized embeddings, and edges connected co-expressed genes. This graph was not cell-specific and was shared across all cells. In our setting, we obtained gene context embeddings for each cell from the scFoundation decoder and set these embeddings as the nodes in the graph (Method). This resulted in a cell-specific gene co-expression graph for predicting perturbations (Fig. 5A).

We trained and tested models on three perturbation datasets which were also used in the original study: the Adamson dataset[48] with 87 1-gene perturbations, the Dixit[49] dataset with 24 1-gene perturbations, and the Norman[53] dataset with 131 2-gene perturbations and 105 1-gene perturbations. We reprocessed the gene numbers to the full gene length of 19,264, ensuring a comprehensive modeling of gene co-expression. For the 1-gene perturbation datasets, we leave a subset of perturbations for testing. For the 2-gene perturbation Norman dataset, we followed the original study to group 3 2-gene perturbations subsets for testing: 0/2, 1/2 and 2/2, where the first number indicated how many perturbations were not seen in the training set.

We computed the mean square error (MSE) of the top 20 differentially expressed (DE) genes between pre- and post-gene expression profiles to evaluate the models' performance. The scFoundation-combined model achieved lower MSE values on all 1-gene perturbation datasets compared to the original GEARS baseline model (Fig. 5B and Fig. S4). For 2-gene perturbations, the model achieved the lowest MSE in the 0/2 unseen case and outperformed across all cases. We further examined the proportion of predicted values that fell within (+-5%) of the true mean expression value of the top 20 DE genes. The results in Fig. 5C demonstrated that the scFoundation-based model exhibited a higher percentage close to 10%, indicating that it provided a more reasonable distribution of post-gene expression values. We then showcased the 2-gene perturbation ETS2+CEBPE in Fig. 5D for an instance of prediction. The scFoundation-based model achieved a more accurate mean post-expression value and a distribution that closely aligned with the ground truth, outperforming the baseline model.

One application for predicting 2-gene perturbations was to classify 2-gene perturbation into different genetic interaction (GI) types. Here, we identified synergy and suppressor GI types by using the magnitude score (Methods). We first computed the PCC between predicted and ground truth magnitude scores of all test set 2-gene perturbations and found that the scFoundation-combined model achieved a higher PCC compared to the baseline (Fig. 5E). Next, we ranked the 2-gene perturbations according to the predicted magnitude score and regarded the top 20 and bottom 20 perturbations as potential synergy and suppressor GI types, respectively. The Venn plot in Fig. 5F revealed that the scFoundation-based model identified a higher number of true perturbations for both synergy and suppressor types.

These results highlighted the cell-specific gene context embeddings obtained from the scFoundation model served as valuable foundational representations for perturbation prediction. Furthermore, the analysis of 2-gene perturbations underscored the model's capability to accurately classify different types of genetic interactions.

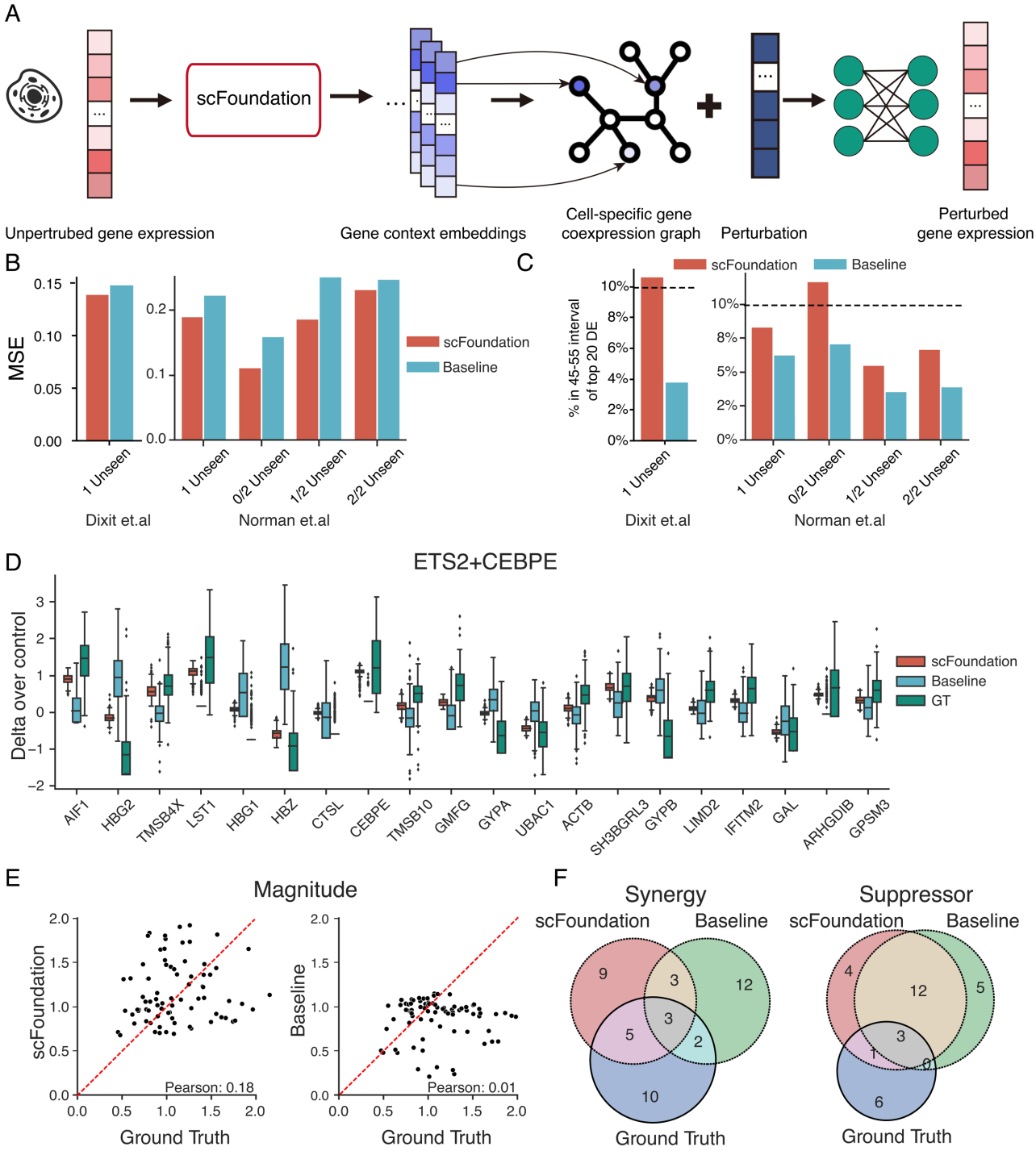

Figure 5. Perturbation prediction tasks using scFoundation gene context embeddings. A) Illustration of the perturbation prediction model based on cell-specific gene embeddings. B) Mean square error between predicted and ground truth post-gene expressions. Results given by the GEARS model using scFoundation cell-specific gene embeddings and gene expression are shown in red and blue, respectively. C) Proportion of predicted values within (+-5%) of the true mean expression value of the top 20 differentially expressed (DE) genes. The black dashed line represents the expected percentage (10%). Colors represent the same results as in panel A. D) Predicted gene expression over control for the top 20 most differentially expressed genes after a combinatorial perturbation (ETS2+CEBPE). The red and blue boxes indicate the gene prediction results by the GEARS model using scFoundation gene embeddings and pre-gene expression, respectively. The green box represents the ground truth post-gene distribution. E) Magnitude scores computed for all test perturbing combinations on the Norman dataset. Each dot represents a specific perturbing combination. The y-axis shows the magnitude score computed from the prediction results, while the x-axis represents the ground truth magnitude score computed using real post-gene expression. F) Top 20 perturbations with synergistic and suppressor gene interaction types identified using scFoundation and baseline methods. The Venn plot illustrates the relationship between the identified perturbation set and the verified perturbation set.

Discussion

Recent breakthroughs in large-scale models demonstrated their ability to grasp complex natural language patterns through self-supervised pre-training. This motivated us to explore whether large-scale models can also be effective for life science "languages", especially for single-cell transcriptomic data, which exhibits large data scales, complex biological patterns, diversity, and technical noise.

We developed the xTrimogene architecture to enable non-destructive modeling of all protein-coding genes and continuous expression values in cells. Combining this architecture with our read depth-aware pre-training task, we presented scFoundation, the largest pre-training foundation model on the single-cell field, with 100 million parameters pre-trained on over 50 million single-cell data.

Our experiments showcased the model's remarkable capabilities across various tasks. The proposed pre-training framework scaled up effectively, surpassing other models in terms of parameter size. scFoundation model enhanced read depth for gene expression without requiring additional fine-tuning, as supported by superior gene expression restoration and cell clustering performance compared to other imputation methods. In the Cancer Drug Response task, the cellular embeddings derived from scFoundation significantly improved the prediction accuracy of drugs' IC50 values. In the single-cell drug sensitivity task, scFoundation effectively integrated the unique characteristics of both bulk and single-cell data, resulting in superior performance in drug prediction. In the perturbation prediction task, our model overcame the limitations of prior approaches by constructing cell-specific gene co-expression maps, yielding the most accurate perturbation predictions to date.

It is worth noting that scFoundation followed a similar training approach to the linear probing[54] in all tasks, which added a task-specific predicting head after the pre-trained model output. The pre-trained model itself was not further fine-tuned, but the downstream task model was trained based on the obtained cell or gene context embeddings. This usage reduced computational and time costs for downstream users and offered flexibility in downstream model design. By adopting this paradigm, we envisioned scFoundation serving as a foundational model in the field of single cells, benefiting researchers and enabling further advancements.

In the future, we will pre-train models with more parameters and larger datasets using our effective pre-training framework. Additionally, the single-cell multi-omics data[55,56] opens up new avenues for modeling cellular regulatory relationships and cell state transitions[57]. These data can be used to train multi-omics large-scale models and provide deeper insights into cellular biological processes, towards modeling the multi-level complex laws of cells.

We expect that our pre-training architecture and the scFoundation model can serve as fundamental contributions supporting both biological large-scale model pre-training and downstream analysis research. We hold high expectations that single-cell pre-trained models will become indispensable tools in driving biological discovery and medical research.

Methods

Pre-training Data collection and preprocessing

Data collection

The human scRNA-seq public data were stored in the Gene Expression Omnibus (GEO) repository, human cell atlas, Single Cell Portal, EMBL-EBI, etc. There were also several studies to integrate human single cells from multiple resources, such as hECA[58], DISCO[59], etc. Each dataset in these databases was linked to a published study and thus had a corresponding DOI id. We manually collected scRNA-seq data from these

databases and removed the dataset with duplicated id. Most of the datasets provide the raw count matrix. For the dataset with normalized expression profiles, we converted them back to the raw count form: We treated the smallest non-zero value in the original matrix as a raw count value of 1, all remaining non-zero values were divided by this smallest value and the integer part was taken. For the dataset with TPM or FKPM expression profiles that cannot be converted back to raw counts, we kept them unchanged.

We collected more than 50 million single cells across multiple organs (e.g. heart, kidney, brain) and tissues (i.e. connective tissues, epithelial tissue, muscle tissue, and nerve tissue). Our database also included cells from various diseases and various cancer dissection regions of different cancer types, aiming to cover all known possible gene expression profiles of the human single cells.

Gene symbol unification

We unified the gene symbols of all raw count gene expression matrices by using the gene symbol mapping reference provided by HUGO Gene Nomenclature Committee. We included human protein-coding genes and common mitochondrial genes, constituting a total of 19,264 genes. If some symbols were missing, we padded them with zero values.

Quality control

To filter extremely low-quality or damaged cells, we kept cells with over 200 genes expressed (i.e., expression vector with non-zero value count > 200) for pre-training by using the Seurat[60] and Scanpy[61] packages.

scFoundation model architecture

We developed the xTrimoGene model as the backbone model of scFoundation. It had three modules: The embedding module converted scalar value into embeddings that were required for the transformer block; The encoder focused on the informative non-zero and non-masked expressed genes; And the decoder integrated information across all genes.

Embedding module

Given a cell's gene expression value vector $\boldsymbol{X}^{\text{input}} \in \mathbb{R}^{n=19264}$, the expression value $\boldsymbol{X}_i^{\text{input}}$ of gene i was a continuous scalar greater than or equal to zero. Unlike the previous language or recently developed single-cell transformer-based model, the embedding module directly converted the scalars into learnable value embeddings without any discretization. And then the value embeddings were added with gene embeddings to form the final input embeddings.

Specifically, the zero and non-zero values were converted in different ways. Zero value was directly converted into a randomly initialized embedding (with dimension d):

$$0 \mapsto \boldsymbol{E}^0 \quad \boldsymbol{E}^0 \in \mathbb{R}^d$$

For non-zero expression value, a look-up table $\boldsymbol{T} \in \mathbb{R}^{b \times d}$ was randomly initialized, where d was the hidden embedding dimension and b was the pre-defined retrieved token number. We used $b = 100$ for our scFoundation model. Then for a non-zero expression value $\boldsymbol{X}_i^{\text{input}}$ of gene i, it was first fed into a linear layer with leaky ReLU activation to get an intermediate embedding $\boldsymbol{v_1}$:

$$\boldsymbol{v_1} = \text{leakyReLU}\left(\boldsymbol{X}_i^{\text{input}} \times \boldsymbol{w_1}\right)$$

where $\boldsymbol{w_1} \in \mathbb{R}^{1 \times b}$ was the parameter. Then another linear layer and a scaling factor α were used to further process $\boldsymbol{v_1}$:

$$\boldsymbol{v_2} = \boldsymbol{v_1} \cdot \boldsymbol{w_2} + \alpha \times \boldsymbol{v_1}$$

where $\boldsymbol{w_2} \in \mathbb{R}^{b\times b}$ and α were the learnable parameters. And the obtained vector $\boldsymbol{v_2}$ was normalized with the SoftMax function to get the attention score vector $\boldsymbol{v_3} \in \mathbb{R}^b$:

$$\boldsymbol{v_3} = [v_{3_0}, \dots, v_{3_i}, \dots] \quad v_{3_i} = \frac{\exp(v_{2_i})}{\sum_{i=1}^{h} \exp(v_{2_i})}$$

And the final value embedding $\boldsymbol{E}_i \in \mathbb{R}^d$ of gene i was the weighted summation of all embeddings in the look-up table $\boldsymbol{T}$:

$$\boldsymbol{E_x} = \boldsymbol{v_3} \cdot \boldsymbol{T}$$

During pre-training, some expression value in the input gene expression $\boldsymbol{X}$ would be randomly replaced by the masked value m, and the masked value were corresponding to a mask embedding:

$$0 \mapsto \boldsymbol{E}^{\mathrm{m}} \quad \boldsymbol{E}^{\mathrm{m}} \in \mathbb{R}^d$$

After converting scalar values into embeddings, an additional gene embedding was necessary to distinguish and indicate different genes. This design was similar to the positional embedding in the conventional language model[15]. We employed a gene look-up table $\boldsymbol{T}^G \in \mathbb{R}^{19266\times d}$ to retrieve embedding for all genes and two indicators T and S, where d is the same hidden dimension size as value embedding. For each gene i, the final input embedding $\boldsymbol{E}_i^{\mathrm{input}}$ was the element-wise sum of the gene and value embeddings:

$$\boldsymbol{E}_i^{\mathrm{input}} = \boldsymbol{E}_i \oplus \boldsymbol{T}_i^G$$

The input embeddings of all genes formed the final input tensor $\boldsymbol{X}^{\mathrm{input}} = [\boldsymbol{E}_1, \boldsymbol{E}_2, \dots, \boldsymbol{E}_{19266}]^T \in \mathbb{R}^{19266\times d}$.

Encoder

The encoder only processed the embeddings of non-zero and non-masked values (i.e. the expressed genes and two total count numbers) and so that the input length of the encoder was about 10% of the full gene length. Denote $S^E = \{S_0^E, S_1^E, \dots, S_K^E\}$ as the index set of non-zero and non-masked values with K elements, the input of encoder was defined as:

$$\boldsymbol{X}^{\mathrm{Enc-input}} = \left[E_{S_0^E}, E_{S_1^E}, \dots\right]$$

The design of encoder greatly reduced the required computational resources, making it possible for the encoder to employ a series of vanilla transformer blocks to capture gene dependency without any kernel or low-rank approximation. The outputs of encoder were intermediate embeddings $\boldsymbol{X}^{\mathrm{inter}}$:

$$\boldsymbol{X}^{\mathrm{inter}} = \mathrm{Trm}(\boldsymbol{X}^{\mathrm{Enc-input}}) \in \mathbb{R}^{K\times d}$$

where Trm represents the transformer blocks. These intermediate embeddings had two usages: 1) they were then sent into the decoder with the zero and mask embeddings. 2) they could be pooled into cellular embeddings for downstream usages.

Decoder

To establish a transcriptome-wide gene regulation relationship, the zero-expressed genes should also be considered for recovering expression values at mask positions. The intermediate embeddings from encoder were concatenated with the zero and mask embeddings to form a decoder input tensor $\boldsymbol{X}^{\mathrm{Dec-input}}$ with full gene length:

$$\boldsymbol{X}^{\mathrm{Dec-input}} = \left[\boldsymbol{X}^{\mathrm{inter}}, \boldsymbol{E}_0^0, \dots, \boldsymbol{E}_{K_0}^0, \boldsymbol{E}_0^{\mathrm{m}}, \dots, \boldsymbol{E}_{K_m}^m\right]^T \in \mathbb{R}^{19266\times d}$$

where K_0 and K_m were the number of zero and masked embeddings, respectively. We used the kernel-based approximation transformer variant Performer[11] as the transformer blocks in the decoder, since the attention calculation was challenging for long sequences[11,62].

$$\boldsymbol{X}^{\mathrm{Out}} = \mathrm{Perfromer}(\boldsymbol{X}^{\mathrm{Dec-input}}) \in \mathbb{R}^{19266\times f}$$

where $\boldsymbol{X}^{\mathrm{Out}} \in \mathbb{R}^{19266\times d}$ was the output tensor of the decoder with dimension f. For predicting the expression value, the embeddings of T and S were dropped and a MLP was followed to project other f-dimension embedding to scalars. These scalars formed a prediction vector $\boldsymbol{P}$:

$$\boldsymbol{P} = \mathrm{MLP}(\boldsymbol{X}^{\mathrm{Out}}) \in \mathbb{R}^{19264}$$

All parameters $\boldsymbol{\Theta} = \{\boldsymbol{E}^0, \boldsymbol{T}, \boldsymbol{w_1}, \boldsymbol{w_2}, \alpha, \boldsymbol{E}^{\mathrm{m}}, \boldsymbol{T}^G, \boldsymbol{\Theta}_{\mathrm{Encoder}}, \boldsymbol{\Theta}_{\mathrm{Decoder}}, \boldsymbol{\Theta}_{\mathrm{MLP}}\}$ were optimized during the pre-training. The detailed hyper-parameter setting of different models could be found in the Table S1.

Read-depth-aware pre-training task

We trained our model with a read-depth-aware (RDA) gene expression prediction task. This task followed a similar self-supervised learning setting but with augmented input data. For each raw pre-training single-cell gene expression vector, we used a hierarchical Bayesian downsampling strategy to generate its low total count variant or unchanged profiles as the input vector. Then we defined two total count indicator T and S and set their values as the total count of the raw and input vectors, respectively. The raw and input vector were normalized and log-transformed.

Then we randomly masked the genes' expressions of the input vector. In this study, we used 30% as our masking ratio for both zero and non-zero values. Then the partial masked input vector was concatenated with two total count indicator T and S and fed into the model. After getting the model-predicted raw gene expression, we conducted the regression loss on the masked genes between the predicted and the raw value. If the input vector was unchanged, the model learned to capture the relation between genes within a single cell. If the input vector was the low total count variant, the model learned the relationship between cells with different read depths.

Total count indicator

Besides gene expression values, there were two additional indicators S and T in our model input, indicating the value of the input and the output total count, respectively. At the pre-training stage, these two indicators were computed from the input and raw gene expression vectors. During the inference, the value of S was still computed from the sum of the input cell's gene expression, but the value of T was set as the desired total count value (e.g., two folds of the input total count).

Hierarchical Bayesian downsampling strategy

For a single-cell gene expression vector $\boldsymbol{X} \in \mathbb{R}^{19264}$, we introduced a two-hierarchy Bayesian sampling to generate its corresponding input vector $\boldsymbol{X}^{input}$. In the first hierarchy, a Bernoulli-distributed random variable γ was introduced to decide whether $\boldsymbol{X}$ would be downsampled. If γ equaled to 0, the input vector $\boldsymbol{X}^{input}$ was the copy of $\boldsymbol{X}$; If γ equaled to 1, the input vector $\boldsymbol{X}^{input}$ was the downsampled variant of vector $\boldsymbol{X}$ given by the second hierarchy. For those gene expression with total count lower than 1,000, we thought their quality were not good for downsampling and fixed $\gamma = 0$.

In the second hierarchy, we downsampled the raw count value of each gene via a binomial distribution:

$$X_i^{input} \sim B(X_i, b)$$

where X_i^{input} was the downsampled raw count value of gene i, X_i was the gene i observed raw count value and b was the downsampling rate. for each cell, all genes shared the same downsampling rate b. And thus the expectation of the downsampled cell's total count satisfying:

$$E\left[TC\left(\boldsymbol{X}^{input}\right)\right] = E\left[X_0^{input} + \ldots + X_N^{input}\right] = E\left[X_0^{input}\right] + \cdots + E\left[X_N^{input}\right]$$
$$= b \cdot X_0 + \cdots + b \cdot X_N = b \cdot TC(\boldsymbol{X})$$

This design ensured that the expectation of the fold change between the raw and downsampled cell's total count was fixed as $1/b$. But for each gene in the $\boldsymbol{X}^{input}$, the expression value is an observation of the random variable.

Further, we let the parameter b followed a beta distribution:

$$b \sim Beta(2,2)$$

This prior distribution guaranteed that the different fold changes between input and output could be seen during training.

Overall, the input vector $\boldsymbol{X}^{input}$ can be defined as:

$$\boldsymbol{X}^{input} = \begin{cases} \boldsymbol{X} & if\ \gamma = 0 \\ [B(X_0, b), B(X_1, b), \ldots, B(X_N, b)] & if\ \gamma = 1 \end{cases} \qquad \gamma \sim Bernoulli(0.5)\ \ b \sim Beta(2,2)$$

Then this input vector would be random masked and sent into the xTrimoGene model.

Mean square error (MSE) loss

Unlike other works that discretize expression values, our model operated on continuous raw gene expression values. So we used the regression loss as the loss function:

$$L = \frac{1}{|M|} \sum_{i=0}^{|M|} (X_i - P_i)^2$$

where M was a binary vector indicating which gene is masked (i.e. value 1 refers to be masked), X_i and P_i were the ground truth and predicted gene expression values, respectively. And | | was a 1-norm operator.

Implementation

The attention mechanism in language modeling determined that increasing sequence length given near quadratic growth of time and space complexity. Even with the model architecture improvements which reduced the complexity from quadratic to linear, experiments with a large training corpus and large model parameter size still required a significant amount of resources. Therefore, time and resource-efficient training techniques were crucial for this work to collect abundant solid experimental results and supported training on 50M data.

HPC cluster infrastructure with NVIDIA Ampere GPU, NVLink for inter-GPU communication inside a server, and a high-speed interconnection network between servers, were utilized as a multi-node deep learning environment to optimize our experiment process.

Since half-precision operations could be executed on FP16 or BFLOAT16 Tensor Core which had 2 times more arithmetic throughput than TF32 on NVIDIA Ampere GPU, besides mixed-precision training decreased both the required amount of memory and the memory bandwidth consumption while maintaining model accuracy, our experiments were conducted with mixed-precision training strategy to gain time saving within a given amount of computational resource.

Distributed Data Parallelism was another widely used training strategy in our work to handle large corpus on HPC cluster, as our model architecture has greatly cut off the necessity of model parallelism in space and time complexity for long sequence modeling. Single Ampere GPU provided a sufficient amount of memory for one model replica of billions of parameters to perform forward and backward passes and gradient accumulation was used to increase the effective batch size to enhance large model training.

To achieve a larger model size without introducing model parallelism, ZeRO-DP stage two[63] and checkpointing technique[64] were experimentally verified reducing model state memory and residual state memory in our environment settings without expanding training time too much while introducing extra communication and computational costs.

For efficient and stable training of the model, we put the layer normalization inside the transformer block and thus the gradients were well-behaved at initialization[65], reducing the hyperparameter tuning cost for the learning rate warm-up stage.

Read depth enhancement analysis

For the gene expression prediction evaluation, we sampled 10,000 cells with a high total count (higher than 1,000) from 50 million single-cell data as the validation dataset. These 10,000 cells were excluded at the training stage. Then we used a binomial distribution to generate the low total count gene expression vector and fed it into our model. We only evaluate non-zero gene expression values considering that 0 expression values do not change in value after downsampling. In addition to using MSE as the evaluation metric, we also used the mean relative error (MRE), which can reflect the relative error:

$$MRE = \frac{1}{|M|}\sum_{i=0}^{|M|}\frac{(X_i - P_i)^2}{X_i}$$

For the clustering analysis, we got the cell embeddings from scFoundation and scVI encoder. For others, we got the imputed gene expression profiles. All methods were used with the default parameter setting. Then we followed the SCANPY pbmc3k tutorial and got the cell cluster by the function 'sc.tl.leiden'.

For the evaluation of the clustering results, we first used ARI and NMI as indicators to evaluate the degree of consistency between the clustering results obtained by different methods and the actual cell type labels. Considering that the acquisition of cluster labels will also be affected by the choice of the clustering algorithm, we used SIL as another evaluation indicator. Compared with ARI and NMI, SIL measures the aggregation degree of true cell type labels on the cell neighborhood maps given by different methods, and thus is independent of the choice of clustering algorithm, reflecting the intrinsic properties of cell representation.

Downstream methods

We dumped the cell embeddings for DeepCDR and SCAD tasks, and gene embeddings for GEARS task. And we trained the downstream model based on these embeddings. All baseline models were trained with default parameters.

DeepCDR

We used the cell line and drug-paired data pre-processed by DeepCDR. The cell line data contains 697 gene expression profiles, and we aligned these genes with our unified list. The drugs were represented as graphs with consistent feature matrices and adjacent matrices sizes.

In total 223 drugs and 561 cell lines data from 31 cancer types were considered and 89,585 and 4,729 cell line-drug samples were used for training and test, respectively. For each cell line, we set both indicators S and T equal to the sum of all gene expression values. And we fed the non-zero gene expression values and two indicators into the model encoder and got the context embedding for each gene. The cell line embedding was obtained by the max-pooling operation for each embedding dimension across all genes.

We trained the baseline model by setting parameters "-use_gexp" as True and "-use_mut" and "-use_methy"

as False. Then for the embedding-based model, we directly replaced the gene expression with the cell embedding and trained the DeepCDR with the same setting. Pearson's correlation coefficient (PCC) was used as the evaluation matrix. For each gene, we computed the PCC between predicted IC50 and truth IC50 across all cell lines. For each cell line, we computed the PCC across all drugs conducted on this cell line.

SCAD

For training the baseline model, we used the processed data provided in their repository. Gene expression values were transformed into the z-score in the processed data. We used all genes and conducted the weighted sampling in the model training process, following the same experimental setting as the original SCAD study.

For training the embedding-based model, we used the normalized gene expression data. For bulk data, we set both S and T equal to the sum of all gene expression values. The same values of S and T would guide the model to keep the original cell line features. For single-cell data, we set the marker S to the sum of all gene expression values, and uniformly set the marker T to 10,000, which empirically is the maximum sequencing depth of a single cell. This setting made the model output consistent in read depth and was as close to Bulk data's read depth as possible. Then the non-zero values of each bulk-level/single-cell gene expression sample and two indicators were fed into the encoder of the pre-trained model. The outputs were the context embeddings of genes for each sample. We found that the best performance can be achieved by concatenating the embeddings obtained in four ways. 1) the max-pooling operation for each embedding dimension across all genes. 2) the mean-pooling operation for each embedding dimension across all genes. 3) the context embedding corresponding to the indictor S. 4) the context embedding corresponding to the indictor T. These four types of embeddings built the new cell embeddings with 3072 dimensions. We used these cells' embeddings to train a new SCAD model.

GEARS

We downloaded the raw gene expression data and unified the gene list to 19,264. We regenerated the gene co-expression network on each dataset. Then we trained the baseline model by setting epoch to 15 and batch size to 30. For the embedding-based model, we first set each cell's T and S values equal to the its total count. Then the gene expression and these two indicators were fed into the model. We dropped the last MLP layer in our model and got the gene context embeddings from decoder. The cell-specific gene context embeddings were set as the node features of the co-expression graph. Then we trained the GEARS model's parameter and froze others. We used the gradient accumulation technique to guarantee the same batch size as the baseline during training.

We followed the definition and metrics used in GEARS and Norman. We focused on the synergy and suppression gene intersection types since they were the most basic types. Identification of these two types was based on the magnitude score which measured the similarity between the two-gene perturbation and combining two single-gene perturbations. Specifically, let the mean change between post- and pre-A perturbed cells as δg_a. A linear model was used to fit the effect of δg_a, δg_b, and δg_{a+b}:

$$\delta g_{a+b} = \mathrm{c_a} \times \delta g_a + \mathrm{c_b} \times \delta g_b + \epsilon$$

where ϵ captures the error in the model fit. We used the robust regression with a Theil-Sen estimator following the same procedure used in []. Using the values of the coefficients, the magnitude was defined as:

$$Magnitude = \sqrt{c_a^2 + c_b^2}$$

Two gene perturbations were ranked by magnitude score, with the top and bottom-ranked being considered synergistic and repressive types, respectively.

References

1. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med* **50**, 1–14 (2018).

2. Jovic, D. *et al.* Single-cell RNA sequencing technologies and applications: A brief overview. *Clinical and Translational Medicine* **12**, e694 (2022).

3. Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C. & Teichmann, S. A. The Technology and Biology of Single-Cell RNA Sequencing. *Molecular Cell* **58**, 610–620 (2015).

4. Regev, A. *et al.* The Human Cell Atlas. *Elife* **6**, e27041 (2017).

5. Svensson, V., Vento-Tormo, R. & Teichmann, S. A. Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols* **13**, 599–604 (2018).

6. Brown, T. B. *et al.* Language Models are Few-Shot Learners. Preprint at https://doi.org/10.48550/arXiv.2005.14165 (2020).

7. Zhao, W. X. *et al.* A Survey of Large Language Models. Preprint at https://doi.org/10.48550/arXiv.2303.18223 (2023).

8. Zhang, R., Luo, Y., Ma, J., Zhang, M. & Wang, S. scPretrain: multi-task self-supervised learning for cell-type classification. *Bioinformatics* **38**, 1607–1614 (2022).

9. Yang, F. *et al.* scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nat Mach Intell* **4**, 852–866 (2022).

10. Cui, H., Wang, C., Maan, H. & Wang, B. scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI. 2023.04.30.538439 Preprint at https://doi.org/10.1101/2023.04.30.538439 (2023).

11. Choromanski, K. *et al.* Rethinking Attention with Performers. Preprint at https://doi.org/10.48550/arXiv.2009.14794 (2022).

12. Ma, X. *et al.* Luna: Linear Unified Nested Attention. Preprint at http://arxiv.org/abs/2106.01540 (2021).

13. Gong, J. *et al.* xTrimoGene: An Efficient and Scalable Representation Learner for Single-Cell RNA-Seq Data. 2023.03.24.534055 Preprint at https://doi.org/10.1101/2023.03.24.534055 (2023).

14. Chen, J. *et al.* Transformer for one stop interpretable cell type annotation. *Nat Commun* **14**, 223 (2023).

15. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint at https://doi.org/10.48550/arXiv.1810.04805 (2019).

16. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research* **30**, 207–210 (2002).

17. Papatheodorou, I. *et al.* Expression Atlas update: from tissues to single cells. *Nucleic Acids Research* **48**, D77–D83 (2020).

18. Seal, R. L. *et al.* Genenames.org: the HGNC resources in 2023. *Nucleic Acids Research* **51**, D1003–D1009 (2023).

19. Kaplan, J. *et al.* Scaling Laws for Neural Language Models. Preprint at https://doi.org/10.48550/arXiv.2001.08361 (2020).

20. van Dijk, D. *et al.* Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell* **174**, 716-729.e27 (2018).

21. Huang, M. *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nat Methods* **15**,

539–542 (2018).

22. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat Commun* **9**, 997 (2018).

23. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat Methods* **15**, 1053–1058 (2018).

24. Baron, M. *et al.* A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Systems* **3**, 346-360.e4 (2016).

25. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* **8**, 14049 (2017).

26. Unger, F. T., Witte, I. & David, K. A. Prediction of individual response to anticancer therapy: historical and future perspectives. *Cell. Mol. Life Sci.* **72**, 729–757 (2015).

27. Liu, Q., Hu, Z., Jiang, R. & Zhou, M. DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics* **36**, i911–i918 (2020).

28. Barretina, J. *et al.* The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**, 603–607 (2012).

29. Iorio, F. *et al.* A Landscape of Pharmacogenomic Interactions in Cancer. *Cell* **166**, 740–754 (2016).

30. Bellamy, D., Celi, L. & Beam, A. L. Evaluating Progress on Machine Learning for Longitudinal Electronic Healthcare Data. *arXiv:2010.01149 [cs, stat]* (2020).

31. Geeleher, P., Cox, N. J. & Huang, R. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome Biol* **15**, R47 (2014).

32. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* **102**, 15545–15550 (2005).

33. Kanehisa, M. & Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* **28**, 27–30 (2000).

34. Saddoughi, S. A., Song, P. & Ogretmen, B. Roles of Bioactive Sphingolipids in Cancer Biology and Therapeutics. in *Lipids in Health and Disease* (eds. Quinn, P. J. & Wang, X.) 413–440 (Springer Netherlands, 2008). doi:10.1007/978-1-4020-8831-5_16.

35. Kurundkar, D. *et al.* Vorinostat, an HDAC inhibitor attenuates epidermoid squamous cell carcinoma growth by dampening mTOR signaling pathway in a human xenograft murine model. *Toxicology and Applied Pharmacology* **266**, 233–244 (2013).

36. Park, H. *et al.* Phase I dose-escalation study of the mTOR inhibitor sirolimus and the HDAC inhibitor vorinostat in patients with advanced malignancy. *Oncotarget* **7**, 67521–67531 (2016).

37. Zibelman, M. *et al.* Phase I study of the mTOR inhibitor ridaforolimus and the HDAC inhibitor vorinostat in advanced renal cell carcinoma and other solid tumors. *Invest New Drugs* **33**, 1040–1047 (2015).

38. Dagogo-Jack, I. & Shaw, A. T. Tumour heterogeneity and resistance to cancer therapies. *Nat Rev Clin Oncol* **15**, 81–94 (2018).

39. McGranahan, N. & Swanton, C. Clonal Heterogeneity and Tumor Evolution: Past, Present, and the Future. *Cell* **168**, 613–628 (2017).

40. Vasudevan, S. *et al.* Drug-Induced Resistance and Phenotypic Switch in Triple-Negative Breast Cancer Can Be Controlled via Resolution and Targeting of Individualized Signaling Signatures. *Cancers* **13**, 5009 (2021).

41. He, M. X. *et al.* Transcriptional mediators of treatment resistance in lethal prostate cancer. *Nat Med* **27**, 426–433 (2021).

42. Jerby-Arnon, L. *et al.* A Cancer Cell Program Promotes T Cell Exclusion and Resistance to Checkpoint Blockade. *Cell* **175**, 984-997.e24 (2018).

43. Zheng, Z. *et al.* Enabling Single-Cell Drug Response Annotations from Bulk RNA-Seq Using SCAD. *Advanced Science* **n/a**, 2204113.

44. Ho, Y.-J. *et al.* Single-cell RNA-seq analysis identifies markers of resistance to targeted BRAF inhibitors in melanoma cell populations. *Genome Res.* **28**, 1353–1363 (2018).

45. Kinker, G. S. *et al.* Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity. *Nat Genet* **52**, 1208–1218 (2020).

46. Bock, C. *et al.* High-content CRISPR screening. *Nat Rev Methods Primers* **2**, 1–23 (2022).

47. Rood, J. E., Maartens, A., Hupalowska, A., Teichmann, S. A. & Regev, A. Impact of the Human Cell Atlas on medicine. *Nat Med* **28**, 2486–2496 (2022).

48. Adamson, B. *et al.* A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell* **167**, 1867-1882.e21 (2016).

49. Dixit, A. *et al.* Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell* **167**, 1853-1866.e17 (2016).

50. Roohani, Y., Huang, K. & Leskovec, J. *GEARS: Predicting transcriptional outcomes of novel multi-gene perturbations*. http://biorxiv.org/lookup/doi/10.1101/2022.07.12.499735 (2022) doi:10.1101/2022.07.12.499735.

51. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat Methods* **16**, 715–721 (2019).

52. Lotfollahi, M. *et al.* Learning interpretable cellular responses to complex perturbations in high-throughput screens. 2021.04.14.439903 Preprint at https://doi.org/10.1101/2021.04.14.439903 (2021).

53. Norman, T. M. *et al.* Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science* **365**, 786–793 (2019).

54. He, K. *et al.* Masked Autoencoders Are Scalable Vision Learners. Preprint at https://doi.org/10.48550/arXiv.2111.06377 (2021).

55. Ma, S. *et al.* Chromatin Potential Identified by Shared Single-Cell Profiling of RNA and Chromatin. *Cell* **183**, 1103-1116.e20 (2020).

56. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat Biotechnol* **37**, 1452–1457 (2019).

57. Kamimoto, K. *et al.* Dissecting cell identity via network inference and in silico gene perturbation. *Nature* **614**, 742–751 (2023).

58. Chen, S. *et al.* hECA: The cell-centric assembly of a cell atlas. *iScience* **25**, 104318 (2022).

59. Li, M. *et al.* DISCO: a database of Deeply Integrated human Single-Cell Omics data. *Nucleic Acids Research* **50**, D596–D602 (2022).

60. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573-3587.e29 (2021).

61. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* **19**, 15 (2018).

62. Beltagy, I., Peters, M. E. & Cohan, A. Longformer: The Long-Document Transformer. Preprint at https://doi.org/10.48550/arXiv.2004.05150 (2020).

63. Rajbhandari, S., Rasley, J., Ruwase, O. & He, Y. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. Preprint at http://arxiv.org/abs/1910.02054 (2020).

64. Chen, T., Xu, B., Zhang, C. & Guestrin, C. Training Deep Nets with Sublinear Memory Cost. Preprint at https://doi.org/10.48550/arXiv.1604.06174 (2016).

65. Xiong, R. *et al.* On Layer Normalization in the Transformer Architecture. Preprint at https://doi.org/10.48550/arXiv.2002.04745 (2020).

Data availability

All data used in this study are publicly available and the usages are illustrated in the Methods. The pre-training datasets were mainly downloaded from GEO (https://www.ncbi.nlm.nih.gov/geo/), Single Cell Portal (https://singlecell.broadinstitute.org/single_cell), HCA (https://data.humancellatlas.org/), EMBL-EBI (https://www.ebi.ac.uk/). The processed data are available from the corresponding author upon reasonable request. The datasets used for downstream tasks can be downloaded from the following link: Baron dataset(https://github.com/mohuangx/SAVER-paper); Zheng68K dataset (http://support.10xgenomics.com/single-cell/datasets); Cancer drug response dataset (https://github.com/kimmo1019/DeepCDR); Single cell drug response classification dataset(https://github.com/CompBioT/SCAD); Perturbation dataset (https://github.com/snap-stanford/GEARS).

Code availability

The code for producing the baseline results is in the downstream methods' GitHub repository. The code for the scFoundation embedding-based model will be released soon.

Acknowledgements

We thank Qijin Yin, Linlin Chao, Zhaoren He from Biomap, Yixin Chen, Chen Li, Haiyang Bian, Jiaqi Li, Tianxing Ma and Dr. Rui Jiang from Bioinfo Division, Tsinghua University for discussions and comments. This work was partially supported by National Natural Science Foundation of China (NSFC) (grants 62250005, 61721003), National Key R&D Program of China (grant 2021YFF1200901) and Tsinghua-Fuzhou Institute for Data Technology (TFIDT2021005).

Author contributions

M.H., J.M., L.S. and X.Z. conceived the study. M.H. Xin Z. (Xin Zeng) and Y.G. collected the downstream datasets involved in this article. Y.G. and L.S. developed data collection criteria and strategies for pre-training. M.H., J.G., Xin Z., C.L., T.W. and X.C. proposed the pre-training framework. M.H., J.G., Xin Z. and C.L. implemented and pre-trained the models. M.H. and J.G. benchmarked all methods. J.G., X.Z., C.L., T.W. X.C., J.M., L.S. and X.Z. provided a lot of advice on pre-training framework design and downstream tasks. M.H., J.G., J.M., L.S. and X.Z. wrote the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information

Evaluation Metrics & Supplementary Figures & Supplementary Tables