scPaLM: Pre-training of Single-cell Language Models through Genetic Pathway Learning

Anonymous ACL submission

Abstract

As scRNA-seq expands dataset richness and advances cellular biology, transformer-based single-cell foundation models have emerged. Despite their efficacy, they face two key challenges: (1) Biological Perspective: They overlook the unordered nature of gene expression and fail to incorporate pathway priors essential for capturing functional interactions. (2) Computational Perspective: The high dimensionality of gene tokens leads to excessive tokenization and a lack of an efficient mechanism for handling long-context sequences. Driven by these challenges, we propose a novel **S**ingle-**c**ell Pre-trained **L**anguage **M**odel via Genetic **Pa**thway Learning, named **scPaLM**[1], that addresses these challenges through three key innovations: ❶ **Permutation-Invariant Embeddeding** where we handle high numer of genes via patching technique at the same time keeping permuation-invaraince; ❷ a **Genetic Pathway Learning** module that is designed to learn discrete representations, enabling the modeling of collective gene behaviors in a data-driven way; ❸ **Cell-level Information Aggregation** that progressively aggregates cell representations into a designated token during the training phase, with a tailored masking strategy and a token-level contrastive regularizer. Comprehensive evaluations across four biological benchmarks demonstrate **scPaLM**'s superiority: it achieves average $\mathbf{10.1}\%$ improvement in cell type annotation compared to scGPT across all datasets, $\mathbf{5.15}\%$ increase in drug response prediction correlation compared to scFoundation.

1 Introduction

Single-cell RNA sequencing has emerged as the state-of-the-art method for elucidating the intricacies and diversity inherent in RNA transcripts at the individual cell level and providing insights into the composition of distinct cell types and their respective functions within tissues, organs, and organisms (Jovic et al., 2022). The massive amount of data generated by scRNA-seq techniques has provided massive information on various cells, enabling a better understanding of them, and hence benefit diverse research areas such as development (Semrau et al., 2017), auto-immune diseases (Gaublomme et al., 2015) and cancer diagnosis or prognosis (Patel et al., 2014).

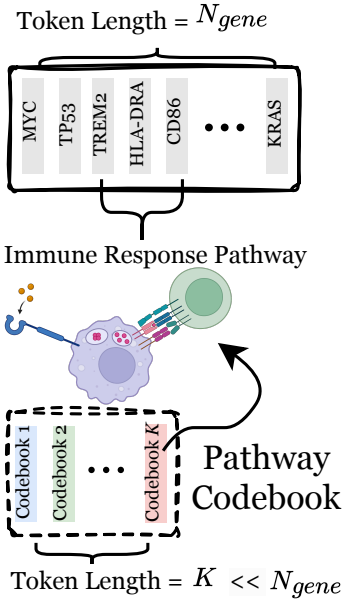

Figure 1: Compared to existing works where N_{gene} often reaches tens of thousands, we leverage the inherent functional similarity among gene groups, such as pathway-level organization (e.g., TREM2, HLA-DRA, and CD86 within the Immune Response Pathway), to enable latent pathway learning with an efficiently reduced token count $K << N_{gene}$ in the latent space.

To effectively model the scRNA-seq data, various computational methods (Cui et al., 2024a; Hao et al., 2024a) with different architecture designs have been proposed. Recent progress in deep learning has inspired the usage of advanced machine learning techniques, such as transformers (Vaswani, 2017), to scRNA-seq data analysis. These models, particularly those inspired by the success of

[1]Source code is provided in supplementary.

BERT (Devlin et al., 2018) GPT (Radford et al., 2019), adopt a token-based approach, treating the expression count of each gene as a "token", a concept widely embraced in natural language processing (NLP). These tokens are then assembled into a "sentence" representing the genetic expression profile of a cell. Such models have demonstrated exceptional performance in capturing the underlying structures inherent to scRNA-seq data and have consistently outperformed conventional algorithms across various downstream tasks, including cell type annotation and bulk drug response prediction (Cui et al., 2024b; Hao et al., 2024b; Theodoris et al., 2023; Yang et al., 2022). Despite its effectiveness, several major bottlenecks remain:

Biological Perspective. Unlike NLP domain, where the order of words carries semantic meaning, gene expression in scRNA-seq does not follow a strict order. Traditional transformer models assume sequential order (like in sentences), but gene expression is inherently unordered—what matters is the presence, absence, and expression levels of genes (Cui et al., 2024b). This naturally raises the challenge of efficiently handling the vast number of genes in single-cell data. Moreover, existing approaches lack explicit mechanisms to incorporate pathway prior knowledge - the well-established biological principle that genes operate in coordinated functional units to execute cellular processes (Cui et al., 2024b; Liang et al., 2023). This oversight limits their ability to model the hierarchical organization of genetic regulation and discover biologically interpretable patterns.

Computational Perspective. Considering that each cell is expressed with vast number of genes, treating each individual gene as a distinct token leads to a substantial increase in the overall token count, resulting in significant computational demands. Traditional approaches to address this issue is to select a subset of highly variable genes, such as the top $2,000$ genes, which can notably reduce the overall gene count (Cui et al., 2024b). However, this gene exclusion inevitably entails the loss of valuable biological information, as certain essential genes, like housekeeping genes, may not exhibit high variability while maintaining pivotal regulatory functions (Joshi et al., 2022).

In this paper, we propose `scPaLM`, a transformer-based model that effectively harnesses massive scRNA-seq data. `scPaLM` incorporates multiple innovative elements: ❶ We propose an **Permutation-Invariant Embedding**, an efficient embedding process that condenses the information from all the genes into a reduced number of tokens by leveraging a symmetric encoder-decoder architecture while ensuring the nature of permutation-invariance, substantially mitigates the computational expenses and facilitates rapid training and inference; ❷ Acknowledging the collective nature of gene functionality, we introduce a **genetic pathway learning**, an encoder which seeks to encode gene tokens into genetically related pathway tokens and acquire discrete representations for them to capture the collective yet distinct functionality of genes; Finally, ❸ to aggregate cell-specific information, we establish a tailored training framework, **Cell-level Information Aggregation** to learn a designated token to represent cells, with the establishment of a masking strategy and a token-level contrastive regularizer. Extensive downstream tasks including cell type annotation and drug response prediction, `scPaLM` achieves greater performance compared to baseline methods, including Geneformer (Theodoris et al., 2023), scGPT (Cui et al., 2023) and scFoundation (Hao et al., 2023).

Our contributions are summarized as follows:

* We propose `scPaLM`, a pretrained foundation model which incorporates biological pathway with efficient tokenziation on massive scRNA-seq data that achieves state-of-the-art performance on various downstream tasks.

* We devise multiple innovative elements that contribute to the success of `scPaLM`: (1) a novel permutation-invariant embedding process that efficiently maps gene expression values into representations for subsequent modeling; (2) a genetic pathway encoder that is designed to model the collective behaviors of genes by learning discrete representations for their tokens; and (3) a training scheme that aggregates cell-specific information into a designated token, and two techniques that augment the aggregation process.

* We demonstrate performance superiority of `scPaLM` on various downstream tasks. For the cell type annotation task, we outperform Geneformer and scGPT by {8.4%,4.9%}/{8.4%,3.0%} and {6.5%,0.9%}/{5.4%,0.2%} in terms of the ARI and NMI scores of two scRNA-seq datasets. For the drug response prediction task, we outperform scFoundation by up to 5.15% in terms of the correlation of IC50 values. We also achieve

state-of-the-art performance on the drug response perturbation prediction task.

2 Related Works

Single-cell Data Analysis. Several methods have been proposed to model transcriptome measurements at the single-cell level that reflect biological diversity (Lopez et al., 2018). MAGIC (Dijk et al., 2017) aims to predict the missing measurements, often referred to as "dropouts", by propagating in a graph constructed based on cell-cell similarity. scImpute (Li and Li, 2018) learns to accurately and robustly identify dropouts and perform imputation on these identified positions. SAVER (Huang et al., 2018) leverages gene-to-gene relationships to recover the expression level of each individual cell. Lopez et al. (Lopez et al., 2018) developed a scalable framework called scVI for probabilistic representation and analysis of gene expression in single cells. More recently, there has been a notable utilization of pretrained transformers in the context of modeling single-cell RNA sequencing (scRNA-seq) data. In the realm of encoder-only transformers, scBERT (Yang et al., 2022) embeds each gene into a token and leverages an efficient transformer to model over $16,000$ genes for each individual cell. Subsequently, scFoundation (Hao et al., 2023) has made advancements in the embedding process introduced by scBERT, which has resulted in enhanced performance. Geneformer (Theodoris et al., 2023) discards the original measurement of transcriptome and constructs input sequences that account for the ranking of measurements across the entirety of the dataset, thereby creating a representation that encapsulates the relative expression levels of all genes within each cell. For decoder-only models, scGPT (Cui et al., 2023) leverages the concept of next token prediction in NLP to iteratively predict the masked genes, creating a novel path for scRNA-seq data modeling. A concurrent work CellPLM (Wen et al., 2023) encodes cell-cell relations by leveraging spatially-resolved transcriptomic data in pre-training. In this work, our objective is to deploy a vector quantization technique to learn discrete genetic pathway representation.

3 Methodology

Overview. A scRNA-seq dataset is usually stored as a matrix $X_{\text{raw}} \in \mathbf{N}^{N_c \times N_g}$, where N_c is the number of cells and N_g is the number of genes. In X, each row represents the expression values of genes in a cell. A transformation (*e.g.*, `log1p`) is usually applied on X_{raw} to obtain X, *i.e.* data with normalized scales (Yeo and Johnson, 2000).

The main components of **`scPaLM`** are transformers (Vaswani, 2017). To facilitate the learning process, we propose a novel embedding process (§3.1), referred to as $\texttt{embed}(\cdot)$, that maps each row of X into N tokens. Note that our embedding process can produce a reduced amount of tokens, which is more memory- and computation-efficient compared to existing works which usually need to construct a large number of tokens (typically equals to N_g). The tokens are then fed into the transformers in **`scPaLM`**, which leverage the attention mechanism to capture the interaction between tokens and learn the biological implications behind the scRNA-seq data. The transformers have multiple layers and process the token representations sequentially with the following formula: $\boldsymbol{h}_i = \text{Layer}_i(\boldsymbol{h}_{i-1})$, where $\boldsymbol{h}_i$ indicates the token representations generated by the i-th transformer layer.

The training process for **`scPaLM`** consists of two distinct stages. *During the initial stage*, we train both an encoder and a decoder using a reconstruction loss. Specifically, the encoder is trained to map the raw gene tokens to tokens that represent genetic pathways (as discussed in §3.2), while the decoder's role is to reverse this mapping, converting genetic pathway tokens back to the original expression levels. *In the second stage*, we train another encoder with an additional token designed to capture cell-specific information (as discussed in §3.3). The two encoders we have trained in these two stages collectively empower various downstream tasks, exhibiting superior performance.

3.1 Permutation-Invariant Embedding

The Tokenization Dilemma in scRNA Modeling Current tokenization strategies for scRNA-seq data face a fundamental trade-off: full-gene tokenization preserves molecular granularity but incurs prohibitive $O(N_g^2)$ computational complexity ($20,000 \lesssim N_g \lesssim 60,000$), while gene filtering discards critical biological signals. Even advanced methods like HVG selection (Cui et al., 2024b) risk eliminating housekeeping genes with low variability but essential functions (Joshi et al., 2022). This dilemma severely limits Transformer-based models' scalability and biological fidelity.

Pathway-Inspired Patch Construction Drawing inspiration from Vision Transformers (Dosovit-

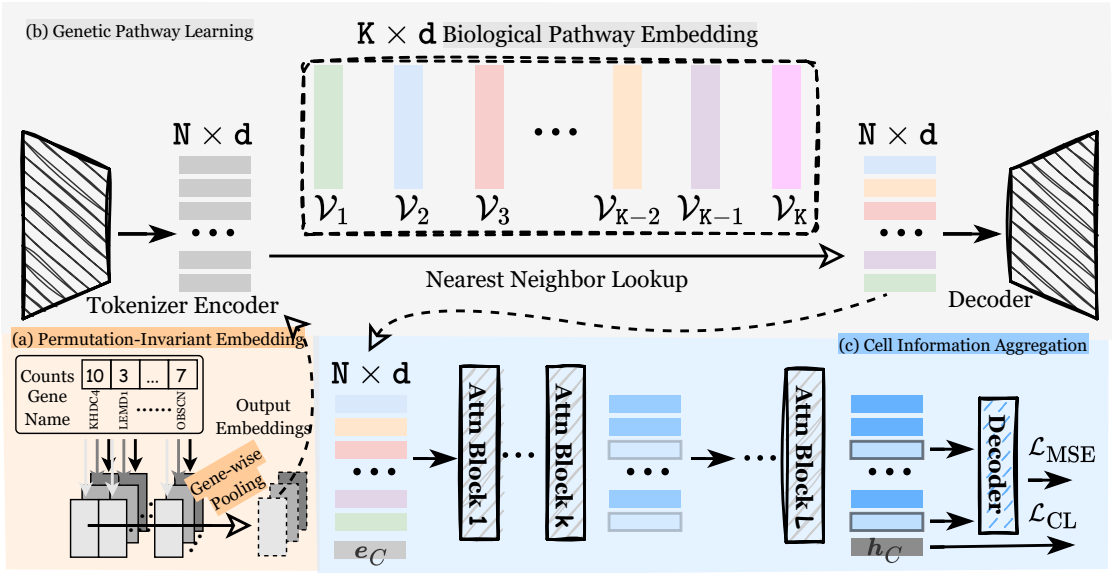

Figure 2: The overview of our framework. Three main innovative components are introduced in our frameworks: an efficient embedding process that is permutation-invariant; a module that captures genetic pathways; and a training framework for cell information aggregation.

skiy et al., 2020), we reconceptualize gene patches as functional units analogous to biological pathways. However, unlike image pixels with spatial coherence, genes lack inherent positional relationships. Direct application of ViT-style patching introduces order sensitivity—permuting gene order alters patch composition, contradicting biological reality where expression vectors are permutation-invariant. To resolve this, we develop a symmetric embedding architecture that dynamically clusters genes into N pathway-aligned patches ($N \ll N_g$) through order-agnostic operations.

Symmetric Embedding Pipeline As detailed in Algorithm 1, our pipeline enforces permutation invariance through three key stages. First, gene expressions are projected into d-dimensional vectors via learnable matrices $\boldsymbol{P}_1 \in \mathbb{R}^{N_g \times d}$, scaled by gene-specific coefficients $\boldsymbol{\alpha}$. A hierarchical pooling strategy then aggregates these embeddings: zero-expression genes are replaced by a learnable embedding and pooled separately, then combined with active genes through concatenation and linear fusion. This two-stage process ensures balanced representation of both silent and active genomic regions. The final embeddings $\bar{\boldsymbol{E}} \in \mathbb{R}^{N \times d}$ are computed through linear interpolation of pooled features with learnable basis vectors $\boldsymbol{P}'$, eliminating positional dependencies. This architecture reduces token count by 99.6% (from $N_g = 60,664$ to $N = 256$) while achieving 16.1% higher clustering accuracy than gene-wise baselines (Table 6), demonstrating superior biological fidelity.

3.2 Genetic Pathway Learning

Biologically, most genes do not function in isolation; instead, they function in concert to perform biological functions (Alexa et al., 2006; Shastry, 2009; Mi et al., 2013). Here, groups of biologically related genes that demonstrate substantial associations with specific biological processes are commonly referred to as *pathways*. Recognizing the activated pathways within a cell holds paramount importance in comprehending its characteristics (Wang and Sherwood, 2011). Despite such significance of pathways current methodologies frequently overlook this aspect.

Pathway Modeling through Discrete Representations Learning. We propose to learn distinct "*pathway*" tokens, represented as *discrete* codes, by training an encoder and a vector quantizer. To be more specifically, the encoder, implemented as a transformer, maps $\bar{\boldsymbol{E}}$ into hidden representations, denoted as $\mathcal{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$. Subsequently, the quantizer learns a codebook $\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K\}$, and associates each $\boldsymbol{e}_i$ with the closest entry in $\mathcal{V}$ in terms of distance. More precisely, for each embedding with index i, we derive the corresponding embedding from the codebook with the following formula: $\boldsymbol{z}_i = \arg\min_{v \in \mathcal{V}} \|\boldsymbol{v} - \boldsymbol{e}_i\|_2$. After acquiring $\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$, we input it into an additional decoder, which is implemented as another transformer model, and obtain the output vector $\boldsymbol{o} \in \mathbb{R}_g^N$. These modules are

trained using reconstruction tasks, where a mean-squared-error (MSE) loss is employed to minimize the dissimilarity between $\boldsymbol{x}$ and $\boldsymbol{o}$. Furthermore, we introduce a commitment loss (Huh et al., 2023) to minimize the distance between each pair of $\boldsymbol{e}_i$ and $\boldsymbol{z}_i$, which has the following form:

$$\mathcal{L}_{\text{cmt}} = \sum_{i=1}^{N} \beta \|\texttt{sg}(\boldsymbol{z}_i) - \boldsymbol{e}_i\| + (1-\beta)\|\boldsymbol{z}_i - \texttt{sg}\ (\boldsymbol{e}_i)\|, \tag{1}$$

where sg indicates the stop-gradient operation. To mitigate the issue of index collapses in VQ techniques, we follow Huh et al. (2023) to regularly replace the unused tokens in codebooks with randomly re-initialized tokens, and leverage an affine parameterization to minimize interval covariate shifts. More details are provided in Appendix C.

3.3 Cell Information Aggregation

While the pathway encoder encodes gene expression values into pathway tokens, it does not provide a cellular-level representation. One possible approach to constructing cell representations involves concatenating the representation of genetic pathway tokens. However, this results in a prohibitively high-dimensional representation, leading to increased computational costs in downstream tasks. Alternatively, using the average representation of pathway tokens, while simpler, yields inferior performance as shown in §5.

Additional tokens to aggregate cell information. To aggregate gene representations effectively and efficiently at the cellular level, we introduce a learnable token, $\boldsymbol{e}_C$, designed to encapsulate cell-specific information. This token is associated with a subset, denoted as $\boldsymbol{z}_{i_1}, \boldsymbol{z}_{i_2}, \ldots, \boldsymbol{z}_{i_{N'}}$, randomly selected with monotonically increasing indices from the set $\mathcal{Z}$, therefore information is transferred to $\boldsymbol{e}_C$. We employ an encoder to convert these tokens into representations, which we denote as $\mathcal{H} = \{\boldsymbol{h}_C, \boldsymbol{h}_{i_1}, \boldsymbol{h}_{i_2}, \ldots, \boldsymbol{h}_{i_{N'}}\}$. Subsequently, we exclude $\boldsymbol{h}_C$, replace the positions of the previously omitted pathway tokens with a common learnable token $\boldsymbol{e}_M$, and proceed to decode this modified embedding using a decoder.

Token Scaling and Reconstruction Despite pathway tokens being fewer than genes, our scaling algorithm (Algorithm 3) effectively aligns their dimensions through linear interpolation. This approach preserves biological fidelity while reducing computational complexity, as evidenced by the superior clustering performance of $\boldsymbol{h}_C$ over gene-averaged baselines in Table 6. The architecture freezes the encoder and quantizer introduced in §3.2, minimizing reconstruction loss between original and decoded expressions, ensuring stable training while maintaining pathway-aware representations.

Contrastive Regularization with Dynamic Pseudo-Labels To enhance cell-specific discriminability, we design a contrastive framework leveraging K-means-derived pseudo-labels. During training, two augmented views ($\mathcal{H}_i$, $\mathcal{H}'_i$) of each cell are generated by masking different pathway token subsets. Embeddings $\boldsymbol{h}_{i,C}$ and $\boldsymbol{h}'_{i,C}$ from the same cell cluster form positive pairs, while cross-cluster cells serve as negatives (limited to K pairs per batch). The contrastive loss

$$\mathcal{L}_{CL} = \sum_{i} -\log \frac{s(\boldsymbol{h}_{i,C}, \boldsymbol{h}'_{i,C})/\tau}{s(\boldsymbol{h}_{i,C}, \boldsymbol{h}'_{i,C}) + \sum_{j \in \text{neg}(i)} s(\boldsymbol{h}_{j,C}, \boldsymbol{h}_{i,C})} \tag{2}$$

maximizes similarity within pseudo-classes through cosine similarity $s(\cdot,\cdot)$, with temperature τ sharpening distinctions (Chen et al., 2020). A fixed-length queue Q stores recent $\boldsymbol{h}_C$ embeddings, enabling periodic K-means updates to adapt pseudo-labels to evolving representations. This implicit modeling of cell-type relationships enforces topological consistency in the latent space, as visualized in Figures 3-4. The detailed pipeline is in Algorithm 2.

4 Experiments

In this section, we detail our experimental setting and demonstrate the superior performance of `scPaLM`. A series of experiments are conducted on various downstream tasks, and we have provided ablation studies in §5 to validate the importance of our proposed techniques.

4.1 Implementation Details

Pretraining Data. `scPaLM` is pretrained on single-cell RNA-seq data covering different types of cells, having in total $43,312,189$ cells and $60,664$ genes collected from CELLxGENE platform (Megill et al., 2021). The statistics and description of the pre-training data are in Appendix B.

Architectures. The overall architecture of `scPaLM` can be split into three parts: (1) embedding layers, where we use mainly MLPs as described in Algorithm 1. The N is set to 256 in our experiment;

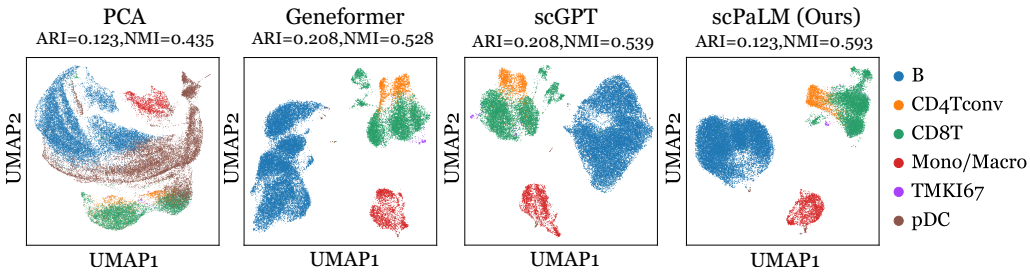

Figure 3: Unsupervised clustering performance on the CLL dataset.

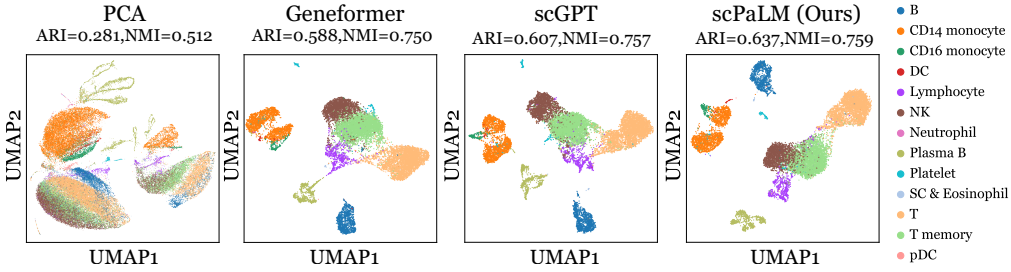

Figure 4: Unsupervised clustering performance on the COVID dataset.

(2) genetic pathway learning, where we introduce an encoder, a decoder, and a quantizer. The encoder and the decoder are developed based on the transformer architecture, which contains 12 layers and a hidden dimension of 768. The quantizer has the same hidden dimension with a codebook size of 128; (3) cell information aggregation, where we introduce another encoder and decoder that have the same architectures and configuration. More details on the architectures can be found in Table 4.

Training Settings. The optimizer we use in our experiments is `AdamW` (Loshchilov and Hutter, 2017). In the first stage of training, where we train the encoder for pathway tokens and the vector quantizer, we adopt a learning rate of 0.001 and a batch size of 128. In the second stage, we train other components with a learning rate of 5×10^{-4} and using the same batch size of 128.

Benchmarks. We compare against baseline methods on various benchmark datasets that are manually excluded from the pretraining data: (1) the CLL (GEO: GSE111014) dataset (Rendeiro et al., 2020), which originally contains 48016 cells with 33694 genes and 6 types of cells. We further filter out cells without type annotations, resulting in 30K cells; (2) the COVID (GEO: GSE150861) dataset (Guo et al., 2020), which contains 11931 cells; (3) the Jurkat from 10x Genomics dataset[2], which contains 3258 cells. We filter out zero-count genes and retain 17753 genes. (4) the PBMC-5k dataset that also comes from 10x Genomics[3] which contains around 5K cells; and finally the Cancer Cell Line Encyclopedia (CCLE) (Barretina et al., 2012) and Genomics of Cancer Drug Sensitivity (GDSC) (Iorio et al., 2016) datasets which are leveraged in the drug response prediction task (§4.3).

Baselines. We compare `scPaLM` with various baseline methods on different tasks. For cell-type annotation, we compare with PCA (where we derive the first 256 principal components on the log-normalized expression values), Geneformer (Theodoris et al., 2023), scGPT (Cui et al., 2023). The latter two are current state-of-the-art algorithms for this task. For imputation, we compare with SAVER (Huang et al., 2018), scImpute (Li and Li, 2018), DCA (Eraslan et al., 2019), which are widely used methods for this task. For drug response prediction, we compare with DeepCDR (Liu et al., 2020) and another transformer-based algorithm, scFoundation (Hao et al., 2023).

4.2 Unsupervised Cell Type Annotation

Our first set of experiments involves applying computational methods on unseen scRNA-seq data and providing type annotations to those unseen cells in an unsupervised manner. We compare the performance of `scPaLM` with three baselines, namely PCA, Geneformer, and scGPT. These experiments are conducted on the CLL and the COVID dataset. Figure 3 and 4 display UMAP visualizations created from the cell representations, *i.e.,* $\boldsymbol{h}_C$. We use the Leiden (Traag et al., 2019) algorithm with a resolution of 1.0 to cluster the embeddings and assess the clustering performance with the adjusted rand index (ARI) and normalized mutual information (NMI) scores. Qualitatively speaking, PCA

[2]Avaiable at https://www.10xgenomics.com/datasets/jurkat-cells-1-standard-1-1-0

[3]Avaiable at https://www.10xgenomics.com/datasets/5k-human-pbmcs-3-v3-1-chromium-controller-3-1-standard

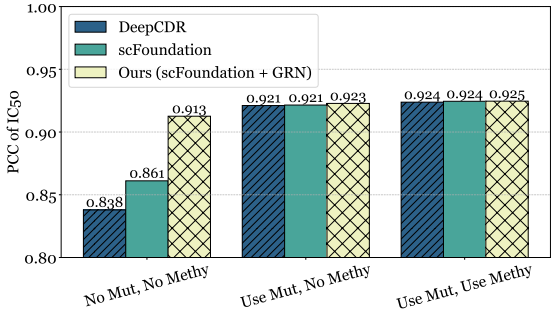

Figure 5: Comparison of Pearson correlation coefficient (PCC ↑) between the predicted and the ground-truth IC50 values using different settings of feature . We compare **scPaLM**'s performance with two baseline algorithms, DeepCDR and scFoundation.

exhibits the poorest results, whereas the UMAPs generated by the other three models demonstrate significantly superior clustering quality. We can also observe that **scPaLM** possesses a smoother and more clustered latent space with respect to the ground-truth cell type labels. Quantitative results also confirm **scPaLM**'s ability to annotate types of cells. Notably, our method achieves higher ARI and NMI scores compared to the baselines by clear margins on both datasets. On CLL, **scPaLM** outperforms the baselines by $0.084 \sim 0.169$ in terms of the ARI score and $0.054 \sim 0.158$ in terms of the NMI score. Similarly, on COVID, it outperforms the baselines by $0.030 \sim 0.356$ in terms of the ARI score and $0.002 \sim 0.247$ in terms of the NMI score. These improvements indicate the high quality of produced embeddings.

4.3 Cancer Drug Response Prediction

Cancer Drug Responses is an important task that can help guide the design of anti-cancer drugs and also understand the cancer biology (Unger et al., 2015). Following the setting in scFoundation (Hao et al., 2023), we combine **scPaLM** with a CDR prediction framework, DeepCDR (Liu et al., 2020), to provide prediction of the IC50 values (*i.e.*, half-maximal inhibitory concentrations) of drugs across different cells. We adopt the settings from scFoundation (Hao et al., 2023) to fuse the extracted representations from gene expression values with the representations of drugs and fit a graph convolution network (GCN) to learn representations that encompass information from multiple sources and modalities. We follow the settings of DeepCDR and experiment with different options: (1) *Use Mut*, which indicates the usage of genomic mutation information; and (2) *Use Methy*, which indicates the usage of DNA methylation data.

From Figure 5, we can observe that both scFoundation and our method outperform the baseline framework DeepCDR significantly and achieve a stronger correlation between the prediction and the IC50 values. Notably, when using no additional information from the mutation and methylation, our method significantly outperforms scFoundation by 5% in terms of the Pearson Correlation Coefficient (PCC). When having additional mutation and methylation information, all methods demonstrate higher PCCs, yet our method remains the top performance among them all.

To have a better understanding of the performance gain, we provide pairwise visualization and case study of the correlation achieved by our method and scFoundation in Figure 6 and 7. Detailed analysis can be found in Appendix E.

4.4 Imputation

Imputation is an important task where the model is asked to recover the expression value of genes within individual cells. It has real-world implications because the measurement of expression levels often exhibits noise (Grün et al., 2014) and suffer from dropout events (Kharchenko et al., 2014). We conduct a series of simulated experiments on the Jurkat and the PBMC dataset to assess **scPaLM**'s ability to accurately predict the expression levels of the missing genes. We randomly sample 10% of genes from each cell with a probability that is proportional to the exponent of negative gene expression values and mask them as 0.

Table 1: Imputation performance of various methods on the Jurkat and the PBMC dataset. We use the rooted mean square error (RMSE) and the mean absolute error (MAE) as the measurements.

Method	Jurkat		PBMC	
	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓
SAVER	0.841	0.664	0.779	0.594
scImpute	1.178	0.838	1.528	1.132
DCA	0.937	0.629	0.833	0.638
scPaLM (Zero Shot)	**0.494**	**0.397**	**0.674**	**0.539**

Table 1 presents the rooted mean square error (RMSE) and the mean absolute error (MAE) between the ground truth and the predicted expression values on the masked genes across different cells. Note that these metrics are calculated based on the log-normalized expression values. Even under a zero-shot setting, **scPaLM** achieves superior performance compared to most baselines, which estimate their parameters on the downstream datasets. This experiment confirms **scPaLM**'s ability in denoising the expression data and capturing the interactions

between cells and genes.

4.5 Genetic Pathway Identification

We finally conducted an experiment to understand the obtained pathway tokens from scRNA-seq datasets. We follow the setting from scGPT (Cui et al., 2023) where we aim to identify genetic pathways on the Immune Human dataset. To associate a gene with a certain pathway token, we first derive the $\mathcal{V}$ for each gene, and for every v, we calculate the associated gene expression vector weighted by the occurrence percentage of v for each cell. Finally, for every v, we obtain the list of associated genes by calculating their relative prevalence. A more detailed algorithm is deferred in §D. We associate 10 genes to each pathway token and run the gene set enrichment analysis (GSEA) algorithm to search for pathways in Reactome Pathway Database (Fabregat et al., 2018). Note that this dataset is not included in our training set; therefore, it constitutes a zero-shot setting. Nevertheless, our method identifies two significant pathways related to the immune system, as shown in Table 7. Particularly, it identifies and clusters the CD1 gene family (CD1E and CD1B), which is involved in antigen presentation that is related to immune reaction.

5 Ablation Studies

Table 2: Clustering performance of different variants for cell representations on the CLL dataset (Rendeiro et al., 2020). We compare the adjusted rand index (ARI), normalized mutual information (NMI), silhouette score (S-score), and clustering time between models.

Method	ARI ↑	NMI ↑	S-score ↑
Mean	0.015	0.059	−0.133
Concatenated	0.181	0.478	0.310
$\boldsymbol{h}_C$ (No CL)	0.275	0.573	0.361
$\boldsymbol{h}_C$ (Ours)	**0.292**	**0.593**	**0.376**

The effectiveness of the cell information aggregation process. We conduct a series of experiments on the CLL dataset to compare two alternatives for building cell representations, where we use the average and the concatenated representations of pathway tokens to represent cells. Table 5 presents the performance on the cell type annotation task, where we can observe that using our cell information aggregation technique yields the best performance among all the variants. We have also conducted an experiment where we do not use the token-level contrastive learning framework to train the embedding of $\boldsymbol{h}_C$. The decreased scores of these experiments demonstrate the importance of the token-level contrastive learning regularizer.

The effectiveness of the pathway encoder. We conduct experiments excluding the genetic pathway learning module discussed in §3.2. Instead of our proposed method, we train the embedding layers and also aggregate information directly from the embeddings of genes on a small subset of training data that have around 100K cells. Table 3 demonstrates the importance of the encoder and the quantizer. We see that the introduced genetic pathway encoder helps improve the clustering performance, improving the metrics by 0.14 and 0.16, respectively. The usage of the quantizer also further improves the performance by an additional 1%.

Table 3: Comparisons on COVID dataset (Guo et al., 2020) with different configurations. The models are trained on a small subset of the pre-training data.

Configuration		ARI ↑	NMI ↑
Encoder	Quantizer		
✗	✗	0.050	0.120
✓	✗	0.191	0.286
✓	✓	**0.201**	**0.291**

The effectiveness of the embedding process. To evaluate the effectiveness of our embedding process, we explore several alternatives and compare them to our proposed embedding process: (1) *per-gene*, where we directly employs gene-specific embeddings $\boldsymbol{E}$. This is a widely adopted option in various methods such as scFoundation (Hao et al., 2023) and Geneformer (Theodoris et al., 2023); (2) *shared-first-layer*, where we employ only a shared $\boldsymbol{P}$ for all the genes. The results are presented in Table 6, where we can observe that these alternatives demonstrate either degraded performance, or suffer from overly high computational cost. The *per-gene* variant results in out-of-memory (OOM) error even using a batch size of 1. Using a shared first layer requires less amount of GPU memory but yields inferior performance compares to our method.

6 Conclusion

This work presents `scPaLM`, a foundation model pre-trained on single-cell RNA-seq data. We devise several novel techniques that efficiently represent gene expression values into tokens, model the collective function of genes, and effectively aggregate cell-specific information into a single token. We evaluate `scPaLM` on a wide range of downstream tasks, and demonstrate it reaches SoTA.

Limitations

Implicit Pathway Modeling. While our genetic pathway learning module captures collective gene behaviors through discrete representations, it currently relies on data-driven discovery rather than explicit integration of established pathway databases (e.g., Reactome or KEGG). Future work could enhance biological interpretability by incorporating curated pathway knowledge through hybrid architectures that combine learned codebooks with prior biological constraints.

Human-Centric Data Bias. Our pre-training dataset primarily focuses on human single-cell transcriptomes from Tabula Sapiens. While this enables strong performance on human biological tasks, the cross-species generalizability of our pathway representations remains uncertain. The evolutionary divergence of gene regulatory networks across species may require specialized adaptation mechanisms when applying `scPaLM` to model non-human organisms.

Ethics Statement

This work adheres to ethical research practices in computational biology. All datasets used for pre-training and evaluation are publicly available through GEO/SRA archives or 10x Genomics, with proper ethical approvals obtained in their original studies. Our framework processes only de-identified genomic data, containing no protected health information. While foundation models like `scPaLM` could theoretically accelerate therapeutic development, we emphasize that any clinical application requires rigorous validation through established biomedical research protocols.

References

Adrian Alexa, Jörg Rahnenführer, and Thomas Lengauer. 2006. Improved scoring of functional groups from gene expression data by decorrelating go graph structure. *Bioinformatics*, 22(13):1600–1607.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, et al. 2012. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Marco Colonna. 2023. The biology of trem receptors. *Nature Reviews Immunology*, 23(9):580–594.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. 2024a. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, pages 1–11.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. 2024b. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, pages 1–11.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. 2023. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, pages 2023–04.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

David van Dijk, Juozas Nainys, Roshan Sharma, Pooja Kaithail, Ambrose J Carr, Kevin R Moon, Linas Mazutis, Guy Wolf, Smita Krishnaswamy, and Dana Pe'er. 2017. Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. *BioRxiv*, page 111591.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. 2019. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, et al. 2018. The reactome pathway knowledgebase. *Nucleic acids research*, 46(D1):D649–D655.

Jellert T Gaublomme, Nir Yosef, Youjin Lee, Rona S Gertner, Li V Yang, Chuan Wu, Pier Paolo Pandolfi, Tak Mak, Rahul Satija, Alex K Shalek, et al. 2015. Single-cell genomics unveils critical regulators of th17 cell pathogenicity. *Cell*, 163(6):1400–1412.

Robert Gray. 1984. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29.

Dominic Grün, Lennart Kester, and Alexander Van Oudenaarden. 2014. Validation of noise models for single-cell transcriptomics. *Nature methods*, 11(6):637–640.

Chuang Guo, Bin Li, Huan Ma, Xiaofang Wang, Pengfei Cai, Qiaoni Yu, Lin Zhu, Liying Jin, Chen Jiang, Jingwen Fang, et al. 2020. Single-cell analysis of two severe covid-19 patients reveals a monocyte-associated and tocilizumab-responding cytokine storm. *Nature communications*, 11(1):3924.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Le Song, and Xuegong Zhang. 2023. Large scale foundation model on single-cell transcriptomics. *bioRxiv*, pages 2023–05.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. 2024a. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, pages 1–11.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. 2024b. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, pages 1–11.

Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. 2018. Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539–542.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. 2023. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. *arXiv preprint arXiv:2305.08842*.

Francesco Iorio, Theo A Knijnenburg, Daniel J Vis, Graham R Bignell, Michael P Menden, Michael Schubert, Nanne Aben, Emanuel Gonçalves, Syd Barthorpe, Howard Lightfoot, et al. 2016. A landscape of pharmacogenomic interactions in cancer. *Cell*, 166(3):740–754.

Chintan J Joshi, Wenfan Ke, Anna Drangowska-Way, Eyleen J O'Rourke, and Nathan E Lewis. 2022. What are housekeeping genes? *PLoS computational biology*, 18(7):e1010295.

Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo. 2022. Single-cell rna sequencing technologies and applications: A brief overview. *Clinical and Translational Medicine*, 12(3):e694.

Peter V Kharchenko, Lev Silberstein, and David T Scadden. 2014. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740–742.

Wei Vivian Li and Jingyi Jessica Li. 2018. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):997.

Qingnan Liang, Yuefan Huang, Shan He, and Ken Chen. 2023. Pathway centric analysis for single-cell rna-seq and spatial transcriptomics data with gsdensity. *Nature communications*, 14(1):8416.

Qiao Liu, Zhiqiang Hu, Rui Jiang, and Mu Zhou. 2020. Deepcdr: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics*, 36(Supplement_2):i911–i918.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. 2018. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Colin Megill, Bruce Martin, Charlotte Weaver, Sidney Bell, Lia Prins, Seve Badajoz, Brian McCandless, Angela Oliveira Pisco, Marcus Kinsella, Fiona Griffin, et al. 2021. Cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices. *bioRxiv*, pages 2021–04.

Huaiyu Mi, Anushya Muruganujan, John T Casagrande, and Paul D Thomas. 2013. Large-scale gene function analysis with the panther classification system. *Nature protocols*, 8(8):1551–1566.

Anoop P Patel, Itay Tirosh, John J Trombetta, Alex K Shalek, Shawn M Gillespie, Hiroaki Wakimoto, Daniel P Cahill, Brian V Nahed, William T Curry, Robert L Martuza, et al. 2014. Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

André F Rendeiro, Thomas Krausgruber, Nikolaus Fortelny, Fangwen Zhao, Thomas Penz, Matthias Farlik, Linda C Schuster, Amelie Nemc, Szabolcs Tasnády, Marienn Réti, et al. 2020. Chromatin mapping and single-cell immune profiling define the temporal dynamics of ibrutinib response in cll. *Nature Communications*, 11(1):577.

Alex Rogozhnikov. 2022. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*.

Stefan Semrau, Johanna E Goldmann, Magali Soumillon, Tarjei S Mikkelsen, Rudolf Jaenisch, and Alexander Van Oudenaarden. 2017. Dynamics of lineage commitment revealed by single-cell transcriptomics of differentiating embryonic stem cells. *Nature communications*, 8(1):1096.

Barkur S Shastry. 2009. Snps: impact on gene function and phenotype. *Single nucleotide polymorphisms: methods and protocols*, pages 3–22.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. 2023. Transfer learning enables predictions in network biology. *Nature*, pages 1–9.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233.

Florian T Unger, Irene Witte, and Kerstin A David. 2015. Prediction of individual response to anticancer therapy: historical and future perspectives. *Cellular and molecular life sciences*, 72:729–757.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Zheng Wang and David R Sherwood. 2011. Dissection of genetic pathways in c. elegans. *Methods in cell biology*, 106:113–157.

Hongzhi Wen, Wenzhuo Tang, Xinnan Dai, Jiayuan Ding, Wei Jin, Yuying Xie, and Jiliang Tang. 2023. Cellplm: Pre-training of cell language model beyond single cells. *bioRxiv*, pages 2023–10.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. 2022. 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.

In-Kwon Yeo and Richard A Johnson. 2000. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959.

A Related Work

Vector Quantization (VQ). In terms of tokenizaiton, VQ (Gray, 1984) is a classical quantization technique in various fields. VQ operates by utilizing a *codebook*, which consists of multiple representations referred to as codes, and associating an input vector with the code within the codebook that is closest in proximity. This approach can be viewed as a form of discrete representation learning since typically only a single token is activated. Researchers have demonstrated that the usage of discrete representation in computer vision can improve the robustness of models. VQ techniques have found application in diverse fields, such as image generation (Van Den Oord et al., 2017; Razavi et al., 2019; Esser et al., 2021), video generation (Yan et al., 2021), and speech recognition (Baevski et al., 2019). In this work, we take a pioneering step by applying the concept of VQ to the field of biosciences, to learn discrete representations capturing genetic pathways.

B Dataset Description

Our pretraining data comes from CELLxGENE (Megill et al., 2021), which has $43,312,189$ cells from more than 8 tissues.

C More Details on Methodology

Algorithm 1 Permutation-Invariant Embedding

Input: A batch of normalized expression value vectors $\boldsymbol{X} \in \mathbb{R}^{B \times N_g}$ where B is the batch size, learnable gene embeddings E_{gene}, learnable projection matrix $\boldsymbol{P}_1 \in \mathbb{R}^{N_g \times d}$, $\boldsymbol{P}_2 \in \mathbb{R}^{d \times d}$, $\boldsymbol{P}' \in \mathbb{R}^{d \times h}$.
Output: Embedding $\bar{\boldsymbol{E}}$ of $\boldsymbol{X}$, $f(\cdot)$ is a symmetric pooling function.
$\boldsymbol{X} \leftarrow \texttt{einsum}(\texttt{"bi,ij->bij"}, \boldsymbol{X}, \boldsymbol{P}_1)$
$\boldsymbol{X} \leftarrow \texttt{leaky\_relu}(\boldsymbol{X})$
$\boldsymbol{E}_{\text{value}} \leftarrow \texttt{einsum}(\texttt{"ij,bij->bij"}, \boldsymbol{\alpha}, \boldsymbol{X}) + \texttt{MLP}(\boldsymbol{X})$
Fill positions of zeros in $\boldsymbol{E}$ with a learnable embedding $\boldsymbol{e}_{\text{zero}}$
$\boldsymbol{E} \leftarrow \texttt{concat}(\boldsymbol{E}_{\text{value}}, \boldsymbol{E}_{\text{gene}})$
$\bar{\boldsymbol{E}} \leftarrow \texttt{einsum}(\texttt{"bj,jl->bjl"}, f(\boldsymbol{E}), \boldsymbol{P}')$
`return` $\bar{\boldsymbol{E}}$

Permutation-Invariant Embedding. Algorithm 1 transforms gene expression vectors into permutation-invariant embeddings. It projects input values into latent features via learnable matrices, combines them with gene embeddings, applies symmetric pooling to aggregate features, and produces final embeddings through linear transformation. This approach reduces computational costs while maintaining invariance to gene order. The `einsum` operation denotes the Einstein summation convention, performing element-wise operations and summation along specified axes indicated by the letters (Rogozhnikov, 2022).

Genetic Pathway Learning. Algorithm 2 outlines the training pipeline for cell representation learning. It first encodes gene expressions into pathway tokens $\mathcal{Z}$ and aggregates cell-level information via a learnable token $\boldsymbol{e}_C$ appended to masked representation. When contrastive learning is enabled, K-Means periodically updates pseudo-labels using stored embeddings in queue Q, and Equation 2 optimizes cluster consistency. The model trains $\boldsymbol{e}_C$, $\boldsymbol{e}_M$, and networks using reconstruction loss for masked tokens while updating $\boldsymbol{h}_C$ in Q.

Algorithm 2 Training Pipeline (One Step)

Input: A batch of gene expression vectors $\boldsymbol{X} \in \mathbb{N}^{B \times N_g}$, current time step T, an interval for re-fit T_r, a K-Means classifier K, a queue Q.
Obtain $\bar{\boldsymbol{E}} \in \mathbb{R}^{B \times N \times d}$ according to §3.1.
Obtain $\mathcal{Z} \in \mathbb{R}^{B \times N \times d}$ according to §3.2.
Randomly select $p\%$ tokens from $\mathcal{Z}$ for each sample and prepend a token $\boldsymbol{e}_C$. Obtain $\mathcal{H}$ and assign clusters.
if use token-level contrastive learning **then**
  if $T \% T_r == 0$ **then**
    Run K-means based on embeddings in Q.
  end if
  Randomly select $p\%$ tokens from $\mathcal{Z}$ for each sample and prepend $\boldsymbol{e}_C$. Obtain $\mathcal{H}'$.
  Calculate the contrastive learning loss according to Equation 2.
end if
Fill $\boldsymbol{e}_M$ into $\mathcal{H}$ at positions previously excluded during sampling.
Train $\boldsymbol{e}_C$, $\boldsymbol{e}_M$, and the two networks with reconstruction loss for excluded tokens (*i.e.*, $\boldsymbol{e}_M$).
Store $\boldsymbol{h}_C$ in Q.

Mask Construction and Output Reshaping. In Algorithm 3, we introduce a simple way to make the shape of the output from the decoder introduced in §3.3 consistent with the original gene expression

vector. Essentially, we flatten the hidden representations, and we assign a region according to the indices of leave-out tokens in which we calculate the MSE loss.

VQ-Techniques For Stable Training. To address the potential index collapse when applying VQ techniques in training neural networks, we follow the pipeline introduced in Huh et al. (Huh et al., 2023). Firstly, they introduce an affine transformation to reparameterize the representation in the codebook with the following formula:

$$\boldsymbol{v}_i = \boldsymbol{c}_{\text{mean}} + \boldsymbol{c}_{\text{std}} \times \boldsymbol{c}_i$$

where the $\boldsymbol{c}_i$ represents the original code vector, and $\boldsymbol{c}_{\text{mean}}$ and $\boldsymbol{c}_{\text{std}}$ indicate the shared affine parameters. Moreover, they introduce several minor modifications to the codebook update process to enhance the stability.

D More Details on Experiments

Model Configurations. Table 4 presents the hyperparameters of our `scPaLM`.

Table 4: Configurations of our `scPaLM`.

Hyperparameters	Value
Hidden Size	768
Intermediate Size	3072
Number of Layers	12
Number of Attention Heads	8
Dropout Probability	0.0
Attention Dropout Probability	0.0

Algorithm 3 Reshaping Masks and Outputs For Loss Calculation.

1: **Input**: a gene expression vector $\boldsymbol{x} \in \mathbb{R}^{N_g}$, the hold-out indices $I = \{i_1, \ldots, i_m\}$, number of tokens N.
2: Calculate the scaling factor $s \leftarrow \lceil N_g/N \rceil$.
3: Initialize a mask vector $\boldsymbol{m} \leftarrow \mathbf{0}^{N_g}$.
4: **for** $j = 1, 2, \ldots, m$ **do**
5: $\quad \boldsymbol{m}_{s\times i_j : s\times(i_j+1)} \leftarrow \mathbf{1}^s$.
6: **end for**
7: Flatten the output from the decoder which also has the shape of $N \times s$ to $1 \times (N \times s)$, and store it as $\boldsymbol{o}$.
8: Crop both $\boldsymbol{o}$ and $\boldsymbol{m}$ to have the length of N_g. Calculate the MSE loss as $\mathcal{L}_{\text{MSE}} = \|\boldsymbol{o} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{m}\|_1$.

Association of Genes with Pathway Tokens. In this section, we describe the methodology employed for associating genes with specific pathway identifiers through an algorithmic approach. The process involves the utilization of a matrix with a dimension of K by N_g, where K represents the number of tokens and N_g the number of genes. For each vector of gene expression, we obtain the set of tokens that are activated within the codebook. Upon activation of the K_i-th token, the corresponding raw gene expression vector is scaled by the frequency of K_i token occurrences among the activated tokens and subsequently aggregated to the K_i-th row of the matrix. This procedure is iterated across the entire gene dataset. Subsequent to the completion of this iterative process, we perform a normalization step on each column, which correlates to individual genes. Following normalization, for each row, we identify and select the genes that exhibit the most significant values (Colonna, 2023).

Table 5: Clustering performance of different variants for cell representations on the CLL dataset. We compare the adjusted rand index (ARI), normalized mutual information (NMI), silhouette score (S-score), and clustering time between models.

Method	ARI ↑	NMI ↑	S-score ↑
Mean	0.015	0.059	−0.133
Concatenated	0.181	0.478	0.310
$\boldsymbol{h}_C$ (No CL)	0.275	0.573	0.361
$\boldsymbol{h}_C$ (Ours)	**0.292**	**0.593**	**0.376**

E More Experimental Results

Additional Drug Response Prediction Result Analysises. In these experiments, we follow the setting of scFoundation and disable the mutation and the methylation features, to focus on the benefit brought by the incorporation of embeddings from gene expression values. From Figure 6 and 7, we can observe that `scPaLM` achieves better PCCs on all but one cancer type and improves the metrics on a majority of cell lines. Following the analysis, we further visualize the best prediction case of the cancer type, namely the low-grade gliomas (LGG) in Figure 7, where we observe both methods achieve high PCC values despite that the IC50 values have a large range from -6 to 6. `scPaLM` outperforms scFoundation by 2% and 4% in terms of the PCC and the Spearman correlation coefficient. These results showcase the effectiveness of `scPaLM`. It

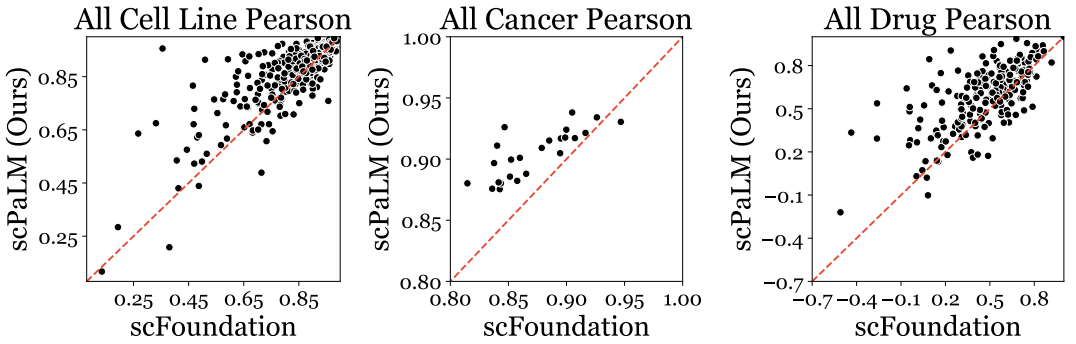

Figure 6: Pairwise visualization of the Pearson correlation coefficient of scFoundation and **scPaLM** based on different grouping strategies. Left: grouping with respect to the cell lines; Middle: grouping with respect to the cancer type; Right: grouping with respect to the drug type. The red lines indicate the relationship of $y = x$.

is also noteworthy that the embeddings generated by **scPaLM** are smaller in dimension compared to those of scFoundation, which implies that **scPaLM** is more efficient in modeling scRNA-seq data.

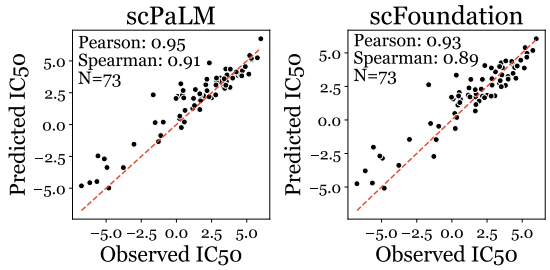

Figure 7: Scatter plots of the predicted and the observed IC50 values on the samples with the cancer type of low-grade gliomas.

Genetic Pathway Results. We present the identified significant genetic pathway in Table 7. Two pathways with p-values that are significantly smaller than 1×10^{-5} are identified, which are all related to immunological reactions.

Table 6: Comparison between different embedding processes. The models are trained on a subset of the pre-training data and evaluated on the COVID dataset.

Embedding Algorithm	Memory Usage ↓	ARI ↑	NMI ↑
Per-gene	>80G	-	-
Shared-first-layer	42378MiB	0.167	0.251
Permutation-invariant (Ours)	43678MiB	0.201	0.291

Comparison of Different Embedding Processes. We conduct an experiment to compare the memory usage and the subsequent clustering performance of models with different embedding processes on the COVID dataset. The results are presented in Table 6.

F Potential Risks

While **scPaLM** demonstrates promising capabilities for therapeutic discovery and cellular analysis, we acknowledge the dual-use potential inherent in any foundational biomedical AI technology. The model's ability to predict cell-drug interactions could theoretically be misapplied to screen compounds with harmful biological activity. To mitigate this risk, we emphasize that any clinical translation of our method must occur within established regulatory frameworks requiring rigorous safety evaluation and ethical oversight. Researchers utilizing this technology should adhere to institutional bio-safety review processes and existing chemical/biological weapons conventions to prevent misuse.

Table 7: Significant pathways identified by GSEA (p-value $< 1 \times 10^{-5}$). Two pathways related to the immune system are selected.

Gene Lists (Token ID)	Term	P-value
CD1E, TREM2, ICAM5, CD1B (25)	Immunoregulatory Interactions Between A Lymphoid And A non-Lymphoid Cell	2.8×10^{-7}
BTN1A1, MRC1, CD1E, TREM2, ICAM5, CD1B (25)	Adaptive Immune System	4.4×10^{-7}