scPerturb: single cell perturbation

Sicheng Yi

05/04/2025

Abstract

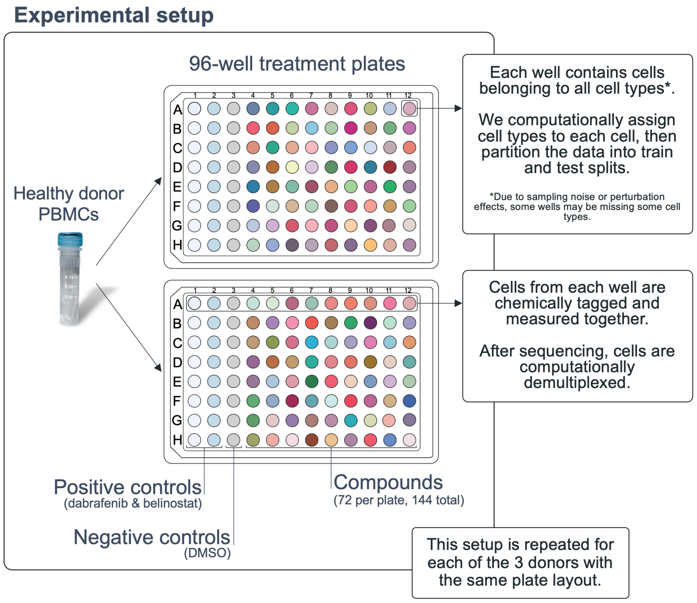
This project investigates how drug perturbations affect gene expression in single cells, with the goal of identifying potential drug candidates for treating leukemia. By modeling the up and down regulation of genes across various peripheral blood mononuclear cell (PBMC) types, including T cells, B cells, NK cells, and myeloid cells, I aim to predict transcriptomic responses to drug compounds. I employ deep learning models, such as multilayer perceptrons (MLPs) and Transformer , to learn from high-dimensional gene expression and molecular features. The results show that these models can more accurately forecast cell-specific differential expression patterns, enabling informed drug screening and prioritization.

Introduction

Each year, millions of people die from cancer, and acute leukemia remains one of the most prevalent and deadly forms. Understanding how chemical drug compounds influence gene regulation in peripheral blood mononuclear cells (PBMCs) is crucial to developing effective treatments. The experimental setup used to study this biological process is illustrated in Figure 1.

In this experiment, blood samples from both healthy individuals and cancer patients are analyzed using single-cell RNA sequencing (scRNA-seq) and chromatin accessibility profiling (scATAC-seq). Over 140,000 individual cells are perturbed with 144 distinct chemical compounds, and their gene expression responses are recorded for more than 18,000 genes. These cells are further categorized into immune subtypes, including CD4+ T cells, CD8+ T cells, natural killer (NK) cells, B cells, and myeloid cells.

As part of the 2023 NeurIPS Kaggle competition, participants were asked to develop models capable of predicting differential gene expression responses based on cell types and small molecule compound features. This presents a challenging high-dimensional regression task due to two main factors: (1) the input features, such as cell type and drug compound, are significantly lower in dimension than the output space of gene expression, and (2) the



Each well contains cells belonging to all cell types*.

We computationally assign cell types to each cell, then partition the data into train and test splits.

*Due to sampling noise or perturbation effects, some wells may be missing some cell types.

Cells from each well are chemically tagged and measured together.

After sequencing, cells are computationally demultiplexed.

This setup is repeated for each of the 3 donors with the same plate layout.

Figure 1: Visualization of PBMC data.

training and test distributions differ, with training samples primarily from T and NK cells, while test samples consist largely of B and myeloid cells.

This project applies and adapts deep learning methods to this biologically grounded prediction task. By augmenting chemical and cellular input features using domain knowledge (e.g., SMILES-based fingerprinting, categorical encoding), and applying multi-target regression models such as multilayer perceptrons (MLPs) and Transformers, I aim to improve generalization to unseen cell types. Additionally, dimensionality reduction techniques like Principal Component Analysis (PCA) are explored to mitigate the computational challenges posed by extremely high-dimensional gene expression outputs. In addition, I investigate the effectiveness of different architectures and pre-processing strategies on a benchmark single-cell genomics dataset.

Related Work

Recent developments in cheminformatics and single-cell perturbation modeling have enabled the application of deep learning to biological response prediction. My work builds on these foundations by combining molecular representations and single-cell transcriptomic.

Randomized SMILES [1]:

They introduced the idea of randomized SMILES strings, demonstrating that varying atom order in molecular SMILES representations can improve generative model performance. This principle motivates our use of molecular fingerprints and SMILES-derived features to represent chemical perturbations more robustly.

scPerturb [2]:

They curated the scPerturb database, which harmonizes 44 single-cell perturbation datasets spanning transcriptomic and proteomics. They propose energy statistics (E-statistics) to quantify perturbation effects across heterogeneous conditions. Their work underscores the need for standardized and scalable evaluation frameworks in perturbation-response modeling.

PBMC scRNA-seq [3]:

They generated a comprehensive single-cell RNA-seq dataset on over 1.3 million PBMCs exposed to various pathogens. Their findings highlight immune cell diversity and context-specific gene regulation, reinforcing the biological motivation for our focus on T, B, NK, and myeloid cells under drug perturbations.

scGen [4]:

They proposed scGen, a deep generative model that uses variational autoencoders to predict cellular responses to perturbations. Their model provides a strong baseline for learning perturbation effects from scRNA-seq data, demonstrating the utility of latent space arithmetic for generalization to unseen conditions.

GCN for Drug Effects [5]:

They applied graph convolution networks (GCN) and transfer learning to predict drug-induced gene expression profiles. This approach illustrates the value of encoding chemical structure explicitly through molecular graphs, a direction we may explore in future work.

Datasets

This project uses data from the scPerturb NeurIPS 2023 Kaggle Competition, which investigates how drug compounds affect gene expression in peripheral blood mononuclear cells (PBMCs) from both healthy and cancer patients. The dataset includes over 140,000 individual cells categorized into major immune cell types: CD4+ T cells, CD8+ T cells, NK cells, B cells, and Myeloid cells. These cells were exposed to 144 different small-molecule drugs, and differential gene expression was measured across more than 18,000 genes.

The main training dataset, provided as a Parquet file (‘de-train.parquet‘), contains rows representing unique combinations of cell type and drug compound. The majority of columns correspond to gene names, with each entry denoting the differential expression level induced by the drug. A positive value indicates gene up-regulation, while a negative value indicates down-regulation. The absolute magnitude reflects the strength of the perturbation effect.

An example of the SMILES string is shown in Table 1, where all entries for the drug clotrimazole share the same molecular representation. A visual rendering of clotrimazole's chemical structure is provided in Figure 5, and its substructure-based encoding is discussed further in the Methods section.

cell_type	**sm_name**	**sm_lincs_id**	**SMILES**	**control**	**A1BG**
NK cells	Clotrimazole	LSM-5341	Clc1ccccc1C(c1ccccc1) (c1ccccc1)n1ccnc1	False	0.104720
T cells CD4+	Clotrimazole	LSM-5341	Clc1ccccc1C(c1ccccc1) (c1ccccc1)n1ccnc1	False	0.915953
T cells CD8+	Clotrimazole	LSM-5341	Clc1ccccc1C(c1ccccc1) (c1ccccc1)n1ccnc1	False	-0.387721
T regulatory cells	Clotrimazole	LSM-5341	Clc1ccccc1C(c1ccccc1) (c1ccccc1)n1ccnc1	False	0.232893

Table 1: differential expression level of each genes perturbed by drug among cell types

The distribution of cell types between training and test datasets is notably imbalanced. As shown in Figure 2, training data predominantly features CD4+/CD8+ T cells and NK cells, while the unseen test data focuses more on B cells and Myeloid cells. This cell-type domain shift presents a significant generalization challenge for predictive models.

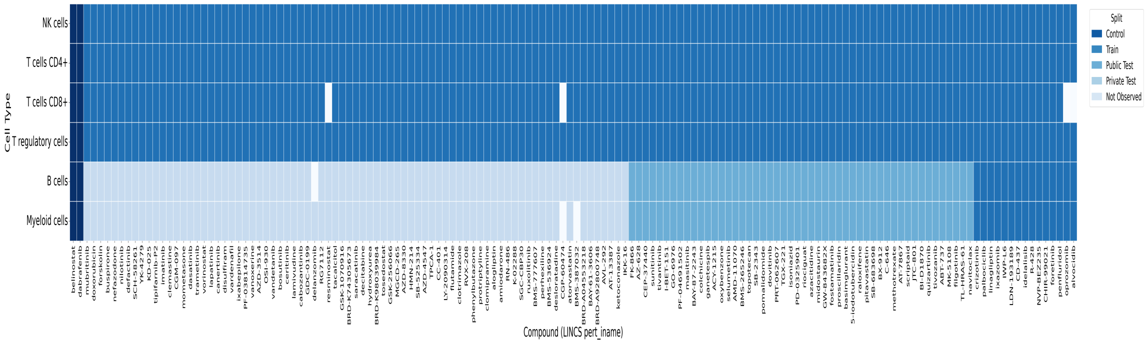


Figure 2: Split of Training Test data

Methods

0.1 EDA

As an initial step, I conducted exploratory data analysis to investigate which genes exhibit strong expression differences between healthy and cancerous conditions. One such gene is **CD69**, which is known to play a role in immune cell activation and has been previously associated with hematological malignancies.

Figures 3 and 4 show the expression levels of CD69 in healthy patients and patients with mixed phenotype acute leukemia (MPAL), respectively. In healthy individuals, CD69 expression is relatively low across most immune cell types (Figure 3). In contrast, MPAL samples show strong overexpression of CD69 in T/NK cells and, notably, in Myeloid cells (Figure 4).

These findings suggest that CD69 overexpression may be indicative of leukemic transformation or immune dysregulation in MPAL. Since over 18,000 genes are profiled in this dataset, identifying compounds that can selectively downregulate oncogene-like targets such as CD69 could be of therapeutic interest in blood cancers.

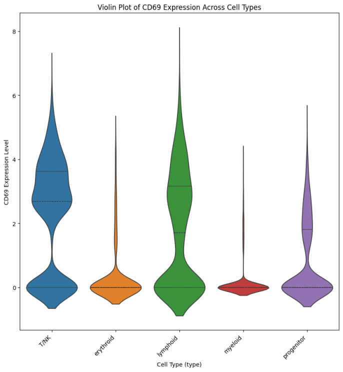


Figure 3: gene CD69 healthy

0.2 Data Augmentation

One of the major challenges in this project is the disparity between the relatively low-dimensional input space—comprising categorical features such as cell types, small molecule (SM) drug names, SMILES strings, and control status—and the high-dimensional output space, which includes differential gene expression values for over 18,000 genes. To bridge this gap and improve model performance, I implemented several strategies for augmenting the input features.

First, I computed summary statistics—mean, standard deviation, and median—of the gene

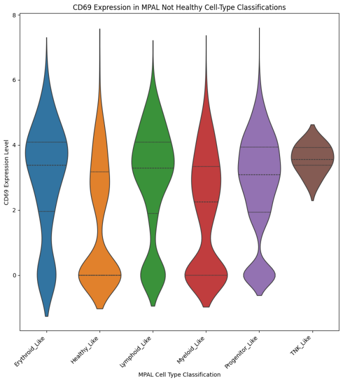


Figure 4: gene CD69 MPAL leukemia

expression levels grouped by both drug and cell type. These statistical features were added to both training and test data. Incorporating these values provided the model with useful priors and significantly improved baseline performance by offering a general view of how each drug-cell pair behaves.

Second, I enriched the chemical representation of drugs by processing their SMILES (Simplified Molecular-Input Line-Entry System) strings. For instance, the compound clotrimazole, shown in Figure 5, was decomposed into chemical substructures (Figures 6 and 7). I then built a sparse count matrix to represent the frequency of these substructures across all 144 compounds. This matrix was used as an additional set of features, enabling the model to capture chemical similarities and differences more explicitly.

Third, I employed the `RDKit` chem informatics toolkit to compute Morgan Finger prints from the SMILES strings. These fingerprints provide a fixed-length numerical encoding that captures the presence and connectivity of molecular substructures. Adding these features allowed the model to learn chemical properties beyond the raw SMILES strings, offering a richer biochemical representation of each drug.

Fourth, since both the cell type and drug compound columns are categorical, I applied standard encoding techniques. One-hot encoding and label encoding were used to transform these categories into numerical formats suitable for input into machine learning models.

Given the large number of additional features introduced—particularly from the sparse substructure count matrix and Morgan Fingerprints—the dimensionality of the input space became very high. To address this and to reduce computational cost, I applied Principal Component Analysis (PCA) to project the augmented input features into a lower-dimensional space while retaining most of the variance. This step not only helped speed up model training but also reduced the risk of overfitting by eliminating noisy and redundant features.

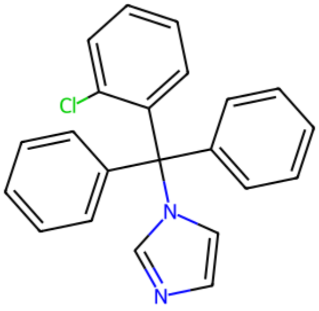


Figure 5: Visualization of clotrimazole.

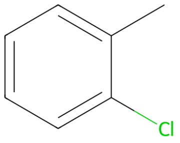


Figure 6: clotrimazole sub module 1

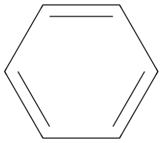
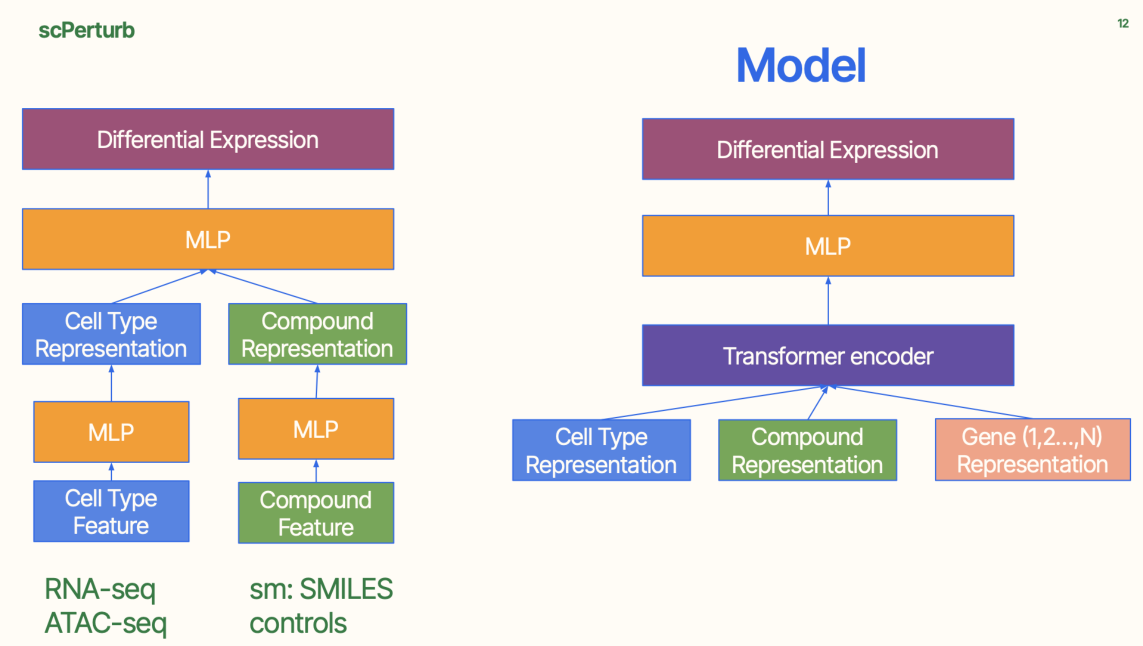


Figure 7: clotrimazole sub module 2

0.3 Models

To model the relationship between cell types, drug compounds, and the resulting gene expression profiles, I implemented and compared two deep learning architectures: a Multi-Layer Perceptron (MLP) and a Transformer-based model, as visualized in Figure 8.

MLP Architecture: The MLP model consists of a deep feedforward network designed to process high-dimensional continuous features. After data augmentation, the input dimension can be as high as 56,000+, so dimensionality reduction using PCA is optionally applied. The architecture includes four fully connected layers with decreasing hidden dimensions: 4096, 2048, 1024, and 512. Each layer is followed by batch normalization, ReLU activation, and dropout to improve generalization. The final output layer maps the 512-dimensional representation to 18,211 gene targets using a linear layer. This model directly



scPerturb

12

Model

Differential Expression

MLP

Cell Type Representation

Compound Representation

MLP

MLP

Cell Type Feature

Compound Feature

RNA-seq
ATAC-seq

sm: SMILES
controls

Differential Expression

MLP

Transformer encoder

Cell Type Representation

Compound Representation

Gene (1,2...,N) Representation

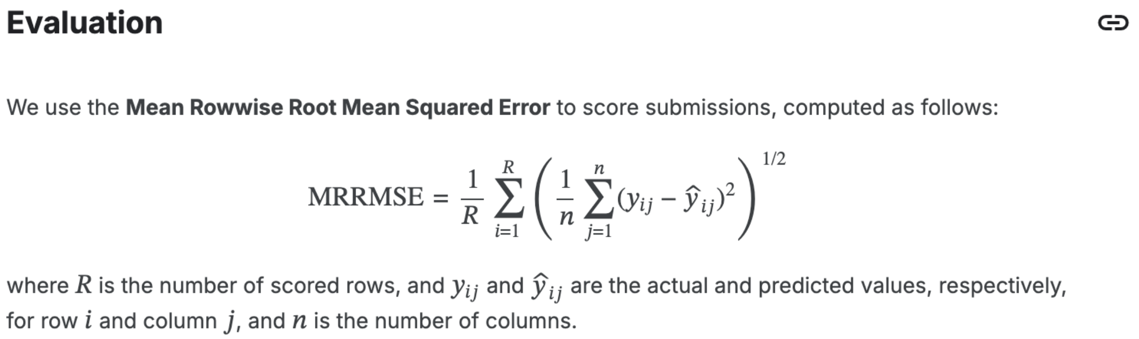
Figure 8: Models

learns a mapping from the augmented numerical features—including statistics (mean, std, median), Morgan fingerprints, and SMILES substructure counts—to gene expression profiles.

Transformer-Based Architecture: The second model is a TabTransformer-like architecture that separately processes categorical and continuous features. Categorical features, such as cell types and drug compounds, are one-hot encoded and projected into an embedding space using a linear layer. These embeddings are then passed through a Transformer encoder with one layer and eight attention heads to capture interactions between discrete categories. In parallel, the continuous features (e.g., gene statistics, SMILES fingerprints) are normalized and projected into the same embedding dimension using a fully connected branch with Layer Norm, ReLU, and dropout. The embeddings from both categorical and continuous branches are concatenated and passed through a final MLP with hidden size 1024 and dropout before producing the final output of 18,211 gene predictions.

Design Motivation: The MLP model leverages the full expressive power of dense connections and is particularly suited to large numerical feature spaces. The Transformer model introduces a more structured inductive bias, allowing it to learn complex interactions among categorical features while still benefiting from continuous inputs. By comparing these two architectures, this project explores how different neural inductive biases affect generalization on a high-dimensional, imbalanced biological prediction task.

Evaluation



Evaluation

We use the **Mean Rowwise Root Mean Squared Error** to score submissions, computed as follows:

$$\text{MRRMSE} = \frac{1}{R}\sum_{i=1}^{R}\left(\frac{1}{n}\sum_{j=1}^{n}(y_{ij}-\widehat{y}_{ij})^2\right)^{1/2}$$

where R is the number of scored rows, and y_{ij} and $\widehat{y}_{ij}$ are the actual and predicted values, respectively, for row i and column j, and n is the number of columns.

Figure 9: MRRMSE loss

To evaluate model performance, I use a custom variant of the Root Mean Square Error (RMSE), computed at the cell-drug pair level. Specifically, for each sample (i.e., a cell type and drug compound pair), I compute the RMSE across the 18,211 predicted gene expression values and then average this score across all samples. This metric is referred to as the Mean Row-wise RMSE (MRRMSE) and is well-suited to this multi-output regression task, as it treats each row independently and prevents genes with inherently higher variance from dominating the loss. The loss function is shown in Figure 9.

Training Setup

Both MLP and Transformer models are trained using the standard mean squared error (MSE) loss. I split the data into training and validation sets and monitored both training and validation loss over 10 epochs. The models are trained using the Adam optimizer with an initial learning rate of 1×10^{-3}, and I use a learning rate scheduler (‘ReduceLROnPlateau‘) to adapt the learning rate based on validation performance. The training loop includes model check pointing, where the model with the lowest validation loss is saved. Figure 9 shows the training and validation loss curves, which help visualize convergence and potential overfitting.

Test Evaluation

After training, I optionally run the model on a held-out test set and compute the final MSE to assess generalization.

Result

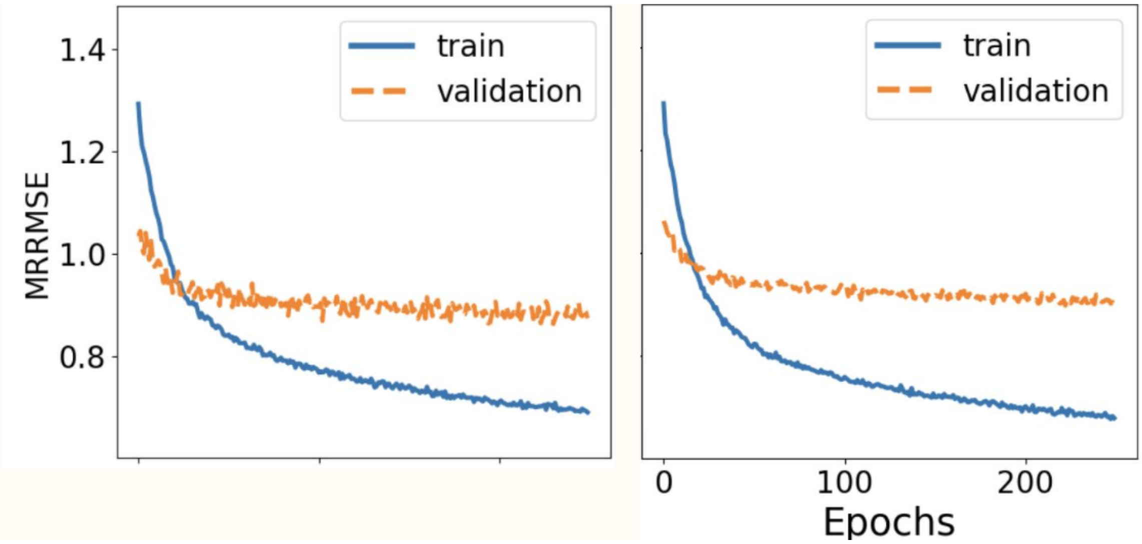


Figure 10: MRRMSE loss

Figure 10 shows the MRRMSE loss curves for both MLP and Transformer models. The MLP model converges more quickly on the validation set, whereas the Transformer requires slightly more epochs to achieve a comparable level of validation loss. However, both models exhibit signs of overfitting after approximately 100 epochs, as indicated by the divergence between steadily decreasing training loss and plateauing validation loss.

MRRMSE	**MLP**	**Transformer**
Validation	0.86	0.90
Test	0.93	0.87

Table 2: MRRMSE scores for MLP and Transformer models on validation and test sets.

As shown in Table 2, while the MLP achieves slightly better performance on the validation set due to faster convergence, it under performs on the hidden test set. In contrast, the Transformer achieves a lower MRRMSE on the test data, suggesting better generalization.

The Transformer model likely benefits from its attention mechanism, which helps capture dependencies between categorical and numerical features. This becomes especially advantageous in the presence of distribution shifts or imbalanced categories—such as the under representation of certain cell types in the training set compared to the test set.

For reference, top-performing models on the Kaggle leader board utilized ensemble techniques combining 4–5 diverse models, achieving MRRMSE scores around 0.75.

Conclusion

This project demonstrates that deep learning combined with feature engineering can significantly enhance the prediction of gene expression changes in single cells subjected to chemical perturbations. By augmenting limited categorical inputs with derived numerical features, such as statistical summaries, SMILES-based substructure counts, and molecular fingerprints, I was able to expand the feature space and improve models toward capturing more biologically meaningful patterns.

Between the two models explored, the Transformer demonstrated superior generalization on the unseen test set, particularly in handling distribution shifts across cell types. This suggests that its attention mechanism is better equipped to model complex, high-dimensional relationships inherent in transcriptomic data. In contrast, the MLP model, while converging faster during training, was more prone to overfitting and exhibited higher test error.

Future work could explore ensemble methods to combine the strengths of multiple models, as well as incorporate graph neural networks to better model molecular structures. In addition, semi-supervised learning could help leverage large amounts of unlabeled perturbation data, potentially improving performance and accelerating the discovery of candidate compounds for leukemia treatment.

Code

The code for scPerturb is available on Github: Github Link

The 2023 NeurIPS Kaggle Competition: Kaggle Link

References

[1] Arús-Pous, J., Johansson, S.V., Prykhodko, O. et al. *Randomized SMILES strings improve the quality of molecular generative models. J Cheminform 11, 71 (2019).* https://doi.org/10.1186/s13321-019-0393-0

[2] Peidli S et al. *scPerturb: harmonized single-cell perturbation data. Nature Methods. (2024).* https://doi.org/10.1038/s41592-023-02144-y

[3] Oelen, R., de Vries, D.H., Brugge, H. et al. *Single-cell RNA-sequencing of peripheral blood mononuclear cells. Nature Communication 13, 3267 (2022).* https://doi.org/10.1038/s41467-022-30893-5

[4] Lotfollahi, M., Naghipourfar, M., Theis, F.J. *scGen predicts single-cell perturbation responses.* Nat Methods 16, 715–721 (2019). https://doi.org/10.1038/s41592-019-0494-8

[5] Cao, Z., Yuan, F. *Predicting drug-induced gene expression profiles using graph convolutional networks and transfer learning.* Bioinformatics 37(19), 3245–3252 (2021). https://doi.org/10.1093/bioinformatics/btab264