SC-MAMBA2: LEVERAGING STATE-SPACE MODELS FOR EFFICIENT SINGLE-CELL ULTRA-LONG TRANSCRIPTOME MODELING

Yalong Zhao[1], **Bowen Zhao**[1,2]*, **Fan Zhang**[1], **Chenfeng He**[1], **Wendao Wu**[1,3]*, **Lipeng Lai**[1†]
[1]XtalPi Innovation Center, XtalPi, Beijing, China
[2]Department of Medicine, McGill University, Canada
[3]School of Mathematical Sciences, Peking University
yalong.zhao@xtalpi.com, bowen.zhao@mail.mcgill.ca, fan.zhang@xtalpi.com
chenfeng.he@xtalpi.com, wuwendao@stu.pku.edu.cn, lipeng.lai@xtalpi.com

ABSTRACT

Single-cell transcriptomics has revolutionized our understanding of cellular heterogeneity, yet modeling ultra-long transcriptome sequences (i.e. number of genes) remains a significant computational challenge. In this study, we introduce SC-MAMBA2, based on the most recent MAMBA2 architecture, as the first application of this architecture integrated with state-space models (SSMs) for single-cell transcriptome modeling. Unlike traditional Transformer-based language models, SC-MAMBA2 leverages the efficiency and scalability of SSMs, enabling to handle longer transcriptome sequences with reduced computational overhead. We introduce unique design adaptations specifically tailored to transcriptome sequences and implement a bidirectional modeling approach under the SSM framework, facilitating comprehensive analysis of whole genome transcriptome sequence. SC-MAMBA2 stands as the largest model in the single-cell transcriptomics domain, with over 150 million parameters, capable of processing transcriptome sequences covering more than 60,000 genes. The model was trained on a dataset of 57 million cells, making it the most comprehensive solution for handling ultra-long sequences to date. Through extensive benchmarking across various downstream tasks, SC-MAMBA2 consistently outperforms state-of-the-art models, demonstrating superior accuracy and computational efficiency. Our results underscore the effectiveness and advanced capabilities of SC-MAMBA2, positioning it as a pivotal tool for future single-cell transcriptome studies.

Keywords single-cell transcriptomics · state-space model · SC-MAMBA2 · ultra-long sequences · gene expression analysis · large-scale modeling · computational biology

1 Introduction

Transfer learning with its applications in computational biology has been instrumental in leveraging large foundation models for the analysis of single-cell transcriptomic data[1]. By pre-training on extensive gene expression datasets, these models acquire a foundational understanding of gene regulation, which can be fine-tuned on specific single-cell transcriptome analysis tasks, such as cell type classification or trajectory inference [2]. This approach circumvents the need for extensive feature engineering, enabling the models to unravel complex patterns within high-dimensional transcriptomic spaces. Consequently, transfer learning facilitates the elucidation of cellular heterogeneity and the discovery of novel biomarkers [3], propelling forward precision medicine and our understanding of intricate biological systems. Overall, the adaptability of pre-trained models, when applied to single-cell transcriptomics, underscores a burgeoning era of transfer learning for single-cell data analysis.

*This work was completed during an internship at XtalPi.

†Corresponding author: lipeng.lai@xtalpi.com

Recent advancements in single-cell transcriptomics have seen the emergence of using innovative methods (e.g., large pre-trained models) to harness the rich biological information, such as scBERT[4], GeneFormer[2], scGPT[5] and scFoundation[6] etc. Among them, scBERT utilizes a BERT-inspired approach to encode gene-gene interactions, offering significant improvements in clustering and classifying cells based on their transcriptomic profiles. GeneFormer introduces a transformer-based model specifically designed to handle ranked gene expression data, providing a robust framework for understanding gene-gene interactions and gene regulation. scGPT adapts the GPT architecture and is a generative pre-trained model to capture the cellular context, enabling enhanced prediction of gene expression patterns. scFoundation amalgamates a vast compendium of single-cell RNA-seq data, leveraging pre-training to establish a comprehensive reference for various cell types. These methods represent a quantum leap in transfer learning applications in single-cell analysis, each contributing to a more nuanced understanding of cellular mechanisms at the single-cell level. However, these models are all based on Transformer architecture, and they still face inherent limitations, particularly the time-consuming inference resulting from the quadratic computation complexity of attention calculation. On the other hand, Mamba, inspired by classical state space models, possesses near-linear scalability concerning sequence length and preserves comparable modeling abilities to transformers[7].

In this work, we introduce SC-MAMBA2, a foundational model for single-cell analysis encompassing over 150 million parameters and pre-trained on an extensive dataset of 57 million cells. Our approach adopts Mamba[7][8] for fast inference and linear scaling of the feature dimensions, which can efficiently model gene-gene interactions within tens of thousands of genes. SC-MAMBA2 devises a unique generative pre-training procedure tailored for transcriptomic data that is inherently non-sequential, while modifying the transformer structure to concurrently acquire representations of both cells and genes. Furthermore, we introduce specialized fine-tuning protocols with distinct objectives to enhance the utility of the pre-trained model across a diverse array of tasks.

The contributions of this work are threefold:

- **Innovative Architecture:** We introduce SC-MAMBA2, the first model to integrate state-space models (SSMs) with the MAMBA framework for single-cell ultra-long transcriptome data. This integration enables efficient and scalable modeling of extensive gene sequences, overcoming the limitations of traditional Transformer-based architectures in handling large-scale biological data.
- **Design Adaptations and Unprecedented Sequence Length Modeling:** We develop unique design modifications tailored to gene sequences and implement a bidirectional modeling approach within the state-space modules. SC-MAMBA2 is capable of modeling full gene sequences encompassing 60,530 genes, representing the largest and most comprehensive sequence length handled in the single-cell transcriptomics domain to date. This capability allows for comprehensive analysis of entire gene transcripts, capturing intricate biological variations and regulatory elements that shorter models cannot accommodate. By successfully modeling such extensive sequences, SC-MAMBA2 provides a more complete and accurate representation of the transcriptome, leading to superior performance in downstream tasks and setting a new benchmark for future studies in the field.
- **Superior Performance:** Through extensive benchmarking, we demonstrate that SC-MAMBA2 outperforms existing state-of-the-art models across multiple downstream tasks, including gene expression quantification, cell type classification, and trajectory inference. This superior performance underscores SC-MAMBA2's efficacy in capturing the full breadth of transcriptomic information while maintaining computational efficiency, establishing it as the most advanced and effective model in the single-cell transcriptomics landscape.

By addressing the computational challenges inherent in single-cell ultra-long transcriptome modeling, SC-MAMBA2 paves the way for more comprehensive and efficient analyses, ultimately contributing to a deeper understanding of cellular biology and the mechanisms driving complex biological systems.

2 Results

We developed SC-MAMBA2, a large-scale model with 30 million parameters to represent 60,531 genes, pre-trained on over 30 million scRNA-seq datasets from CELLxGENE[9]. SC-MAMBA2 is specifically designed for the single-cell domain, offering extensive gene context length, coverage, and scalability. To adapt to the unique characteristics of single-cell data, we modified Mamba into a bidirectional architecture to efficiently learn relationships among all genes. This scalable state space model (SSM) leverages bidirectional contextual relationships, enhancing the capacity for single-cell analysis. SC-MAMBA2 fine-tuned on specific tasks has demonstrated superior performance over benchmark methods in cell annotation, multi-batch data integration, multi-omics data integration, and perturbation prediction. These results highlight the model's capability to accurately capture complex dependencies between genes in single cells.

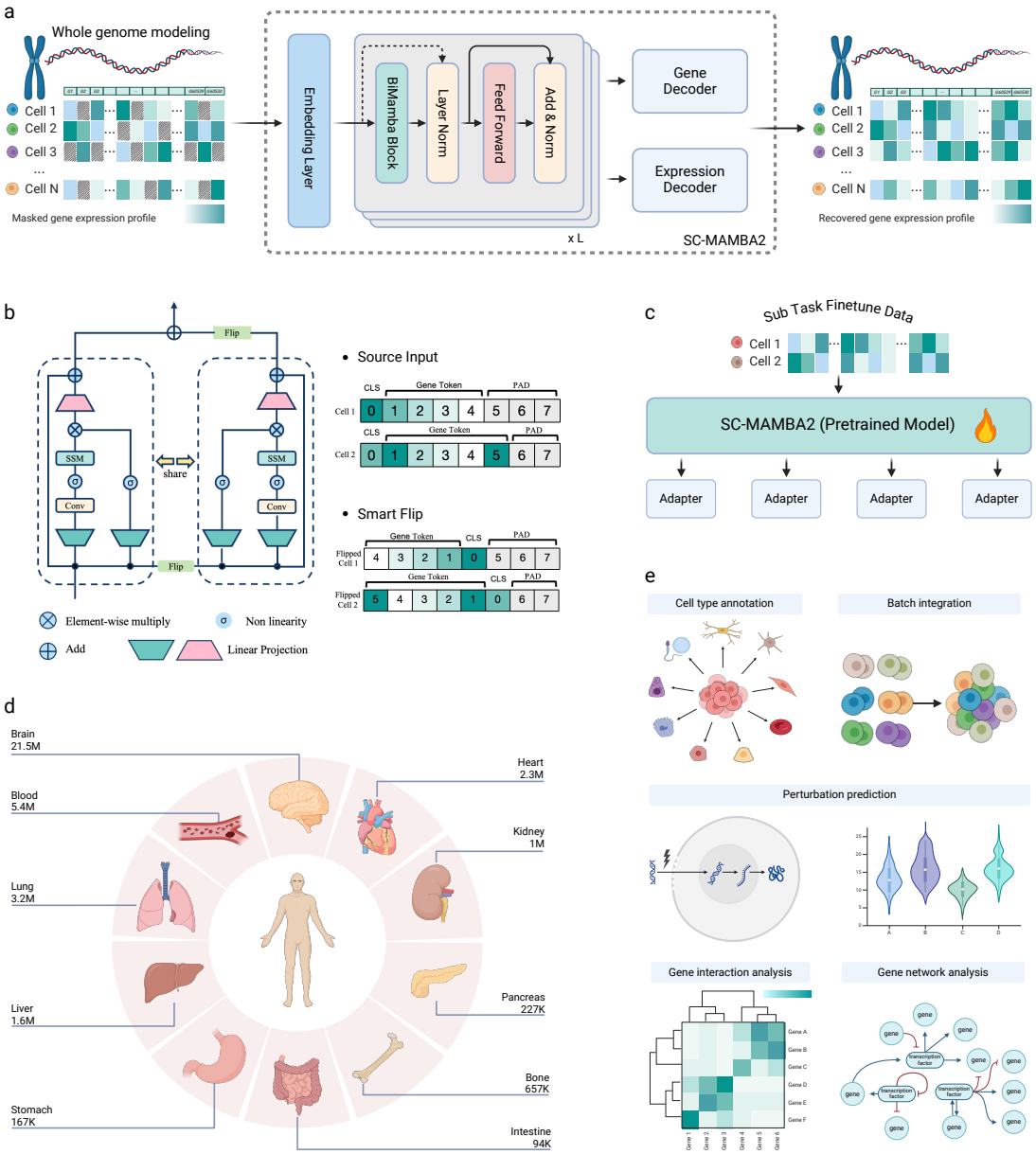


Figure 1: **Schematic overview of the SC-MAMBA2 architecture and its applications in single-cell transcriptomics. a,** Whole genome modeling pipeline using SC-MAMBA2. The masked gene expression profiles from multiple cells are processed through the SC-MAMBA2 architecture. The output is then passed to both Gene and Expression Decoders, recovering the gene expression profile of each cell. **b,** Smart Flip bidirectional input mechanism using state-space models (SSMs). The architecture supports both source input, where the gene tokens are processed in a specific order, and a Smart Flip mechanism, where the sequence is reversed, enhancing bidirectional transcriptome modeling. **c,** Adapter-based fine-tuning of the SC-MAMBA2 pre-trained model for sub-task adaptation using specific data. **d,** Extensive and diverse training data sourced from CELLxGENE, covering multiple organs and tissues. **e,** Downstream applications of SC-MAMBA2, including cell type annotation, batch integration, perturbation prediction, gene interaction analysis, and gene regulatory network inference, showcasing its versatility in various single-cell tasks.

2.1 SC-MAMBA2 improves multi-batch and multi-omic integration

Multi-batch integration. Integrating multiple scRNA-seq datasets from different batches poses unique challenges in simultaneously preserving the biological variance of integrated data and removing technical batch effects. To integrate

sequencing samples, we fine-tuned SC-MAMBA2 in a self-supervised manner by learning unified cell presentations that recover masked gene expression. In our benchmarking experiments, we compared SC-MAMBA2 with scGPT and three popular integration methods: scVI[10], Seurat[11] and Harmony[12]. The evaluation was conducted in datasets peripheral blood mononuclear cell (PBMC) 10k (two batches)[13]. In the PBMC 10k dataset, SC-MAMBA2 successfully separated all cell types. The scGPT demonstrated a 5%-10% improvement over scVI Seurat and Harmony with an increase in AvgBIO from 0.724 to 0.821, the superior integration performance of SC-MAMBA2 was further supported by its high biological conservation score, with an AvgBIO score of 0.836, which was 1.83% higher than scGPT (Fig 3a, Fig 3b).

Multi-omic integration. Single-cell multi-omics data, which combine multiple views of genetic regulation such as epigenetic, transcriptomic and translation activities, present a unique challenge in aggregating cell representations while preserving biological signals. SC-MAMBA2 addresses this challenge by effectively extracting integrated cell embeddings across different omics datasets. In the case of the 10x Multiome PBMC dataset[14], which includes joint gene expression and chromatin accessibility measurements, we compared SC-MAMBA2 with transformer based model scGPT and two state-of-the-art methods, scGLUE[15] and Seurat (v.4). SC-MAMBA2 presented more defined cluster structures than scGPT, Although the improvement in the AvgBIO score is only 0.5%, SC-MAMBA2 stands out as the only method that successfully generates a distinct cluster for CD14+ and CD16+ naive cells (Fig 3c, Fig3d). Fig 3e showed the UMAP plot of paired gene expression and protein abundance dataset from bone marrow mononuclear cells (BMMCs)[16]. This dataset contains 90,000 cells and 12 batches and 48 cell types. SC-MAMBA2 also yielded a marginal performance improvement in the AvgBIO score compared with scGPT.

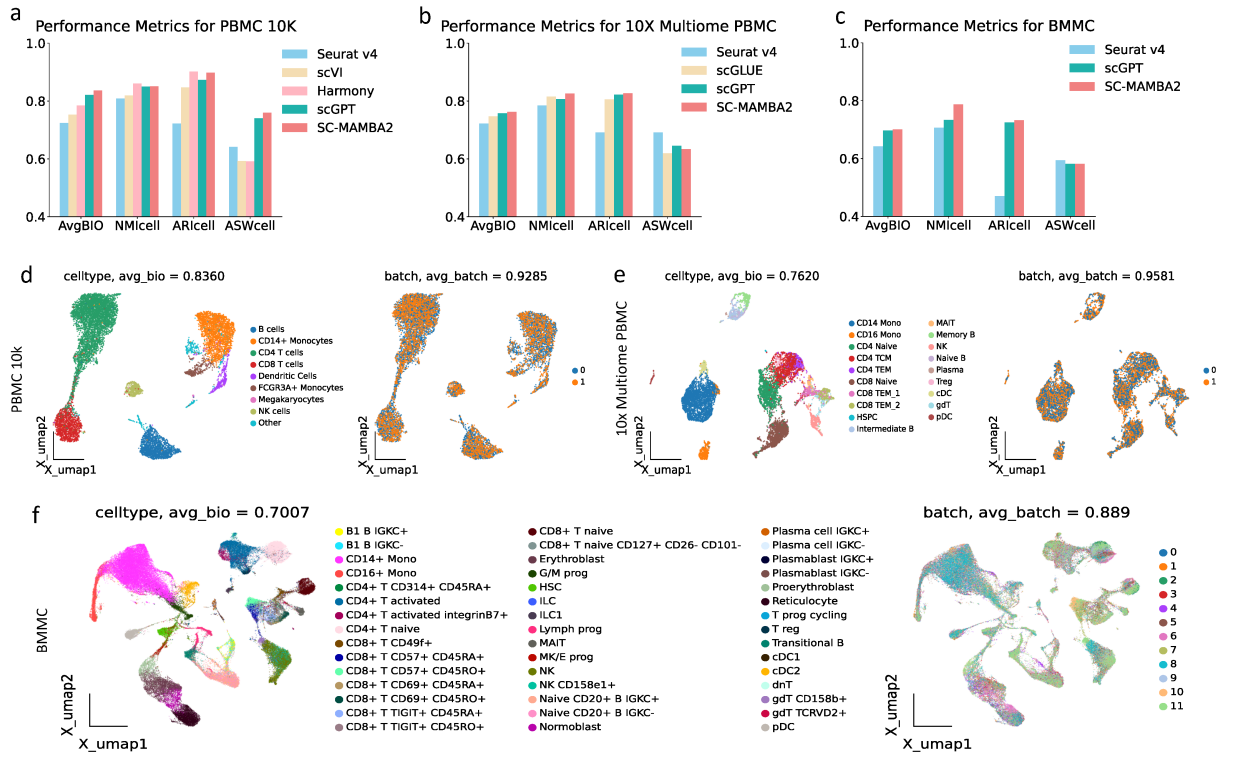


Figure 2: **SC-MAMBA2 performance on multi-batch and multi-omic integration**. **a.** Comparison of clustering performance among Seurat v4, scVI, Harmony, scGPT and SC-MAMBA2 on the PBMC 10K dataset. **b.** Comparison of clustering performance among Seurat v4, scGLUE, scGPT and SC-MAMBA2 on the 10X multiome PBMC dataset. **c.** Comparison of clustering performance among Seurat v4, scGPT and SC-MAMBA2 on the BMMC dataset **d.** The UMAP of fine-tuned SC-MAMBA2 on the PBMC 10k dataset for the cell type-clustering and batch integration task. The UMAP plot of learned cell embeddings is colored by cell types (left) and different batches (right). **e, f.** The UMAP of fine-tuned SC-MAMBA2 on the PBMC 10k multiome dataset (e) and BMMC (f) dataset for the multi-omics cell type-clustering (left) and batch integration task (right)

2.2 SC-MAMBA2 performance on cell type annotation

To address the cell type-annotation task, we conducted fine-tuning of the model using a reference set that contained ground truth labels, and subsequently assessed the annotation performance using a separate query set. We retained the common set of gene tokens between the pretrained foundation model and the reference set. Prior to fine-tuning the model, gene expression values underwent normalization, log transformation, and binning. The fine-tuned model was initialized using all pretrained model weights, with the exception of the output cell type classifier, which was initialized randomly. During the training process, all gene tokens, regardless of their expression values being zero or non-zero, were included. The objective during the fine-tuning process was to minimize the classification loss using the cell type-classification fine-tuning approach.

We fine-tune the pretrained SC-MAMBA2 for cell type annotation on three datasets: myeloid, ms and hPancreas Notably, scMamba achieved higher accuracy, precision, recall and macro f1 compared with scGPT (Fig. 3). Fig 3 visualizes the cell embedding in the fin-tuned SC-MAMBA2 in three datasets, which demonstrate high intra-cell type similarities. We benchmarked the fine-tuned scMamba against three other transformer-based methods, TOSICA[17], scBERT and scGPT across the three datasets. Our results indicate that SC-MAMBA2 consistently outperforms the other models across all datasets. In the Myeloid dataset, while scGPT demonstrated a substantial improvement over TOSICA with an increase in accuracy from 0.488 to 0.642, scMamba marginally enhanced these results, achieving an accuracy of 0.644. Notably, scMamba's precision and recall were 0.381 and 0.366, respectively, surpassing scGPT by 1.5% and 1.9%. In the Multiple Sclerosis dataset, scMamba achieved an accuracy of 0.869, outstripping scGPT by 1.3%. Similarly, precision and recall were slightly higher for SC-MAMBA2, with increases of 0.3% and 1.0%, respectively. The MacroF1 score also reflected this enhancement, with scMamba achieving 0.723 compared to scGPT's 0.703. The hPancreas dataset displayed the highest accuracy rates, with scMamba reaching 0.975, a 0.7% increase over scGPT. Here, scMamba's precision and recall saw a more pronounced improvement of 4.1% and 4.3%, respectively, compared to scGPT. Fig 3a, Fig 3b and Fig 3c offers a visualization of the cell embeddings post fine-tuning with SC-MAMBA2. These visualizations highlight the strong similarity within each cell type, indicating the model's effectiveness in capturing the nuances of cell type-specific expression profiles.

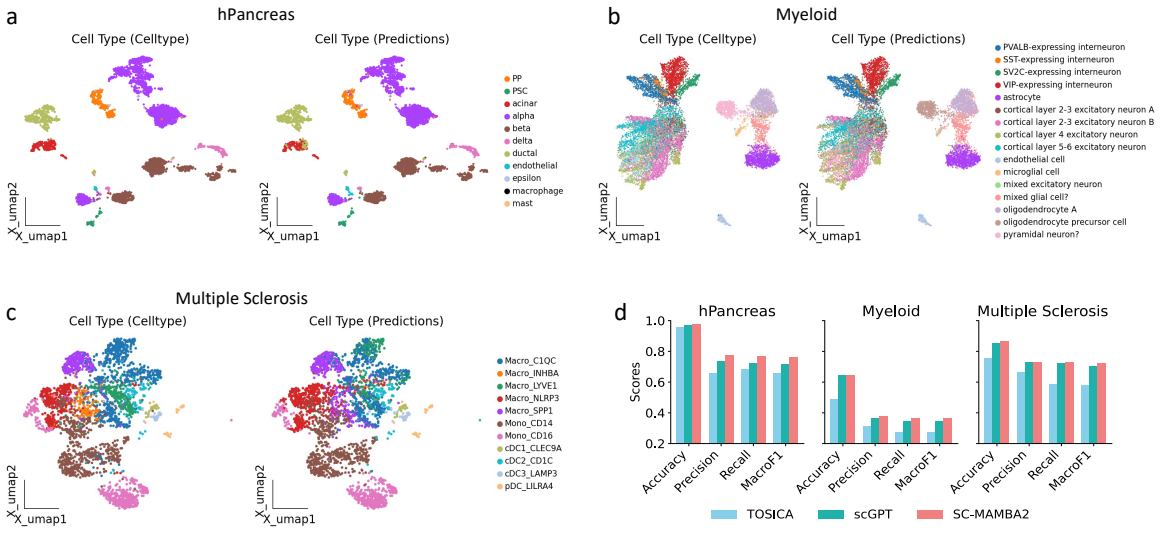


Figure 3: **Cell type annotation with SC-MAMBA2**. UMAP of gene expression of cells from the human pancreas dataset (a), myeloid (b) and multiple sclerosis (c), colored by the cell types annotated in the original study (left) and by the cell types predicted by the fine-tuned SC-MAMBA2. (right). (c), The barplot of accuracy, precision, recall and macro f1 score compared between SC-MAMBA2 and TOSICA, scGPT in three datasets

2.3 SC-MAMBA2 improves In silico perturbation and reverse perturbation prediction

Gene regulation functions as a hierarchical network, wherein a change in the expression of one gene can potentially influence the expression of multiple other genes [18]. Predicting the expression level changes of other genes in response to a single gene (i.e. perturbation) can facilitate the elucidation of the hierarchical structure within gene networks. On the other hand, identifying key genes responsible for alterations in gene expression in specific status (i.e. reverse

perturbation), such as diseases, can inform the development of therapies aimed at the core regulatory elements that drive the disease, rather than focusing on peripheral downstream effects that may not be directly disease related [5].

In this study, we evaluated SC-MAMBA2 and scGPT using the Norman dataset [19], which includes 131 two-gene perturbations and 105 single-gene perturbations, to assess the models' capability for both perturbation and reverse perturbation prediction tasks. Specifically, we first fine-tuned the models using a subset (n = 138) of the perturbations and tasked them with predicting the post-perturbation gene expression. The fine-tuned models were then applied to predict gene expression level changes after other unseen perturbations (Fig. 3a). SC-MAMBA2 consistently outperformed scGPT, achieving a higher Pearson correlation by at least 5% margins when correlating with changes of measured gene expression. Notably, SC-MAMBA2 performed particularly well for two-gene perturbations when neither gene was observed during the fine-tuning process, suggesting SC-MAMBA2 generalizes better for completely novel perturbed genes.

We also evaluated the fine-tuned models for the reverse perturbation task, aiming to predict the source perturbed genes from resulting cell states. This application is ideal for identifying potential therapeutic gene targets for certain diseases. Following the approach described by Cui H. et al.[5] , we demonstrated the effectiveness of SC-MAMBA2 on reverse perturbation using the Norman dataset [19]. Experiment shows SC-MAMBA2 achieved higher accuracy than scGPT. By exploring the combination space of top20 expression genes (Fig. 4c), SC-MAMBA2 accurately identified three out of seven unseen perturbations in the test dataset within its top 1 predictions, whereas scGPT correctly predicted only one. When extending the evaluation to the top 1–8 predictions, SC-MAMBA2 consistently outperformed scGPT (Fig. 4d) by achieving a higher number of accurate predictions.

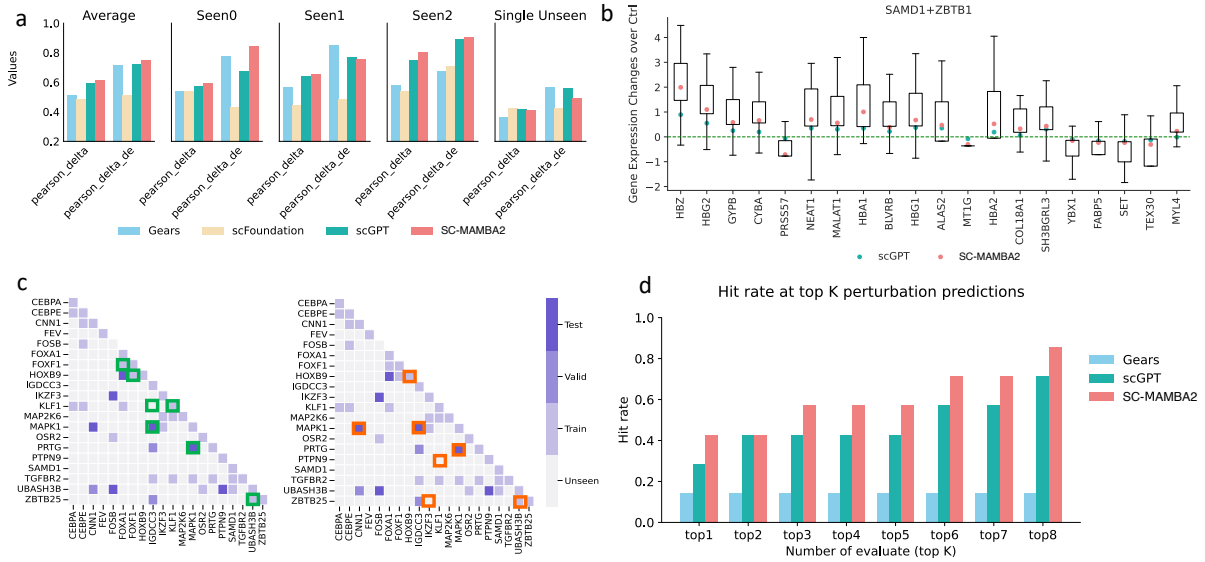


Figure 4: **Evaluation of models accuracy for perturbation and reverse perturbation prediction**. **a.** SC-MAMBA2 scGPT and GEARS[20] performance on perturbation task. Perturbations in the test dataset were separated into 4 categories: Single Unseen (single-gene perturbed and this gene was not in the fine-tune training dataset), Seen0 (two-genes perturbed and both genes were not in the fine-tune training dataset), Seen1 (two-genes perturbed and one of the genes was in the fine-tune training dataset) and Seen2 (two-genes perturbed and both genes were in the fine-tune training dataset). The averaged metrics were also shown (left). Two metrics were included for evaluating the accuracy: pearson_delta (pearson correlation between the predicted versus observed gene expression after subtracting the control cells' gene expression profile) and pearson_delta_de (pearson_delta of the all the differentially expressed genes) **b.** Example perturbation (SAMD1+ZBTB1) prediction results, showing the distribution of predicted and observed gene expression for the top 20 differentially expressed genes. The box denotes the interquartile range of expression change. The green dot represent the prediction value from scGPT and the rid dot represent the . **c.** Visualization of scGPT and SC-MAMBA2 reverse perturbation prediction results. Heatmap show the possible perturbation combinations. Grid color represents the experiment type (train, valid, test and unseen). Green square boxes are scGPT prediction results (left) and red boxes are SC-MAMBA2 results (right). **d.** Comparison of Reverse Perturbation Accuracy Between scGPT and SC-MAMBA2. The hit rate was defined as the number of correct perturbations accurately predicted within the top K (1 to 8) rankings.

3 Discussion

In this work, we established SC-MAMBA2, so far the largest foundation model for modeling complex single-cell gene expression profile. The efficiency of the Mamba algorithm enabled SC-MAMBA2 to handle transcriptomic data with sequence lengths (i.e., the number of genes) previously unattainable. Experiments demonstrated the advantages of extending the sequence length, allowing SC-MAMBA2 to outperform several state-of-the-art single-cell foundation models on various downstream tasks. As a foundation model for modeling single-cell gene expression profiles, besides the downstream tasks we demonstrated, SC-MAMBA2 can also be fine-tuned for other tasks, such as gene network inference, in-silico treatment analysis, and more [2]. The computational efficiency of SC-MAMBA2 holds significant potential for modeling complex gene regulations as more data becoming available. The continuous accumulation of single-cell data in the future is expected to enhance SC-MAMBA2's accuracy and effectiveness.

4 Methods

4.1 State Space Models

State Space Models (SSMs) are traditionally used to describe the dynamics of continuous systems by transforming an input sequence $x(t) \in \mathbb{R}$ into a latent state representation $h(t) \in \mathbb{R}^N$. This latent state is then utilized to generate an output sequence $y(t) \in \mathbb{R}$. The mathematical formulation of an SSM is given by:

$$\dot{h}(t) = \hat{A}h(t) + \hat{B}x(t), \quad y(t) = Ch(t), \tag{1}$$

where $\hat{A} \in \mathbb{R}^{N\times N}$, $\hat{B} \in \mathbb{R}^{N\times 1}$, and $C \in \mathbb{R}^{1\times N}$ are parameters. To effectively incorporate continuous SSMs into deep learning architectures, discretization is essential. This involves introducing a time step parameter $\Delta \in \mathbb{R}$ and applying the zero-order hold (ZOH) technique for discretization.

Through this process, the continuous matrices $\hat{A}$ and $\hat{B}$ are transformed into their discrete counterparts, A and B. Consequently, Equation (1) is reformulated in discrete form as shown in Equation (2), making it more suitable for implementation in modern computational frameworks:

$$h(t) = Ah(t-1) + Bx(t), \quad y(t) = Ch(t), \tag{2}$$

where $A = e^{\Delta\hat{A}}, \quad B = (\Delta\hat{A})^{-1}(e^{\Delta\hat{A}} - I)\hat{B} \approx \Delta\hat{B},$

and I denotes the identity matrix. Additionally, the process described in Equation (2) can be implemented globally in a convolutional manner as:

$$y = x * \mathbf{K}, \quad \mathbf{K} = \left(CB, CAB, \ldots, CA^{L-1}B\right), \tag{3}$$

where $\mathbf{K} \in \mathbb{R}^L$ represents the convolution kernel. Recent work by Mamba proposes a method in which the parameters B, C, and Δ are input-independent, addressing limitations inherent in previous Linear Time Invariant (LTI) SSM models. This enhancement improves the adaptability and performance of SSMs.

State Space Duality

Mamba2 recently introduced the concept of State Space Duality (SSD), simplifying the matrix $\mathbf{A}$ into a scalar. This specific case of selective State Space Models (SSMs) can be applied in both linear and quadratic forms. Without loss of generality, the matrix transformation form of selective state space models is represented as follows:

$$y(t) = \sum_{i=1}^{t} C_t^T \mathbf{A}_{t:i+1}\mathbf{B}_i x(i), \quad \text{where} \quad \mathbf{A}_{t:i} = \prod_{i=2}^{t} \mathbf{A}_i, \tag{4}$$

$$y = \text{SSM}(\mathbf{A}, \mathbf{B}, \mathbf{C})(x) = \mathbf{F}x, \quad \text{where} \quad \mathbf{F}_{ji} = C_j^T \mathbf{A}_{j:i}\mathbf{B}_i. \tag{5}$$

When A_i is reduced to a scalar, the quadratic form of Equation (4) can be reformulated as:

$$y = \mathbf{F}x = \mathbf{M} \cdot \left(C^T \mathbf{B}\right) x, \quad \text{where} \quad \mathbf{M}_{ij} = \begin{cases} A_{i+1} \times \cdots \times A_j & \text{if } i > j \\ 1 & \text{if } i = j \\ 0 & \text{if } i < j, \end{cases} \tag{6}$$

while its linear form is expressed as:

$$h(t) = \mathbf{A}_t h(t-1) + \mathbf{B}_t x(t), \quad y(t) = C_t h(t). \tag{7}$$

4.2 SC-MAMBA2

Pre-processing

In the data preprocessing stage, we first perform expression level statistics on the data. The number of genes expressed by each cell is denoted as N_i. To ensure uniform input dimensions for batch processing in neural networks, we specify a maximum sequence length M. Depending on N_i, we process each cell's expression data as follows:

$$X_i = \begin{cases} x_i, & \text{if } N_i = M \\ \text{pad}(x_i, M), & \text{if } N_i < M \\ \text{random_sample}(x_i, M), & \text{if } N_i > M \end{cases} \tag{8}$$

where X_i is the processed expression vector of length M for cell i, and x_i is the original expression vector.

The embedding of SC-MAMBA2 consists of two main parts: gene name encoding and expression value encoding.

Gene Name Encoding

Each gene g_j is tokenized and represented as a high-dimensional vector, similar to word embeddings in NLP. Specifically, each gene is indexed based on the dataset's gene list (e.g., `CELLxGENE`) and encoded via a gene name encoder:

$$G_j = \text{GeneNameEncoder}(g_j), \quad j = 1, 2, \ldots, M \tag{9}$$

where $G_j \in \mathbb{R}^d$ denotes the gene embedding in d-dimensional space.

This tokenization creates a dictionary that maps each gene name g_j to its corresponding vector, forming the set:

$$G = \{G_1, G_2, \ldots, G_M\} \tag{10}$$

These embeddings are then utilized for downstream analysis, enabling a structured and learnable representation of gene names.

Expression Value Encoding

The expression values are binned according to their relative expression levels across cells. This binning process helps mitigate batch effects. Let B denote the number of bins. Each binned value e_j is then passed through an expression value encoder, which uses a continuous mapping to project the binned expression values into a d-dimensional space:

$$E_j = \text{ExpressionValueEncoder}(e_j; B), \quad j = 1, 2, \ldots, M \tag{11}$$

where $E_j \in \mathbb{R}^d$ represents the encoded expression value in the d-dimensional space, and B indicates the total number of bins used in the binning process.

Cell Embedding

The embedding C_j for each gene in a cell is computed by adding the gene name encoding and the expression value encoding:

$$C_j = G_j + E_j, \quad j = 1, 2, \ldots, M \tag{12}$$

The complete embedding sequence for cell i is:

$$C_i = \{C_1, C_2, \ldots, C_M\} \in \mathbb{R}^{M \times d} \tag{13}$$

To facilitate the self-supervised training objective, we mask a certain proportion of the encoded expression values E_j:

$$\tilde{E}_j = \text{Mask}(E_j), \quad j = 1, 2, \ldots, M \tag{14}$$

The model is then trained to predict the masked expression values $\tilde{E}_j$ based on the context provided by the unmasked tokens. This masking mechanism allows the model to learn representations that generalize across different cells and genes.

Training

Since the original Mamba model is autoregressive and cannot capture bidirectional contextual relationships, we specifically designed the BiMamba module to process the embedding sequence C_i. The input sequence C_i is first reversed to create a "flipped" sequence:

$$\text{Flipped } C_i = \text{Reverse}(C_i) \tag{15}$$

Both C_i and Flipped C_i are independently processed by weight-shared unidirectional Mamba modules:

$$C_i^{(l)} = \text{Mamba}(C_i), \quad \text{Flipped } C_i^{(l)} = \text{Mamba}(\text{Flipped } C_i) \tag{16}$$

The flipped sequence is then reversed back to its original order:

$$\text{Flipped } C_i^{(l)} = \text{Reverse}(\text{Flipped } C_i^{(l)}) \tag{17}$$

The outputs from both directions are combined through summation:

$$\hat{C}_i = C_i^{(l)} + \text{Flipped } C_i^{(l)} \tag{18}$$

This combined sequence $\hat{C}_i$ is passed to the next layer in the BiMamba block. The Smart Padding mechanism ensures that only meaningful tokens (e.g., CLS token and gene tokens) are flipped, while padding tokens remain in their original positions to prevent artifacts during the reversing process.

Inspired by the Transformer architecture, conventional attention modules are replaced with BiMamba modules to construct a deep architecture capable of capturing bidirectional relationships. The final embedding sequence C_i is passed through a stack of L BiMamba layers, where the output of the l-th layer is given by:

$$C_i^{(0)} = C_i \tag{19}$$

$$C_i^{(l)} = \text{BiMamba}_l(C_i^{(l-1)}), \quad \forall l \in [1, L] \tag{20}$$

where $C_i^{(l)}$ denotes the output of the l-th BiMamba layer. Each BiMamba_l builds upon the representation learned from the previous layer, and the output of each layer is used as the input to the next layer in the stack. This formulation ensures that the model progressively refines its representation, effectively capturing complex bidirectional relationships within the gene expression data.

loss

The gene expression prediction module processes each gene token to output the predicted binned expression value $\hat{E}_j$. The loss function is computed as the mean squared error (MSE) between the predicted binned expression values $\hat{E}_j$ and the original binned expressions E_j:

$$\mathcal{L} = \frac{1}{M} \sum_{j=1}^{M} \left(\hat{E}_j - E_j \right)^2 \tag{21}$$

This loss function enables end-to-end training of the model, allowing it to learn to predict masked expression values based on the context. By stacking multiple BiMamba layers, the model is trained to capture complex bidirectional relationships within the gene expression data.

4.3 Cell type annotation

To address the cell type annotation task, we fine-tuned the model using a reference dataset that included true labels and subsequently evaluated its annotation performance on an independent query dataset. We retained the gene markers shared between the pre-trained base model and the reference dataset. During fine-tuning, all gene markers were considered, regardless of whether their expression values were zero. Prior to fine-tuning the model, the gene expression values underwent normalization, log-transformation using Scanpy[21], and binning consistent with the pre-training process. The fine-tuned model was initialized with the weights of the pre-trained model, except for the cell type classifier, which was randomly initialized. During fine-tuning, the objective was to minimize the classification loss using a cell type classification fine-tuning approach. We used sklearn[22] to compute the supervised evaluation metrics.

4.4 Batch and Multi-omics integration

In the multi-batch integration task, we first matched the genes in the downstream task data with the SC-MAMBA2 vocabulary to select the shared genes, and then identified the highly variable genes across multiple batches. Similar to the cell type annotation task, we did not filter out genes with zero expression in each cell. All filtered genes, regardless of whether their expression values were zero or non-zero, were utilized. Additionally, we inferred the CLS token for each cell as input and employed an adversarial domain adaptation approach[23] to map these tokens to the same space, achieving batch correction.

For single-cell multi-omics integration tasks, to leverage the pretrained single-cell transcriptomics weights, we converted the data into transcriptomics format. Specifically, single-cell ATAC-seq data was transformed into a gene activity matrix using ArchR[24], while single-cell protein data did not require conversion. We applied preprocessing similar to that used for multi-batch data integration, obtaining a CLS token for each cell as its representation. Additionally, we added modality tokens to indicate the data source for each cell. After obtaining the integrated cell representations, we employed an adversarial domain adaptation approach to map these representations into the same space, achieving multi-omics data integration.

We used scib[25] to evaluate the integration performance, where AvgBio represents the average of ARI, NMI, and ASW cell scores.

4.5 Perturbation and reverse perturbation

To prepare for the perturbation prediction task, we began by selecting highly variable genes (HVGs) and preprocessing their expression values before initiating model training. We utilized the parameters of the embedding layers and transformer layers from the pre-existing model to initialize the fine-tuned model. During the fine-tuning process, we included all gene tokens, encompassing both zero and non-zero expression values. Notable adjustments were made to the input for the perturbation prediction task. Firstly, we employed log1p-transformed expression values as both input and target values, deviating from binned values, in order to enhance the accuracy of predicting absolute post-perturbation expression for this particular task. Secondly, we introduced a binary condition token at each input gene position to signify whether the gene had been perturbed.

We adopted the perturb-GEP fine-tuning objective with additional modifications to the training setup. Instead of utilizing masked and unmasked expression values of the same cell as the input and learning target, we opted to use a control cell as the input and the perturbed cell as the target. This was achieved by randomly pairing a non-perturbed control cell with each perturbed cell to form input–target pairs. The input values encompassed all gene expression values in the control cells. As a result, the model learned to forecast post-perturbation responses based on control gene expression and the perturbation token.

Data availability

The pretraining datasets were sourced from the CELLxGENE census (release version 1 July 2024, accessible at https://chanzuckerberg.github.io/cellxgene-census/python-api.html and https://cellxgene.cziscience.com/). For annotation tasks, the MS dataset was retrieved from the Gene Expression Atlas (https://www.ebi.ac.uk/gxa/sc/experiments/E-HCAD-35), while the myeloid dataset is publicly available via GEO (GSE154763). The processed human pancreas dataset was obtained from https://github.com/JackieHanLab/TOSICA. Reference mapping utilized the Lung-Kim dataset from the Curated Cancer Cell Atlas (https://www.weizmann.ac.il/sites/3CA/lung), and the processed COVID-19 dataset was accessed from https://github.com/theislab/scarches-reproducibility. Perturbation prediction tasks used the Norman and Adamson datasets from Harvard Dataverse (https://dataverse.harvard.edu/api/access/datafile/6154020 and https://dataverse.harvard.edu/api/access/datafile/6154417). The Replogle dataset was retrieved from https://gwps.wi.mit.edu/. For batch integration, the PBMC 10k dataset was accessed via scVI tools (https://scvi-tools.org/), while the perirhinal cortex dataset was sourced from the CELLxGENE Human Brain Cell Atlas v1.0 (https://cellxgene.cziscience.com/collections/283d65eb-dd53-496d-adb7-7570c7caa443). The multi-omic integration task utilized the 10x Multiome PBMC dataset (https://scglue.readthedocs.io/en/latest/data.html), the BMMC dataset (GSE194122) from GEO, and the ASAP PBMC dataset from https://github.com/PeterZZQ/scMoMaT/tree/main/data/real/ASAP-PBMC.

Acknowledgments

We are deeply grateful to the XtalPI Innovation Center (XIC) for the financial support and provision of essential computational resources that made this research possible. Special thanks go to Dr. Zhenghui Wang from XIC for her

invaluable insights and constructive feedback throughout the study. We also appreciate the collaborative support of our partner, Gnosis Neurodynamics, whose inspiring discussions significantly contributed to the development of this work. Lastly, we acknowledge the Chan Zuckerberg Initiative for providing the CELLxGENE dataset, as well as all other publicly available datasets utilized in this study, which have been instrumental in advancing our research and the field of single-cell transcriptomics.

References

[1] Nicolae Sapoval, Amirali Aghazadeh, Michael G Nute, Dinler A Antunes, Advait Balaji, Richard Baraniuk, CJ Barberan, Ruth Dannenfelser, Chen Dun, Mohammadamin Edrisi, et al. Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, 13(1):1728, 2022.

[2] 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. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

[3] Axel Kowald, Israel Barrantes, Steffen Möller, Daniel Palmer, Hugo Murua Escobar, Anne Schwerk, and Georg Fuellen. Transfer learning of clinical outcomes from preclinical molecular data, principles and perspectives. *Briefings in Bioinformatics*, 23(3):bbac133, 2022.

[4] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

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

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

[7] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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

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

[11] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[12] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[13] Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, et al. A python library for probabilistic analysis of single-cell omics data. *Nature biotechnology*, 40(2):163–166, 2022.

[14] 10X Genomics. PBMC from a healthy donor, single cell multiome ATAC gene expression demonstration data by Cell Ranger ARC 1.0.0. `https://support.10xgenomics.com/single-cell-multiome-atac-gex/datasets/1.0.0/pbmc_granulocyte_sorted_10k`, 2020.

[15] Zhi-Jie Cao and Ge Gao. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, 40(10):1458–1466, 2022.

[16] Malte D Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann T Chen, Louise Deconinck, Angela M Detweiler, Alejandro A Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021.

[17] Jiawei Chen, Hao Xu, Wanyu Tao, Zhaoxiong Chen, Yuxuan Zhao, and Jing-Dong J Han. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1):223, 2023.

[18] Lesley T MacNeil and Albertha JM Walhout. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome research*, 21(5):645–657, 2011.

[19] Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

[20] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2024.

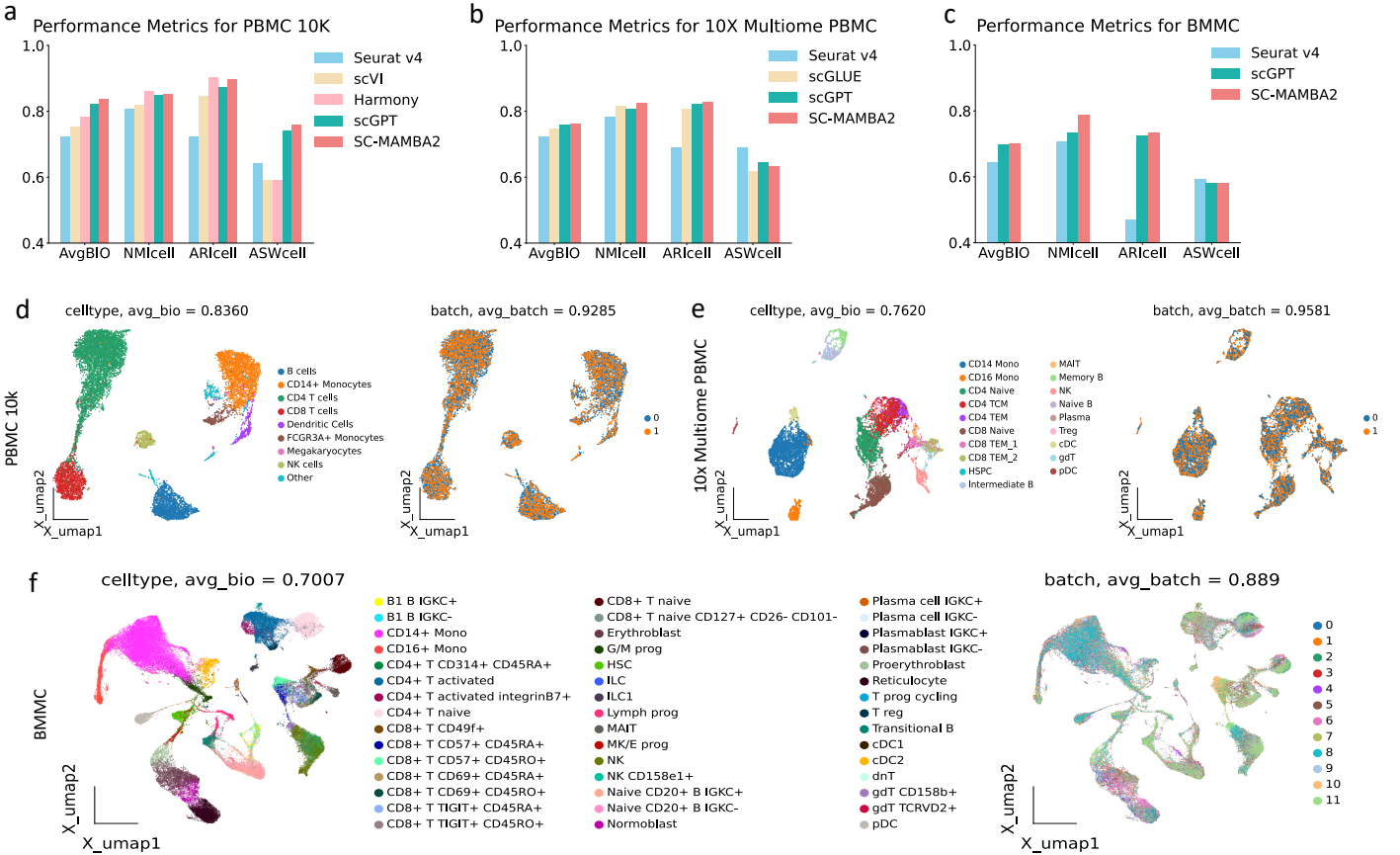
[21] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

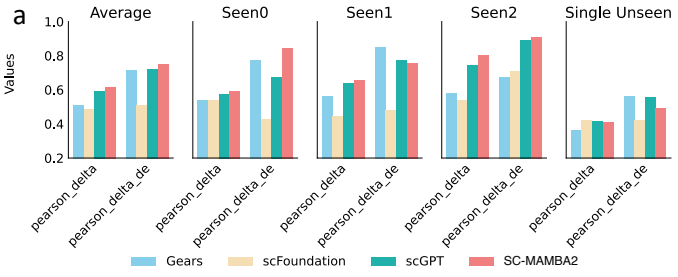
[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

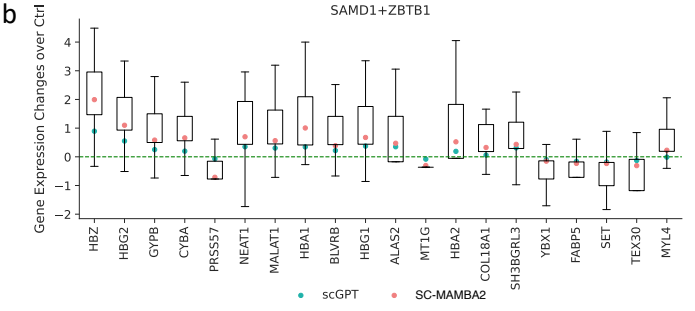
[23] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

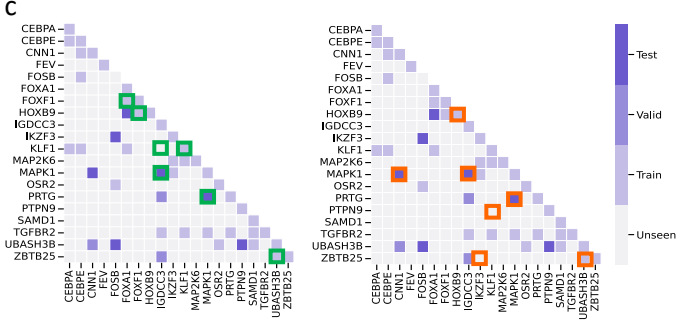
[24] Jeffrey M Granja, M Ryan Corces, Sarah E Pierce, S Tansu Bagdatli, Hani Choudhry, Howard Y Chang, and William J Greenleaf. Archr is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nature genetics*, 53(3):403–411, 2021.

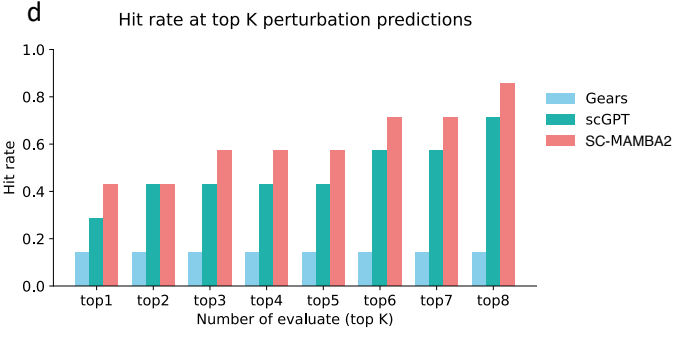
[25] Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

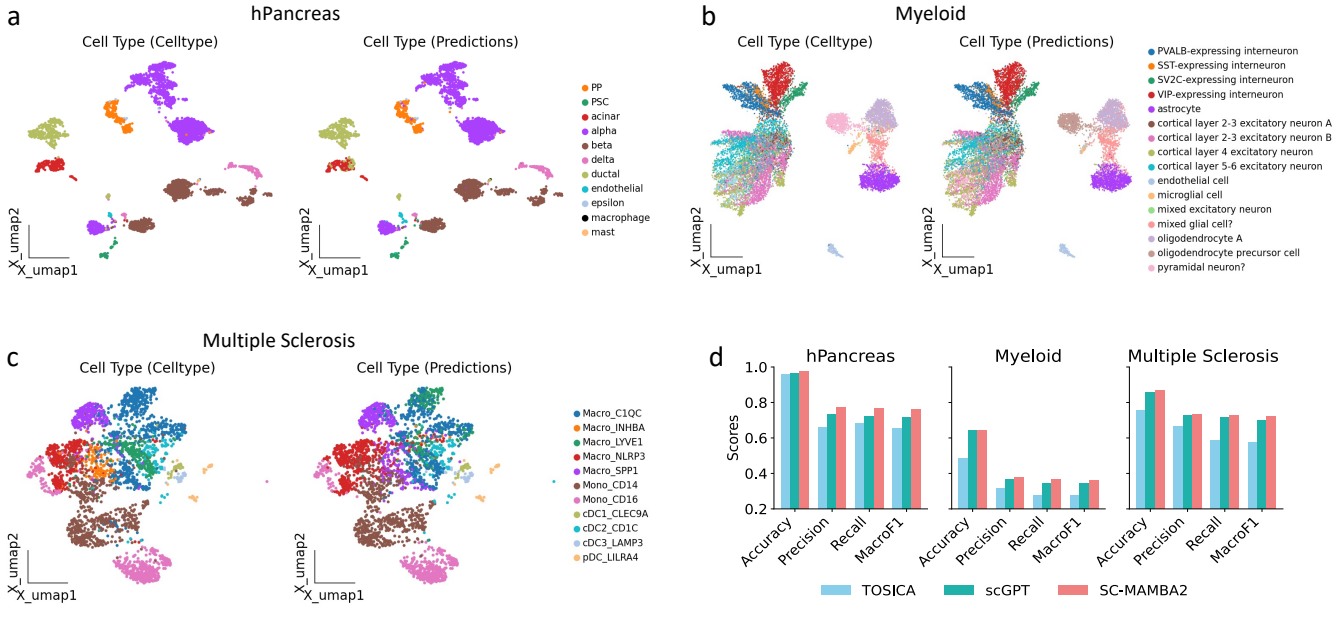

a
Performance Metrics for PBMC 10K
1.0
0.8
0.6
0.4
AvgBIO
NMIcell
ARIcell
ASWcell
Seurat v4
scVI
Harmony
scGPT
SC-MAMBA2
b
Performance Metrics for 10X Multiome PBMC
Seurat v4
scGLUE
scGPT
SC-MAMBA2
c
Performance Metrics for BMMC
Seurat v4
scGPT
SC-MAMBA2
d
celltype, avg_bio = 0.8360
batch, avg_batch = 0.9285
PBMC 10k
B cells
CD14+ Monocytes
CD4 T cells
CD8 T cells
Dendritic Cells
FCGR3A+ Monocytes
Megakaryocytes
NK cells
Other
X_umap1
X_umap2
e
celltype, avg_bio = 0.7620
batch, avg_batch = 0.9581
10x Multiome PBMC
CD14 Mono
CD16 Mono
CD4 Naive
CD4 TCM
CD4 TEM
CD8 Naive
CD8 TEM_1
CD8 TEM_2
HSPC
Intermediate B
MAIT
Memory B
NK
Naive B
Plasma
Treg
cDC
gdT
pDC
f
celltype, avg_bio = 0.7007
batch, avg_batch = 0.889
BMMC
B1 B IGKC+
B1 B IGKC-
CD14+ Mono
CD16+ Mono
CD4+ T CD314+ CD45RA+
CD4+ T activated
CD4+ T activated integrinB7+
CD4+ T naive
CD8+ T CD49f+
CD8+ T CD57+ CD45RA+
CD8+ T CD57+ CD45RO+
CD8+ T CD69+ CD45RA+
CD8+ T CD69+ CD45RO+
CD8+ T TIGIT+ CD45RA+
CD8+ T TIGIT+ CD45RO+
CD8+ T naive
CD8+ T naive CD127+ CD26- CD101-
Erythroblast
G/M prog
HSC
ILC
ILC1
Lymph prog
MAIT
MK/E prog
NK
NK CD158e1+
Naive CD20+ B IGKC+
Naive CD20+ B IGKC-
Normoblast
Plasma cell IGKC+
Plasma cell IGKC-
Plasmablast IGKC+
Plasmablast IGKC-
Proerythroblast
Reticulocyte
T prog cycling
T reg
Transitional B
cDC1
cDC2
dnT
gdT CD158b+
gdT TCRVD2+
pDC
0
1
2
3
4
5
6
7
8
9
10
11
X_umap1
X_umap2

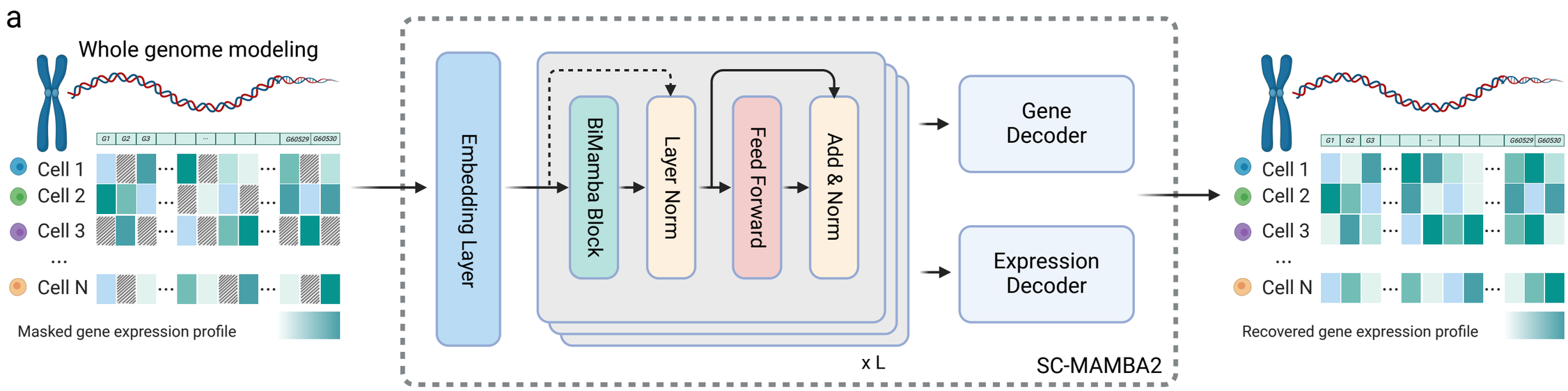

a
Average
Seen0
Seen1
Seen2
Single Unseen
Values
1.0
0.8
0.6
0.4
0.2
pearson_delta
pearson_delta_de
Gears
scFoundation
scGPT
SC-MAMBA2

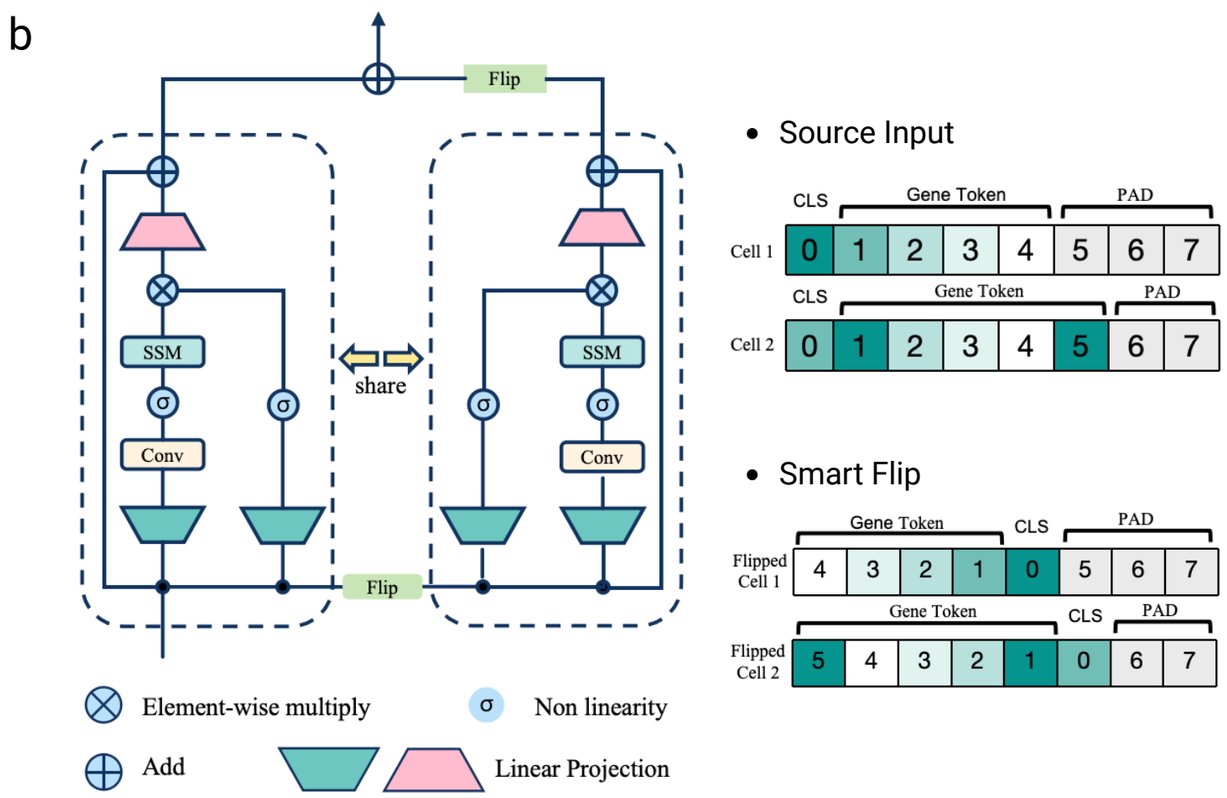

b
SAMD1+ZBTB1
Gene Expression Changes over Ctrl
4
3
2
1
0
−1
−2
HBZ
HBG2
GYPB
CYBA
PRSS57
NEAT1
MALAT1
HBA1
BLVRB
HBG1
ALAS2
MT1G
HBA2
COL18A1
SH3BGRL3
YBX1
FABP5
SET
TEX30
MYL4
scGPT
SC-MAMBA2

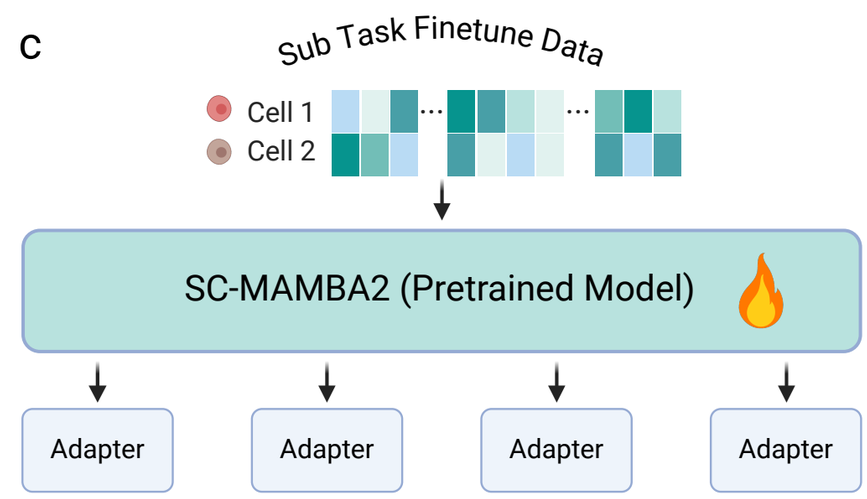

c
CEBPA
CEBPE
CNN1
FEV
FOSB
FOXA1
FOXF1
HOXB9
IGDCC3
IKZF3
KLF1
MAP2K6
MAPK1
OSR2
PRTG
PTPN9
SAMD1
TGFBR2
UBASH3B
ZBTB25
Test
Valid
Train
Unseen

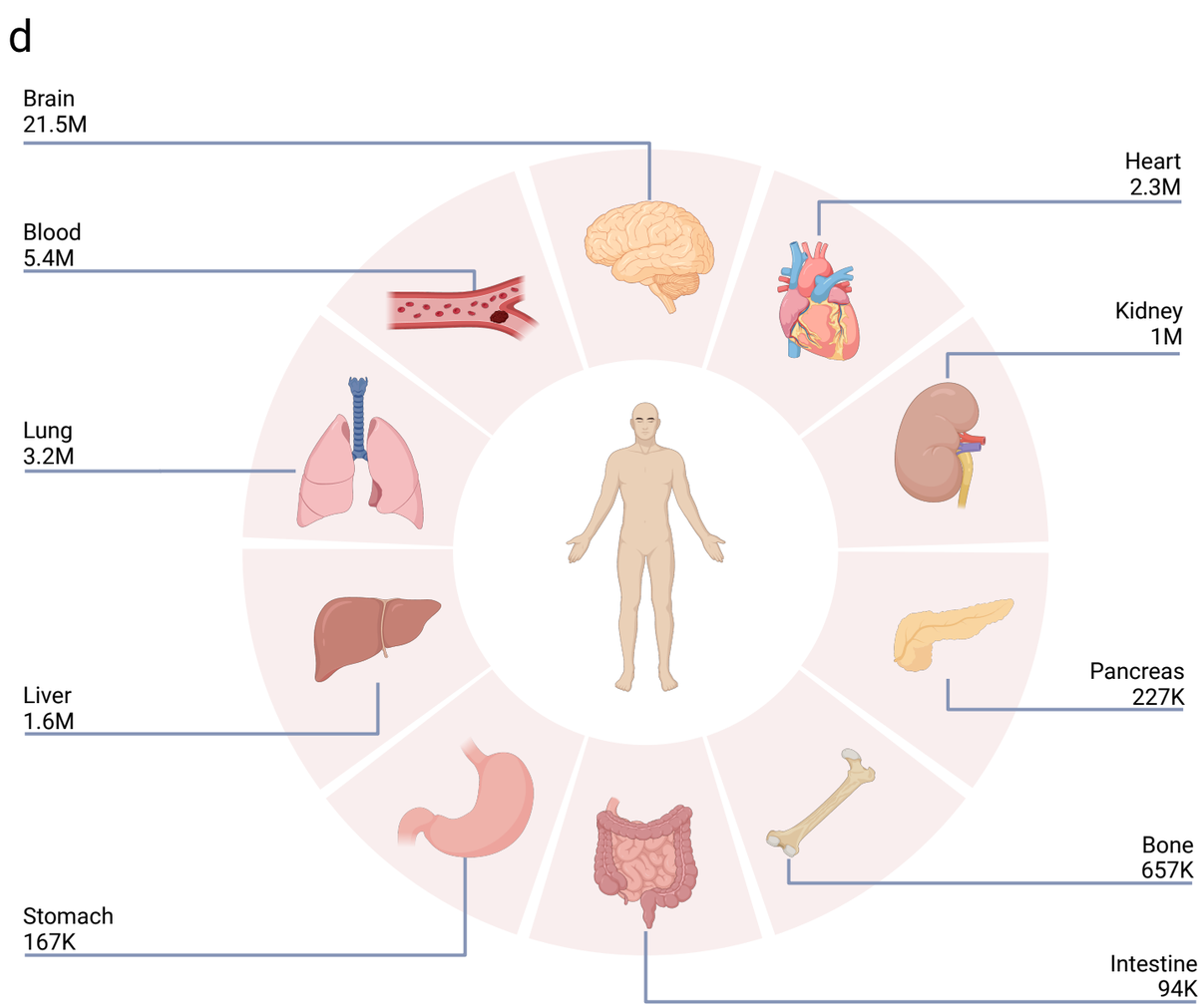

d
Hit rate at top K perturbation predictions
Hit rate
1.0
0.8
0.6
0.4
0.2
0.0
top1
top2
top3
top4
top5
top6
top7
top8
Number of evaluate (top K)
Gears
scGPT
SC-MAMBA2

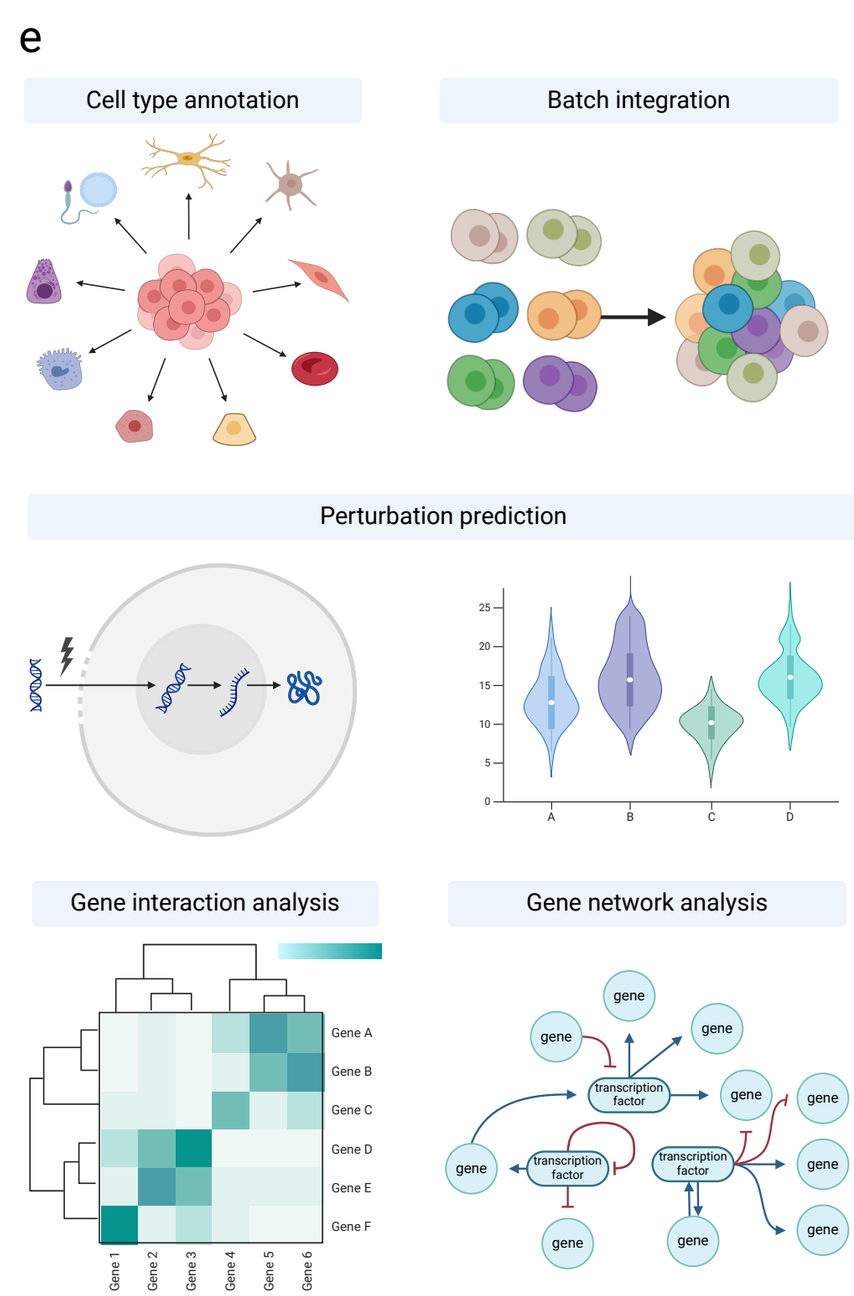

a
hPancreas
Cell Type (Celltype)
Cell Type (Predictions)
PP
PSC
acinar
alpha
beta
delta
ductal
endothelial
epsilon
macrophage
mast
X_umap2
X_umap1
b
Myeloid
Cell Type (Celltype)
Cell Type (Predictions)
PVALB-expressing interneuron
SST-expressing interneuron
SV2C-expressing interneuron
VIP-expressing interneuron
astrocyte
cortical layer 2-3 excitatory neuron A
cortical layer 2-3 excitatory neuron B
cortical layer 4 excitatory neuron
cortical layer 5-6 excitatory neuron
endothelial cell
microglial cell
mixed excitatory neuron
mixed glial cell?
oligodendrocyte A
oligodendrocyte precursor cell
pyramidal neuron?
X_umap2
X_umap1
c
Multiple Sclerosis
Cell Type (Celltype)
Cell Type (Predictions)
Macro_C1QC
Macro_INHBA
Macro_LYVE1
Macro_NLRP3
Macro_SPP1
Mono_CD14
Mono_CD16
cDC1_CLEC9A
cDC2_CD1C
cDC3_LAMP3
pDC_LILRA4
X_umap2
X_umap1
d
hPancreas
Myeloid
Multiple Sclerosis
Scores
1.0
0.8
0.6
0.4
0.2
Accuracy
Precision
Recall
MacroF1
TOSICA
scGPT
SC-MAMBA2


a
Whole genome modeling
Cell 1
Cell 2
Cell 3
Cell N
Masked gene expression profile
Embedding Layer
BiMamba Block
Layer Norm
Feed Forward
Add & Norm
x L
Gene Decoder
Expression Decoder
SC-MAMBA2
Recovered gene expression profile


b
Flip
SSM
Conv
share
Element-wise multiply
Non linearity
Add
Linear Projection
Source Input
CLS
Gene Token
PAD
Cell 1
Cell 2
Smart Flip
Flipped Cell 1
Flipped Cell 2


c
Sub Task Finetune Data
Cell 1
Cell 2
SC-MAMBA2 (Pretrained Model)
Adapter
Adapter
Adapter
Adapter


d
Brain
21.5M
Blood
5.4M
Lung
3.2M
Liver
1.6M
Stomach
167K
Heart
2.3M
Kidney
1M
Pancreas
227K
Bone
657K
Intestine
94K


e
Cell type annotation
Batch integration
Perturbation prediction
Gene interaction analysis
Gene network analysis
Gene A
Gene B
Gene C
Gene D
Gene E
Gene F
Gene 1
Gene 2
Gene 3
Gene 4
Gene 5
Gene 6
gene
transcription factor