scLong: A Billion-Parameter Foundation Model for Capturing Long-Range Gene Context in Single-Cell Transcriptomics

Ding Bai[1, *], Shentong Mo[1, *], Ruiyi Zhang[2, *], Yingtao Luo[3], Jiahao Gao[4], Jeremy Parker Yang[5], Qiuyang Wu[6], Digvijay Singh[7], Hamidreza Rahmani[8], Tiffany Amariuta[4, 9], Danielle Grotjahn[8], Sheng Zhong[6], Nathan Lewis[10, 6], Wei Wang[5, 11], Trey Ideker[4, 6], Pengtao Xie[2, 1, 4] ✉, and Eric Xing[1, 3] ✉

[1]Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, UAE
[2]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA
[3]Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[4]Department of Medicine, University of California San Diego, La Jolla, CA 92093, USA
[5]Department of Chemistry and Biochemistry, University of California San Diego, La Jolla, CA 92093, USA
[6]Department of Bioengineering, University of California San Diego, La Jolla, CA 92093, USA
[7]School of Biological Sciences, University of California San Diego, La Jolla, CA 92093, USA
[8]Department of Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, CA 92037, USA
[9]Halıcıoğlu Data Science Institute, University of California San Diego, La Jolla, CA 92093, USA
[10]Department of Pediatrics, School of Medicine, University of California San Diego, La Jolla, CA 92093, USA
[11]Department of Cellular and Molecular Medicine, School of Medicine, University of California San Diego, La Jolla, CA 92093, USA

Single-cell RNA sequencing (scRNA-seq) has revolutionized the study of cellular heterogeneity by providing gene expression data at single-cell resolution, uncovering insights into rare cell populations, cell-cell interactions, and gene regulation. Foundation models pretrained on large-scale scRNA-seq datasets have shown great promise in analyzing such data, but existing approaches are often limited to modeling a small subset of highly expressed genes and lack the integration of external gene-specific knowledge. To address these limitations, we present scLong, a billion-parameter foundation model pretrained on 48 million cells. scLong performs self-attention across the entire set of 28,000 genes in the human genome. This enables the model to capture long-range dependencies between all genes, including lowly expressed ones, which often play critical roles in cellular processes but are typically excluded by existing foundation models. Additionally, scLong integrates gene knowledge from the Gene Ontology using a graph convolutional network, enriching its contextual understanding of gene functions and relationships. In extensive evaluations, scLong surpasses both state-of-the-art scRNA-seq foundation models and task-specific models across diverse tasks, including predicting transcriptional responses to genetic and chemical perturbations, forecasting cancer drug responses, and inferring gene regulatory networks.

Correspondence: ***epxing@cs.cmu.edu, p1xie@ucsd.edu***
**** Equal contribution***

Introduction

Single-cell transcriptomics enables the study of gene expression at the individual cell level, offering insights into cellular heterogeneity that bulk methods cannot reveal (1–3). It allows for the identification of rare cell populations (4), uncovers cell-cell interactions (5), and provides a detailed map of gene regulation (6), making it an essential tool for advancing personalized medicine, drug discovery, and understanding cellular diversity. Foundation models have shown great promise in analyzing single-cell transcriptomics data (7–11). Pretrained on large-scale single-cell RNA sequencing (scRNA-seq) datasets using self-supervised learning (12), these models can capture complex gene expression patterns across diverse cell types. One of the key mechanisms in these models is self-attention (13), which computes relationships between genes by allowing every gene to attend to every other gene. This helps the model capture important gene interactions, contextualize gene expression, and understand long-range dependencies between genes. With fine-tuning, foundation models can be adapted for various downstream tasks, such as cell type classification (7), gene perturbation prediction (14), and reconstruction of gene regulatory networks (15), even in settings with limited data.

Despite the significant progress foundation models have made in analyzing single-cell transcriptomics data, they still face critical limitations that hinder their ability to fully capture the complexity of gene expression. One key limitation is that, to save computational cost, these models typically perform self-attention on a small subset of genes (e.g., 2048 in Geneformer (8) and scFoundation (10), and 2000 in scGPT (9)), often selected based on high expression levels (8). This approach excludes many lowly expressed genes which play essential roles in cellular processes and regulatory networks (16–18). By restricting self-attention to only a fraction of the transcriptome, current models miss important regulatory signals and fail to capture long-range gene interactions across the entire genome (19, 20). As a result, they provide incomplete representations of gene regulatory mechanisms, overlooking subtle but critical gene interactions that are key to understanding complex cellular functions. Another limitation is the lack of integration of external gene-specific knowledge, such as that provided by the Gene Ontology (21), which encodes relationships among genes, biological processes, and molecular functions. Current models (8–10) rely predominantly on patterns derived solely from gene expression data, which restricts their ability to capture context related to gene functions and regulatory interactions. Without leveraging such rich functional information, these models may struggle to fully understand the roles of genes, especially in cases where direct expression data offers limited insight into a gene's activity within a broader regulatory framework.

To overcome these limitations, we present scLong, a billion-parameter scRNA-seq foundation model. First, instead of focusing on a small subset of genes, scLong performs self-attention across the entire human transcriptome, encompassing around 28,000 genes. This enables the model to capture long-range interactions and dependencies between all genes, including those with low expression levels that may still play crucial roles in cellular processes. By including every gene in the analysis, scLong offers a more comprehensive and unbiased representation of gene regulatory networks, avoiding the pitfalls of restricting attention to highly expressed genes. Second, scLong integrates external gene knowledge from the Gene Ontology using a graph convolutional network (14, 22) to learn gene representations. This allows the model to incorporate hierarchical and functional relationships between genes, providing deeper functional context to its predictions. By leveraging this structured information, scLong enhances its ability to interpret gene functions and interactions, even when direct expression data is sparse or ambiguous. Together, these two mechanisms - self-attention across all genes and integration of Gene Ontology knowledge - enable scLong to generate more accurate, interpretable, and functionally relevant representations, effectively addressing the limitations of current foundation models in transcriptomics data analysis. scLong has one billion parameters, making it ten times larger than the previously largest scRNA-seq foundation models, including scFoundation (10) and GeneCompass (11), which each have 100 million parameters.

Results

scLong overview. scLong takes a cell's gene expression vector as input, generating a representation for each element in the vector (Fig. 1a). Each element corresponds to a specific gene, with its value indicating the level of gene transcription into RNA at a given moment, which may reflect potential protein production. scLong includes a gene encoder, an expression encoder, and a contextual encoder. The expression encoder, a multi-layer perceptron (MLP), produces a representation vector for each scalar expression value. The gene encoder leverages Gene Ontology (21) to extract a representation vector for each gene. For each element in the expression vector - defined by a gene ID and its expression value - we combine the gene's representation (from the gene encoder) with its expression representation (from the expression encoder) to represent the element. These element representations are then fed into the contextual encoder, which learns contextualized representations that capture relationships among elements (Methods).

The gene encoder constructs a gene graph using the Gene Ontology and applies a graph convolutional network (14, 22) to this graph to learn gene representations. The Gene Ontology (GO) (21) offers a structured vocabulary for describing gene functions, organized into three primary domains: Biological Process, which refers to the biological roles or processes in which a gene is involved, such as cell division or metabolic pathways; Molecular Function, which specifies the biochemical activities of a gene product, such as enzyme activity or binding; and Cellular Component, indicating the cellular locations where a gene product operates, such as the nucleus or mitochondria. Each gene's functions are annotated with GO terms from this vocabulary. The gene graph is constructed based on the method in (14), where each node represents a gene. For each pair of genes, u and v, the Jaccard index is calculated to measure the overlap between their sets of annotated GO terms. If the overlap is sufficiently high, an edge is added between the two genes in the graph. The gene graph captures functional relationships between genes based on shared GO annotations. Genes with overlapping GO terms are connected, reflecting similarities in biological processes, molecular functions, and cellular localization. For example, genes involved in related biological processes, such as metabolic pathways, are linked, suggesting shared roles in complex cellular functions. Genes with similar molecular functions, like enzymatic activities or binding properties, are also connected, indicating biochemical similarities or cooperative interactions. Additionally, genes localized to the same cellular components, such as the nucleus or mitochondria, are linked, suggesting potential spatial co-localization. On top of the gene graph, we construct a graph convolutional network (GCN) (23), which learns representations for each gene. Through a process called message passing, the GCN enables each node to aggregate information from its neighboring nodes, effectively capturing the relationships between genes.

The contextual encoder employs self-attention (13) to capture long-range relationships between genes in the context of the input cell. It takes the initial representations generated by the gene and expression encoders and learns a contextualized representation for each element. Self-attention calculates pairwise correlations among elements, capturing their interdependencies. To balance computational efficiency with representation quality, we use a large Performer (24) encoder and a mini Performer encoder to process elements with varying expression levels. Specifically, we rank each cell's gene expression elements in descending order, dividing them into two groups: a high-expression group, containing the top-ranked elements, and a low-expression group with the remaining ones. The high-expression group, which carries core biological information critical for modeling gene interactions and regulatory pathways, is processed by the larger Performer encoder with more layers and parameters. The low-expression group, offering less critical information, is processed by the smaller encoder, optimizing computational efficiency.

While low-expression genes are less prominent in terms of overall abundance, they play essential roles in a range of biological processes and cannot be disregarded. Many low-expression genes are involved in regulatory mechanisms that influence the behavior of high-expression genes, acting as switches or modulators in complex cellular networks (16).

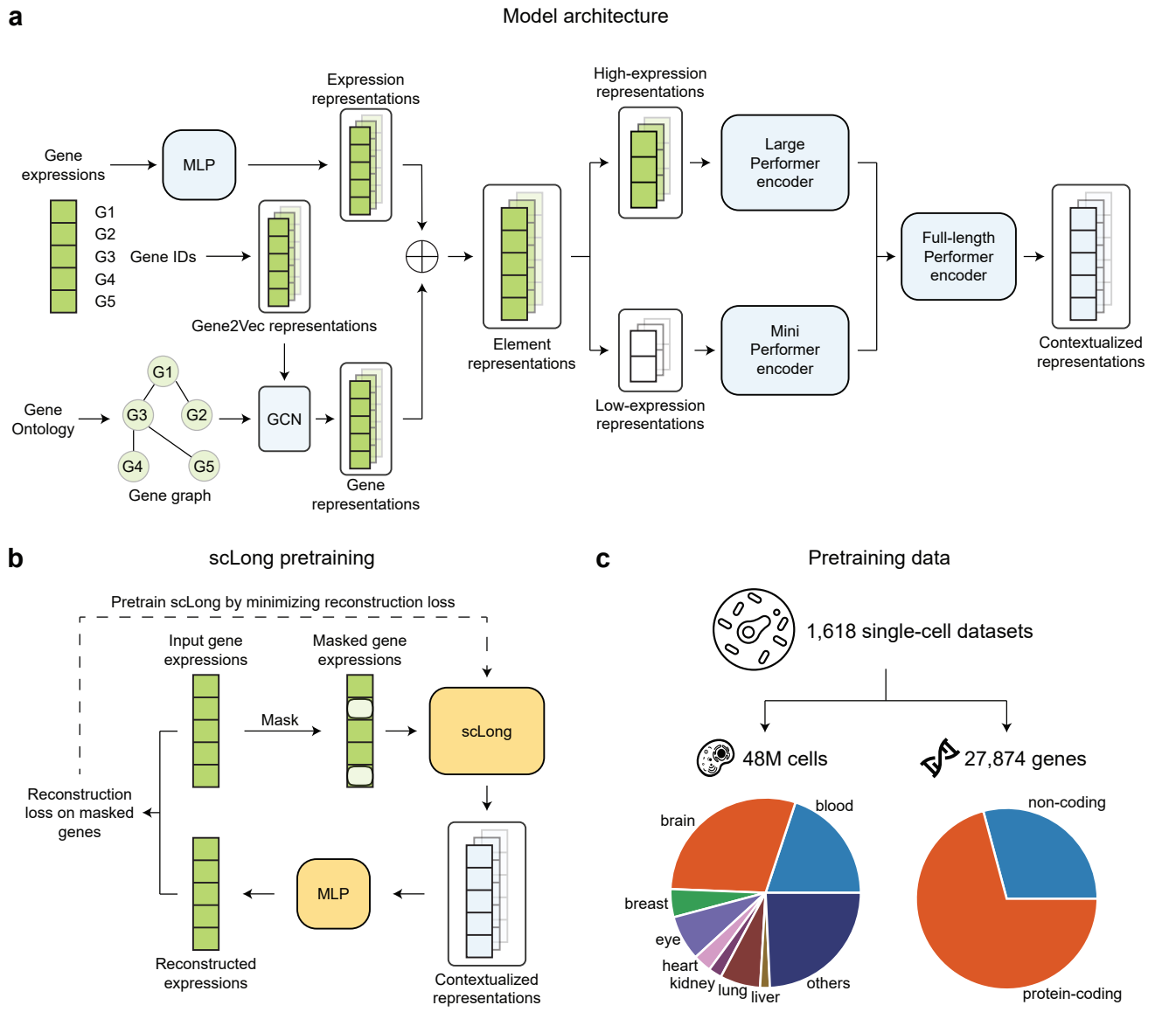


Fig. 1 | scLong, a scRNA-seq foundation model with one billion parameters pretrained on 48 million cells, captures long-range context across 27,874 genes by employing a dual encoder architecture and leveraging Gene Ontology knowledge. a, Model architecture of scLong. scLong generates a representation for each element in a cell's gene expression vector using three main components: a gene encoder, an expression encoder, and a contextual encoder. The expression encoder, a multi-layer perceptron (MLP), produces a representation vector for each scalar expression value, while the gene encoder utilizes Gene Ontology to derive a representation vector for each gene. These representations are combined for each element and fed into the contextual encoder, which learns context-aware representations that capture inter-element relationships. Specifically, the gene encoder constructs a gene graph from Gene Ontology and applies a graph convolutional network (GCN) to learn gene-specific representations. To capture long-range relationships between genes, the contextual encoder leverages self-attention. To optimize efficiency and representation quality, scLong employs two Performers of different sizes, with high-expression elements processed by a larger Performer for detailed interaction modeling, and low-expression elements by a smaller Performer for efficiency. The outputs from these two encoders are then passed through a final full-length Performer, generating the final scLong representations. **b**, scLong is pretrained by reconstructing masked expression values. For each input cell, we randomly mask a subset of expression values and use scLong to learn representations for both the masked and unmasked elements. The representations of the masked elements are passed to an MLP-based decoder to predict their expression values. A reconstruction loss is calculated between the predicted and actual values, and pretraining involves minimizing this reconstruction loss. **c**, The pretraining data for scLong includes 48 million cells and 27,874 genes (approximately 20,000 protein-coding and 8,000 non-coding genes) derived from 1,618 scRNA-seq datasets spanning over 50 tissues.

These genes can also be crucial in rare or specialized cell types, where their subtle expression may drive specific phenotypes or responses to environmental stimuli (17, 18). Ignoring them could lead to incomplete models that overlook important aspects of cellular function. Moreover, low-expression genes often participate in context-specific pathways that become active only under certain conditions, such as stress responses, immune signaling, or disease progression (17, 18). These genes may also be important for rare cell populations, whose contributions to tissue function

or disease states could be missed if low-expression signals are not adequately represented (19, 20). Thus, while high-expression genes often drive primary biological processes, low-expression genes provide the fine-tuned regulation and specialized functions necessary for a complete understanding of cellular behavior.

After processing by the large and mini Performer encoders, each element obtains a contextualized representation vector of uniform dimension. These vectors are then input into a full-length Performer encoder, which performs self-attention across all elements. This final encoder, configured with the same number of layers as the mini encoder, produces the final representations for each expression element.

To pretrain scLong, we compiled a large-scale scRNA-seq dataset comprising approximately 48 million human cells from diverse tissues and cell types (Fig. 1c), covering 27,874 human genes (Methods). Pretraining involves reconstructing masked expression values (12) (Fig. 1b). For each input cell, we randomly mask a subset of values, then use scLong to learn representations for both masked and unmasked elements. The representations of masked elements are fed into a decoder to predict their expression values. A reconstruction loss is calculated between the predicted and actual values. Pretraining is performed by minimizing these reconstruction losses (Methods).

scLong predicts transcriptional outcomes of genetic perturbations. Predicting transcriptional outcomes of genetic perturbations involves forecasting how changes to specific genes, such as knockouts or overexpressions, impact the overall gene expression profile of a cell (25, 26). This capability is essential for understanding gene function and regulatory networks, as each perturbation can reveal how genes interact with each other and contribute to cellular behavior. Accurately predicting transcriptional outcomes offers a deeper understanding of pathways associated with disease, helping to pinpoint potential therapeutic targets and advance precision medicine. Additionally, in synthetic biology, understanding transcriptional responses supports the design of gene circuits and engineered cells with specific desired properties.

For this task, the input comprises a cell's pre-perturbation gene expression vector and its corresponding perturbation conditions, while the output is the cell's post-perturbation gene expression vector. We utilized scLong to generate representations for pre-perturbation gene expressions and employed GEARS (14) to derive representations for the perturbation conditions (Fig. 2a). These representations were summed and processed by a GEARS decoder to predict the post-perturbation gene expression vector (Methods). The perturbation conditions included both single and double gene perturbations, where either one or two genes were altered simultaneously in each cell. The Norman dataset (26), consisting of 91,205 cell samples, 5,045 genes, and 236 unique perturbation conditions, was used for this task, providing a training set of 58,134 cells, a validation set of 6,792 cells, and a test set of 26,279 cells. Each test sample was categorized into one of four scenarios: 1) neither gene in a double-gene perturbation conducted on the test sample is present in the training data (Seen 0/2); 2) one gene in a double-gene perturbation is absent from the training data (Seen 1/2); 3) both genes in a double-gene perturbation are present in the training data (Seen 2/2); and 4) the gene in a single-gene perturbation is absent from the training data (Seen 0/1). This categorization helps assess the model's ability to generalize to unseen perturbations. Following GEARS, prediction performance was evaluated using two metrics: Pearson correlation and mean squared error (MSE) on the top 20 differentially expressed (DE) genes (14) (Methods). We compared scLong with three state-of-the-art scRNA-seq foundation models: Geneformer (8), scGPT (9), and scFoundation (10), as well as GEARS (14), a task-specific approach developed to predict gene expression outcomes following genetic perturbations. Geneformer, pretrained on 29.9 million cells, has 30 million parameters. scFoundation, with 100 million parameters, was pretrained on 50 million cells, while scGPT, with 50 million parameters, was pretrained on 33 million cells. Unlike the others, GEARS does not involve pretraining. Geneformer, scGPT, and scFoundation integrate GEARS with their pretrained models to predict perturbational effects (Methods).

scLong outperforms the four baseline models in most cases, across both Pearson correlation and MSE metrics, and under various test scenarios, including Seen 0/2, Seen 1/2, Seen 2/2, and Seen 0/1 (Fig. 2b). The improvement of scLong is particularly notable in the Seen 0/1 and Seen 0/2 scenarios, where the perturbation conditions in the test data are not encountered during training. For example, in the Seen 0/1 scenario, scLong achieved a Pearson correlation of 0.63, compared to 0.56, 0.57, 0.58, and 0.58 for GEARS, Geneformer, scGPT, and scFoundation, respectively. In the Seen 0/2 scenario, scLong obtained an MSE of 0.17, while the baseline models recorded errors of 0.22, 0.19, 0.20, and 0.19, respectively. This demonstrates that scLong has a stronger out-of-domain generalization capability compared to the baseline models.

scLong's superior performance over existing foundation models, including Geneformer, scGPT, and scFoundation, can be attributed to its two key advantages: comprehensive self-attention across all genes and the integration of Gene Ontology knowledge. First, scLong's self-attention spans all 28,000 genes, capturing interactions among both highly and lowly expressed genes, unlike baseline models that restrict attention to a small subset of highly expressed genes. Although low-expression genes are often less abundant, they play essential roles in gene regulation and cellular signaling, acting as modulators that influence how high-expression genes respond to perturbations (16–18). By attending to all genes, scLong identifies a more complete picture of regulatory dynamics, capturing subtle but important gene

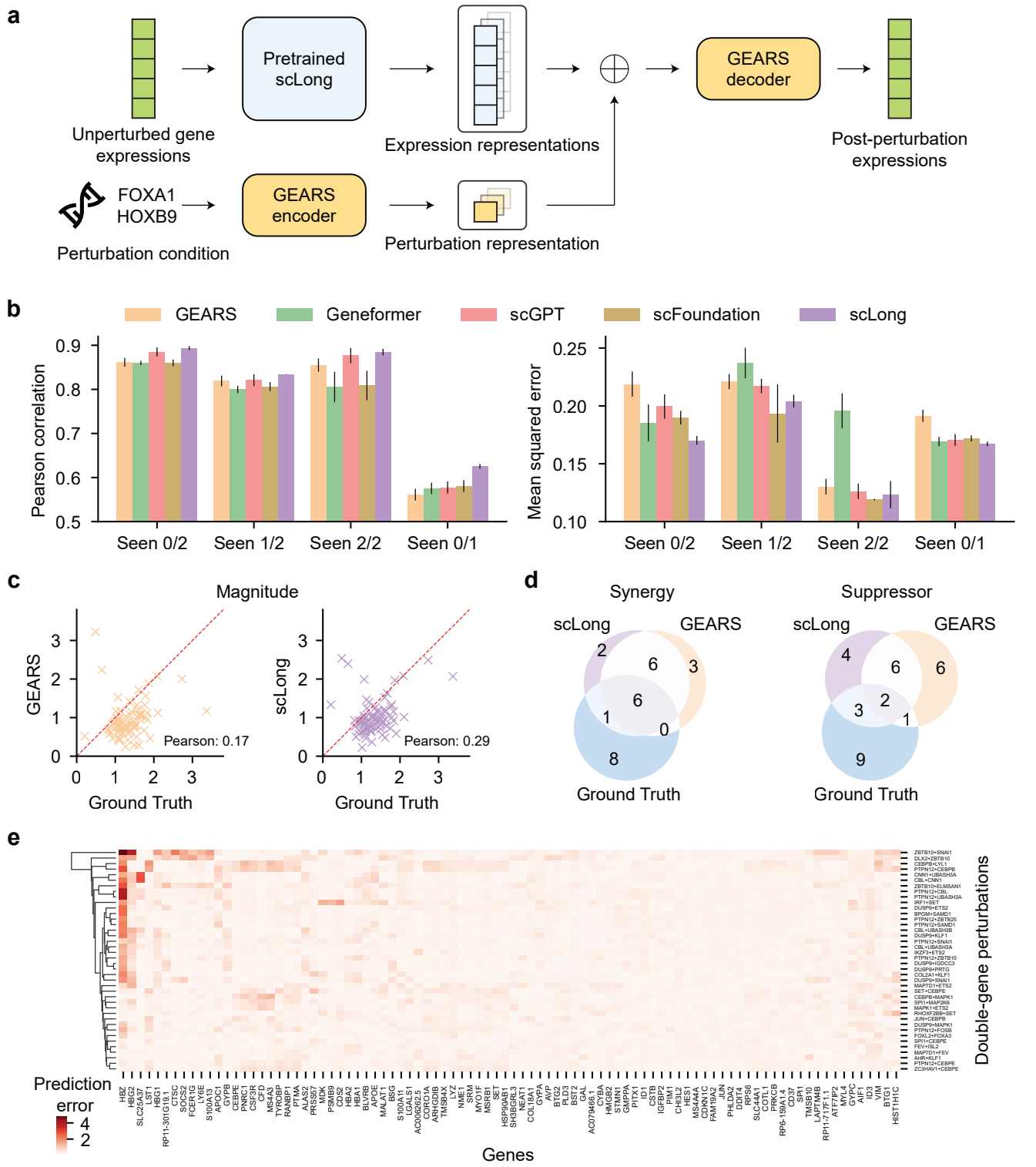


Fig. 2 | scLong surpasses state-of-the-art scRNA-seq foundation models and task-specific methods in predicting transcriptional outcomes of genetic perturbations. a, Model architecture for fine-tuning the pretrained scLong to predict transcriptional outcomes of genetic perturbations. **b**, scLong outperforms scRNA-seq foundation models, including Geneformer, scGPT, and scFoundation, as well as the task-specific GEARS method, in terms of Pearson correlation (higher is better) and mean squared error (lower is better) on the top 20 differentially expressed genes across four testing scenarios: Seen 0/2, Seen 1/2, Seen 2/2, and Seen 0/1. **c**, In classifying double-gene perturbations into two genetic interaction types, synergy and suppressor, scLong's magnitude score achieves a significantly higher Pearson correlation with the ground truth than GEARS, underscoring its enhanced capability to distinguish between these interaction types. Each cross denotes a double-gene perturbation. **d**, The top-15 synergistic double-gene perturbations identified by scLong show a greater overlap with the ground truth compared to GEARS, and the same holds for suppressor double-gene perturbations. This further demonstrates that scLong provides more accurate predictions of synergistic and suppressive interactions in double-gene perturbations. **e**, scLong's mean absolute prediction errors for individual genes (columns) across different double-gene perturbation conditions (rows). The 90 genes and 40 conditions with the largest errors were visualized. Hierarchical clustering of error patterns (row vectors) effectively groups perturbation conditions involving the same gene together.

interactions crucial for accurately predicting the transcriptional effects of genetic perturbations. This broad gene attention allows scLong to account for dependencies and feedback mechanisms that baseline models may overlook due to their limited gene focus. Second, scLong incorporates Gene Ontology (GO) knowledge through a graph convolutional network, providing each gene with a representation enriched by its biological functions, processes, and cellular roles. Gene Ontology offers structured, hierarchical insights that enable the model to understand not only direct gene interactions but also the broader functional context of each gene's role within the cellular environment (21, 26, 27). In the context of genetic perturbations, such functional insights are vital, as they allow the model to infer how perturbing one gene might affect other related genes within the same biological pathways or processes. Baseline models that lack GO knowledge miss this critical functional layer. Together, scLong's inclusive self-attention and GO-enhanced representations equip it to generate highly context-aware predictions, leading to its stronger performance in predicting transcriptional responses to genetic perturbations.

scLong's enhanced performance over the task-specific model GEARS is largely due to its extensive pretraining on 48 million scRNA-seq data - a foundational step that GEARS lacks. This large-scale pretraining enables scLong to learn generalizable patterns in gene expression across diverse cell types and conditions, equipping it with a comprehensive understanding of cellular behaviors, gene interactions, and regulatory networks. For the task of predicting transcriptional responses to genetic perturbations, these pretraining benefits are crucial. Genetic perturbations often result in complex regulatory cascades and cross-gene effects that are not fully represented within narrow task-specific datasets. Through exposure to tens of millions of expression profiles, scLong learns robust gene representations that capture both common and rare expression patterns, including context-specific dependencies likely to be triggered by perturbations. This pretraining allows scLong to identify subtle transcriptional shifts and regulatory changes that may be pivotal in accurately predicting perturbation outcomes. In contrast, GEARS, lacking this extensive pretraining, relies solely on task-specific data, which limits its exposure to the broad spectrum of cellular states and gene regulatory mechanisms that scLong acquires through pretraining. Consequently, GEARS may struggle to capture nuanced gene interactions or transcriptional changes, particularly in response to less common or complex perturbations. Additionally, scLong's pretraining fosters the formation of robust gene representations, reducing overfitting to specific datasets and enhancing its predictive performance across diverse perturbation scenarios. This foundational knowledge enables scLong to make more accurate, context-aware predictions of transcriptional outcomes, leading to its stronger performance relative to GEARS in the task of genetic perturbation response prediction.

We evaluated scLong's capability to classify double-gene perturbations into two genetic interaction (GI) types: synergy and suppressor. Following (26), we used a magnitude score to distinguish these interaction types. This score measures the correlation between the effect of a double-gene perturbation (i, j) and the linear combination of the effects of the corresponding single-gene perturbations i and j (Methods). Higher scores indicate stronger synergy between i and j. For each double-gene perturbation, we calculated magnitude scores using ground-truth post-perturbation expressions, scLong-predicted post-perturbation expressions, and GEARS-predicted post-perturbation expressions. We then computed Pearson correlation coefficients (PCC) between scLong and the ground truth, as well as between GEARS and the ground truth. scLong achieved a higher PCC than GEARS (Fig. 2c), demonstrating its superior ability to identify the true GI type. To further illustrate this, we ranked the magnitude scores of ground truth, scLong, and GEARS in descending order. The top 15 and bottom 15 double-gene perturbations were classified as having synergy and suppressor GI types, respectively. For both interaction types, the overlap between scLong and the ground truth exceeded that of GEARS and the ground truth (Fig. 2d), further demonstrating that scLong more accurately predicts synergistic and suppressive gene interactions in double-gene perturbations.

Fig. 2e shows scLong's mean absolute prediction errors for individual genes (columns) across various perturbation conditions (rows). We visualized 90 genes and 40 perturbation conditions with the highest prediction errors. Hierarchical clustering of perturbation conditions was performed, grouping those with similar error patterns (row vectors). The clustering results are sensible: conditions involving the same gene, such as CEBPB+LYL1 and PTPN12+CEBPB, or CNN1+UBASH3A and CBL+CNN1, were grouped together. Under conditions involving CEBPE (e.g., ZC3HAV1+CEBPE, PTPN12+CEBPE), scLong achieved near-zero errors across all genes. This is because CEBPE has minimal regulatory influence on other genes; perturbing CEBPE does not markedly affect gene expression, which simplifies the prediction of transcriptional changes. In contrast, conditions involving ZBTB10, SNAI1, or DLX2 exhibit notably higher prediction errors, as these genes exert substantial regulatory influence on others. Perturbing them triggers significant transcriptomic shifts, posing a greater challenge for accurate prediction. Generally, prediction errors are low across most genes; however, errors for the HBZ gene are particularly high due to its sensitivity to regulatory effects from other genes. Perturbing these regulators substantially alters HBZ expression, making its post-perturbation state more challenging to predict.

scLong predicts transcriptional outcomes of chemical perturbations. Beyond predicting genetic perturbations, we applied scLong to predict gene expression profiles in response to de novo chemical perturbations, which is crucial for drug discovery and personalized medicine (28). By

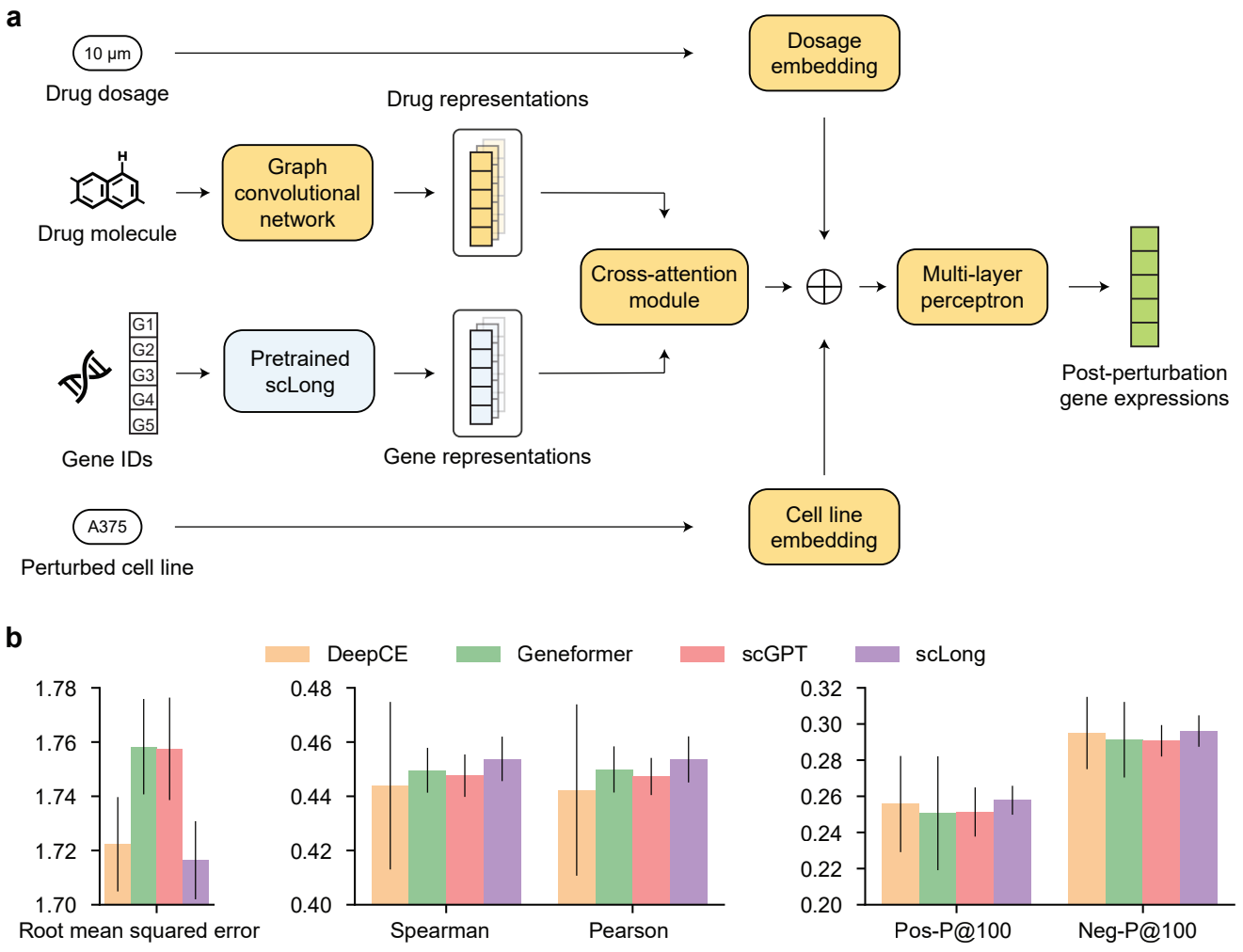


Fig. 3 | scLong outperforms existing scRNA-seq foundation models and specialized methods in predicting transcriptional outcomes of chemical perturbations. **a**, Model architecture for fine-tuning the pretrained scLong for this prediction task. **b**, scLong demonstrates superior results over scRNA-seq foundation models, including Geneformer and scGPT, as well as the task-specific DeepCE method, across metrics including root mean squared error (RMSE), Spearman and Pearson correlations, Pos-P@100 and Neg-P@100. Higher values indicate better performance for all metrics except RMSE.

forecasting how novel compounds affect gene activity, researchers can rapidly screen for potential therapeutic effects or adverse reactions, significantly accelerating the drug development process. This capability also provides insights into the molecular pathways and cellular processes targeted by new compounds, helping to uncover their specific mechanisms of action. Additionally, it reduces the need for extensive experimental validation, saving time and resources, while enabling more precise, data-driven decisions in both clinical and research settings.

In this task, we used a subset of the L1000 dataset (29), which contains 7 distinct cell lines, 978 genes, and 810 drug compounds, with drugs tested at 6 different dosage levels. The prediction model takes two inputs: 1) the index of the perturbed cell line and 2) the molecular graph and dosage of the drug used to perturb it. The output is the gene expression profile of the cell line after perturbation. The dataset does not include pre-perturbation gene expression data. Each data sample in L1000 consists of these inputs and outputs, totaling 5,005 examples, with 3,965 used for training, 544 for validation, and 496 for testing. We used scLong to extract representation vectors for each gene and a graph convolutional network (GCN) to extract representations from the drug molecule graph (Fig. 3a). These representations are passed through a multi-head cross-attention module (13), combined with embeddings of cell line indices and dosage information, and then fed into an MLP to predict post-perturbation gene expression (Methods). We compared scLong with two foundation models, Geneformer and scGPT, as well as the task-specific model DeepCE (30). Evaluation metrics included root mean square error (RMSE), Spearman and Pearson correlation scores, and top-100 precision for the highest (Pos-P@100) and lowest (Neg-P@100) predicted expression values (Methods). For RMSE, lower values indicate better performance, while higher values are better for the other metrics.

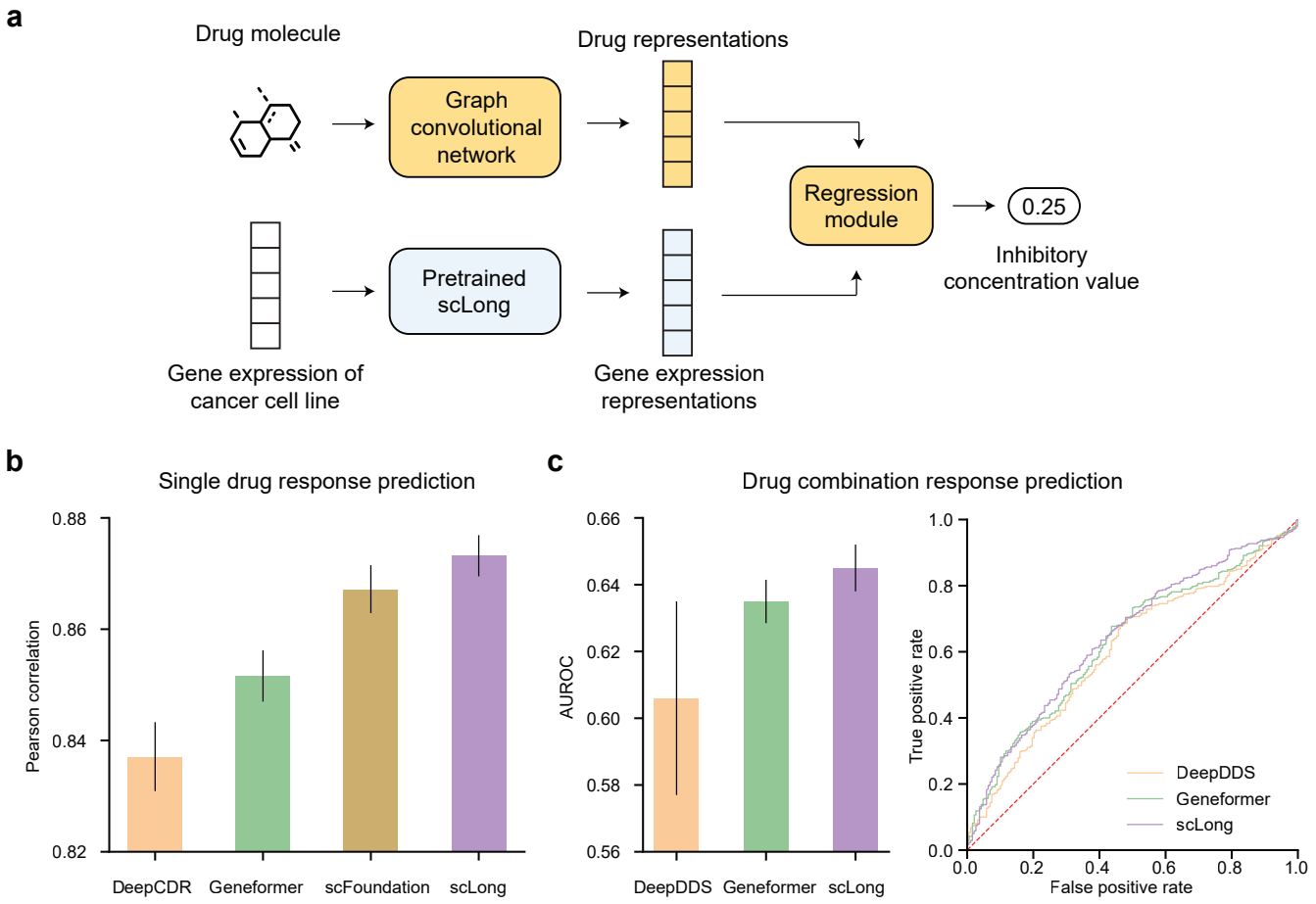


Fig. 4 | scLong surpasses existing scRNA-seq foundation models and task-specific methods in predicting cancer cell responses to individual drugs and synergistic drug combinations. a, Model architecture for fine-tuning the pretrained scLong for this prediction task. **b**, scLong achieves higher Pearson correlation and area under the receiver-operating characteristic curve (AUROC) than foundation models including Geneformer and scFoundation, as well as specialized approaches including DeepCDR and DeepDDS.

scLong outperformed the Geneformer and scGPT foundation models, as well as the task-specific model DeepCE, across all evaluation metrics (Fig. 4a). The reasons for this performance advantage align with those observed in predicting transcriptional outcomes of genetic perturbations. Specifically, scLong's comprehensive self-attention across all genes and integration of gene-specific knowledge from the Gene Ontology contributed to its advantage over Geneformer and scGPT. Additionally, scLong outperformed DeepCE due to its extensive pretraining on 48 million cells.

scLong predicts cancer drug response. Cancer drug response prediction involves forecasting how individual cancer cells or tumors will react to specific treatments (31). This process is essential because cancer is a highly heterogeneous disease, and not all patients respond to the same drugs in the same way. By accurately predicting drug response, personalized treatment plans can be developed, improving the effectiveness of therapies and minimizing adverse effects. It enables oncologists to tailor treatment strategies based on the molecular profile of a patient's cancer, leading to better outcomes. Additionally, it accelerates drug discovery by identifying promising drug candidates and reducing the need for extensive clinical trials.

In this task, the input includes the molecular structure of a potential cancer drug and the bulk gene expression profile of a cancer cell line. The output is a prediction of the drug's efficacy against the cancer cell line, measured by its half-maximal inhibitory concentration (IC50) value (32). We use scLong to extract a representation vector from the input gene expression data, which is then concatenated with the drug molecule representation obtained through a graph convolutional network (33) (Fig. 4a). The combined representation is subsequently fed into a regression module to predict the IC50 value (Methods). We used the dataset from DeepCDR (34), which includes 102,074 training examples and 5,372 testing examples. We compared scLong with Geneformer and a task-specific model, DeepCDR. Pearson correlation was used as the evaluation metric, where higher values indicate better performance. scLong outperformed both baselines, with a Pearson correlation score of 0.873, surpassing Geneformer's score of 0.852 and DeepCDR's 0.837 (Fig. 4b).

We further explored whether scLong improves the prediction

of cancer cell responses to synergistic drug combinations, focusing on the response to drug pairs rather than individual drugs (35). Drug combinations can target multiple pathways or mechanisms simultaneously, potentially leading to better therapeutic outcomes than single-drug treatments. They can also decrease the likelihood of drug resistance developing, as it is harder for cancer cells to adapt to multiple pharmacological agents at once. Despite its promise in cancer therapy, the exponential increase in potential drug pairings poses a significant challenge in identifying the most effective combinations. For this task, the input consists of a cancer cell line and a drug pair, while the output is a binary label indicating whether the cell responds. The model architecture closely resembles that used for single-drug response prediction (Methods). We used a large-scale oncology screening dataset (36) with over 12,000 examples for training and a separate dataset from AstraZeneca's drug combination dataset (37) with 668 samples for testing. The test dataset has a different distribution from the training data, allowing us to assess the models' out-of-distribution generalization capabilities. We compared scLong with Geneformer and a task-specific model DeepDDS (38). scLong outperformed both baselines in terms of the area under the receiver operating characteristic curve (AUROC) (Fig. 4c).

scLong's superior performance over the Geneformer and scGPT foundation models is again attributed to its comprehensive self-attention across both highly and lowly expressed genes and its integration of Gene Ontology knowledge. First, low-expression genes, while less abundant, often act as modulators within cellular pathways, serving as "switches" or fine-tuners in signaling and gene regulation that indirectly influence how high-expression genes respond to drug interventions (17, 18, 20). In cancer cells, these subtle regulatory roles are particularly significant, as low-expression genes can control key pathways linked to drug resistance, cell survival, or proliferation (16, 18–20). By attending to low-expression genes, scLong captures a more complete view of the cellular network, identifying intricate dependencies and regulatory feedback loops that are essential for accurately predicting cancer cell responses to drug interventions. Second, incorporating functional relationships from Gene Ontology enriches scLong's predictions by embedding structured knowledge about gene functions and pathways. Since drugs often target or disrupt specific cellular processes, GO allows the model to recognize interactions among genes involved in critical processes like apoptosis, cell proliferation, and drug metabolism - central to a cancer cell's response to treatment. Additionally, GO annotations enable scLong to identify context-dependent roles of genes, including those that might be inactive under normal conditions but become essential under drug-induced stress. This integration of GO knowledge allows scLong to make more context-aware predictions, enhancing its ability to anticipate complex drug response patterns in cancer cells.

scLong's superior performance over task-specific models, including DeepCDR and DeepDDS, is attributed to its extensive pretraining on 48 million scRNA-seq data points. Predicting cancer drug response requires understanding the complex interactions and regulatory networks that dictate how cancer cells respond to treatment (32). This pretraining exposes scLong to a diverse range of gene expression profiles, enabling it to learn the underlying relationships and dependencies among genes, including those critical for modulating drug response in cancer cells. Additionally, gene expression patterns associated with drug resistance, metastasis, or apoptosis often appear only in specific cancer subtypes or under certain treatment conditions (32, 39). By capturing these rare but pivotal patterns, pretraining equips scLong to better understand and predict responses in heterogeneous cancer cell populations.

scLong infers gene regulatory networks. Gene Regulatory Networks (GRNs) represent the intricate interactions between genes and their regulators, such as transcription factors, that control gene expression within cells (40, 41). These networks determine which genes are turned on or off, guiding important cellular activities like differentiation, proliferation, and responses to environmental signals. Inferring GRNs from experimental data, such as single-cell RNA sequencing, is crucial for uncovering the regulatory mechanisms that drive these processes. Reconstructing these networks provides valuable insights into the molecular foundations of health and disease, highlighting key regulatory elements that may serve as potential therapeutic targets or biomarkers. Accurate GRN inference also enhances the ability to model and predict cellular behavior in response to specific conditions or treatments.

The input for this task consists of gene expression vectors from a collection of cells, with the output being a gene regulatory network represented as an adjacency matrix. We used gene expression data from $N_c = 758$ human embryonic stem cells (hESC) (42, 43), encompassing $N_g = 17,735$ genes. The pretrained scLong model is applied to extract representations of these genes, and an adjacency matrix is generated by calculating cosine similarities between these gene representations (Fig. 5a). This matrix is further refined using the beta variational autoencoder (44) in DeepSEM (45) (Methods). We evaluated this GRN by comparing it to a ground-truth GRN derived from ChIP-Seq (46) data. Area under the precision–recall curve ratio (AUPR) and early precision ratio (EPR) (42), where higher values indicate better performance, were used as evaluation metrics (Methods). We compared scLong's performance with the Geneformer foundation model and the task-specific DeepSEM method. scLong outperformed both Geneformer and DeepSEM across both metrics (Fig. 5b), demonstrating that its learned gene representations effectively capture gene interactions.

scLong's self-attention mechanism operates across the entire set of approximately 28,000 genes, encompassing both highly expressed and lowly expressed genes. This broad inclusion allows scLong to detect critical regulatory patterns

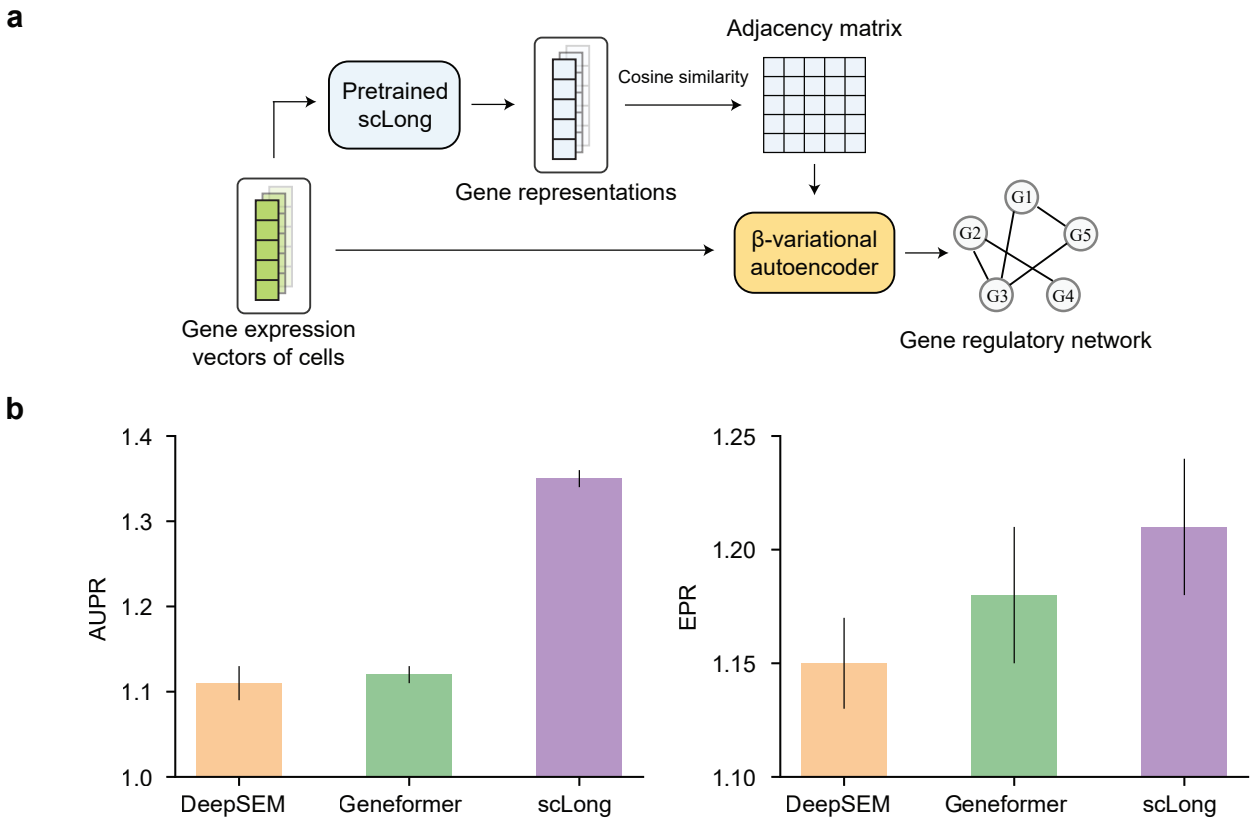


Fig. 5 | scLong outperforms existing scRNA-seq foundation models and task-specific methods in gene regulatory network inference. a, Model architecture for fine-tuning the pretrained scLong for this inference task. **b**, scLong achieves higher area under the precision-recall curve ratio (AUPR) and early precision ratio (EPR) compared to the Geneformer foundation model and the task-specific DeepSEM method.

that might be overlooked when focusing solely on highly expressed genes, as Geneformer does. Lowly expressed genes play crucial roles in cellular regulation, including acting as fine-tuners in regulatory networks or contributing to rare but essential cellular processes (20). By accounting for the expression patterns of these genes, scLong can infer a more complete and biologically accurate regulatory network. This comprehensive approach enables scLong to capture a broader range of gene interactions and dependencies, which are particularly important for uncovering regulatory relationships that influence rare or condition-specific cellular states. Furthermore, scLong leverages the Gene Ontology to construct functionally enriched representations of genes, adding another layer of precision in regulatory inference. By incorporating Gene Ontology, scLong is effectively pre-conditioned with structured knowledge about gene functions, pathways, and interrelations. This knowledge-rich representation provides a strong inductive bias, guiding scLong in recognizing functionally relevant connections between genes and their roles in regulatory networks. In contrast, Geneformer lacks this knowledge-based guidance, limiting its ability to identify meaningful relationships in cases where gene expression alone does not reveal functional interactions.

scLong clusters marker genes associated with different cell types. Fig. 6 illustrates the clustering of gene representations obtained from scLong for two cell types in the Zheng68K dataset (47). For each cell type, we randomly sampled 50 cells, used scLong to extract their gene representations, computed pairwise cosine similarity between genes, and conducted hierarchical clustering on the resulting similarity matrix (Methods). Displayed are the 50 genes with the highest similarity scores. The results show that marker genes for each cell type, highlighted in red, are grouped into the same cluster. These cell-type-specific genes are generally highly expressed within their respective cell types. Marker genes and non-marker genes are assigned to separate clusters. These results demonstrate scLong's capability to capture gene co-expression patterns within specific cell types.

Discussion

scLong offers a valuable advancement in single-cell transcriptomics, providing a scalable model that accommodates the full spectrum of gene expression within single cells. Its billion-parameter architecture and dual encoder strategy allow it to handle both high- and low-expression genes, addressing limitations in existing models that often overlook low-expression genes critical for cellular regulation. By integrating gene-specific information from the Gene Ontology, scLong brings contextual depth to its predictions, enhancing its ability to capture nuanced interactions across diverse cellular contexts. This comprehensive approach not only

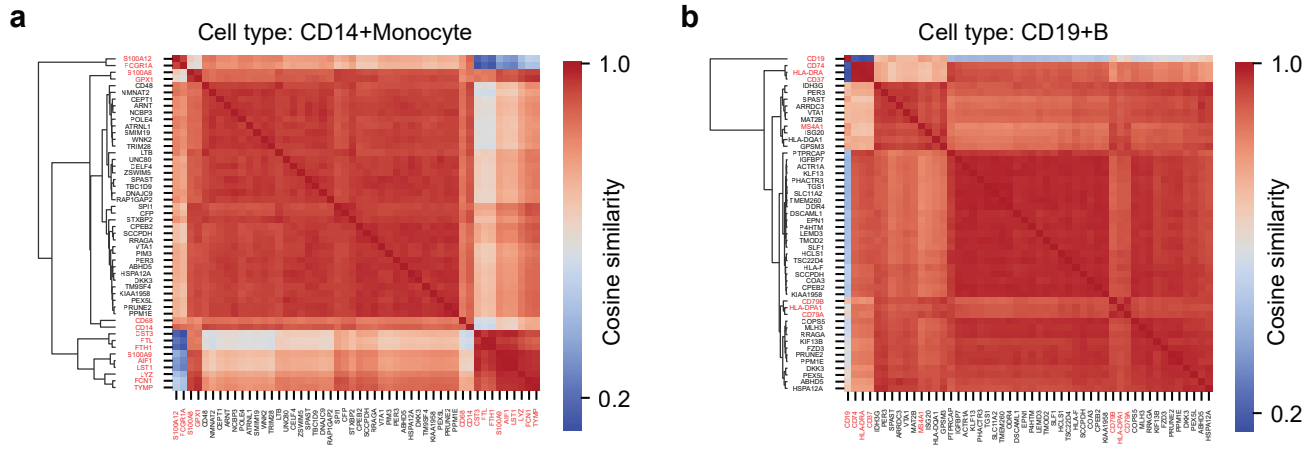


Fig. 6 | scLong groups together the marker genes of the same cell type. Hierarchical clustering was performed on cosine similarity matrices derived from gene representations extracted by scLong for two cell types: CD14+ monocytes and CD19+ B cells. The top 50 genes with the highest similarity scores are displayed. Marker genes (highlighted in red) for each cell type cluster together, while marker and non-marker genes are separated into distinct clusters.

improves prediction accuracy but also broadens the model's applicability in studying condition-specific responses and complex gene regulatory mechanisms. These capabilities make scLong a valuable tool for advancing research in precision medicine, drug discovery, and cellular biology, supporting new insights into gene expression dynamics and informing more targeted therapeutic approaches.

scLong's ability to outperform both state-of-the-art foundation models for scRNA-seq and task-specific models across diverse downstream tasks underscores its robustness and adaptability in single-cell transcriptomics. Its strong performance in predicting transcriptional responses to genetic and chemical perturbations highlights its potential to aid in uncovering gene functions, regulatory pathways, and cellular responses under various conditions - essential for understanding disease mechanisms and identifying therapeutic targets. In the context of cancer research, scLong's accurate predictions of drug responses, both for individual drugs and synergistic drug combinations, present valuable insights for precision oncology. This capability could facilitate personalized treatment approaches by helping to identify the most effective therapies based on specific cancer cell profiles, potentially improving patient outcomes and reducing adverse effects. Additionally, scLong's success in gene regulatory network inference signifies its capacity to map complex interactions among genes, supporting efforts to model cellular processes and regulatory circuits more precisely.

Balancing computational efficiency and representation quality presents a fundamental challenge in large-scale models like scLong, where both attributes are crucial yet often in conflict. Achieving high-quality representations typically requires complex processing, such as applying self-attention across long expression vectors, to capture intricate gene relationships accurately. However, this comprehensive modeling approach incurs significant computational costs, as attention operations scale quadratically with the number of elements, making such methods infeasible for gene expression vectors with tens of thousands of elements. Strategies aimed at enhancing efficiency, such as reducing the number of layers, attention heads, or hidden dimensions, often sacrifice representation richness and granularity, limiting the model's ability to detect subtle yet significant gene interactions. Some approaches improve efficiency by shortening vector length, excluding low-expression genes under the assumption that they contribute less to primary cellular insights (9, 10). While this reduces memory and computational loads, it inherently sacrifices the model's quality, as many low-expression genes are crucial for regulatory functions and context-specific cellular responses. scLong addresses this trade-off through a dual encoder strategy that selectively applies a larger encoder to high-expression genes, which typically convey more essential functional information, while a smaller encoder processes low-expression genes. This selective approach optimizes computational resources, allowing scLong to maintain high-quality representations for critical elements while managing efficiency. By retaining all genes in its representation and adjusting resource allocation appropriately, scLong preserves essential interactions among both high- and low-expression genes, achieving a balance between computational efficiency and comprehensive representation quality.

Despite its advancements, scLong has certain limitations that merit consideration. The model's billion-parameter architecture, although optimized for efficiency, still demands significant computational resources for training and inference, which may hinder accessibility for groups lacking high-performance infrastructure. Additionally, scLong relies on static, predefined relationships from sources like the Gene Ontology, which, while providing valuable contextual information, may restrict adaptability to dynamic gene interactions and condition-specific regulatory changes not represented in these databases. Another limitation is the

potential sensitivity of scLong's performance to the choice of high- and low-expression gene thresholds in its dual encoder design; selecting these thresholds inappropriately could lead to suboptimal representations, particularly in cell types with unusual gene expression distributions. Addressing these limitations could make scLong a more versatile and broadly applicable tool in single-cell transcriptomics research.

Future work on scLong can focus on several key areas to further enhance its capabilities and broaden its applications. One promising direction is the incorporation of additional biological datasets, such as pathway databases (48), protein-protein interaction networks (49), and epigenetic data (50), to enrich the context-awareness of the model and improve its ability to capture more complex regulatory mechanisms. Expanding the model's pretraining on diverse datasets from various species and tissues could also boost its generalizability across different biological contexts. Another area for improvement is model interpretability; future versions of scLong could integrate more advanced explainability techniques, such as attention-based visualization tools or saliency maps (51), to provide clearer insights into the gene interactions driving its predictions. Additionally, exploring methods to reduce the computational demands of training and deploying scLong, such as model pruning (52) or distillation (53), would make the model more accessible to a wider range of researchers. Finally, applying scLong to novel downstream tasks, such as predicting cell signaling pathways (5) or identifying gene interactions in rare cell populations, could further validate its versatility and expand its impact in single-cell biology.

References

1. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. *Nature*, 583 (7817):590–595, 2020.
2. Silvia Domcke, Andrew J Hill, Riza M Daza, Junyue Cao, Diana R O'Day, Hannah A Pliner, Kimberly A Aldinger, Dmitry Pokholok, Fan Zhang, Jennifer H Milbank, et al. A human cell atlas of fetal chromatin accessibility. *Science*, 370(6518):eaba7612, 2020.
3. Xiaoping Han, Ziming Zhou, Lijiang Fei, Huiyu Sun, Renying Wang, Yao Chen, Haide Chen, Jingjing Wang, Huanna Tang, Wenhao Ge, et al. Construction of a human cell landscape at single-cell level. *Nature*, 581(7808):303–309, 2020.
4. Dominic Grün, Anna V. Lyubimova, Lennart A. Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander van Oudenaarden. Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature*, 525:251–255, 2015.
5. Shengquan Jin, Christian F. Guerrero-Juarez, Longhao Zhang, et al. Inference and analysis of cell-cell communication using cellchat. *Nature Communications*, 12:1088, 2021. doi: 10.1038/s41467-021-21246-9.
6. Molly Gasperini, Andrew J. Hill, José L. McFaline-Figueroa, Beth Martin, Seungsoo Kim, Melissa D. Zhang, Dana Jackson, Anh Leith, Jacob Schreiber, William S. Noble, Cole Trapnell, Nadav Ahituv, and Jay Shendure. A genome-wide framework for mapping gene regulation via cellular genetic screens. *Cell*, 176(1):377–390.e19, 2019. doi: https://doi.org/10.1016/j.cell.2018.11.029.
7. 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*, 4:852–866, 2022.
8. 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, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618: 616–624, 2023.
9. Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21:1470–1480, 2024. doi: 10.1038/s41592-024-02201-0.
10. Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Le Song, and Xuegong Zhang. Large scale foundation model on single-cell transcriptomics. *Nature Methods*, 21:1481–1491, 2024. doi: 10.1038/s41592-024-02305-7.
11. Xiaodong Yang, Guole Liu, Guihai Feng, Dechao Bu, Pengfei Wang, Jie Jiang, Shubai Chen, Qinmeng Yang, Yiyang Zhang, Zhenpeng Man, Zhongming Liang, Zichen Wang, Yaning Li, Zheng Li, Yana Liu, Yao Tian, Ao Li, Jingxi Dong, Zhilong Hu, Chen Fang, Hefan Miao, Lina Cui, Zixu Deng, Haiping Jiang, Wentao Cui, Jiahao Zhang, Zhaohui Yang, Handong Li, Xingjian He, Liqun Zhong, Jiaheng Zhou, Zijian Wang, Qingqing Long, Ping Xu, The X-Compass Consortium, Hongmei Wang, Zhen Meng, Xuezhi Wang, Yangang Wang, Yong Wang, Shihua Zhang, Jingtao Guo, Yi Zhao, Yuanchun Zhou, Fei Li, Jing Liu, Yiqiang Chen, Ge Yang, and Xin Li. Genecompass: Deciphering universal gene regulatory mechanisms with knowledge-informed cross-species foundation model. *Cell Research*, 2024. doi: 10.1038/s41422-024-01034-y.
12. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.
13. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
14. Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, pages 1–9, 2023.
15. Hantao Shu, Jingtian Zhou, et al. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501, 2021. doi: 10.1038/s43588-021-00099-8.
16. Ying Sha, John H. Phan, and May D. Wang. Effect of low-expression gene filtering on detection of differentially expressed genes in rna-seq data. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6461–6464, 2015.
17. Hongmin Zhao, Hongli Yu, Jianhua Zheng, Ning Ning, Fanglan Tang, Yang Yang, and Yan Wang. Lowly-expressed lncrna gas5 facilitates progression of ovarian cancer through targeting mir-196-5p and thereby regulating hoxa5. *Gynecologic oncology*, 151(2):345–355, 2018.
18. Liang Yang, Shohei Takuno, Elizabeth R Waters, and Brandon S Gaut. Lowly expressed genes in arabidopsis thaliana bear the signature of possible pseudogenization by promoter degradation. *Molecular biology and evolution*, 28(3):1193–1203, 2011.
19. Zhipeng Zhou, Yunkun Dang, Mian Zhou, Lin Li, Chien-hung Yu, Jingjing Fu, She Chen, and Yi Liu. Codon usage is an important determinant of gene expression levels largely through its effects on transcription. *Proceedings of the National Academy of Sciences*, 113 (41):E6117–E6125, 2016.
20. Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539–542, 2018.
21. Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, Midori A Harris, David P Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C Matese, Joel E Richardson, Martin Ringwald, Gerald M Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25(1): 25–29, May 2000. doi: 10.1038/75556.
22. Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
23. Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6861–6871. PMLR, 2019.
24. Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.
25. Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7):1853–1866, 2016.
26. 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.
27. Ding Bai, Caleb N Ellington, Shentong Mo, Le Song, and Eric P Xing. Attentionpert: accurately modeling multiplexed genetic perturbations with multi-scale effects. *Bioinformatics*, 40(Supplement_1):i453–i461, 2024.
28. Bissan Al-Lazikani, Udai Banerji, and Paul Workman. Combinatorial drug therapy for cancer in the post-genomic era. *Nature biotechnology*, 30(7):679–692, 2012.
29. Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David D. Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie Donahue, Bina Julian, Mariya Khan, David Wadden, Ian C. Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Federica Piccioni, Sarah A. Johnson, Nicholas J. Lyons, Alice H. Berger, Alykhan F. Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Y. Takeda, Roger Hu, Desiree Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Peyton Greenside, Nathanael S. Gray, Paul A. Clemons, Serena Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, 2017. doi: https://doi.org/10.1016/j.cell.2017.10.049.
30. Thai-Hoang Pham, Yijie Wang, Jie Xu, and Ping Zhang. A deep learning framework for high-throughput mechanism-driven phenotype compound screening and its application to covid-19 drug repurposing. *Nature Machine Intelligence*, 3:247–257, 2021. doi: 10.1038/

s42256-020-00285-9.
31. Brent M. Kuenzi, Jeongsik Park, Shih-Han Fong, Kyla S. Sanchez, Jacob Lee, Jason F. Kreisberg, Jing Ma, and Trey Ideker. Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer Cell*, 38(5):672–684.e6, Nov 2020. doi: 10.1016/j.ccell.2020.09.014.
32. Florian T Unger, Irene Witte, and Kerstin A David. Prediction of individual response to anticancer therapy: historical and future perspectives. *Cellular and molecular life sciences*, 72:729–757, 2015.
33. Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
34. Qiao Liu, Zhiqiang Hu, Rui Jiang, and Mu Zhou. DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics*, 36:i911–i918, 12 2020. doi: 10.1093/bioinformatics/btaa822.
35. Shan Zhao, Tomohiro Nishimura, Yibang Chen, Evren U Azeloglu, Omri Gottesman, Chiara Giannarelli, Mohammad U Zafar, Ludovic Benard, Juan J Badimon, Roger J Hajjar, et al. Systems pharmacology of adverse event mitigation by drug combinations. *Science translational medicine*, 5(206):206ra140–206ra140, 2013.
36. J. O'Neil, Y. Benita, I. Feldman, M. Chenard, B. Roberts, Y. Liu, J. Li, A. Kral, S. Lejnine, A. Loboda, W. Arthur, R. Cristescu, B.B. Haines, C. Winter, T. Zhang, A. Bloecher, and S.D. Shumway. An unbiased oncology compound screen to identify novel combination strategies. *Molecular Cancer Therapeutics*, 15(6):1155–1162, June 2016. doi: 10.1158/1535-7163. MCT-15-0843. Epub 2016 Mar 16.
37. M. P. Menden, D. Wang, M. J. Mason, B. Szalai, K. C. Bulusu, Y. Guan, T. Yu, J. Kang, M. Jeon, R. Wolfinger, et al. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature Communications*, 10 (1):1–17, 2019.
38. Jinxian Wang, Xuejun Liu, Siyuan Shen, Lei Deng, and Hui Liu. DeepDDS: deep graph neural network with attention mechanism to predict synergistic drug combinations. *Briefings in Bioinformatics*, 23(1):bbab390, 09 2021. doi: 10.1093/bib/bbab390.
39. Paul Geeleher, Nancy J Cox, and R Stephanie Huang. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome biology*, 15:1–12, 2014.
40. Camille Berthelot, Diego Villar, Julie E Horvath, Duncan T Odom, and Paul Flicek. Complexity and conservation of regulatory landscapes underlie evolutionary resilience of mammalian gene expression. *Nature ecology & evolution*, 2(1):152–163, 2018.
41. Dawn Thompson, Aviv Regev, and Sushmita Roy. Comparative analysis of gene regulatory networks: from network reconstruction to evolution. *Annual review of cell and developmental biology*, 31(1):399–428, 2015.
42. Anushka Pratapa, Amogh P Jalihal, Justin N Law, Aditya Bharadwaj, and T M Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods*, 17(2):147–154, 2020. doi: 10.1038/s41592-019-0690-6.
43. Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James A Thomson. Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17:1–20, 2016.
44. Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
45. Hantao Shu, Jingtian Zhou, Qiuyu Lian, Han Li, Dan Zhao, Jianyang Zeng, and Jianzhu Ma. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7):491–501, 2021.
46. Shinya Oki, Tazro Ohta, Go Shioi, Hideki Hatanaka, Osamu Ogasawara, Yoshihiro Okuda, Hideya Kawaji, Ryo Nakaki, Jun Sese, and Chikara Meno. Chip-atlas: a data-mining suite powered by full integration of public chip-seq data. *EMBO reports*, 19(12):e46255, 2018.
47. Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1): 14049, 2017.
48. Antonio Fabregat, Sarah Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Cristina del-Toro Roca, Kevin Rothfels, Carlos Sevilla, Veronica Shamovsky, Scott Shorser, Thawfeek M. Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Research*, 46(D1):D649–D655, Jan 2018. doi: 10.1093/nar/gkx1132.
49. Damian Szklarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars J Jensen, and Christian von Mering. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, 49(D1):D605–D612, 11 2020. doi: 10.1093/nar/gkaa1074.
50. Jason D. Buenrostro, Bryan Wu, U. Litzenburger, William J. Greenleaf, and Howard Y. Chang. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523:486–490, 2015. doi: 10.1038/nature14590.
51. Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336 – 359, 2016.
52. Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In *Neural Information Processing Systems*, 2015.
53. Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.
54. Andrey Barski, Suresh Cuddapah, Keji Cui, Tae-Young Roh, Dustin E. Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keqiang Zhao. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823–837, May 2007. doi: 10.1016/j.cell.2007.05.009.

Methods

Collection and preprocessing of large-scale transcriptomics pretraining data. We collected scRNA-seq data from three public repositories: CELLxGENE[1], Cell Blast[2], and the Human Cell Atlas[3]. Initially, around 1,600 datasets were downloaded, comprising over 60 million cells. We filtered out non-human datasets and excluded those containing fewer than 1,000 genes. Additionally, datasets normalized using unknown methods were removed. After this filtering process, 848 datasets remained.

Next, we performed gene selection. First, we removed all non-human genes from the 848 datasets, leaving approximately 66,000 human genes. From these, we selected the top 20,000 genes with the highest number of non-zero entries across the datasets. Additionally, we included all 19,748 protein-coding genes (55), all 20,480 genes from the Gene Ontology (GO) (21), and all 20,184 genes from Gene2Vec (56). To eliminate duplicates, we mapped gene IDs from these different sources - represented as gene symbols or NCBI IDs - into a unified format based on Ensembl IDs. After removing duplicates, we obtained a final list of 27,874 unique genes.

For each cell, we created a 27,874-dimensional gene expression vector based on the 27,874 selected genes, where the j-th element represents the expression value of gene j in that cell. If a gene was not expressed in the cell, its value was set to 0. A cell was removed if it had fewer than 300 non-zero expression values. We then checked whether the expression values were in raw counts or already log1p normalized. If an expression value x was in raw count, we applied log1p normalization to it: $x \leftarrow \log(x/10000 + 1)$. Next, we adjusted the normalized expression values by magnifying or clipping them so that the maximum value in each cell's expression vector was 10. If the maximum value in an expression vector exceeded 10, all values greater than 10 were set to 10. If the maximum value was less than 10, each value in the vector was scaled by dividing it by the maximum value and then multiplying by 10. After removing duplicate cells, we retained 48,024,242 unique cells.

scLong model architecture. Each element in a gene expression vector contains two components: the gene ID and its expression value. scLong employs a gene encoder to generate a representation vector for the gene and an expression encoder to produce a representation vector for the expression value. The final representation for each element is obtained by adding these two vectors. The expression encoder is a multi-layer perceptron (MLP) that takes the scalar expression value as input and outputs a representation vector. This MLP consists of two layers, with ReLU activation (57), and generates a representation vector with a dimension of 200.

[1] https://cellxgene.cziscience.com/datasets
[2] https://cblast.gao-lab.org/
[3] https://www.humancellatlas.org/

The gene encoder first uses Gene2vec (56) to obtain an initial 200-dimensional representation for each gene. It then constructs a gene graph based on the Gene Ontology (GO), which, along with the initial representations, is input into a graph convolutional network (GCN) (23) to learn a refined representation for each gene. The gene graph is constructed as follows (14): for each gene pair, u and v, we retrieve their annotated GO terms from the Gene Ontology, denoted as N_u and N_v. GO terms are standardized categories that describe various attributes of genes, focusing on their roles, processes, and locations within a cell. They are organized into three main categories: Molecular Function, Biological Process, and Cellular Component. Each gene can be associated with multiple GO terms, providing a comprehensive view of its functional and spatial characteristics within cellular and molecular systems. We then compute the Jaccard index $J_{u,v} = \frac{|N_u \cap N_v|}{|N_u \cup N_v|}$ between the two sets of GO terms, which quantifies the fraction of shared GO terms and indicates the functional similarity of each gene pair. Using this similarity measure, we construct a graph where each gene is represented as a node, and edges are assigned between gene pairs with high Jaccard index values. Specifically, for each gene u, we select the top 20 genes v_i with the highest J_{u,v_i} values and connect them to u.

A one-layer GCN is constructed on the gene graph, taking $\mathbf{X}$, a matrix containing initial representation vectors generated by Gene2vec, as input. This GCN learns refined 200-dimensional representations $\mathbf{X}'$ for all genes using the following update equation:

$$\mathbf{X}' = \mathbf{D}^{-\frac{1}{2}} \widehat{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}} \mathbf{X} \mathbf{\Theta}, \tag{1}$$

where $\mathbf{\Theta}$ contains the GCN's weight parameters. Here, $\widehat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, with $\mathbf{A}$ representing the adjacency matrix of the gene graph and $\mathbf{I}$ as the identity matrix. $\mathbf{D}$ is a diagonal matrix with entries $D_{ii} = \sum_{j=1}^{K} \widehat{A}_{ij}$, where K is the total number of genes.

Given the extracted representation vectors for each element in the input gene expression vector, we feed them into self-attention layers (13) to learn enhanced representations of these elements. Self-attention computes pairwise correlations between elements, capturing the relationships among them. To balance computational efficiency with representation effectiveness, we employ a large Performer (24) encoder and a mini Performer encoder to process elements with varying expression magnitudes. First, we rank the elements in the gene expression vector in descending order of expression values and select the top 4,096 with the highest values for processing by the large Performer encoder. This encoder applies self-attention across all 4,096 high-expression elements, comprising 42 Performer layers with 32 attention heads and a hidden dimension of 1,280, and produces 200-dimensional output vectors. The remaining $K - 4,096$ elements, where $K = 27,874$ represents the total number of genes, are processed by a mini Performer encoder tailored for lower expression values. This encoder performs self-attention across all $K - 4,096$ elements. This mini encoder has 2 layers, 8 attention heads, and a hidden dimension of 200, yielding 200-dimensional output representations as well. After processing by these encoders, each element in the input expression vector has a 200-dimensional representation, derived from either the large or mini encoder. These representations are then fed into a full-length Performer encoder, which performs self-attention across all 27,874 elements. This final encoder has 2 layers, 8 attention heads, and a hidden dimension of 200. The resulting representations from the full-length encoder serve as the final outputs of scLong and are utilized for a range of downstream tasks.

scLong pretraining. scLong was pretrained using a masked value reconstruction task. In this approach, 15% of the non-zero values in each input gene expression vector were randomly masked, and the model was trained to predict the masked values based on the unmasked portions of the vector. The 15% masking ratio followed that used in BERT (12). Let M_x represent the set of indices corresponding to masked gene expressions in an input gene expression vector x. We create a masked expression vector x' by assigning a special symbol [MASK] to each masked gene while leaving unmasked values intact:

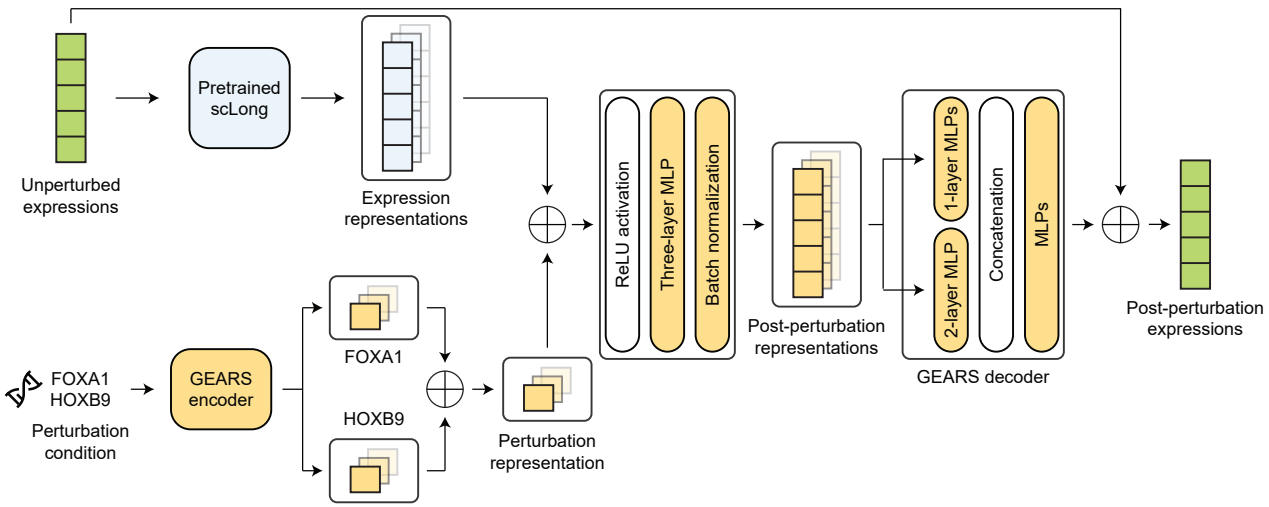
$$x'_i = \begin{cases} \text{[MASK]}, & \text{if } i \in M_x \\ x_i, & \text{if } i \notin M_x \end{cases}$$

We then obtain a representation vector for each element in x'. For unmasked expression values x_i, we apply an MLP to generate their representation vectors as previously described. For each [MASK] symbol, we use a learnable representation vector specific to [MASK]. These representation vectors for elements in x' are subsequently fed into the remaining layers of scLong to compute a final representation for each element. Finally, the representation vector corresponding to each masked gene is processed through a gene-specific MLP, producing a scalar representing the reconstructed value for that gene's masked expression. Let $\hat{x}_i$ and x_i represent the reconstructed value and the ground truth (pre-masking) value of a masked gene i, respectively. The reconstruction loss is measured as the mean squared error between $\hat{x}_i$ and x_i. Pretraining is performed by minimizing the reconstruction loss across the dataset. Let D denote the entire pretraining dataset. The overall pretraining loss is defined as:

$$\mathcal{L} = \frac{1}{|D||M_x|} \sum_{x \in D} \sum_{i \in M_x} (\hat{x}_i - x_i)^2, \tag{2}$$

where $|\cdot|$ denotes the size of the set.

During pretraining, we divided the 48 million cells into two sets: 95% for training and 5% for validation. The pretraining process was implemented using PyTorch Distributed Data Parallelism (58) and half-precision BFLOAT16 operations. The model was trained across 12 machines, each equipped with 8 A100 GPUs (80GB memory per GPU).



Extended Data Fig. 1 | Model architecture for predicting transcriptional responses to genetic perturbations.

Training was conducted over 5 epochs, taking a total of 35 GPU days. The batch size per GPU was set to 1, with a gradient accumulation step size of 200. We employed the Adam (59) optimizer with default hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and no weight decay. A cosine annealing scheduler was used to adjust the learning rate. The first cycle step size was 15, with cycle step magnification of 2. The maximum and minimum learning rates for the first cycle were 5×10^{-5} and 10^{-6}, respectively. The linear warmup step size was 5. The max learning rate decreased by a factor of 0.9 in each cycle.

Prediction of transcriptional responses to genetic perturbations. For this task, we used the Norman dataset (26), preprocessed with GEARS (14), which includes 91,205 cell samples and 5,045 genes. The dataset features 236 perturbation conditions: 105 involving single-gene perturbations, such as 'FOXA1' and 'HOXB9', and 131 involving double-gene perturbations, such as '{ZBTB10, SNAI1}', '{CDKN1A, CDKN1B}', and '{FOXA1, HOXB9}'. Each double-gene perturbation is a combination of two single-gene perturbations. We adopted the data split used in GEARS, comprising a training set of 58,134 cells, a validation set of 6,792 cells, and a test set of 26,279 cells. Each test sample was assigned to one of four categories: 1) neither gene in a double-gene perturbation appears in the training data (Seen 0/2); 2) one gene in a double-gene perturbation is absent from the training data (Seen 1/2); 3) both genes in a double-gene perturbation are present in the training data (Seen 2/2); and 4) the gene in a single-gene perturbation is absent from the training data (Seen 0/1).

Extended Data Fig. 1 illustrates the model architecture used for this downstream task. The input includes a gene expression vector of a cell prior to perturbation and the associated perturbation condition. The output is the gene expression vector of the cell following perturbation. We use the pretrained scLong model to derive a representation vector for each element of the pre-perturbation expression vector, while the GEARS method generates a representation for the perturbation condition. These vectors are then combined and processed through the GEARS decoder to predict the post-perturbation gene expression vector. Specifically, GEARS generates a 200-dimensional representation vector for each single-gene perturbation. For a double-gene perturbation, its representation is obtained by summing the vectors of the two individual single-gene perturbations it comprises. The representation vector for the perturbation condition is added to the representation vector of each element in the gene expression vector extracted by scLong. A ReLU activation is then applied to each dimension of the resulting vectors. Each vector is subsequently passed through a three-layer MLP with hidden dimensions of 200, 400, and 200, followed by a batch normalization (60) layer, producing a 200-dimensional post-perturbation representation for each expression element. Finally, each post-perturbation representation vector is processed by a decoder to generate post-perturbation values. The decoder begins with a one-layer MLP, which takes a post-perturbation representation as input and outputs an initial predicted post-perturbation value. Simultaneously, the decoder concatenates the post-perturbation representations of all expression elements, passing this combined vector through a two-layer MLP (with hidden dimensions of 5045 and 200) to produce a 200-dimensional vector. This vector is then concatenated with the initial predicted post-perturbation value of each element and fed into another MLP, which outputs an additional scalar prediction for each element. This scalar is then added to the corresponding pre-perturbation expression value to yield the final predicted post-perturbation value.

We assess the discrepancy between predicted and ground-truth post-perturbation expression vectors, denoted as $\widehat{\mathbf{y}}$ and $\mathbf{y}$ respectively, using a composite loss function adopted from GEARS, which includes an autofocus loss and a direction-aware loss. Let c denote the perturbation

condition corresponding to $\mathbf{y}$ and S_c represent the set of post-perturbation expression vectors in the training data resulting from applying c. Define Z_c as the subset of genes that exhibit non-zero expression in at least one vector within S_c. The autofocus loss is defined as follows::

$$\mathcal{L}_{\text{af}} = \frac{1}{|Z_c|} \sum_{k \in Z_c} \left(\widehat{y}_k - y_k\right)^4. \quad \textbf{(3)}$$

Let $\mathbf{x}$ denote the pre-perturbation expression vector corresponding to $\mathbf{y}$. The direction-aware loss function assesses the alignment of directional changes between the predicted and actual post-perturbation expressions relative to their pre-perturbation states:

$$\mathcal{L}_{\text{d}} = \frac{1}{|Z_c|} \sum_{k \in Z_c} \left[\text{sign}\left(\hat{y}_k - x_k\right) - \text{sign}\left(y_k - x_k\right)\right]^2, \quad \textbf{(4)}$$

where $\text{sign}(\cdot)$ denotes the sign function, which determines the sign of a real number a:

$$\text{sign}(a) = \begin{cases} -1 & \text{if } a < 0, \\ 0 & \text{if } a = 0, \\ 1 & \text{if } a > 0. \end{cases}$$

The model's total loss function is a sum of the autofocus loss and the direction-aware loss. During training, the model aims to minimize this total loss across all data examples.

The hyperparameters for our method were mostly the same as those used in GEARS. The hidden dimension of the GEARS encoder and decoder was set to 1024, with ReLU as the activation function. Model weights were optimized using the Adam (59) optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, $\lambda = 5 \times 10^{-4}$, learning rate $\gamma = 10^{-3}$). Training was conducted across 4 GPUs, with a batch size of 16 per GPU. With 8 gradient accumulation steps, the effective overall batch size was $16 \times 8 \times 4 = 512$. Our model was trained for 16 epochs, while GEARS was trained for 20 epochs to replicate their reported results. Early stopping was applied when performance on the validation set began to decline.

To evaluate the model's performance, we employed two metrics: Mean Squared Error (MSE) and Pearson Correlation Coefficient (PCC), focusing on the top 20 differentially expressed (DE) genes. The set of top 20 DE genes for a given perturbation condition c, denoted as D_c, was identified as the 20 genes with the highest variance across the expression vectors in S_c, the set of post-perturbation expression vectors under condition c. For each expression vector - predicted ($\widehat{\mathbf{y}}$), ground truth ($\mathbf{y}$), and pre-perturbation ($\mathbf{x}$) - under perturbation condition c, we extracted the subvector corresponding to D_c, denoted as $\hat{\mathbf{y}}[D_c]$, $\mathbf{y}[D_c]$, and $\mathbf{x}[D_c]$, respectively. The MSE on D_c is calculated as:

$$\text{MSE}(D_c) = \frac{1}{|D_c|} \sum_{k \in D_c} \left(\hat{y}_k - y_k\right)^2. \quad \textbf{(5)}$$

The PCC on D_c assesses the correlation between the predicted and actual changes in expression from pre- to post-perturbation:

$$\text{PCC}(D_c) = \text{PCC}(\widehat{\mathbf{y}}[D_c] - \mathbf{x}[D_c], \mathbf{y}[D_c] - \mathbf{x}[D_c]), \quad \textbf{(6)}$$

where the Pearson Correlation Coefficient $\text{PCC}(\mathbf{v}, \mathbf{u})$ between two n-dimensional vectors, $\mathbf{v}$ and $\mathbf{u}$, is given by:

$$\text{PCC}(\mathbf{v}, \mathbf{u}) = \frac{\sum_{i=1}^{n}(v_i - \bar{v})(u_i - \bar{u})}{\sqrt{\sum_{i=1}^{n}(v_i - \bar{v})^2}\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2}}. \quad \textbf{(7)}$$

In this formula, $\bar{v}$ and $\bar{u}$ represent the mean values of $\mathbf{v}$ and $\mathbf{u}$, respectively. For each of the four test sample categories - Seen 0/2, Seen 1/2, Seen 2/2, and Seen 0/1 - we calculated the $\text{MSE}(D_c)$ and $\text{PCC}(D_c)$ for each sample. The overall performance for each category was then determined by averaging $\text{MSE}(D_c)$ and $\text{PCC}(D_c)$ across all samples in that category.

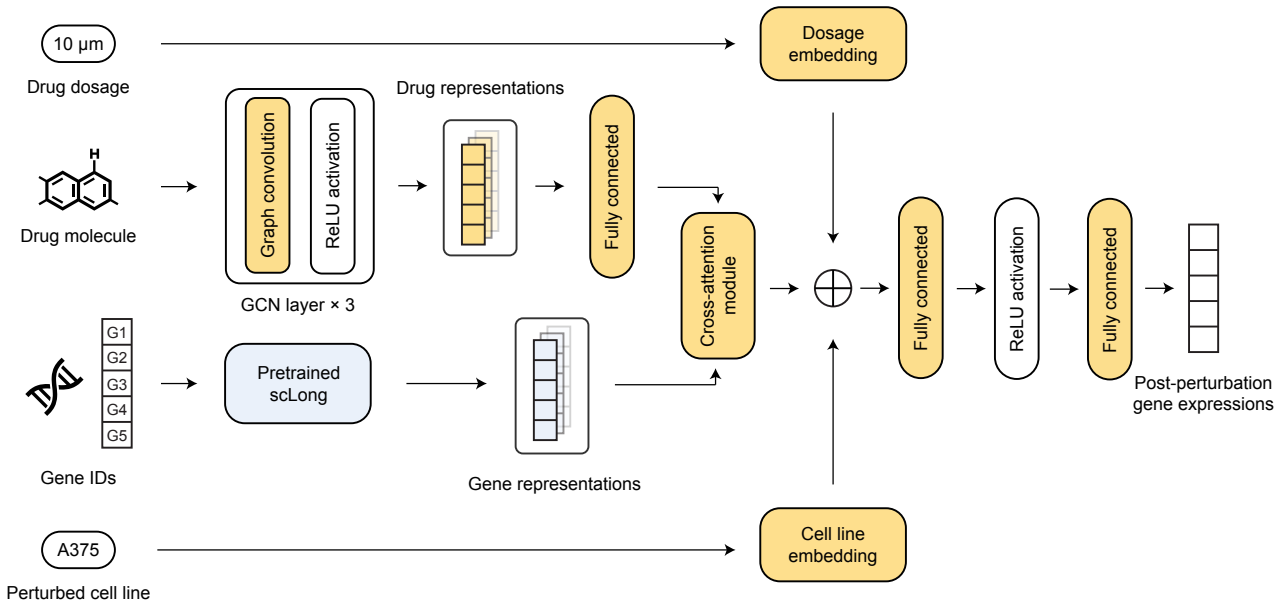
In classifying gene interaction types, we utilize a magnitude score (14). For a double-gene perturbation $\{i, j\}$, scLong's magnitude score is defined as follows. Let $\mathbf{x}$ denote a cell's pre-perturbation expression vector, and $\widehat{\mathbf{y}}$ represent the post-perturbation expression vector predicted by scLong for the same cell under perturbation condition $\{i, j\}$. The difference $\widehat{\mathbf{y}} - \mathbf{x}$ is considered the perturbation effect of $\{i, j\}$ for this cell. Repeating this process for all test cells, we calculate the average perturbation effect, denoted as $\mathbf{\Delta}_{i,j}$. Similarly, we compute the average perturbation effect $\mathbf{\Delta}_i$ for single-gene perturbation i and $\mathbf{\Delta}_j$ for single-gene perturbation j. We then solve the following equation:

$$\mathbf{\Delta}_{i,j} = \alpha \mathbf{\Delta}_i + \beta \mathbf{\Delta}_j + \mathbf{c}, \quad \textbf{(8)}$$

where α and β are scalars representing the linear combination coefficients, and $\mathbf{c}$ is an offset vector. The magnitude score is then defined as $\sqrt{\alpha^2 + \beta^2}$. Hierarchical clustering in Fig. 2e was conducted using the Seaborn library (61).

Prediction of transcriptional responses to chemical perturbations. In this task, we used a subset of the L1000 dataset (30), which comprises 7 distinct cell lines, 978 genes, and 810 drug compounds, each tested at 6 dosage levels. The prediction model takes two inputs: 1) the index of a perturbed cell line, and 2) the molecular graph and dosage level of the drug used to induce the perturbation. The model's output is the gene expression profile of the cell line following perturbation. The dataset does not provide pre-perturbation gene expression data. Each example in the L1000 dataset includes these inputs and outputs, with a total of 5,005 examples divided into 3,965 for training, 544 for validation, and 496 for testing.

Extended Data Fig. 2 illustrates the model architecture for this task. The perturbed cell line is encoded using a 7-dimensional one-hot vector, where each dimension represents one of the 7 cell lines, and each is associated



Extended Data Fig. 2 | Model architecture for predicting transcriptional responses to chemical perturbations.

with a 4-dimensional learnable embedding. Similarly, the drug dosage is encoded using a 6-dimensional one-hot vector, with each dimension corresponding to one of the 6 dosages, and each is linked to a 4-dimensional learnable embedding. For each of the 978 genes, scLong extracts a 200-dimensional representation vector as the output of its graph convolutional network (GCN) built on the gene graph. This vector is then passed through a linear projection layer to generate a 512-dimensional representation. We employ a GCN to extract a representation vector for each atom in the input drug molecule graph. The GCN consists of three convolutional layers, each with a hidden dimension of 128. Next, cross-attention (13) is applied between genes and drug atoms to capture their interaction patterns. Specifically, the representation vectors of drug atoms extracted by the GCN serve as both the key and value vectors in the cross-attention module, while the gene representation vectors, obtained from scLong and the subsequent linear layer, serve as the query vectors. Prior to cross-attention, the drug atom representations are mapped to 512-dimensional vectors via a linear projection layer to align with the gene representation dimensions. The cross-attention module consists of 2 attention layers, 4 attention heads, and a hidden dimension of 512. Finally, the gene representations from the cross-attention module are integrated with drug, cell line, and dosage information. Specifically, the 512-dimensional representation vector of each gene obtained from cross-attention is concatenated with a 4-dimensional cell line embedding, a 4-dimensional dosage embedding, and a 128-dimensional drug representation vector averaged across the representations of all atoms in the drug. The resulting concatenated vector is then passed through a 2-layer MLP to predict the post-perturbation expression for each gene. These two layers have dimensions of 648 and 1, respectively, with ReLU serving as the activation function. The predictions for each gene are concatenated to form the final predicted gene expression vector.

The model was trained by minimizing the mean squared error (MSE) between the predicted and ground-truth post-perturbation gene expression vectors, using the Adam optimizer with a fixed learning rate of 2e-4 (without learning rate scheduler), betas of (0.9, 0.999), epsilon of 1e-08, and no weight decay. The training was conducted with a batch size of 16 for a maximum of 100 epochs. The final performance was evaluated on the test set, using the checkpoint that achieved the best validation performance.

We compared our method with Geneformer and the task-specific DeepCE model, all of which have similar model configurations, differing only in their approaches to gene representation extraction. In DeepCE, gene representations are derived from the STRING protein interaction network (49) using the Node2Vec method (62). Geneformer obtains representations from its pretrained model. Spearman and Pearson correlation coefficients, root mean squared error (RMSE), and precision for top-K positive and negative predictions were used as evaluation metrics. Given predicted and ground truth gene expression vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$, we first rank the values in each vector in descending order. Let $R(X) = (R(x_1), \ldots, R(x_n))$ and $R(Y) = (R(y_1), \ldots, R(y_n))$ represent the rank positions for values in X and Y, respectively. The Spearman correlation (SC) score is defined as: $SC(X,Y) = \rho(R(X), R(Y))$, where ρ denotes Pearson correlation. The root mean squared error (RMSE) between X and Y is calculated as:

$$RMSE(X,Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}. \quad \textbf{(9)}$$

To compute the positive precision at K (Pos-P@K), we identify the top-K genes with the highest expression values in Y as G_y and in X as G_x. Pos-P@K is defined as:

$$Pos\text{-}P@K = \frac{1}{K}|G_y \cap G_x|, \quad \textbf{(10)}$$

reflecting the proportion of genes in the predicted set G_x that are also present in the ground truth set G_y. The negative precision at K (Neg-P@K) is computed analogously, using the lowest-expressed genes in X and Y. In our calculations, we set $K = 100$.

Prediction of cancer drug responses. The dataset (34) for this task was created by combining the Cancer Cell Line Encyclopedia (CCLE) (63) and the Genomics of Cancer Drug Sensitivity (GDSC) (64) datasets, resulting in 561 cancer cell lines and 238 drugs. Each cell line is represented by a bulk gene expression vector of 697 genes. Out of the total $561 \times 238 = 133,518$ possible (cell line, drug) interaction pairs, approximately 19.5% (26,072) had missing IC50 values, leaving 107,446 complete pairs. From these, 95% were allocated for training and 5% for testing.

Extended Data Fig. 3 illustrates the model architecture for this task. The input consists of the molecular structure of a potential cancer drug and the bulk gene expression profile of a cancer cell line. The output is a prediction of the drug's efficacy against the cancer cell line, quantified by its half-maximal inhibitory concentration (IC50) value. We use the scLong model to extract a representation vector from the gene expression data, which is concatenated with the drug molecule representation obtained via a graph convolutional network (GCN). This combined representation is then passed through a regression module to predict the IC50 value. Specifically, the scLong model processes a 697-dimensional gene expression vector as input, generating a 100-dimensional representation for each gene, resulting in a 697×200 matrix. We apply average pooling across the 200 dimensions to reduce this matrix to a 697-dimensional vector. This representation is then passed through a two-layer MLP, with hidden dimensions of 256 and 100, respectively. The output of this MLP is a final 100-dimensional representation that captures the cell line's features. The GCN consists of 3 convolutional layers, each with a hidden dimension of 100 and using the ReLU activation function. It learns a 100-dimensional representation for each atom in the molecular graph. To obtain a single representation for the entire molecule, average pooling is applied across the atom-level vectors. The regression module comprises a one-layer MLP with a hidden dimension of 300, followed sequentially by a convolutional neural network (CNN) with 3 layers and a final linear layer that outputs a scalar. The CNN layers have filter numbers and kernel sizes of (30, 150), (10, 5), and (5, 5), respectively. The scalar from the linear layer is passed through a sigmoid function to predict the IC50 value. To prevent overfitting, a dropout (65) rate of 0.1 is applied to all layers.

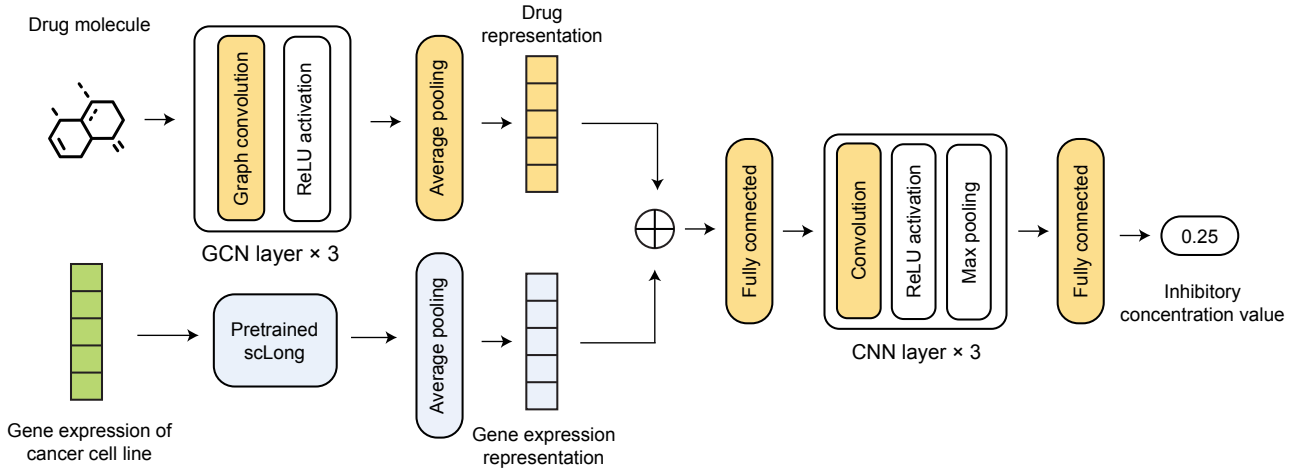
The model was trained using a mean squared error (MSE) loss function, which measures the difference between the predicted and ground truth IC50 values. We optimized the model parameters using the Adam optimizer (59), with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate was set to a fixed value of 0.001, without the use of a learning rate scheduler. Training was conducted for 500 epochs with a batch size of 64.

We compared our method with two baseline approaches: DeepCDR (34) and Geneformer. The only difference between these baselines and our method lies in how the representation vectors are extracted from the input gene expression data, while the rest of the model architecture and hyperparameter settings remain identical. In DeepCDR, the raw gene expression vector is directly fed into the regression module without learning additional representations. For the Geneformer baseline, we used the pretrained Geneformer model to extract a representation from the gene expression vector before passing it to the regression module. We used Pearson correlation, defined similarly to Equation 7, as the evaluation metric.

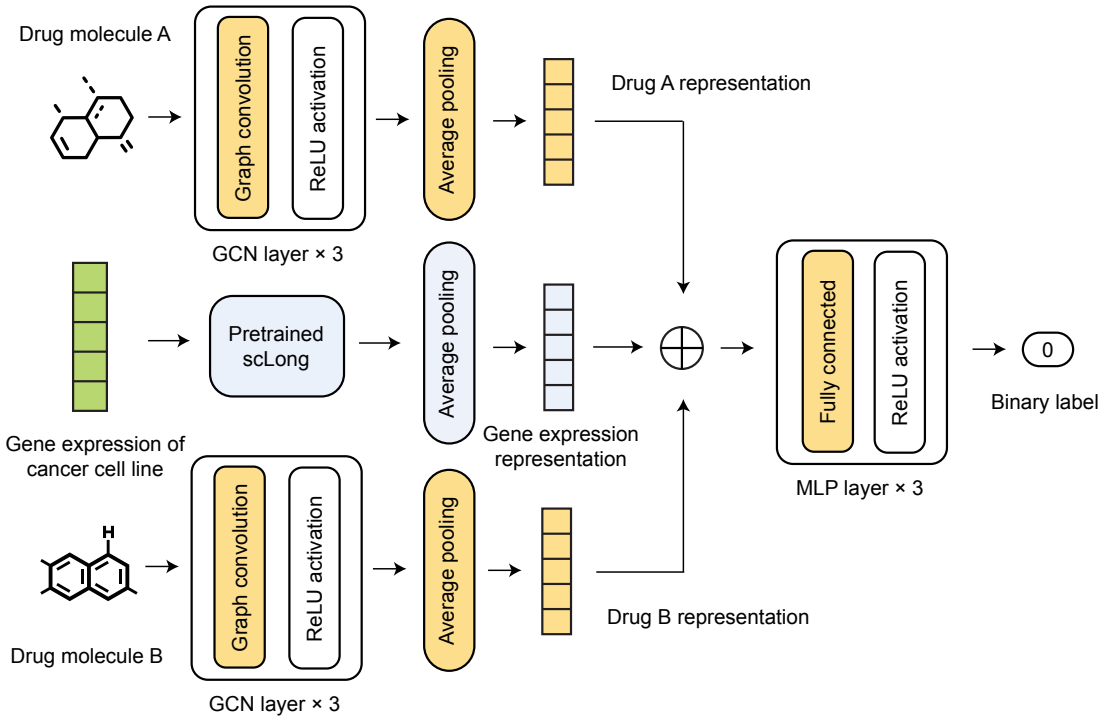
Prediction of cancer cell line responses to synergistic drug combinations. In this task, each data sample consists of a bulk gene expression vector from a cancer cell line, a pair of drugs, and a binary label indicating whether the drug combination is effective against the cell line. The training dataset, sourced from (36), includes 12,415 examples, spanning 36 anti-cancer drugs and 31 human cancer cell lines. The test dataset, obtained from an independent source (37), contains 668 examples, covering 57 drugs and 24 cancer cell lines. Both datasets include gene expression values for 954 genes.

Extended Data Fig. 4 illustrates the model architecture for this task, which closely resembles the architecture used for single-drug response prediction (Extended Data Fig. 3). First, the 954-dimensional gene expression vector is processed by the scLong model, yielding a (954, 200) representation matrix. After applying average pooling across the 200 dimensions, we obtain a 954-dimensional representation vector. This vector is then passed through a 3-layer MLP with hidden dimensions of 512, 256, and 128, using the ReLU activation function. For the drug pair input, two separate 3-layer GCNs are employed to generate a representation for each drug. The GCN layers use ReLU as the activation function, with hidden dimensions of 1024, 512, and 156, respectively. The representations of the two drugs are concatenated with the gene expression representation obtained from the MLP. The combined representation is then passed through another 3-layer MLP to predict the binary output label, with ReLU activation and hidden dimensions of 1024, 512, and 128, respectively.

The model was optimized using cross-entropy loss with the Adam optimizer (59). The learning rate was set to 1e-4, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$, and no weight decay.



Extended Data Fig. 3 | Model architecture for predicting cancer cell responses to individual drugs.

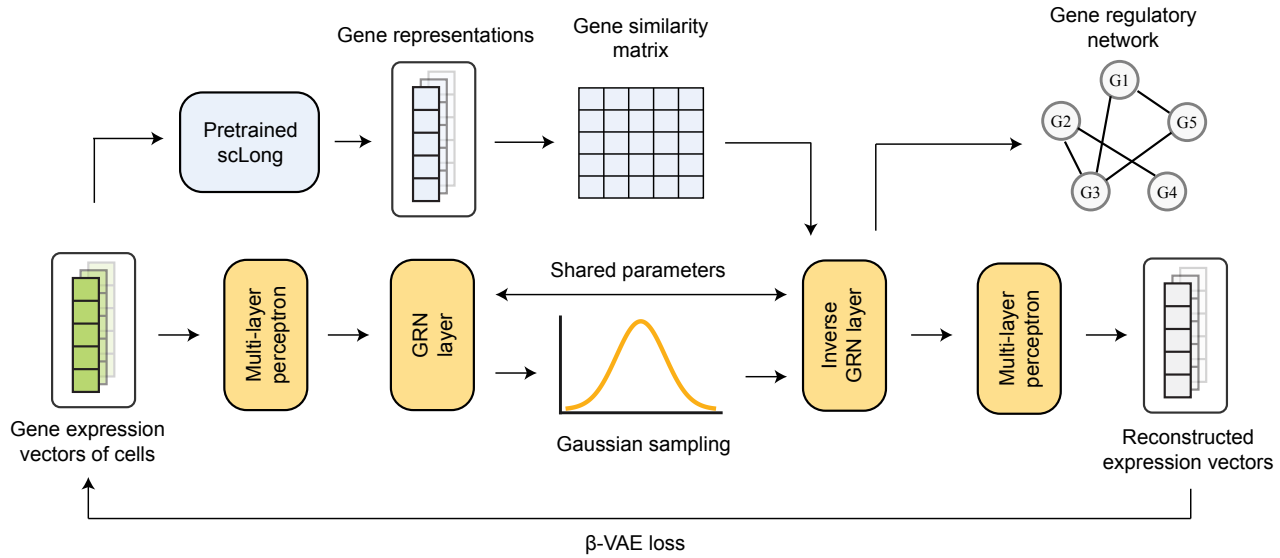


Extended Data Fig. 4 | Model architecture for predicting cancer cell responses to synergistic drug combinations.

The model was trained with a batch size of 256 for 1,000 epochs. The baseline method, DeepDDS (38), directly used raw gene expression vectors as the representations of cell lines, without learning additional latent representations. For the Geneformer baseline, the Geneformer model was used to extract representations from the gene expression vectors. All other model settings and hyperparameters were kept consistent with our method. The performance of the models in this binary classification task was evaluated using the AUROC score.

Inference of gene regulatory networks. The input for this task consists of gene expression vectors from a collection of cells, with the output being a gene regulatory network represented as an adjacency matrix. In this matrix, each element indicates the interaction strength between two genes. We utilized gene expression data from $N_c = 758$ human embryonic stem cells (hESC) (42), covering $N_g = 17,735$ genes.

Extended Data Fig. 5 presents the model architecture used for this task. Initially, we applied the pretrained scLong model to extract a $N_g \times 200$ representation matrix from each cell's gene expression vector, where each row is a 200-dimensional representation vector of gene expression elements. Aggregating these matrices across cells yields a



Extended Data Fig. 5 | Model architecture for inferring gene regulatory networks.

$N_c \times N_g \times 200$ tensor. By averaging over the last dimension of this tensor, we obtain a $N_c \times N_g$ matrix, where each column serves as a new representation vector for a gene across all cells. We then compute a $N_g \times N_g$ adjacency matrix A by calculating the cosine similarity between the N_c-dimensional representation vectors of each pair of genes, capturing gene-gene relationships. Following DeepSEM (15), we use the preliminary adjacency matrix A as an initialization for further refinement with a beta variational autoencoder (beta-VAE) (44). The beta-VAE framework uses A to model gene expression vectors and consists of a probabilistic encoder and decoder. The encoder defines a conditional distribution $p(z|x)$, where x is a gene expression vector and z is a 128-dimensional latent vector representing x. The decoder defines a conditional distribution $p(x|z)$. Both distributions are parameterized as multivariate Gaussians, with the mean and covariance for $p(z|x)$ computed by an encoder network receiving x as input, and for $p(x|z)$, by a decoder network taking z as input. The encoder network comprises a three-layer MLP followed by a GRN layer parameterized by A, while the decoder network includes a reverse GRN layer also parameterized by A, followed by a three-layer MLP. The GRN layer applies a linear transformation with parameters $I - A$, and the reverse GRN layer applies another linear transformation with parameters $(I - A)^{-1}$. The total loss for the beta-VAE, a weighted sum of reconstruction and KL divergence losses, optimizes the encoder and decoder MLPs and refines the adjacency matrix A to produce the final inferred GRN, leveraging the reparameterization trick (66). Both the encoder and decoder MLPs have a hidden dimension of 128 and use Tanh as the activation function.

The model was optimized using the RMSProp optimizer (67) with a learning rate of 2e-5 for the adjacency matrix and 1e-4 for other parameters. We set $\alpha = 0.99$, $\epsilon = 1e-08$, weight decay to 0, and momentum to 0. A linear learning rate scheduler was applied with a step size of 0.99 and $\gamma = 0.95$. Training was conducted with a batch size of 64 over 120 epochs.

We compared our method with DeepSEM and Geneformer, keeping most model configurations consistent. The primary difference lies in how each approach initializes the adjacency matrix A. DeepSEM initializes A with a uniform distribution in the range (0, 2e-4), Geneformer initializes A using cosine similarity between gene representation vectors derived from Geneformer, and our method initializes A based on gene representations extracted by scLong, as described earlier. To evaluate these methods, we used a gene regulatory network derived by ChIP-Seq (54) as the ground truth. This network comprises 2,762 nodes representing genes, 436,563 edges representing gene interactions, and includes 487 transcription factors (TFs). We compared the GRNs inferred by different methods against this ground-truth GRN. Given the high sparsity of the ground-truth network - only a small fraction of gene pairs exhibit regulatory interactions (436,563 out of a possible 17,735×17,735 pairs) - we employed early precision ratio (EPR) (42) and area under the precision–recall curve ratio (AUPR) (42) as evaluation metrics. EPR is defined as the ratio between the Pos-P@K score of an inferred adjacency matrix (as previously specified) and that of a random predictor. Here, we set K to $436,563$, which corresponds to the number of edges in the ground truth GRN. Specifically, the edges with the top-K values from the inferred adjacency matrix A form an edge set E_p, and we calculate the proportion of these edges that also appear in the ground truth edge set E_g using the formula $\frac{1}{K}|E_p \cap E_g|$. The random predictor estimates the presence of an edge between a gene pair with a probability of $p = 436,563/(17,735 \times 17,735)$, which represents the ratio of edges in the ground truth GRN to the

total possible edges. The Pos-P@K score for the random predictor is thus p. AUPR is computed as the ratio of the area under the precision-recall curve (AUPRC) for the inferred adjacency matrix to that of the random predictor, where the random predictor's AUPRC equals p.

Clustering of gene representations extracted by scLong. For each cell type in the Zheng68K dataset (47), which includes 11 cell types, 68,000 cells, and 20,000 genes, we randomly sampled 50 cells. Using the pretrained scLong model, we extracted representations for each cell's gene expression elements. To generate an overall representation vector for each gene, we averaged its representations across the 50 sampled cells. We then calculated the cosine similarity between each gene pair, where the cosine similarity of two vectors, $\mathbf{x}$ and $\mathbf{y}$, is given by $\frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$, with $\mathbf{x}\cdot\mathbf{y} = \sum_i x_i y_i$ as the dot product and $\|\mathbf{x}\| = \sqrt{\mathbf{x}\cdot\mathbf{x}}$ as the L^2 norm. Cosine similarity values range from -1 to 1. We selected the 50 genes with the highest cumulative similarity scores for further analysis. Hierarchical clustering was then performed on the similarity matrix using the *clustermap* function from the Seaborn (61) library. The Zheng68K dataset includes known marker genes identified from prior studies for each cell type (47). Marker genes, or cell-type-specific genes, are typically expressed at high levels in a specific cell type and at low levels in others (68). These genes are essential for manual or semi-supervised cell classification (69, 70), and the dataset provider used them to classify cells into the 11 defined types.

Data availability

The dataset curated and utilized in this study can be accessed at https://mbzuaiac-my.sharepoint.com/:f:/g/personal/ding_bai_mbzuai_ac_ae/EpvKzQW4hI5Bnb88-iM7vE0B_e2_U5r_ZGXb_FILCLTw3Q

Code availability

The source code for this work is available at https://github.com/BaiDing1234/scLong

References

55. Peter W Harrison, M Ridwan Amode, Olanrewaju Austine-Orimoloye, Andrey G Azov, Matthieu Barba, If Barnes, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, Simarpreet Kaur Bhurji, Sanjay Boddu, Paulo R Branco Lins, Lucy Brooks, Shashank Budhanuru Ramaraju, Lahcen I Campbell, Manuel Carbajo Martinez, Mehrnaz Charkhchi, Kapeel Chougule, Alexander Cockburn, Claire Davidson, Nishadi H De Silva, Kamalkumar Dodiya, Sarah Donaldson, Bilal El Houdaigui, Tamara El Naboulsi, Reham Fatima, Carlos Garcia Giron, Thiago Genez, Dionysios Grigoriadis, Gurpreet S Ghattaoraya, Jose Gonzalez Martinez, Tatiana A Gurbich, Matthew Hardy, Zoe Hollis, Thibaut Hourlier, Toby Hunt, Mike Kay, Vinay Kaykala, Tuan Le, Diana Lemos, Disha Lodha, Diego Marques-Coelho, Gareth Maslen, Gabriela Alejandra Merino, Louisse Paola Mirabueno, Aleena Mushtaq, Syed Nakib Hossain, Denye N Ogeh, Manoj Pandian Sakthivel, Anne Parker, Malcolm Perry, Ivana Piližota, Daniel Poppleton, Irina Prosovetskaia, Shriya Raj, José G Pérez-Silva, Ahamed Imran Abdul Salam, Shradha Saraf, Nuno Saraiva-Agostinho, Dan Sheppard, Swati Sinha, Botond Sipos, Vasily Sitnik, William Stark, Emily Steed, Marie-Marthe Suner, Likhitha Surapaneni, Kyösti Sutinen, Francesca Floriana Tricomi, David Urbina-Gómez, Andres Veidenberg, Thomas A Walsh, Doreen Ware, Elizabeth Wass, Natalie L Willhoft, Jamie Allen, Jorge Alvarez-Jarreta, Marc Chakiachvili, Bethany Flint, Stefano Giorgetti, Leanne Haggerty, Garth R Ilsley, Jon Keatley, Jane E Loveland, Benjamin Moore, Jonathan M Mudge, Guy Naamati, John Tate, Stephen J Trevanion, Andrea Winterbottom, Adam Frankish, Sarah E Hunt, Fiona Cunningham, Sarah Dyer, Robert D Finn, Fergal J Martin, and Andrew D Yates. Ensembl 2024. *Nucleic Acids Research*, 52(D1):D891–D899, 11 2023. doi: 10.1093/nar/gkad1049.
56. Jingcheng Du, Peilin Jia, YuLin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, 20(1):82, Feb 2019. doi: 10.1186/s12864-018-5370-x.
57. Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.
58. Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12): 3005–3018, August 2020. doi: 10.14778/3415478.3415530.
59. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
60. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 448–456, 2015.
61. Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021.
62. Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
63. Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam Margolin, Sungjoon Kim, Christopher Wilson, Joseph Lehar, Gregory Kryukov, Dmitriy Sonkin, Anupama Reddy, Manway Liu, Lauren Murray, Michael Berger, John Monahan, Morais Paula, Jodi Meltzer, Adam Korejwa, Judit Jané-Valbuena, and Levi Garraway. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483:603–607, 03 2012. doi: 10.1038/nature11003/.
64. Francesco Iorio, Theo Knijnenburg, Daniel Vis, Graham R Bignell, Michael Menden, Michael Schubert, Nanne Aben, Emanuel Gonçalves, Syd Barthorpe, Howard Lightfoot, Thomas Cokelaer, Patricia Greninger, Ewald van Dyk, Han Chang, Heshani de Silva, Holger Heyn, Xianming Deng, Regina K Egan, Qingsong Liu, and Mathew Garnett. A landscape of pharmacogenomic interactions in cancer. *Cell*, 166, 07 2016. doi: 10.1016/j.cell.2016.06.017.
65. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.
66. Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
67. Geoffrey Hinton. Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012. Coursera: Neural Networks for Machine Learning.
68. Yixuan Qiu, Jiebiao Wang, Jing Lei, and Kathryn Roeder. Identification of cell-type-specific marker genes from co-expression patterns in tissue samples. *Bioinformatics*, 37(19):3228–3234, 04 2021. doi: 10.1093/bioinformatics/btab257.
69. Hannah A Pliner, Jay Shendure, and Cole Trapnell. Supervised classification enables rapid annotation of cell atlases. *Nature methods*, 16(10):983–986, 2019.
70. Ze Zhang, Danni Luo, Xue Zhong, Jin Huk Choi, Yuanqing Ma, Stacy Wang, Elena Mahrt, Wei Guo, Eric W Stawiski, Zora Modrusan, et al. Scina: a semi-supervised subtyping algorithm of single cells and bulk samples. *Genes*, 10(7):531, 2019.