Cell2Sentence: Teaching Large Language Models the Language of Biology

Daniel Levine [*1] **Sacha Lévy** [*1] **Syed Asad Rizvi** [*1] **Nazreen Pallikkavaliyaveetil** [1] **Xingyu Chen** [1] **David Zhang** [1] **Sina Ghadermarzi** [1] **Ruiming Wu** [2] **Zihe Zheng** [1] **Ivan Vrkic** [3] **Anna Zhong** [1] **Daphne Raskin** [1] **Insu Han** [1] **Antonio Henrique de Oliveira Fonseca** [1 4] **Josue Ortega Caro** [1 4 5] **Amin Karbasi** [1] **Rahul M. Dhodapkar** [6] **David van Dijk** [1 4 7]

Abstract

We introduce Cell2Sentence (C2S), a novel method to directly adapt large language models to a biological context, specifically single-cell transcriptomics. By transforming gene expression data into "cell sentences," C2S bridges the gap between natural language processing and biology. We demonstrate cell sentences enable the fine-tuning of language models for diverse tasks in biology, including cell generation, complex cell-type annotation, and direct data-driven text generation. Our experiments reveal that GPT-2, when fine-tuned with C2S, can generate biologically valid cells based on cell type inputs, and accurately predict cell types from cell sentences. This illustrates that language models, through C2S fine-tuning, can acquire a significant understanding of single-cell biology while maintaining robust text generation capabilities. C2S offers a flexible, accessible framework to integrate natural language processing with transcriptomics, utilizing existing models and libraries for a wide range of biological applications.

1. Introduction

Large language models (LLMs) such as GPT have demonstrated powerful capabilities in natural language processing tasks including question answering, summarization, and text generation (Vaswani et al., 2017; Radford et al., 2018; Devlin et al., 2019; OpenAI, 2023; Touvron et al., 2023; Anil et al., 2023). However, their application to complex fields like biology, especially in single-cell transcriptomics, poses a novel challenge. Traditional methods in this domain, largely reliant on specialized neural network architectures, do not leverage the potential of LLMs' pretrained knowledge and linguistic understanding.

In this paper, we introduce Cell2Sentence (C2S), a method designed to adapt LLMs for transcriptomics. C2S converts single-cell gene expression data into textual sequences by rank-ordering gene names in descending order of expression levels. This formatting enables LLMs to process and interpret this information (see Figures 1 and 2) while also maintaining the richness and complexity in single-cell data (Figures 3 and 4). Another advantage of C2S is that it takes advantage of the highly optimized and user-friendly open source libraries for transformer models such as Hugging Face (Wolf et al., 2020).

This approach not only allows the models to generate biologically relevant cells and predict cell types but also facilitates the generation of descriptive natural language text from single-cell data. This work demonstrates that combining C2S with LLMs significantly enhances their performance in transcriptomic tasks, notably outperforming models trained solely on C2S.

In summary, our key contributions are:

1. Introducing Cell2Sentence, an *effective method* for representing single-cell data as text sequences.
2. Fine-tuning language models on cell sentences to generate cells, predict combinatorial cell labels, and interpret single-cell data in natural language.
3. Providing a *simple and modular framework* for adapting language models to transcriptomics using popular language model tools and libraries.

In the following sections, we detail the C2S data transformation, model fine-tuning, and evaluate our fine-tuned models

[*]Equal contribution [1]Department of Computer Science, Yale University, New Haven, CT, USA [2]School of Engineering Applied Science, University of Pennsylvania, Philadelphia, PA, USA [3]School of Computer and Communication Sciences, Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland [4]Department of Neuroscience, Yale School of Medicine, New Haven, CT, USA [5]Wu Tsai Institute, Yale University, New Haven, CT, USA [6]Keck School of Medicine, University of Southern California, Los Angeles, CA, USA [7]Department of Internal Medicine (Cardiology), Yale University, New Haven, CT, USA. Correspondence to: Rahul M. Dhodapkar <rahul.dhodapkar@med.usc.edu>, David van Dijk <david.vandijk@yale.edu>.

Copyright 2024 by the author(s).

on various biological tasks. We conclude with a discussion on the broader implications and future directions for merging natural language processing with transcriptomic data analysis through representation learning. We plan to open-source our software and cell sentence datasets.

2. Background and Related Work

2.1. Large Language Models

LLMs have transformed natural language processing, demonstrating versatility in tasks ranging from text classification to text generation. Pioneering architectures such as LSTM (Hochreiter & Schmidhuber, 1997), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2020) have excelled in text classification, while models like LlaMA-2 (Touvron et al., 2023) and Falcon (Almazrouei et al., 2023) have advanced question answering. Text generation has seen remarkable strides with T5 (Raffel et al., 2020), GPT-3 (Brown et al., 2020), and BART (Lewis et al., 2020). For an extensive exploration of LLMs' evolution, refer to (Zhao et al., 2023).

2.2. Single-Cell Foundation Models

Parallel to LLMs, deep learning in single-cell transcriptomics has progressed significantly. Models like NeuCA (Li & Feng, 2023), ACTINN (Li & Feng, 2023), and scVI (Lopez et al., 2018) have been instrumental in cellular annotation. Tools such as scGen (Lotfollahi et al., 2019) and SAUCIE (Amodio et al., 2019) have addressed batch effect removal, while scVI and DeepImpute (Arisdakessian et al., 2019) have pioneered data imputation. Centralized repositories like GEO (Barrett et al., 2012), CellxGene (Megill et al., 2021), and HCA (Regev et al., 2017) further propelled the field, leading to the development of foundation models like scGPT (Cui et al., 2023b) and Geneformer (Theodoris et al., 2023).

2.3. Prompt Fine-Tuning

Prompting, a technique that has been popularized since the introduction of GPT-2 (Radford et al., 2019), has become key to eliciting specific behaviors from LLMs (Lester et al., 2021; Gao et al., 2021; Li & Liang, 2021). The availability of datasets such as Alpaca (Taori et al., 2023a) and tools like Flan (Wei et al.; Longpre et al., 2023; Chung et al., 2022), along with parameter-efficient tuning methods (Hu et al., 2022), has enabled the customization of LLMs for specific tasks. A comprehensive survey of these methods is available in (Liu et al., 2023).

2.4. Multimodal Training and Cross-Modality Encoding

Our C2S approach uniquely integrates multimodal learning by transforming single-cell data into text before it is embed (Baltrušaitis et al., 2018; Barua et al., 2023), diverging from traditional schemes that encode modalities separately. This strategy parallels the visual language modeling of (Wu et al., 2007), but uniquely applies to single-cell transcriptomics using modern embedding techniques, marking a novel direction in multimodal machine learning.

3. Methods

Cell2Sentence transforms single-cell expression data into sentences of gene names rank ordered by decreasing transcript abundance as shown in Figure 2.

3.1. Data transformation

Single-cell RNA sequencing produces transcript count matrices that represent the genetic profiles of individual cells. Most current computational models in single-cell biology handle data in $\mathbb{R}^{c \times n}$, posing scalability challenges with larger datasets. We propose transforming expression matrices into gene sequences as a solution to enable the use of LLMs (Cui et al., 2023a; Hou & Ji, 2023) and other transformer-based architectures (Yang et al., 2022) for single-cell data analysis.

Let C denote a cell by gene count matrix with n rows and k genes, with $C_{i,j}$ denoting the number of RNA molecules observed for gene j in cell i. We filter cells with fewer than 200 genes expressed and genes expressed in less than 200 cells. Cell-wise quality control metrics are then calculated based on mitochondrial gene counts using the Scanpy Python library (Wolf et al., 2018). Cells with over 2500 counts or more than 20% mitochondrial transcript counts are excluded. The count matrix is then row-normalized (summing to 10,000) and log-normalized (Haque et al., 2017), yielding the preprocessed transcript count matrix C'. We summarize this normalization step as:

$$C'_{i,j} = \log_{10}\left(1 + 10^4 \times \frac{C_{i,j}}{\sum_{j=1}^{k} C_{i,k}}\right) \tag{1}$$

We denote the rank-order transformation applied on C' as S, and the sequence of gene names resulting from $S(C_i)$ as cell sentence s_i for each cell i in the preprocessed count matrix. In practice, we apply the preprocessing and rank-order transformation S on each individual single-cell dataset, providing a flexible process for converting traditional single-cell gene expression count matrices to cell sentences.

While genes are not intrinsically ordered in transcript matrices, their expression patterns have been shown to follow inverse-rank frequency patterns (Furusawa & Kaneko, 2003;

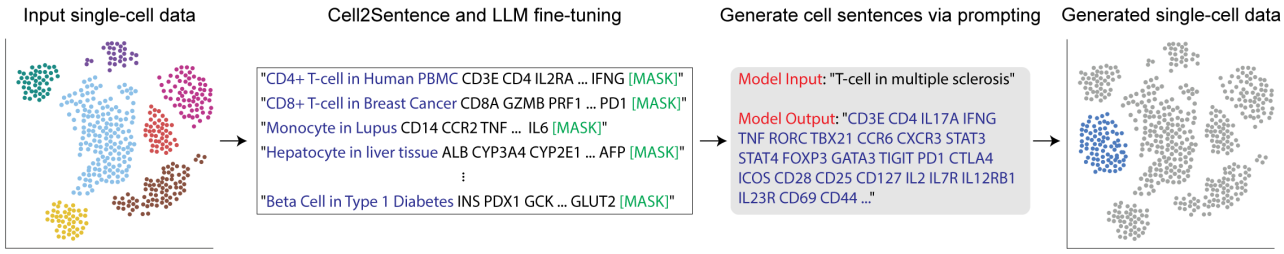

Figure 1. Overview of the Cell2Sentence framework. Input single-cell data, including metadata, are converted into cell sentences for language model fine-tuning. At inference, new cell sentences are generated and can be converted back to gene expression space.

Qiu et al., 2013), thus establishing a steady relationship between a gene's expression level within a cell and its rank among the genes expressed in that cell. We model this inverse-rank relationship with a log-linear distribution and approximate it in log-log space using a linear regression (Dhodapkar, 2022).

Given single-cell dataset which underwent rank-order transformation S, let r_i denote the log of the rank of gene i in C', and e_i the original expression of gene i. We fit a linear model to predict e_i from r_i during the initial conversion to cell sentence format, resulting in a fitted slope and intercept value which are saved for each converted dataset (see Figures 7). The linear model has form $e_i = a_d \times r_i + b_i$, given dataset d and $\{a_d, b_d\} \in \mathbb{R}^2$.

We postprocess generated cell sentences by ignoring invalid gene names and averaging the rank of duplicate genes. The fitted linear model is then applied to the log-rank of generated genes to convert back to expression. Any gene not present in a cell sentence is considered to have zero expression. We define the average rank of a generated gene g_i^{gen} belonging to the set of unique genes $G_U \subseteq S$ as follows:

$$r_i^{\text{gen}} = \frac{1}{|G|} \sum_{j=1}^{|G|} \text{rank}(g_j^{\text{gen}}) \tag{2}$$

where $G = \{g_1^{\text{gen}}, g_2^{\text{gen}}, \ldots, g_n^{\text{gen}}\} \subseteq S$ is the set of duplicate generated genes for g_i^{gen}, and r_i^{gen} denotes the average rank of gene g_i^{gen} in the generated cell sentence. This yields the following formulation for expression value vector for the generated cell

$$e_i^{\text{gen}} = \begin{cases} a_d \times \log(r_i^{\text{gen}}) + b_d & \text{if } g_i^{\text{gen}} \in G \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In practice, we consider a global dictionary of all gene names seen in single-cell datasets, which dictates the size of the resulting gene expression vector of the cell.

Next, we consider the robustness of the C2S transformation when converting cells to sentences and reverting back to expression.

3.2. Transformation robustness

We find that transforming expression data to cell sentences is a robust and reversible operation, and cells converted to text and back to expression incur minimal information loss, with over 81% of the variation in the gene expression being captured by the linear regression on an immune tissue dataset (see Figures 3 and 4). Relationships between normalized gene expression and rank on a variety of human tissue datasets are shown in Figures 6 and 7 (see Appendix A). As the linear model only requires the log-rank of a gene to approximate its expression level, any gene sequence can be converted to expression, including generated cell sentences.

3.3. Tasks

The objective of this work is to train language models to generate single-cell data and derive biological insights from single-cells in natural language. We summarize the tasks implemented in our experiments (see Section 4):

- Generate cell sentences: C2S models are trained to generate gene sequences from prompts, optionally conditioned on additional metadata.
- Predict cell labels: biological experiments often involve combinatorial labels (e.g., patient and sample metadata) which C2S models can learn to predict from cell sentences directly in text.
- Derive natural language insights: C2S analyzes single-cell data to extract high-level information on expression dynamics by pairing relevant natural language with cell sentences.

Examples of prompts and responses for each of these tasks are shown in Figure 5.

4. Experiments

In this section we benchmark Cell2Sentence on conditional cell generation, combinatorial cell label prediction, and ab-

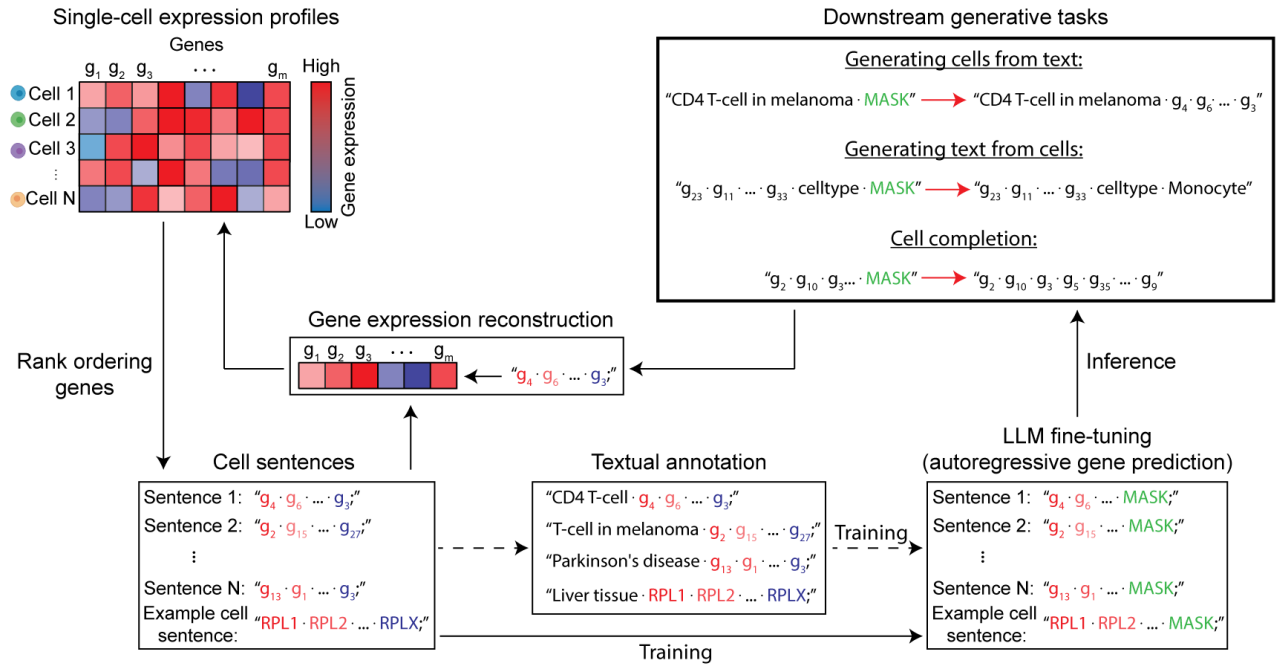

Figure 2. Detailed overview of the Cell2Sentence framework. Single-cell gene expression profiles are transformed into cell sentences through expression rank ordering of gene names. These cell sentences can be annotated with biological metadata, including cell type, tissue, or disease. Subsequently, language models are fine-tuned using the cell sentences. At inference, cell sentences are generated both conditionally (e.g., given a cell or tissue type) and unconditionally, and natural language is produced from text (e.g., cell type label, abstract summary). The generated cell sentences can then be converted back into gene expression profiles.

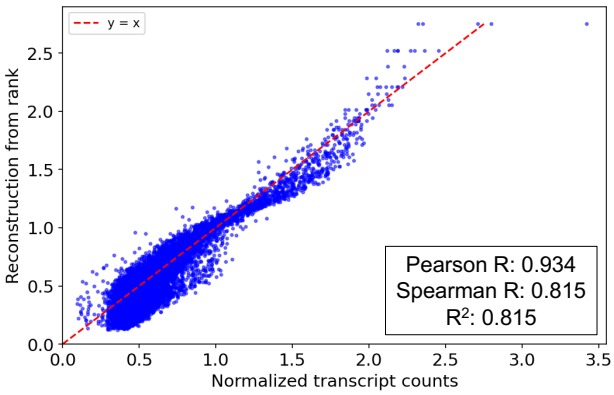

Figure 3. Expression reconstruction from rank using a linear regression.

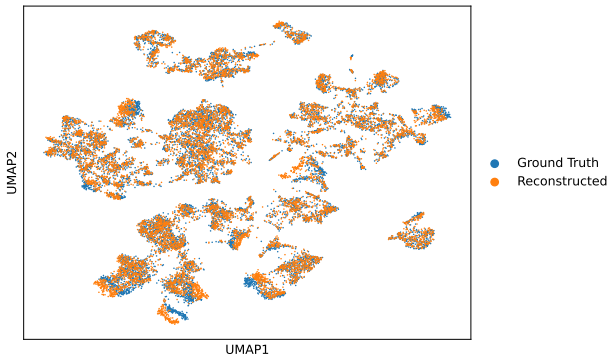

Figure 4. UMAP of ground-truth expression (blue) and reconstructed expression (orange) from cell sentences overlaid.

stract generation from cell sentences. These experiments are conducted using models fine-tuned either with an immune tissue dataset (Domínguez Conde et al., 2022) or with a large scale multi-tissue dataset (Megill et al., 2021). In both cases, we fine-tune GPT-2 (small, medium and large) (Radford et al., 2019) using cell sentences truncated to 100 genes due to resource constraints. We also fine-tune Pythia-160m (Biderman et al., 2023), which is based on the GPT-NeoX architecture (Black et al., 2022) and uses rotary embeddings (Su et al., 2024). For the latter, we set the model's maximum input sequence length to 9200 tokens during fine-tuning, permitting manipulation of full cell sentences. We structure our experiments by first fine-tuning language models on large

Conditional Cell Generation
Prompt: Generate the 100 highest expressed genes listed in descending order for a long-lived plasma cell
Response: **MT-V1 RPS9 [...] RPS9 RPL8**

Cell Label Prediction
Prompt: Identify the cell type most likely associated with these 100 highly expressed genes listed in descending order: DIF3 RPS11 [. . .] RPP4 RPS22
Response: **The cell type corresponding to these genes is a CD4-T cell.**

Natural Language Insights from Data
Prompt: Extract and summarize biological insights given this cell's 100 highest expressed genes: MALAT1 VIM [. . .] RPL10A S100A6
Response: **This study used single-nucleus RNA sequencing to comprehensively map the cell types in the adult human brain.**

Figure 5. Example of Cell2Sentence prompts and responses for generating cell sentences from text, predicting complex natural language labels, and generating biological insight from a single cell sentence.

single-cell datasets, and optionally continuing training for downstream evaluation tasks (see Appendix B.2).

4.1. Fine-Tuning Datasets

We focus our experiments on two datasets with extensive natural language metadata and labels, allowing to leverage the capabilities of base models.

Immune tissue (Domínguez Conde et al., 2022) proposes a large human immune tissue single-cell dataset with cell type annotations. After transformation, we obtain 273,502 cell sentences, each paired with one of 35 cell type labels. We hold out 20% of cell sentences for validation (10%) and testing (10%). We derive three tasks for this dataset:

- Unconditional cell generation (generate a random cell without any specified cell label).
- Conditional cell generation (generate a cell sentence given a specific cell type).
- Cell type prediction (predict the cell type in natural language given a cell sentence prompt).

We create 20 prompt templates to embed cell sentences and labels in natural language.

Multi-tissue (Megill et al., 2021) provides access to hundreds of human and mouse single-cell datasets. We select 99 human single-cell datasets and convert each of them to cell sentences (see Section 3), yielding a total of 37M cells (including 19 held out studies, representing 2.7% of all cell sentences). Every cell sentence is paired with a tissue label derived from the study's metadata (e.g., the tissue label for a dataset containing "Brain" and "Liver" cells will be "Brain, Liver"). We find a total of 11 unique tissues, and 42 unique tissue combinations across this dataset. Additionally, we generate synthetic abstract summaries to augment original abstracts for the multi-tissue dataset (see Appendix B.1). We derive the following five tasks for this multi-tissue dataset:

- Tissue type prediction and conditional generation (similarly to the previous cell type prediction and conditional generation prompts).
- Generate abstract summaries from cell sentences (prompted with a cell sentence, the model generates an abstract summary for the corresponding study).
- Generate a cell sentence given an abstract summary (similar to generating from tissue or cell type, but instead leveraging natural language from abstract summaries).

We also include an unconditional generation prompt template, where models are only prompted to generate a cell sentence.

4.2. Experiment 1: conditional cell generation

Objective: Accurate generation of different cell types is crucial for generative approaches on single-cell data, as it enables downstream analysis. Our aim is to establish the quality of various cell generation methods by using distributional and classification metrics. We show that C2S models outperform state-of-the-art baselines in generating synthetic single-cell data given cell type conditions.

Methodology: We train C2S on the immune tissue dataset from (Domínguez Conde et al., 2022). Cell sentences are appended to text prompts indicating its cell type in order for the model to learn conditional cell generation (Figure 5). Our approach was compared against several established generative single-cell methods, including scVI, scGen, scDiffusion, and scGPT (Lopez et al., 2018; Lotfollahi et al., 2019; Luo et al., 2024; Cui et al., 2023a). For scGen, we randomly split our training data in half and artificially set one group as *control* and the other as *stimulation*, and we randomly pair cells of the same cell type. scGPT's unique

conditional generation method was described in (Cui et al., 2023a), but no code for this method was made publicly available as of this writing. We mimic their method by adding an embedding layer for cell type labels and add the embedding to all tokens while generating random samples of 1000 genes until 36000 genes are generated. A held out test dataset of 500 cells per cell type (17,500 cells in total) with 36,503 genes was used to evaluate all models. We run our experiments 5 times using 5 different samples of our test dataset and report the mean score and standard deviation. Further details on training and data are supplied in Appendix B.3.

Evaluation: The performance of C2S was quantitatively evaluated using k-NN (K-Nearest Neighbors) accuracy and Gromov-Wasserstein distance, measured against the original ground truth cells. The k-NN classifier was fit on the test dataset, and we used the values $k = 3, 5, 10$, and 25. Gromov-Wasserstein distance was used to compare the similarity and structural alignment between generated outputs, which is not always captured by k-NN accuracy due to its local nature. Overall, our evaluation metrics give a well-rounded assessment of each model's ability to generate biologically plausible cells.

Results: As shown in Table 1, our C2S model outperformed all other methods. Notably, our C2S-trained Pythia-160m model outperforms the foundation model scGPT and also scDiffusion, which requires over 900 million parameters in its standard implementation for the immune tissue dataset. The performance of scVI and scGEN was similar due to their similar architectures, both being variational autoencoders. Our results show that causal language modeling can be adapted to single cell generation and outperform other commonly used generative architectures.

4.3. Experiment 2: cell label prediction

Objective: The second experiment evaluate C2S's capacity for cell label classification with complex, combinatorial cell labels involing multiple metadata information about cells. We hypothesize that the natural language generation capabilities of C2S from pretraining will transfer well to complex label classification compared to baseline methods.

Methodology: We compared C2S against several single-cell analysis methods, including two state-of-the-art (SOTA) single-cell foundational models, scGPT (Cui et al., 2023a) and Geneformer (Cui et al., 2022), as well as two non-single cell baseline classifiers. To evaluate the generalization ability of the model to out-of-distribution data, we utilized two bulk datasets - L1000 (Subramanian et al., 2017) and GTEx (Consortium, 2020) - as benchmark datasets, which follow a different data distribution compared to the single-cell datasets which comprised the pretraining datasets for each foundation model. We also evaluate all models on a human PBMC single-cell dataset treated with various combinatorial cytokine stimulations (Dong et al., 2023), as a third benchmark dataset.

For each dataset, metadata associated with the cell - cell type, tissue, drug perturabtions, dosage information, stimulations, etc. - was gathered to create information-rich multi-part labels for each cell/bulk sample in each dataset. Predicting combinations of labels is expected to be a challenging task, particular in a limited data scenario. We constrained the number of training samples for each dataset to roughly give a few samples per combinatorial class, resulting in a dataset where there are few examples per class.

Evaluation: We report the classification accuracy and Area Under the Receiver Operating Characteristic Curve (AUROC) metrics across 3 experiment repeats on each dataset, as reported in Table 2. Exact combinatorial label matching was calculated for each dataset, in addition to another set of metrics giving models partial credit for getting individual parts of the combinatorial label correct.

Results: As indicated in Table 2, C2S outperformed baseline methods in both exact and partial label accuracy, demonstrating its capability in complex label prediction through natural language. The experiment particularly highlights C2S's adaptibility to out-of-distribution data, which was not present anywhere in its training distribution, while also learning to predict complex multi-part labels.

4.4. Experiment 3: abstract summary generation

Objective: We demonstrate the ability for C2S models to generate meaningful text carrying biological insight given a single cell sentence. Additionally, we show that C2S performance cannot be achieved by large language models without fine-tuning despite their extensive pretraining.

Methodology: We use GPT-2 small fine-tuned on cell sentences truncated to the top-100 most expressed genes from the multi-tissue dataset (see Appendix 4.1). At inference, we prompt C2S models with a natural language prompt and a single truncated cell sentence sampled from one of 19 held out evaluation studies. We generate abstract summaries with 30 cell sentences sampled from each evaluation study. This approach allows for a direct comparison of the generated abstracts with the ground-truth abstracts from associated evaluation publications. We benchmark against large language models in a 10-shot prompting setting. Each baseline model is prompted with a cell sentence and tasked with producing biological insights from the cell as an abstract summary (see the full prompt in Appendix E.1). We compare our approach against OpenAI's GPT-3.5-Turbo-1106, as well as Mistral-

Table 1. Results on immune cell conditional generation quality. For each measure we sample 500 cells with replacement from each cell type in our held out immune dataset for comparison. The k-nearest neighbors (k-NN) classifier is fit on the held out immune cells with their cell types used as labels. The true label of a generated cell is the cell type used for its conditional generation. Gromov-Wasserstein (GW) distance is measured between all generated and all held out cells. Our full cell generation C2S model outperforms all models.

MODEL	k-NN (↑) 3	5	10	25	GW (↓)
SCGEN	0.2376 ± 0.0112	0.2330 ± 0.0093	0.2377 ± 0.0053	0.2335 ± 0.0041	315.9505 ± 1.2431
SCVI	0.2436 ± 0.0062	0.2400 ± 0.0064	0.2425 ± 0.0034	0.2348 ± 0.0032	302.1285 ± 0.9338
SCDIFFUSION	0.2335 ± 0.0125	0.2288 ± 0.0111	0.2368 ± 0.0067	0.2306 ± 0.0049	72.0208 ± 0.3937
SCGPT	0.1838 ± 0.0086	0.1788 ± 0.0169	0.1811 ± 0.0149	0.1882 ± 0.0071	2989.8066 ± 4.9229
C2S (PYTHIA-160M)	$\mathbf{0.2588 \pm 0.0061}$	$\mathbf{0.2565 \pm 0.0060}$	$\mathbf{0.2746 \pm 0.0073}$	$\mathbf{0.2715 \pm 0.0070}$	$\mathbf{54.3040 \pm 0.3410}$

Table 2. Experimental results on downstream cell label classification. Cell labels are composed of multiple combinatorial metadata parts, including cell type, perturbations, and dosage information. Accuracy and area under ROC curve is computed on model predictions versus ground truth combinatorial labels, with partial credit given for partial misclassifications.

	MODEL	CYTOKINE STIMULATION ACC	AUROC	L1000 ACC	AUROC	GTEX ACC	AUROC
PARTIAL LABEL	k-NN CLASSIFIER	0.462 ± 0.0047	0.550 ± 0.0064	0.592 ± 0.0054	0.740 ± 0.0036	0.492 ± 0.0047	0.662 ± 0.0035
	XGBOOST	0.515 ± 0.0175	0.631 ± 0.0202	0.389 ± 0.0066	0.534 ± 0.0044	0.482 ± 0.0127	0.633 ± 0.0120
	GENEFORMER	0.600 ± 0.0170	0.722 ± 0.0145	0.419 ± 0.0153	0.632 ± 0.0181	0.500 ± 0.0013	0.649 ± 0.0025
	SCGPT	0.419 ± 0.0001	0.500 ± 0.0000	0.334 ± 0.0076	0.500 ± 0.0000	0.270 ± 0.0522	0.500 ± 0.0000
	C2S (GPT-2 LARGE)	$\mathbf{0.639 \pm 0.0049}$	$\mathbf{0.767 \pm 0.0049}$	$\mathbf{0.631 \pm 0.0031}$	$\mathbf{0.768 \pm 0.0021}$	$\mathbf{0.575 \pm 0.0035}$	$\mathbf{0.713 \pm 0.0014}$
FULL LABEL	k-NN CLASSIFIER	0.060 ± 0.0044	0.509 ± 0.0013	0.168 ± 0.0041	0.583 ± 0.0020	0.112 ± 0.0027	0.500 ± 0.0001
	XGBOOST	0.087 ± 0.0088	0.528 ± 0.0050	0.124 ± 0.0032	0.556 ± 0.0018	0.107 ± 0.0080	0.500 ± 0.0003
	GENEFORMER	0.134 ± 0.0161	0.555 ± 0.0085	0.102 ± 0.0288	0.549 ± 0.0146	0.123 ± 0.0005	0.500 ± 0.0000
	SCGPT	0.049 ± 0.0002	0.500 ± 0.0000	0.026 ± 0.0001	0.500 ± 0.0000	0.006 ± 0.0028	0.500 ± 0.0000
	C2S (GPT-2 LARGE)	$\mathbf{0.149 \pm 0.0057}$	$\mathbf{0.564 \pm 0.0030}$	$\mathbf{0.202 \pm 0.0059}$	$\mathbf{0.600 \pm 0.0029}$	$\mathbf{0.152 \pm 0.0062}$	$\mathbf{0.574 \pm 0.0032}$

7B-Instruct (Jiang et al., 2023) and Mixtral-8x7B-Instruct (Jiang et al., 2024) quantized to 4-bit precision with AWQ (Lin et al., 2023). We also benchmark the performance of GPT-2 small and large pretrained checkpoints without C2S fine-tuning. The pretrained GPT-2 models are prompted in a zero-shot fashion due to their small contexts—the models are unable to accommodate the 4,000 tokens required for the 10-shot prompting setting (more details about the evaluation can be found in Appendix E).

Evaluation: Generated summaries are embedded and evaluated for pairwise similarity relative to the ground-truth abstracts (Xiao et al., 2023). We compute the mean pairwise cosine similarity between evaluation studies to test whether generated abstracts are significantly correlated to their ground-truths, more so than to other original abstracts. We apply the T and KS test on the mean cosine similarities. We also compute embedding distribution correlations (Pearson's, Spearman's) and distances (MMD, Wasserstein) between generated and original abstracts.

Results: We find that C2S was more adept at generating abstracts that closely align with the ground truth, compared to baseline large language models. Table 3 shows that only C2S is able to generate differentiated abstracts given a prompt cell sentence and large context. Additionally, C2S-generated summaries lie closer to the held out study abstracts in embedding space by a 50% margin on MMD (see Table 3). This experiment brings empirical evidence for the utility of cell sentences to derive biological insights from single-cell data. C2S learns to associate relevant language (e.g., tissue type, condition) to complex gene sequences, going beyond capabilities acquired during pretraining. We present a qualitative comparison of generated abstract summaries in Appendix Figure 10.

5. Discussion

Cell2Sentence is a novel approach for training large language models using single-cell transcriptomics data, converting gene expression profiles into sequences of text called *cell sentences*. This method involves ranking gene names by their expression levels to create a reversible encoding of biological data with minimal loss of information. Language models fine-tuned on these cell sentences outperform other foundation models on embedding tasks and generative tasks. Cell sentences, which can be integrated with textual annotations, are versatile for generation and summarization tasks, benefiting from natural language pretraining. We show pretrained language models trained on combinations of text

Table 3. Experimental results on abstract summary generation. This table displays the outcomes of statistical analyses (T-test, KS test) to evaluate the mean cosine similarities between embeddings of generated abstracts and their respective original abstracts (where $i = j$) as well as with different original abstracts (where $i \neq j$). Additionally, it details the Maximum Mean Discrepancy (MMD) and Wasserstein distance (W) comparisons of the embeddings from generated abstracts against those of original abstracts. The results indicate that C2S significantly surpasses baseline methods by achieving a higher correlation with the original abstracts' embeddings. Furthermore, the embeddings from C2S-generated summaries exhibit closer alignment to those of the original abstracts compared to baseline approaches.

MODEL	T TEST (↑)	KS TEST (↑)	MMD (↓)	W (↓)
C2S (GPT-2 SMALL)	$\mathbf{2.96}, \boldsymbol{p = 0.003}^*$	$\mathbf{0.35}, \boldsymbol{p = 0.023}^*$	$\mathbf{0.198 \pm 0.004}$	$\mathbf{0.414 \pm 0.006}$
C2S (GPT-2 LARGE)	$\mathbf{2.49}, \boldsymbol{p = 0.013}^*$	$\mathbf{0.35}, \boldsymbol{p = 0.020}^*$	$\mathbf{0.202 \pm 0.003}$	$\mathbf{0.421 \pm 0.004}$
GPT-3.5-TURBO-1106	$1.23, p = 0.220$	$0.21, p = 0.392$	0.298 ± 0.004	0.490 ± 0.008
MIXTRAL-8X7B-INSTRUCT AWQ	$-1.20, p = 0.233$	$0.24, p = 0.246$	0.639 ± 0.016	0.544 ± 0.005
MISTRAL-7B-INSTRUCT	$-8.64, p = 0.384$	$0.23, p = 0.299$	0.754 ± 0.010	0.584 ± 0.004
GPT-2 SMALL	$1.31, p = 0.896$	$1.52, p = 0.783$	1.045 ± 0.009	0.752 ± 0.004
GPT-2 LARGE	$-1.44, p = 0.885$	$1.81, p = 0.581$	0.939 ± 0.006	0.701 ± 0.016

and cell sentences leads to nascent capabilities of drawing insight from data not inherent to large language models. We leave the possibility of exploring the emergent properties of large language models with model and data scale to future work.

Broader Impact

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of Cell2Sentence, particularly in the field of biomedical research. We hope that pretrained language models fine-tuned on single cell sentences will aid biological data generation and processing, uncovering valuable insights through a natural language perspective.

References

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Alhammadi, M., Daniele, M., Heslow, D., Launay, J., Malartic, Q., Noune, B., Pannier, B., and Penedo, G. The falcon series of language models: Towards open frontier models. 2023.

Amodio, M., van Dijk, D., Srinivasan, K., et al. Exploring single-cell data with deep multitasking neural networks. *Nat Methods*, 16:1139–1145, 2019. doi: 10.1038/s41592-019-0576-7.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A. C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., and Wu, Y. Palm 2 technical report, 2023.

Arisdakessian, C., Poirion, O., Yunits, B., et al. Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell rna-seq data. *Genome Biol*, 20:211, 2019. doi: 10.1186/s13059-019-1837-6.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41 (2):423–443, 2018.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C. L., Serova, N., Davis, S., and Soboleva, A. NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1):D991–D995, 11 2012. ISSN 0305-1048. doi: 10.1093/nar/gks1193. URL https://doi.org/10.1093/nar/gks1193.

Barua, A., Ahmed, M. U., and Begum, S. A systematic literature review on multimodal machine learning: Ap-

plications, challenges, gaps and future directions. *IEEE Access*, 11:14804–14831, 2023. doi: 10.1109/ACCESS.2023.3243854.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. Scaling instruction-finetuned language models, 2022.

Consortium, G. The gtex consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509): 1318–1330, 2020.

Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., and Wang, B. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, pp. 2023–04, 2023a.

Cui, H., Wang, C., Maan, H., and Wang, B. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, 2023b. doi: 10.1101/2023.04.30.538439. URL https://www.biorxiv.org/content/early/2023/05/01/2023.04.30.538439.

Cui, Z., Liao, Y., Xu, T., and Wang, Y. Geneformer: Learned gene compression using transformer-based context modeling. *arXiv preprint arXiv:2212.08379*, 2022.

Dao, T. FlashAttention-2: Faster attention with better parallelism and work partitioning. 2023.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Dhodapkar, R. M. Representing cells as sentences enables natural-language processing for single-cell transcriptomics. *bioRxiv*, 2022. doi: 10.1101/2022.09.18.508438. URL https://www.biorxiv.org/content/early/2022/09/19/2022.09.18.508438.

Domínguez Conde, C., Xu, C., Jarvis, L., Rainbow, D., Wells, S., Gomes, T., Howlett, S., Suchanek, O., Polanski, K., King, H., et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376 (6594):eabl5197, 2022.

Dong, M., Wang, B., Wei, J., de O. Fonseca, A. H., Perry, C. J., Frey, A., Ouerghi, F., Foxman, E. F., Ishizuka, J. J., Dhodapkar, R. M., et al. Causal identification of single-cell experimental perturbation effects with cinema-ot. *Nature Methods*, 20(11):1769–1779, 2023.

Eldan, R. and Li, Y. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Furusawa, C. and Kaneko, K. Zipf's law in gene expression. *Physical review letters*, 90(8):088102, 2003.

Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*, 2021.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Haque, A., Engel, J., Teichmann, S. A., and Lönnberg, T. A practical guide to single-cell rna-sequencing for

biomedical research and clinical applications. *Genome medicine*, 9(1):1–12, 2017.

HF Canonical Model Maintainers. gpt2 (revision 909a290), 2022. URL https://huggingface.co/gpt2.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Hou, W. and Ji, Z. Reference-free and cost-effective automated cell type annotation with gpt-4 in single-cell rna-seq analysis. *bioRxiv*, pp. 2023–04, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation, 2021.

Li, Z. and Feng, H. *NeuCA: NEUral network-based single-Cell Annotation tool*, 2023. R package version 1.6.0.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), jan 2023. ISSN 0360-0300. doi: 10.1145/3560815. URL https://doi.org/10.1145/3560815.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Lopez, R., Regier, J., Cole, M., Jordan, M., and Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat Methods*, 15(12):1053–1058, 2018. doi: 10.1038/s41592-018-0229-2.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lotfollahi, M., Wolf, F., and Theis, F. scgen predicts single-cell perturbation responses. *Nat Methods*, 16(8):715–721, 2019. doi: 10.1038/s41592-019-0494-8.

Luo, E., Hao, M., Wei, L., and Zhang, X. scdiffusion: conditional generation of high-quality single-cell data using diffusion model. *arXiv preprint arXiv:2401.03968*, 2024.

Megill, C., Martin, B., Weaver, C., Bell, S., Prins, L., Badajoz, S., McCandless, B., Pisco, A. O., Kinsella, M., Griffin, F., et al. Cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices. *bioRxiv*, pp. 2021–04, 2021.

OpenAI. Gpt-4 technical report, 2023.

Qiu, X., Wu, H., and Hu, R. The impact of quantile and rank normalization procedures on the testing power of gene differential expression analysis. *BMC bioinformatics*, 14: 1–10, 2013.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Regev, A., Teichmann, S. A., Lander, E. S., Amit, I., Benoist, C., Birney, E., Bodenmiller, B., Campbell, P., Carninci, P., Clatworthy, M., Clevers, H., Deplancke, B., Dunham, I., Eberwine, J., Eils, R., Enard, W., Farmer, A., Fugger, L., Göttgens, B., Hacohen, N., Haniffa, M., Hemberg, M., Kim, S., Klenerman, P., Kriegstein, A., Lein, E., Linnarsson, S., Lundberg, E., Lundeberg, J., Majumder, P., Marioni, J. C., Merad, M., Mhlanga, M., Nawijn, M., Netea, M., Nolan, G., Pe'er, D., Phillipakis, A., Ponting, C. P., Quake, S., Reik, W., Rozenblatt-Rosen, O., Sanes, J., Satija, R., Schumacher, T. N., Shalek, A., Shapiro, E., Sharma, P., Shin, J. W., Stegle, O., Stratton, M., Stubbington, M. J. T., Theis, F. J., Uhlen, M., van Oudenaarden, A., Wagner, A., Watt, F., Weissman, J., Wold, B., Xavier, R., Yosef, N., and Participants, H. C. A. M. Science forum: The human cell atlas. *eLife*, 6:e27041, dec 2017. ISSN 2050-084X. doi: 10.7554/eLife.27041. URL https://doi.org/10.7554/eLife.27041.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., Gould, J., Davis, J. F., Tubelli, A. A., Asiedu, J. K., et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171 (6):1437–1452, 2017.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023a.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm.stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023b.

Theodoris, C. V., Xiao, L., Chopra, A., Chaffin, M. D., Al Sayed, Z. R., Hill, M. C., Mantineo, H., Brydon, E. M., Zeng, Z., Liu, X. S., et al. Geneformer: Transfer learning enables predictions in network biology. *Nature*, pp. 1–9, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Wolf, F. A., Angerer, P., and Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing, 2020.

Wu, L., Li, M., Li, Z., Ma, W.-Y., and Yu, N. Visual language modeling for image classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pp. 115–124, 2007.

Xiao, S., Liu, Z., Zhang, P., and Muennighoff, N. C-pack: Packaged resources to advance general chinese embedding, 2023.

Yang, F., Wang, W., Wang, F., Fang, Y., Tang, D., Huang, J., Lu, H., and Yao, J. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. A survey of large language models, 2023.

A. Method Details

We further evaluate the robustness of the Cell2Sentence transformation across 127 single-cell datasets retrieved from (Megill et al., 2021). Among those are the 99 human single-cell datasets introduced in Section 4, as well as 28 mouse single-cell datasets. We quantify the reconstruction performance using a linear regression between log-rank and normalized expression on Pearson's R, Spearman's R and R-squared coefficients (see Figure 6). We also visualize the linear linear between log-rank and normalized expression across 12 human datasets in Figure 7. Interestingly, emerging patterns seem to hold across data samples within tissues (e.g., kidney and blood tissue samples achieve relatively similar reconstruction performance).

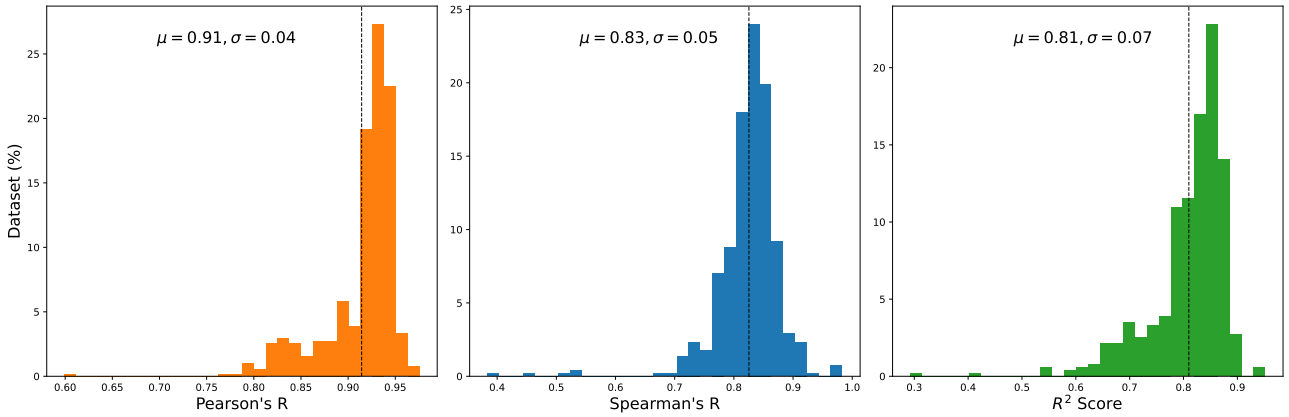

Figure 6. Distributions of Peason's R, Spearman's R and R^2 statistics for linear expression reconstruction from log-rank using across 127 distinct scRNA-seq datasets. Both mean and standard deviation are shown for each score, with the mean represented by a dashed line on the x-axis.

B. Experimental Details

B.1. Training Data Augmentation

We augment the original abstracts of studies in the multi-tissue dataset and generate 5,000 summaries for each abstract with GPT-3.5-Turbo-1106, yielding 495,000 unique abstract summaries (Eldan & Li, 2023; Taori et al., 2023b). The prompt employed to generate the abstracts is presented in Figure 8.

B.2. Evaluation Datasets

Cell label prediction Our evaluation benchmarks model performance across three distinct datasets, incorporating both single-cell and bulk data. Each dataset introduces unique challenges for label classification, stemming from their combinatorial label structures and the presence of out-of-distribution samples:

1. Human PBMC single-cell (Dong et al., 2023): This dataset comprises 250 PBMC cells distributed among 7 cell types, incorporating 10 different stimulations and distinguishing between acute and chronic stimulations. The evaluation set encompasses approximately 120 unique label combinations, challenging the model's ability to navigate sparse label spaces.

2. L1000 (Subramanian et al., 2017): Comprising bulk RNA sequencing data, the L1000 dataset is considered out-of-distribution for our task. It features 2,000 bulk samples, with labels denoting the cell line, drug compound (with 20 possible values), dosage, and perturbation time, leading to a total of 4,000 potential label combinations. This dataset serves to evaluate the model's generalization capabilities across a broad spectrum of biological conditions and experimental interventions.

3. GTEx (Consortium, 2020): Similarly utilizing bulk RNA sequencing data, the GTEx dataset includes 1,000 bulk samples. Labels combine patient age, death condition (such as ventilator use or sudden death), and tissue type,

amounting to around 1,000 distinct label combinations. Through this dataset, the model's ability to deduce complex biological states from significantly diverse data compared to single-cell observations is scrutinized.

B.3. Cell Generation

We train scGPT with sequence lengths of 1200 nonzero genes as described in (Cui et al., 2023a) for 1 epoch. In order for the model to generate zero-expressed genes, we train for 3 more epochs on randomly sampled collections of 2000 genes including zero-expressed genes with the same masking ratios used during the first phase of finetuning. The reported results show the model trained after its first epoch since the model had converged and stayed at the same loss for the remaining epochs. Even on a p4d.24xlarge AWS instance with 8 A100 40GB GPUs, half-precision, and flash attention 2, we found it difficult to fit longer sequences without memory issues. Although Pythia-160m has approximately 3 times as many parameters as scGPT, it was able train on the same instance 4 times faster per epoch despite having 4.5 times more tokens per sample.

To generate cells at inference, we start with 1000 genes from a randomly sampled cell in the training dataset and generate autoregressively using the previously generated 1000 genes as context until 36,000 genes are generated. The cell type label is embedded and added to all other tokens, so the model receives two conditional signals during inference.

C. Training Details

We provide further details on our training configuration for the pretraining for experimental results and fine-tuning settings. We refer to natural language as "NL" and Cell2Sentence as "C2S", and refer to a model that has been solely trained on cell sentences as "C2S", in contrast to a pretrained model fine-tuned on cell sentences which we refer to as "NL + C2S". We use sequence lengths of 1024 for GPT-2 and 9200 for Pythia-160m and train on all tokens. We use the AdamW optimizer (Loshchilov & Hutter, 2017) and flash attention (Dao et al., 2022; Dao, 2023). We find that C2S models largely benefit from natural language pretraining as opposed to starting training from randomly initialized weights as explained below (see Appendix C.1).

C.0.1. Pretraining

The GPT-2 small model is initialized with 12 layers and 768 hidden dimensions, and the medium model with 24 layers and 1024 hidden dimensions, as detailed in (Radford et al., 2019). We employ a learning rate of 6×10^{-4} with a cosine scheduler and 1% warmup ratio. For the GPT-2 medium model, we accumulate gradients over 16 steps. The effective batch sizes for the small and medium models are of 10 and 48 examples. Each model is trained using a single A5000 GPU over two days. Model weights are randomly initialized using a Xavier normal distribution (Glorot & Bengio, 2010).

We train a Byte Pair Encoding (BPE) tokenizer (Sennrich et al., 2015) on the full cell sentence dataset, including NL prompts and cell type labels, yielding a vocabulary of 9,609 tokens. The training set contains approximately 30 million tokens, averaging 740 tokens per example. Due to the smaller embedding space, the initialized models contain slightly fewer parameters than their counterparts pretrained on a vocabulary of 50,257 tokens (93M for the small model and 313M for medium model). The resulting corpus exhibits sparse NL tokens due to short and repetitive prompts. Despite instruction corpora being traditionally used to fine-tune pretrained models for question answering tasks, we adopt this setting during pretraining to mirror our fine-tuning setup described in Section C.0.2. We hypothesize that the semantic variability from prompting patterns might implicitly regularize token and positional embeddings, with natural language tokens acting as class tokens.

We emphasize that the loss is computed on both the prompt and the associated label (i.e. cell type). Not doing so would cause embeddings of the prompt tokens to remain random, impairing the capacity of the model to learn the conditional relations between prompt and label tokens. We evaluate the capacity of our model to generate valid genes and maintain an accurate sequence length (here, of 100 genes) and present the results in Table 7. We find that both pretrained models are able to generate sequences of 100 genes without significantly deviating from the mean. The models also both achieve over 97% and 96% accuracy in gene validity and uniqueness.

C.0.2. Fine-tuning

Models are initialized using pretrained weights retrieved from the Hugging Face model hub (HF Canonical Model Maintainers, 2022). We employ a cosine scheduler. On both models, we accumulate gradients over 16 steps and use batch sizes of

Table 4. Correlation metrics for averaged generated cells per cell type against original expression values. The "random" model was trained from scratch on cell sentences. The "pretrained" model was fine-tuned on cell sentences. Correlation metrics are computed on individual cell types and then averaged.

MODEL	PEARSON R	R^2
GPT-2 SMALL, RANDOM	0.947	0.873
GPT-2 SMALL, PRETRAINED	**0.984**	**0.949**

eight examples (yielding an effective gradient update batch size of 128 examples). Each model is trained using a single A5000 GPU. While we experimented with applying efficient fine-tuning techniques (e.g. LoRA (Hu et al., 2022)), fully fine-tuned models outperformed alternatives in gene uniqueness and validity assessments. We notably found LoRA to yield highly variable generation patterns, with uniqueness of genes in generated sentences as low as 70%. Unlike for our pretraining setup, we apply the instruction fine-tuning task in a classical manner, computing the loss exclusively on labels. We use the pretrained GPT-2 tokenizer, which averages around 233 tokens per training samples (yielding a total of 9M training tokens).

Similarly to the process detailed in Section C.0.1, we examine the coherence of generated output using sequence length, as well as accuracy in gene validity and uniqueness. We find that the fine-tuned model outperform the pretrained models by generating genes with over 99% validity and 98% uniqueness on average (see Table 7). While both models achieve reliable performance by these standard metrics, we conclude that our fine-tuned models are consistently outperforming the pretrained models in generating real human genes, which are only rarely duplicated within cell sentences.

C.1. Comparing fine-tuning and pretraining

We test whether using pretrained GPT-2 weights yields performance improvements over only training from randomly initialized weights on cell sentences. We therefore train two GPT-2 small models on a random subset of 49,920 cell sentences spanning 17 cell types from the immune tissue dataset. To evaluate the ability of our trained models to generate realistic cells, we consider the average generated cell of each of the 17 cell types in our immune tissue dataset, and compare it with the average real cell of each cell type in Table 4. Across 17 different cell types, generated cells from fine-tuned models show high correlation with real cells, capturing over 94% of the variation in the expression of an average cell. We note that initializing a model with a pretrained language model outperforms training from scratch, indicating that there is mutual information which allows a model to better understand cell sentence generation.

We also trained an additional two GPT-2 medium models to assess the performance improvements brought by scaling the parameter count. Table 5 show the *k*-NN accuracy of each model for different number of neighbors. The cell type of cells generated by C2S models are accurately classified with a *k*-NN classifier, achieving a peak of accuracy of 54%. Table 6 underscores the necessity of NL pretraining for accurate cell type identification. A significant performance decline is observed when using models that have not undergone NL pretraining, thereby confirming that the models are not merely memorizing the conditioning text. Despite 1) the limited scope of natural language text in our training prompts relative to the pretraining corpus, and 2) permitting the models to train on these natural language prompts, models without NL pretraining failed to acquire meaningful natural language embeddings. Furthermore, a modest performance increment is observed as the scale of pretrained models increases.

D. Inference Details

At inference, we follow the training procedure highlighted in Section C. We set the hyper-parameters top_p $= 0.9$ and temperature $= 0.7$ to promote diversity in cell generation. For conditional cell generation and unconditional cell generation used in our experiments (see Section 4), we randomly sample prompt templates as input, inserting the cell type or sentence where needed for conditional generation. For autoregressive cell type prediction or abstract generation, we randomly sample templates as in training.

All outputs are generated until an end-of-sequence (EOS) token is predicted. Post-generation, gene and cell type extraction is done using regex to remove prompts. For evaluation, we retain invalid genes and average ranks of duplicate genes, rearranging sequences as needed. When reverting back to expression values, invalid genes are ignored, but the rank values

Table 5. k-NN classification accuracy results against ground truth data. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. k-NN classifier is fitted on ground truth cell sentences and used to predict the cell type label of generated cell sentences from different trained models. k-NN classification is done both in cell sentence space using Levenshtein distance (Lev.), as well as after converting back to expression vectors (Expr.). "Real cells" indicates k-NN classification fit on ground truth cell sentences and used to predict a separate sample of ground truth cell sentences.

MODEL	k=5		k=10		k=25		k=50	
	EXPR.	LEV.	EXPR.	LEV.	EXPR.	LEV.	EXPR.	LEV.
GPT-2 SMALL (C2S)	24.56	16.02	23.79	17.72	23.51	19.17	22.84	19.76
GPT-2 SMALL (NL + C2S)	52.55	37.63	52.19	41.20	51.42	43.42	49.51	44.44
GPT-2 MEDIUM (C2S)	26.36	17.58	25.48	19.18	24.55	20.67	23.44	21.32
GPT-2 MEDIUM (NL + C2S)	**54.67**	**38.60**	**54.52**	**41.34**	**53.27**	**43.93**	**51.80**	**44.73**

Table 6. Quantification of autoregressive cell type prediction on unseen cells. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. These results show test accuracy significantly improves with NL pretraining. The scores are computed on unseen immune tissue test data and weighted by the distribution of labels.

MODEL	ACCURACY	F1	PRECISION	RECALL
GPT-2 SMALL (C2S)	29.33	17.46	13.27	29.33
GPT-2 SMALL (NL + C2S)	69.95	68.56	69.44	69.95
GPT-2 MEDIUM (C2S)	29.17	17.41	13.35	19.17
GPT-2 MEDIUM (NL + C2S)	**74.26**	**73.69**	**74.16**	**74.26**

are preserved, e.g. if an invalid gene appears in position 3 and a valid gene appears in position 4, the invalid gene is ignored, but the valid gene retains a rank of 4.

Utilizing pretrained LLMs offers the advantage of using highly optimized, open-source libraries for inference. Similarly to the training setup, we make use of flash attention (Dao et al., 2022; Dao, 2023) and batched inference to accelerate generation. Inference for GPT-2 medium (345M parameters) can be done on a single A5000 GPU with 24GB of VRAM and a batch size of 100 without running out of memory. For GPT-2 small (117M parameters), the batch size can be increased to 250. We did not determine the exact maximum batch sizes, so these values can likely be increased further. On average, the number of tokens in the prompts and top 100 genes combined was around 350. For example, the GPT-2 small model takes approximate 20 minutes to generate 500 cells from each of the 35 cell types found in the immune tissue dataset. Model quantization was not required but may be useful for future experiments with larger models.

E. Evaluation

E.1. In-Context Learning Abstract Generation

We evaluate the capacity of closed- and open-source LLMs to generate biological insights from cell sentences in a 10-shot prompting setting. Each model is provided with a sequence of pairs of cell sentences and abstract summaries. We evaluate GPT-3.5-Turbo-1106 with the prompt presented in Figure 9. We use an identical prompt for Mixtral-8x7B-Instruct and Mistral-7B-Instruct, while removing the instruction to return JSON format as this is not known to be a feature of these models. This 10-shot prompting approach yields a context length of around 4000 tokens. As GPT-2 small and GPT-2 large are limited to 1024 in-context tokens, we use the same prompt as for the C2S models, in which we only provide a minimal instruction in natural language paired with a cell sentence.

Table 7. Quality of generated outputs. "NL + C2S" means pretrained on natural language and then trained on cell sentences. "C2S" means no pretraining and just trained on cell sentences. This table shows 1. models trained using Cell2Sentence are able to generate real genes with few duplicates and invalid genes and 2. models pretrained with natural language generate more accurately. The metrics are computed across all 35 cell types seen during training with 500 cells generated per cell type and then averaged across all generated cells (top 100 genes) from the immune tissue dataset. The valid genes percentage shows the number of genes generated that are real genes including duplicates. The generated length is the number of genes generated regardless of their validity. The unique gene ratio is the ratio of unique valid genes to the generated length.

MODEL	GEN. LENGTH	VALID GENES %	UNIQUE GENES %
GPT-2 SMALL (C2S)	101.54	97.84	97.31
GPT-2 SMALL (NL + C2S)	99.84	99.60	98.88
GPT-2 MEDIUM (C2S)	100.62	97.51	96.69
GPT-2 MEDIUM (NL + C2S)	99.80	**99.70**	**99.47**

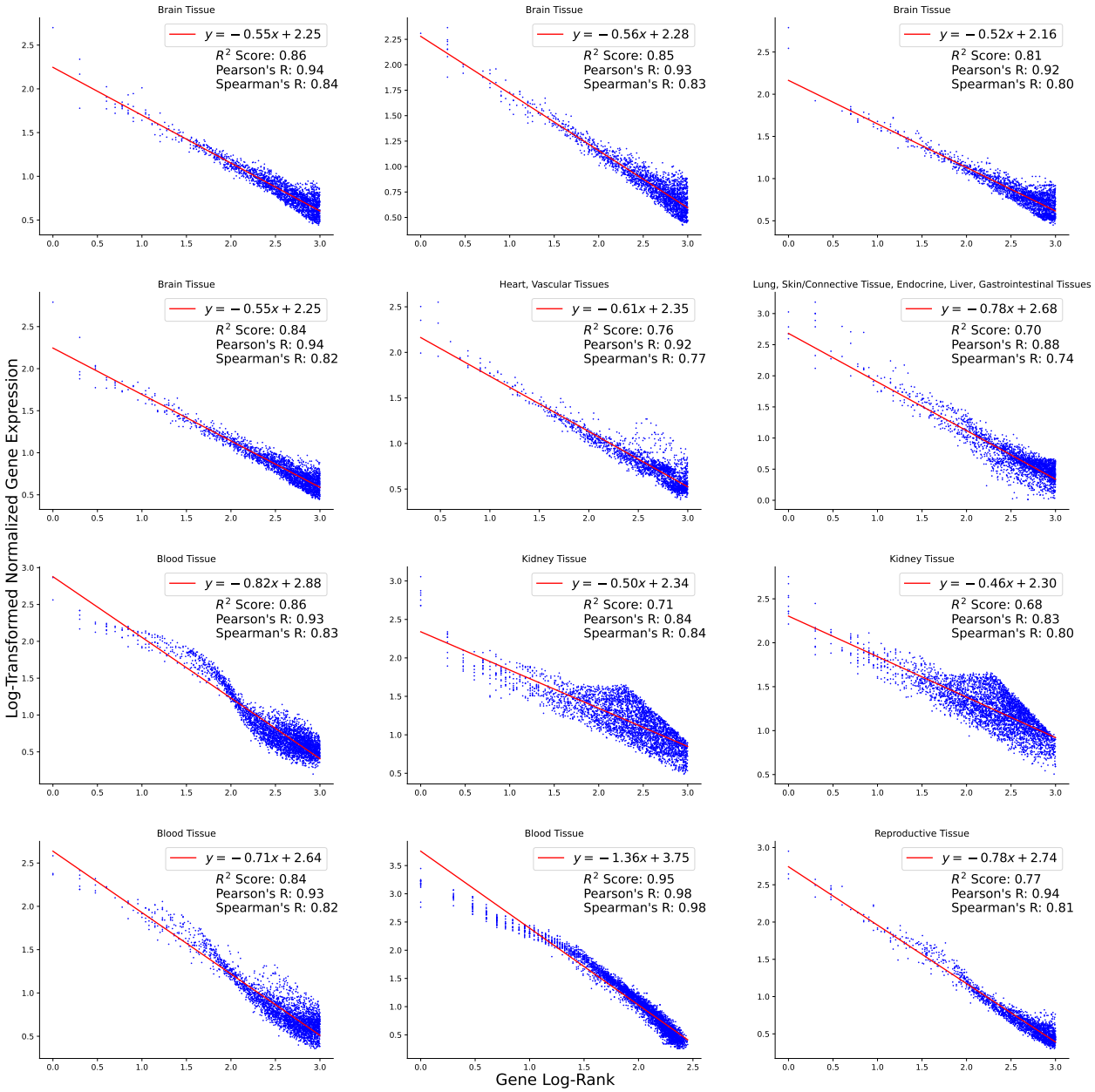

Figure 7. Scatter plots showing the log-linear relationship between gene expression and rank across 12 diverse scRNA-seq datasets. The red line shows the fitted linear model, with Pearson's R, Spearman's R, and R^2 quantifying goodness of fit. The high correlation values demonstrate that gene rank encodes expression in a consistent, reversible way across datasets. This enables translating between the text domain of cell sentences and original gene expression.

You are a helpful assistant and PhD in single-cell genomics. Your task is to concisely summarize abstracts of scientific publications while conveying the key results from the paper. Keep your summary under 70 words.

Abstract: <ABSTRACT>

Figure 8. Prompt employed to generate synthetic abstract summaries using GPT-3.5-Turbo-1106.

You are an AI assistant with specialized expertise in biology, particularly skilled in single-cell transcriptomics data analysis. Your role is to provide concise, insightful summaries of research studies centered on specific cells, emphasizing their most prominent features derived from the top-100 expressed genes in the sampled cell. Utilize this gene expression data to deduce the cell's condition and the type of tissue it originates from. This prompt includes various examples where short abstracts were effectively generated from single-cell data:
context. Deliver your summary in a structured JSON format: 'abstract': <ANSWER>.

Abstract: <ABSTRACT>
Cell sentence: <CELL_SENTENCE>

Generate an abstract for the following cell sentence:
Cell sentence: <CELL_SENTENCE>

Make sure to format your answer in JSON {'abstract': <ANSWER>}.

Figure 9. In-context learning prompt employed to generate abstract summaries from cell sentences with GPT-3.5-Turbo-1106.

A) Fallopian tube (FT) homeostasis requires dynamic regulation of heterogeneous cell populations and is disrupted in infertility and ovarian cancer. Here we applied single-cell RNA-seq to profile 59,738 FT cells from 4 healthy pre-menopausal subjects. The resulting cell atlas contains 12 major cell types representing epithelial, stromal and immune compartments. Re-clustering of epithelial cells identified 4 ciliated and 6 non-ciliated secretory epithelial subtypes, two of which represent potential progenitor pools: one leading to mature secretory cells, while the other contributing to either ciliated cells or one of the stromal cell types. To understand how FT cell numbers and states change in a disease state, we analyzed 17,798 cells from two hydrosalpinx samples and observed shifts in epithelial and stromal populations, and cell type-specific changes in extracellular matrix and TGF- signaling, underscoring fibrosis pathophysiology. This resource is expected to facilitate future studies to understand fallopian tube homeostasis in normal development and disease.

B) Single-cell analysis of postmenopausal ovary and fallopian tube revealed 17 cell clusters in the ovary and 22 in the fallopian tube, defining distinct cell types in each tissue. Notably, ciliated and secretory epithelial cells were in the fallopian tube. This work contributes to the Human Cell Atlas initiative and provides valuable insights into normal ovarian and fallopian tube biology.

C) This study utilized single-cell gene expression analysis to investigate the molecular characteristics of smooth muscle cells present in human vascular tissue. Through comprehensive profiling of gene expression patterns, the researchers identified key genes involved in smooth muscle cell function, including those related to contractility and regulatory processes. These findings contribute to our understanding of the molecular mechanisms underpinning vascular smooth muscle cell physiology.

D) This single-cell transcriptomic analysis of multiple human tissue types revealed the top-100 expressed genes, which include mitochondrial genes such as MT-CO3, MT-CO2, and MT-CO1, metabolic genes like FABP6 and EEF1A1, and genes involved in immune response and cell adhesion, such as B2M, HLA-A, and CLDN3. Notable findings include the expression of genes related to hypoxia response (PRAP1, PHGR1, TFF3, and TKFC) and immune regulation (TNFSF4, CDHR5, and FOS), providing insights into tissue-specific features and potential signaling pathways. The data provides a valuable resource for further investigation into tissue-specific gene expression patterns and disease mechanisms.

Figure 10. Comparison between an original study abstract (A), and abstract summaries generated with GPT-2 small fine-tuned with C2S (B), GPT-3.5-Turbo-1106 (C), and Mixtral-8x7B-Instruct (D). Only the C2S-generated abstract is relevant to the ground-truth study abstract, despite all summaries being semantically correct. Note that this study was held out from the training set.