Université Paris Cité

École doctorale Informatique, télécommunications et électronique de Paris (EDITE) - ED130

Institut Pasteur, Département de Biologie Computationnelle, Machine Learning for Integrative Genomics (ML4IG)

Transformers on single-cell RNA-sequencing data as large cell models

Par Jérémie KALFON

Thèse de doctorat d'Intelligence Artificielle et Décision (IAD)

Dirigée par Laura CANTINI

Et par Gabriel Peyré

Présentée et soutenue publiquement le 25 Mars 2026

Devant un jury composé de :

Charlotte Bunne, Assistant Professor École polytechnique fédérale de Lausanne	Rapportrice
Valentina Boeva, Assistant Professor École polytechnique fédérale de Zurich	Rapportrice
Simona Cristea, Group leader Dana Farber Cancer Institute Harvard	Membre invitée
Thomas Walter, DR Mines Paris, Université Paris Sciences et lettres	Examinateur
Laurent Jacob, DR Sorbonne Université	Examinateur
Laura CANTINI, Chargé de recherche HDR Université Paris Cité	Directrice de thèse
Gabriel Peyré, DR Ecole Normale Supérieure	Directeur de thèse

Résumé

Titre: L'intelligence artificielle sur l'expression génétique par cellule unique comme modèle cellulaire.

Mots clefs: scRNA-seq ; Modèles fondamentaux ; Transformers ; Réseaux de régulation génique ; Apprentissage profond ; Apprentissage zero-shot ; Analyse inter-espèces ; Représentation ; Débruitage ; Évaluation

Résumé: Cette thèse présente des avancées fondamentales dans l'application de l'apprentissage profond basé sur les transformers aux données de séquençage ARN unicellulaire, avec des applications à l'inférence de réseaux de régulation génique et à l'apprentissage de représentations cellulaires. Nous avons développé scPRINT (single-cell PRe-trained Inference of Networks with Transformers), un modèle cellulaire de grande taille entraîné sur plus de 50 millions de cellules permettant l'inférence de réseaux géniques spécifiques à chaque cellule à l'échelle du génome. Grâce à des innovations architecturales majeures—incluant l'encodage génique basé sur les protéines, la tokenisation apprise de l'expression, et l'encodage positionnel génomique—scPRINT surpasse les méthodes existantes sur plusieurs benchmarks tout en démontrant des capacités zero-shot pour la classification de types cellulaires, la correction d'effets de lot, et le débruitage d'expression.

Nous avons introduit Xpressor, une architecture à attention croisée permettant aux modèles fondamentaux d'apprendre à travers les échelles biologiques, des molécules aux tissus. Cette architecture compresse les représentations au niveau génique en vecteurs d'états cellulaires de dimension réduite via une approche de goulot d'étranglement informationnel, tout en permettant l'ajustement fin multi-échelle de modèles de langage protéique utilisant des tâches cellulaires. Xpressor a amélioré la précision de prédiction des types cellulaires ainsi que la qualité des représentations cellulaires par rapport aux architectures standard.

Ces travaux préliminaires ont conduit au développement de scPRINT-2, entraîné sur 350 millions de cellules provenant de 16 organismes—le plus grand modèle fondamental unicellulaire à ce jour. Grâce à une évaluation additive systématique de 42 configurations de modèles, nous avons validé les décisions de conception clés et atteint des performances de pointe en classification zero-shot des types cellulaires, un débruitage supérieur dans tous les contextes, et les meilleurs scores de correction d'effets de lot de sa catégorie. Nous avons démontré une généralisation sans précédent aux modalités de transcriptomique spatiale et aux organismes non vus durant l'entraînement, des capacités de raisonnement contrefactuel, ainsi qu'une analyse de réseaux géniques inter-espèces, ouvrant de nouvelles perspectives.

Ce travail établit des normes pour les modèles fondamentaux unicellulaires à travers une évaluation rigoureuse, des implémentations open-source, et une utilité biologique démontrée. Nous avons publié sept paquets Python, des benchmarks complets, et des outils accessibles à la communauté.

Abstract

Title: Transformers on single-cell RNA-sequencing data as large cell models

Keywords: sc-RNA-seq ; Foundation models ; Transformers ; Gene Regulatory Networks ; Deep Learning ; Zero-shot Learning ; Cross-species Analysis ; Embeddings ; Denoising ; Benchmarking

Abstract: This thesis presents foundational advances in applying transformer-based deep learning to single-cell RNA sequencing data, with applications to gene regulatory network inference and cellular representation learning. We developed scPRINT (single-cell PRe-trained Inference of Networks with Transformers), a large cell model trained on over 50 million cells that enables genome-wide, cell-specific gene network inference. Through novel architectural innovations—including protein-based gene encoding, learned expression tokenization, and genomic positional encoding—scPRINT outperforms existing methods on multiple benchmarks while demonstrating zero-shot capabilities for cell type classification, batch correction, and expression denoising.

We introduced Xpressor, a cross-attention framework enabling foundation models to learn across biological scales, from molecules to tissues. This architecture compresses gene-level representations into lower-dimensional cell-state vectors through an information bottleneck approach, while enabling multi-scale fine-tuning of protein language models using cellular tasks. Xpressor improved cell-type prediction accuracy and embedding quality over standard architectures.

Building on these foundations, we developed scPRINT-2, trained on 350 million cells across 16 organisms—the largest single-cell foundation model to date. Through systematic additive benchmarking of 42 model configurations, we validated key design decisions and achieved state-of-the-art performance with zero-shot cell-type classification accuracy, superior denoising across all contexts, and best-in-class batch integration scores. We demonstrated unprecedented generalization to unseen spatial transcriptomics modalities and organisms, counterfactual reasoning capabilities, and cross-species gene network analysis.

This work establishes new standards for single-cell foundation models through rigorous benchmarking, open-source implementations, and demonstrated biological utility, contributing seven Python packages, comprehensive benchmarks, and accessible tools to the community.

Essentially, all models are wrong,
but some are useful

George E. P. Box - 1976

A Ph.D. is not a scprint,
it is a marathon

Someone

Remerciements

Je tiens à remercier chaleureusement mes collègues du laboratoire ML4IG de l'institut Pasteur ainsi que ceux du centre de science des données de l'ENS et tout particulièrement Jules Samaran, Remi Trimbour et Geert Huizing, pour leur accueil et leur aide durant cette thèse.

Je veux aussi bien sûr remercier Laura Cantini et Gabriel Peyré, mes co-encadrants, pour leur acceuil, leur soutien et leur disponibilité durant ces années. Leur expertise, leurs conseils et leur rigueur scientifique m'ont été d'une aide précieuse dans la réalisation de ce travail. Je suis entêté, je ne suis pas toujours clair, mais ils ont toujours su m'orienter et me guider dans la bonne direction.

Je remercie ma compagne, Juliette Hirsch, pour avoir toujours été à mes côtés ainsi que pour ses conseils, ses questions et son écoute constante. Malgré le stress qui n'a pas toujours fait ressortir le meilleur chez moi, elle a su être le soleil dans mes journées et la patience dans mes doutes.

Je remercie Patrice, Johanna, Marie-Astrid et Emmanuel Kalfon, sans qui je n'en serais pas ici aujourd'hui et qui m'ont conduit, par leur confiance, à faire ce travail. Je remercie Genevieve et Henri Spanjersberg ayant respectivement fait de la recherche en biologie et la même école que moi 50 ans auparavant ont su m'apporter l'intéret pour la science et l'ingérnierie. Je remercie Lucien Kalfon pour m'avoir montré qu'on pouvait rêver à de grandes choses. Finalement, Je tiens à remercier tout particulièrement Monique Obineau, pour son soutien indéfectible, sa bienveillance et son amour inconditionnel, je n'en serais pas là sans elle, son écoute et ses mots.

Je remercie Alex Wolf, Sergei Ribakov, Brice Rafestin et beaucoup d'autres pour leur aide dans le développement informatique des méthodes que je présente ici.

Je remercie les membres de l'institut Pasteur, du département de biologie computationnelle, le hub de bio-informatique, du département de Mathématiques appliquées de l'ENS, ainsi que ceux du supercalculateur Jean Zay sans qui ces méthodes n'auraient pu être entraînées.

Je remercie mes amis, Baptiste Tesson, Oscar Simon, Pier Michele Kaubari, Emile D'allens, Louis Cauquelin, Suzanne Lazarus, Anne-lise Aupetit, Quentin Van Straaten, sans qui je savais d'avance que je n'aurais été capable d'accomplir ce travail.

Je remercie aussi les membres de Nucleate, Whitelab Genomics, Blossom, dot Omics, Biographica et d'autres pour leur intérêt et soutien durant cette thèse.

Personal Motivation

This Ph.D. started relatively late in my career. This chapter explains the personal and professional path that led to this thesis, and the objectives I set for myself when starting it.

0.1 Background to the thesis

0.1.1 The PiPle Project

Many of the opportunities I had after school have been very exciting. Initially, I decided to create a company called PiPle with a friend, Paul Best, who is now a Post-doctoral Researcher at the University of Vienna in Machine Learning for bio-acoustics.

Funnily enough, it was completely unrelated to biology. We worked on creating novel means of communication. We had—and still have—big ideas for improving utterly inadequate messaging apps, emails, and similar tools through machine learning and innovative design. Doing this, we learned a lot about managing complex projects, selling ideas, building large codebases, teamwork, and designing interfaces.

However, we did not gain enough traction from this, and after a year of hard work, we felt the road ahead was paved with too many sacrifices.

0.1.2 The Broad Institute

I passed on Ph.D. opportunities a second time to work at the Broad Institute instead. Having visited the labs, Boston, and Kendall Square, I knew this was the kind of experience I wanted, and Ph.D.s seemed long and cumbersome. At Broad, I worked on many very-high-impact research projects, and I felt I was part of something bigger than myself. I published as the first author and even started my own research projects, which would inform the thesis I am presenting here.

While I still understood that a Ph.D. was the best place to undergo such projects, I was uncertain about the specifics. I also understood the length, harshness, and sometimes arbitrary nature of U.S. Ph.D. programs. I also wanted to continue working on team-based projects and wanted to experience the start-up environment.

0.1.3 Whitelab Genomics

Along with other personal decisions, it led me to return to France and take on the role of team lead for the computational biology group at Whitelab Genomics in Paris.

Figure 1: The Whitelab Genomics team in September 2023, in its future4care offices

At Whitelab, I learned how to build a team and manage people. I learned a lot about what it means to grow companies from 10 to 50 people. I also learned about the biotech industry and how to build and sell such products.

Whitelab had a good mix of expertise in computational biology, machine learning, structural biology, and business development (see Figure 1). While starting the first project there, I significantly enhanced the potential of foundation models for the biotech industry.

From DNA language models to cell foundation models and knowledge-graph-based models, it became clear that they would be the path forward for aggregating sparse, disparate information across many fields of biology and medicine.

0.1.4 Starting the Ph.D.

I was not looking for any other positions and intended to stay at least a few years to assess how we had grown during that time.

However, I was already in contact with Laura Cantini, with whom I had previously discussed Ph.D. projects. At some point, Laura came back to me with this Ph.D. proposal. I spent the better part of a month in a challenging position, thinking about which decision would not become a regret in the future.

There was no perfect time to do this, but it felt like it was now or never. I was also very impressed by the level of various Ph.D. students in the labs of Laura and Gabriel. Seeing people 4 years younger than me already with such a high level of expertise and knowledge was very humbling. Finally, the Ph.D. topic and group were really on point with what I wanted to do. But mostly, my work/life environment was welcoming, surrounded by family, friends, and activities. I knew what I wanted to work on and what I wanted to learn.

Therefore, I decided to start this Ph.D. journey.

0.2 Personal objectives

This is copied from my initial objectives written in my research proposal at the start of the Ph.D.

I had the chance to see many friends completing their Ph.D.s before starting mine. A main mistake I saw during one's Ph.D. is not seeing the time passing by. My goal for this Ph.D. was to be as product-first as I was at Whitelab Genomics. Delivering results quickly & improving until it is publishable. This mistake, thinking "Well, I have 3 years…", is at least partly responsible for the stress, the crash, and the unpreparedness for what some students might experience after the Ph.D. Thus, I plan to give myself a short timeline, knowing I will likely go over. And I will prepare everything around this idea. I will also start to prepare for what is next from the get-go.

To do that best, one needs to take the opportunity of the Ph.D. to make connections with other labs (industry or academic). Moreover, a good piece of advice I have been given is to *know what you want to do and what you don't want to do.* Know what you are here for. Learn to say no. And I learned to say no in the last 4 years. My goal is to work on large models & large datasets, mostly in transcriptomics, and always to go back to first principles and biology. I also know I want to make something useful, create something that can be a stepping stone for others. Something that affects the community. I know that to do that, you have to go the extra mile in terms of development and be honest with yourself about any shortcomings.

Finally, I have been fortunate to become addicted to my work. I like working hard and taking on challenges. But for this to happen, I need to keep enjoying what I am doing. I also wish to have no regrets about this decision. Thus, my final goal is to enjoy it as much as I can.

Figure 2: A view of the Pasteur Institute in Paris, where I did my Ph.D. Adopted from *CIS de l'institut Pasteur* [1].

Contents

List of Figures

List of Tables

Liste des abréviations

AI Artificial Intelligence. The simulation of human intelligence processes by machines, especially computer systems. xxi, 18, 60

AIVC Artificial Intelligence for a Virtual Cell. xxi

AP-MS Affinity Purification Mass Spectrometry. xxi, 38, 125

ARI Adjusted Rand Index. A measure of the similarity between two data clusterings. xxi

ASO Antisense Oligonucleotide. xxi

ASW Average Silhouette Width. A measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). xxi

ATAC-seq Assay for Transposase-Accessible Chromatin using sequencing. A technique used in molecular biology to assess genome-wide chromatin accessibility. xxi, 14, 123

AUPRC Area Under the Precision-Recall Curve. A performance metric for binary classification problems. xxi, 34, 47–49, 51, 107, 125, 155

BERT Bidirectional Encoder Representations from Transformers. A transformer-based machine learning technique for natural language processing pre-training. xxi, 28, 42

BPH Benign Prostatic Hyperplasia. xxi, 35, 41, 42, 55–60, 159

BS-seq Bisulfite Sequencing. The use of bisulfite treatment of DNA before routine sequencing to determine the pattern of methylation. xxi, 14, 17

CAF Cancer-Associated Fibroblast. A cell type within the tumor microenvironment that promotes tumorigenic features. xxi, 57

CCE Categorical Cross Entropy. A loss function used in multi-class classification tasks. xxi

cFM Cell Foundation Model. A foundation model trained on cellular data. xxi, 35, 86, 87, 90, 92, 94, 96, 98

CGT Cell Graph Transformer. xxi

ChIP-seq Chromatin Immunoprecipitation Sequencing. A method used to analyze protein interactions with DNA. xxi, 14, 34, 50–52, 61, 159

CNRS Centre National de la Recherche Scientifique. The French National Centre for Scientific Research. xxi, 39

CRISPR Clustered Regularly Interspaced Short Palindromic Repeats. A family of DNA sequences found in the genomes of prokaryotic organisms such as bacteria and archaea, used in gene editing. xxi, 15, 164

CxG CellxGene. xxi, 34, 36, 41–45, 53, 54

DNA Deoxyribonucleic Acid. Molecule that carries genetic information for the development and functioning of an organism. viii, xxi, 9, 10, 12–14, 17, 18, 35, 39, 45, 61, 86, 97

ECM Extracellular Matrix. xxi, 35, 37, 42, 59, 60, 122

ENCODE Encyclopedia of DNA Elements. A public research project which aims to identify all functional elements in the human genome sequence. xxi, 34, 47–50

EPR Early Precision Ratio. xxi, 34, 36, 47–49, 51, 92

ESM Evolutionary Scale Modeling. Protein language models trained on evolutionary data. xxi, 34–36, 44–46, 48, 87–90, 93–98, 101, 123, 126, 159, 160

FFPE Formalin-Fixed Paraffin-Embedded. xxi

FM Foundation Model. A large machine learning model trained on a vast amount of data at scale that can be adapted to a wide range of downstream tasks. xxi

FSQ Finite Scalar Quantization. A quantization method for latent representations. xxi, 102

FSQ-VAE Finite Scalar Quantization Variational Autoencoder. xxi, 102

GELU Gaussian Error Linear Unit. xxi

GEO Gene Expression Omnibus. A public functional genomics data repository supporting MIAME-compliant data submissions. xxi, 37

GN Gene Network. xvii, xxi, 27, 41–43, 47, 48, 52, 53, 59–61, 125, 126

GNN Graph Neural Network. A class of artificial neural networks for processing data that can be represented as graphs. xxi, 30–32, 37, 126, 132, 133, 147, 160, 161

GPT Generative Pre-trained Transformer. A type of large language model (LLM) and a prominent framework for generative artificial intelligence. xxi

GPU Graphics Processing Unit. A specialized electronic circuit designed to manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. xxi, 20, 43, 60, 97, 108, 119

GRN Gene Regulatory Network. A collection of molecular regulators that interact with each other and with other substances in the cell to govern the gene expression levels of mRNA and proteins. xxi, 11, 26, 27, 30–32, 34, 38, 41, 42, 45, 46, 48, 49, 101, 159

GSEA Gene Set Enrichment Analysis. A computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states. xxi, 47

gwps Genome-wide Perturb-seq. xxi, 38, 52, 125

IB Information Bottleneck. A method for extracting relevant information from an input variable. xxi, 101

iLISI Integration Local Inverse Simpson's Index. A metric to quantify the degree of mixing of datasets in an integrated embedding. xxi

ipTM interface predicted Template Modeling score. xxi, 126

kBET k-nearest-neighbor Batch Effect Test. A metric to quantify batch effects in single-cell RNA-seq data. xxi

KNN K-Nearest Neighbors. A non-parametric method used for classification and regression. xxi

KO Knockout. A genetic technique in which one of an organism's genes is made inoperative. xxi

latex Is a mark up language specially suited for scientific documents. xxi

LLM Large Language Model. A language model notable for its ability to achieve general-purpose language generation and understanding. xxi, 21, 28

lncRNA Long Non-Coding RNA. A type of RNA, defined as being transcripts with lengths exceeding 200 nucleotides that are not translated into protein. xxi, 10

log1p Logarithm of (1 + x). xxi

LoRA Low-Rank Adaptation. A technique for fine-tuning large language models. xxi, 87, 97

LR Learning Rate. A hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. xxi, 145

LSE Log-Sum-Exp. xxi

mFM Molecular Foundation Model. A foundation model trained on molecular data. xxi, 35, 86, 87, 96, 97

MHC Major Histocompatibility Complex. A set of cell surface proteins essential for the acquired immune system to recognize foreign molecules. xxi, 38

miRNA MicroRNA. A small single-stranded non-coding RNA molecule (containing about 22 nucleotides) found in plants, animals and some viruses, that functions in RNA silencing and post-transcriptional regulation of gene expression. xxi, 10

ML Machine Learning. A field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. xxi, 18, 20, 39

MLP Multi-Layer Perceptron. A class of feedforward artificial neural network. xxi, 34, 36, 45, 87, 88, 90, 97, 98, 130, 135–137, 141, 144, 145, 159, 160

MMD Maximum Mean Discrepancy. A kernel-based statistical test used to determine whether two given samples are drawn from the same distribution. xxi, 148, 149

mRNA Messenger RNA. A single-stranded RNA molecule that corresponds to the genetic sequence of a gene, and is read by a ribosome in the process of synthesizing a protein. xxi, 10, 14

MSA Multiple Sequence Alignment. The alignment of three or more biological sequences (protein or nucleic acid) of similar length. xxi, 94

MSE Mean Squared Error. A measure of the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. xxi, 36, 111, 130, 137, 145

MVC Model-View-Controller. xxi

nFM Nucleotide Foundation Model. A foundation model trained on nucleotide sequences. xxi, 35, 86, 87, 94, 96, 97

NGS Next-Generation Sequencing. A high-throughput method used to determine the sequence of nucleotides in DNA or RNA samples. xxi, 13

NLP Natural Language Processing. A subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language. xxi

NMI Normalized Mutual Information. A normalization of the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation). xxi

NN Neural Network. A method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. xxi, 17, 19, 21

NNZ Number of Non-Zeros. xxi, 37, 112, 115, 116, 118, 131, 149, 150, 160

ODE Ordinary Differential Equation. xxi, 46, 61

OR Odds Ratio. xxi

OT Optimal Transport. A mathematical theory that deals with the problem of finding the most efficient way to move objects from one location to another. xxi

PCA Principal Component Analysis. A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. xxi, 93, 120, 132

PE Positional Encoding. A mechanism in Transformers to inject information about the relative or absolute position of the tokens in the sequence. xxi

PEFT Parameter-Efficient Fine-Tuning. Approaches to fine-tune large models with a small number of parameters. xxi

perturb-seq A technique combining CRISPR-based gene perturbations with single-cell RNA sequencing to study gene function and regulatory networks. xxi, 15

pLM Protein Language Model. A language model trained on protein sequences. xxi

PPI Protein-Protein Interaction. The physical contact between two or more protein molecules as a result of biochemical events. xxi, 38, 61

QKV Query, Key, Value. The three matrices used in the attention mechanism of transformers. xxi

RBP RNA Binding Protein. xxi

RNA RiboNucléique Acide. Polymeric molecule essential in various biological roles in coding, decoding, regulation and expression of genes. xxi, 9, 10, 14, 18, 25, 26, 35, 41, 42, 61, 86, 97

RNA-seq RNA Sequencing. A technique used to analyze the transcriptome of gene expression patterns. xxi, 14, 16, 28, 39, 45, 86, 105

rRNA Ribosomal RNA. A type of non-coding RNA which is the primary component of ribosomes, essential to all cells. xxi, 10

scATAC-seq Single-cell ATAC-seq. A method to map chromatin accessibility at the single-cell level. xxi, 17, 41

scFM Single-cell Foundation Model. A foundation model specifically designed for single-cell biology tasks. xxi, 33, 36, 37, 105–115, 119, 123, 126, 127, 138, 139, 142

scGEN Single-cell Generative. A tool for predicting single-cell perturbation responses. xxi, 55

scGPT Single-cell Generative Pretrained Transformer. A foundation model for single-cell biology based on the GPT architecture. xxi, 34, 36, 39, 42, 45–49, 51–53, 55, 98, 113

scIB Single-cell Integration Benchmark. A benchmark for single-cell RNA-seq integration methods. xxi, 34, 36, 37, 55, 92, 95, 107, 111, 118, 120, 122, 151, 155, 156

scPRINT Single-cell PRe-trained INference Transformer. The model developed in this thesis. xvii, xxi, 34–39, 41–60, 87–89, 91, 92, 96, 99, 101, 105, 106, 108–127, 143–147, 155, 159, 160

scRNA-seq Single-Cell RNA-Sequencing: Method to measure the RNA content of a cell. xxi, 14, 30, 31, 41, 45, 46, 50, 54–56, 58, 60, 61, 117, 137, 152, 159, 163

scVI Single-cell Variational Inference. A probabilistic framework for analyzing single-cell RNA sequencing data. xxi, 55, 99

SEM Structural Equation Modeling. xxi, 34, 46, 47, 49, 51, 52

SGD Stochastic Gradient Descent. An iterative method for optimizing an objective function with suitable smoothness properties. xxi, 20, 21

siRNA Small Interfering RNA. A class of double-stranded RNA non-coding RNA molecules, typically 20-24 base pairs in length, similar to miRNA, and operating within the RNA interference (RNAi) pathway. xxi, 10

SMILES Simplified Molecular Input Line Entry System. A specification in the form of a line notation for describing the structure of chemical species using short ASCII strings. xxi, 86, 97

SOTA State of the Art. xxi, 34, 41–43, 46, 52–55

TF Transcription Factor. A protein that controls the rate of transcription of genetic information from DNA to messenger RNA, by binding to a specific DNA sequence. xxi, 10, 34, 41, 45–52, 142

tFM Tissue Foundation Model. A foundation model trained on tissue data. xxi, 35, 86, 87, 96

TME Tumor Microenvironment. The environment around a tumor, including the surrounding blood vessels, immune cells, fibroblasts, signaling molecules and the extracellular matrix. xxi, 35, 42, 57, 60, 159

tRNA Transfer RNA. An RNA molecule that helps decode a messenger RNA (mRNA) sequence into a protein. xxi, 10

UCE Universal Cell Embedding. A foundation model for single-cell biology. xxi, 45

UMAP Uniform Manifold Approximation and Projection. A dimension reduction technique that can be used for visualization similarly to t-SNE, but also for general non-linear dimension reduction. xxi, 108, 117, 120, 122

VAE Variational Autoencoder. A type of artificial neural network used to learn efficient data codings in an unsupervised manner. xxi, 17, 37, 87, 91, 102, 120, 121, 136, 143, 144, 146, 148

VQ-VAE Vector Quantized Variational Autoencoder. A generative model that learns discrete latent representations. xxi, 102

W2 Wasserstein-2 Distance. xxi

XPEFT XPressor-based Parameter-Efficient Fine-Tuning. xxi, 37, 113

ZINB Zero-Inflated Negative Binomial. A distribution used to model count data that has an excess of zero counts. xxi, 36, 88, 99, 100, 107, 111, 117, 129, 136, 137, 145

Résumé en Français

La cellule constitue l'unité fondamentale du vivant, et la compréhension de ses mécanismes de régulation représente un enjeu majeur de la biologie moderne. Les réseaux de régulation génique (GRN) modélisent les interactions moléculaires qui gouvernent l'expression des gènes au sein de chaque cellule, influençant sa différenciation, sa réponse aux stimuli et son homéostasie. L'inférence précise de ces réseaux à partir de données transcriptomiques unicellulaires demeure toutefois un défi considérable en biologie computationnelle, en raison de l'explosion combinatoire des interactions possibles, du bruit inhérent aux mesures, et de la rareté des ground truths validées expérimentalement. Dans ce contexte, les modèles fondamentaux basés sur les transformers offrent une approche prometteuse pour apprendre des représentations cellulaires riches à partir de grandes quantités de données.

Cette thèse présente des avancées fondamentales dans l'application de l'apprentissage profond basé sur les transformers aux données de séquençage ARN unicellulaire (scRNA-seq). Elle s'articule autour de trois publications principales, chacune abordant un aspect complémentaire : l'inférence de réseaux de régulation génique, l'apprentissage multi-échelle entre modèles biologiques, et la validation systématique des choix architecturaux pour les modèles fondamentaux unicellulaires ainsi qu'un modèle qui bat l'état de l'art sur de nombreuses tâches de biologie cellulaire.

Chapitre 1 : scPRINT — Pré-entraînement sur 50 millions de cellules pour l'inférence de réseaux géniques

Le premier chapitre présente scPRINT (single-cell PRe-trained Inference of Networks with Transformers), un modèle cellulaire de grande taille conçu pour l'inférence de réseaux géniques spécifiques à chaque cellule à l'échelle du génome. scPRINT a été entraîné sur plus de 50 millions de cellules provenant de la base de données CellxGene, représentant environ 80 milliards de tokens couvrant plusieurs espèces, maladies et origines ethniques. L'objectif initial de cette thèse était d'utiliser des réseaux de neurones graphiques (GNN) pour améliorer l'inférence de GRN. Cependant, nous avons rapidement constaté que les GNNs ne constituaient pas l'approche optimale : l'absence de structures graphiques connues a priori et leurs propriétés de mise à l'échelle limitées nous ont conduits à adopter les transformers, qui peuvent être vus comme des GNN opérant sur des graphes entièrement connectés.

Le modèle introduit plusieurs innovations architecturales majeures. Premièrement, un encodage génique basé sur les protéines utilisant les embeddings du modèle de langage protéique ESM2, ce qui réduit le nombre de paramètres tout en permettant la généralisation inter-espèces. Deuxièmement, une tokenisation apprise de l'expression génique via des perceptrons multicouches (MLP), qui surpasse les méthodes de discrétisation manuelles utilisées par les modèles existants. Troisièmement, un encodage positionnel génomique qui capture les patterns de co-régulation entre gènes chromosomiquement proches. Le pré-entraînement repose sur trois tâches complémentaires : une tâche de débruitage par suréchantillonnage de transcrits, une tâche d'apprentissage par goulot d'étranglement pour la compression et la reconstruction d'embeddings cellulaires, et une tâche de prédiction de labels avec classification hiérarchique pour obtenir des embeddings cellulaires disentangled représentant différentes facettes phénotypiques.

Une contribution critique est notre méthode d'extraction de réseaux géniques spécifiques à chaque cellule à partir des matrices d'attention du transformer, inspirée d'approches similaires dans ESM2 pour la prédiction de contacts protéiques. Nous avons rendu cette approche extensible pour calculer des réseaux à l'échelle du génome pour des milliers de cellules sur du matériel grand public. Nous avons également introduit un mécanisme de sélection des têtes d'attention, où un sous-ensemble de têtes est choisi en fonction de leur corrélation avec des réseaux de vérité terrain connus, améliorant significativement la qualité des réseaux inférés.

Pour remédier au manque d'évaluation standardisée dans le domaine, nous avons créé BenGRN et GRnnData, des suites de benchmarking pour l'inférence de GRN utilisant plusieurs types de validations : des réseaux basés sur la littérature (Omnipath), intersections ChIP-seq/perturb-seq spécifiques aux types cellulaires, et données perturb-seq à l'échelle du génome. Nos résultats démontrent que scPRINT surpasse toutes les autres méthodes — incluant scGPT, Geneformer v2, DeepSEM et GENIE3 — sur la majorité des benchmarks. Sur le benchmark Omnipath couvrant 26 types cellulaires, scPRINT a retrouvé 67% plus de connexions que GENIE3 et a montré un enrichissement supérieur pour les facteurs de transcription et leurs cibles validées par ENCODE (20% des facteurs de transcription avec enrichissement significatif, contre 0% pour scGPT).

Au-delà de l'inférence de réseaux, scPRINT démontre des performances compétitives même sans ré-entraînement sur des tâches orthogonales : débruitage d'expression (égalant les méthodes de l'état de l'art comme MAGIC et KNNsmoothing2, et les surpassant sur les types cellulaires rares) ; classification de types cellulaires (62% de précision sans ré-entraînement sur plus de 200 types cellulaires, surpassant les méthodes basées sur les gènes marqueurs comme CellTypist) ; et correction d'effets de lots (obtenant des scores compétitifs sans utiliser de labels de lot). Ces capacités émergent naturellement de l'apprentissage de représentations cellulaires de qualité. L'application à un atlas de 83 000 cellules de tissu prostatique normal et pré-cancéreux (HBP) a permis d'identifier des marqueurs précoces du microenvironnement tumoral dans des lymphocytes B à mémoire, ainsi que des hubs de gènes différentiellement présents dans les fibroblastes associés au HBP. Une étude approfondie de ces réseaux révèle des voies interconnectées liant l'échange ionique, le remodelage de la matrice extracellulaire, le stress oxydatif et l'inflammation chronique : des signatures d'états prémalins. Ce chapitre a conduit à une publication dans Nature Communications.

Chapitre 2 : Xpressor — Vers des modèles fondamentaux apprenant à travers les échelles biologiques

Le deuxième chapitre présente Xpressor, un cadre architectural et d'entraînement permettant l'apprentissage inter-échelle entre modèles fondamentaux biologiques. Ce travail adresse une limitation importante : les modèles fondamentaux existant à différentes échelles biologiques — molécules, séquences, cellules, tissus — fonctionnent de manière isolée, incapables d'exploiter le flux d'information entre les échelles. Nous proposons que le vocabulaire de chaque échelle peut être vu comme une représentation compressée de l'échelle inférieure : les acides aminés sont composés d'atomes, les gènes sont définis par des acides aminés, les cellules par l'ensemble de leurs gènes, et les tissus par l'organisation de leurs cellules.

L'architecture Xpressor introduit un mécanisme de compression basé sur l'attention croisée qui transforme les représentations de haute dimension en vecteurs de dimension réduite. Des blocs transformer additionnels effectuent une attention croisée entre les embeddings de sortie d'un modèle fondamental et un ensemble de tokens latents appris, créant un goulot d'étranglement qui compresse m tokens de dimension d_c en n tokens cellulaires de dimension d_t, où $n \ll m$ et $d_t < d_c$. Ce même transformer peut ensuite décompresser ces représentations par attention croisée avec des tokens de position (ou d'identité). De plus, l'espace latent du goulot est régularisé par des pertes contrastives et des classificateurs spécifiques à chaque token, garantissant que chaque token capture des informations biologiques distinctes.

Notre seconde contribution est une méthode d'ajustement de modèles d'échelle inférieure lors de l'entraînement de modèles d'échelle supérieure. Nous démontrons ceci en utilisant ESM2 (modèle de langage protéique) comme modèle d'échelle inférieure et scPRINT comme modèle d'échelle supérieure. Plutôt que d'utiliser des embeddings ESM2 figés comme tokens géniques, nous ajoutons un adaptateur MLP entraînable qui transforme chaque embedding protéique durant le pré-entraînement de scPRINT.

Les résultats empiriques montrent des améliorations substantielles : la précision de prédiction des types cellulaires augmente de 0,60 à 0,72 (+20%) avec l'architecture Xpressor par rapport au pooling standard, et la qualité des embeddings s'améliore de 0,48 à 0,52 (+8%). L'ajustement multi-échelle améliore également la prédiction des types cellulaires de 0,60 à 0,70 (+17%) et l'inférence de réseaux géniques sur le benchmark Omnipath de 2,0 à 2,4 EPR (+20%). Ce chapitre a conduit à un poster à ICML 2025 et est actuellement en révision pour publication dans Bioinformatics Advances.

Chapitre 3 : scPRINT-2 — Vers la nouvelle génération de modèles fondamentaux cellulaires et de benchmarks

Le troisième chapitre présente scPRINT-2, un modèle fondamental unicellulaire de nouvelle génération dont les décisions de conception ont été systématiquement validées via un cadre

d'évaluation additive sans précédent. Ce travail comble une lacune critique : alors que de nombreux modèles fondamentaux unicellulaires ont été proposés, l'importance relative de leurs choix architecturaux, de leurs stratégies d'entraînement et de leurs modalités de données n'avait auparavant jamais été rigoureusement évaluée de manière isolée.

Nous avons conçu un benchmark complet évaluant 42 configurations différentes de composants, certaines variantes étant entraînées avec différentes initialisations pour générer des bornes d'erreurs statistiques. Cette évaluation a révélé plusieurs résultats clés : le débruitage est supérieur au masquage comme tâche de pré-entraînement ; l'expression non normalisée surpasse l'entrée normalisée ; les tokens géniques basés sur ESM surpassent les embeddings appris de zéro ; l'encodage de localisation génomique améliore la convergence du modèle ; la perte MSE surpasse la ZINB en moyenne, mais une perte hybride ZINB+MSE offre le meilleur équilibre ; et la taille du modèle corrèle avec l'amélioration de l'inférence de réseaux et de la prédiction de types cellulaires.

Pour l'entraînement, nous avons assemblé le plus grand corpus unicellulaire à ce jour : plus de 350 millions de cellules provenant de 16 organismes eucaryotes couvrant plus d'un milliard d'années d'évolution, intégrant des données de CellxGene, du jeu de données Tahoe-100M et de la base scBasecount (20 000 jeux de données GEO retraités), totalisant 25 To de données uniques avec environ 400 000 gènes distincts et 4 764 labels cellulaires différents répartis en 140 000 groupes. Nous avons démontré que la diversité des états cellulaires et la qualité des données importent davantage que le simple nombre de cellules : réduire à 200 jeux de données humains n'a causé qu'une perte minimale de performance, tandis que l'utilisation exclusive de jeux de données à faible diversité a provoqué un effondrement des performances. Nous avons introduit des stratégies d'échantillonnage pondéré par cluster et par qualité pour traiter efficacement ces données hétérogènes.

scPRINT-2 incorpore 12 innovations distinctes validées par notre benchmark, incluant l'architecture XPressor, un encodeur d'expression basé sur GNN exploitant l'information de voisinage, l'attention criss-cross (un mécanisme d'attention sous-quadratique inspiré des Recurrent Interface Networks), une compression basée sur VAE avec pertes de dissimilarité entre tokens cellulaires, et une perte de classification hiérarchique actualisée pénalisant les prédictions en fonction de la distance ontologique plutôt que de la simple exactitude binaire.

Sur le benchmark Open Problems, scPRINT-2 a atteint 75% de précision en classification zero-shot de types cellulaires, surpassant scPRINT-1 (47%) et tous les autres modèles fondamentaux zero-shot (40–60%). Avec notre méthode XPEFT, scPRINT-2 a surpassé toutes les méthodes supervisées et non supervisées existantes sur cette plateforme. Pour le débruitage d'expression, scPRINT-2 est devenu état de l'art, surpassant MAGIC dans tous les contextes testés. Pour l'intégration de lots, les performances zero-shot de scPRINT-2 ont dépassé toutes les autres méthodes.

La capacité de généralisation du modèle au-delà de sa distribution d'entraînement constitue un résultat particulièrement marquant. Sur des données de transcriptomique spatiale Xenium (une modalité absente de l'entraînement), scPRINT-2 a réussi à débruiter l'expression, à imputer 5 000 gènes non observés avec des scores de corrélation comparables aux gènes débruités, et à produire des prédictions biologiquement significatives. Sur des tissus pulmonaires de chat et de tigre (organismes non vus durant l'entraînement), scPRINT-2 a

atteint une précision suffisante sur 500 types cellulaires pour parfois corriger les annotations d'experts, avec une précision atteignant 95% après ajustement fin via XPEFT. L'architecture XPressor a également permis la génération contrefactuelle : en remplaçant les embeddings cellulaires spécifiques à l'organisme de cellules de souris par des embeddings humains, nous avons généré des profils d'expression « humanisés » avec un enrichissement de 58% dans les gènes différentiellement exprimés correctement prédits. Nous avons en outre montré que des embeddings géniques biologiquement significatifs — enrichis pour des voies biologiques connues plutôt que pour de simples valeurs d'expression — n'émergent qu'avec un support architectural explicite pour la compression et la reconstruction. Ce chapitre a conduit à une prépublication actuellement en révision à Nature Methods.

Autres contributions et perspectives

Au-delà de ces trois chapitres, cette thèse a donné lieu à de nombreuses contributions additionnelles. Nous avons publié sept paquets Python en open source : scPRINT et scPRINT-2 (modèles et scripts d'entraînement), scDataloader (chargement efficace de milliers de jeux de données unicellulaires avec échantillonnage pondéré sur des milliards d'éléments), BenGRN (benchmark pour l'inférence de GRN), GRNNdata (manipulation conjointe de réseaux et de données unicellulaires), Xpressor (reproduction des expériences du chapitre 2), Simpler Flash (mécanismes d'attention efficaces incluant notre attention flash criss-cross), et Hierarchical Classifier (classification hiérarchique pour données unicellulaires). Des efforts d'accessibilité importants ont été réalisés, incluant des tutoriels sur Google Colab, le déploiement sur le hub de modèles Chan–Zuckerberg et la plateforme Superbio.ai, ainsi qu'une implémentation Docker pour le benchmarking sur la plateforme Open Problems. Des efforts de vulgarisation ont été menés via des articles pour l'Institut Pasteur et le CNRS, des vidéos YouTube, plus de 25 présentations invitées, et cinq posters dans des conférences internationales.

En perspective, les principaux défis identifiés concernent la diversité et la qualité des données (diversité génétique, technologies de séquençage plus profondes), l'intégration de modalités supplémentaires (ATAC-seq, protéomique, transcriptomique spatiale) et de données de perturbation à l'échelle des modèles fondamentaux. À terme, l'état de l'art se dirigera vers la construction d'un modèle cellulaire virtuel piloté par l'IA, combinant pré-entraînement multi-échelle, apprentissage par renforcement avec retour actif, et raisonnement via des modèles de langage, dans une boucle « laboratoire dans la boucle » reliant expériences in silico et in vitro.

Background

This chapter provides the background needed to understand the contributions of this thesis. We begin with the motivations for modeling the cell, then introduce the molecular biology of gene regulation, the sequencing technologies that made large-scale cellular measurements possible, the computational tasks that have emerged from these data, and finally the artificial intelligence methods we build upon.

In the mid-17th century, Robert Hooke made a groundbreaking discovery while observing a piece of cork through his microscope. He observed structures that he named "cells" and, as a result, marked the beginning of cellular biology [6]. Cells have since been identified as life's fundamental structural and functional units, and biologists have endeavored to map the diverse cell types that comprise multicellular organisms. Additionally, they have sought to understand the transient cell states that occur during development, disease progression, and tissue regeneration [7].

0.1 The promises of cellular biology

Understanding cells at a fundamental level opens the door to transformative applications.

0.1.1 Drug Design

Before developing a drug for a disease, one must understand the disease and identify a potential target gene or set of target genes. It refers to the genes in specific cell types that need to be reactivated, deactivated, or modified to address the disease's underlying mechanism.

But drugs don't have to be small molecules. CAR-T cell therapies have revolutionized blood cancer treatment by modifying a patient's own immune cells to fight the cancer (see Figure 3). Similar approaches could be developed for many other conditions [8]. Here, the drug becomes a cell.

Helping create these cellular drugs, as well as more classic ones, is one of the possible applications of the work we will present in this thesis.

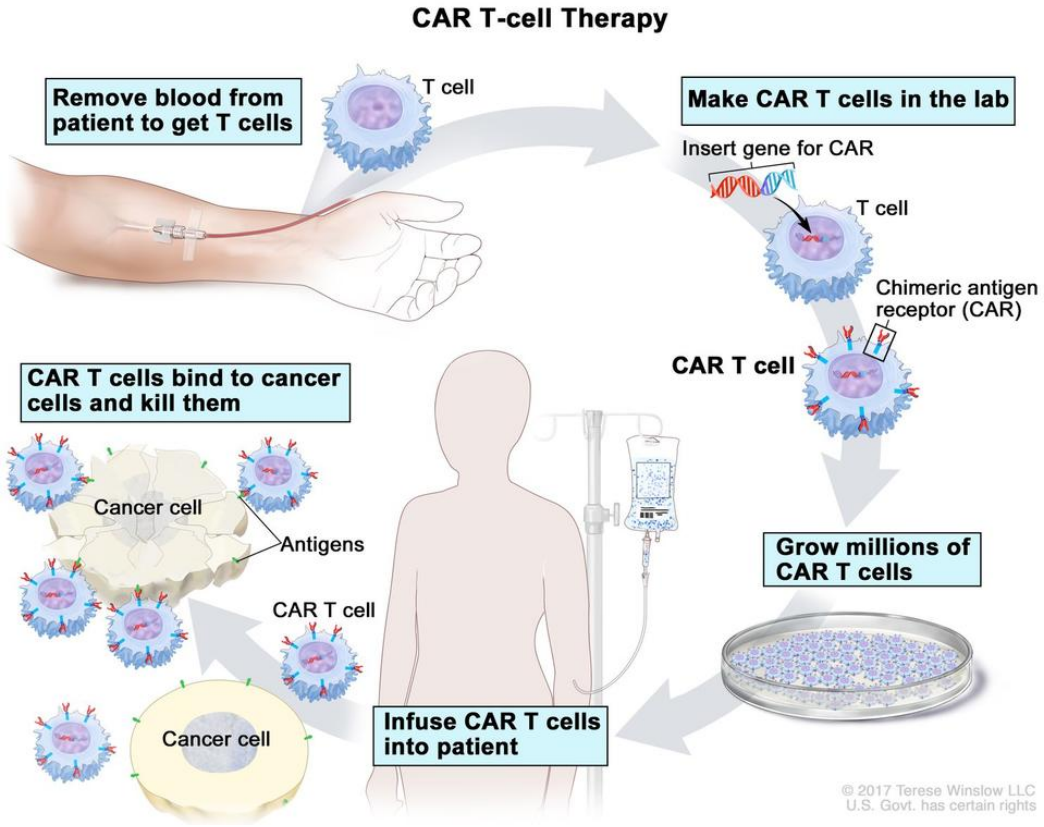


Figure 3: CAR-T cell therapy. Illustration of the CAR-T cell therapy process. It is one example of cell therapy. Adopted from Winslow LLC [9]

0.1.2 Other Applications

But life is everywhere, and cellular engineering has already helped us make better crops, create synthetic meat, and design fungi that remove pollution. When people think about nanorobots, they should think about engineered cells instead [10]. Finally, Richard Feynman famously said, "What I cannot create, I do not understand." Therefore, the modeling of the cell stands as a key milestone in cellular biology, and, indeed, one cannot fulfill the aforementioned promises without a correct cellular blueprint.

This realization has driven hundreds of companies, from tech to bio, and dozens of institutes to pursue efforts to create virtual cellular models [5]. These virtual cells aim to simulate and predict cellular behavior computationally, enabling *in silico* experimentation before costly wet-lab validation.

0.2 GRN and the Cell

To build such a cell model, we must first understand the biological machinery they aim to represent. In this section, we review the key molecular components of the cell, focusing on the regulatory mechanisms that govern gene expression.

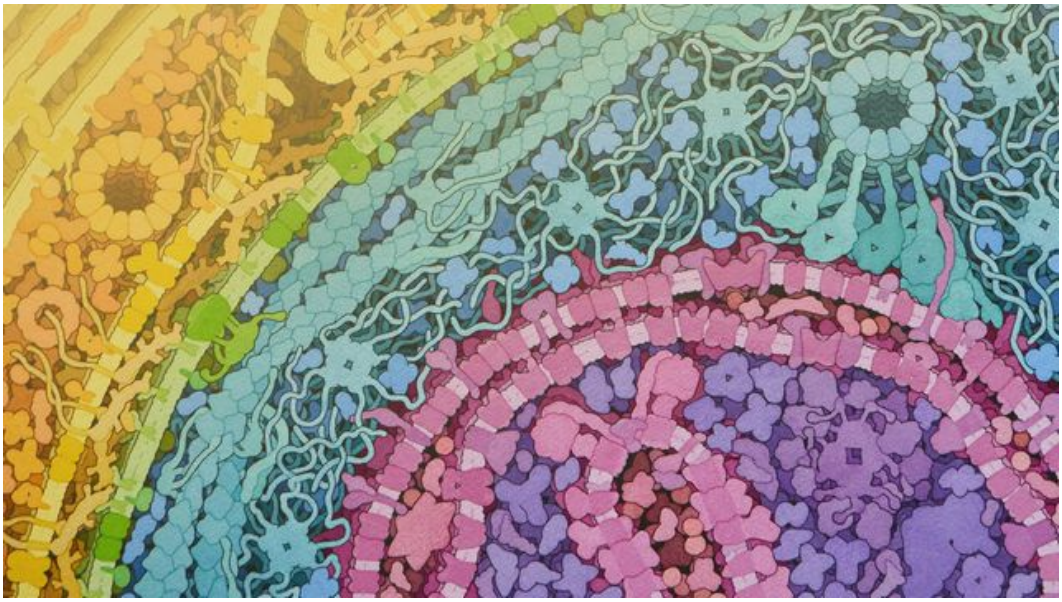


Figure 4: Image of a cell. Artist representation of a small part of a eukaryotic cell from cryo-ET images. Adopted from Goodsell [11]

The cell is the fundamental unit of life and is composed of various components, including proteins, nucleic acids, lipids, and carbohydrates (see Figure 4). Each of these components plays a crucial role in the cell's structure and function. Proteins are responsible for most cellular processes, while nucleic acids (DNA and RNA) carry genetic information. Lipids form cell membranes, and carbohydrates serve as energy sources and structural components.

It is at the Institut Pasteur in the 1950s that André Lwoff, Jacques Monod, and Agnes Ullmann made significant discoveries about the role of messenger RNA, gene regulation, and genetic programs in cellular function. Together with François Jacob (see Figure 5), yet another Pasteur Institute scientist, they proposed the operon model of gene regulation in prokaryotes, which explained how genes are turned on and off in response to environmental signals. For their discoveries, François, André, and Jacques were awarded the Nobel Prize in Physiology or Medicine in 1965 [12]. It is again at the Institut Pasteur, near Monod's and Jacob's buildings, that this Ph.D. was undertaken, aiming to understand further mRNA's role and the cell's regulation using AI models.

0.2.1 RNA

Among the cell's molecular components, RNA plays a particularly central role. RNA biology is a critical aspect of cellular function, encompassing processes such as transcription, translation, and regulation. Transcription is the process by which DNA is copied into RNA, which then serves as a template for protein synthesis during translation. Regulation of these processes is essential for maintaining cellular homeostasis and responding to environmental changes. This regulation can occur at multiple levels, including transcriptional control, RNA processing, and post-translational modifications [7].

The RNA hypothesis posits that RNA molecules were the first self-replicating entities,

Figure 5: Lwoff, Jacob, Monod in their Pasteur Institute Office. Adopted from *François Jacob, Jacques Monod et André Lwoff discutant de la structure des protéines et de leurs éléments de symétrie, en s'aidant pour cela de cartes à jouer* [13]

leading to the evolution of life as we know it. This hypothesis suggests that early life forms relied on RNA for both genetic information storage and catalytic functions, paving the way for the development of DNA and proteins, showing how RNA might be one of the most central components of the cell [14].

Many different types of RNA exist, each with distinct functions. Messenger RNA (mRNA) carries genetic information from DNA to ribosomes for protein synthesis, while transfer RNA (tRNA) and ribosomal RNA (rRNA) allow translation of mRNAs into proteins. Other types of RNA, such as small interfering RNA (siRNA) and microRNA (miRNA), are involved in gene regulation and silencing. Long-non-coding RNAs (lncRNAs) also play crucial roles in regulating gene expression and chromatin structure [15].

0.2.2 Gene Expression

The various types of RNA described above participate in a tightly regulated process called gene expression. Gene expression is the process by which information from a gene is used to synthesize a functional gene product, typically a protein. This process involves several key steps, including transcription, where DNA is transcribed into mRNA, and translation, where mRNA is translated into a protein (see Figure 6). Transcription factors (TFs) are proteins that bind to specific DNA sequences to regulate the transcription of genes. They play a crucial role in determining which genes are expressed in a cell at any given time, influencing cellular function and identity [7].

Transcription factors also interact with other proteins, such as cohesin, which helps maintain chromatin and the specific 3D structure of the DNA. Chromatin is the complex of DNA and proteins that forms chromosomes within the nucleus of eukaryotic cells. The organization of chromatin is essential for regulating gene expression, as it determines the accessibility of DNA to transcription machinery (see Figure 7).

Understanding the transcriptional rules and grammar of TF binding is helping us engineer

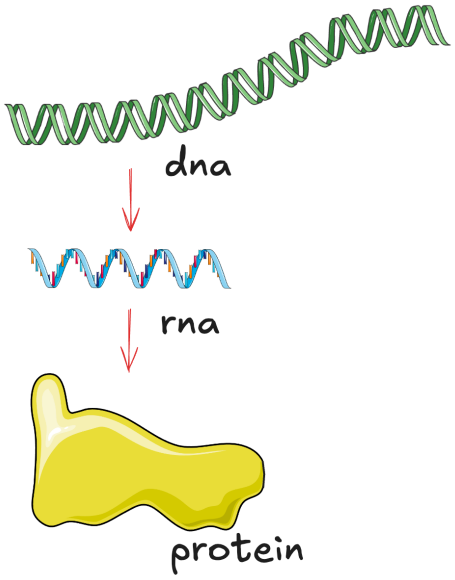


Figure 6: The central dogma of biology also represents the classic view of how gene expression occurs.

bacteria and eukaryotic cells to express specific genes.

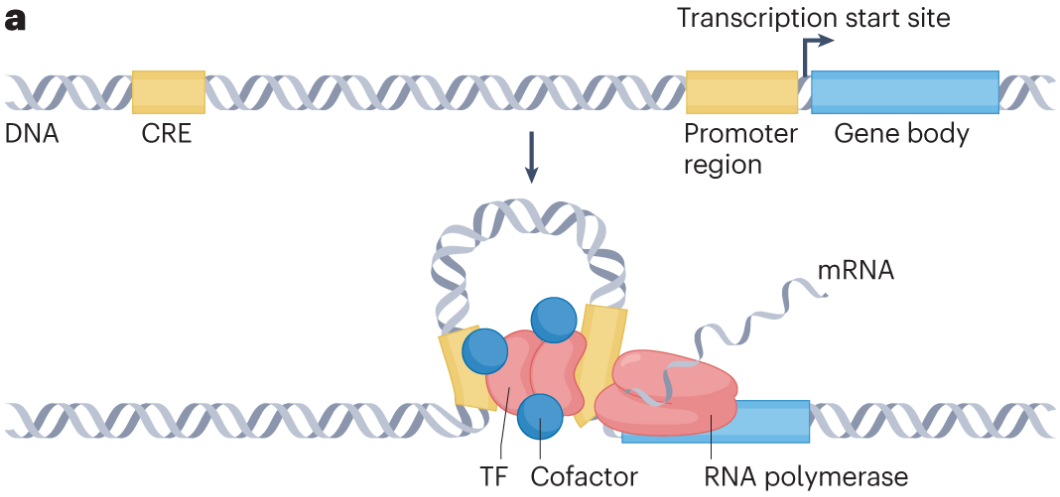


Figure 7: Image of the classical view of gene regulation whereby transcription factors and cofactors bound to cis-regulatory elements and each other recruit the transcriptional machinery at the promoter region of the transcription start site. Adopted from Badia-i-Mompel et al. [16].

0.2.3 Gene Regulatory Networks

Given this complexity, biologists have relied on the concept of gene regulatory networks (GRNs) to simplify the complex interactions within the cell [16]. GRNs are networks of molecular interactions that govern gene expression levels in a cell. They consist of genes, transcription factors, and other regulatory elements that interact to control the timing and level of gene expression (see Figure 8). Although very coarse and likely incomplete,

These modeled interactions provide insights into how cells might respond to various stimuli, differentiate into specific cell types, and maintain homeostasis. They are used by researchers every day through pathway, regulon, and other ontological relationship databases. They help us understand diseases, their mechanisms, and improve crop quality and yield.

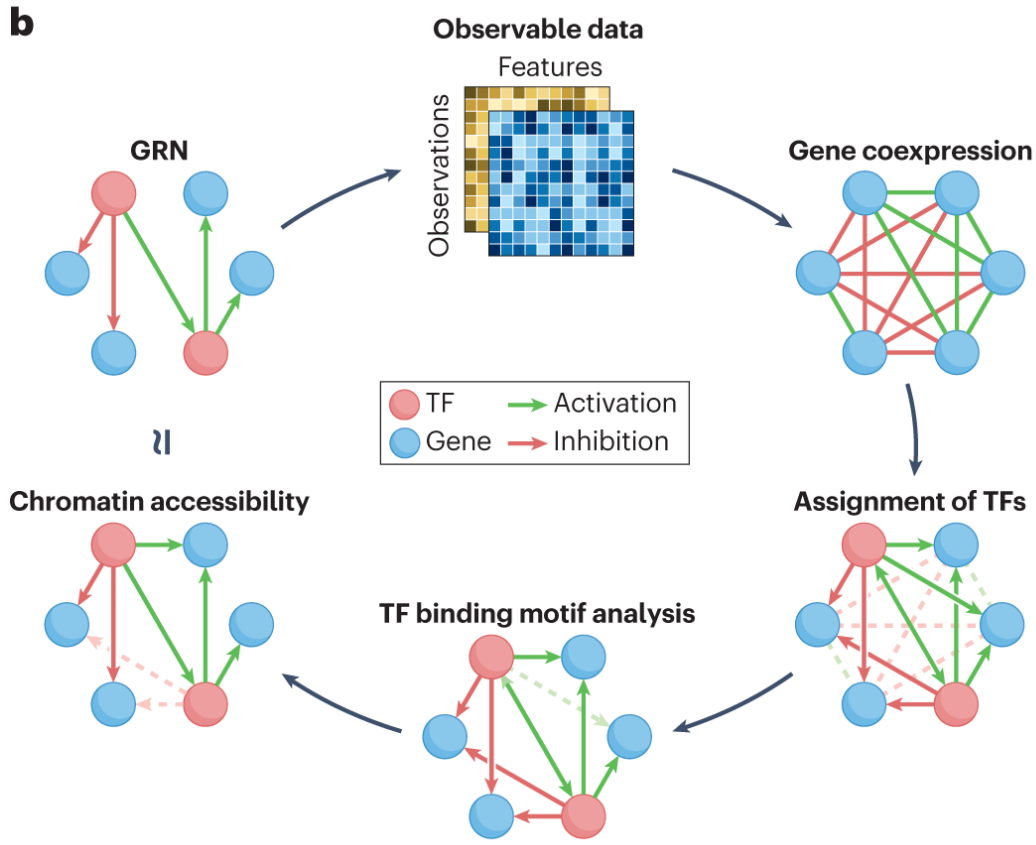


Figure 8: Inferring gene networks using cellular measurements. In this loop, we see the classical approach, where multiple measurements are combined with prior knowledge and heuristics to design a network that better understands the observations and makes predictions. Adopted from Badia-i-Mompel et al. [16].

0.3 Single-Cell Genomics

While GRNs provide a conceptual framework for cellular regulation, measuring gene expression at scale requires advanced experimental technologies. In this section, we trace the evolution of sequencing methods from early techniques to modern single-cell approaches that generate the data our models are trained on.

0.3.1 Sequencing

The study of gene expression at scale started within the field of genomics. Genomics is the study of an organism's complete set of DNA, including all of its genes. It involves sequencing

and analyzing the entire genome to understand its structure, function, and evolution. Then, the development of high-throughput sequencing technologies revolutionized genomics, allowing researchers to sequence entire genomes quickly and cost-effectively.

Initially, DNA sequencing was performed using Sanger sequencing (see Figure 9), the first method developed for this purpose. It involves selectively incorporating chain-terminating dideoxynucleotides during DNA replication, enabling researchers to determine the nucleotide sequence of a DNA molecule. This method was labor-intensive and time-consuming, but it laid the foundation for modern sequencing techniques [17].

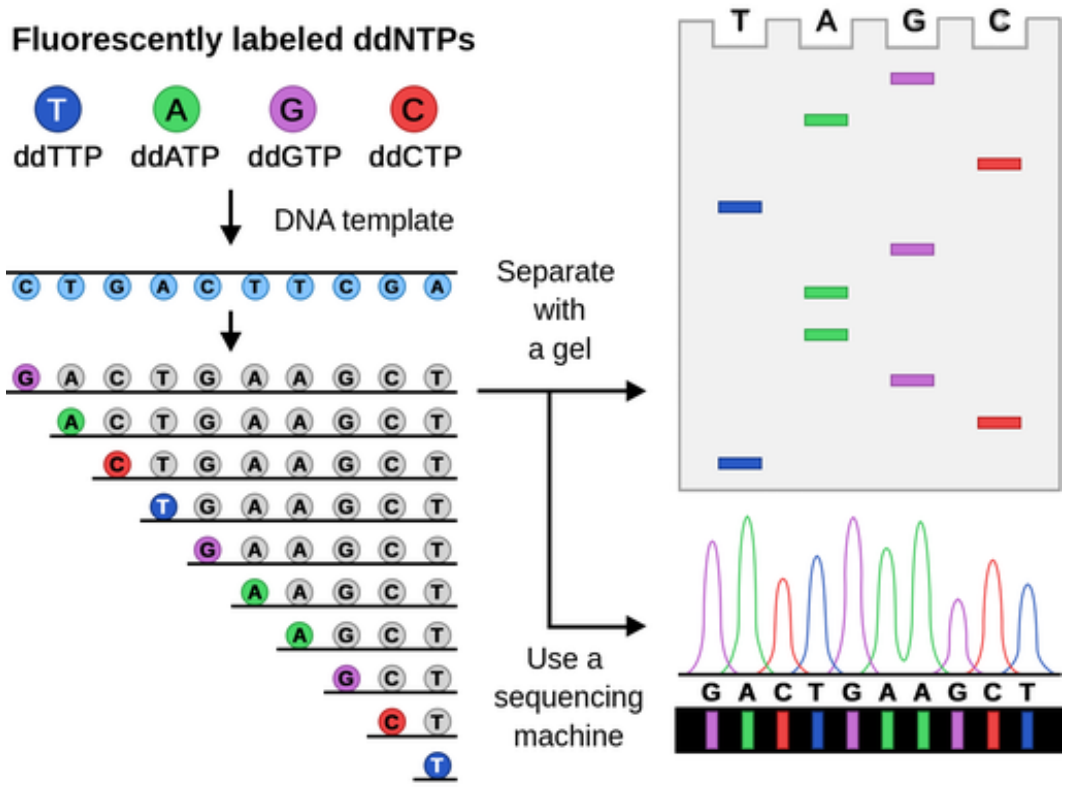


Figure 9: Main method for sequencing DNA. Sanger sequencing uses the sequential addition of fluorescent complementary nucleotides to read the genome one nucleotide at a time. A highly parallelized version is being used in most modern sequencers. Adopted from *dna sequencing* [18].

0.3.2 Next-Generation Sequencing

Sanger sequencing paved the way for a revolution in throughput. Nowadays, we use next-generation high-throughput sequencing (NGS) technologies, which allow for the massively parallel sequencing of millions of DNA fragments—also called reads—simultaneously (see Figure 10). It has significantly reduced the time and cost required for genome sequencing, enabling large-scale genomic studies and personalized medicine approaches [19].

Reads, small chunks of DNA, often likened to tiny puzzle pieces, are multiplied and sequenced in parallel. The resulting sequences are then aligned (or mapped) to a reference genome, which serves as a template for assembling the reads into a complete genome

sequence. The average number of overlapping reads that cover a specific region of the genome is called sequencing depth or coverage. Higher sequencing depth generally leads to more accurate and reliable results, as it reduces the likelihood of errors and increases the confidence in variant detection.

In the last decade, large-scale sequencing efforts such as the 1 million genomes project, the Human Genome Project, and the 1000 Genomes Project have provided valuable insights into human genetic variation and disease susceptibility [20, 21]. Genetic sequencing now allows us to define the genetic basis of many diseases, identify which drug might work for specific patients, and establish follow-ups for high-risk patients. It is increasingly driving clinical decisions and is becoming a standard part of patient care.

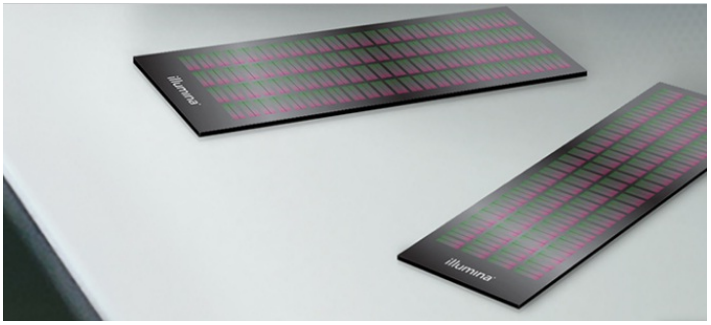


Figure 10: Image of an Illumina next-generation sequencing chip. Adopted from *Infinium Global Screening Array-48 Kit* [22].

0.3.3 New Modalities in Sequencing

Beyond reading the genome itself, sequencing also enabled many new applications, such as the study of gene expression. Here, we use a sequencer to read the mRNAs present in specific tissues, a process known as RNA sequencing (RNA-seq) [23]. Other examples abound, such as the sequencing of DNA states, such as methylation (BS-seq), open chromatin (ATAC-seq), and chromatin immunoprecipitation (ChIP-seq), which provide a view of how the genome is being read at a point in time. From DNA and its mutations to its state in different contexts and how these lead to changes in RNA's expression and state, we have begun to develop a holistic view of various cellular mechanisms [24].

However, these methods provided only an average across cells within a tissue, not individual-cell data. In 2014, the first single-cell RNA sequencing (scRNA-seq) methods were developed, allowing researchers to analyze gene expression at the single-cell level [25]. It was a breakthrough in genomics, enabling the study of cellular heterogeneity and the identification of rare cell populations that had previously been missed in bulk analyses.

Since then, single-cell sequencing technologies have advanced rapidly, with new methods being developed to sequence other omics modalities at the single-cell level. Studies conducted on tens of thousands of cells in the 2010s are now done on millions of cells [26], generating what has been called cell atlases.

In our work, we are gathering all publicly available scRNA-seq datasets and atlases

across tissues, diseases, and species to train foundation models.

0.3.4 Future Directions

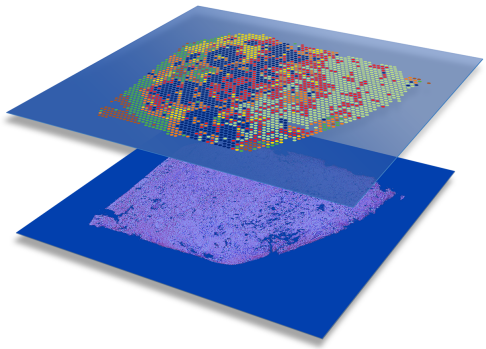


Figure 11: Illustration of spatial transcriptomics showing how it can be used to extract additional information from a tissue slide. Adopted from *Visium Spatial* [27]

Single-cell genomics continues to expand in both scale and scope. Recently, the development of spatial transcriptomics and imaging techniques has allowed researchers to study the spatial organization of gene expression within tissues, providing a more comprehensive view of cellular function in its native context [28, 29] (see Figure 11). Protein measurements are also being developed, unlocking an additional layer of information [30, 31]. Current applications have primarily focused on understanding diseases and drug effects within tissues. The technique allowed the identification of hundreds of new cell types and states, and improved the study of cellular development and differentiation. It had a substantial impact on cancer, neurological diseases, and immunology [32, 33].

Finally, the development of genetic perturbation techniques such as CRISPR-cas9 [34, 35], combined with sequencing (perturb-seq), enables us to go beyond observations and begin to understand the causal relationships between genes and their functions.

Indeed, these CRISPR screens enable researchers to systematically knock down genes in individual cells and observe changes in gene expression, providing further insights into gene function and regulatory networks. However, these techniques are in their infancy and still limited in scale, scope, and especially, quality.

0.4 Current Single-Cell Tasks

The wealth of single-cell data described above has given rise to a set of standard computational tasks and analysis pipelines. While many computational tools exist for these tasks, benchmarks that align with real use cases from the user's perspective remain scarce. Fortunately, the field has matured sufficiently to enable the creation of standardized tasks and pipelines for understanding, assessing, and analyzing data. Understanding these tasks

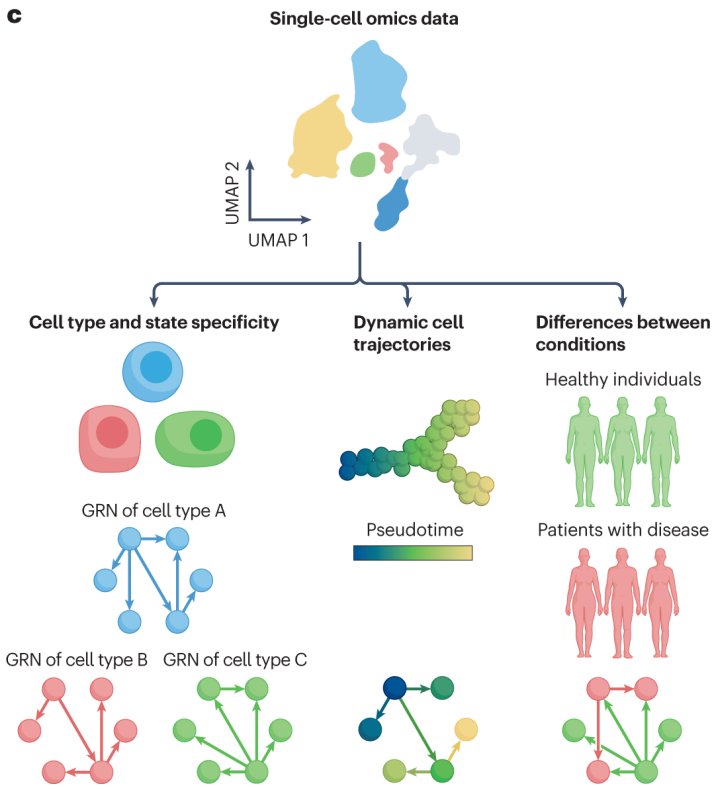


Figure 12: Single-cell data analysis pipelines and their relationship to gene networks inference. Adopted from Badia-i-Mompel et al. [16]

is crucial for this thesis, as they define both our pretraining objectives and evaluation benchmarks.

In single-cell RNA-seq data, a typical pipeline is as follows and contains several possible steps (see Figure 12). We categorize them by their role in our work: *preprocessing* (applied before model training), *pretraining tasks* (used to train our foundation models), *zero-shot tasks* (evaluated without fine-tuning), and *fine-tuning tasks* (requiring task-specific training):

1. **Alignment** [*preprocessing*]. It all starts with preprocessing the raw sequencing data: detecting/imputing the cell index, aligning the sequencing reads with a reference genome's gene locations, detecting low-quality cells, reads, doublet events, cell death events, and more [36]. These choices will introduce biases into the output dataset. This step is performed upstream of our models.

2. **Normalization and clustering** [*preprocessing*]. The data might be normalized to correct for sequencing depth and other gene-level biases, and cell clusters would be defined based on expression profile similarity. Finally, differential expression analysis is performed, in which clusters of cells are compared to identify genes that are differentially expressed between them [37]. Our models take normalized counts as input but learn their own expression tokenization.

3. **Batch correction / atlas alignment** [*zero-shot*]. The dataset might be aligned to a reference atlas when one exists. It is done using batch-correction methods, often built

around nearest-neighbor mapping, matrix factorization, or neural networks (NN), such as variational auto-encoders (VAEs) [38]. Our models perform batch correction zero-shot through learned invariant representations.

4. **Annotation and labeling** [*zero-shot / fine-tuning*]. Cluster-level and dataset-level labels might be inferred, such as cell type, tissue, disease, and age. These often come from prior knowledge, manual annotation based on differential expression features, automated tools, or alignment with other prelabeled datasets [39]. Our models achieve zero-shot cell-type classification via label-prediction pretraining and can be fine-tuned for specific annotation tasks.

5. **Denoising / imputation** [*pretraining task / zero-shot*]. In cases where specific clusters contain a low number of cells, denoising or zero-imputation methods can be used [40], but they haven't proven consistently useful in practice because they rely on cluster-level information. We use denoising as a core pretraining objective, and our models can denoise zero-shot [41].

6. **Multimodal integration** [*not addressed*]. If users have access to datasets of the same tissue from other modalities (scATAC-seq, BS-seq, protein measurements, imaging, etc.), multimodal alignment methods can be applied. Such alignments help bridge the gap between genotype (e.g., DNA mutations) and phenotype (expression, protein levels, pathology). This thesis focuses on scRNA-seq; multimodal integration remains future work.

7. **Trajectory inference** [*not addressed*]. If the dataset was measured in a non-static context, one can infer "cellular trajectories", i.e., how cells transition from one state to another based on many single-cell snapshots [42]. We do not directly address trajectory inference in this thesis.

8. **Spatial analysis and cell-cell interactions** [*zero-shot generalization*]. If the dataset contains spatial information, one can infer cell-cell interactions based on proximity and expression profiles [43]. We demonstrate zero-shot generalization to spatial transcriptomics data in scPRINT-2.

9. **Perturbation response prediction** [*fine-tuning*]. If the dataset includes perturbation experiments, one can predict how cells respond to specific perturbations, such as drug treatments or genetic modifications [44]. We benchmark perturbation prediction as a fine-tuning task and develop counterfactual reasoning capabilities.

These analyses and tools are available in a set of packages called scverse, which our work relies heavily on and has contributed to.

Having established the biological context and computational tasks that motivate this work, we now turn to the artificial intelligence methods that can be used to tackle them.

0.5 AI and Neural Networks

A virtual cell model has been the dream of computational and systems biologists for decades. Initially, these models were based on simplified representations of cellular processes, often focusing on specific pathways or interactions. The models examined chemical reaction parameters involving proteins, RNAs, and DNA. However, in addition to computational challenges, these models failed to generate realistic predictions of cellular behavior.

Nowadays, the idea has emerged that artificial intelligence techniques could help us solve some of these problems. But first, what is AI, and what is the difference between machine learning (ML), data science, and informatics?

0.5.1 Definitions

Data Science encompasses the gathering, management, and analysis of data in information systems. **Machine Learning** happens when one uses statistical methods to generate predictions from data. But while these methods can be seen in the framework of statistics, they also have an underpinning in other domains, like information systems, neuroscience (with neural networks), statistical physics, psychology, and applied mathematics (with Algebra, Topology, Analysis, and Optimization).

Artificial Intelligence (AI) is a broad term that has had multiple meanings in society and culture. For many, it mainly refers to applications of machine learning methods to human and animal-related tasks, such as understanding images, videos, speech, and text, as well as robot manipulation. In the field, it has often been a much broader term, encompassing knowledge bases, statistical methods, and neural networks (see Figure 13).

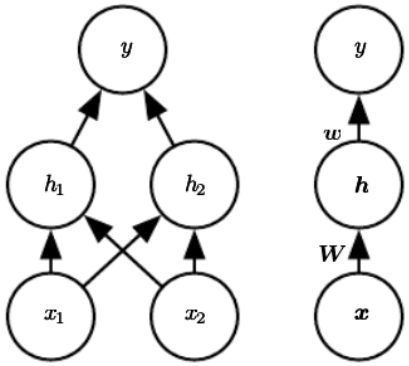


Figure 13: Example of a feed-forward neural network architecture, drawn in 2 different styles. Adopted from Goodfellow, Bengio, and Courville [45].

Recently, Machine Learning (ML) has made great strides in many areas, primarily due to a significant increase in data generation, along with improvements in optimization methods and neural networks. In this Ph.D., we are piggybacking on these improvements and performing data science and machine learning on single-cell data.

We now present an overview of these methods and provide intuition for why they work.

0.5.2 Learning Paradigms and Their Applications to Single-Cell Data

Machine learning methods can be categorized into three main paradigms, each with distinct applications in our field:

Supervised learning involves training models on labeled data to predict outcomes. In single-cell genomics, this can include cell-type classification from expression profiles, disease-state prediction, and perturbation response modeling. These methods require curated annotations, which are often expensive and inconsistent across datasets. It is used, for example, when classifying cell type.

Unsupervised learning discovers structure in unlabeled data. Traditional applications include dimensionality reduction (PCA, UMAP) for visualization, clustering for cell type discovery, and matrix factorization for batch correction. It is used, for example, to learn representations of cells.

Self-supervised generative learning represents the paradigm underlying foundation models. It can be seen as an instance of unsupervised learning, where models are pretrained on large unlabeled datasets using proxy tasks—such as predicting masked genes or reconstructing corrupted inputs—that force them to learn meaningful representations. These representations can then be fine-tuned for downstream tasks or used zero-shot. This thesis develops foundation models in this paradigm, using denoising and expression reconstruction as pretraining objectives. Generative AI has led the second big AI revolution of this decade with tools to generate images, videos, music, voice, and text [46, 47, 48].

0.5.3 Representation Learning and Embeddings

Most modern machine learning tools rely on representation learning, converting concepts and objects into high-dimensional vectors called embeddings [45]. In deep Neural Networks (NN), these embedding vectors are processed through layers of mathematical operations, often involving matrix multiplications and non-linear activation functions. The layers are designed to learn hierarchical representations of the input data, capturing increasingly complex features as data passes through the network. For single-cell data, this means learning representations in which similar cells (by type, state, or function) cluster together in embedding space, enabling downstream tasks such as classification and trajectory inference. Such tools have been heavily applied in single-cell genomics, with many methods relying on representation learning to denoise, align, and analyze data [38].

Recently, specific neural network architectures with powerful scaling properties, called transformers, have become ubiquitous. Researchers have extended their capabilities to scientific domains, including weather prediction [49] and, as we explore in this thesis, cellular modeling. But why and how are these models so powerful? While this remains an active area of research, essential theories have been presented.

0.5.4 Why Does It Work? Architectural Innovations

Contrary to previously accepted machine learning dogma, deep learning researchers showed in the 2010s that increasing the number of parameters did not necessarily lead to overfitting. Instead, thanks to regularization methods such as dropout and weight decay, models learned more complex patterns in the data.

Several key architectural innovations enabled modern deep learning:

Skip connections (residual connections) prevent vanishing gradients by allowing information to flow directly across layers. This innovation led to architectures such as ResNet [50], enabling the training of much deeper networks: a major component of the deep learning revolution of the 2010s. It is also widely used in transformer-based models.

Normalization layers (batch normalization, layer normalization) stabilize training by normalizing intermediate activations, allowing higher learning rates and faster convergence. Layer normalization is particularly important for transformers and is used in all models.

Tokenization and attention make the models both more parallelizable and more complex. Indeed, it allows models to work on matrix inputs, where each input value becomes a vector of numbers (also called an embedding or token). For text, this involves subword units; for single-cell data, we can let our imagination run free; it could be genes, cells, molecules, indeed, we will be using both in our models. We then use classical neural networks for per-token processing and the attention mechanism to enable tokens to interact with one another. Taken together, these make the model even more parallelizable in both depth and width.

GPUs or Graphical Processing Units, which are the final enablers allowing us to reach unprecedented scales through large, very efficient parallel matrix operations.

These innovations are particularly relevant for genomics: attention mechanisms can capture gene-gene interactions, parallelization enables processing thousands of genes, and proper tokenization of expression values is crucial for learning meaningful representations. Indeed, they have already greatly impacted biology across multiple fronts, a famous example being the recent Nobel Prize-winning AlphaFold2 model, built on modified attention architectures [51].

The final piece of the puzzle is in how we train such models.

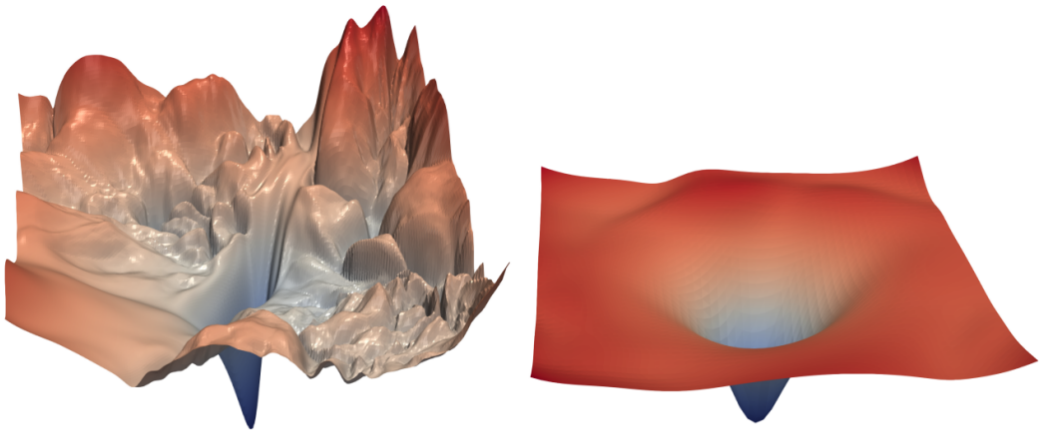
0.5.5 Optimization and Loss Landscapes

Interestingly, whether small or large, neural network or otherwise, almost all ML methods are trained using variants of gradient descent optimization methods. These methods aim to minimize a loss function, which quantifies the difference between the model's predictions and the actual data. By iteratively adjusting the model's parameters in the direction that reduces the loss, these optimization algorithms help the model learn from the data.

But this is through *stochastic* gradient descent (SGD) methods like Adam that we suc-

cessfully train NNs [52, 53]. Indeed, using only a small subset of the data at each training step is not only much faster to minimize the loss function, but it also helps escape local minima and saddle points [54].

To understand this, we need to understand the loss landscape. Imagine a 3D landscape where the height represents the loss value, and the two other ”surface” dimensions are the model's parameters (see Figure 14). The goal of the model is to find the lowest point in this landscape, which corresponds to the best set of parameters to fit the data. However, this landscape is very complex, with many local minima and saddles, which would prevent the model from reaching a nice minimum; it wanders blindly and can only sense its immediate surroundings.



(a) without skip connections (b) with skip connections

Figure 14: Visualization of a loss landscape with and without skip connections. The figure shows the loss values on the z-axis and color-coded by parameters on the x- and y-axes. Adopted from Li et al. [54]

In deep neural networks, there are not just two but billions of dimensions, however. In this context, the loss landscape doesn't contain that many local minima; there is always a possible direction to decrease the loss. Furthermore, SGD, by being stochastic, alters the loss landscape each time, allowing the model to easily find an escape direction from a local minimum or a saddle point (a locally flat surface).

Behind this unexpected behavior is the unsettling theory of emergence. Or how small objects can combine and interact in ways that would be unexpected and difficult to predict based on their individual properties. This theory tries to explain phenomena in dunes, snowflakes, ant colonies, and life itself, which might explain how large neural networks achieve such complex behaviors [55].

0.5.6 Transformer Architectures: Encoder vs. Decoder

For text, images, videos, and audio, large language models (LLMs) are now ubiquitous. In scientific domains, we call similar models trained on all available data in a given modality

foundation models. These models represent a paradigm shift from small, task-specific neural architectures to larger, transformer-based architectures (see Figure 15), trained across entire data domains, to generalize to unseen datasets and novel tasks [56].

Two main transformer architectures exist, with different implications for genomics:

Encoder-only models (e.g., BERT [57]) use **bidirectional** attention, where each token can attend to all other tokens. This is well-suited for understanding and classification tasks. For single-cell data, bidirectional attention allows each gene to see all other genes, naturally modeling gene-gene interactions.

Decoder-only models (e.g., GPT [58]) use **causal/unidirectional** attention, where each token can only attend to previous tokens. They are trained with **autoregressive** prediction: predicting the next token given previous ones.

In single-cell genomics, the choice is consequential: genes lack a natural ordering (unlike words in sentences), making bidirectional architectures more appropriate. Our models use encoder-style bidirectional attention, treating genes as an unordered set where each gene can attend to all others. This design choice enables learning symmetric gene-gene relationships suitable for GRN inference.

Structured and sparse attention. Standard attention allows every token to attend to every other token, with complexity $O(n^2)$. However, biological systems have inherent structure: genes interact through regulatory networks, proteins form complexes, and cells communicate through defined pathways, motivating variants of the attention mechanism that incorporate prior knowledge or learn sparse patterns. **Graph attention networks** [59] restrict or bias attention to predefined edges in a graph, allowing tokens to attend only to their neighbors. It is particularly relevant for genomics, where protein-protein interaction networks or known regulatory relationships can guide attention.

Similar ideas were used in the development of AlphaFold2 and AlphaFold3 [60, 51]. In text processing, similar examples abound of **Sparse attention** mechanisms like Longformer [61] and BigBird [62], which combine local windowed attention with global tokens, achieving linear complexity while maintaining expressiveness.

Finally, while some of these methods already decrease the computational complexity of attention, **efficient attention** methods such as hyper attention, Performer, and others [63, 64, 65] approximate the attention mechanism using low-rank approximations, dropping blocks, or using kernel methods.

For single-cell models processing thousands of genes, such efficiency gains are essential. In our work, we explore how attention patterns learned during pretraining can bias them, and how we can reduce their computational complexity.

0.5.7 The Pretraining-Fine-tuning-Zero-shot Pipeline

The foundation model paradigm involves three phases that we leverage extensively in this thesis:

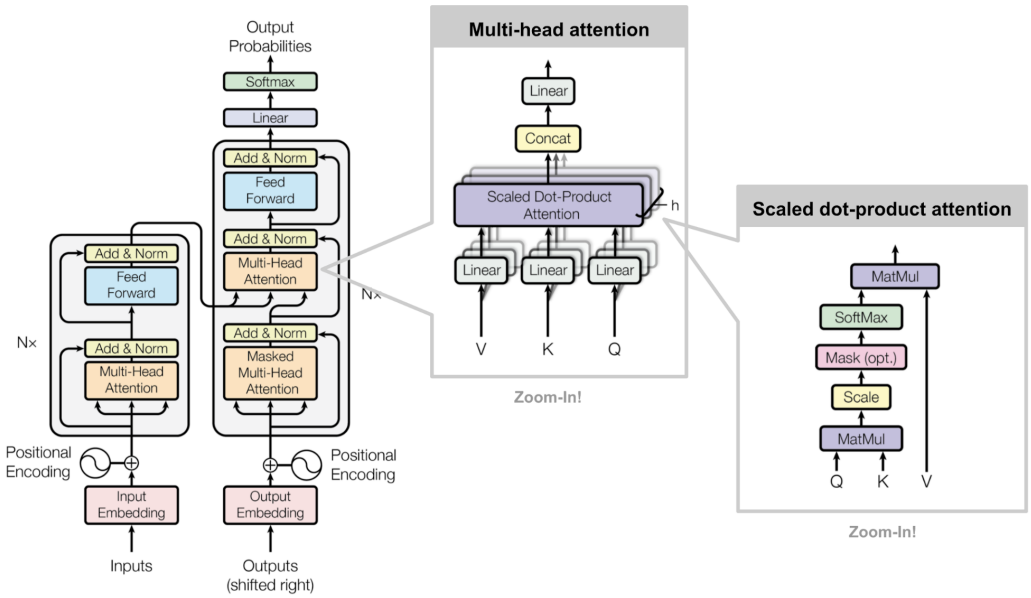


Figure 15: Transformer architecture overview. From Attention is all you need. The leftmost block is an encoder transformer, the longer block to its right is a decoder transformer. Nowadays, most methods are either one or the other, but many combinations, such as the one presented here, can be created. Adopted from Vaswani et al. [66].

Pretraining uses self-supervised tasks on large unlabeled datasets. For language, this is next-token prediction; for single-cell data, we use expression denoising (reconstructing original counts from corrupted inputs), bottleneck learning (compressing and reconstructing cell representations), and hierarchical label prediction. Pretraining teaches the model general patterns; for our models, this includes gene co-expression patterns, cell-type signatures, and regulatory relationships.

Fine-tuning adapts pretrained models to specific tasks using smaller labeled datasets. The pretrained weights provide a strong initialization, enabling faster convergence and better performance than training from scratch. We fine-tune for tasks like perturbation prediction, where limited experimental data is available.

Zero-shot inference applies pretrained models directly to new tasks without any task-specific training, thus testing whether pretraining has learned genuinely transferable representations.

In this thesis, we will examine pretrained foundation models in both their zero-shot and fine-tuned abilities, but we are most interested in the representations they learn during pretraining.

This pipeline is particularly valuable for biology, where labeled data is expensive and inconsistent. By pretraining on millions of cells, single-cell foundation models might learn robust representations that transfer across tissues, diseases, and even species.

Introduction

0.6 Motivation and problem setting

The cell is the fundamental unit of life, composed of various components including proteins, nucleic acids, lipids, and carbohydrates. Despite decades of research, we still lack the ability to accurately predict how a cell will respond to a given stimulus, design targeted therapies with high confidence, or engineer cellular behavior from first principles. The central problem this thesis addresses is: *can we build computational models that learn meaningful representations of cellular state from large-scale transcriptomic data, and can these representations be used to infer gene regulatory relationships and generalize to unseen biological contexts?*

The objectives of cellular biologists are to understand and control cells, with the dream of engineering life from plants to animals and even generating entirely new synthetic life [10].

Achieving this vision requires a deep understanding of cellular mechanisms. Before developing a drug for a disease, for instance, one must identify the target genes in specific cell types that need to be reactivated, deactivated, or modified to address the disease's underlying mechanism. Understanding mRNA and siRNA has already enabled potent therapies, including some of the well-known COVID-19 vaccines. Unfortunately, many RNA types remain poorly understood, and their functions are an active area of research. In eukaryotic cells, like our own, RNAs are produced through gene expression and are actively regulated by the cell.

But drugs are not the only application of cellular understanding. Life is everywhere, and engineering has already helped us make better crops, create synthetic meat, and design fungi that remove pollution. Yet cells are extraordinarily complex, and we remain limited by our understanding of their inner workings.

Recent advances in single-cell sequencing have begun to change this. Technologies have advanced rapidly, with studies conducted on tens of thousands of cells in the 2010s now scaling to millions [26], generating what have been called cell atlases. This explosion of data has driven hundreds of companies and dozens of institutes to pursue virtual cellular models [5]—computational systems that aim to simulate and predict cellular behavior, enabling *in silico* experimentation before costly wet-lab validation.

0.6.1 Current Challenges

Single-cell sequencing itself comes with a set of challenges that directly motivate the methods developed in this thesis. The main issues are:

1. **Sparsity and noise.** Most current single-cell sequencing methods capture only 10-20% of transcripts, leading to many zeros ("dropouts") in the data. Our models address this through denoising pretraining tasks and learned expression tokenization.
2. **Batch effects.** Strong biases in data generation make cross-dataset analysis challenging. Our foundation models learn batch-invariant representations through large-scale pretraining across hundreds of datasets.
3. **Limited coverage.** Many tissues, rare cell types, and non-model organisms remain undersequenced. We demonstrate cross-species generalization by training on 16 organisms and testing on unseen species.
4. **Missing modalities.** Spatial context and protein levels are often unavailable. We show zero-shot generalization to spatial transcriptomics data without spatial-specific training.

These challenges define the benchmarking framework we use to evaluate our models and motivate our architectural choices.

Beyond these data-level challenges, biological complexity itself poses fundamental obstacles. A single human cell contains approximately 20,000 protein-coding genes, but gene regulation extends far beyond simple on/off switches: it involves combinatorial control by transcription factors and cofactors, epigenetic modifications, post-transcriptional regulation by non-coding RNAs, alternative splicing, protein-protein interactions, and metabolic feedback loops (see 0.2). These processes interact across multiple spatial and temporal scales—from millisecond signaling cascades to days-long differentiation programs—creating a system whose emergent behavior cannot be easily predicted from its individual components. It is this multi-layered complexity that makes purely mechanistic modeling insufficient and motivates the data-driven approaches developed in this thesis.

0.6.2 Gene Regulatory Networks

Gene regulatory networks (GRNs) provide a simplified yet powerful framework for understanding how genes interact within cells. The foundations of this field were laid at the Institut Pasteur in the 1950s, where André Lwoff, Jacques Monod, Agnes Ullmann, and François Jacob (see Figure 16) made seminal discoveries about messenger RNA, gene regulation, and the operon model [12]. It is again at the Institut Pasteur, near Monod's and Jacob's buildings, that this Ph.D. was undertaken, aiming to understand further mRNA's role and the cell's regulation using AI models.

Traditional computational cell models based on chemical reaction parameters attempted to simulate these regulatory dynamics but failed to generate realistic predictions of cellular

Figure 16: Lwoff, Jacob, Monod in their Pasteur Institute Office. Adopted from *François Jacob, Jacques Monod et André Lwoff discutant de la structure des protéines et de leurs éléments de symétrie, en s'aidant pour cela de cartes à jouer* [13]

behavior, largely because they could not capture the full combinatorial complexity of gene interactions [16]. Moreover, as discussed in the Background chapter, gene regulation extends well beyond transcription factors: it involves cofactor proteins, non-coding RNAs, RNA maturation, and protein translation, with interactions spanning all molecular layers. Gene regulatory networks (GRNs) provide a simplified framework for these interactions, while gene networks (Gene Network (GN)s) encompass a broader set of relationships, including protein-protein interactions and metabolic pathways.

Current GRN inference methods suffer from additional practical limitations: they typically operate on only a small subset of genes (often restricted to known transcription factors), process only a limited number of cells (failing to scale to modern atlas-sized datasets), and do not fully exploit the richness of single-cell expression profiles [67]. Despite this long history, inferring accurate cell-type-specific GRNs remains a major bottleneck in computational biology. Current methods face several challenges: (1) the combinatorial explosion of possible gene-gene interactions makes exhaustive testing infeasible, (2) correlation-based methods cannot distinguish direct from indirect effects [68], (3) perturbation data is expensive and limited in scale, and (4) ground truth networks for validation are sparse and context-specific [69]. These limitations motivate the development of new approaches, including the foundation model-based methods presented in this thesis.

0.7 Foundation Models

The challenges outlined above—noisy, sparse, and heterogeneous single-cell data combined with the combinatorial complexity of gene regulation—demand models that can learn from massive datasets without requiring exhaustive manual annotation. Traditional approaches, whether statistical (correlation-based GRN inference) or mechanistic (chemical kinetics simulations), have not scaled to the complexity of the problem. The emergence of modern AI—particularly self-supervised transformers pretrained on large datasets—has opened a new avenue: foundation models for single-cell biology. The key question is whether the

paradigm that transformed natural language processing can be adapted to biological data, where tokens lack a natural ordering, expression values are continuous, and the underlying generative process is fundamentally different from language.

0.7.1 LLMs

In Language modelling, large language models (LLMs) have demonstrated that self-supervised pretraining on massive text corpora produces representations that generalize across a wide range of tasks. Models such as BERT [57] and GPT [58] learn contextual relationships between tokens—words or subword units—through tasks like masked language modeling or next-token prediction. The success of these models relies on three pillars: (1) the transformer architecture, which captures long-range dependencies through attention mechanisms; (2) scale, both in data and parameters, which enables emergent capabilities; and (3) the self-supervised paradigm, which removes the need for labeled data during pretraining. These principles have inspired a wave of foundation models in scientific domains, from protein sequences (ESM2 [70]) to molecular structures (AlphaFold [51]), and, as we discuss below, to single-cell transcriptomics.

0.7.2 Bio-Foundation Models

The first practical example of a single-cell (RNA-seq) foundation model was scBERT, released in 2021. However, it was only used and benchmarked for cell type classification and pretrained on 1 million single cells [71]. The first foundational model with broader claims, Geneformer, was released a year later [72] (see Figure 17). The authors demonstrated the model's ability to perform various single-cell tasks, including cell-type classification, gene regulatory network inference, and perturbation prediction. Geneformer was trained on a much larger dataset of 33 million single cells.

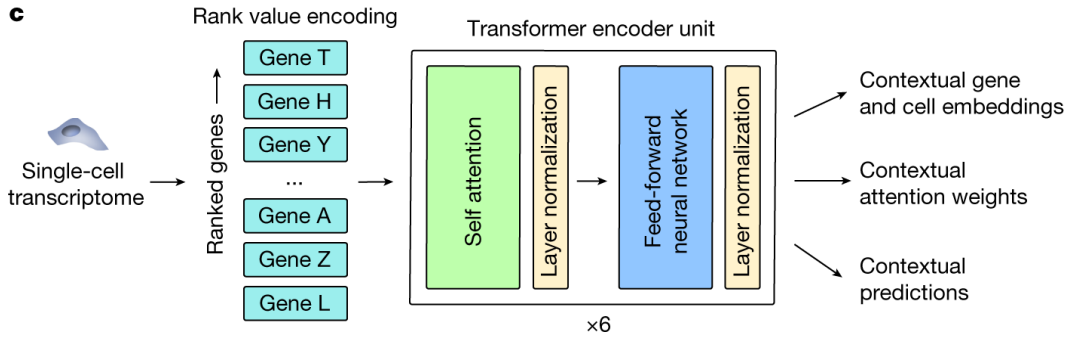


Figure 17: The Geneformer model, where genes are represented as words and cells as sentences, where genes are ordered by their expression level. Adopted from Theodoris et al. [72].

However, Geneformer, like scBERT, was essentially an LLM (BERT) applied directly to single-cell data. In this context, words are gene names, listed in order of expression level in the cell to form a sentence. This design choice raises questions about whether such direct adaptations from NLP are optimal for biological data.

0.7.3 Current Single-Cell Foundation Models and Their Limitations

In 2023, a year after Geneformer, several additional foundation models were released. scGPT [73] showcased a GPT-style architecture and presented various losses for fine-tuning. It was the first example of systematic fine-tuning in single-cell and a more in-depth benchmark across four abilities: cell type prediction, gene network inference, perturbation prediction, and batch correction. However, it did not outperform state-of-the-art methods [74, 75]. At the same time, Universal Cell Embedding (UCE) [76] demonstrated cross-species training to achieve state-of-the-art cross-species cell embeddings, introducing a contrastive loss function for cell representation learning (see Figure 18).

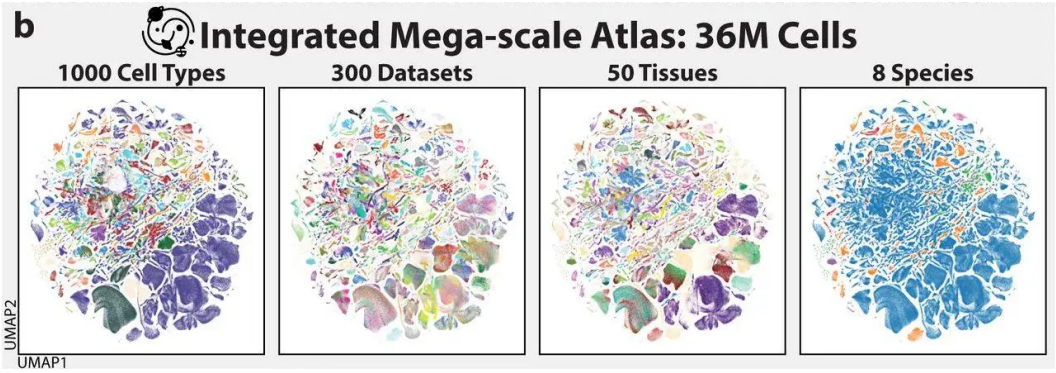


Figure 18: Low-dimensional visualization of universal cell embeddings across species. Each point is a cell positioned near similar cells according to this foundation model. Adopted from Rosen et al. [76].

Finally, scFoundation [77], despite being closed-source, showcased a truly novel architecture specifically built for single-cell data and a novel training method based on the noise-to-sequencing-depth relationship.

Key bottlenecks in single-cell foundation models. At the start of this Ph.D., we identified several limitations in existing approaches that motivated our contributions:

1. **Expression tokenization.** Existing models used hand-crafted binning or rank-ordering of expression values. We hypothesized that learned tokenization could better capture the biological signal.

2. **Gene representation.** Most models learned gene embeddings from scratch, ignoring rich prior knowledge from protein sequences. We introduced protein-based gene encoding using ESM2 embeddings.

3. **Genomic context.** Gene position on chromosomes affects co-regulation, but this was ignored. We added genomic positional encoding.

4. **GRN inference.** Claims about GRN inference were not rigorously benchmarked. We developed BenGRN, a comprehensive benchmarking suite.

5. **Scalability.** Transformer quadratic complexity limited genome-wide analysis. We developed efficient attention mechanisms for large-scale inference.

6. **Reproducibility.** Many models were not open-source or reproducible. We committed to releasing all code, models, and benchmarks.

7. **Cross-species generalization.** Training was limited to human/mouse. We scaled to 16 organisms.

0.8 Scientific aim

The limitations identified above—in both GRN inference methods and existing single-cell foundation models—point to a clear need: models that not only perform well on standard benchmarks but whose representations can be mechanistically interpreted to yield biological insight. Beyond building better architectures, we must understand *what* these models learn, *whether* their improvements are genuine or artifacts of evaluation choices, and *how* to make them practically useful for the biological community. This requires rigorous benchmarking, reproducible training, and systematic ablation of design decisions. In what follows, we describe the initial aims that guided this Ph.D. and how they evolved into the contributions presented in the subsequent chapters.

0.8.1 Initial aim

At the start of the project, we wanted to understand how single-cell foundation models worked—or whether they worked at all—and to improve gene regulatory network (GRN) inference using single-cell RNA sequencing (scRNA-seq) data, sensing a possible interplay between the two. Before presenting our contributions, it is important to reflect on what we initially set out to achieve. This being a Thesis by Articles, each of the three main chapters corresponds to a specific scientific publication.

This Ph.D. project initially aimed to develop new deep learning approaches, possibly using graph neural network architectures on large scRNA-seq datasets, to assess their predictive performance on high-quality benchmarks and package them as an open-source Python library. Our principal idea was to use Graph Neural Networks (GNN)s. GNNs are a class of deep learning layers designed to operate on graph-structured data. They are specifically tailored to handle modalities where edges connect the different input elements (nodes or vertices) [78, 79, 80] (see Figure 19).

Traditional neural networks are primarily designed to process grid-like data, such as images, or sequential data, such as text. However, GNNs extend this capability to graph-structured data by incorporating a pooling operation across connected nodes.

Objectives. We wished to improve GRN predictions from scRNA-seq data. Our approach was:

1. To use larger neural network models that scale linearly with the dataset size, taking advantage of the tens of millions of data points now becoming available.

2. To use novel GNN layers that can reduce the model's "search space" by constraining the set of possible topologies it learns.
3. To improve the pretraining and fine-tuning of these models to the predictive task they have to perform, and the constraints of the system they are predicting.
4. To formulate better layers that correspond to the sparse interactions between genes and our current knowledge about their functions.
5. To create formal and rational benchmarks that best capture the ability of a GRN methodology.
6. To assess predictions and any usefulness or lack of it by having biologists test hypotheses using the model.

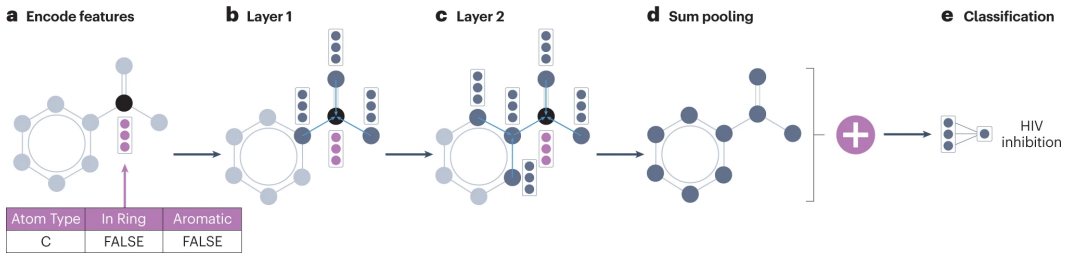


Figure 19: Illustration of the graph neural network mechanism, update and pooling (e.g., summing) across multiple connected nodes represented as vectors. Adopted from Corso et al. [81].

0.8.2 Potential Impacts

From these initial objectives, we envisioned several impacts. This Ph.D. project will contribute to methodological breakthroughs by providing new tools and methods for applying neural networks to unstructured data such as scRNA-seq, and to improve the state of the art in GRN prediction.

The proposed methodologies will impact computational (bioinformatics, machine learning) and biomedical fields. The new architectures might address challenges faced by related fields such as environmental research, industrial biotechnology, and biofuel studies. The improved GRN predictions will enhance our understanding of cellular processes, potentially leading to new therapeutic targets and strategies for treating diseases. The open-source Python library will democratize access to these tools, enabling researchers worldwide to apply them to their data and questions.

0.9 Thesis scope & contributions

Given the limitations of existing approaches—graph neural networks that do not scale, foundation models with unverified claims, and a lack of rigorous benchmarks—this thesis addresses the problem of building single-cell foundation models that are simultaneously scalable, biologically meaningful, and rigorously evaluated. We investigate whether modern transformer architectures can learn cellular representations that capture gene regulatory

relationships from single-cell RNA sequencing data. We explore their utility for tasks such as denoising, imputation, cell type annotation, and batch correction, and through this lens we design improvements via better architectures and training strategies.

0.9.1 Limitations

While the initial objectives centered on graph neural networks, early investigations revealed fundamental limitations of this approach for GRN inference. First, reliable ground-truth GRNs are largely unavailable, making it impossible to initialize a model from a known graph topology. Second, GNNs do not scale well to the tens of thousands of genes present in a cell, and systematic benchmarks consistently show them underperforming transformer-based models on comparable tasks [82, 83].

These findings led us to adopt transformers instead, which can be viewed as GNNs operating on fully-connected graphs [84, 85, 86, 87]. The principal challenge with transformers is their quadratic complexity with respect to the number of input tokens. Addressing this scalability bottleneck—making transformers scale sub-quadratically with the number of input genes and cells—became one of the central contributions of this thesis.

Notably, transformer-based models can also be used to infer putative GRNs directly from non-graph input data [73, 72], a capability that standard GNNs cannot achieve. This observation further reinforced our decision to build upon the transformer paradigm.

It is also important to note that this thesis does not specifically address perturbation response prediction, temporal dynamics, or spatial transcriptomics as primary modeling targets. While we demonstrate zero-shot generalization to spatial data and discuss perturbation prediction as a fine-tuning task, dedicated modeling of these modalities remains an important direction for future work.

Finally, while we managed to initiate some collaborations, fully achieving cross-disciplinary validation proved difficult—an unsurprising reality in the current landscape. This experience reinforced the need to make foundation models more accessible, which became one of the contributions of this thesis, represented not only in the effort to release easy-to-use open-source models but also in various side contributions and outreach efforts.

0.9.2 Scope of the Thesis

The limitations and opportunities described above progressively shaped the scope of this Ph.D. As we benchmarked existing single-cell foundation models and their claimed abilities, we encountered numerous shortcomings—ranging from poor usability and lack of reproducibility in pretraining to questionable architectural decisions and inconsistent evaluation practices. These observations led us to create our own model and, more broadly, to pursue a research program organized around the following axes:

1. **Benchmarking and evaluation.** We developed standardized, biologically grounded benchmarking suites (BenGRN, GRnnData) for GRN inference and contributed bench-

marks to the Open Problems platform, addressing the lack of rigorous and reproducible evaluation in the field.

2. **Reproducibility and accessibility.** We committed to fully open-source releases of all models, training code, datasets, and documentation, and deployed models on community platforms (CZ Virtual Cell Models, Superb.io) with tutorials and containerized benchmarks.

3. **Novel architectures.** We designed the Xpressor cross-attention compression mechanism for learning across biological scales (Chapter 2), criss-cross attention for subquadratic scaling, and GNN-based expression encoders for leveraging neighborhood information (Chapter 3).

4. **Improved training strategies.** We systematically evaluated pretraining tasks (denoising vs. masking), loss functions (MSE, ZINB, and hybrids), input representations (normalized vs. raw counts), and gene tokenization approaches through an additive benchmarking framework (Chapter 3).

5. **Zero-shot and generative capabilities.** We demonstrated zero-shot performance on denoising, cell-type classification, batch correction, and spatial transcriptomics, as well as counterfactual generation through the Xpressor architecture (Chapters 1 and 3).

6. **Applications to biological discovery.** We applied our models to real biological systems, including prostate tissue atlases and cross-species macrophage analysis, recovering known biology and generating testable hypotheses (Chapters 1 and 3).

7. **Understanding model limitations.** Through systematic ablations and cross-model comparisons, we characterized the conditions under which foundation models succeed or fail, informing future model development.

Improving scFMs to generate better representations of cells, genes, and their networks thus became the central objective of this Ph.D. We also sought to create benchmarks better suited to the single-cell genomics field, driven by real-life applicability rather than artificial metrics. Indeed, current methods often relied on synthetic data and ground truths unrepresentative of real biological systems, and the known single-cell standardized benchmarks were rarely used by early scFM papers.

0.10 Chapters Overview & Main Contributions

This thesis is structured around three main publications, each presented as a chapter. Below, we provide detailed summaries of our contributions and results.

0.10.1 Chapter 1: scPRINT: pretraining on 50 million cells allows robust gene network predictions

In this chapter, we present scPRINT (single-cell PRetrained Inference of Networks with Transformers), a large cell model designed for cell-specific gene network inference at the genome scale. This work addresses a fundamental challenge in cellular biology: inferring the network of molecular interactions that governs cell behavior.

Model architecture and training innovations. We trained scPRINT on more than 50 million cells from the CellxGene (CxG) database, representing approximately 80 billion tokens across multiple species, diseases, and ethnicities. Our model introduces several architectural innovations: (1) a protein-based gene encoding using ESM2 embeddings, which reduces parameters while enabling cross-species generalization; (2) a learned expression tokenization via MLP rather than hand-crafted binning; and (3) positional encoding of genomic location to capture co-regulation patterns. We designed three complementary pretraining tasks: a denoising task (transcript upsampling), a bottleneck learning task (embedding compression and reconstruction), and a label-prediction task with hierarchical classification for disentangled cell embeddings that represent different phenotypic facets.

Gene network inference methodology. A critical contribution is our method for extracting cell-specific gene networks from the transformer's attention matrices, inspired by similar approaches in ESM2 for protein contact prediction. We made this approach scalable to compute genome-wide networks for thousands of cells on commodity hardware. We also introduced an attention head selection mechanism that selects a subset of heads based on their correlation with known ground-truth networks, significantly improving network quality in larger models.

Comprehensive benchmarking framework. We created BenGRN and GRnnData, novel benchmarking suites for GRN inference that address the lack of standardized evaluation in the field. We benchmarked scPRINT against scGPT, Geneformer v2, DeepStructural Equation Modeling (SEM), and GENIE3 using multiple ground truth types: literature-based networks (Omnipath), cell-type-specific ChIP-seq/perturb-seq intersections (MCalla et al.), and genome-wide perturb-seq data. Our results demonstrate that scPRINT outperforms all other methods on most benchmarks. On the Omnipath benchmark across 26 cell types, scPRINT recovered 67% more connections than GENIE3 and showed superior enrichment for TFs and their ENCODE-validated targets (20% of TFs with significant enrichment, compared to 0% for scGPT). On the MCalla et al. cell-type-specific ground truth, scPRINT consistently outperformed all methods on both AUPRC and Early Precision Ratio (EPR) metrics.

Zero-shot capabilities on orthogonal tasks. Beyond gene network inference, we demonstrated that scPRINT's learned cell model enables competitive zero-shot performance on denoising, cell type prediction, and batch effect correction—without fine-tuning. For denoising, scPRINT matches State of the Art (SOTA) methods (MAGIC, KNNsmoothing2) on bulk populations and outperforms them on rare cell types where neighborhood-based methods fail. For cell type classification, scPRINT achieves 62% accuracy as a zero-shot predictor across 200+ cell types, outperforming marker-based methods like CellTypist. For batch effect correction, scPRINT achieves competitive scIB scores without using batch labels,

outperforming all methods that similarly do not require batch annotation.

Biological application and discovery. We applied scPRINT to an atlas of 83,000 cells from normal and Benign Prostatic Hyperplasia (BPH) prostate tissues. In rare switched memory B cells, we identified early TME markers, including BAG5, a known B-cell-associated prostate cancer marker. In fibroblasts, our gene networks revealed differential hub genes between normal and BPH-associated cells, recovering known biology around PAGE4 and uncovering interconnected pathways linking ion exchange, Extracellular Matrix (ECM) remodeling, oxidative stress, and chronic inflammation—hallmarks of premalignant states.

0.10.2 Chapter 2: Xpressor: Towards foundation models that learn across biological scales

In this chapter, we present Xpressor, a framework and architecture enabling cross-scale learning between biological foundation models. This work addresses a fundamental challenge: while foundation models exist at multiple biological scales (molecules, sequences, cells, tissues), they operate in isolation, unable to leverage the rich interconnections between scales.

Motivation and conceptual framework. We begin with a comprehensive review of foundation models across four biological scales: mFMs for atomistic molecular representations, nFMs for nucleotide and amino acid sequences (DNA, RNA, proteins), cFMs for cellular abundance profiles, and tFMs for tissue-level spatial organization. We argue that information flows between scales: lower-scale models (e.g., protein sequences) can improve input representations for higher-scale models (e.g., cells), while relationships learned at higher scales can inform lower-scale representations. Each scale's vocabulary can be seen as built from the compressed representations of the scale below—amino acids from atoms, genes from proteins, cells from genes, tissues from cells.

The Xpressor architecture. Our first contribution is a cross-attention-based compression mechanism called Xpressor that transforms high-dimensional gene-level representations into lower-dimensional cell-state vectors. The architecture introduces additional transformer blocks that perform cross-attention between the output embeddings of a foundation model and a set of learned latent tokens. It creates a bottleneck that compresses m gene tokens of dimension d_c into n cell tokens of dimension d_t, where $n \ll m$ and $d_t < d_c$. Critically, the same transformer can then decompress these cell representations back to gene-level predictions using cross-attention with gene ID tokens. This compression/decompression framework is grounded in the information bottleneck theory of Tishby et al., where the goal is to retain maximal information about relevant variables while achieving compression. We further regularize the latent space using contrastive losses between embedding dimensions and dimension-specific classifiers, ensuring each cell embedding dimension captures distinct biological information.

Multi-scale fine-tuning approach. Our second contribution is a method for fine-tuning lower-scale models using upper-scale tasks via adapter layers. We demonstrate this using ESM2 (a protein language model) as the lower-scale model and scPRINT as the upper-

scale model. Rather than simply using frozen ESM2 embeddings as gene tokens, we add a trainable MLP adapter that transforms each protein embedding during scPRINT's pretraining. We provide a formal proof that such an MLP has sufficient capacity to learn any arbitrary mapping—including acting as a lookup table that assigns each of D proteins to a unique learned output. This allows the adapter to enrich ESM2's representations (which encode protein sequence, evolutionary constraints, and structure) with co-expression information learned from millions of single-cell profiles.

Empirical results on the scPRINT benchmark gymnasium. We evaluate both contributions on three tasks from the scPRINT benchmark: cell-type prediction, embedding quality (scIB score for batch correction and biological consistency), and gene network inference (EPR on genome-wide perturb-seq and Omnipath ground truths). For the Xpressor architecture versus standard class-pooling (as used in scGPT), we observe substantial improvements: cell-type prediction accuracy increases from 0.60 to 0.72 (+20%), and embedding quality improves from 0.48 to 0.52 (+8%), while gene network inference remains comparable. For multi-scale fine-tuning, comparing frozen ESM2 embeddings versus fine-tuned ones, we see cell-type prediction improve from 0.60 to 0.70 (+17%), embedding quality from 0.48 to 0.49, and gene network inference improves on the Omnipath benchmark from 2.0 to 2.4 EPR (+20%). Notably, fine-tuned ESM2 embeddings outperform both frozen ESM2 and randomly initialized embeddings across nearly all metrics.

0.10.3 Chapter 3: scPRINT-2: Towards the next-generation of cell foundation models and benchmarks

In this chapter, we present scPRINT-2, a next-generation single-cell foundation model whose design decisions were systematically validated through an unprecedented additive benchmarking framework. This work addresses the critical gap in the field: while many scFMs have been proposed, the relative importance of their architectural choices, training strategies, and data modalities has never been rigorously assessed in isolation.

The additive benchmark: a systematic evaluation framework. We designed a comprehensive benchmark to evaluate 42 different configurations of scFM components, including pretraining databases, architectures, and training tasks. Each model variant was trained 6 times across multiple seeds to generate statistical error bounds, and evaluated on a gymnasium of tasks: cell-type classification, batch correction (scIB scores), expression denoising, and gene network inference. Our benchmark revealed several key findings: (1) denoising is superior to masking as a pretraining task for classification and embedding quality; (2) un-normalized expression outperforms normalized input; (3) ESM-based gene tokens significantly outperform learned embeddings from scratch; (4) genomic location encoding improves model convergence; (5) MSE loss outperforms ZINB on average, but a hybrid ZINB+MSE loss provides the best balance between accuracy and expressivity; and (6) model size correlates with improved gene network inference and cell-type prediction.

The scPRINT-2 corpus: the largest single-cell database to date. We assembled a pretraining database of over 350 million cells from 16 eukaryotic organisms spanning more than one billion years of evolution. This corpus integrates data from CxG, the Tahoe-100M

dataset, and the scBasecount database (20,000 reprocessed GEO datasets), totaling 25 TB of unique data with approximately 400,000 distinct genes and 4,764 different cell labels across 140,000 cell groups. We demonstrated that cell-state diversity and data quality are more important than sheer cell count — reducing to 200 human datasets caused only a minimal performance decrease, whereas using low-diversity datasets alone caused performance to plummet. We introduced cluster-weighted sampling and Number of Non-Zeros (NNZ)-weighted sampling to address dataset imbalances, enabling effective training on this heterogeneous corpus.

Architectural innovations. scPRINT-2 incorporates 12 distinct contributions validated through our benchmark. Key innovations include: (1) the XPressor architecture, a cross-attention-based compression mechanism that transforms gene-level representations into cell-level tokens and back, enabling the model to be generative; (2) a GNN-based expression encoder that leverages neighborhood information from similar cells or spatial neighbors; (3) criss-cross attention, a sub-quadratic attention mechanism inspired by Recurrent Interface Networks that dramatically improves training speed while retaining model capabilities; (4) VAE-based compression with dissimilarity losses between cell tokens, improving batch correction; and (5) an updated hierarchical classification loss that penalizes predictions based on ontological distance rather than binary correctness.

State-of-the-art performance across benchmarks. On the Open Problems benchmark (November 2025), scPRINT-2 achieved 75% zero-shot cell-type classification accuracy, outperforming scPRINT-1 (47%) and all other zero-shot scFMs (40-60%). With our XPressor-based Parameter-Efficient Fine-Tuning (XPEFT), scPRINT-2 surpassed every existing supervised and unsupervised method on the platform. For expression denoising, scPRINT-2 became state-of-the-art, outperforming MAGIC across all tested contexts, with particularly strong improvements on low- and mid-quality datasets where the GNN encoder can leverage neighbor information. For batch integration, scPRINT-2's zero-shot performance exceeded all other methods, and fine-tuned performance achieved the best overall scIB scores.

Generalization to unseen modalities and organisms. We demonstrated scPRINT-2's ability to generalize beyond its training distribution. On Xenium spatial transcriptomics data (a modality absent from training), scPRINT-2 successfully denoised expression, imputed 5,000 unseen genes with correlation scores matching denoised genes, and produced biologically meaningful cell-type and disease predictions. On cat and tiger lung tissues (organisms not seen during training), scPRINT-2 achieved 42% cell-type classification accuracy across 500 possible labels, with differential expression analysis confirming that scPRINT-2 sometimes corrected expert annotations. With cluster-based logits averaging and XPEFT fine-tuning, accuracy improved to 95%.

Counterfactual reasoning and generative capabilities. The XPressor architecture enables scPRINT-2 to perform counterfactual generation. We demonstrated this by replacing organism-specific cell embeddings from mouse cells with human embeddings to generate "humanized" mouse expression profiles. The Wasserstein-2 distance between these counterfactual profiles and real human cells decreased significantly, and over-representation analysis showed 58% enrichment in correctly predicted differentially expressed genes. Pathway analysis revealed biologically meaningful differences in immune function, membrane-ECM interactions, and tissue elasticity.

Gene embeddings and network inference. We showed that the XPressor architecture produces output gene embeddings with meaningful biological clustering (enriched for known pathways), whereas standard transformers without XPressor produce embeddings that merely encode expression values. For gene network inference, we introduced a computationally intensive extraction method biased toward co-expressed genes. Benchmarking against six ground-truth networks (including the cellmap Affinity Purification Mass Spectrometry (AP-MS) data, human interactome, and Genome-wide Perturb-seq (gwps)), scPRINT-2 showed improved performance on odds-ratio metrics. We demonstrated cross-species gene network analysis in macrophages, identifying conserved hub genes involved in feroptosis, pathogen phagocytosis, and MHC pathways. We also showed how scPRINT-2's predictions can cross-validate PPI ground truths, identifying connections (HLA-DRA/CD74, B2M/B2M) that RoseTTAFold2-PPI missed but AlphaFold-Multimer confirmed.

0.10.4 Impacts Beyond Publications

Beyond the three main publications, this thesis produced several additional contributions:

- The first ones are six open source Python packages named:

  scPRINT: where the model is made available, together with training scripts and notebooks to use the model, functions to download and preprocess data, and more. https://github.com/cantinilab/scPRINT

  scPRINT-2: where the second model is made available, together with training scripts and notebooks to use the model, functions to download and preprocess data, and more. https://github.com/cantinilab/scPRINT-2

  scDataloader: a package to load thousands of large single-cell datasets efficiently, with preprocessing, filtering, and loading options. It also allows a first-of-its-kind efficient weighted random sampling over billions of elements. https://github.com/jkobject/scDataLoader

  Bengrn: a package to benchmark GRN inference methods on single-cell data, using multiple types of metrics and ground truth networks. https://github.com/jkobject/Bengrn

  GRNNdata: a package to work with gene regulatory networks and single-cell data jointly, using the Anndata format. https://github.com/cantinilab/GRNNdata

  Xpressor: a package to reproduce the second paper's experiments and create an Xpressor model from scratch. https://github.com/cantinilab/XPressor

  Simpler Flash: Initially a package to facilitate the use of flash attention before it became part of the pytorch implementation itself. It now includes multiple types of efficient attention mechanisms, such as softpick-flash and our flash criss-cross attention mechanism. https://github.com/jkobject/simpler_flash

  Hierarchical Classifier: a package to implement hierarchical classification for single-cell data. https://gist.github.com/jkobject/5b36bc4807edb440b86644952a49781e

- Another contribution, as previously mentioned, is around accessibility. Not only did I release model weights and inference code, but I also provided easy-to-use inference tools, pretraining methods, and datasets, training traces, and documentation. Moreover, tutorials were implemented in Google Colab, and versions of the models got released on the Chan–Zuckerberg model hub (https://virtualcellmodels.cziscience.com/) and Superb.io's platform (https://superbio.ai).

- Finally, I implemented Docker containers for scPRINT, scGPT, and Geneformer for benchmarking on the open problems platform, and participated in creating and improving two benchmarks on the platform.

- In addition to publication, I wrote a blog post with Lamin.ai on training on large datasets. I also wrote with multiple x-plainers on X, LinkedIn, Bluesky, and Threads, as well as my personal website, to share some of our findings with a possibly wider audience. Similarly, I wrote vulgarisation articles for the Pasteur Institute's and CNRS's websites and created one of the most viewed videos on the Pasteur Institute's YouTube and Instagram accounts, presenting my work. I was also highlighted on Whitelab's blog posts and released four YouTube videos featuring diverse presentations of my work.

- Other outreach efforts were done through conference presentations and invited talks with:

  A participation in three international ML conferences

  Over 25 invited talks and five poster presentations.

- Finally, I also contributed to the European start-up ecosystem by translating work from Academia to Industry. I joined a worldwide organization called Nucleate to help master's students, PhDs, and Post-docs translate their research into start-ups. I also worked as a consultant for four start-ups: Whitelab Genomics, Biographica, Blossom, and dot-omics, assisting them in developing strategies for foundation models in single-cell RNA-seq and DNA sequencing.

scPRINT: pretraining on 50 million cells allows robust gene network predictions

1.1 Summary

A cell is governed by the interaction of myriads of macromolecules. Inferring such a network of interactions has remained an elusive milestone in cellular biology. Building on recent advances in large foundation models and their ability to learn without supervision, we present scPRINT, a large cell model for the inference of gene networks pretrained on more than 50 million cells from the CxG database. Using innovative pretraining tasks and model architecture, scPRINT pushes large transformer models towards more interpretability and usability when uncovering the complex biology of the cell. Based on our atlas-level benchmarks, scPRINT demonstrates superior performance in gene network inference to the SOTA, as well as competitive zero-shot abilities in denoising, batch effect correction, and cell label prediction. On an atlas of BPH, scPRINT highlights the profound connections between ion exchange, senescence, and chronic inflammation.

1.2 Introduction

Understanding the cellular mechanism is considered a milestone in biology, allowing us to predict cell behavior and the impact of drugs and gene knock-outs[88, 89, 90, 91, 92]. A cell is regulated by a complex interplay of myriads of macromolecules that define its state. We can simplify these interactions via a GN[16] (GN). Many approaches have been developed to infer these networks, focusing on TF-to-gene links using single-cell omics data modalities like scRNA-seq and scATAC-seq[93, 94, 95, 96, 80, 97, 83, 98, 99, 100, 101, 102]. This gene network subset regulating the cell gene expression levels is often called a GRN. However, many other gene products than TFs impact RNA abundances in the cell, like RNA-RNA and protein-TF interactions[103, 104, 105, 106, 107]. Most GRN inference methods do not

scale to the number of genes and cells present in single-cell RNA datasets, and they need many cells, thus impairing their ability to reconstruct cell-state-specific networks. Other methods consider datasets where differentiating cells can be ordered temporally to predict more causal GRNs. While this approach is interesting, temporal ordering is often hard to predict[98, 108].

Benchmarks like BeeLine[109] and MCalla et al.[110] have shown that despite the existence of many methods, GN inference remains a challenging problem. Indeed, it is underconstrained and has limited prior knowledge. New foundational models trained on tens of millions of measurements could help solve these difficulties. Transformers like BERT[66, 57] have gained traction in computational biology and have held promise to learn a model of the cell that would translate across many tasks of cellular biology, such as cell type annotation, batch-effect correction, perturbation prediction, and gene network inference[72]. Among them, scGPT[73] got much attention, proposing a novel encoding of genes and their expression, a new pretraining methodology similar to autoregressive pretraining in language models, and the possibility of extracting GRN from its model (see 1.5. methods).

Inspired by these efforts, we propose scPRINT (single-cell PRetrained Inference of Networks with Transformers), a foundation model designed for gene network inference. scPRINT brings inductive biases and pretraining strategies better suited to GN inference while answering issues in current models (see Supplementary Table 5.1.1). scPRINT outputs cell type-specific genome-wide gene networks but also generates predictions on many related tasks, such as cell annotations, batch effect correction, and denoising, without fine-tuning.

We extensively benchmark scPRINT on challenging gene network inference tasks, from literature-based networks to cell type-specific ones generated via orthogonal sequencing methods. We show that scPRINT outperforms the SOTA on most of these atlas-level benchmarks. In addition, our model focused on GN inference, is also competitive on a compendium of tasks like denoising, cell type prediction, and embedding with batch effect correction. This suggests that by learning a cell model, scPRINT gains zero-shot abilities in many tasks of cellular biology. We use scPRINT to analyze an atlas of normal and senescent prostate tissues where we identify rare cell populations with early markers of the TME in B-cells. In fibroblasts, we study gene networks and recover known hubs such as PAGE4, linking the senescence of fibroblasts to changes in the ECM and downstream inflammation. We find key interconnected pathways of the oxidative stress response and extracellular matrix building via metal and ion exchange in the gene network of BPH-associated fibroblasts. We also show that healthy and disease-related cells exhibit different network patterns, demonstrating that scPRINT can help identify novel pathways and targets while considering them in their specific cellular and molecular contexts.

scPRINT[111] (https://github.com/cantinilab/scPRINT) is a fast and open-source tool that can be readily integrated into the bioinformatics pipeline. We make public the code and model weights, but also the pretraining strategies, datasets, and our own dataloader for use with vast training sets like the CxG databas[112]. We also release a Gene Network benchmarking suite: *BenGRN*[113] and *GrnnData*[114].

1.3 Results

1.3.1 scPRINT: a scRNAseq foundation model for gene network inference

We propose scPRINT (Figure 1.1A), a SOTA bidirectional transformer designed for cell-specific gene network inference at the scale of the genome. scPRINT is trained with a custom weighted-random-sampling method[115] over 50 million cells from the CxG[112] database from multiple species, diseases, and ethnicities, representing around 80 billion tokens (see 1.5. Methods). We train scPRINT at various scales (from 2M to 100M parameters) and very efficiently by using flashattention2[116], e.g., only requiring an A40 GPU for 48 hours to train our medium model, significantly reducing the barrier to entry for any computational biology lab (see Supplementary Table 5.3).

To push scPRINT to learn meaningful GNs and their underlying cell model, we design a unique set of pretraining tasks, as well as expression encoding and decoding schemes (Figure 1.1B).

scPRINT's pretraining is composed of three tasks which loss are added and optimized together: a denoising task, a bottleneck learning task, and a label prediction task. The objective is to let scPRINT learn to represent meaningful gene connections while also endowing it with a breadth of zero-shot prediction abilities.

Indeed, similarly to ADImpute[117, 40], we expect a good gene network to help denoise an expression profile by leveraging a sparse and reliable set of known gene-gene interactions.

We implement this denoising task as the upsampling of transcript counts per cell (see 1.5. Methods). While most other methods have been using masking as a pretraining task, our method is related to the downsampling and masking task of scFoundation[77]. We show that this strategy performs better than masked language modeling and gives scPRINT the ability to upsample any expression profile.

In addition, we expect that a cell model tasked to compress expression profiles into embeddings can learn the regularities of modules and communities of gene networks. Therefore, the bottleneck learning task drives scPRINT to generate an embedding and a cell expression profile from its embedding only. The embedding is generated by scPRINT and is used again, this time without the cell expression values, to regenerate the true profile (see 1.5. Methods).

Finally, the cell's gene network should represent the cell state and its different phenotypic facets. Effectively, scPRINT generates not just one embedding per cell but multiple. A hierarchical classifier is then applied to each distinct cell embedding to predict its associated class, such as cell type, disease, sex, organism, ethnicity, and sequencing platform. The embeddings thus become disentangled, each representing a specific facet of the cell state[118]. This last training task pushes the large cell model and its gene network to represent the cell state.

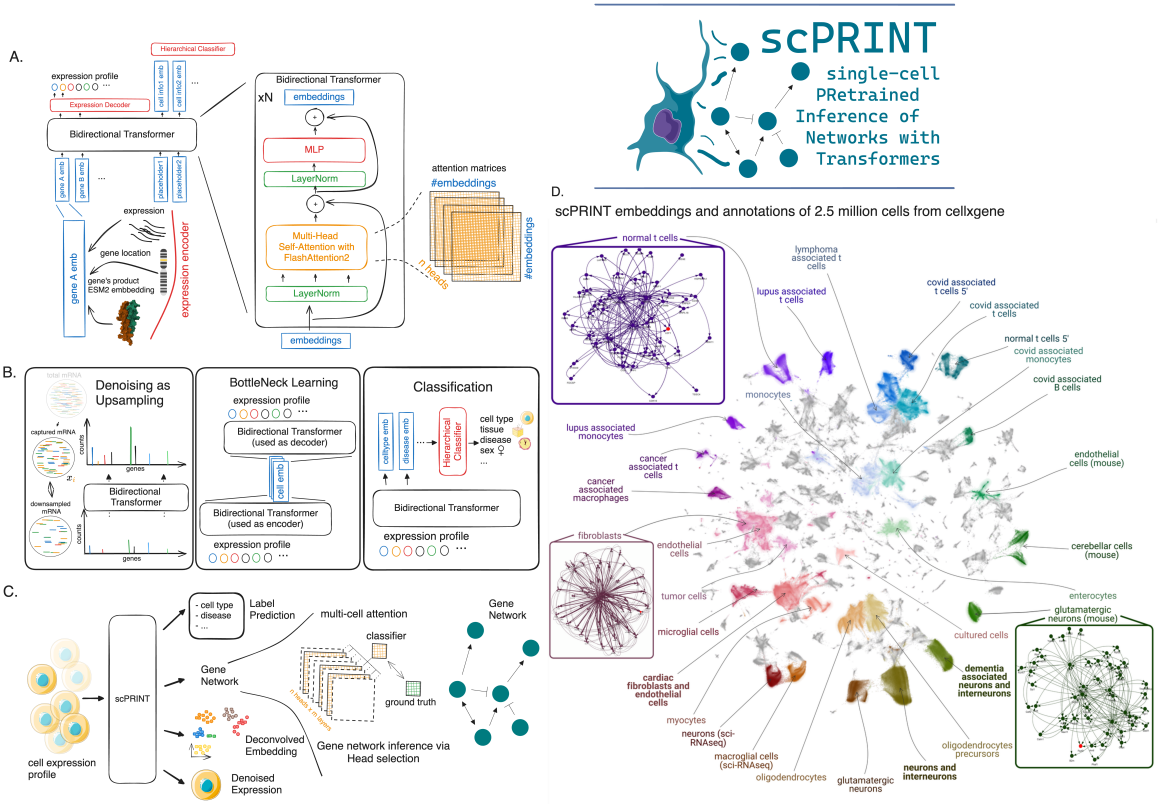


Figure 1.1: Presentation of the scPRINT model and training. (a) Schematic representation of scPRINT with its bidirectional encoder, gene expression embedding encoding via gene location, matched protein ESM2 embedding, and gene expression. (b) scPRINT pretraining tasks: Denoising task whose goal is to recover the known transcriptomic profile from a purposefully downsampled expression profile. Bottleneck learning reconstructs the expression of requested genes using only their cell embedding. The same model is used for both The encoding and decoding steps. Hierarchical classification is achieved by applying a hierarchical classifier to each disentangled embedding. This pushes the first embedding to contain cell type info, the second embedding to contain disease info, and so on (see 1.5. methods). (c) The different outputs in scPRINT. scPRINT generates label predictions of cell type, tissue, disease, sex, sequencer, ethnicity, and organism. scPRINT generates multiple embeddings (which we call disentangled embedding), a general one, as well as a specific embedding for each class. scPRINT also generates a reconstructed expression profile at any requested sequencing depth (i.e., total transcript count) (denoising). scPRINT also generates a Gene Network by selecting and combining various attention heads into a gene x gene matrix. (d) Example of a scPRINT output from a random subset of 2.5 million cells from the CxG database. Embeddings and labels are generated by scPRINT, together with the example cell type-specific gene networks. We show only subparts of the networks extracted from a central node, represented in red.

Thanks to the CxG database requirement for complete annotations and our innovative hierarchical classifier, we have added label prediction as part of the pretraining of scPRINT. While the assumption is that in other modalities, the scarcity and noisiness of such labels make it infeasible, we show that this approach is a net positive in our case (see Supplementary Table 5.4; 1.5. Methods). Indeed, it helps us disentangle the various cell embeddings and performs zero-shot predictions on unseen datasets. These disentangled embeddings are opening a future possibility to perform counterfactual generation: mixing embeddings representing different facets of cell states, e.g., fibroblast + cancer + pancreas tissue + female,

to generate novel unseen expression profiles.

scPRINT converts the gene expression of a cell to an embedding by summing three representations or tokens: its id, expression, and genomic location (Figure 1.1A, see 1.5. Methods). scPRINT encodes the gene IDs using protein embeddings. This gene representation is made using the ESM2[119] amino-acid embedding of its most common protein product (see Supplementary Figure 5.2.1). First proposed in UCE[76], the model learns to leverage representations that can potentially apply to unseen genes and species, using the structural and evolutionary conservation of the sequence encoded by ESM2. While drastically reducing the number of weights used in the model compared to scGPT and Geneformer (see 1.5. Methods), this representation also contains some priors needed to infer protein-protein[120] interactions (Figure 1.1A).

The gene's expression is tokenized via a MLP using log-normalized counts. This MLP lets the model learn a metric behind gene expression, whereas scGPT and Geneformer apply a specific prior for the encoding of their gene expression (see 1.5. Methods).

Finally, we help the model know that genes with similar locations tend to be regulated by identical DNA regions, using the positional encoding of their location in the genome (see 1.5. Methods).

These three embeddings are summed and then concatenated across all the genes expressed in a cell together with additional placeholder cell embeddings to form the transformer model's input.

scPRINT is pretrained using 2,200 randomly selected expressed genes in a cell profile. If a cell doesn't have enough expressed genes, the list is padded with randomly selected unexpressed genes. A context of 2200 genes, while not genome-wide, captures all the expressed genes in more than 80% of the cell profiles in the CxG database. We also show that scPRINT can make predictions on much larger sequences of genes at inference time without using attention approximation methods[63].

Using unexpressed genes, combined with the denoising task, let scPRINT discriminate the true zeros from dropouts in scRNA-seq47. The expression decoder of scPRINT further helps model this statistic of the data. It is a zero-inflated negative binomial graphical model inspired by previous literature in single-cell RNA-seq modeling48. Here, the loss (also used for bottleneck learning) is thus the log-likelihood of the gene expression given the distribution parameters.

As shown in Figure 1.1C, at inference time, scPRINT can generate multiple outputs across any scRNA-seq-like cellular profile of various mammalian species without fine-tuning. Figure 1.1D shows scPRINT's prediction at the scale of an atlas of 2M randomly sampled cells from CxG. From its pretraining, scPRINT performs denoising, label prediction, and cell embedding without fine-tuning. However, a critical emergent output of scPRINT is its cell-specific gene networks. Following a similar approach to ESM2, we generate cell-level gene networks via the bidirectional transformer's input-wise weighted matrices, called attention matrices -or heads-. They represent general gene-gene connections and can be subsetted to TF-gene connections (i.e., GRNs). Remarkably, we made this approach scalable enough to compute attention heads-based gene networks for 1 to 10,000 cells, at the genome

scale, with commodity hardware and in a few minutes. These networks both showcase the ability of scPRINT to model cellular biology and help make it a more explainable tool for the community, showing the network assumptions made during inference. The attention heads are either all aggregated by averaging or can be selected to better reflect connections of interest (Figure 1.1C). This is done using the average of the heads most correlated with literature or perturbation-based ground truth networks. Finally, while we do not assess scPRINT's ability to model inhibition due to the scarcity of such annotations, we leave open the possibility of using our head selection technique for such a task.

Similarly to what has already been done in ESM2 and the Large Language Model literature[121, 122, 123], we deeply investigate the meaning of attention matrices in the context of cellular biology, an aspect under-studied in the literature of foundation models applied to genomics.

In the following sections, we benchmark scPRINT on gene network inference against scGPT, DeepSEM[102], GENIE3[124], and Geneformer v2[125], the updated version of Geneformer. scGPT and Geneformer v2 are highly cited and published transformer models for single-cell scRNA-seq, mentioning the inference of gene interactions[73]. DeepSEM is an autoencoder model jointly learning its weights and a gene network matrix. GENIE3 generates networks via regression by finding the set of genes that best predict another gene's expression. It is one of the top-performing and most used methods for GRN inference (see 1.5. Methods). However, it suffers from very long run times and high memory requirements (see Supplementary Table 5.1.4).

1.3.2 scPRINT recovers biological features in its gene networks

We benchmark scPRINT against the SOTA based on whether their recovered networks contain meaningful biological knowledge. We consider two main benchmarking methodologies, one using a simulated expression profile from a well-established biological network. Because simulated data does represent real cell expression data (see 1.5. Methods), our second and main approach focuses on biological features of a network inferred from real cell expression profiles. Indeed, we assume that a meaningful gene network should have some of its hub nodes being TFs. TFs should be more connected to their known target, on average. We should recover known gene-gene connections and expect enrichment of cell type-specific marker genes in the network.

We compare each gene network inference method's ability to recover a known network from 1000 simulated single-cell scRNA-seq expression profiles generated by the Sergio Ordinary Differential Equation (ODE) model[126] from the ground truth network Regnetwork[127] (see 1.5. Methods). Only scPRINT was able to recover meaningful connections (see Supplementary Table 5.1.5). One explanation is that through its training, scPRINT has learned the common gene connections that also exist in the RegNetwork ground truth.

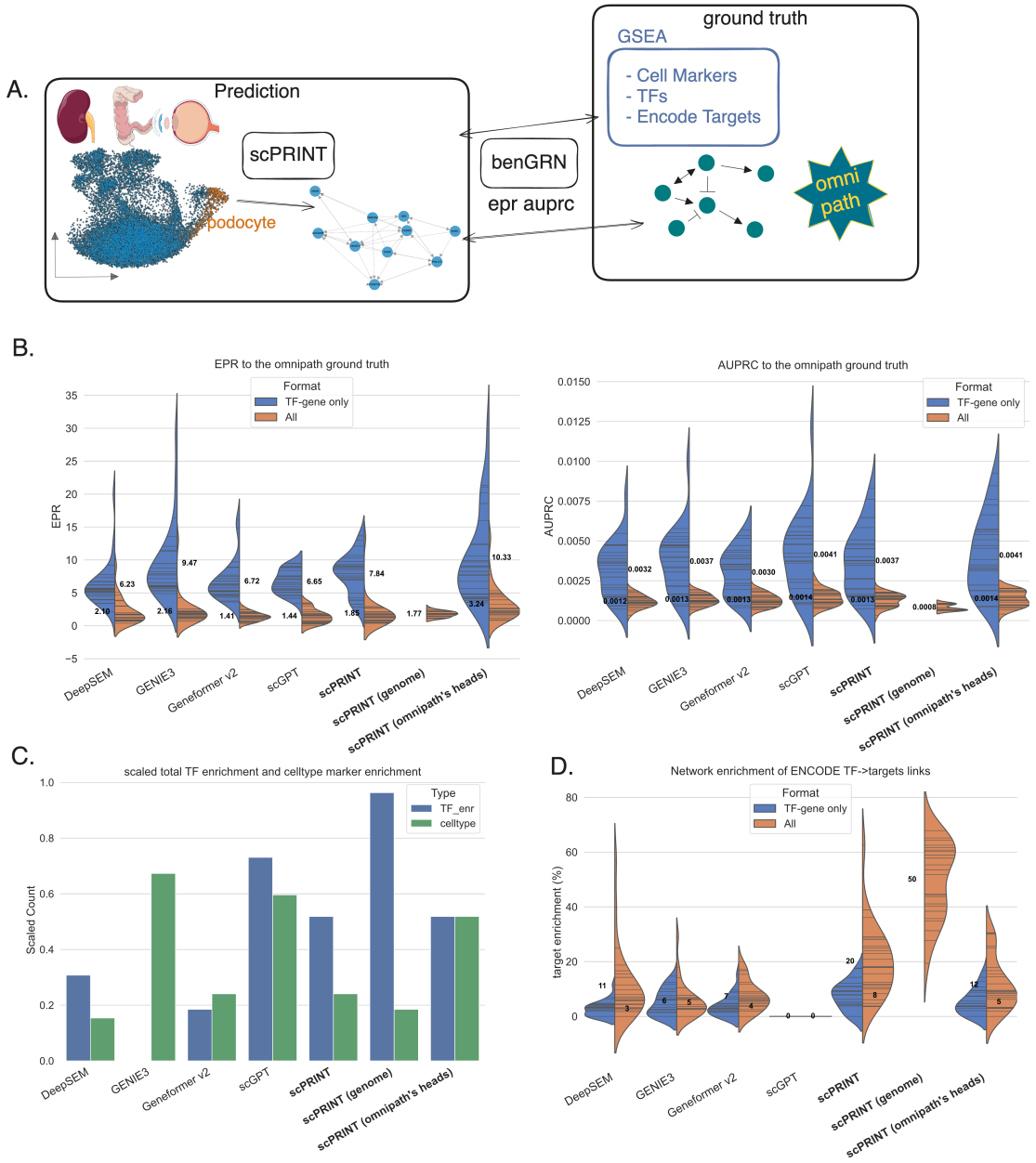


Figure 1.2: Analysis of the gene networks generated by scPRINT. (a) We extract cell type-specific gene networks for each cell type in the dataset (n=26 cell types across 3 datasets). We perform Gene Set Enrichment Analysis (GSEA)[128] on the network's nodes (n=4000 genes). We compute the ability of the edges to recover the Omnipath ground truth's connections. (b) Violin plot of the ten different AUPRC and EPR values obtained when comparing the inferred cell type-specific networks with the Omnipath network for scPRINT: average of all attention heads, scPRINT (genome): same scPRINT version but computing a genome-wide gene network, scPRINT (omnipath's heads): same scPRINT version but with attention heads selected using a subset of omnipath, scGPT, DeepSEM, Geneformer v2, and GENIE3, when considering only TF-gene connection or all gene-gene connections. (c) Violin plot of the average number of TF with enrichment for their ENCODE target in each cell-type-specific network. (d) Number of GNs with a significant enrichment of TFs and of their cell type's marker genes.

On gene network inference from real expression data, we noticed that depending on cell

type and datasets, the different tools could vary greatly in the similarity of their GNs to the Omnipath[129] ground truth. Because of this, we focused our benchmark on three randomly selected test datasets of kidney, retina, and colon tissues comprising 26 cell types[130, 131, 33] (see 1.5. Methods, per dataset results in Supplementary Figure 5.2.2). Of note is that we could not determine if these datasets were used during the training of scGPT or Geneformer.

We build one network per cell type, using the same 1024 cells and their 5000 most differentially expressed genes for all benchmarked methods. We evaluate the quality of the networks based on their overlap with Omnipath. We also compute the network enrichment for cell type markers, TFs, and ENCODE TF targets using the prerank[128] algorithm (Figure 1.2A).

Although the scGPT code mentions GRN inference only using perturb-seq data, we reapply the same method without the perturbation-baseline comparison. This is to make it comparable with other benchmarked methods and because most of our datasets are not perturbation-based. Similar to what is presented in its paper, we use the mean of the attention matrices across cells and the four attention heads of the last layer of the human pretrained model. We retain this method across our benchmarks for scGPT (see 1.5. Methods). We apply a similar strategy for Geneformer (see 1.5. Methods).

For scPRINT, we generate three network versions: one simply called scPRINT, based on the average of all heads in the model. scPRINT (omnipath's heads), based on the average of heads selected with our abovementioned head selection method inspired by ESM2, and scPRINT (genome), which is like the scPRINT network but uses our method to generate genome-wide networks (see 1.5. Methods) instead of using the 5000 most differentially expressed genes. Indeed, in transformer models, the choice of attention heads is important. Although transformers can learn the causal structure of their input, it has been shown that some attention heads, especially in larger networks, can become unused, containing predominantly random connections[132]. Some work has been done at pruning these heads[133] or forcing a head selection mechanism during inference and training[134]. For scPRINT (omnipath's heads), we select heads based on a linear classifier's prediction of the best set of heads to predict a subset of Omnipath (see 1.5. Methods). Similarly to the scPRINT network, these heads are then averaged to generate the scPRINT (omnipath's heads) gene network. To perform this selection, we split the omnipath dataset into train/test and select heads, using 50% of the ground truth and only the first cell type of each dataset. We then use the same combination of heads across all other cell types. This shows that our selection process builds consistent networks across cell types and parts of the ground truth. This innovative approach contrasts with previous ones like scGPT's and GENIE3 by using part of an available ground truth to select heads.

First, we look at how much information from Omnipath is contained in the inferred networks. Omnipath contains around 90,000 curated gene-gene connections, mainly from the literature. These connections are cell type agnostic, and most are TF - gene. On this benchmark, we evaluate the networks based on AUPRC and EPR, two metrics often used in GRN benchmarks[109] (see 1.5. Methods), where we define our task as a binary classification of connections on all gene-gene pairs. Due to the row-wise normalization of networks generated by all methods, and because Omnipath has many sources with only a few targets (see Supplementary Figure 5.2.3), we here use the transpose of our inferred networks when

making comparisons with Omnipath (see 1.5. Methods). In Figure 1.2B, we can see that scPRINT (omnipath's heads) outperforms all methods on average across all cell types. While scPRINT (omnipath's heads) uses some ground truth information to select its head, we see that scPRINT still outperforms scGPT and Geneformer v2 on the EPR metric, showing that its top predicted edges more closely match the ground truth.

AUPRC results are very low overall because we do not expect most Omnipath connections to be present in the cell type's gene network, as many connections in Omnipath might only be true in some cellular contexts. Moreover, we do not expect most connections in our generated network to exist in Omnipath as it only contains a small fraction of all real gene-gene connections. Although overall AUPRC values are small, we can see that both scGPT and scPRINT outperform the other methods in the number of connections recovered. Indeed, on average, scGPT and scPRINT respectively recover 42% and 67% more connections than GENIE3.

However, GENIE3 is often used by biasing the method to only predict TF-gene connections (see 1.5. Methods). This type of network, usually called a GRN, is most often used, given the importance of TFs in regulating gene expression. To compare the other methods to this GRN version of GENIE3, we also use a GRN version of their networks by subsetting them to TF-gene connections only. In this context, all the methods significantly improve their predictions without altering their relative performances (Figure 1.2B). This is unsurprising, considering that Omnipath is strongly biased towards TF-gene interactions.

Interestingly, we have seen that smaller scPRINT models containing fewer heads perform better when taking the average of their heads. In contrast, head selection is often more advantageous in larger models with more heads (see Supplementary Table 5.1.7). As presented at the beginning of the results section, it might be that as models become larger and less regularized, some heads tend to become unused and contain mostly noise. As a consequence, a head selection is advantageous in larger models.

We also expect biologically meaningful gene networks to have their central nodes enriched for TFs. In addition, because these networks are cell type-specific, we expect their central nodes to be enriched for some marker genes of their associated cell types (see 1.5. Methods). In this regard, both scGPT and scPRINT achieve very similar and strong network enrichment for TFs compared to GENIE3, DeepSEM, and Geneformer v2, whose networks are not enriched for TFs (Figure 1.2C).

Moreover, amongst the 178 cell types we have marker gene sets for in pangaloDB[135], all methods find some enrichment, especially GENIE3 and scGPT (see 1.5. Methods). We notice that selecting heads based on Omnipath significantly improves scPRINT's network enrichment for cell-type markers. Of note, our goal is not to annotate cell types from the gene network but mainly to showcase the network's cell type specificity.

Finally, we also examine how much the connections of each TF are enriched for that TF's target. Here, scPRINT overperforms all other methods (Figure 1.2D). In the scPRINT networks, 20% of the Transcription Factors for which we have data on ENCODE have connections significantly enriched for their ENCODE-validated gene targets[136]. Interestingly, only our large cell model achieved a great performance, and scGPT did not display any enrichment across the 26 cell types assessed. While we acknowledge that ENCODE is used

in the Omnipath database, we cannot expect Omnipath to represent the ENCODE targets. Indeed, it combines and processes 57 additional data sources to build its consensus network.

scPRINT (genome) has been added despite its performance not being comparable to other methods. Indeed, comparing its overlap with Omnipath is unfair as it includes many more genes and connections, many of which will have almost no data on this ground truth. While scPRINT (genome) showcases our ability to generate genome-wide networks, it also shows strong performances in TF enrichment and ENCODE TF-target enrichments. This highlights that even at such a large scale, networks generated by scPRINT are enriched in biological knowledge gained solely from its pretraining tasks.

Overall, we have shown that scPRINT generates, in one forward pass, cell type-specific gene networks that are biologically meaningful. We will now examine them using cell type-specific ground truths extracted from orthogonal experiments.

1.3.3 scPRINT outperforms the state of the art on cell type-specific ground truths

Although we have shown that our networks represent meaningful biology, the Omnipath ground truth is literature-based and not cell type-specific. Here, we use two different modalities, perturb-seq[34], and ChIP-seq[137], as ground truths to compare predicted gene networks against.

In the MCalla et al.[110] ground truth, ChIP-sequencing and perturb-seq are intersected to get at the small subset of possibly direct connections between TFs and genes for both human and mouse embryonic stem cells (ESC) (Figure 1.3A, see 1.5. Methods). We have seen that these ground truth networks show a different pattern than literature-based networks (see Supplementary Figure 5.2.3). Some TFs regulate only a few genes, whereas others are highly connected.

To generate our networks, we use as input one human and two mouse ESC scRNA-seq datasets from MCalla et al. with the addition of another human dataset from Yan et al.[138] Networks are generated over the same 1024 cells, and the 5000 most variable genes for all methods. For scPRINT, three networks have been generated: one averaging all the attention heads (scPRINT), one averaging heads selected based on how well they predicted Omnipath ground truth data: scPRINT (omnipath's heads), and one averaging heads selected from one of the MCalla ground truths: scPRINT (Han et al.'s heads). For more details, see the results section 2: scPRINT recovers biological features in its gene networks. Of note, due to the small amount of genes assessed in the ground truth, we do not add the genome-wide network version here. Moreover, only the TF version of GENIE3 and the TF-gene subsets of the other method's networks are used since the ground truth only contains TF-gene connections.

Contrary to Omnipath, some elements in these biological networks are highly connected, whereas many others display no connections. This imbalance means that a method predicting only the highly connected TFs will perform well on the MCalla et al. benchmark. As a consequence, we are not transposing the attention matrix as done in the previous section.

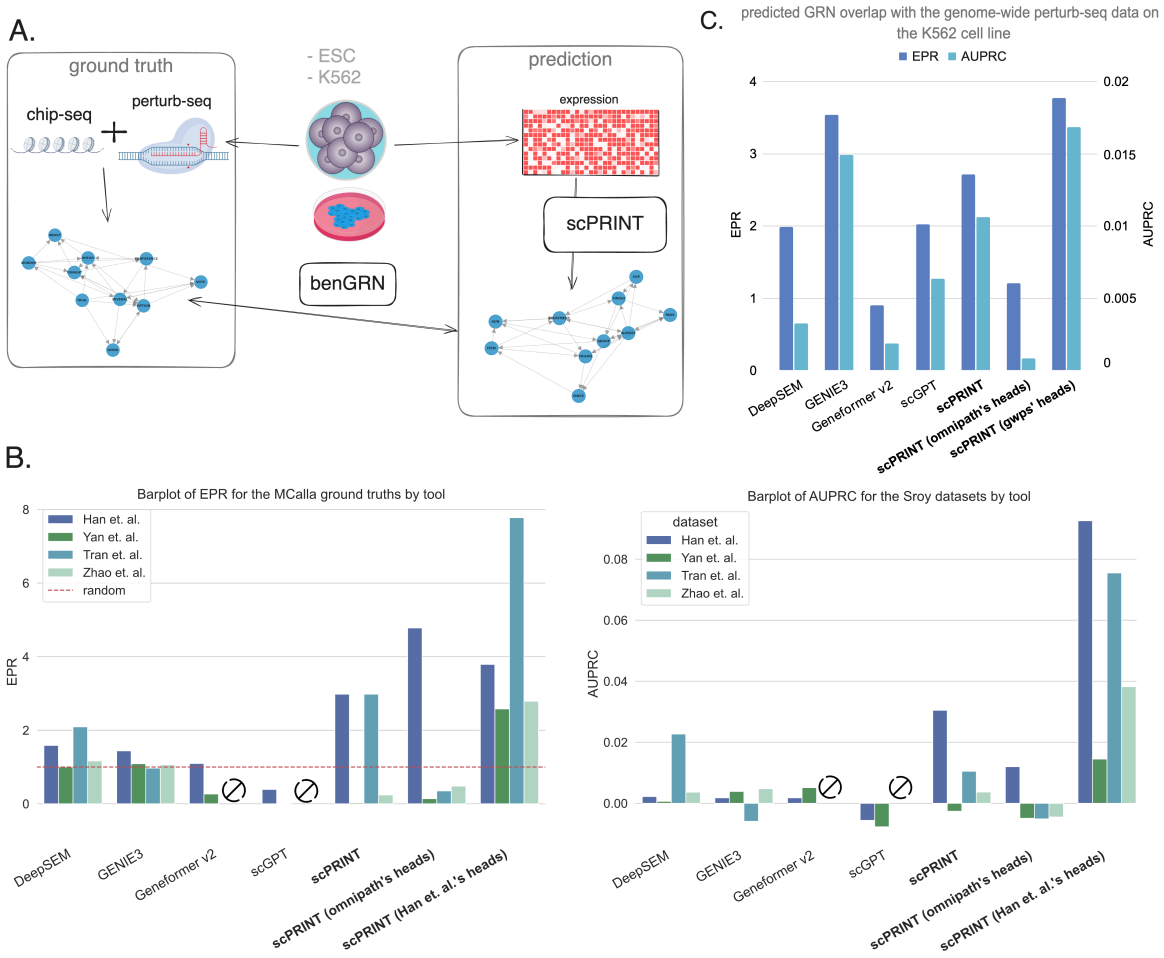


Figure 1.3: scPRINT GN inference performance on cell-type specific ground truths. (a) The ground truths are generated via orthogonal sequencing assays on the same cell type. ChIP-seq and perturb-seq are intersected for the MCalla et al. dataset on human (hESCs) and mouse (mESCs) Embryonic Stem Cells, whereas perturb-seq on the K562 cell line is only used for the genome-wide perturb-seq ground truth. (b) Performance of scPRINT, scPRINT (omnipath's heads): same scPRINT version but with attention heads selected using a subset of omnipath, scPRINT (Han et al.'s heads): same scPRINT version but with attention heads selected using a subset of the Han et al.'s ground truth dataset, compared to GENIE3, DeepSEM, Geneformer v2, and scGPT on the MCalla et al. ground truth using the AUPRC and EPR on two human and two mouse ESC datasets. (c) Same as (b) but on the genome-wide perturb-seq dataset with scPRINT (Han et. al.'s heads) replaced with scPRINT (gwps' heads): same scPRINT version but with attention heads selected using a subset of the genome-wide perturb-seq ground truth. EPR and AUPRC are provided here in one barplot, left to right.

Based on both AUPRC and EPR, scPRINT outperforms all other methods on this benchmark (Figure 1.3B). This means, for example, that when training GENIE3 to only predict a gene's expression based on TF expressions, it is not selecting the right TFs amongst the set of a few dozen assessed in MCalla et al.

scGPT, Geneformer v2—and, in a few cases, scPRINT—can have values worse than random guessing. Thus, their predictions are often specific to some TFs but not necessarily the right ones (Figure 1.3B).

It also appeared that selecting heads based on Omnipath, although helping slightly in

one instance, is not a net benefit for this dataset (see Supplementary Table 5.1.9). This makes sense since MCalla et al. itself does not overlap much with Omnipath (see Supplementary Table 5.1.7). However, selecting heads based on the ground truth itself, only using 50% of the connections available, shows substantial improvement. These same heads also show reliable behavior when using them on the second dataset of the same species.

This shows that scPRINT can better decipher direct from indirect TF-gene connections than scGPT, DeepSEM, Geneformer v2, and GENIE3, although more tests would likely be needed.

However, the results also highlight that the high imbalance (i.e., TFs being not connected or highly connected) combined with the dataset size (i.e., only a few dozen TFs assessed) and the low number of cells make the results in MCalla et al. very variable. Some of this might be true biology or explained by ChIP-seq, which can be very noisy depending on the quality of its antibodies[139].

To answer this issue, we selected another dataset: genome-wide perturb-seq (gwps)[140]. Here, we measured the effect on transcription of knocking out all expressed genes in the K562 cell line. We transformed it into a network using a cutoff of 0.05 on the significance level of each gene's differential expression before and after the KO of each other gene. Although this does not tell us which connections are direct or indirect, we now have a much broader set of connections over thousands of genes and better statistics to assess our gene network inference methods.

GENIE3 performs best, directly followed by scPRINT. Interestingly, Geneformer v2 shows poor performance (Figure 1.3C). Perturbation experiments are known to correlate somewhat to expression correlation, and this might explain GENIE3's strong performance. However, when using our head selection mechanism, scPRINT (gwps' heads) outperforms GENIE3. Again in this dataset, selecting heads based on Omnipath does not help; the small overlap between the gwps network and the Omnipath ground truth network seems likely to be the culprit (see Supplementary Table 5.1.7). These overlaps show that the three ground truth networks are very different and that a different set of heads predicts each type of ground truth. We also assess the networks on the TF-gene only subset of the gwps ground truth. Here, we see a large drop in performances for most methods, except GENIE3 (see Supplementary Figure 5.2.4).

Finally, we have seen that on both MCalla and gwps, scPRINT also predicts networks that agree with the Omnipath ground truth and are again enriched for cell type markers and TFs (see Supplementary Tables 5.1.8 and 5.1.9).

Since GNs can be seen as approximations of a cell model, we expect that when a tool has good internal cell models, it should generate meaningful results on tasks such as denoising, cell type prediction, embedding and batch effect correction, perturbation prediction, trajectory inference, and more. We will now focus on three tasks orthogonal to GN inference to compare the ability of scPRINT to the SOTA.

1.3.4 scPRINT is competitive on tasks orthogonal to GN inference

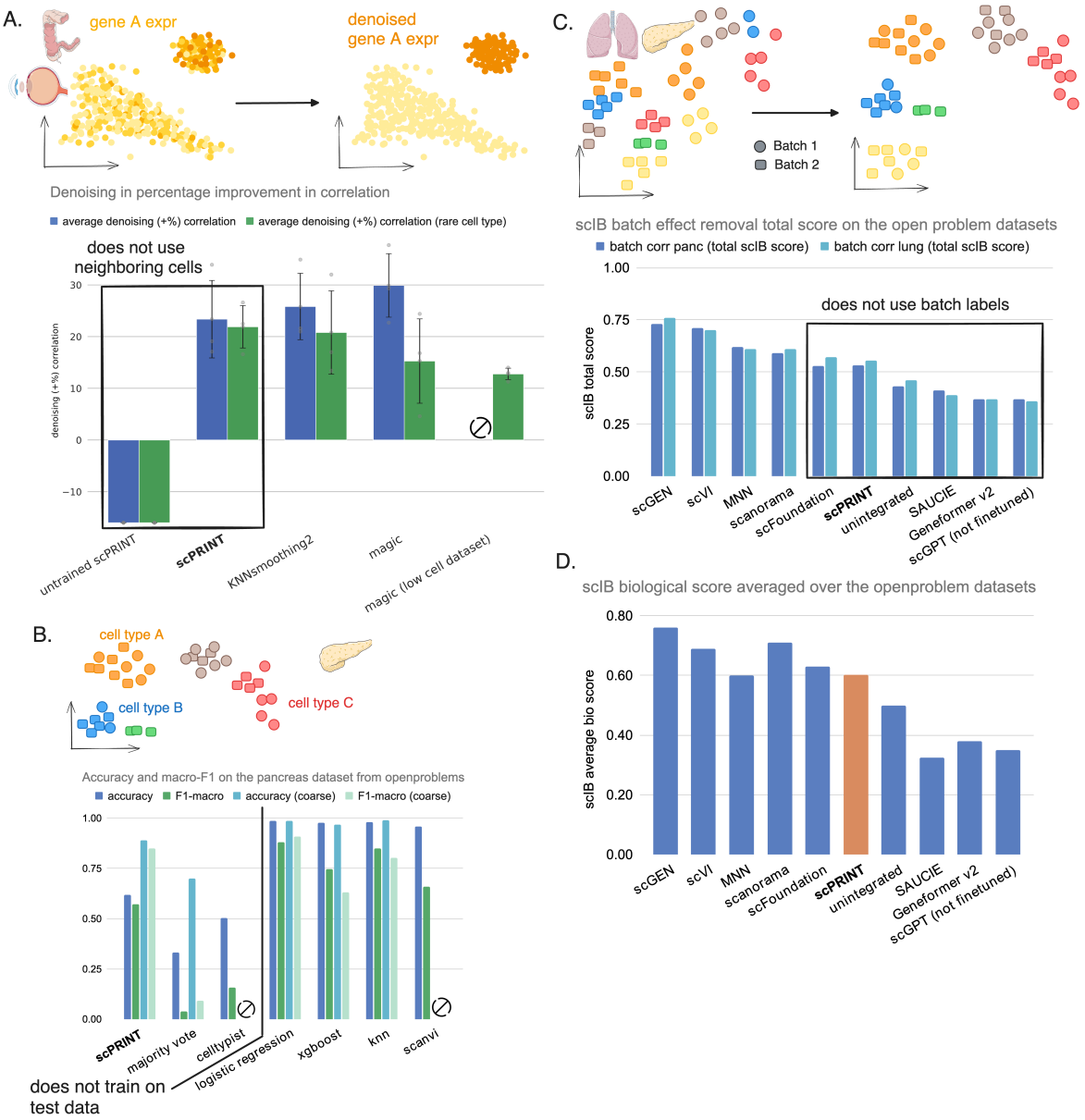


Figure 1.4: Benchmark of scPRINT on orthogonal tasks to GN inference. (a) Performance for a denoising task compared to SOTA methods MAGIC and knnsmooth2 on 3 datasets (ciliary body, colon, and retina tissues) from CxG. Here, we generate a noisy profile by downsampling 70% of the cell transcripts and computing the Spearman correlation increase of the correlation between the denoised and the true profile compared to the one between the noisy and the true profile. (b) Performance on cell-type label prediction compared to SOTA methods as well as CellTypist. Showing accuracy, F1 and macro-F1 scores for the open-problems human pancreas dataset. (c) The performance of scPRINT as well as scGPT and Geneformer v2 on batch effect correction on the human pancreas and lung datasets from the openproblems challenge showing the scIB aggregated score. They are compared to SOTA methods which results were extracted from the openproblems benchmark. Unintegrated means only PCA was applied. (d) The scIB avgBIO score on both datasets.

To test the quality of the cell model learned by scPRINT, we now consider denoising, cell type prediction, and batch effect correction as a representative set of classic scRNA-seq and cellular biology benchmarks.

Similarly to our pretraining task, we simulate lower transcript count profiles and then ask scPRINT and two other SOTA methods, MAGIC[41] and KNNsmoothing2[141], to recreate the true expression profile. We use Spearman correlation to the original gene expression profile as our metric. In Figure 1.4A, we show the increase in correlation after denoising the downsampled profile on 3 test set datasets, composed of ciliary body, colon, and retina tissues[131, 142, 143], randomly selected from CxG (see 1.5. Methods).

ScPRINT is competitive with both SOTA methods, while contrary to MAGIC and KNNSmoothing2, it operates independently over each cell in the test set (see 1.5. Methods). We have also seen a 10% variability in denoising ability across the different datasets used (see Supplementary Table 5.1.10). This was similar across all tools and possibly related to the number of genes expressed in each dataset.

However, these test cases mostly contain very similar cell states, whereas denoising is helpful in cases with rare cell types or transitory cell states that have low cell counts by default. We show that since scPRINT does not aggregate profiles over neighboring cells, it outperforms MAGIC and KNNsmoothing2 in rare cell states subsets of the datasets (respectively: pericytes microfold cells of epithelium of small intestine and microglial cells) with around 10 to 200 cells (Figure 1.4A, Supplementary Figure 5.2.5). Computing MAGIC and KNNsmoothing2 over only this rare cell population gives even lower performances for MAGIC and creates an error for KNNsmoothing2 (see Supplementary Table 5.1.10). These results suggest that a good cell model reliably using learned gene-gene interactions can help denoise an expression profile.

For cell type classification, we expect scPRINT to be able to find sets of genes that can predict a cell type across multiple batches and under the high dropout rate of single-cell scRNA-seq. To evaluate cell type classification, we use the multi-batch benchmark pancreas dataset of openproblems, its metrics, preprocessing, and hyperparameter choices (see 1.5. Methods)[144, 145].

scPRINT is a zero-shot predictor of cell labels. Indeed, it does not need to train on the dataset itself to make its predictions, unlike other methods that often need to use more than 70% of the test dataset for training. scPRINT also makes predictions over more than 200 cell type labels, while other methods often only predict a few cell types. Conversely, the other classifier methods, like Logistic Regression or XGBoost, and previous foundation models are trained or fine-tuned on the test dataset, thus giving a strong advantage over scPRINT. We, therefore, also compare scPRINT to the marker-based classifier CellTypist[146] and its pancreas marker database (see 1.5. Methods). A method that also does not use the labels of the test dataset.

scPRINT reaches 62% classification accuracy, largely outperforming CellTypist (Figure 1.4B, Supplementary Figure 5.2.6). Interestingly, with the macro F1 score, which considers each cell type group equally regardless of its size, scPRINT achieves similar results to the SOTA[145] methods: Logistic Regression and XGBoost. This is probably because scPRINT is not influenced by the number of cells in each category.

In addition, we have noticed that scPRINT is challenged by some specific pancreatic cell types in this dataset. Indeed, scPRINT often switches the assignment of A, B, D, and E cells. Thus, when using the coarser "endocrine pancreatic cell" label to define these cell types, we see a big improvement in the accuracy and macro-F1 score of scPRINT, even outperforming SOTA methods.

Here, we have shown the accuracy of scPRINT independently of cell neighborhood. However, like gene marker-based methods, scPRINT can annotate cell types in novel datasets. In this context, its predictions could be smoothed and improved using majority voting over predefined cell clusters.

Finally, scPRINT predictions are given as probability vector overall cell type labels. They can be used to display the top K labels and learn about the model's uncertainty.

Thanks to its disentangled embeddings, scPRINT can generate cell representations that partially remove batch effects from cell profiles. On the human pancreas and lung datasets of open problems[147], we see that, based on the scIB metrics, scPRINT shows convincing batch effects removal ability, while not on par with the SOTA methods scGEN and scVI (Figure 1.4C, Supplementary Figure 5.2.7). Concerning foundation models, scPRINT and scFoundation show strong zero-shot performances compared to Geneformer v2 and scGPT. Except for Geneformer v2, scGPT, and scFoundation, we did not rerun previous algorithms for this benchmark and show their performances from the openproblems portal (open-problems-v2.3.6, march 2024). However, we also ran the Geneformer v2 and scGPT foundation models on the openproblems benchmark and showed that without fine tuning on this specific dataset, they are not able to meaningfully correct for batch effect (see 1.5. Methods).

Moreover, scPRINT is one of the few methods that do not train on the test dataset and do not use already annotated batch labels. When only looking at methods that do not use batch labels as prior information, e.g., SAUCIE[148], LIGER[149], scPRINT is the top performer. We have also noticed that the scPRINT cell embeddings preserve biological information competitively to SOTA methods (Figure 1.4D, Supplementary Figure 5.2.8). This also exemplifies that a reliable cell model can perform well at disentangling the different facets of a cell expression profile and its underlying batch effect.

Overall, we have seen that scPRINT can achieve zero-shot performances on par with many famous single-cell scRNA-seq tools on multiple important tasks of single-cell biology, showing that our architecture and foundational pretraining tasks are a powerful new foundation for large cell models.

1.3.5 scPRINT highlights the role of ion exchange and fibrosis in the ECM of Benign Prostatic Hyperplasia

To showcase the ability of scPRINT, we focus on premalignant neoplasms from an atlas of two studies of human prostate tissues[32]. The data contains both normals and pre-cancerous lesions, also called BPH, across sequencers and age groups. Starting from post-alignment raw counts, scPRINT generates a consistent and batch-corrected embedding of the datasets

(Figure 1.5A, Supplementary Figure 5.2.9). scPRINT also annotates the cell type, sequencer, sex, ethnicity, and disease type of each cell with an accuracy of 0.71, 0.99, 0.99, 0.95, and 0.85, respectively.

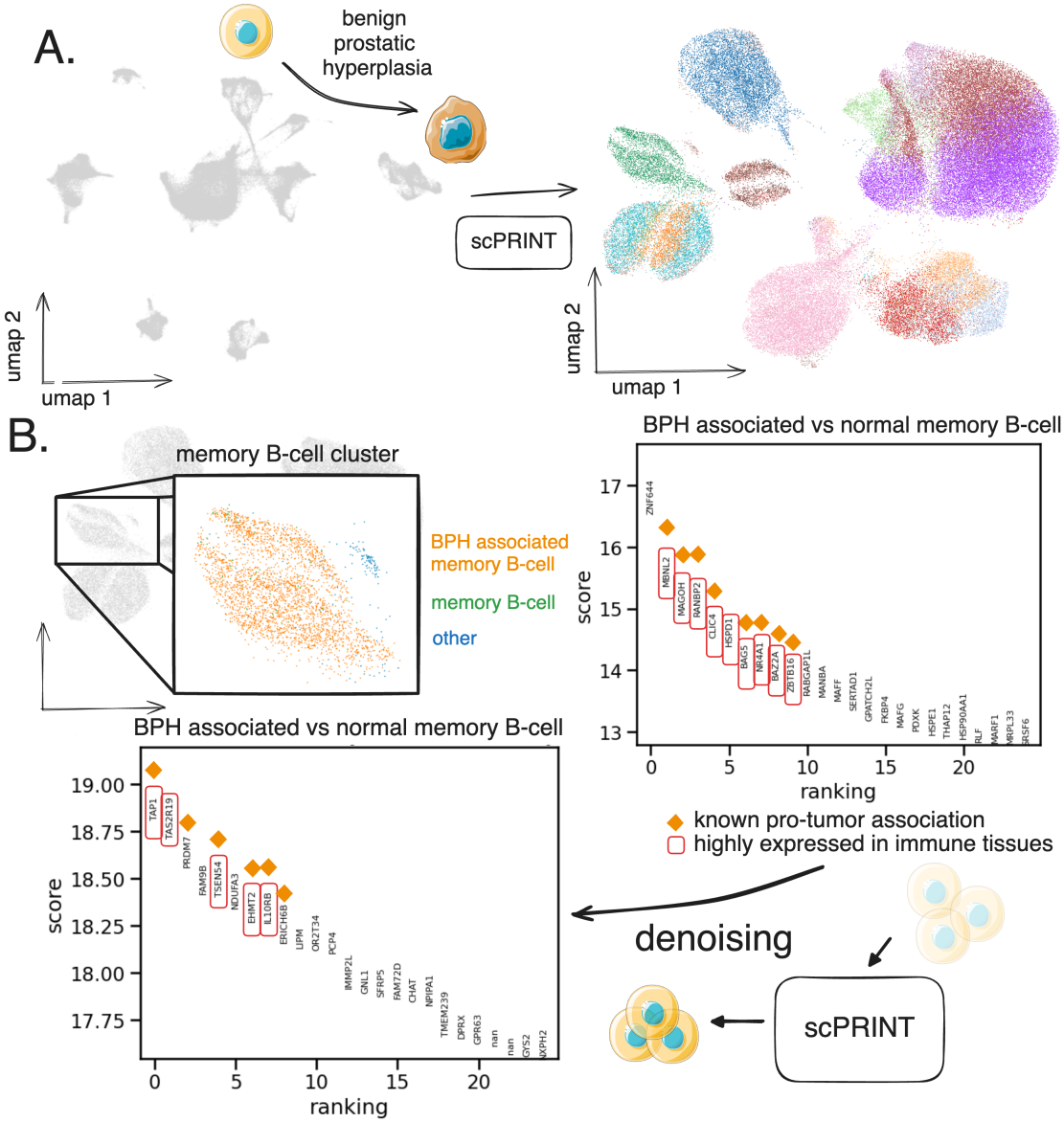


Figure 1.5: scPRINT-based bioinformatics analysis of early prostate cancer. (a) Single-cell scRNA-seq atlas of BPH and normal prostate tissues of 83,000 cells given to scPRINT. scPRINT generates a set of embeddings and label predictions for each cell. To clean our predictions, we drop cell types with less than 400 cells and diseases with less than 1000 cells, replacing them with the "other" label (see Supplementary Figure 5.2.9). (b) Zooming in on one cluster, we see annotations of a switched memory B-cell cluster, some labeled "benign hyperplasia" and others "normal". Differential expression analysis on the two groups of B-cells showing enrichment of B-cell & cancer markers when assessing its top 10 genes. We performed upsampling of the transcript count before performing a new differential expression analysis where we now see new genes amongst the top 10 differentially expressed ones some of them also associated with cancer and immune tissues.

We then focus on a switched memory B-cell cluster composed of a group of cells labeled as benign prostatic hyperplasia and another as normal (Figure 1.5A). B-cells are known to be dominant in prostate cancer and are often switched memory B-cells[150]. First, we show that they differentially express many known B-cell markers (see Supplementary Figure 5.2.10). In addition, when comparing the BPH to the normals B-cells, we recover that the top 10 BPH B-cells differentially expressed genes contain many known cancer markers, B cell markers, and a specific B-cell associated prostate cancer markers: BAG5[151] (highlighted in Figure 1.5B, Supplementary Table 5.1.11). Moreover, many other genes have evidence in other cancers, like CLIC4, known to be involved in the maintenance of the TME in breast cancer[152].

However, the number of healthy cells, especially normal memory B-cells, in this dataset is small: only 26. By performing denoising, we can recover genes that might have been missed during differential expression analysis of such a low cell count. Increasing the counts of all the genes by a factor of ten and re-doing differential expression analysis highlights some new genes whose differential expression scores are even higher than those previously cited.

Interestingly, amongst them, TSEN54, EHMT2, and IL10RB are known to impact the function of B-cells in malignancies (see Supplementary Table 5.1.11). Other genes have evidence in immunity and cancer, like TAP1, which is known to be highly expressed in immune organs and is an immunomodulation gene known to play many roles in various cancers[153], while some genes have, of yet unknown significance, like LIP, whose paralog LIPA is a known cancer target[154] (Figure 1.5B).

This demonstrates how scPRINT can embed, align, and annotate diverse datasets in a meaningful way so that one can then analyze specific and rare cell clusters to recover both known and new biology.

Finally, for the second part of the analysis, we move to another cell type of interest: fibroblasts. Fibroblasts are known to be involved in cancer[155], also called cancer-associated fibroblasts (CAFs), of which many subtypes exist, with different roles in tumor progression and invasion[156]. In our dataset, we can see a large cluster of cells labeled as “fibroblast of connective tissue of glandular part of prostate”, of which 500 are coming from normal tissues, and 600 are coming from hyperplasia and are possible precursors of CAFs (Figure 1.6A). Interestingly, 40% of the cells annotated as BPH-associated fibroblasts are coming from healthy tissue, according to the authors of the dataset. However, it is known that more than 50% of adult males over the age of 50 will have BPH[157]. Thus, one possibility is that some of the fibroblasts of these healthy tissues already present patterns of gene activation similar to those of pre-cancerous ones.

We generate a gene network of the BPH and normal fibroblasts using the 4000 most variable genes and taking the average over all heads in the network (Figure 1.6A). Looking at the top 15 hubs, using degree centrality, we can see S100A6 as the top element in normal fibroblasts. This gene is known to be a fibroblast and epithelial cell marker that regulates, among other things, cell cycle and differentiation[158, 159]. We also see MIF, IGFBP7, and other genes involved in immune signaling and growth[160, 161, 162].

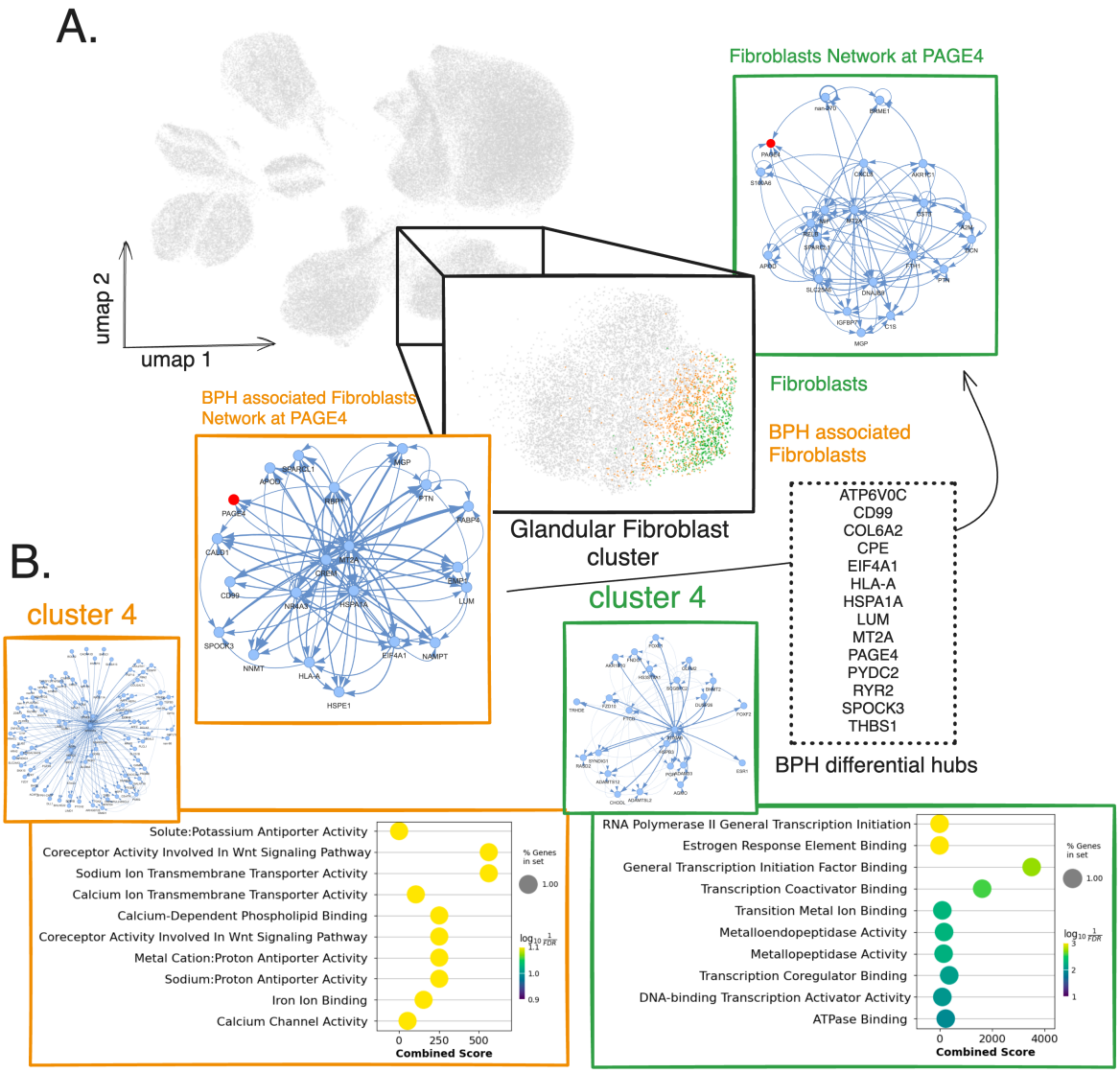


Figure 1.6: scPRINT-based bioinformatics analysis of early prostate cancer predicts disease cell-type specific gene networks. Continuing on the single-cell scRNA-seq atlas of BPH and normal prostate tissues of 83,000 cells given to scPRINT (a) Zooming in on another cluster from scPRINT's cell embeddings and annotations, we see a group labeled as "fibroblast of connective tissue of glandular part of prostate", some labeled as "benign prostatic hyperplasia", and others "normal". We generate gene networks from each and highlight a sub-network of the PAGE4 differential hub gene in BPH, showing different connection strengths and patterns between normal and BPH-associated fibroblasts. (b) Left to right: gene-set enrichment analysis, using Enrichr, of the gene community 4 found by the Louvain algorithm in the BPH-associated fibroblast gene network, same but on the normal fibroblast gene network. It shows the top 10 most strongly enriched gene sets from GO_MF_2023 according to q-value (i.e. adjusted p-value).

However, some of these genes are not in common with the BPH fibroblasts ones. Over the set of 2881 common nodes between the two networks, the genes HSPA1A, MT2A, SPOCK3, ATP6V0C, DEFA1, EIF4A1, and CD99 are considered differential hubs (i.e., more central) in the BPH fibroblasts compared to normal ones (see Figure 1.6A, Supplementary Table 5.1.12).

Another definition of centrality, eigenvector centrality, recovers 55% of the genes already identified as hubs, plus some new ones. As an example, Prostate Associated Gene 4 (PAGE4), which is part of the GAGE family of genes, is expressed in a variety of tumors and reproductive tissues, especially BPH, where it is related to oxidative stress response and fixation (i.e., anti-invasion)[163, 32, 164]. Interestingly, although the networks share 75% of their genes, they only share 50% of their edges when considering the top 20 edges per gene. It shows that over the same set of genes, scPRINT discovers distinct gene networks across biological contexts. Taking as an example the differential hub PAGE4 (Figure 1.6A), we see that it is connected to many of the top 15 hub nodes in the BPH network, such as MT2A, HSPA1A, SPOCK3, and CD99. This shows a master node sub-network linking metal and ion exchange, oxidative stress response, and inflammation[165, 166, 167, 168]. Some genes are also part of the IL24 signaling inflammatory pathway (EIF4A1;COL6A2;HLA-C;HSPE1), and the secretory senescence phenotype (H2AZ1;UBE2S;UBE2C;IGFBP7)[169], hallmarks of fibrosis and malignancies[170, 171]. The PAGE4 network in normal fibroblasts, while having some elements in common, like metal transport, is much less connected (seen by the strength of the edges in Figure 1.6A). It also contains a different set of genes, which are less related to senescence, inflammation, and ion exchange (see Supplementary Figure 5.2.11).

Furthermore, we can use these networks, defined over only a few cells, to perform community detection. Taking community 4, containing 92 genes and defined with the Louvain algorithm on the BPH-associated fibroblasts GN, we see two hub nodes: SPOCK3 and HERC3 (Figure 1.6B). Interestingly, not much is known about those genes except that HERC3 has been linked to inflammation and the ECM via metallopeptidase and the NCOA1 gene[172]. SPOCK3, moreover, is known to be related to prostate malignancies and collagen in the ECM[166]. Gene set enrichment tells us that the genes in this subnetwork are primarily related to calcium, sodium, iron, and metal transport, validating the evidence around HERC3 and SPOCK3 (Figure 1.6B). In normal fibroblast, however, taking the community most associated with metal transport (community 4, see details in Supplementary Figure 5.2.12 and Methods) shows RNASEK, SELENOM, and an unknown ubiquitin ligase, paralog of ITCH. While RNASEK is related to RNA degradation, its expression has been linked to a lower risk of prostate cancer[173]. SELENOM is of unknown function, but some SEL proteins have been related to cell adhesion[174] .

Through its networks, scPRINT highlighted the role of ion exchange and fibrosis in the ECM in BPH. While some of the same genes would have been found from differential expression analysis, these results show us how gene networks can be used to describe the intersection of genes and their molecular functions. Putting genes into the context of their connections, one can validate known functions or relate them to new ones. From such contextualization, a picture starts to emerge, whereby through specific genes, glandular fibroblasts in senescence enter a wound-healing state. This fibrosis is caused by the export of more metal and ions to generate ECM and change its acidity levels. This might cause a loss in tissue flexibility and potentially create oxidative stress[175]. In our networks, these pathways seem connected to inflammation. Chronic inflammation and wound healing states are hallmarks of BPH and a predisposition to future malignancies[176, 177].

1.4 Discussion

We can simplify the complex macromolecular interactions governing a cell through what is often referred to as a gene network. However, creating such a network in a meaningful way remains a challenging task.

We have created and benchmarked scPRINT, a novel single-cell scRNA-seq foundational model trained on more than 50 million single-cell profiles across tissues, diseases, and species contexts. scPRINT uses three foundational pretraining tasks, as well as new encoding and decoding mechanisms specifically designed for gene expression data. Although it has not been directly trained for it, scPRINT generates gene networks. These networks can be used to better understand the model predictions and help make more informed decisions about the significance and role of a potential target. Finally, we present a mechanism to best select heads containing the known biology of these networks. This approach also helps users fine-tune the type of network they are interested in. Given the discrepancy amongst ground truth networks, we advise users to consider using all-head averaging and to only revert to head selection when some high-confidence interactions are available. Indeed, general collections like Omnipath did not improve performance in most of our tests.

We show that we outperform other foundation models on most of our benchmarks while using a similar model size. We believe that our inductive biases and training procedures helped scPRINT achieve such a performance. Moreover, while GENIE3 is still a competitive tool, we outperformed it on many of our benchmarks, showing that pushing training to millions of cells and large parameter sizes will be an essential direction for further work on gene network inference.

In addition, contrary to any other method assessed, our large cell model can also achieve zero-shot performances on par with many famous single-cell scRNA-seq tools on multiple important tasks. While some specialized tools might be better suited to some use cases, scPRINT's versatility and speed make it a worthwhile alternative in many instances. Indeed, users can directly use scPRINT in their bioinformatics workflows with commodity hardware (1 CPU, 1 GPU with 10GB of memory and 16GB of memory).

Finally, we put scPRINT to the test on a challenging atlas of normal and senescent prostate tissues showing BPH. We identify rare cell populations with early markers of TME in B-cells. In fibroblasts, we study gene networks and recover known hubs such as PAGE4, thereby linking the senescence of fibroblasts to changes in the ECM and downstream inflammation. We find key interconnected pathways of the oxidative stress response and extracellular matrix building via metal and ion exchange in the gene network of BPH-associated fibroblasts. We also show that healthy and disease-related cells exhibit different network patterns, demonstrating that scPRINT can help identify novel pathways and targets while considering them in their specific cellular and molecular contexts.

An assumption in natural language processing is that fewer inductive biases make for better models. Our work shows that adding good inductive biases and rethinking architectures will likely be important directions for AI models in biology.

A challenging aspect of GN inference is that no perfect ground truths exist, and many

GN methods are, unfortunately, benchmarked on ODE-generated mock-up expression data. In contrast, ChIP-seq, perturb-seq, and literature-based ground truths remain scarce and ambiguous. With BenGRN and GRnnData, our suite of tools for benchmarking Gene Networks inferred from single-cell scRNA-seq, we present an extensive set of real-world ground truths representative of the diversity of networks we can assess. However, improvement in performance and benchmarking will need to come from innovative experimental approaches that can produce causal, genome-wide, and cell-type-specific networks containing the many different types of connections and regulations that exist, from PPI, RNA-DNA, RNA-protein, to inhibition, activation, cooperation, and more.

We acknowledge that work remains to be done, from the transformer's ability to generate graphs to their explainability and the breadth of tasks they can undertake. Questions still remain regarding the pretraining tasks and how to integrate additional data modalities into foundational models.

Transcription is much more complex than what gene networks currently represent. In the future, we expect such large cell models to work in tandem with new sequencing techniques measuring information such as time, space, protein amounts, DNA configuration, and non-coding RNA species to solve the gap in our understanding and our ability to model cell biology.

1.5 Methods

we propose scPRINT, a foundation model designed for gene network inference. ScPRINT brings novel inductive biases and pretraining strategies better suited to GN inference while answering issues in current models. scPrint outputs cell type-specific genome-wide gene networks but also generates predictions on many related tasks, such as cell annotations, batch effect correction, and denoising, without fine-tuning.

1.5.1 Architecture

The model architecture is composed of:

- An encoder that takes the raw data and embeds it in a high-dimensional space used by the transformer.
- A bidirectional multi-head transformer
- A decoder to transform the expression embeddings into expression values
- A decoder that transforms the cell embeddings into cell-specific label prediction over a range of classes.

Expression encoder

In scPRINT, each gene in a cell is converted to an embedding: It corresponds to the sum of 3 different elements:

1. An embedding representing the gene itself (see Supplementary Table for model embedding size). ESM2 embedding of each gene's most common protein product was used to represent that gene. While imperfect in some ways, this inductive bias allows the model to learn representations that potentially apply to even unseen genes from unseen species or integrate specific genetic mutations into its representation. First implemented in UCE, this provides the model information related to the gene product's structure, ontology, and similarity to other genes. This also speeds up the training greatly, particularly for small models. We show that this is a great gene representation, but that model performance can be increased by refining gene embeddings further during training. However, we elect not to do so to maintain the model's versatility in working on unseen genes.

We encode the genes' embeddings using ESM2. The mapping process happens the following way:

- A gene name is mapped to its canonical protein name using Ensembl.
- We recover the protein sequence of the protein using Ensembl
- We use the protein sequence to generate an embedding using ESM2 by averaging all the amino-acid output embeddings, as done in the ESM2 paper.

With the embedding function provided in our code, one can easily do this with any species in Ensembl.

scPRINT can effectively be retrained with any set of gene embeddings, which can be frozen during training or used only for initialization (tried, for example, in our ablation studies, Table S3).

2. An embedding of the gene location in the genome. This has also been proposed in UCE and helps the model understand that genes with similar locations tend to be regulated by similar regulatory regions, a relationship well-known in cellular biology.

We encode the genes' locations using positional encoding. Every gene less than 10,000 bp from the next is said to be in the same location; otherwise, we increment location by 1. We do this for all genes in the Ensembl database per species.

We then embed these locations by applying the Positional Encoding (PE) algorithm of Vaswani et al. .

3. An embedding of the gene expression in the cell. For this, we embed the gene's expression using an MLP. While GeneFormer devised a ranking strategy based on a gene expression compared to a baseline expression, scGPT instead used binning of log normalized counts. On our end, we haven't found that this approach was the simplest, nor was it performing better than only using the log-transformed counts. We thus directly take the log-transformed counts

$$\mathrm{e_{i,j}} = MLP(log_2(x_{i,j} + 1)),\ x_{i,j} \in \mathbb{R},\ \mathrm{e_{i,j}} \in \mathbb{R}^d \tag{1.1}$$

where $\exp r_{i,j}$ is the embedding of the expression, $x_{i,j}$ is the expression value of the gene j in the cell i, and the MLP is a two-layer neural network, where each layer is composed of

$$Dropout(ReLU(LayerNorm(Linear(\mathrm{e_{i,j}})))) \tag{1.2}$$

where the Dropout rate is fixed at 0.1, and the dimensions are specified as $1 \longrightarrow d$ for the first layer of the MLP and $d \longrightarrow d$ for the second layer, with d representing the model dimension.

Of Note: Geneformer used positional encoding to encode gene expression, a function often used to encode the position of words in a text. Similarly to gene name token, scGPT learned an embedding for different ranges of expression values, binning them to remove sampling noise.

Both approaches apply a specific prior for the metric that defines expression. Geneformer defines expression amount as ranking based on how each gene is expressed in the cell compared to its average across all cells. Unregarding the batch effect issues, this is an assumption that expression values are not meaningful and only the ranking of the relative abundance is meaningful information. Meanwhile, scGPT has the bias that an expression of 1, 2, or 3 are the same and that an expression 1, and 5 are different by some amount learned by the model.

By using an MLP with two layers, we effectively let the model learn the metric of transcription expression. Moreover, again, we decrease the number of parameters used compared to scGPT while being able to make predictions on count values unseen during training, such as those of bulk or pseudo-bulk RNAseq.

Finally, when encoding a cell expression profile, only a subset of 2200 genes is used during pretraining. If less than 2200 genes are expressed, we randomly choose 2200 expressed genes and pad them with randomly sampled unexpressed genes (meaning with an expression value of 0). This approach allows the model to see different patches of the same cell profile during training. We chose 2200 genes as 2/3rds of the cells in cellxgene had less than this number of genes expressed, striking a balance between computation and gene usage.

We decided to add unexpressed genes because, combined with our denoising methodology, this lets the model figure out that some genes are true 0s during training. In contrast, others are only caused by dropout and a function of the transcript counts. This causes scPRINT to model dropout as a function of read depth (i.e., total transcript count).

Moreover, this completes the minibatch by token matrix without padding and fully utilizes the GPU during the attention computation.

Of note, some models have been able to reach context lengths of 20,000 genes using the performer architecture. Performer is an often-cited method and part of the literature on attention approximation. However, most state-of-the-art transformer models do not use attention approximation as they are known to lead to worse performance.

Moreover, in cellxgene, more than 80% of the cells have less than 2200 genes being measured. This means that most of the memory and compute power is likely lost on tokens that are almost always zeros due to dropout.

The full set of embeddings of cell i sent to the transformer is the matrix X_i where

$$X_i = [\mathrm{g}_0 + \mathrm{e}_{\mathrm{i},0} + \mathrm{l}_0, \mathrm{g}_1 + \mathrm{e}_{\mathrm{i},1} + \mathrm{l}_1, ..., \mathrm{e}_{\mathrm{i,t}}, \mathrm{p}_{\mathrm{default}}, \mathrm{p}_{\mathrm{celltype}}, \mathrm{p}_{\mathrm{disease}}, ...] \tag{1.3}$$

where g_j is the gene j encoding, $\mathrm{e}_{\mathrm{i,j}}$ is the encoding of the expression of gene j in cell i, l_j is the gene j location encoding, and p_A is a learnt embedding for the class A.

The total count information is stored separately and encoded similarly to the expression,

$$\mathrm{e}_{\mathrm{t,i}} = MLP(log_2(1 + t_i)), \text{ where } t_i = \sum_j x_{i,j} \tag{1.4}$$

with $x_{i,j}$ the expression value of gene j in cell i, and the MLP is a two-layer neural network similar to the previous one.

The full cell total count (t) lets scPRINT model its denoising based on this required total count parameter.

The placeholder tokens (total count, default cell embedding, cell type, disease, sex, ethnicity, assay, organism) are learned embeddings that stay the same across all inputs. They only act as placeholders for the model to fill in during the forward process. At the transformer's output, they will have been modified to contain the embeddings requested. At least two are used, one containing the default cell embedding and another the profile's total depth. More tokens can be used, one for each predicted cell label.

Model

The model is a bidirectional autoencoder similar to BERT with n layers, h attention heads, and a dimension of d. It uses the flashattention2 methodology implemented in Triton to compute its attention matrix. It uses the pre-normalization technique, with a sped-up layer norm implemented in Triton's tutorial. It uses a stochastic depth with increasing dropout probability.

It has a 2-layer MLP with a 4x width increase in its hidden layer and a GELU activation function.

Expression decoder

scPRINT uses a novel expression decoder for foundation models, which outputs the parameters of a zero-inflated negative binomial (*ZiNB*) function for each gene i in cell j. The *ZiNB* distribution is defined as

$$X \sim ZiNB(\mu, \theta, \pi) \tag{1.5}$$

where the parameters $\mu,\ \theta,\ \pi$ are obtained from a multi-layer perceptron (MLP) applied to the expression embeddings outputted by the transformer model at its last layer (e), which are the:

$$\mu, \theta, \pi = MLP(\mathrm{e}) \tag{1.6}$$

The MLP is a two-layer neural network with dimensions [d, d, 3]

Based on the work of Jiang et al., zero inflation is the best distribution when considering a broad range of transcriptomic measurements, where some have enough dropouts, and a zero inflation term is needed to model it. In our case, and similarly to scVI, we define our *ZiNB* as

$$ZiNB(x|\mu, \theta, \pi) = \pi\delta_0(x) + (1 - \pi)NB(x|\mu, \theta) \tag{1.7}$$

where $\delta_0(x)$ is a point mass at zero, and $NB(x \mid \mu, \theta)$ is the negative binomial distribution with mean μ and dispersion θ.

With these parameters, the negative binomial distribution is represented in the following way

$$NB(x|\mu, \theta) = \frac{\Gamma(x + \theta)}{x!\Gamma(\theta)} \left(\frac{\mu}{\mu + \theta}\right)^x \left(\frac{\theta}{\mu + \theta}\right)^\theta \tag{1.8}$$

where μ is the mean and θ the overdispersion parameter, representing the inverse of the dispersion. From Hibe et al., we know that this is a parameter change from the most used probability mass function (PMF) given by

$$P(X = x) = \binom{x + r - 1}{x}(1 - p)^r p^x \tag{1.9}$$

where r is the number of successes, p is the probability of success, and k is the number of failures.

One can interpret such a negative binomial distribution as a Poisson distribution with an additional overdispersion term that makes the variance not tied to the mean. In scPRINT, we use the zero-inflated Poisson for count downsampling as we can't easily infer the gene overdispersion parameter from each cell profile. By removing this zero-inflated Poisson from the gene expression profile, we keep the potential overdispersion in the profile (see the Negative Binomial to Poisson relationship section in Methods).

Compared to scVI, where the overdispersion parameter θ is learned for each gene, we make scPRINT output it together with $\mu,\ \pi$ (see Supplementary Figure 5.2.13)

Effectively, the model learns that the dispersion might change depending on the gene, the sequencer, the cell type, and the sequencing depth.

Class decoder

scPRINT also outputs a variety of class embeddings, such as default cell embedding, cell type embedding, disease embedding, etc., by filling the different placeholder tokens given as input (see the Expression encoder section in the Methods).

Effectively, for each class, we have the model learn to produce a new disentangled embedding (e.g., cell type, disease, tissue, age). This means the model uses an MLP to transform each token where A is a class. For each, we jointly train a classifier:

$$\widehat{\mathrm{c_A}} = \sigma(MLP_A(\widehat{\mathrm{e}_A})) \tag{1.10}$$

where:

- $\widehat{\mathrm{c}_A}$ represents the logits for a class A of a dimension d_A whose size corresponds to the number of labels.
- σ denotes the Sigmoid activation function.
- MLP_A stands for the Multi-Layer Perceptron trained to predict the logits of the class A.
- $\widehat{\mathrm{e}}_A$ is the output embedding for the class A of dimension d.

However, some classes, like cell type, have up to 800 labels. Fortunately, cellxgene classes follow an ontology, a robust structure that defines relationships among the labels. We reduce the size of the output labels by training the model only on the leaf labels in the ontology hierarchy (i.e., the most precise available). For cell types, this represents around 400 different labels (see Supplementary Table 5.1.13).

Thus, when a label is not very specific for a cell type (e.g., neuron), the model will predict the best leaf label (e.g., dopaminergic neuron). This way, we can generate meaningful training signals from even very coarse labels (see The classification task section in methods for more information and definition of the loss). We only apply this hierarchical classifier to the cell type, disease, and assay labels.

In the following section, we show how we train such classifiers. During the classifiers' training, we sum up their loss without applying any scaling between the different classes.

1.5.2 Ablation study

We perform an ablation study of multiple of our additions in scPRINT for its medium size version. Removing positional encoding, replacing log-normalization with a total-normalization, replacing denoising with masking, using the cell-gene product method

of scGPT vs our own encoder-decoder approach to learn a cell embedding, using 2 vs 4 heads per attention blocks, not using weighted random sampling, not freezing the gene ID embeddings, and using mean-squared-error instead of the ZINB loss. For each, we re-train scPRINT entirely on the same dataset and validate its test performance with our automated benchmark platform. We provide the results in Table S3.

1.5.3 Pretraining

The three tasks of the multi-task pretraining are the denoising task, the classification task, and the bottleneck learning task. While the denoising loss enhances the model's ability to find meaningful gene-gene connections, the other two try to make the model and its underlying networks more robust and cell-type-specific. All three losses are summed without rescaling.

Optimization method

The optimization is done with fused ADAMW, with a weight decay of 0.01. We noticed a total inability to learn when using base ADAM, which has a similar weight decay. This can be explained by a known inequivalence issue in ADAM.

We use the stochastic weight averaging method during training with a learning rate of 0.03.

During pre-training, the hyperparameters are set to dropout of 0.1, a learning rate (LR) of 1e-4, the precision is set to 16-mixed with residuals in fp32. We clip gradients to 100 and train in many sub-epochs of 7000 training batches and 2000 validation batches with a warmup duration of 500 steps.

Across epochs, we use a linear LR decrease of 0.6 with a patience of 1 and stop training after three consecutive increases in validation loss (patience: 3). In the final layer of the class decoders, we initialize values to a normal distribution around 1 for weights, 0 for biases, and -0.12 for biases.

Our batch size is 64, and we use a pre-norm strategy for the transformer with a linearly increasing stochastic depth dropout rate of 0.02 per layer. We use a noise parameter of 60%. We split the cells in the datasets into 98% train and 2% validation and reserve at minimum 2% of separated datasets for testing.

Finally, we use weighted random sampling on our training data based on the different class values we have to predict. We use a factor of 50, meaning the rarest elements will, on average, be sampled only 50 times less than the most common ones. The sampling factor used for each group is then $\frac{50}{count+50}$, instead of $\frac{1}{count}$ where count is the number of cells in each group.

The classification task

We perform label prediction during pretraining for different classes, currently: cell type, disease, sequencer, ethnicity, sex, and organism. Due to issues in the ontologies, we have omitted tissue type and age classes.

Due to the hierarchical structure of the prediction, we also created a hierarchical loss. Here, we compute the loss regularly when the label is a leaf label. Otherwise, we replace all associated leaf labels to the given label by the log-sum-exp, such that for a cell label, the loss is:

$$Loss_{classification} = CE(\sigma(\overline{\mathrm{c}}, \mathrm{c})) \tag{1.11}$$

with:

$$\overline{\mathrm{c}} = \begin{cases} \widehat{\mathrm{c}} & \text{if } \{i | c_i = 1\} \subseteq T \\ LSE\left(\widehat{\mathrm{c}}_d\right) || \widehat{\mathrm{c}}_{\sim d} & \text{else} \end{cases} \tag{1.12}$$

where:

- $\widehat{\mathrm{c}}$ is the predicted vector with dimension equal to the number of leaf labels
- T being the set of label indices marking the labels that are leaf labels.
- $\widehat{\mathrm{c}}_d = \{\widehat{c_i}, \forall\, i \in T\}$ all the values in vector $\widehat{\mathrm{c}}$ whose indices are in T. Same for c.
- $\widehat{\mathrm{c}}_{\sim d} = \{\widehat{c_i}, \forall\, i \notin T\}$ all the values in vector $\widehat{\mathrm{c}}$ whose indices are not in T. Same for c.
- LSE is the log-sum-exp operation

The CE (cross-entropy) is defined as:

$$CE(\mathrm{p}, \mathrm{q}) = -\sum_u q_u \log(p_u) \tag{1.13}$$

And the LSE (log-sum-exp) is defined as

$$LSE(X) = \log\left(\sum_{p \in X} e^p\right) \tag{1.14}$$

This loss allows the classifier to learn even in cases where the labels can be of varying coarseness without the coarseness of some labels impacting the ability of the model to predict the true fine-grained labels (see Supplementary Figure 5.2.14)

The loss is hierarchical for the classes: cell type, disease, sequencer, ethnicity; the labels follow a hierarchy defined by (Cell Ontology, MONDO, EFO, HANCESTRO), respectively.

We do not compute the loss for cells where a class has an unknown label. We perform these classification tasks in one pass, using the embeddings generated directly from the downsampled expression profile.

The denoising task

Similarly to ADImpute, we expect a good gene network to help denoise an expression profile by leveraging a sparse and reliable set of known gene-gene interactions. In addition, we expect a good cell model to help embed and reconstruct an expression profile by leveraging the regularities of modules and communities within its network.

We view denoising similarly to upsampling, and inversely, we view adding noise as downsampling a cell profile.

Noise is similar to downsampling because of the distribution we are working with. Note that contrary to vision tasks (e.g. diffusion models), where additive Gaussian noise is added, in the context of expression data, where the distribution is often seen as a Poisson, NB, or ZINB, the data is already noisy, and the more counts are sampled, the less noise. No information is similar to not sampling data.

We downsample an expression profile using a zero-inflated Poisson model of the data. With this formulation, on average, half of the counts to be dropped are dropped by randomly removing a number of reads per gene, given by sampling from a Poisson whose lambda parameter is proportional to the number of counts in that gene. The remaining half of the counts to be dropped are dropped by randomly setting some genes to 0, i.e. a complete dropout of that gene. It is to be noted that with this definition of downsampling, the exact average amount of counts dropped for both parts depends slightly on the dropout *r*. During our pretraining, *r* is set to 0.6, meaning, on average, 60% of the transcript counts are dropped per cell.

Let $\mathrm{x_i}$ be the gene expression vector of cell i with dimensions n_{genes}; we create a downsampled *version* by doing

$$\widehat{\mathrm{x}}_\mathrm{i} = \max((\mathrm{x_i} - \mathrm{p_i}) \cdot \pi_\mathrm{i}, 0) \tag{1.15}$$

with:

- $\mathrm{p_i} \sim Poisson(\mathrm{x_i} \times r \times 0.55)$ a vector of size n_{genes} where the poisson is samples for each element $\mathrm{x_i}$ of x
- $\pi_\mathrm{i} = I(u \geq r \times 0.55)$ a vector of size n_{genes} , the binary mask vector indicating non-dropout genes.
- $\mathrm{u_i} \sim Uniform(0, 1)$, a vector of size n_{genes}. of random values drawn from a uniform distribution.
- $\cdot$ denotes the element-wise multiplication.

- r being the dropout amount. We scale it by a tuning hyperparameter of 0.55 instead of 0.5 for numerical reasons.

The goal of the model is then using $\widehat{\mathrm{x_i}}$ as an input to output the parameters μ_i , θ_i , π_i of a *ZINB* distribution of the true profile $\mathrm{x_i}$, all vectors of size n_{genes}. The contribution of cell i to the loss is then computed as the negative log-likelihood of the count data given the distribution parameters being generated by the model

$$Loss_{denoising} = Loss_{ZINB} = -\frac{1}{n_{gene}m}\sum_{i=0,j=0}^{n_{gene},m}\log(L(x_{i,j}|\mu_{i,j},\theta_{i,j},\pi_{i,j})) \tag{1.16}$$

where n_{gene} is the size of the expression profile $\mathrm{x_i}$, m is the size of the minibatch and

$$L\left(x|\mu,\theta,\pi\right) = \begin{cases} \frac{\pi}{\pi-\theta\cdot(\log(\theta)-\log(\theta+\mu))} & \text{if } x=0 \\ \frac{\left(\frac{\mu}{\theta+\mu}\right)^{x}\cdot\Gamma(x+\theta)\cdot\sigma(-\pi)}{\exp(\pi)\cdot\left(\frac{\mu}{\theta+\mu}\right)^{\theta}\cdot\Gamma(\theta)\cdot\Gamma(x+1)} & \text{if } x>0 \end{cases} \tag{1.17}$$

with σ the sigmoid function.

We show that models trained with such a framework perform better than regular MSE-trained models (see Supplementary Table 5.4), for which one only outputs one value instead of three, directly representing the data's log-transformed count. In this case, the loss is the mean squared error between the predicted and true count values.

scPRINT effectively lets the user choose between the three formulations: *ZINB* with a *ZINB* loss, NB with an NB loss, and direct log-transformed count reconstruction with an *MSE* loss.

However, we have noted that the *NB* and *ZINB* loss still have some notable issues. They can easily overflow, especially when working with lower precision systems (like fp16, bf16, etc). These losses are also proportional to the total expression count, meaning cells with higher expression will have a higher loss on average. It also appears that the log-likelihood cannot go below ~1.1 loss on average and plateaus quickly. This makes evaluation of the loss less practical when comparing models. Finally, this minimal loss also depends on the total number of zeros in the true expression dataset, as the zero-inflation part of the loss converges smoothly to 0.

The bottleneck learning task

Bottleneck learning is a method that drives the model to generate a cell expression profile only from its embedding. Cell-embedding which can be passed again to that same model without the gene expression information, such that from the cell-embedding only, scPRINT can re-generate the cell's expression profile. The model thus finds the best compression of the cell's expression according to the information-theoretic theorem by Tishbi et al. .

While many transformer models and Geneformer directly use the average of gene embeddings to generate a cell embedding, this will likely squash the expression information. scGPT used another methodology (called MVC) to generate an embedding vector such that

$$x_{i,j} = \mathrm{e_i} \odot \mathrm{g_j} \tag{1.18}$$

where $x_{i,j}$ is the expression of gene j in cell i, and $\odot$ is the dot product. For each gene embedding $\mathrm{g_j}$, the embedding only contains information about the gene name, not gene expression. Regular MSE on each $x_{i,j}$ is then used as the training loss.

This pushes the cell embedding $\mathrm{e_i}$ to contain all the expression information of the cell i.

This is less computationally intensive to train than our bottleneck learning method. However, we have noticed poorer reconstruction through this methodology than ours (see Supplementary Table 5.4).

In our case, we consider that our model scPRINT can act as two parts of an autoencoder. The encoding part is when we give scPRINT the expression profile of a cell and retrieve a set of disentangled cell embeddings (see the Class decoder section of the methods). The decoder part is when we provide scPRINT only the gene labels without their corresponding expression values and the disentangled cell embedding in place of the empty placeholder embeddings (see Supplementary Figure 5.2.15).

This means the encoder is considered as

$$\mathrm{e_{A,i}} = scPRINT([\mathrm{g_O} + \mathrm{e_{0,i}} + \mathrm{l_0}, \mathrm{g_1} + \mathrm{e_{1,i}} + \mathrm{l_1}, ..., \mathrm{p_A}]) \tag{1.19}$$

where $\mathrm{e}_{A,i}$ is the output embedding of the placeholder embedding token A for the cell i (in our case, we use multiple (default, totalcount, cell_type, disease, sex, organism, ethnicity, sequencer). Then the decoder is defined as

$$\mu_\mathrm{i}, \theta_\mathrm{i}, \pi_\mathrm{i} = scPRINT([\mathrm{g_O} + \mathrm{l_0}, \mathrm{g_1} + \mathrm{l_1}, ...], \mathrm{e_{0,i}}, \mathrm{e_{1,i}}, ..., \mathrm{e_{t,i}}) \tag{1.20}$$

With μ_i , θ_i , π_i vectors of size n_{genes}. Finally, the loss is given by the ZINB loss:

$$Loss_{bottleneck} = \sum_{i=0}^{m} Loss_{ZINB}(\mathrm{x_i}|\mu_\mathrm{i}, \theta_\mathrm{i}, \pi_\mathrm{i}) \tag{1.21}$$

where $\mathrm{x_i}$ is the cell i expression profile and m the minibatch size.

Implementing a set of disentangled embeddings is not straightforward. In our case, we push the embeddings to be as different from one another as possible with a contrastive loss defined as

$$Loss_{contrastive} = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{i'}^{m} 1 - \cos(\mathrm{e_i}, \mathrm{e_{i'}}) \tag{1.22}$$

where e_i and $e_{i'}$ are the cell embeddings, m is the minibatch size, and cos denotes the cosine similarity. This pushes each embedding to represent the correct information using the classifiers. However, more is needed to remove all the batch effects or entirely prevent information leakage across embeddings.

Finally, we have also used the classifier output logits as cell embeddings. This works particularly well for cell type, disease, or sequencer classes containing many labels. It has been shown that classifier logit outputs behave similarly to embeddings and, in our case, offer an even better removal of the batch effects (See Supplementary Figure 5.2.7).

For the bottleneck loss, we directly reconstruct expression using the cell embeddings generated from the noisy, downsampled expression profile of the denoising process, doing the entire process in one single pass. We sum all the losses without scaling them:

$$Loss = Loss_{contrastive} + Loss_{bottleneck} + Loss_{denoising} + Loss_{class} \tag{1.23}$$

1.5.4 scDataloader

Parallel to this work, we worked with Lamin.ai to develop a dataloader for large cell atlases, described and benchmarked in Rybakov et al.. One key advantage of this dataloader is its ability to perform weighted random sampling on hundreds of millions of cells without being a bottleneck during pretraining. scDataloader samples cells amongst the 800+ datasets of cellxgene's mid-2023 release, using the cell labels to inform how rare the specific combination of labels is.

From this, the dataloader produces a cell sampling weight, rescaled with a hyperparameter. The dataloader will sample, with replacement, more consistently rare cell types than more common ones.

We have produced an additional wrapper package around the laminDB "mapped-dataset" called scDataloader. scDataloader works with lamin.ai but can also interface with scVI and AnnData formats to enable downloading, preprocessing, and QC of large single-cell databases and datasets. It is very flexible and can represent expression data in the formats used by scPRINT, scGPT, and Geneformer. It also implements a lightning datamodule scheme and command line interfaces for quick setup (see Supplementary Figure 5.2.16).

Overall, we preprocess each of the 1200 datasets in cellxgene by only keeping primary cells from either humans or mice and dropping all the spatial omics datasets. Spatial omics are not true single-cell assays, and we decided for now not to include them. We also drop any cells with less than 200 expressed genes. Finally, we drop any resulting dataset smaller than 100 cells, with less than 10,000 genes, or from which more than 95% of the cells have been removed. This results in a new database of 54,084,961 cells and 548 datasets.

We believe that the weighted random sampling strategy allowed our pre-training to be much faster by creating more diverse minibatches.

1.5.5 Extracting meta-cell gene networks from attention matrices in scPRINT

Transformers compute multiple attention matrices per layer, called attention heads. This is done by splitting the generated $\boldsymbol{K}$, $\boldsymbol{Q}$, and $\boldsymbol{V}$ embedding into m sub-embeddings, thus defining m attention heads. Each attention head computes the attention matrix via the equation:

$$\text{softmax}\left(\frac{\text{QK}^T}{\sqrt{d_k}}\right) \tag{1.24}$$

However, we would want to aggregate those over multiple cells from a similar cell state to increase the signal obtained from only one cell. We are doing so by averaging the Keys and Queries embeddings over the set of cells U passed to the model:

$$\text{softmax}\left(\frac{mean_U(\text{Q}) \cdot mean_U(\text{K})^T}{\sqrt{d_k}}\right) \tag{1.25}$$

By doing this, the attention matrix behaves as if each query vector for cell i was "looking" across the key vectors of all the cells in U.

The resulting object is a row-wise normalized n^*n matrix, where n is the size of the input context (i.e. the number of genes passed to the model). However, we also include the possibility to generate large matrices and gene networks, referred to as genome-wide gene networks. We take the average over different sets of expressed genes for each cell in the set U. This allows us to compute a genome-wide attention matrix while only doing forward passes on smaller subsets of the genome per cell.

1.5.6 Heads selection

With scPRINT, we present a method to select heads based on some available ground truth data. This is inspired by the ESM2 paper and uses a somewhat similar method. Using all the available attention matrices from all of the model's heads, we use a linear classifier RidgeClassifier from scikit-learn (with an L2 penalty set to 1, a positivity constraint on the coefficients, and without an intercept) to classify the ground truth's edges based on a combination of each head. The classifier converts the target values into {-1, 1} equals to {no connections, connections} and then treats the problem as a regression task with mean squared error.

Instead of taking the classifier's output, we use the average of the subset of heads associated with a non-zero coefficient in the classifier, without weighting them. Thus, the classifier only serves as a means to select the heads with relevant information in predicting a ground truth of interest and decreases the possibility of overfitting (see Figure 1C).

1.5.7 Normalization and network interpretation

In scPRINT and scGPT, the attention matrix is normalized via the softmax function over the query (i.e., row) dimensions. This means that all row elements sum up to 1 or that the same mass flows from each network component. This rescaling is essential as it corrects that some row element scales can be much higher than others in the attention matrix. Similarly, in regularized models like GENIE3, only a small set of genes are connected for each gene in the matrix, meaning all genes have directed edges toward a small subset of genes. Thus, our interpretation is that the row elements are the targets in our network, each connected to a small subset of genes. The column elements are thus the regulators and can regulate many / most genes in the network.

For biological ground truths like MCalla et al. and gwps, which fit this assumption of highly connected regulators and sparsely regulated targets, we directly compare them to the inferred network. Tables S12 and S13 show that this performs better than taking the opposite view by transposing the inferred networks.

This assumption is challenged for Omnipath, which has most of its elements connected to a sparse set of other elements (see Supplementary Figure5.2.3). Due to the sparsity of connections for regulators (i.e., sources) in the ground truth network and the large number of regulators (8000+), the methods are challenged and perform much better when taking the transpose of their network and matching the regulators to the sources and sources to regulators.

1.5.8 Simulated datasets, BoolODE and Sergio

BoolODE is a method to generate count data via a stochastic differential equation applied over a user-defined Boolean network. It was used and developed as part of the BEELINE benchmark algorithm, which was created as an improvement over the GeneNetWeaver algorithm. However, this model is still very simple compared to cell biology. Due to its computational complexity, it can only model up to a couple hundred gene relationships over a few dozen genes.

Sergio, a slightly more recent ODE model marks an improvement over BoolODE on the size of the networks it can simulate (up to a thousand genes) and its similarity to scRNAseq data.

Indeed, Sergio's simulated data is not similar to real expression data. This means that the biases that Transformer models learn should not help them predict Sergio's data. Correlation and regression-based methods do not have biases. They are therefore expected and have traditionally shown better performance on these benchmarks.

We generated the Sergio ground truth network and simulated single cell expression by using the notebook: https://github.com/g-torr/SERGIO/blob/v2/minimal_example.ipynb from the repository: https://github.com/g-torr/SERGIO which present some debugs and improvements to the initial repository: https://github.com/PayamDiba/SERGIO. Indeed only this fork of the initial Sergio repository led us to successfully generate a

network.

We used RegNetwork as input and simulated 1000 cells from its 3546 connections over 813 genes with default parameters from the notebook.

1.5.9 BenGRN and gene network metrics

We use the packages benGRN and GRnnData released with this manuscript to work With Gene networks and perform our benchmarks.

Our three main metrics are EPR, AUPRC, and enrichment. They all take advantage of the fact that the predictions are generated as scores over edges between nodes:

- We have computed the Early Precision Ratio (EPR) as the diagnostic odds ratio: (TP x TN) / (FP x FN) at the cutoff of the scores giving K positive predictions, where K is the number of positive elements in the ground truth.
  In this context, 1 is a random prediction, and inf is a perfect prediction; values below one mean that inverting the predictor would provide better results.
- Area Under the Precision-Recall Curve (AUPRC) is the area (computed with the composite trapezoidal rule) under the curve defined by the precision ($PR = TP / (TP + FP)$) and recall ($RE = TP / (TP + FN)$) where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. This curve is obtained through a range of cutoffs going from 0 predicted positives to all predicted positives. Here, we compute a version of the AUPRC where the floor of the area is not given by the Precision=0 line but by the line of the prevalence of the positive class. Moreover, we do not interpolate the curve between the last recall value and the perfect recall: 1. We do this to properly compare AUPRC values across benchmarks and models. Random precision values are given in the supplementary data.
- Enrichment is computed using the prerank methodology, where, given an ordered set of genes, it is computed by:
  - 1. Summing all scores of edges of the matrix row-wise. (Target - Hub) Or
  - 2. Summing all scores of edges of the matrix column-wise. (Regulators - Hub) Or
  - 3. Computing the eigenvector centrality of nodes in the graph using NetworkX's implementation. Prerank's background comprises all the genes in the set (centrality).

Of note, we did not design an automated method for cell-type enrichment. Instead, the assessment of whether or not a network is enriched for the correct cell type is done manually, identifying cell type names in the top 10 cell types listed in the enrichment results of the network.

1.5.10 Other evaluation metrics

All evaluation metrics from the section "scPRINT is competitive on tasks orthogonal to GN inference" of the results come from the openproblems benchmark and are standards in the field.

scIB's batch correction score is an average of the avgBatch score and the avgBio score, which are themselves averaged over many scores. Details of each value are available in our package's notebooks.

- scIB avgBio is a combination of label-based and label-free metrics using for example: the Adjusted Rand Index (ARI) and the Normalized Mutual Information (NMI) on clusters computed from the K-Nearest Neighbor graph. Other scores are used, some using the conservation of trajectories and of the cell cycle variance, and some on the rare cell population conservation, overlap of highly variable genes (see scIB), and more.
- scIB avgBatch is a similar combination of label-based and label-free metrics, using, for example, the average connectivity across clusters of different batches: ASW, the graph integration local inverse Simpson's Index: graph iLISI, the k-nearest-neighbor Batch Effect Test (kBET), and more.

Finally, we also use two metrics in our classification task:

- Macro-F1: also called macro-average, is the average of the F1 score across each class in a multi-class task. Where the F1 score is: $2 \times \frac{PR*RE}{PR+RE}$.
- Accuracy: the accuracy is computed as $\frac{TP + TN}{TP+TN +FN+FP}$

1.5.11 Denoising Benchmarks

To validate the denoising ability of scPRINT, MAGIC, and KNNsmoothing2, our test function, available in the scPRINT package, uses a representative subset of 10,000 cells of each dataset to generate the denoised expression over the 5000 most variable genes in this dataset.

Before that, counts are removed from the dataset following the same procedure as done for scPRINT's pretraining (see The denoising task section of the methods).

For each cell, we compare the denoised and un-denoised profiles to the true profile (e.g. before denoising). We compute the Spearman's correlation over the genes initially expressed in the cell, taking the average across all cells. We do not use the unexpressed genes as we are working with a dataset with high dropout and expect that a good denoiser will set genes that are 0 in the profile with some value. We notice that this improves the score of all denoising methods and makes more sense given the data.

For the rare cell population test, we keep everything similar but compute only the Spearman correlation over a rare cell population in the dataset.

We run KNNsmoothing2 with default parameters and a K of 10. We run MAGIC using the Scanpy implementation with default parameters and the approximate solver for computational speed. When computing KNNsmoothing2 or MAGIC over a small set of cells we use a K of 5 for the nearest neighbors.

1.5.12 Fine-tuning

Contrary to most other foundation models for scRNAseq, we do not finetune scPRINT at any moment in our benchmark and all results are provided for the pre-trained model only.

While we haven't assessed fine-tuning we believe this is an important feature of foundation models and release various scPRINT models so that they can be re-trained, fine-tuned, and modified by the community for novel tasks or to improve its performance on the tasks we have presented.

1.5.13 State-of-the-art methods used in benchmarking

All methods presented here generate networks from their input data. Given gene-level expression data, they will generate gene-networks. Without additional information, no method can distinguish the type of molecular interactions that underpin their predicted network edges.

Gene network inference with an ensemble of trees (GENIE3)

Developed originally for bulk transcriptional data, *GENIE3* computes the regulatory network for each gene independently. It uses a random forest, a weak learner ensemble method, to predict the expression profile of each target gene from profiles of all the other genes. The weight of an interaction comes from the feature importance value of an input gene in the predictor for a target gene's expression pattern. Aggregating these weighted interactions over all the genes yields the regulatory network. This method was the top performer in the DREAM4 in silico network challenge (multifactorial subchallenge).

GENIE3 can be seen as a generalization of correlation-based methods for inferring gene networks. Instead of looking at genes that correlate most with another gene, GENIE3 finds how to combine a set of correlated genes to get an even better correlation. We run GENIE3 on raw counts as it is said from both the BEELINE benchmark and the R package vignette that GENIE3 can be run on either log normalized or raw count data and that while it will change the results, there are no preferred methods. This is something we have also noted in our trials.

We use all default parameters and choose 100 trees for computational feasibility reasons.

We compute the networks on the same set of cells and genes as the other methods.

We also use a TF-gene only version of the method where the regression is performed only using the expressed transcription factors instead of all expressed genes as input. This is the most used version of *GENIE3* and is much faster.

DeepSEM

DeepSEM is an autoencoder model made for gene network inference. It learns to decompose a set of cells as a set of embedding and an adjacency matrix (i.e., a gene network). The formula of the VAE then becomes: $X = f_1((I - W^\top)^{-1}Z)$, for the decoder and $Z = (I - W^\top)^{-1}f_2(X)$ for the encoder, where X is the expression data, Z is the embedding dimension, W is the adjacency matrix, I the identity, and f_1, f_2 are MLPs.

We preprocess the anndata by normalizing gene expression to 10,000 genes, applying a logp1 transformation, and then computing the z-score per gene, as explained in the associated research paper.
We use DeepSEM with default parameters and on the same set of cells and genes as the other methods. We use the DeepSEM-provided functions for loading and parsing Anndatas.

Single-cell generative pretraining transformer (scGPT)

scGPT is a transformer-based model of roughly 100M parameters, pre-trained with a generative process similar to Language models. scGPT proposes to build similarity networks based on the output gene embeddings of the model but also based on its attention matrices. It computes networks as the difference between the rank-normalized version of the average attention matrix in a baseline expression profile vs a perturbed one in perturb-seq data. The attention matrix is the average of attention matrices over the heads of the last layer and over the cells given to the model.

We run scGPT following the examples given in their "Tutorial_Attention_GRN.ipynb" notebook.

We use the "scGPT_human/best_model.pt" from the list of available models with default parameters. All runs are in our fork: "https://github.com/jkobject/scGPT" in the "mytests/" folder. Similarly, we take the mean over cells and over the heads of the last layer. We compute softmax similarly to the attention computation but without applying the rescaling factor $\sqrt{d_k}$. We finally drop the first element corresponding to the cell embedding token.

We extract cell embeddings from scGPT by directly using the cell embedding token of the model without fine-tuning it on a batch correction task. This is done in order to compare it to scPRINT which is itself not fine-tuned. We compute the networks on the same set of cells and genes as the other methods.

Geneformer

Geneformer is a BERT model. Gene expression data is transformed into a sentence or genes ordered by their scaled expression. It is trained with mask language modeling and contains somewhere around 80M parameters. We use the new versions of 2024 Geneformer models trained on 100M cells (2x more than scPRINT). We follow the preprocessing and inference scripts used in the geneformer huggingface repository and notebooks: https://huggingface.co/ctheodoris/Geneformer/tree/main. Our inference script updates to extract gene networks from Geneformer are available in our scPRINT repository: https://github.com/cantinilab/scPRINT/tree/dev/tools.

We extract gene networks from Geneformer using the mean of all attention heads per cell. Since Geneformer only uses expressed genes in a cell, we have to map the attention matrices back to the full network size before computing its average over cells, taking into account the NaN values. We compute the networks on the same set of cells and genes as the other methods.

We extract a cell embedding from Geneformer using the cell embedding from the "gf-12L-95M-i4096_MTLCellClassifier_CELLxGENE_240522" model that has been fine-tuned on predicting the cell labels of cellxgene datasets.

scFoundation

scFoundation is a foundation model for single-cell RNAseq based on the xtrimogene architecture. It was built by the Biomap company. It is able to work on the full genome sequence of transcripts for each cell by considering the high number of zeros and embedding them separately. The tool is aimed at performing a range of tasks, such as denoising, embedding, and predicting perturbation response. It has been trained with a mixed masking and denoising pre-training. However, we could not compare scFoundation to scPRINT and MAGIC on the denoising benchmark, as scFoundation's denoising only happens at the level of the cell embedding at inference time.

We could not validate scFoundation on our Gene network inference benchmark as extracting a network from the attention matrices was much more complex due to the xtrimogene architecture. scFoundation mentions the generation of gene modules using clustering of its output gene embeddings. It also mentions the interference of gene networks. However, it is achieved using RcisTarget, a prior gene network based on motif analysis. This approach is not comparable to the gene networks generated by scPrint, Geneformer, and scGPT. Indeed, RcisTarget could be applied to every model we have benchmarked and would prevent us from doing an unbiased benchmark. Neither our approach nor Hao et al.'s could extract gene networks directly from scFoundation. It is being left to further investigations.

For batch effect correction, we use scFoundation with default parameters and follow the steps for cell embedding in the "model/README.md" file in their GitHub repository: https://github.com/biomap-research/scFoundation. However, we give scFoundation single cell profiles of the 5000 most variable genes in each dataset. This is because we could not run scFoundation on genome-wide expression profiles with our GPU. We then apply a PCA

to the output embedding to reduce the dimensionality from 3224 to 512. This is because the initial dimension was too high for scIB to compute a score from on our machine (40CPU Intel Xeon, 32GB RAM + 64GB SWAP, GPU NVIDIA A4500 with 20GB of memory).

Marker-based cell type prediction with CellTypist

To showcase the novel ability of scPRINT to perform zero-shot prediction of cell type labels, we use the CellTypist method, which similarly performs de-novo prediction of cell type labels given its precomputed databases of cell type markers.

CellTypist works by mapping cell gene expression to genes known to be specifically expressed in combination in a cell type. Thus, it predicts cell type from these marker genes.

We use it with default parameters on the normalized and log-transformed counts over the full set of genes in the dataset. We use the 'Adult_Human_PancreaticIslet' database, which contains markers for 14 cell types and overlaps with only four of the cell types in the dataset.

We decided to still use it as is to showcase the marker-based method's inability to recover the full set of cells and the tradeoff between the number of cell types and accuracy.

Fortunately, these four cell types (A, B, D, PP) represent 70% of the dataset. With its current database, CellTypist can only reach a maximum accuracy of 70%. Even when taking this into account, CellTypist only overperforms scPRINT on the accuracy metric and by roughly 9 points.

Classification benchmark and associated methods

Our classification benchmark is run using following the openproblems benchmark. It uses the same input, output data, and metric. It also similarly splits the train-test by batch and preprocesses the expression matrix to what is presented in the open problem benchmarks.

For this task, methods can access the full set of genes by default. scPRINT will use its random sampling of genes approach with a context of 4000 genes. Classifiers like logistic regression and xgboost were run according to the openproblem process, using the 25 principal components of the count normalized, logp1 transformed expression data. CellTypist was run on the normalized and logp1-transformed cell expression profile.

1.5.14 Ground truth preparation

McCalla et al.

For the MCalla et al. dataset, we downloaded the data from the supplementary datasets of their paper . After undoing the logp1 transform, we re-generate the true count expression matrix from the normalized one by dividing the expression of each cell by the smallest value

in its expression profile. This fully recovered the true counts, all values being integers. For the additional human dataset we used, we downloaded it from the gene expression atlas database.

We used the intersection (gold standard) ground truth dataset for both human and mouse, converting this list of sources to target genes into a directed binary network.

Omnipath

We generate the Omnipath network using all the interactions from the Omnipath Python package, excluding small molecules, lncRNAs, and any element without a unique HGNC symbol. We then transform it into a directed binary network of source to target. These interactions are extracted from the literature and represent mainly TF to gene connections as well as many protein-protein interaction connections and a small number of other connections known from the literature like RNA-RNA interactions, protein-RNA interactions, and more. All interactions are mapped back to their gene IDs, generating a gene-gene network encompassing the various interactions the genes and their molecular products can have.

Gene networks from genome-wide perturb-seq

We created a gene network from the genome-wide perturb-seq dataset using the supplementary matrix containing the results of differential expression in the dataset. This matrix represents the multiple hypothesis testing corrected p-values of a differential expression test of cells with KO of gene A compared to the baseline cell expression. This is available for all 8000+ expressed genes in the K562 cell line. We used a cutoff of 0.05 on these values to define the directed binary connection between genes.

This effectively gives a gene x gene-directed binary graph that tells if a statistically significant connection exists from the source $gene_A$ to the target $gene_B$ according to genome-wide perturb-seq.

For all ground truths, download, preprocessing, and extraction of the network and expression data are available in the BenGRN package.

1.5.15 Details on the Benign Prostatic Hyperplasia analysis

We download our dataset from cellxgene under the reference: *574e9f9e-f8b4-41ef-bf19-89a9964fd9c7*.

We preprocess the dataset using scDataloader's preprocessing function. We generate embedding and classification using 3000 expressed genes in each cell. Similarly to pretraining, we take 3000 randomly expressed genes; if less than 3000 are expressed, we complete with randomly selected unexpressed genes. We display embeddings generated using the cell type classifier logits (see section The classification task in methods)

We use the Scanpy toolkit to generate our Umap plots directly from the embeddings, as well as our differential expression results and our clusters. We define the clusters using the Louvain algorithm with 10 k-nearest-neighbors and a resolution of 1. We perform denoising on 5000 genes per cell selected similarly to the embedding and classification part. We use the 4000 most variable genes in each cell type to generate our gene networks in the BPH and normal fibroblasts.

On the gene networks, we perform gene set enrichment with the Enrichr method on the GO_MF_2023 gene sets. For community detection, we use the Louvain algorithm with parameter 1.5. We perform analysis only on the communities with between 200 and 20 genes. (4 and 5 in the BPH-associated fibroblasts, 3 and 4 in the normal fibroblasts)

All analysis and results are available in the *cancer_usecase_1* and *cancer_usecase_2* notebooks.

1.5.16 Negative Binomial to Poisson relationship

As explained in The denoising task and Expression decoder section of the methods, in our model, we have used the ZINB as our loss, an extension of the NB distribution to zero-inflated data.

Moreover, we have also used the zero-inflated Poisson mechanism to downsample the cell expression profiles. These are consistent because we can view the Poisson distribution as a NB without overdispersion. The relationship between *NB* and *Poisson* is given by making the dispersion term go to 0 and the inverse dispersion term $\theta \to \infty$. Doing so, the term $\frac{\theta}{\theta+\mu}$ approaches 1. Thus, the PMF simplifies to:

$$P(X = x) \approx \frac{\Gamma(x+\theta)}{x!\Gamma(\theta)} 1^{\theta} \left(\frac{\mu}{\theta+\mu}\right)^{x} \tag{1.26}$$

For large θ, we use Stirling's approximation of the Gamma function: $\Gamma(\theta) \approx \sqrt{2\pi\theta}(\frac{\theta}{e})^{\theta}$

we get:

$$\Gamma(x+\theta) \approx \sqrt{2\pi(x+\theta)} \left(\frac{x+\theta}{e}\right)^{x+\theta} \tag{1.27}$$

$$\Gamma(\theta) \approx \sqrt{2\pi\theta} \left(\frac{\theta}{e}\right)^{\theta} \tag{1.28}$$

Simplifying the ratio of the Gamma functions:

$$\frac{\sqrt{2\pi(x+\theta)} \left(\frac{x+\theta}{e}\right)^{x+\theta}}{\sqrt{2\pi\theta} \left(\frac{\theta}{e}\right)^{\theta}} = \sqrt{\frac{x+\theta}{\theta}} \left(\frac{x+\theta}{\theta}\right)^{\theta} \left(\frac{x+\theta}{e}\right)^{x} \tag{1.29}$$

For large θ, $\frac{x+\theta}{\theta} \sim 1$, so: $\sqrt{\frac{x+\theta}{\theta}} \approx 1$

$$(\frac{x+\theta}{\theta})^{\theta} \approx 1$$

Thus, the expression simplifies to:

$$P(X=x) \approx \frac{1}{x!}\left(\frac{\mu}{\theta+\mu}\right)^{x}\left(\frac{\theta+x}{\theta}\right)^{x} \tag{1.30}$$

Finally, $(\frac{x+\theta}{\theta+\mu})^{x} \approx 1$ for large θ, so:

$$\lim_{\theta\to\infty} P(X=x) = \frac{\mu^{x}}{x!}e^{-\mu} \tag{1.31}$$

This is the PMF of the Poisson distribution with mean μ.

1.5.17 Data availability

The model weights are publicly available on Hugging Face. Pre-training logs to assess the model's training are available on Weights and Biases. The full pre-training dataset is publicly available on CellxGene under its census data release version: LTS 2023-12-15, accessible at https://cellxgene.cziscience.com/. All other datasets used in this work can be downloaded through their respective public databases via the helper scripts on the scPRINT, BenGRN, GRnnData, and scDataLoader packages. Source data are provided to re-generate the figures. Code to generate the large UMAP of Figure 1 is available as a notebook on GitHub at https://github.com/cantinilab/scPRINT/blob/1.6.4/figures/nice_umap.ipynb. Code to re-generate the source data is available as notebooks on our Github.

1.5.18 Code availability

The code and notebooks used to develop the model, perform the analyses, and generate results in this study are publicly available and have been deposited in cantinilab/scPRINT at https://github.com/cantinilab/scPRINT under MIT license. The specific version of the code associated with this publication is archived in the same repository under the tag 1.6.4 and is accessible via https://github.com/cantinilab/scPRINT/tree/1.6.4/ and DOI:10.5281/zenodo.14749466.

Additional developed packages for this analysis are defined in the pyproject file and project submodules. They are available on GitHub:

- **GrnnData**: https://github.com/cantinilab/GRnnData, DOI:10.5281/zenodo.10573141

- **BenGRN**: https://github.com/jkobject/benGRN, DOI:10.5281/zenodo.10573209
- **scDataLoader**: https://github.com/jkobject/scDataLoader, DOI:10.5281/zenodo.10573143
- **scGPT and notebooks to reproduce the results**: https://github.com/jkobject/scGPT/tree/main/mytests

Xpressor: Towards foundation models that learn across biological scales

2.1 Summary

Biological foundation models now exist across four scales: molecules, sequences, cells, and tissues; yet, they operate in isolation. We present a general approach, called Xpressor, for multi-scale learning via (1) compression, using a cross-attention mechanism, and (2) fine-tuning of the compression model. Applied to a cell foundation model, our gene-to-cell compression approach improves cell-type prediction (+28%) and embedding quality (+8%). Our amino-acid-to-gene fine-tuning approach also yields improvements across all metrics. The Xpressor represents a first step towards models that bridge biological scales.

2.2 Introduction

While it could be said of most domains, biology is a great example of a system that processes information across scales, from molecules to tissues. Foundation models now excel at specific scales: e.g., protein structures [70], cell states [178, 73], but operate in isolation, unable to leverage cross-scale connections.

Our premise is that lower-scale information (e.g., molecular) can improve higher-scale representations (e.g., cellular), and vice versa [5, 179]. While learning across all scales simultaneously is infeasible, we propose to compose pre-trained "uniscale" models through architectural updates and fine-tuning (see Figure 2.1).

2.2.1 Bio-foundation models across scales

Bio-foundation models exist at four biological scales (see Supplementary Material 2.4.5 for an extended review):

Molecular FMs (mFMs) encode molecules via SMILES or 3D coordinates to predict binding, solubility, and dynamics [180, 181].

Nucleotide FMs (nFMs) use transformer-based language models on DNA, RNA, and protein sequences [70, 182]. We include protein language models in this category, as their distinctions are blurring [183].

Cell FMs (cFMs) are trained on single-cell RNA-seq abundance matrices using encoder transformers [178, 73, 72]. Challenges remain around data noise and coverage [184].

Tissue FMs (tFMs) learn spatial cell relationships from microscopy using vision transformers [185, 186]. Their tokens could leverage cell representations from cFMs.

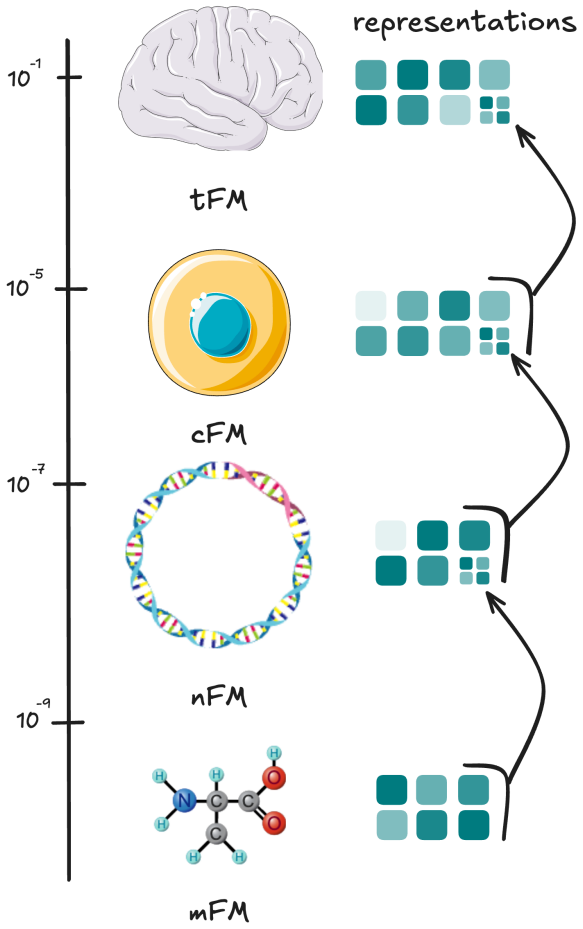


Figure 2.1: Biological scales and how foundation model representations at each scale can inform adjacent scales. In red, we apply our proposed Xpressor approach.

2.2.2 Existing approaches

Architectural modifications: compressed representations Prior work on biological representations includes VAEs [38, 187] and quantized embeddings [188, 189]. Simple pooling of transformer outputs performs poorly [190, 191]; state-of-the-art methods instead use cross-attention mechanisms. We adopt a similar approach to compress foundation model outputs into lower-dimensional vectors.

Training modifications: fine-tuning Fine-tuning approaches range from full model updates to parameter-efficient methods like LoRA [192, 193] and adapter layers [194], small MLPs in-between model layers or outputs, each offering a lightweight and parameter-efficient approach. We will use adapter layers as a simpler proxy to Xpressor for fine-tuning lower-scale models.

2.2.3 Contributions

Following up on these recent advances, we propose:

- A cross-attention "compressor" block whose goal is to compress a foundation model's output embeddings into a small set of low-dimensional vectors, called the *Xpressor* (Cross-Attention Compressor transformer). This is learnt using an auto-encoding approach with a reconstruction loss. The Xpressor is modality-agnostic and can be used by mFMs, nFMs, cFMs, tFMs, or any other transformer-based foundation model, unregarding of the pre-training tasks (see Figure 2.3A).
- A multi-scale fine-tuning approach using Xpressor. This allows the fine-tuning of models from one level using the upper-scale model's task (see Figure 2.3B).

2.3 Xpressor

2.3.1 Background

We use scPRINT [178] as our cFM and ESM2 [70] as our nFM. Below, we detail scPRINT's architecture and training, which inform our design choices.

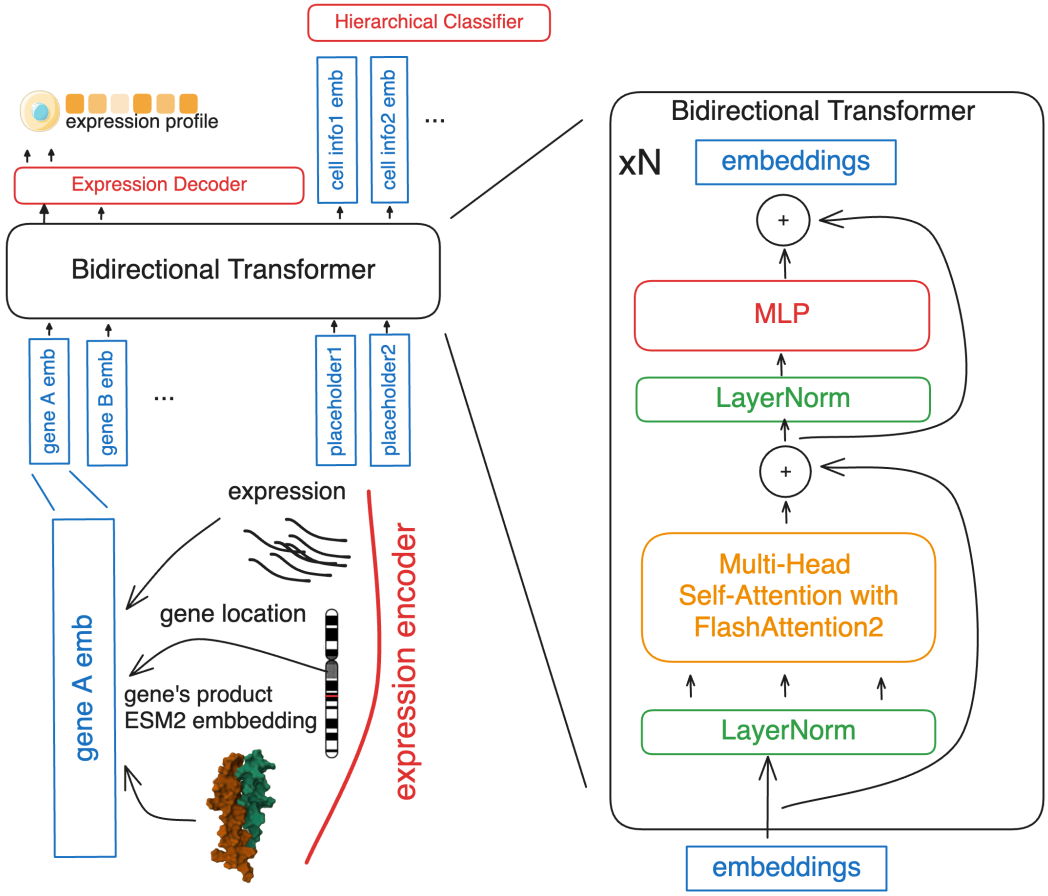


Figure 2.2: Overview of scPRINT's architecture: gene embeddings are formed by summing gene ID, expression, and positional embeddings. The transformer processes these along with cell and label tokens. The decoder outputs parameters of a zero-inflated negative binomial distribution for each gene. Inspired by the original Figure 1.A in Kalfon et al. [178].

Architecture. scPRINT is a bidirectional transformer trained on 50M+ cells from CellxGene [112]. Each gene in a cell is encoded as the sum of three embeddings: (1) a *gene ID embedding* from ESM2's mean-pooled amino acid embeddings of the gene's protein product, providing evolutionary and structural priors; (2) an *expression embedding* from an MLP applied to log-normalized counts $e_{i,j} = \text{MLP}(\log_2(x_{i,j} + 1))$; and (3) a *positional embedding* encoding genomic location, since co-located genes share regulatory regions. Additionally, learned placeholder tokens for cell embeddings and labels (cell type, disease, sex, etc.) are concatenated to form the input (see Figure 2.2).

The model processes 2,200 genes per cell during training (covering 80% of cells in CellxGene), padding with unexpressed genes when needed. This lets the model distinguish true zeros from dropouts. The expression decoder outputs parameters of a zero-inflated negative binomial (ZINB) distribution (μ, θ, π) for each gene, modeling both the mean expression and overdispersion.

Pretraining. scPRINT uses three jointly optimized tasks: (1) *denoising*, reconstructing expression from downsampled profiles (60% dropout), which encourages learning gene-

gene interactions [117]; (2) *bottleneck learning*, compressing and reconstructing profiles through cell embeddings (see Figure 2.5 in Supplementary Material 2.4.6); and (3) *label prediction*, classifying cell type, disease, and other annotations using a hierarchical classifier that handles ontology-structured labels of varying granularity.

Gene networks. scPRINT extracts cell-specific gene networks from its attention matrices, similar to ESM2's contact prediction. These networks can be subsetted to transcription factor-gene connections and benchmarked against ground-truth networks.

For protein representations, ESM2 [70] learns evolutionary constraints from amino acid sequences and enables structure prediction via ESMfold [195].

2.3.2 Approach

Our first contribution is the compression of output embeddings of foundation models using a transformer block and a bottleneck-learning method (whose basic theory is presented in Supplementary Material 2.4.7): we call it the Xpressor (see Figure 2.3A). Compression / decompression is the key mechanism for transferring representations across scales (see Supplementary Material 2.2.1). To do so, we introduce an additional set of transformer blocks called "Xpressor blocks". In the context of scPRINT, these blocks represent cell features.

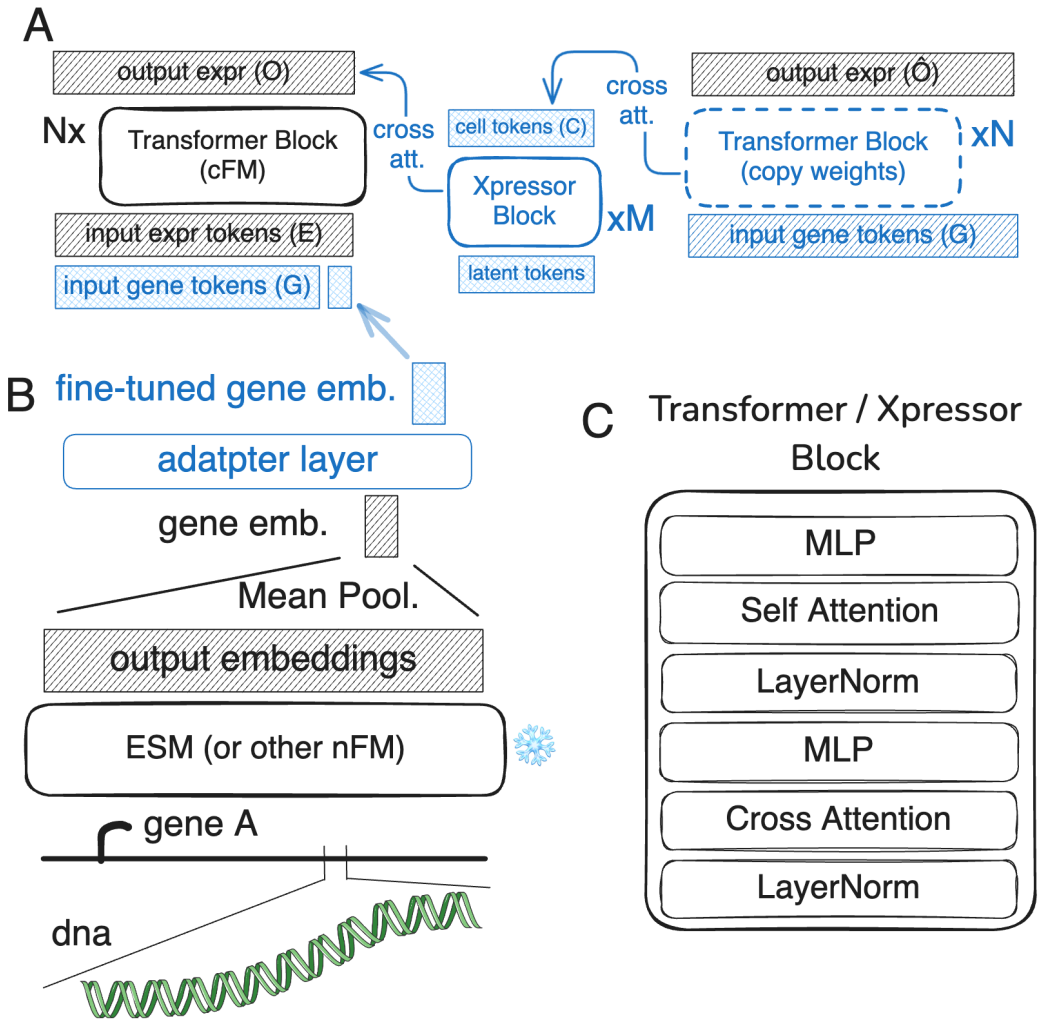


Figure 2.3: Overview of the Xpressor architecture and multi-scale fine-tuning approach applied to a cell foundation model. A. The Xpressor architecture, composed of M layers, shows how gene-level representations are compressed into cell-state vectors through cross-attention over the output embeddings of a transformer, composed of N layers. These compressed representations are then decompressed back using the same initial transformer model with cross-attention, given the initial gene-level tokens. B. Example of the multi-scale fine-tuning setup illustrating how the adapter layer enables joint training of gene-level representations that are then used by a cFM. C. Detailed structure of the transformer and Xpressor blocks showing the cross-attention and self-attention sub-blocks. Blue blocks are our contributions. Shaded blocks indicate inputs and outputs.

We keep scPRINT's input the same: we use a set of summed up gene expression and gene ID tokens. The first ones are generated using an MLP for each gene expression value in cell j, and the others are generated from ESM2's output embeddings for each gene, aggregated with mean-pooling. The newly proposed Xpressor block uses as input a set of learned latent tokens $\boldsymbol{T}$. It then performs cross-attention between the last layer of the gene embeddings and the latent tokens (see Figure 2.3A). The goal is for the Xpressor blocks to be of smaller dimensions and context size than the main blocks, such that we end up with $\boldsymbol{C}_j$, a set of n tokens of dimension d_t generated from the encoded gene expression and ID matrices $\boldsymbol{E}_j$ and $\boldsymbol{G}$. Where $\boldsymbol{G}$ and $\boldsymbol{E}_j$ are sets of m tokens of size d_c representing the IDs of the genes

and their corresponding expression in cell j, respectively, where $d_c < d_t$ and $n << m$. In this context, for a cell j, scPRINT does:

$$\boldsymbol{O}_j = \text{scPRINT}(\boldsymbol{E}_j + \boldsymbol{G}) \tag{2.1}$$

Where scPRINT implies N layers of transformer blocks performing self-attention (see figure 2.2 for the transformer architecture). Xpressor is then applied on its output:

$$\boldsymbol{C}_j = \text{Xpressor}(\boldsymbol{O}_j, \boldsymbol{T}) \tag{2.2}$$

Where Xpressor implies M layers of transformer blocks performing cross-attention on the output embeddings $\boldsymbol{O}_j$ and self attention on $\boldsymbol{T}$, with $\boldsymbol{T}$, the learned set of input cell tokens, and $\boldsymbol{C}_j$ being the cell tokens associated with the input expression $\boldsymbol{E}_j$.

Both scPRINT's and Xpressor's transformer blocks possess a cross-attention architecture (see Figure 2.3C) such that we can finally do:

$$\hat{\boldsymbol{E}}_j = \text{scPRINT}(\boldsymbol{C}_j, \boldsymbol{G}) \tag{2.3}$$

Where scPRINT this time performs cross-attention on the cell tokens $\boldsymbol{C}_j$ and self-attention on the gene ID tokens $\boldsymbol{G}$ to reconstruct the expression values $\hat{\boldsymbol{E}}_j$.

textcolorredIn this example, the decompression is done with gene ID tokens as input only ($\boldsymbol{G}$) (see Figure 2.3A). These tokens are expression independent and thus do not depend on j. In the context of protein language models, for example, this would be replaced by positional tokens.

As shown in Figure 2.3A, the *Transformer* blocks are applied twice: first as an encoder (self-attention only), then as a decoder alongside the *Xpressor*. Unlike the original transformer [66], cross-attention is performed first. Related ideas appear in Lee et al. [190], where a single cross-attention layer, together with prompting and fine-tuning, is used to compress the context of a large language model.

In the cell foundation model context, the goal of the *Xpressor* is to perform compression of the gene id & expression tokens into a set of cell tokens similar to the classical information bottleneck from Tishby, Pereira, and Bialek [196] (see Supplementary Material 2.4.7). And vice-versa, to decompress the cell tokens back into an expression profile. This, together with the scPRINT's denoising and classification tasks, compose our pre-training objective (see Supplementary Material 2.4.6).

In our case, each embedding corresponds to a different cell component. At training time, we present multiple losses to both regularize it and ensure differences across them, similar to what can be done in VAEs (see Supplementary Material 2.4.8).

Table 2.1: Comparison of cell embedding approaches

Model	Cell Label Pred.	Embed. Quality	Gene-Net Infer.
Class-pooling	0.56	0.49	3.8,1.5
Xpressor	**0.72**	**0.53**	**4.1,1.7**
negative ctrl.	0.00	0.37	1.0,1.0
default methods	0.5	0.44	3.5,2.2

2.3.3 Results

We show that such an instantiation of the transformer leads to better performance over the gymnasium of tasks available in the scPRINT cFM.

Indeed, we now look at three specific tasks: cell-type prediction, embedding quality, and gene-network inference. The tasks are the same as presented in Kalfon et al. [178].

"Embedding quality" refers to the average scIB [144] score for batch correction and biological consistency of cell embeddings. In this context, scIB assesses the quality of embeddings using measures of similarity, nearest neighbors, and clustering.

Cell-label predictions are generated using a classifier on top of the cell embeddings generated by each model. We follow the approach of Kalfon et al. [178] here, which was recently presented with a different mechanism in Wang et al. [197]. This classification task allows us to see how one can steer the model's embeddings to represent meaningful biological features.

Finally, we display two different metrics for gene-network inference. The gene network inference benchmark aims to estimate the quality of the self-attention matrices by comparing them to a gene-gene ground-truth matrix. Here we use EPR, an odds-ratio measure in which, e.g., a value of N means that the predictions are N times as likely to be correct as a random guess. One is the EPR score on the genome-wide perturb-seq gene network from BenGRN [178], while the second is the average EPR of multiple predicted gene networks across various cell types compared to BenGRN's omnipath ground-truth gene network [129].

In our comparison, the class-pooling approach is used, similarly to other cell foundation models, like scPRINT & scGPT [73, 178], where a class token is added to the model's input and an additional loss is placed on it: $argmin_{C_j}(||E_j - \boldsymbol{C}_j\boldsymbol{G}_j^T||_2)$. Both models use the same latent dimensions, architectures, training paradigm, and number of input tokens for both genes and cells.

We see that the Xpressor outperforms the simpler class-pooling approach on embedding quality and cell-label prediction, while gene-network inference results remain roughly the same.

We have also added two baselines: a negative control using an untrained model, and a default method where, for each task, common unsupervised single-cell methods are used: PCA for embedding quality, CellTypist [198] for cell-label prediction, and GENIE3 [67] for gene-network inference.

We will now see how we can further train -or fine-tune- these representations using information from the upper scale. While Xpressor layers with their small set of low-dimensional tokens are best suited for this task, we will focus on a commonly available foundation model without modifying its architecture.

2.4 Multi-scale Fine-tuning

2.4.1 Background

To merge foundation models, we need a way to connect the lower-scale models to the upper one. It had been proposed in Rosen et al. [76] and Kalfon et al. [178] to use protein language model-based representations, like those of ESM2, as input tokens for the models. This decreases the number of parameters the model has to learn; it allows the model to work on genes unseen at training time; moreover, it lets the model use information it would not have gained otherwise, such as protein structure, homology, and mutations.

2.4.2 Approach

We propose going beyond simply reusing lower-scale models' representations and fine-tuning them during the pre-training of the upper-scale model. We would use a foundation model pretrained with Xpressor blocks and updating only the Xpressor blocks during fine-tuning to achieve high parameter efficiency.

However, this exercise would require us to pre-train and assess a second foundation model. We opt instead for a proxy to Xpressor: using an adapter layer (see Figure 2.3B) on top of a mean-pooling of ESM2's output embeddings. This approach is strictly less expressive than Xpressor and serves as a lower bound on the performance gains from multi-scale fine-tuning. With the adapter layer, each output embedding matrices $\boldsymbol{S}_k$ of a genetic sequence k of length N from ESM2 is mapped to a point in space $\boldsymbol{i}$ using a smooth & trainable function $MLP()$, a 2-layer neural-network:

$$\boldsymbol{i}_k = MLP(\frac{1}{N}\sum_{i=0}^{N} \boldsymbol{s}_{k,i}) \tag{2.4}$$

By using a neural-network, we allow for an interpolation of the initial lower-scale representation towards a representation containing features learned from the upper-scale data [199]. In our case, we use ESM2 as the lower-scale model and scPRINT as the upper-scale

Table 2.2: Comparison of input-gene embedding approaches

Model	Cell Label Pred.	Embed. Quality	Gene-Net Infer.
Random init.	0.62	0.48	4.5,1.0
ESM2 frozen	0.56	0.49	3.8,1.5
ESM2 fine-tuned	**0.70**	**0.49**	**4.5,2.4**

model. The initial ESM2 embedding is known to capture the protein's sequence, evolutionary similarity, and constraints.

Indeed, this is what allows this representation to replace the multiple sequence alignment (MSA) step in ESMfold [195]. We posit that this initial embedding already contains the information necessary to understand some rules governing gene interactions (homology and similar evolutionary constraints). However, representations from ESM2 differ significantly from those of single-cell foundation models. Our goal is to enrich these representations with knowledge gained from co-expression information across millions of cells.

2.4.3 Results

We show that a cFM trained on the pooled embeddings of a pretrained nFM performs better on most tasks from the Kalfon et al. [178] gymnasium benchmark than one trained on learned representations (see Table 2.2). This is possible because we allow the model to start from a very rich representation instead of a random set of vectors, while still giving it the flexibility to incorporate additional knowledge. Each foundation model tested uses the same latent dimensions, architectures, training, and number of input tokens. We report performance at the last epoch; training is stopped after 20 epochs.

We also show the difference in cell embeddings obtained between the regular transformer and the Xpressor (see Figure 2.4). The dataset is a challenging mix of modalities with varying batch effects and levels of noise. Cell types are also quite similar, making the task more difficult. We can see that the Xpressor embeddings exhibit more structure and better resolve different cell types than a transformer with class-pooling.

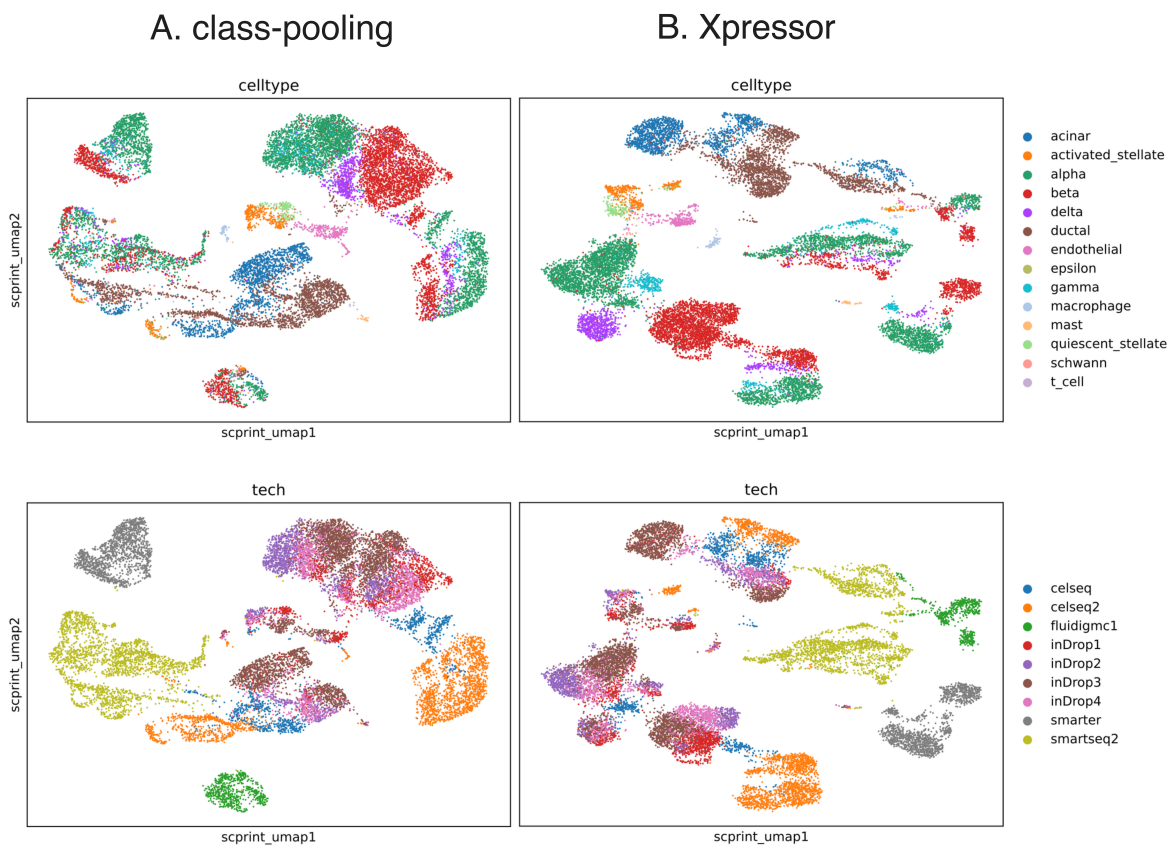


Figure 2.4: Comparison of cell embeddings between the regular transformer with class-pooling (left), scIB: 0.43, and the Xpressor (right), scIB: 0.48. The Xpressor embeddings exhibit greater structure and better resolve different cell types.

Using ESM2's embeddings enables scPRINT to work on genes and sequences unseen at training time, learn from an unlimited number of species, and integrate DNA, RNA, and protein-level information, such as mutations and structural variants.

Finally, unlike other methods, this version does not require updating the original model and can be added to the new model. Moreover, with this approach, scPRINT still maintains the ability to work on genes and sequences unseen during training.

2.4.4 Applications

This multi-scale fine-tuning approach has applications in any context where one wants to compress a foundation model's representation. We have discussed examples in improving foundation model's biological representation of cells and proteins. We have shown how this can be used to create multi-scale model compositions.

In the context of cell foundation models, we included the Xpressor into a larger methodological development: scPRINT-2 [200]. In this same work, we investigate the biological relevance, applicability, and generalization abilities of scPRINT-2.

But this approach can also be used in other contexts. For example, one could use it in language models to compress context windows or in vision transformers to compress image patches of large images for use in other tasks.

Scope and Limitations

While our framework applies in principle to all biological scales (mFM → nFM → cFM → tFM), we focused our empirical validation on the nFM-to-cFM connection (ESM2 to scPRINT). Extending this work to other scale transitions would require substantial effort across multiple dimensions:

Implementation and expertise. Each modality uses distinct architectures, data formats, and training paradigms. Molecular foundation models employ equivariant networks with 3D coordinate inputs; tissue models use vision transformers on multi-channel images. Reimplementing, updating, and retraining these models requires deep domain expertise in each field.

Benchmarking infrastructure. Comprehensive evaluation requires modality-specific benchmarks that capture biologically meaningful tasks. While scPRINT provides a gymnasium of tasks for cFMs, equivalent benchmarks for cross-scale learning at other transitions (e.g., mFM-to-nFM or cFM-to-tFM) do not yet exist and would need to be developed.

Computational resources. Training foundation models at scale demands significant compute. Full retraining of models like ESM2 or tissue foundation models to incorporate cross-scale objectives would multiply this cost.

Importantly, our approach offers two paths forward for other modalities:

1. **Full retraining with multi-scale objectives**: Models like ESM2 could be retrained from scratch, adding Xpressor blocks which will then be fine-tuned during the training of upper-scale models (e.g., cell FMs), enabling end-to-end cross-scale learning.
2. **Post-hoc Xpressor addition**: Xpressor blocks can be added on top of *pretrained* foundation models and trained for reconstruction without ever modifying the base model weights. This lightweight approach preserves the original model's capabilities while enabling compression and cross-scale transfer, and requires only a fraction of the original training compute.

The second approach is particularly promising for rapid adoption, as it allows practitioners to leverage existing pretrained models (ESM2, Nucleotide Transformer, DINO-based tissue models) without fully retraining them, adding only the Xpressor compression layer and fine-tuning on cross-scale objectives.

Conclusion

We proposed a framework for compositional hierarchical foundation models across biological scales. Rather than training end-to-end, we compose models that distill key information for adjacent scales, enabling a shared vocabulary for biological entities across molecules, cells, and tissues.

We have presented one small piece in this approach, in which a cell foundation model (scPRINT) leverages and fine-tunes a protein sequence foundation model (ESM2). We have also shown how Xpressor can compress the output representations of transformers into a small set of lower-dimensional vectors, bridging proteins to cells. Such an approach could be used to bridge molecules to proteins and cells to tissues by using compressed representations that are then fine-tuned. This is a promising backbone architecture for a general model going from atoms to tissues.

Future work should focus on using Xpressor's representations to power upper-scale models or the ability to learn a Xpressor on top of a pre-trained foundation model. The Xpressor approach could also be extended to decoder-based language models. Finally, fine-tuning using an adapter layer suffers from a major drawback: the non-additivity of MLPs, and therefore the limited use of such fine-tuned models in other contexts beyond their compressed representations. Implementing intelligent GPU scheduling and using LoRA-like methods to fine-tune only Xpressor blocks will allow for more complex fine-tuning in GPU-rich settings. We will need to show that this approach can be applied to other scales of biological representation and to generate benchmarks that better capture the diversity of real-world biological tasks across these scales.

Supplementary

2.4.5 Extended review of foundation models across scales

Molecular foundation models (mFM) model molecules with atomistic precision using SMILES notation or 3D coordinates [180, 181, 201]. They incorporate symmetry invariances [78] and can predict binding affinities, solubility, and dynamics, though matching precise molecular dynamics methods remains challenging [202, 203].

Nucleotide foundation models (nFM) analyze DNA, RNA, and protein sequences using transformer architectures [66, 204]. Protein language models like ESM2 [70] enable structure prediction, while DNA/RNA models focus on regulatory mechanisms [182, 205, 183]. These representations encode evolutionary constraints and can inform protein-protein interactions [206]. Future nFMs could use compressed mFM representations as tokens, enabling unified modeling of nucleotides and their modifications [207].

Cell foundation models (cFM) are trained on single-cell RNA-seq abundance matrices [5, 178, 72, 73, 77, 76]. Despite promising benchmarks, experimental validations remain

challenging due to data noise, limited coverage, and species bias [184, 74, 112]. Distilling sequence-level knowledge onto cFMs could improve learning of regulatory mechanisms.

Tissue foundation models (tFM) learn spatial cell relationships from microscopy using vision transformers [185, 186, 208, 209]. Spatial transcriptomics modalities provide rich channel information but at limited resolution [210]. Challenges include data accessibility, 2D limitations, and measuring cell communication [211]. tFMs could use cFM cell representations as tokens to predict cell presence in spatial context.

2.4.6 Detailed scPRINT architecture and training

We provide here additional details on scPRINT's architecture and training procedure, which serve as the foundation for our Xpressor contributions.

Expression encoder

Each gene j in cell i is represented as the sum of three embeddings:

$$\mathrm{x}_{i,j} = \mathrm{g}_j + \mathrm{e}_{i,j} + \mathrm{l}_j \tag{2.5}$$

where $\mathrm{g}_j \in \mathbb{R}^d$ is the gene identity embedding, $\mathrm{e}_{i,j} \in \mathbb{R}^d$ is the expression embedding, and $\mathrm{l}_j \in \mathbb{R}^d$ is the genomic location embedding.

Gene identity embedding. The gene embedding g_j is derived from ESM2 by mean-pooling over all amino acid embeddings of the gene's canonical protein product. This provides evolutionary and structural priors while enabling generalization to unseen genes.

Expression embedding. The expression value is encoded via a two-layer MLP:

$$\mathrm{e}_{i,j} = \mathrm{MLP}(\log_2(x_{i,j} + 1)), \quad x_{i,j} \in \mathbb{R}_{\geq 0} \tag{2.6}$$

where each MLP layer applies: $\mathrm{Dropout}(\mathrm{ReLU}(\mathrm{LayerNorm}(\mathrm{Linear}(\cdot))))$ with dropout rate 0.1. This lets the model learn an appropriate metric for expression values, unlike binning (scGPT) or ranking (Geneformer) approaches that impose specific priors.

Genomic location embedding. Genes within 10kb are assigned the same location index, then encoded using sinusoidal positional encoding [66]. This captures the fact that co-located genes often share regulatory regions.

The full input to the transformer for cell i is:

$$X_i = [\mathrm{x}_{i,1}, \ldots, \mathrm{x}_{i,m}, \mathrm{e}_{t,i}, \mathrm{p}_{\text{cell}}, \mathrm{p}_{\text{type}}, \mathrm{p}_{\text{disease}}, \ldots] \tag{2.7}$$

where $\mathrm{e}_{t,i} = \mathrm{MLP}(\log_2(1 + t_i))$ encodes the total count $t_i = \sum_j x_{i,j}$, and $\mathrm{p}.$ are learned placeholder tokens for cell embeddings and label predictions.

Transformer architecture

scPRINT uses a bidirectional transformer with n layers, h attention heads, and dimension d. Key implementation choices include:

- FlashAttention2 for efficient attention computation
- Pre-normalization with fused LayerNorm
- Stochastic depth with linearly increasing dropout (0.02 per layer)
- 2-layer MLP with 4× hidden dimension and GELU activation

During training, 2,200 genes are sampled per cell (covering >80% of CellxGene cells). When fewer genes are expressed, the input is padded with randomly sampled unexpressed genes, allowing the model to distinguish true zeros from dropouts.

Expression decoder

The decoder outputs parameters of a zero-inflated negative binomial (ZINB) distribution for each gene:

$$\mu_j, \theta_j, \pi_j = \text{MLP}(\text{o}_j) \tag{2.8}$$

where o_j is the transformer output for gene j. The ZINB distribution is defined as:

$$\text{ZINB}(x \mid \mu, \theta, \pi) = \pi \cdot \delta_0(x) + (1 - \pi) \cdot \text{NB}(x \mid \mu, \theta) \tag{2.9}$$

where $\delta_0(x)$ is a point mass at zero, and the negative binomial is:

$$\text{NB}(x \mid \mu, \theta) = \frac{\Gamma(x + \theta)}{x!\,\Gamma(\theta)} \left(\frac{\mu}{\mu + \theta}\right)^x \left(\frac{\theta}{\mu + \theta}\right)^\theta \tag{2.10}$$

Here μ is the mean and θ is the inverse dispersion. Unlike scVI, which learns a fixed θ per gene, scPRINT predicts it dynamically, allowing the model to capture context-dependent overdispersion.

Pretraining objectives

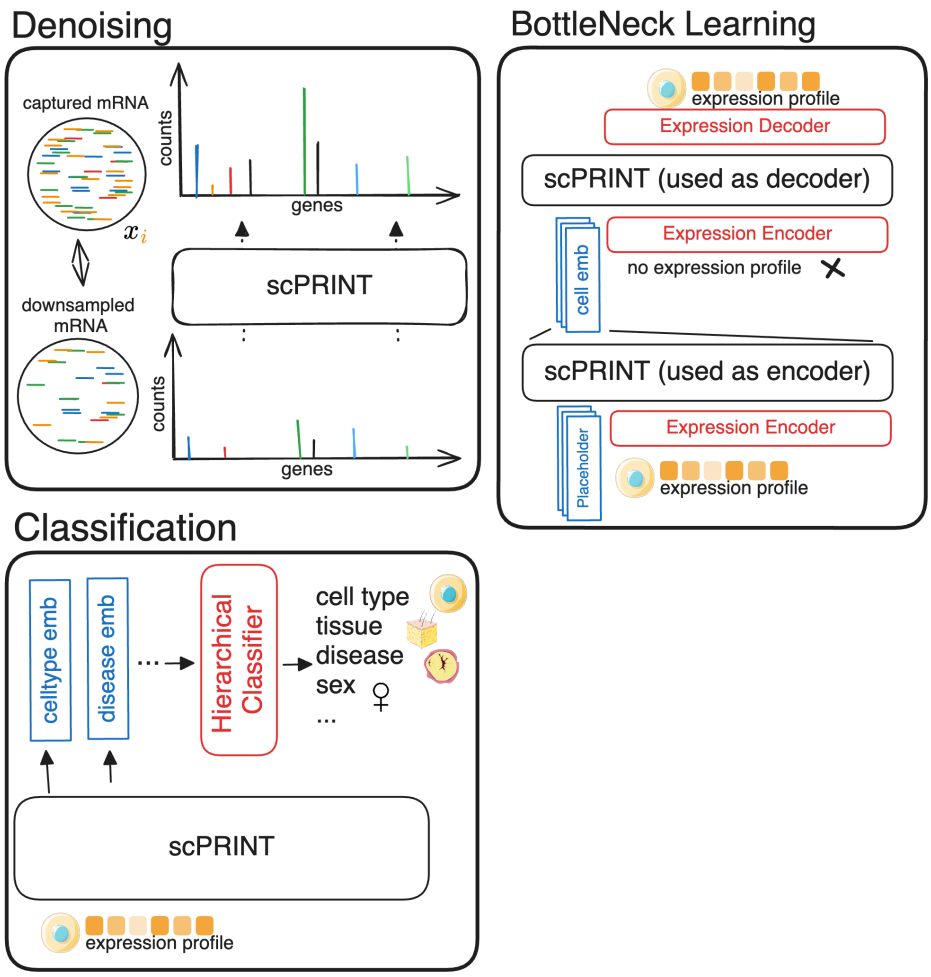


Figure 2.5: Overview of scPRINT's pretraining tasks. Three pretraining objectives are jointly optimized: denoising (reconstructing from downsampled profiles), bottleneck learning (compression through cell embeddings), and label prediction (hierarchical classification of cell annotations). Inspired by the original Figure 1.B in Kalfon et al. [178].

The total loss is: $\mathcal{L} = \mathcal{L}_{\text{denoise}} + \mathcal{L}_{\text{bottleneck}} + \mathcal{L}_{\text{classify}}$

Denoising loss. Expression profiles are downsampled using a zero-inflated Poisson model with dropout rate $r = 0.6$:

$$\tilde{x}_{i,j} \sim \text{ZiPoisson}(x_{i,j}, r) \tag{2.11}$$

The model reconstructs the original counts, with loss given by the negative log-likelihood under the ZINB decoder (see Figure 2.5).

Bottleneck loss. The model must reconstruct expression from cell embeddings alone (without gene expression inputs), encouraging compression of cellular state into the placeholder tokens (see Figure 2.5).

Classification loss. For hierarchical labels (cell type, disease, assay), we use a modified cross-entropy that handles ontology structure. For non-leaf labels, we apply log-sum-exp over descendant leaf labels:

$$\mathcal{L}_{\text{cls}} = \text{CE}(\bar{\text{c}}, \text{c}), \quad \bar{c}_k = \begin{cases} \hat{c}_k & \text{if } k \in \mathcal{T} \\ \text{LSE}(\{\hat{c}_l : l \in \text{desc}(k)\}) & \text{otherwise} \end{cases} \tag{2.12}$$

where $\mathcal{T}$ is the set of leaf labels and $\text{desc}(k)$ are descendants of label k (see Figure 2.5).

Optimization

Training uses fused AdamW with weight decay 0.01, learning rate 10^{-4}, and stochastic weight averaging (SWA) with LR 0.03. Sub-epochs consist of 7,000 training and 2,000 validation batches with a 500-step warmup. Learning rate decays by 0.6× with patience 1, and training stops after 3 consecutive validation loss increases. Weighted random sampling with factor 50 balances rare cell types: sampling weight $\propto \frac{50}{\text{count}+50}$.

Gene network extraction

Cell-specific gene networks are extracted from attention matrices, following ESM2's approach for contact prediction. For a cell i, the gene-gene attention matrix $A_i \in \mathbb{R}^{m \times m}$ captures learned interactions. These can be thresholded to obtain sparse networks or subset to transcription factor-gene edges for gene regulatory network (GRN) inference. Attention heads can be selected based on correlation with ground-truth networks to focus on biologically meaningful interactions.

More details on scPRINT's pre-training and architecture, datasets preprocessing, and gene network extraction are available in Kalfon et al. [178].

2.4.7 argument about the Tishby et al. bottleneck learning approach

We present the Information Bottleneck (IB) method, a technique that defines compression as a learning approach. Using the information theory framework, it seeks a stochastic mapping $p(t|x)$ that compresses the input variable X into a representation T, while preserving as much information as possible about the relevant variable Y. The trade-off is controlled by the Lagrange multiplier $\beta \geq 0$. The IB objective is to minimize the following Lagrangian:

$$\mathcal{L}_{\text{IB}}[p(t|x)] = I(X;T) - \beta\, I(T;Y), \tag{2.13}$$

where $I(\cdot;\cdot)$ denotes mutual information.

Under the Markov constraint $\text{Y} \leftrightarrow \text{X} \leftrightarrow \text{T}$, The optimization leads to the following

self-consistent equations:

$$p(t|x) = \frac{p(t)}{Z(x,\beta)} \exp\left(-\beta\, D_{\mathrm{KL}}\big(p(y|x) \,\|\, p(y|t)\big)\right), \tag{2.14}$$

$$p(t) = \sum_x p(x)\, p(t|x), \tag{2.15}$$

$$p(y|t) = \frac{1}{p(t)} \sum_x p(y|x)\, p(x)\, p(t|x), \tag{2.16}$$

where:

- $D_{\mathrm{KL}}\big(p(y|x) \,\|\, p(y|t)\big)$ is the Kullback-Leibler divergence between the conditional distributions $p(y|x)$ and $p(y|t)$,
- $Z(x,\beta)$ is the normalization factor ensuring that $\sum_t p(t|x) = 1$.

2.4.8 FSQ and other contrastive losses on the cell embeddings

While D_{KL} over a non-informative Gaussian prior is a common formulation for regularizing the embedding space in VAEs, other formulations have been used, such as with the VQ-VAE and Finite Scalar Quantization Variational Autoencoder (FSQ-VAE). In these contexts, the D_{KL} is replaced with a discretization objective tailored to the respective quantization schemes.

VQ-VAE. Value Quantized (VQ)-VAE employs a *codebook* of size C, where each codebook entry is a d-dimensional vector. The encoder produces a continuous latent vector, which is then mapped to its nearest codebook entry (a hard quantization). A commitment loss term encourages the encoder's outputs to stay close to the chosen codebook vector, making the entire latent representation discrete at the vector level.

FSQ-VAE. By contrast, Finite Scalar Quantization (FSQ)-VAE discretizes each latent dimension *independently*. Specifically, the encoder outputs d values, each constrained to lie within a bounded range (e.g., $[-1, 1]$). Each dimension is then quantized into one of M discrete levels within that range. This dimension-wise quantization can be implemented as either a hard nearest-bin assignment or a differentiable approximation thereof. Because FSQ enforces scalar-level discretization, it provides a simpler and more fine-grained alternative to VQ's vector-level codebook approach, while still offering strong regularization of the latent space.

Contrastive regularization across embedding dimensions. We further encourage each of the d embedding dimensions to encode distinct information by adding a contrastive loss between them. Specifically, we compute pairwise similarities among embedding elements

and penalize redundancy, thus pushing each dimension to capture complementary features. A general contrastive loss for this purpose can be written as

$$\mathcal{L}_{\text{contrastive}} = \sum_{i=1}^{d} \sum_{j \neq i} \ell\big(\mathrm{e}_i, \mathrm{e}_j\big), \tag{2.17}$$

where e_i denotes the i-th embedding dimension and ℓ is a contrastive loss function (e.g., InfoNCE [212]) that encourages *dissimilarity* among different embedding components.

Dimension-specific classifiers. To further steer each dimension's content, one can add a separate classifier on top of each dimension to learn about different classes. The classifier for dimension i is trained via a cross-entropy loss

$$\mathcal{L}_{\text{cls}}^{(i)} = -\sum_{c} y_c \log p\big(c \mid \mathrm{e}_i\big), \tag{2.18}$$

where y_c is the ground-truth label and $p\big(c \mid \mathrm{e}_i\big)$ is the predicted probability for class c. Summing these per-dimension losses yields an overall classification objective

$$\mathcal{L}_{\text{cls}} = \sum_{i=1}^{d} \mathcal{L}_{\text{cls}}^{(i)}. \tag{2.19}$$

Together, the contrastive and classification losses ensure each embedding dimension captures unique, discriminative information, resulting in more expressive representations.

Software and Data

The software and data for training scPRINT, as well as gymnasium tasks and code to reproduce the results of the manuscript, are available at https://github.com/cantinilab/Xpressor.

WandB logs are available in the following link: https://api.wandb.ai/links/ml4ig/h370j6io

Model checkpoints are available at the following link: https://huggingface.co/jkobject/scPRINT/tree/main

Acknowledgments

The project leading to this manuscript has received funding from the Inception program (Investissement d'Avenir grant ANR-16-CONV-0005) L.C. and the European Union (ERC StG, MULTIview-CELL, 101115618) L.C. We acknowledge the help of the HPC Core Facility of the Institut Pasteur and Déborah Philipps for the administrative support. L.C.

The work of G. Peyré was supported by the French government under management of Agence Nationale de la Recherche as part of the 'Investissements d'avenir' program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute). G.P.

Impact Statement

This paper presents work aimed at advancing the fields of computational biology and machine learning. No ethical issues are raised by the work other than what is typically noted in computational biology and foundation model papers. It might have an impact on building better models for drug discovery, target discovery, and improving our understanding of biological systems.

scPRINT-2: Towards the next-generation of cell foundation models and benchmarks

3.1 Summary

Cell biology has been booming with foundation models trained on large single-cell RNA-seq databases, but benchmarks and capabilities remain unclear. We propose an additive benchmark across a gymnasium of tasks to discover which features improve performance. From these findings, we present scPRINT-2, a single-cell Foundation Model pretrained across 350 million cells and 16 organisms. Our contributions in pretraining tasks, tokenization, and losses made scPRINT-2 state-of-the-art in expression denoising, cell embedding, and cell type prediction. Furthermore, with our cell-level architecture, scPRINT-2 becomes generative, as demonstrated by our expression imputation and counterfactual reasoning results. Finally, thanks to our pretraining database, we uncover generalization to unseen modalities and organisms. These studies, together with improved abilities in gene embeddings and gene network inference, place scPRINT-2 as a next-generation cell foundation model.

3.2 Introduction

For the last few years, Single-Cell Foundation Models (scFMs), also known as Virtual Cell models, have provided early approaches to modeling the cell using single-cell RNA-seq data as their primary modality[73, 72, 71, 76]. The field has been booming with these transformer-based machine learning models trained on large databases of tens of millions of cells. The models themselves contain tens to hundreds of millions of parameters and are trained on unsupervised (or semi-supervised) tasks such as predicting masked gene expression or denoising expression. They can then be used as is to examine their learned

representations or fine-tuned to transfer their knowledge across a range of everyday tasks in that modality. Many examples have now been proposed, such as predicting single-cell perturbation responses, patient drug responses, and disease states; annotating cells; correcting for batch effects; improving noise levels; imputing unseen gene expression or modality; generating gene networks; identifying cell niches; and more[213, 214, 71, 215, 216, 217, 218, 219, 220, 221, 222].

While many AI Virtual Cell models and scFMs exist, little has been done regarding their comparison[223, 224, 225, 226, 227]. A crucial question remains: how to validate the impact of the different proposed methods, regardless of implementation, datasets, or model size. Indeed, reproducing results has been challenging for many, and the literature has yielded discordant conclusions about the performance and capabilities of these models. Showing they often underperform simpler approaches on classification, batch correction, and perturbation prediction[223, 224, 68, 184, 228]. Much work remains to get to feature-rich, easy-to-use scFMs. Models that allow inference in minutes, along with well-crafted reproducible benchmarks that demonstrate how scFMs uniquely solve essential problems in single-cell biology. Open-sourcing not just model weights but their pretraining tasks and datasets.

On this front, scPRINT was released as part of a second batch of scFM, presenting contributions in terms of usability and reproducibility while also showcasing pretraining strategies, data encoding, and decoding[178]. scPRINT was trained on 50 million cells using a multitask pretraining strategy that included expression denoising, autoencoding, and cell-label prediction. It also presented an in-depth benchmark that examined the foundation model's zero-shot performance on these tasks, as well as its internal gene network representation and fidelity compared to multiple ground truths.

Building on these strengths and moving towards the next generation of scFMs, we here use scPRINT (which will be referred to as scPRINT-1) as the reference to showcase an extensive additive benchmark of scFM attributes. We address several key questions about the importance of diverse architectures, datasets, and training modalities. This additive benchmark aims to understand the relative importance of these different features in our task gymnasium, examining the choice of model architecture and pretraining tasks across 42 different scenarios. In these scenarios, we propose a breadth of novel components for scFMs. In addition to those 12 distinct contributions, we also examine various pretraining datasets, compiling a 350-million-cell database—the largest to date—with over 16 organisms.

As a result of the benchmark, we derive a next-generation scFM, extbfscPRINT-2. scPRINT-2 improves upon the previous generation of models by leveraging our database, the scPRINT-2 corpus, and multiple data augmentation approaches. It uses a set of updated pretraining tasks and losses, improving its accuracy in challenging and unseen contexts. Finally, it is equipped with graph-based encoders and the XPressor architecture, enabling unprecedented expression imputation, high-quality zero-shot embeddings, and counterfactual reasoning. We dive into these specific contributions by examining multiple use cases, highlighting behaviors that are often overlooked or under-assessed in classical benchmarks.

scPRINT-2, its dataloader, pretraining datasets, preprocessing, task functions, pretrained

weights, as well as the additive benchmark training traces and all 42 models' weights are fully open-sourced and available under the GPL-v3 License.

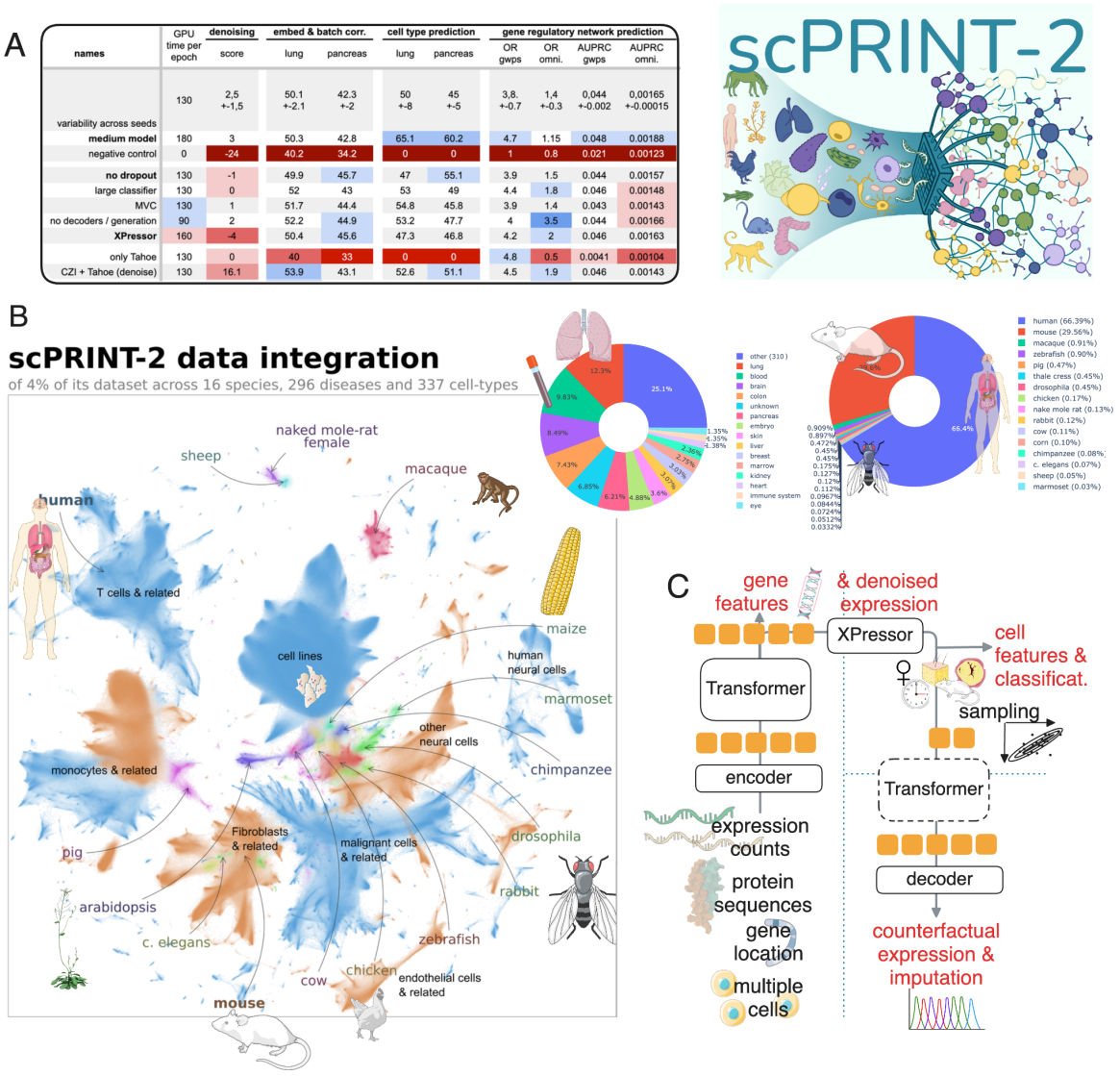
3.3 Results

3.3.1 Decoding the impact of a foundation model's architecture through an additive benchmark

Many scFMs have been developed in single-cell genomics. They have mostly been studied in isolation, using their own benchmarks. While most of them maintained relatively similar architectures, the impact of each design's decisions was never thoroughly assessed. For example, scPRINT-1 uses a denoising reconstruction task similar to scFoundation. Still, scFoundation uses the mean-squared-error (**MSE**) for the reconstruction loss, whereas scPRINT-1 uses the zero-inflated negative-binomial loss (**ZINB**) (see Methods). scGPT and Geneformer utilize masking, but scGPT bins expression counts (**binning**), while scPRINT-1 does denoising and employs a continuous embedding with a log transform and a pseudocount of 1 (**logp1**)[73, 72]. Other models, like cellPLM, instead use a contrastive learning approach, which encourages embeddings of perturbed and unperturbed cell profiles to be more similar to each other than those of different cell profiles. This method is also known as InfoNCE or Contrastive Cell Embedding (**CCE**) (see Methods)[212].

Additive benchmark

To address the lack of a consistent assessment of these models, we have designed a benchmark to comprehensively evaluate the various components of scFMs, including pretraining databases, architectures, and training tasks. This benchmark is based on a gymnasium of tasks similar to those presented in Kalfon et al.[178] (see Figure 3.1; see Table 3.1). The scFM gymnasium assesses each model's ability to predict labels, remove batch effects, denoise, and impute gene expression, as well as discover known gene-gene relationships at different stages of training. For embeddings and cell type classification, we use the scIB and accuracy scores over the same ground-truth test datasets as in Kalfon et al. (see Methods). For denoising, we evaluate the model's ability to recover the noised expression profile of cells from a test dataset, as measured by the improvement in correlation with the ground-truth profile after denoising. For gene-network inference, we examine the Odds Ratio (OR) and AUPRC scores of the model's ability to recover a ground-truth gene network from expression data alone (see Methods).



names	GPU time per epoch	denoising score	embed & batch corr. lung	embed & batch corr. pancreas	cell type prediction lung	cell type prediction pancreas	gene regulatory network prediction OR gwps	OR omni.	AUPRC gwps	AUPRC omni.
variability across seeds	130	2,5 +-1,5	50.1 +-2.1	42.3 +-2	50 +-8	45 +-5	3,8. +-0.7	1,4 +-0.3	0,044 +-0.002	0,00165 +-0.00015
medium model	180	3	50.3	42.8	65.1	60.2	4.7	1.15	0.048	0.00188
negative control	0	-24	40.2	34.2	0	0	1	0.8	0.021	0.00123
no dropout	130	-1	49.9	45.7	47	55.1	3.9	1.5	0.044	0.00157
large classifier	130	0	52	43	53	49	4.4	1.8	0.046	0.00148
MVC	130	1	51.7	44.4	54.8	45.8	3.9	1.4	0.043	0.00143
no decoders / generation	90	2	52.2	44.9	53.2	47.7	4	3.5	0.044	0.00166
XPressor	160	-4	50.4	45.6	47.3	46.8	4.2	2	0.046	0.00163
only Tahoe	130	0	40	33	0	0	4.8	0.5	0.0041	0.00104
CZI + Tahoe (denoise)	130	16.1	53.9	43.1	52.6	51.1	4.5	1.9	0.046	0.00143

Figure 3.1: Presentation of the scPRINT-2 model, pretraining dataset, and additive benchmark. (a) The additive benchmark example table with its gymnasium scores across the scFM's features. (b) Our scPRINT-2 corpus pretraining dataset, with 16 organisms across 300+ tissues. UMAP of 15 million cells from the corpus integrated using scPRINT-2. Colors represent species. (c) The scPRINT-2 model, its input data, and its different outputs. Source data are provided as a Source Data file.

The base model, on which the additive benchmark is performed (see Figure 3.1, Table 3.1, and Methods), is trained on the CxG database, comprising 500 carefully annotated human and mouse datasets. Its training lasts for a maximum of 20 epochs, each of 20,000 steps, with a minibatch size of 64. We encode the gene expression using the scPRINT-1 approach and decode it with the MSE method. The base model's pretraining task uses a 30% gene expression mask. We pretrain the models 6 times across multiple seeds to generate error bounds. Using Flash-Attention-3, the 20M parameters model trains on 1 H100 GPU for 2 days.

While we will not delve into the details of each feature assessed (see Methods), our benchmark broadly highlights several key points.

Regarding the tasks, we have confirmed what Kalfon et al. and De Waele et al. previously showed: that denoising is superior to masking as a pretraining task for single-cell data in classification and embedding tasks[178, 229]. Similarly, un-normalized expression is better than normalizing it at the input. Classification also serves as a good supplement to the pretraining task, as without it, we observe a slight decrease in performance (see Table 3.1).

We also present, as part of our study, the **scPRINT-2 corpus**, which comprises more than 350 million single cells (see Figure 3.1). This is the largest dataset ever assembled, consisting of data from the Chan Zuckerberg Institute's Cellxgene (**CxG**), the **Tahoe**-100M dataset, and the scBasecount database, which contains 20,000 reprocessed datasets from the Gene Expression Omnibus[112, 230, 231]. The cells themselves are derived from 16 different eukaryotic organisms, spanning more than 1 billion years of evolution. The dataset comprises approximately 400,000 distinct genes, 4,764 different labels, and around 140,000 cell groups, totaling 25 TB of unique data[115]. Our database contains nine main classes: *cell type, disease, age, tissue of origin, assay, ethnicity, sex, cell culture,* and *organism.*

Thanks to this database, we demonstrated the growing importance of data selection in pretraining scFMs. Indeed, when using the Tahoe-100M database solely for pretraining, the model's overall performance plummets, as the sequencing depth and diversity are low despite the large number of cells.

However, including this lower-diversity dataset with the high-diversity CxG database and carefully considering the cell-state imbalances results in only a noticeable decrease in denoising performance. Interestingly, using all available datasets did not change performance across our benchmarks. Reducing the training database to a random subset of only **200 human datasets only,** led to a minimal decrease in denoising and cell type prediction. This shows again that the benchmark fails to highlight abilities on more diverse cell types and organisms[232, 233]. But it also indicates diminishing returns in adding more datasets—diversity in cell states and organisms being much more important than cell count.

We thus preprocessed each dataset by removing all duplicates, filtering for low-quality cells, aligning metadata to the CxG ontologies, and computing cell-cell similarity profiles and clusters. It allowed us to introduce multiple data augmentation techniques, such as varying the input context length (**var. context**) during training and randomly creating **meta-cells**, which are averages of similar cell expression profiles across K-nearest neighbors (K-NN) (see Results section 3). Interestingly, we observe that both methods tend to improve the model's performance in most metrics, even though these models do not examine more cells overall. This highlights the importance of effective data augmentation techniques for scFM pretraining[234].

	names	GPU time per epoch	denoising score	embed & batch corr. lung	embed & batch corr. pancreas	cell type prediction lung	cell type prediction pancreas	gene regulatory network prediction OR gwps	gene regulatory network prediction OR omni.	gene regulatory network prediction AUPRC gwps	gene regulatory network prediction AUPRC omni.	run id
Base	ross seeds (masking; ZINB loss; ² + continous expr. emb.; classif. + generative task)	130	2,5 +- 1,5	50.1 +-2.1	42.3 +-2	50 +- 8	45 +- 5	3,8. +-0.7	1,4 +- 0.3	0,044 +-0.002	0,00165 +-0.00015	blooming-dew-714, driven-valley-750, summer-deluge-783, silent-poltergeist-843, unraveling-pumpkin-841
	medium model	180				65.1	60.2	4.7		0.048	0.00188	solar-durian-637
	negative control		-24	40.2	34.2	0	0	1	0.8	0.021	0.00123	
architecture	**no dropout**		-1		45.7		55.1					expert-feather-748
	large classifier		0						1.8		0.00148	chocolate-snowball-718
	MVC	130							1.4		0.00143	celestial-sun-749
	no decoders / generation	90			44.9				3.5			northern-voice-777
	XPressor	160	-4		45.6				2			crimson-wildflower-791
data	CxG (CellxGene's Census)			52.3					2			faithful-dragon-663
	CxG + Tahoe (denoising)		16.1	53.9			51.1		1.9			lurking-cat-846
	only Tahoe		0	40	33	0	0	4.8	0.5	0.0041	0.00104	young-bush-669
	all databases (denoise)	140			39							macabre-apparition-844
	200 human datasets only		0			40.3			1.8			playful-frost-804
	sampling without replacement		0		45	36	34		2.4			autumn-aardvark-702
	cluster-based sampling only			43.3								rosy-firefly-805
attention	softpick								1.8			eldritch-fang-834
	criss-cross (larger context)	80	5.6						x	x	x	uncanny-raven-835
	hyper (denoise, larger context)								0.6	0.04	0.00115	silver-grass-803
loss	contrastive learning (masking + denoising)					39.5					0.00149	northern-frog-797
	elastic cell similarity			52.7			34.9					hopeful-monkey-796
	no embedding ind loss								2.3			devoted-wave-795
	ZINB+MSE (denoising)		25.5		48					0.04		firm-silence-747
	MSE		-4	54	46	62						generous-dawn-666
	VAE compressor	140				38	27					wild-terrain-694
pretraining task	**var. context (larger context)**	170	29.1	53	46		52.2			0.038	0.00146	apricot-snowflake-756
	TF masking								2.3			winter-meadow-772
	denoising		21	52.6	45.1		54.5					efficient-firebrand-753
	no classification				40	0	0				0.00129	copper-frost-625
	adv. classifier (+larger classif)										0.0014	sunny-morning-629
input	sum normalization (denoise)		12.8	45.6	46.5	21.4	22.9	2.4	1	0.029	0.00136	snowy-galaxy-744
	no random level of denoising		19	54.1	45.3				2	0.041		divine-monkey-798
	GNN	150	44	48		38	35				0.00128	balmy-totem-727
	meta-cell			52.8	47.7		51.3			0.04		unique-dawn-806
	binning		0		45.5		52			0.047		twilight-breeze-874
	using only expressed genes											sage-snow-873
	without gene location		3.4	36.2	35	4	5.9	4.8		0.048		youthful-snowflake-792
	learn gene emb (denoising)				45.1					0.041		jumping-night-755
	fine-tuned ESM3										0.00181	not-snowflake-755
Main	small model (V2)	1820	44	53	49					0.041		honest-vortex-815
	medium model (V2)	5600	x	x	x	x	x	x	x	x	x	x
	medium model (V1)	520		52.6	45.6	61.8	57.6		2.2	0.041		bewitched-poltergeist-857
	small model (V1)	160	31.7	52.4	50						0.00138	dry-smoke-852

Table 3.1: Table representing the results of the additive benchmark on 42 models, over multiple metrics: batch correction and cell embedding quality, denoising quality, cell type prediction, and gene network inference. Additional information on the different components is available in the methods section. Bold elements are the features that are part of the scPRINT-2 foundation model.

Regarding architecture, we recomputed results from the XPressor manuscript[235], which showed that this architecture improves the embedding quality of scFMs (see Results section 4; see the full table in Supplementary Table 5.13). We also demonstrate that using ESM-based gene ID tokens leads to much better performance than learning gene tokens from scratch[236]. Providing each gene's genomic location as additional input information significantly improves model convergence. However, we also noticed that when they do converge, models without gene location information can perform well. We have noticed that model size correlates with higher scores, at least for gene network inference and cell-type prediction. Using a Graph Neural Network (**GNN**) encoder shows significant improvements, with only a slight decrease in the cell-type prediction task (see Results Section 3; see Methods). Additionally, our sub-quadratic attention mechanism, Criss-cross

attention, also shows substantial benefits with no reduction in performance (see Results section 4; see Methods).

Moreover, MSE, on average, outperforms ZINB as a loss function while decreasing the model's expressivity (see Methods). A good proposed middle ground is the ZINB+MSE loss (see Results Section 3; see Methods).

Some unexpected results showed that omitting the decoder part of scPRINT-1 led to stronger performance; however, this comes at the cost of generative abilities and decreased cell-embedding fidelity. Indeed, despite its importance for understanding scFMs' behavior and feature importance, we have noted that our benchmark does not yet capture the full breadth of abilities that scFMs do or should have. For example, both scIB and classification scores are very dependent on the dataset' s quality and its labels. Scores presented here show only a facet of the model's ability. We might be interested in the model's performance up-to-convergence instead of stopping them at 20 epochs or looking at unseen species, or assays at training. This is a first attempt to benchmark scFMs, but more extensive efforts will be needed.

scPRINT-2

Overall, we have examined the performance improvements driven by our 12 distinct contributions across 42 training runs. Based on these results and our own considerations, we have elected a set of features to create scPRINT-2, a next-generation cell foundation model (see Supplementary Figure 5.28; see Methods). We highlight its architecture in Figure 3.1; scPRINT-2 is currently available in a small version with only 20M active parameters. Its encoder-compressor-decoder architecture produces cell- and gene-level outputs at multiple levels, working on one or more cells at a time.

Furthermore, to aid in the exploration of this largest-ever cross-organism single-cell dataset, we release all of the 350 million cells in the scPRINT-2 corpus, aligned into an atlas by scPRINT-2, of which 1% are directly accessible through an interactive visualization (see Figure 3.1, see Data availability) along with scPRINT-2 cell label predictions for all classes. This should enable never-before analysis and exploration of single-cell RNA-seq data.

But the additive benchmark leaves some questions unanswered about the effect of combining these features up-to-convergence and the models' abilities on unseen modalities, tasks, and species. In the following sections, we will focus on 1. looking at more diverse and truthful datasets in size, quality, and source domains; 2. using more scores and ground truth validations; 3. defining tasks that better reflect the possibilities and real-life use of these models.

3.3.2 A diverse dataset of 350 million cells pushes generalization to unseen organisms

One of the most critical features of foundation models (FMs) is the breadth of their training dataset. From vision to language, AI advancement has been driven by training models on ever-larger datasets[183, 237, 238, 185, 239]. Nowadays, most scFMs are trained on 20 to 50 million cells, except the recently released Geneformer-v2 and STATE-SE models, which have been trained on roughly 300 million cells[125, 240].

scPRINT-2 pretraining corpus

In conjunction with our model's architecture, the scPRINT-2 corpus and its 16 organisms enable generalization to organisms unseen during training. This broader cell type diversity, however, comes with additional challenges: annotation quality has decreased due to missing annotations in scBasecount. Additionally, the skew toward low sequencing depth and highly similar cells has increased with the inclusion of spatial transcriptomics datasets and less curated databases such as Tahoe-100M and Arc's scBasecount (see Methods).

Fortunately, a key feature of our dataloader, scDataLoader[178], is its ability to perform weighted random sampling, thereby mitigating the heavy dataset imbalances that currently exist across diverse cell types, sequencing methodologies, and different organisms assessed. We thus present methods to successfully train scPRINT-2 on this large dataset. The first, called cluster-weighted sampling, allows datasets with unclear annotations to benefit from weighted random sampling by defining clusters of high expression similarity (see Methods). This lets us define cell states without requiring any label information and perform sampling that is aware of the different cell states, regardless of the size of each cluster. We address the second issue of uneven cell quality by also skewing sampling toward cells with more non-zero genes (NNZ). Both methods were enabled on such a vast database thanks to essential updates to scDataLoader. This re-weighting is performed jointly with weights on cell type, disease, organism, and sequencer labels, thereby addressing the size/diversity issues that plague these larger cell databases[75].

Interestingly, the number of training steps required to achieve convergence increased only 2-fold, indicating that, as in scPRINT-1, the model did not sample as many cells as actually exist in the pretraining dataset before reaching convergence. However, with data augmentation and nearest-neighbor sampling, the model still encountered roughly 2 billion distinct input cell profiles during pretraining, corresponding to 2000 cell profiles per step.

After implementing this feature and training scPRINT-2, its cell-type classification performance on the validation dataset was 76%. For its other predicted labels, its performance was 59% (disease), 96% (ethnicity), 96% (assay), 94% (age), 100% (cell culture), 100% (organism), 93% (sex), and 70% (tissue of origin).

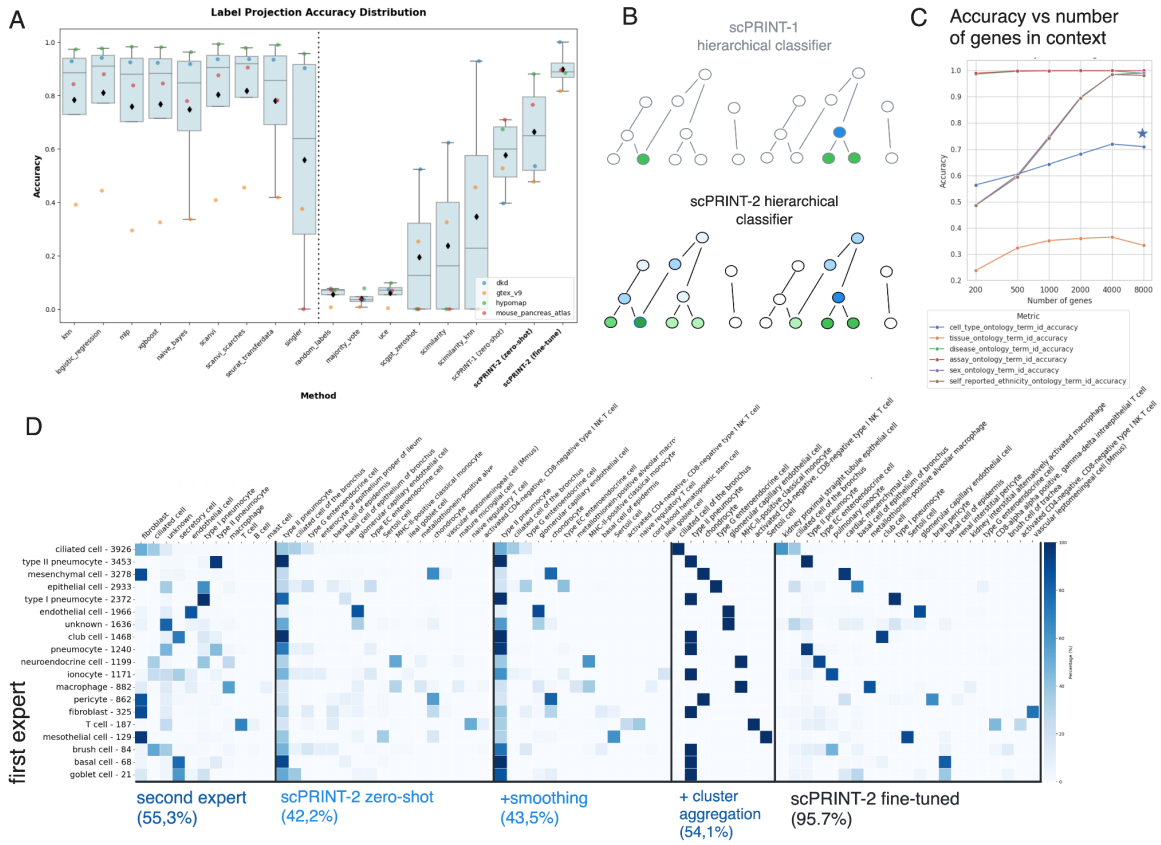


Figure 3.2: Presentation of the updated classifier and results on classification tasks. (a) Open Problems benchmark results and comparison of scPRINT-1 and zero-shot and fine-tuned scPRINT-2. (b) Illustration of our updated hierarchical classifier loss. (c) Unseen organisms cell type classification for cat and tiger datasets, across two experts and scPRINT-2 zero-shot, after label smoothing, after cluster aggregation, and after fine-tuning. (d) Change in classification accuracy as the number of genes in context increases for high-quality single-cell datasets. The star represents the model's score when label smoothing is used.

On the live benchmark Open Problem from November 2025, it achieved an average zero-shot performance of 75%, putting it above scPRINT-1 (47%) and other zero-shot FMs (40-60%), even above Liger, a supervised technique[241] (see Figure 3.2; see Methods). But scPRINT-2 was the only scFM with UCE that could run on all datasets[76]. Against the two human datasets on which Scimilarity-KNN could be run, it performed slightly better than scPRINT-2. This is most likely due to the smaller capacity of scPRINT-2 (20M parameters) compared to scimilarity (100M parameters), as we also observed in our additive benchmark (see Table 3.1). Another likely reason is that the model likely saw those datasets more often during pretraining, since it is trained only on CxG's human datasets.

We then performed fine-tuning using our XPressor-based Parameter-Efficient Fine-tuning (XPEFT), in which we fine-tune only the XPressor layers of scPRINT-2 (see Methods). In this context, we show that scPRINT-2 fine-tuned outperforms every existing supervised and unsupervised method on the Open-problem (see Results section 4; see Methods)[242]. We observed similar trends in the macro-F1 scores (see Supplementary Figure 5.29). Of note, neither scGPT nor Geneformer are currently tested in their fine-tuned version on the platform.

These performances are enabled in part by our update to scPRINT-1's hierarchical classification loss (see Figure 3.2). The scPRINT-1 classifier generates predictions for all possible labels in a hierarchical ontology, while producing logits only for the leaf labels. To predict the other labels, it only has to aggregate their leaf logits. In scPRINT-2, we improve on this loss by using the entire ontological graph, meaning that, e.g., given a ground truth of *olfactory neuron,* we will penalize a prediction of *inhibitory neuron* less overall than a non-neuron label, like *fibroblast.* In conjunction with our weighted sampler, this allows the model to learn rich gradients from a low volume of data.

scPRINT-2 generalizes to unseen classification tasks

We have, however, noticed that classification performance does not generalize sufficiently to correctly recover the exact phylogenetic relationships within organisms or, similarly, within ethnicities (see Supplementary Figures 5.30, 5.31, 5.32). This could be biased heavily by tissue representation in rare ethnicities and organisms. However, some relationships were found, such as *Singaporean Indian/Singaporean Chinese, Korean/Japanese/Chinese, American/Latin American,* or *Macaque/Marmoset/Chimpanzee, Drosophila/C. elegans, Human/Mouse, Pig/Cow*, suggesting that with greater diversity and representation, scFMs might learn this relationship classification of gene expression on their own.

We show that this does not prevent scPRINT-2 from generalizing to unseen organisms. Using a randomly selected tomato plant dataset and its corresponding ESM3 gene embeddings, unseen at training time, scPRINT-2 generates an organism label prediction for the two plant organisms it knows about 67% of the time. This is despite the very low prevalence of these organisms in the pretraining dataset (see Figure 3.1). For a horse dataset, scPRINT-2 predicted mammalian organisms 72% of the time.

Unfortunately, these datasets lacked cell-type annotations. Using well-annotated datasets from Zhong et al.[243] of cat and tiger lung tissues, organisms not seen at training time, we generate cell type predictions using scPRINT-2 and achieved a prediction accuracy of 42.2% across the 500 potential cell type leaf labels scPRINT-2 knows about. While this score may seem low compared to supervised approaches, it is worth noting that labels from a secondary source were available in the datasets. Comparing them to the initial ground truth, we found only a 55.3% agreement between the two. Furthermore, we noticed that for some cells, annotations were quite different, such as: *fibroblast* being labelled as *ciliated cell*, *macrophage* as *neuroendocrine cell, and ionocyte* as *secretory cell.*

Given the low correspondence between the two expert annotations, we wanted to determine which was correct between scPRINT-2 zero-shot or the expert ground-truth labels. We conducted a differential expression analysis between cells labeled as *type 2 pneumocyte* by scPRINT-2 (zero-shot) but as *macrophage* by the ground truth (see Supplementary Figure 5.33). We saw that the most highly differentially expressed genes were *MAGI1*, *NPNT*, *TEAD1*, and *LMO7*, which are involved in cell-cell junctions, epithelial cells, alveolar cells, and lung tissues. Moreover, the first differentially expressed gene was *SFTPC*, a known "type 2 pneumocyte" marker. This means that, even in this challenging unseen-organism dataset, scPRINT-2 seems to legitimately correct expert annotations. This showcases strong generalization to unseen organisms.

To further improve scPRINT-2's accuracy, we use a method first presented in Hu et al. to aggregate predictions based on **nearest neighbor smoothing** of the model's class logits (see Methods)[244, 245]. This approach increased accuracy in most of our use cases but yielded a small 1.3% improvement here. We also provide tools to perform **top-K predictions** and **confidence-based selection**. This means that scPRINT-2 can list multiple putative labels for each cell. When multiple labels have high logits, it can output their shared parental label for that cell instead. When labels disagree, or the logits are low, scPRINT-2 can output an "unknown" label instead. Using both approaches together, we get an additional 3% improvement in accuracy, with 10% of the cells now listed as "unknown".

Additionally, the low accuracy is also related to scPRINT-2 predictions being cell-specific, whereas most ground truth labels are cluster-specific. We propose a **cluster-based logits averaging,** which can be viewed as an extreme case of smoothing (see Methods). With this tool, scPRINT-2 performance increased by 12% (see Figure 3.2). Beyond improved accuracy, these inference-time contributions significantly enhance the usefulness of scFM-based cell annotation for biologists.

Finally, we also demonstrate that with our XPEFT method (presented further in Results section 4), scPRINT-2 can improve its predictions to 95% accuracy in the test subset, while preserving some fine-grained cell-type distinctions not present in the training data (see Figure 3.2).

We then assessed scPRINT-2' s performance as we increased the number of genes in context. We used a Smart-seq-v4 dataset from Jorstad et al., averaging around 6000 NNZ genes per cell (see Supplementary Figure 5.34)[246]. As shown in Figure 3.2, we observed an overall increase in prediction accuracy across all labels as we increased the context from 200 to 8000 genes, even though scPRINT-2 was pretrained on only 3200 genes, demonstrating generalization to larger input contexts. Interestingly, classes such as sex and ethnicity reached much better predictive accuracy as we increased the number of genes. When using only the most expressed genes in context, we observed that cell types, which are often defined by highly expressed canonical genes, remained relatively high, even with only 200 genes in context (see Supplementary Figure 5.35).

Training scFMs on large dataset sizes does not necessarily improve the model's performance. It is the breadth of cell types, conditions, organisms, and cell quality that produces real generalization abilities. We showcased it here, with scPRINT2 able to label unseen organisms, improving its predictions across various context lengths and rare modalities. We also showed scPRINT-2 reaching state-of-the-art classification accuracy with our fine-tuning.

We will now see how some of our contributions in training loss and data augmentation can similarly improve performance in denoising and imputation in unseen modalities.

3.3.3 A multi-cell denoising auto-encoder task unlocks new modalities and performances

Not all single-cell datasets are at the sequencing depth and quality of Smart-seq-v4. On average, single-cell data has very low depth, preventing scFMs from learning features that

may only be seen in higher-quality cellular profiles.

Meta-cells and graph neural network encoder

In addition to biasing sampling toward cells with more non-zero genes (NNZ), scPRINT-2's dataloader now uses neighborhood information, whether defined in expression space or via spatial transcriptomics (see Figure 3.3; see Methods). This allows users to create models that take into account nearest neighbor cells during pretraining. This can be done, for example, by creating **meta-cells**. Meta-cells average the expression over the cell and its neighbors to artificially create a higher-depth cell with less dropout. We demonstrate that this approach achieves improved results across multiple model metrics, but not in denoising (see Table 3.1). While 17% of cells in the dataset have more than 2600 non-zero values, 11% had at least 3200. With NNZ-weighted sampling, we reach 33%. By adding metacells, half of our input expression profiles now have more than 3200 NNZ elements—allowing us to extend scPRINT-2's context to 3200 genes.

However, one can go beyond meta-cells and, instead of averaging, use a graph neural network (**GNN**) (see Figure 3.3; see Methods)[247, 81]. In this case, the set of neighbors' expressions is encoded in the input token of the transformer. We show that this improves the model's denoising ability. However, we also noticed a decrease in cell embedding and classification (see Table 3.1). Further experiments showed that this was mitigated with longer training time. As in the variable context case, we variably select 0 to 6 neighbors per minibatch, so the model learns to use a variable number of cell neighbors (see Supplementary Figure 5.36, see Methods for details on the choice of neighbors).

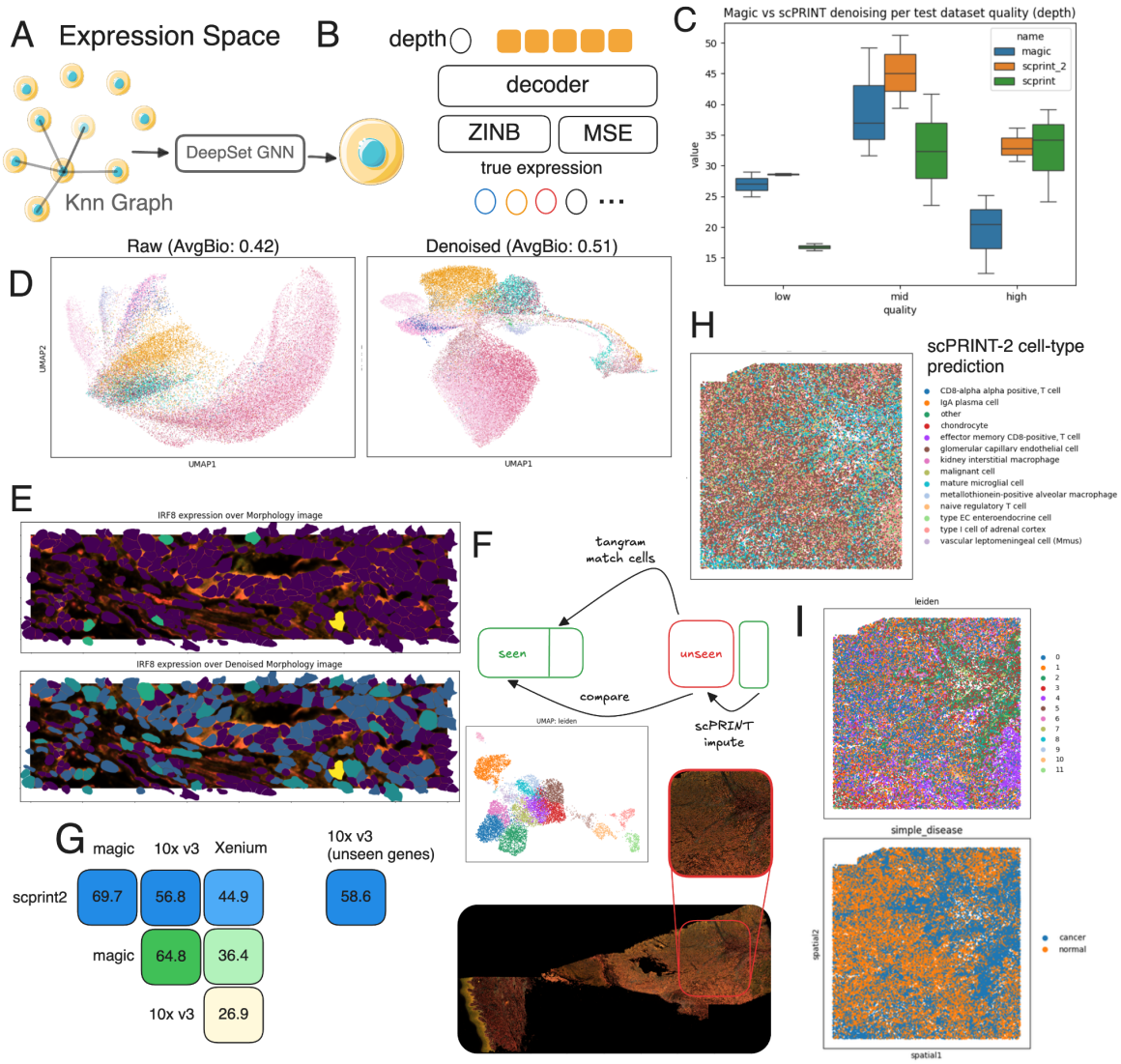


Figure 3.3: Presentation of the expression encoder and decoders and performance on denoising and imputation tasks. (a) Overview of scPRINT-2's multi-cell expression encoder and (b) scPRINT-2's expression decoder loss. Circles represent scalar values, orange blocks represent vectors. (c) Benchmark of scPRINT-2 on expression denoising over nine datasets of varying quality, compared to MAGIC and scPRINT-1. (d) UMAP of the Xenium's patches of cells' expression pre/post denoising with scPRINT-2. (e) Expression denoising of IRF8 with scPRINT-2 over a sub-patch of the Xenium melanoma dataset with cell contour overlaid. (f) Overview of the patch selection in the Xenium dataset, and of the mapping and pseudo-imputation with Tangram using a matched melanoma 10x v3 scRNA-seq dataset. (g) Correlation-based denoising & imputation scores of scPRINT-2 and denoising of MAGIC on the matched dataset. (h) scPRINT-2 cell type prediction over the Xenium melanoma patch. (i) Expression-based clusters and scPRINT-2 disease prediction of cells from the Xenium melanoma patch analyzed. Source data are provided as a Source Data file.

Pushing our analysis further, we realized that a mix of both scores, which we call **ZINB+MSE** (see Figure 3.3; see Methods), yields a better denoising score while retaining the ability to model zero inflation and uncertainty (see Table 3.1). Together, these updates have already made scPRINT-2 better than scPRINT-1 and even better than MAGIC on our denoising benchmarks (see Table 3.1, see Figure 3.3)[41]. While these results are already

state-of-the-art, we wanted to explore the effects of denoising and how to assess our model in unseen contexts.

Looking at denoising scores across technologies, we notice that scPRINT-1 tends to perform much better on datasets with higher NNZ genes, i.e., higher-quality datasets (see Figure 3.3, see Methods). However, within each dataset, scPRINT-1 struggles more with low-depth cells than MAGIC & scPRINT-2, which is more consistent overall. We explain this paradox by the fact that, beyond NNZ genes, the high-quality dataset often exhibits lower biases in the distribution of NNZ genes per cell (see Supplementary Figure 5.37). This also explains why MAGIC and scPRINT-2 perform better than scPRINT-1 in these biased datasets. Indeed, they can look at the neighbor's expression and model the expression biases this way. This usage explains the significant improvements in the low- and mid-quality datasets, making scPRINT-2 state-of-the-art across all tested contexts and modalities using its estimate of zero-inflation.

scPRINT-2 generalizes to unseen denoising tasks

Additionally, we decided to look at performance on a Xenium dataset, a modality completely absent from scPRINT-2's training (see Methods)[248]. We elected to use a large, recent skin melanoma dataset with a 5000-gene panel, reaching the upper limit of what is doable with current technology.

A first proof of scPRINT-2's denoising is the scIB biological truthfulness of the Xenium dataset, which improves over the raw expression embedding when using its embeddings (see Figure 3.3; see Supplementary Figure 5.38; see Supplementary Table 5.14). To further assess how well scPRINT-2 can denoise this unseen data modality, we leverage the optimal transport-based method Tangram[249]. We used Tangram to map each Xenium cell to another cell in a non-spatial 10X v3 dataset of similar skin melanoma[250] (see Figure 3.3). Here, the mapping quality is low due to many differences between the two technologies, e.g., number of cells, number of genes per cell, or biases in cell and gene types (see Supplementary Figure 5.39). Still, using the 10X v3 dataset as ground truth, we can see that MAGIC and scPRINT-2 recreate an expression profile that correlates more than 30% better with the 10X dataset than does Xenium (see Figure 3.3). There, MAGIC creates expression profiles closer to the 10X ones, while scPRINT-2 remains closer to the initial Xenium profiles, and both scPRINT-2 and MAGIC tend to agree more with each other than with anything else (see Figure 3.3).

Overall, this suggests that using a tool like scPRINT-2 might be a better alternative for denoising and imputing expression from Xenium than using a secondary non-spatial 10X dataset and aligning it with Tangram.

At the same time, MAGIC can only perform denoising and cannot impute expression for unseen genes. We thus use scPRINT-2 to impute a random subset of 5000 genes present only in the 10X v3 dataset. Interestingly, we noticed that feeding all 5000 (expressed in Xenium) + 5000 (unexpressed in Xenium) genes in context did not lead to good imputation. However, using scPRINT-2's generative architecture, we directly decoded the 5000 10X-only genes from the scPRINT-2's cell tokens generated on the 5000 Xenium genes (see Supplementary

Figure 5.40). We show that this imputation scores as high as the denoised Xenium genes (see Figure 3.3).

Finally, we also wanted to examine scPRINT-2's cell-label predictions on this unseen modality. While we did not have access to ground-truth labels in this dataset, we could already spot-check the validity of the predictions. Indeed, many cell types were labeled as *basal* or *epidermis*, with numerous immune cell labels in the cancer-induced lesion in the tissue (see Figure 3.3). This entire lesion region was labeled as *cancer* by scPRINT-2. This was striking as it contained mostly non-cancerous activated immune cells (see Figure 3.3; see Supplementary Figure 5.41). It likely reflects the biases of the pretraining dataset, where disease labels are often applied at the dataset level rather than the cell level, making scPRINT-2's disease predictions sometimes imprecise. Thankfully, many cells had the cell-type label '*malignant cell*'. These cells were distributed throughout the tissue and showed a strong signal for the five key literature melanoma genes (*BCL2, IGF1, EGFR, FGFR2, SOX10*) (see Supplementary Figures 5.42 and 5.43).

Overall, we have seen how scPRINT-2 can be used on challenging modalities to augment a given dataset with cell label predictions, expression denoising, and gene imputation. Showing yet again another axis of generalization. We will now focus on how some of our contributions in training loss and data augmentation can similarly improve performance in denoising and imputation in unseen modalities.

3.3.4 An efficient, hierarchical attention architecture makes scPRINT-2 generative

Efficient attention architectures and compression methods

Implementing transformer models on new modalities is a potent way to rethink some of their mechanisms. A common issue with transformer models is their memory and compute requirements, which grow quadratically with their context length (e.g., the number of genes in their input). This is even more pronounced in bidirectional transformers like most scFMs. With the introduction of scPRINT-1, we presented a model that could train in 3 days on a regular A40 GPU and on 50M cells, an order of magnitude faster than most similar scFMs. A first contribution to the scPRINT-2 architecture is the addition of state-of-the-art approaches to reduce the memory footprint and increase training speed. We modified the attention mechanism in multiple ways, using grouped-query attention (GQA) to reduce memory usage. We benchmarked additional attention mechanisms alongside Flash-Attention-3 to assess their performance and their speed.

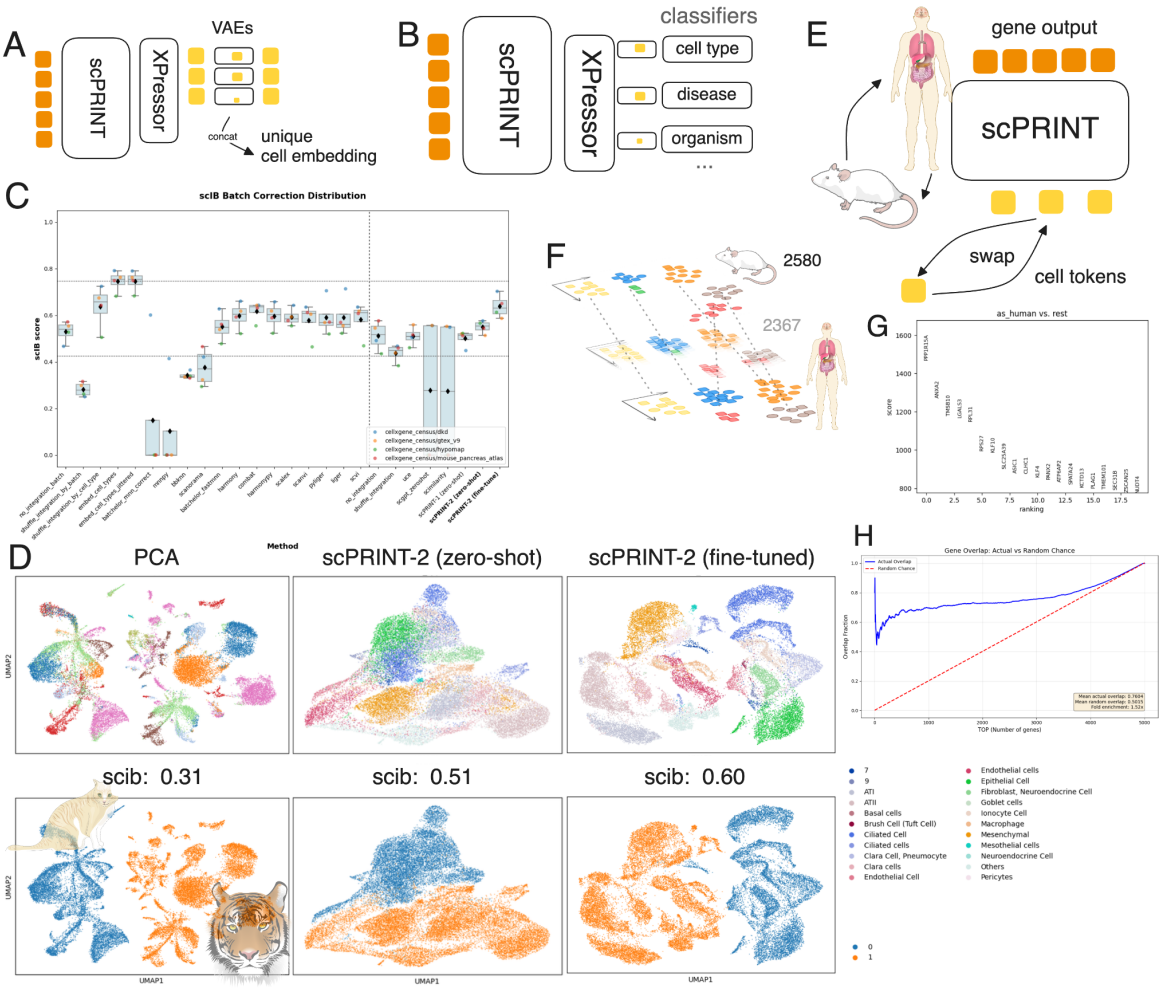


Figure 3.4: Presentation of the XPressor architecture and performance on cell embedding tasks. (a) Presentation of the XPressor with VAE-based compression. (b) Schematic representation of going from expression to classification with scPRINT-2, XPressor, and VAE-based compression. (c) Open-Problem scores for scPRINT-2 across all methods. (d) UMAPs of, respectively, PCA embeddings, scPRINT-2 zero-shot cell-type embeddings, and scPRINT-2 fine-tuned cell-type embedding colors by known cell types and batches, with scIB total scores. (e) Schematic representation of the counterfactual generation using scPRINT-2's embedding and replacing them for the organism class from mouse to human. (f) Illustration of the decrease in distance between initially unrelated datasets from applying this counterfactual approach. (g) Differentially expressed genes post vs pre mouse "humanization" with scPRINT-2. (h) Over-representation plot of the top positively differentially expressed genes in both human-like mouse and real human vs. mouse; the red line indicates random chance. Source data are provided as a Source Data file.

A first one is flash-**hyper attention**, which computes specific attention only on sets of keys and queries known to be similar via locality-sensitive hashing and clustering[64]. A second one is flash-**softpick attention**, a rectified softmax that decreases hyperactivation of specific tokens, often called attention sinks[251]. We also present our own subquadratic attention mechanism: **criss-cross attention** (see Methods), inspired by advanced concepts such as the Recurrent Interface Network (RIN) and the Induced Set Attention Block (ISAB)[252, 253]. It compresses attention by sketching it in context, using a doubly cross-attention mechanism with a set of latent tokens that get updated across layers (see

Supplementary Figure 5.44). We show that only criss-cross attention dramatically improved the model' s speed while retaining all its abilities (see Table 3.1). However, it is not yet compatible to retrieve gene networks from; for this reason, our scPRINT-2 architecture, for now, uses flash-attention-3 and XPressor.

On another direction, while single-cell analysis has leveraged VAEs for years to generate meaningful compressed representations of cells, transformers inherently lack this ability[254, 255, 256]. We use the **XPressor** architecture presented in Kalfon et al.[235], which compresses output gene embeddings into a set of cell embeddings and decompresses them back into their original gene embeddings (see Figure 3.1, Figure 3.4, and Methods). This innovative architecture draws on ideas that have existed in the transformer literature for several years[253, 257, 258, 259, 190]. We show in our ablation study that using XPressor results in a slightly better cell representation overall, but does not meet the statistical threshold. This difference might be explained by the limit in the number of epochs and the model's smaller size compared to Kalfon et al. (see Table 3.1, see Supplementary Table 5.13). We include an extension to this approach, in which one appends VAEs to each output embedding of XPressor to regularise the different cell embeddings generated by the model (see Figure 3.4). This addition allows us to choose a specific dimension for each cell embedding that is lower than that of XPressor. A second constraint is defined by applying the Kullback-Leibler divergence (KL) loss (see Figure 3.1, see Methods). This creates an information bottleneck for the different cell embeddings, pushing the model to select only the minimum amount of relevant information to represent the label. While our ablation study does not show improvement in cell embeddings with this approach, this is likely because each method was trained for only 20 epochs. Indeed, the VAE-infused model is taking longer to learn to classify cells. However, the batch correction score improved significantly, indicating that the different cell tokens mainly contained information about the class they encoded (see Supplementary Materials). Now that we have highly compressed cell-level embeddings (i.e., tokens), we can apply a **dissimilarity loss** between each for a given cell. This actively pushes them to be as different as possible (see Supplementary Figure 5.45; see Methods). We demonstrate that this tends to slightly improve the model' s output embedding in our ablation study (see Table 1).

These architectural changes make scPRINT-2 much more efficient at compression and zero-shot batch-correction. Indeed, on the open problem's benchmark, we observe an overall improvement over scPRINT-1, again becoming the state-of-the-art zero-shot method on the platform (see Figure 3.4). This zero-shot performance increase is solely due to the improvement in the batch-correction score from using our VAE method (see Supplementary Figure 5.46). We then fine-tune the XPressor architecture alone – our XPEFT approach – to further learn to remove batch effects and predict expert-annotated cell-type labels. We add a Maximum Mean Discrepancy (MMD) loss (see Methods) that penalizes the distance between batch elements[260, 261]. Doing so, we observe a jump in scIB scores, especially in biological truthfulness, as measured by the scIB metrics (see Figure 4C; Supplementary Figure 5.47), making scPRINT-2 the best-performing method in the benchmark.

scPRINT-2 generalizes to unseen cell embedding tasks

We then wanted to push our analysis further and test the zero-shot organism-level integration of scPRINT-2 on organisms unseen during training. Again, using our cat and tiger dataset presented in the second result section, we saw that already, scPRINT-2's general cell embedding performs better than doing no correction and keeps lot of biological truthfulness, as shown by the scIB score of 0.44 vs 0.37 for PCA (see Figure 3.4, see Supplementary Figures 5.48, 5.49, see Supplementary Table 5.15). Then, as often, taking the cell-type-specific embedding further increases the biological truthfulness to 0.49, mainly by generating a more faithful biological representation, as reflected in the scIB scores (see Figure 3.4, Supplementary Figures 5.49, 5.50, Supplementary Table 5.15). Again, using XPEFT, we achieve a tremendous 0.60 scIB score, placing us among the top 3 best-performing models in this category, behind SATURN and scGEN (see Figure 3.3, Supplementary Figures 5.49, 5.51, Supplementary Table 5.14). We note that even in this domain, many cell types didn't overlap across organisms. It is a common behavior in this benchmark, and similar cell types now almost overlap in the UMAP, hinting at shared neighbors (see Figure 3.3, see Methods)[262].

Finally, we wanted to examine the model's ability not only to integrate cellular profiles but also to generate entirely new ones at inference time in a zero-shot manner by combining cell tokens (see Figure 3.4). We first approach it using a matched mouse-human multi-organ atlas from Zhong et al.[243]. We then generated cell embeddings for all cells and computed an average "human"-ness cell embedding using the *organism* embeddings of all human cells. We regenerate an expression profile using 1. the human gene embedding and 2. the mouse cell embeddings, replacing the organism cell embedding with the human one (see Figure 3.4 and Methods). We thus generate a set of human-like cell expression profiles from mouse expression profiles. Using the 5000 most variable orthologous genes, we indeed observed a decrease in the Wasserstein-2 (W2) distance on this counterfactual conversion to human (see Figure 3.4, see Methods)[263, 264]. Applying a similar approach, but this time to generate females from males in the human dataset, we also notice a similar reduction in expression W2-distance from 1076 to 938.

Looking at how cell expression patterns change after this transition, we found that most of the top differentially expressed genes are the same as those identified in the differential expression analysis of the real human dataset (see Figure 3.4, see Supplementary Figure 5.52). Computing an over-representation test, we observe a robust 58% enrichment compared to random, with more than half of the top differentially expressed genes correctly predicted by scPRINT-2 in both over- and under-expressed genes (see Figure 3.4, see Supplementary Figures 5.53, 5.54). Looking at *Reactome_2022* pathway enrichments, we see multiple pathways related to immune system function, membrane-ECM (Extra-Cellular Matrix) interactions, and tissue elasticity, as well as many other molecular-level pathways (see Supplementary Figure 5.55). These align with previous analyses highlighting ECM and immune function differences between human and mouse tissues[265, 266].

Overall, we have shown that an entirely novel architecture and a set of learning constraints enable scPRINT-2 to generate high-quality embeddings in a zero-shot manner. Thanks to its multi-organism training, this can be extended to unseen species, while achieving even stronger results with fine-tuning. We have also demonstrated how one can use the

scPRINT-2's cell embeddings to generate counterfactual cellular profiles. This makes it a strong contender for performing atlas-scale analysis across tissues, diseases, and organisms, by learning to disentangle each cell component. We will now see how other parts of the models can be used to extract additional information.

3.3.5 High-quality contextual gene representations from scPRINT-2

scPRINT-2 has rich gene embeddings

scFMs don't just provide cell-level embedding, they have also been used to generate contextual gene-level embeddings given a cell's expression profile or to predict gene-gene connections. The model' s gene embeddings can be used for fine-tuning, such as to predict ATAC-seq activities or gene essentiality[73, 72]. We investigate the gene embeddings produced by scPRINT-2 and then delve into how its gene networks can be better extracted and assessed.

A good output gene embedding is also defined by the quality of its input. With scPRINT-2, we introduced a fine-tuning adapter layer on top of ESM3's protein embeddings, jointly trained with the model (see Methods). This approach is one of the few that improve gene network inference without decreasing any other metrics in our additive benchmark (see Table 3.1). It allows us to update gene representations during pretraining while maintaining the ability to work with unseen representations, e.g., from unseen species (see Figure 3.5).

It remains unclear, however, what the right approach is for selecting output gene embeddings, with some heuristics proposing using the last or second-to-last layer. Using our regular transformer model trained with masking, we demonstrate that its output gene embeddings contain only their own expression values (see Figure 3.5). However, when trained with the Xpressor architecture, clusters of genes appear (see Figure 3.5). This is sensible because Xpressor forces gene embeddings to be rich in meaning, as the compression block must query them. We have, however, noticed that for regular models that are not fully trained (only 20 epochs), the output gene embedding still contains some input ESM3 features (see Supplementary Figure 5.56). The number of enriched pathways in its output gene embedding cluster is still significantly less than for scPRINT-2's XPressor architecture (see Figure 3.5; see Methods)

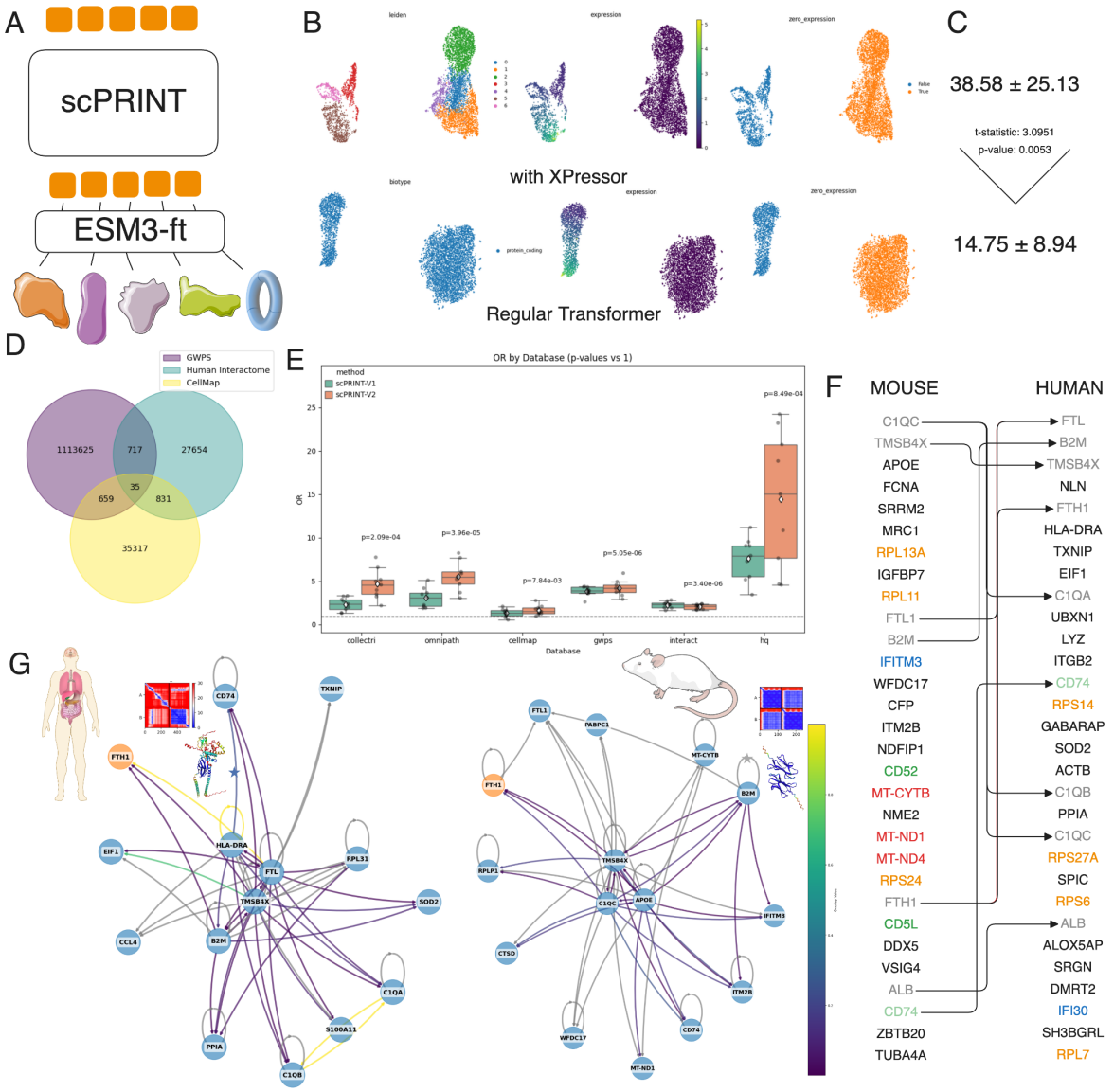


Figure 3.5: Presentation of the ESM3 fine-tuning and gene network study. (a) Illustration of fine-tuning of ESM3 while training scPRINT-2 using an adaptor layer. (b) Comparison of gene output-embeddings for a random cell in a model with the XPressor architecture and a model trained without. (c) On the side, the average number of pathways shown to be enriched in the gene output embedding clustering of each method using three main pathway databases. The number below is on the non-fully trained regular transformer; otherwise, no pathways are enriched (see Methods). (d) Comparison of ground truth networks' overlap between cellmap, the human interactome, and genome-wide perturb-seq. (e) Benchmark over six ground truth gene networks of scPRINT-1's gene networks with its extraction method and scPRINT-2's gene networks with its extraction method, over nine different human cell types from the same dataset. (f) Comparison of the top-30 hub nodes on both gene networks. Arrows link similar genes, and colors represent similar gene groups. (g) Subset of a gene network generated by scPRINT-2 seeded at FTL1, on human macrophage cells, and on mouse macrophage cells, edge color represents the RoseTTAFold2-PPI scores for these connections, grey means no score was computed. The AlphaFold-Multimer structure and amino-acid distance map are provided for the star-marked connections. Source data are provided as a Source Data file.

extracting gene networks from scPRINT-2

Thanks to the transformer architecture, one can go beyond gene output embeddings to examine gene-gene interactions via the model's attention layers. Following the tests reported in Kalfon et al., we observed, on average, no dramatic performance gains across the methods we tested (see Table 3.1). An issue we noticed is that the problem is not well-defined. Indeed, the ground truths widely disagreed with one another (see Figure 3.5; see Methods). Between the genome-wide perturb-seq (gwps) ground truth and omnipath, only 800 gene-gene connections were in common over the hundreds of thousands that each contained. This suggests that diversity of ground truth will be key to showcasing the breadth of potential gene-gene connections in the cell.

We thus gathered a new set of ground truth gene networks (GN)s from recent works. Our first approach was to use protein-binding datafrom AP-MS experiments within the O2US cell line, called the *cellmap*[267] (see Methods). Additionally, thanks to protein structure models, we are now able to compute putative interactions across millions of protein pairs; a first version of this analysis has been defined in the human *interactome* (see Methods). But here again, the disagreement was significant, with only ~1-4% of the connections in each ground truth being found in another, and no connections were reliably found across all five ground truths (see Supplementary Figure 5.57).

Acknowledging these disagreements, we benchmarked them against nine human cell types from the same dataset using scPRINT-1 and scPRINT-2. We use a gene network extraction method that is more computationally demanding but biases the network towards co-expressed genes (see Methods). We see that scPRINT-2's performance was often greater or similar across all benchmark networks, as indicated by the odds-ratio measures (see Figure 3.5; see Methods). We did not see a similar trend, however, on AUPRC (see Supplementary Figure 5.58). This suggests that our method is more accurate for its top-K connections. Indeed, the strongest human interactome connections were overrepresented in scPRINT-2, more so than in scPRINT-1.

cross-organism gene network analysis

To continue on our cross-organism analysis, we also aimed to further characterize some of the genes observed in our previous human/mouse datasets by interrogating the cell-specific GN identified by scPRINT-2 in *Macrophage* cells from both mammals. Looking at their hub nodes, we see that many are common and represent key conserved cell immune pathways, such as *feroptosis*, *vitamin B12*, and *Pathogen Phagocytosis Pathways* (*WikiPathway_2023_Human*), with genes like *C1Qs, RPs, ALB, and APOE*[268, 269]. These mainly relate to the macrophage' s internal machinery, which is designed to eat and destroy pathogens. Other genes were clear markers of macrophages (*CD74;LYZ) and/or immune cells (HLA-DRA, B2M) or their pathways, such as interferons alpha/gamma and MHC Class II (MSigDB_Hallmark_202076–78).* Interestingly, these networks share only 30% similarity when considering the top 20 connections for each gene. But what seemed like differences in connections and top 50 hub genes tended to disappear after thorough analysis, such as with the Ribosomal proteins, which are related in the kinds of pathways they are part of, or in their relationships

in the PPI_Hub_Proteins database[270] (see Figure 3.5).

We then extracted a subset of the macrophage networks, seeded at the *FTH1* gene, for both organisms, focusing on the top 15 connected nodes and their top 60 edges (see Figure 3.5; see Methods). We observed a set of hub genes in both subnetworks, with some genes being shared between human and mouse. Interestingly, these hub genes had more interactions in the human interactome ground truth than non-hub genes. We also noticed that the "hub-ness" of the subnetworks can be very variable and seems to depend on the "seed" gene (see Supplementary Figure 5.59).

By overlaying the human interactome ground-truth values on our subnetworks, we found that only a small subset of connections was marked as valid (i.e., score above 0.6) in the ground truth (see Figure 3.5). In the mouse *Macrophage* subnetwork, almost no connections were recovered, but this may be explained by the fact that the ground truth is the "human" interactome, computed using human proteins rather than mouse proteins. We thus wondered whether we could use scPRINT-2 to cross-validate the interactions present in this ground truth. Indeed, we know that the human interactome values are not directly computed from AlphaFold-multimer's interaction probability (interface predicted Template Modeling score (ipTM)); they come from a simpler model called "RoseTTAFold2-PPI". Testing a couple of connections predicted to be low ipTM by RoseTTAFold2-PPI but found by scPRINT-2, we readily identified two: HLA-DRA/CD74 and B2M/B2M, which, when passed to AlphaFold-Multimer, indeed formed an interaction with an ipTM of more than 0.6. This showcases the potential of scPRINT-2 in this domain and future directions for GN inference.

We have seen here how scPRINT-2's output gene embeddings and attention matrices can be used to extract meaningful biological insights and drive hypothesis generation in a cell-to-cell, state-specific manner. These outputs can also be used for fine-tuning purposes and in explainable AI-driven analysis. We also pushed our GN analysis further, defining additional benchmarks and a more powerful GN extraction mechanism. We demonstrated cross-species analysis and presented the tantalizing possibility of merging foundation models at different scales, including ESM3 fine-tuning, AlphaFold Multimer, RoseTTAFold2-PPI, and scPRINT-2. While these are just examples, they demonstrate what aggregating multiple bodies of evidence across scales can achieve for genetic interaction predictions. A first step towards using scFMs, protein Language Models, and structural models in coordination, to shed light on the cellular machinery.

3.4 Discussion

In this work, we present a gymnasium of tasks to benchmark scFMs in multiple contexts. Together with an efficient and reproducible pipeline, we test the benefits of 42 different parts of scFMs structures, encoding, and training. In this additive benchmark, 12 of these are our own contributions to scFMs, including GNN-based expression encoding, cross-foundation model fine-tuning, sub-quadratic attention mechanisms, and rich losses. This massive benchmark is the first of its kind for scFMs and assesses four different tasks. It allowed us

to identify bottlenecks and limitations, issues that we solved in subsequent analysis. Indeed, future benchmarks will benefit from using more diverse datasets, tasks, and ground truths.

We have also presented the largest pretraining database to date, encompassing more organisms, conditions, and data modalities. We have seen that, while more work is needed to obtain higher-quality, well-annotated datasets, our dataloader and preprocessing pipeline have made the most of this vast database.

Using the best feature combinations from our additive benchmarking, we build and train a next-generation cell Foundation Model, scPRINT-2. We demonstrate that, although currently 5 times smaller, scPRINT-2 outperforms scPRINT-1 across all benchmarks tested. On denoising, scPRINT-2 becomes state-of-the-art, and with our fine-tuning approach, it also outperforms every other method on the batch-correction and classification tasks of the open-problem benchmarks.

We then challenge scPRINT-2 on tasks of high relevance for cellular biology, highlighting some pitfalls in current benchmarks. We show that scPRINT-2 acquires generalizable abilities across unseen modalities and organisms, while remaining consistent in its predictions. We demonstrate it across many tasks, including cross-organism integration, unseen gene imputation, and counterfactual reasoning.

Finally, we present tools for easily extracting labels, cell-specific gene embeddings, imputing gene expression, performing gene network inference, and working with organisms unseen during pretraining. We believe our results demonstrate many domains where scFMs might confidently replace approaches that rely on heuristics, atlases, and a variety of tools and packages. However, much work remains.

Current ground-truth cell annotations are cluster-based and obfuscate the complexity of cellular states by inherent clustering biases. Batch correction metrics are similarly biased, and top scores can be easily gamed; gene network ground-truths are not cell type specific and likely filled with false negatives. Data diversity and quality are the principal pretraining bottlenecks, and efforts will be needed to improve foundation models. Many other key modalities, such as measuring time and perturbation effects, remain scarce. They will become increasingly helpful for enriching the future comprehensive benchmarks of next-generation cell foundation models.

Our analysis and contributions highlight powerful features of scFMs and provide guidance for designing benchmarks that better highlight their strengths and weaknesses. scPRINT-2 presents a direction for future improvements, with more specialized architectures and using a combination of biological FMs working jointly across modalities and scales. This next-generation scFM is a step forward in the design of AI for cell biology.

3.5 Methods

We present an additive benchmark with over a dozen contributions to the pre-training tasks, losses, and architecture of single-cell foundation models. Along with it, **scPRINT-2** (pronounced “sprint”), a next-generation model trained on the best-performing contributions.

We analyze its out-of-distribution generalization and present methods for querying and fine-tuning it to solve various tasks. We will go through the specific techniques that made it possible.

3.5.1 Additive benchmark

We now describe in matched order with respect to Table 1, the methods behind the multiple contributions tested in our additive benchmark (see Results section 1). We bolded the ones that are further defined in the methods. In this benchmark, we are using and testing the:

1. **"base model",** every subsequent element is applied to the base model
2. "medium model", larger base model, see the base model section
3. "negative control", untrained base model

Architecture

4. "no dropout", where we remove the dropout initially applied in the base model
5. "large classifier", where the classifier sizes are increased from [input - output] in the base model to [input - 256 - output]
6. "MVC", where we replace the base model's decoder with the cell embedding's MVC approach of scGPT[73]
7. "no decoders/generation", where we removed the base model's decoder, getting a masking+classification only pre-training

Data

8. replacing our pre-training dataset with "Tahoe" 's 100M dataset
9. Chan Zuckerberg Institute ("CZI")'s cellxgene database (version 2024)
10. replacing our pretraining dataset with "CZI + Tahoe" with Tahoe's 100M database
11. replacing our pretraining dataset with "all databases", both CZI, Tahor, and Arc's scBasecount[231, 230]
12. replacing our pretraining dataset with "only 200 random" human datasets
13. replacing our sampling with a "sampling without replacement"
14. **replacing our sampling with "cluster-based sampling only"**
15. **adding "meta-cell" during pre-training**

Attention

16. replacing FA3 with "softpick" attention, using the approach of Zuhri et al.[251]
17. replacing FA3 with "hyper"-attention, using the approach of Han et al.[64]
18. replacing self-attention with **"criss-cross" attention layers**
19. **adding "XPressor" layers**

Losses

20. **adding "contrastive learning"**
21. **adding "elastic cell similarity"**
22. **"no embedding independence loss", removing the embedding independence loss**
23. replacing the ZINB loss with Mean Squared Error ("MSE")-loss
24. **replacing the ZINB loss with "ZINB+MSE" loss**
25. **adding a "VAE compressor" loss to the Base model**

Tasks

26. **adding "variable context length" and a larger context**
27. **replacing masking with a Transcription Factor "(TF)-masking" task**
28. replacing masking with "denoising", using the approach in scPRINT, with a random level of denoising (see below)
29. "no classification", removing the classification pre-training task
30. **adding an "adversarial classifier"**

Input

31. replacing log1p normalization with "sum normalization" where each expression profile is normalized to sum to 10,000
32. "no random level of denoising" where we remove the random level of denoising, see the denoising section
33. **where we replace the expression encoder with a Graph Neural Network ("GNN") encoder**

34. where we replace the continuous expression encoder with a "binning" version, following the approach of scGPT[73]
35. where we are "using only expressed genes", as in scGPT and geneformer
36. "without using gene location", removing the gene location information in the input tokens.
37. "learn gene embedding" where we replace the ESM3 gene embedding with learnt embeddings, as in scGPT and Geneformer.
38. **replacing the ESM3 gene encoder with a "fine-tuned ESM3" gene encoder**

The full training traces of the entire additive benchmark are available on weights and biases:

https://wandb.ai/ml4ig/scprint_ablation/reports/scPRINT-2-additive-benchmark--VmlldzoxNTIyOTYwNA

Base model and training

The additive benchmark is performed on a small model with 18.2M parameters, an embedding dimension of 256, and 8 layers and 4 heads. The model trains for 20 epochs of 20,000 batches of 64 cells per batch. Validation is performed on 10,000 minibatches. We otherwise use the same optimizer and hyperparameters as for scPRINT-2 (see **pre-training** in Methods)

Gene expression is encoded using ESM3 embedding, with gene location and MLP-based expression encoding added, as described by Kalfon et al. The output is decoded using an MLP that takes the output embeddings and depth information, then outputs a scalar expression value.

The base model is trained on CZI's cellxgene census dataset, version 2024 (compared to 2022 in Kalfon et al.). The pre-training task uses a 30% gene expression mask with an MSE loss (as is common for BERT-like encoder transformers)[73, 76, 72]. The Base model also uses a multi-cell-token generative loss as described in Kalfon et al.[111]. It also performs matched multi-class hierarchical classification, as defined below (see **Hierarchical classifier** in the Methods**).** Finally, it also uses a dissimilarity loss between each of our cell embeddings for a given cell (see embedding independence in Methods).

Each of these decisions is assessed within our additive study.

We pre-train the base model 6 times across multiple seeds to generate error bounds. We train using Flash-Attention-3 on 1 H100 GPU, each training of 20 epochs taking roughly 2 days. Some runs were done on A100s and V100s; we thus had to rescale the time duration for some of these runs.

Some additive study runs use denoising as a training strategy or larger context lengths when it seemed likely that this would best highlight the abilities and shortcomings of the benchmarked element.

The **medium model** size uses an embedding dimension of 512, with 16 layers and 8 heads.

The **negative control** is a model that was not trained at all.

Weighted sampling

The goal of weighted random sampling is to de-bias regular random sampling of cells in contexts where many cells have similar profiles and expression patterns, while others are rare cell types.

We use weighted random sampling on our training data based on all the different class values we have to predict. We use a factor of S_1, meaning the rarest elements will, on average, be sampled only S_1 times less than the most common ones. The sampling factor used for each group is then $\frac{S_1}{c+S_1}$, instead of $\frac{1}{c}$, where c is the number of cells in each group.

Cluster-weighted sampling

The goal of cluster-weighted sampling is to improve weighted sampling in the condition where cell-type annotations are poor or non-existent.

For cluster-weighted sampling, we simply use the labels obtained by applying Leiden clustering to the K-NN graph of cells for each dataset during preprocessing. We used a resolution of 1 and 15 neighbors. We merge clusters if their centroid correlation exceeds a threshold (here 94%). This cluster label is then treated similarly to other labels, such as *cell_type*, *sequencer*, etc.

In this context, within datasets that lack information about tissue of origin or sequencer, or that belong to the same categories, cells from cluster 1 will be sampled with equal weight from those datasets. The sampling is not dataset-specific. This decision arises because most datasets contain some information about their tissue of origin or disease, and cluster sizes of data from the same tissue/disease often represent similar cells.

This can be applied to any dataset for training models.

Depth-weighted sampling

The goal of depth-weighted sampling is to sample cells with higher quality, in terms of the number of genes expressed, more often.

For depth-weighted sampling, we scale each cell' s sampling probability by its non-zero (NNZ) gene count. Similarly, we scale this value, but this time we apply a sigmoid function beforehand to reduce the impact of extreme values.

The values shown for Input were the ones we used across our research and were selected manually.

Algorithm 1 Depth-weighted sampling using sigmoid scaling

Require: nnz: number of non-zero genes per cell
Require: midpoint = 2000: sigmoid midpoint
Require: steepness = 0.003: sigmoid steepness parameter
Require: scale = 1000: scaling factor
Ensure: sampling probability for each cell

▷ Apply sigmoid transformation centered at midpoint
sigmoid_values $\leftarrow \frac{1}{1+\exp(-\text{steepness}\times(\text{nnz}-\text{midpoint}))}$
▷ Scale sigmoid output to range [1, scale]
probability $\leftarrow 1 + (\text{scale} - 1) \times \text{sigmoid\_values}$
return probability

This can be applied to any single-cell dataset for training models.

Multi-cell sampling

For all our datasets, our preprocessing pipeline computes a K-NN graph from the PCA of the scaled, log-transformed expression data. For each sampled cell, scDataloader also retrieves its k-NN cell ID and loads them, along with their distance information. Here, we set K to 6 and the PCA components to 200.

We set the number of PCA components to 200 to retain as much information as possible, while accounting for rare cells whose expression might have only a small impact on the first PCA components.

We set K to 6 to balance the computational resources required to sample 6 times more cells per minibatch with the need for enough neighboring cells. Indeed, these computational resources are more prevalent for smaller models that perform fast iterations across many cells than for larger models. 6 neighbors per sampled cell was our limit for a small foundation model like scPRINT-2. We also note that there is likely a rapid diminishing returns beyond 6 to 15 cells for most datasets as we start sampling more often cells that are less similar to the center one.

During scPRINT-2 training, we select 0 to 6 neighbors per minibatch, so the model learns to use a variable number of cell neighbors.

GNN Expression encoder

The goal of the GNN expression encoder is to increase the information the foundation model can obtain from 1 cell to a set of neighboring cells, thereby dramatically reducing input noise.

The GNN takes multiple expression values as input, optionally along with corresponding cell-cell distances, and returns a vector encoding this information. Both continuous and

GNN encodings can be configured to receive either logp1-transformed expression data, sum-normalized expression, or both. The GNN follows the DeepSet[247] implementation:

$$E_j = \text{DeepSet}(x_{ij}, n_{ij}, d) = \phi_1(\phi_2(x_{ij})||\phi_3(n_{ij}, d_{ij})) \tag{3.1}$$

Where:

- x_j is the center cell's expression for the gene j
- $n_{ij} \in \mathfrak{R}^k$ is the K nearest neighbor cell's expression for gene j and cell i
- $d_{ij} \in \mathfrak{R}^k$ is the distance of each neighboring cell to the center one
- ϕ_i are MLPs.
- $||$ is the concat operation

We selected K to be a random number between 0 and 6 during training and 6 at inference.

ESM3 fine-tuning gene-encoder

The goal of ESM3 fine-tuning is to get the best of both worlds between learning token features from the data and using learnt protein representations from a pLM as a prior.

We encode/tokenize gene IDs using ESM3[271]. The mapping process happens in the following way:

- A gene name is mapped to its canonical protein name using Ensembl114.
- We recover the protein sequence of the protein using Ensembl
- We use the protein sequence to generate an embedding using ESM3 by averaging all its amino-acid output embeddings.

For the fine-tuning part, we reuse the fine-tuning approaches presented in Kalfon et al., which place an additional adapter layer after mean-pooling and before feeding the protein representation to the model. Interestingly, using gene expression as a further signal to the adaptor layer often led to training instability.

Biased attention

The goal of biased attention is to orient our attention matrix towards genetic interaction priors to improve learning and the model's biological fidelity.

We leveraged the Rcistarget computation and ranking of the human genome for 10kb down- and upstream of each target gene[94], available at the Aerts Lab cistarget databases[1]. Using this information, we generate a weight matrix M that links each motif-defined TF to its target genes.

Given this matrix, we bias the attention matrix for all heads and layers using the attn_mask parameter of the torch.nn.functional.scaled_dot_product_attention function.

It appears in the attention computation like so: $\text{softmax}\left(\frac{QK^\top}{\sqrt{d}} + M\right)V$ where M is the attn_mask matrix and is real-valued.

Criss-Cross attention

The goal of criss-cross attention is to create an efficient attention mechanism by learning, in context, a factorisation of each attention matrix.

In criss-cross attention, we replace the self-attention mechanism with a double cross attention between the N input elements and M latent tokens (see Supplementary Figure 5.45). This is thus replacing a N^2 computation with a $2NM$ one, hence going below the quadratic bottleneck of attention. This bears resemblance to the ISAB architecture, XPressor, and perceiverIO[252, 235, 258, 257, 259, 190]. M, in our case, is set to 10: our 9 predicted classes plus an additional token.

Effectively, the M latent tokens are learnt at the first layer of the models. At the same time, they could also be generated from a sketching or principal components analysis (PCA) of the input tokens. They also get updated during the attention computation, so that at the second layer.

We replace the traditional attention computation $X_{l+1} = \text{Attention}(X_l, X_l, X_l) + X_l$, where Attention takes as input the Query, Key, Value elements, with:

$$\text{Attention}(X_1, X_2, X_3) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V \tag{3.2}$$

With

- $Q = X_1 W_Q, K = X_2 W_K, V = X_3 W_V$
- $W_Q, W_K, W_V \in \mathfrak{R}^{dxd}, X \in \mathfrak{R}^{Nxd}$

In self-attention $X_1 = X_2 = X_3$

[1] https://resources.aertslab.org/cistarget/databases/homo_sapiens/hg38/refseq_r80/mc_v10_clust/gene_based/hg38_10kbp_up_10kbp_down_full_tx_v10_clust.genes_vs_motifs.rankings.feather

In Criss-Cross attention, the algorithm becomes:

$$V_{l+1} = Att(V_l, X_l, X_l) + V_l \tag{3.3}$$

for the latent update and

$$X_{l+1} = Att(X_l, V_l, V_l) + X_l \tag{3.4}$$

for the main update

with $X_l \in \mathfrak{R}^{Nxd}$ the main embeddings and $V_l \in \mathfrak{R}^{Mxe}$the latent embeddings

XPressor model

The goal of the XPressor architecture, as presented in Kalfon et al., is to replace and generalize the class-pooling of other transformer models and the bottleneck learning of scPRINT. This makes the model more powerful at encoding cell-level features while also separating cell-level tokens from gene-level tokens. Finally, it enables a new mode of Parameter-Efficient Fine-Tuning. This bears similarities to the ideas presented in criss-cross attention above.

The **Xpressor** block uses as input a set of learned latent tokens T. It then performs cross-attention between the last layer of the gene embeddings and the latent tokens. The goal is for the **Xpressor** layers to be of smaller dimensions and context size than the main transformer layers, such that we end up with C_j a set of n tokens of dimension d_t generated from the encoded gene expression and ID matrices E_j , and G. Where $G$and $E_j$are sets of m tokens of size d_c representing the IDs of the genes and their corresponding expression in cell \$j\$, respectively, where $d_c < d_t$ and $n << m$:

$$O_j = Transformer(E_j, G) \tag{3.5}$$

$$C_j = Xpressor(O_j, T) \tag{3.6}$$

For a cell j, with the **Xpressor** being initialized with a learned set of input cell tokens, and C_j being the cell tokens associated with the input E_j.

The **Transformer** and **Xpressor** are both transformers with l_1 and l_2 layers, respectively. Indeed, we have designed both layers to contain a cross-attention architecture (see Figure 4A, Supplementary Figure 5.28) such that we can also do:

$$O_j = Transformer(C_j, G) \tag{3.7}$$

With O_j the output of the **Transformer** when using the **Xpressor** representation as input.

We add an optional MLP after cross-attention to transform the embeddings before the self-attention round. In our example, the decompression is performed using gene ID tokens as input only. These tokens remain the same for all cells of a given organism and thus do not depend on j. In the context of protein language models, for example, this would be replaced by positional tokens.

As shown in Supplementary Figure 5.28, the **Transformer** blocks are applied twice. The first application serves as an "encoder", using only self-attention, while the **Xpressor** and the second application of the **Transformer** blocks act as "decoders". We follow these definitions from the original "Attention is All You Need" paper[66]. It should be noted that, in our case, cross-attention is performed before self-attention.

Related ideas have also been explored in the NVEmbed paper, where the authors propose a cross-attention-based method to update tokens using "latent" tokens and some additional prompting tricks[190].

XPressor can be applied during pre-training or fine-tuning to replace mean-max-class pooling in Foundation models.

VAE-based compressor model

The goal of the VAE-based compressor is to reduce information sharing between output embeddings by penalizing the amount of information each embedding stores (see Figure 3.4, Supplementary Figure 5.28)[272].

Each VAE-based compressor is explicitly applied to a cell embedding, compressing it into a relevant latent dimension. It has a 2-layer MLP encoder and a 2-layer MLP decoder. In cases where only a small set of possible elements exists, such as in sex embeddings or cell culture, one can use the Finite Scalar Quantization (FSQ)-VAE[189].

FSQ-VAE discretizes each latent dimension **independently**. Specifically, the encoder outputs d values, each constrained to lie within a bounded range (e.g., [-1, 1]). Each dimension is then quantized into one of M discrete levels within that range (in our case 2). This dimension-wise quantization can be implemented as either a hard nearest-bin assignment or a differentiable approximation thereof. Because FSQ enforces scalar-level discretization, it provides a simpler and more fine-grained alternative to VQ' s vector-level codebook approach, while still offering strong regularization of the latent space.

In our case, all VAEs with fewer than 8 latent dimensions used the FSQ-VAE approach.

It can be applied on top of any output embedding at pre-training or fine-tuning.

ZINB+MSE loss

The goal of the ZINB+MSE loss is to make the model' s expression-level prediction as precise as possible (thanks to the MSE) while preserving the ZINB' s expressivity and uncertainty estimation.

scPRINT-2 uses a novel expression decoder for foundation models that outputs the parameters of a zero-inflated negative binomial (ZINB) distribution for each gene i in cell j. The *ZINB* distribution is defined as

$$X \sim ZINB(\mu, \theta, \pi) \tag{3.8}$$

Where the parameters μ, θ, π are obtained from a multi-layer perceptron (MLP) applied to the expression embeddings outputted by the transformer model at its last layer (e), which are the:

$$\mu, \theta, \pi = MLP(e||d) \tag{3.9}$$

The MLP is a two-layer neural network with dimensions [*d+1, d*, 3], where || denotes the concatenation operation.

Based on the work of Jiang et al.[273], zero inflation is the best distribution for a broad range of transcriptomic measurements, as some measurements exhibit sufficiently high dropout rates and require a zero-inflation term to model them. In our case, and similarly to scVI[254], we define our *ZINB* as

$$ZINB(x|\mu, \theta, \pi) = \pi\delta_0(x) + (1 - \pi)NB(x|\mu, \theta) \tag{3.10}$$

Where $\delta_0(x)$ is a point mass at zero, and $NB(x|\mu, \theta)$ is the negative binomial distribution with mean μ and dispersion θ.

Compared to scVI, where the overdispersion parameter θ is learned for each gene, we make scPRINT-2 output it together with μ, π (see Supplementary Figure 5.40)

Effectively, the model learns that dispersion may vary across genes, sequencers, cell types, and sequencing depths.

In addition, the loss adds an MSE term computed from the μ and θ output of the MLP, comparing for a gene i,

$$e_i = \mu_i \times (1 - \sigma(\pi_i)) \tag{3.11}$$

to the logp1-transform of the expression using mean-squared-error.

Where e_i is the predicted expression of gene i and σ is the sigmoid function:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(e_i - \log_2(x_i + 1))^2 \tag{3.12}$$

The zinb+mse loss is the addition of both losses with a scale parameter, here:

$$L_{ZINB+MSE} = L_{ZINB} + 0.5 \times L_{MSE} \tag{3.13}$$

This loss comes as a replacement for the classical MSE or ZINB in scRNA-seq models.

Embedding contrastive loss

The goal of this contrastive loss is to remove some batch effect by pushing cell embeddings obtained from the expression profile after different perturbations to be more similar to each

other than they are from cell embeddings of other cell profiles, using the InfoNCE[212] loss:

Algorithm 2 Contrastive Loss (InfoNCE)

Require: x: embeddings of cells post perturbation A [batch_size × feature_dim]
Require: y: embeddings of same cells post perturbation B [batch_size × feature_dim]
Require: τ: temperature scaling parameter = 0.3
Ensure: contrastive loss value

▷ Compute cosine similarity matrix

$S \leftarrow \text{cosine\_similarity}(x, y)/\tau$

▷ Where $S[i, j] = \frac{x[i] \cdot y[j]}{\|x[i]\| \|y[j]\| \tau}$

▷ Create positive pair labels

$\text{labels} \leftarrow [0, 1, 2, \ldots, \text{batch\_size} - 1]$

▷ Compute cross-entropy loss

$\text{loss} \leftarrow \text{cross\_entropy}(S, \text{labels})$

▷ Which expands to:

$\text{loss} \leftarrow -\sum_i \log \left(\frac{\exp(S[i,i])}{\sum_j \exp(S[i,j])} \right)$

return loss

This loss can be added to any scFMs at pre-training or fine-tuning (see Supplementary Figure 5.45).

Elastic cell similarity loss

The goal of this loss is to reduce batch effects by pushing cells that are similar to become more similar and cells that are dissimilar to become more dissimilar[73].

We implement the **cell similarity loss** of scGPT, where, given cell embeddings $e \in \mathbb{R}^{m \times d}$, where m is the number of cells and d is the embedding dimension:

$$L_{similarity} = \frac{1}{m(m-1)} \sum_{i \neq j} 1 - (\max(0, \hat{e}_i^\top \hat{e}_j) - \tau)^2 \tag{3.14}$$

Where:

$\hat{e}_i = e_i / \|e_i\|_2$ is the L2-normalized embedding of the cell i

τ is the similarity threshold (default 0.3)

$m(m-1)$ is the number of off-diagonal pairs

This loss can be added to any scFMs.

Embedding independence loss

The goal of the embedding independence loss is to push the different class-level embeddings of a cell to encode distinct information by making them orthogonal (see Supplementary

Figure 5.45).

Implementing a set of disentangled embeddings is not straightforward. In our case, we push the embeddings to be as different from one another as possible, with an **independence loss** defined as

$$L_{independence} = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{i'}^{m} 1 - cos(e_i, e_{i'}) \tag{3.15}$$

where e_i and $e_{i'}$ are the cell embeddings, m is the minibatch size, and cos denotes the cosine similarity. This pushes each embedding to represent different information from the others.

This loss can be added to any scFMs at pre-training or fine-tuning.

Hierarchical classifier loss

The goal of the hierarchical classifier is to enable efficient label predictions for a set of related labels defined by a known graph.

The scPRINT-1 classifier generates predictions for all possible labels in a hierarchical ontology, while producing logits only for the most fine-grained elements. To predict the other elements, it only has to aggregate their children' s logits. We improve this loss in scPRINT-2 by using the entire ontological graph: e.g., if a cell is an *olfactory neuron*, then it is also a neuron. If the classifier predicts *glutaminergic neuron*, it is wrong at this level but correct for *neuron*, meaning we penalize it less overall than a non-neuron label, like *fibroblast* (see Figure 3.2).

In conjunction with our weighted sampler, this allows the model to learn rich gradients from a low volume of data. We also implement two additional classes for predictions in our hierarchical classifier compared to scPRINT-1: age and tissue of origin.

During pre-training, we perform label prediction for different classes, e.g., cell type, disease, assay, age, tissue, ethnicity, sex, and organism. We created a specific relabeling of the age label that could be very fine-grained, e.g., 2 weeks, 3 weeks, 35 years old, 36 years old, into biologically relevant groups such as *embryo, fetal*, *6-month-old*, *1-year-old*, *adolescent*, young adult, and so on. We mapped both human and mouse data this way to a common age profile. These were the only two species with such labels available. The labels follow a hierarchy defined by ontologies: the Cell Ontology for cell type, MONDO for disease, EFO for assay, HANCESTRO for ethnicity, HSAPDV for age, UBERON for tissue, NCBITaxon for organism, and EFO for sex[274, 275, 232]. We do not compute the loss for cells with the unknown label.

The algorithm thus becomes:

The hierarchical loss is available as a standalone function on GitHub Gist[2].

[2]https://gist.github.com/jkobject/5b36bc4807edb440b86644952a49781e

Algorithm 3 Hierarchical Classification Loss

Require: pred: predicted logits [batch_size × n_leaf_labels]
Require: cl: ground truth labels [batch_size]
Require: labels_hierarchy: binary matrix [n_parent_labels × n_leaf_labels]
Ensure: hierarchical binary cross-entropy loss

```
newcl ← zeros[batch_size × n_leaf_labels]                ▷ Initialize target matrix
weight ← ones[batch_size × n_leaf_labels]                ▷ Initialize weight matrix
for each sample i where cl[i] ∈ [0, n_leaf_labels) do
    newcl[i, cl[i]] ← 1                                  ▷ Handle leaf labels
end for
for each sample i where cl[i] = −1 do
    weight[i, :] ← 0                                     ▷ Handle unknown labels
end for
for each sample i where cl[i] ≥ n_leaf_labels do
    parent_idx ← cl[i] − n_leaf_labels
    children ← children_of_parent[parent_idx]
    weight[i, children] ← 0                              ▷ Zero out weights for unknown children
    newcl[i, children] ← 1                               ▷ Set targets for all children
end for
for each parent p do
    addpred[p] ← logsumexp(pred[:, children_of_parent[p]])
    addnewcl[p] ← max(newcl[:, children_of_parent[p]])
    addweight[p] ← addnewcl[p] / sqrt(|children_of_parent[p]|)
end for
pred ← concat(pred, addpred)
newcl ← concat(newcl, addnewcl)
weight ← concat(weight, addweight)
return binary_cross_entropy_with_logits(pred, newcl, weight)
```

This loss replaces a classical pytorch classifier loss, such as binary_cross_entropy_with_logits.

Variable context length

The goal of the variable context length method is to decrease the model's bias toward a specific number of elements in context.

Indeed, we noticed that at inference time, the model's performance could be lower in variable-context situations (e.g., on gene-panel datasets or when using only expressed genes). We thus introduced a **variable-context** training scheme in which the model's context sometimes drops by a random amount (see Table 3.1; see Methods). This makes the model less biased toward a specific input context during inference and decreases training time (see Supplementary Materials). Again, here we see strong consistent improvement in the model's performance across our additive benchmark.

This can be applied to any transformer models where the number of elements in context can be chosen arbitrarily.

Adversarial classifier loss

The goal of the Adversarial classifier is to remove batch effect[44].

The adversarial classifier is applied only to the *cell_type* cell embedding and is tasked to classify the organism of origin for each cell. It uses the same MLP as regular classifiers (2 layers, 256 as inner dimension). We use the reverse_gradient operation on top of a simple softmax-based binary cross-entropy classifier loss as follows:

Algorithm 4 Adversarial Classifier Loss

Require: e: input cell embedding tensor [batch_size × feature_dim]
Require: c: input ground truth label [batch_size]
Ensure: cross-entropy loss
$e \leftarrow \text{grad\_reverse}(e)$ ▷ reverse gradient for adversarial behavior
logits $\leftarrow \text{MLP}(e)$ ▷ compute logits from embedding
return $\text{cross\_entropy}(\text{logits}, c)$

We use it to predict both organisms and sequencers. Sequencers are mapped to a set of coarser labels, as we cannot use the hierarchical classifier in an adversarial context. Indeed, as it is sigmoid-based, it could easily set all label logits to -inf.

This loss can be added during pre-training or finetuning of a foundation model, provided batch labels are available.

Algorithm 5 Gradient Reversal

Require: e: input tensor [batch_size × feature_dim]
Require: λ: scaling factor for gradient reversal = 1
Ensure: tensor with reversed gradients
  if forward pass **then**
    return e (identity function)
  end if
  if backward pass **then**
    e.grad_input $\leftarrow -\lambda \times e$.grad_output
    return reversed gradients
  end if
  return grad_input, None

TF-masking task

The goal of the Transcription Factor (TF) masking task is to push the model to pay more attention to TFs than to other genes.

For the Transcription Factor masking task, we reuse the classic 30% masking task used in the base model (see Base Model). We then list the ENSEMBL IDs of all 13,000 TFs across our 16 organisms and sample our mask, giving increased weight to the TFs. Here, the weight is set up to be 10 for TFs and 1 for the rest.

The tool can be applied to any other set of genes as a replacement for classical masking in scFMs.

3.5.2 Additive Benchmark's datasets

The gene network analysis is performed on a test kidney single-cell dataset, using 1000 cells from the same cell type, and is compared with the omnipath ground truth (also known as the omnipath benchmark) across all cell types. It is also performed for 1000 K562 cells, comparing it to a network assembled from all genes "i" whose expression changes significantly when gene "j" is perturbed, using a genome-wide perturb-seq dataset called GWPS benchmark[140].

Knowing that perturb-seq still often implies cell-type- and patient-specific off-target effects and cannot detect many direct effects[276, 277, 92, 278].

The cell type prediction uses accuracy, and batch correction uses scIB v2, as in Kalfon et al.[178]. Both the lung and pancreas datasets have also been used in Kalfon et al. They are test datasets, removed from the pre-training corpus, and both come from the initial scIB paper[144].

3.5.3 scPRINT-2

The model architecture is composed of:

- An **encoder/tokenizer** that takes multiple inputs, such as raw expression data, gene names, and gene locations, and embeds them in a high-dimensional space used by the transformer.
- A **trunk** with a bidirectional multi-head transformer, an XPressor bidirectional multi-head transformer, and a set of VAEs applied to each XPressor output embeddings.
- A **class decoder** that transforms the output cell embeddings of the XPressor into cell-specific label prediction logits over a range of classes.
- An **expression decoder** to transform the output embeddings into expression values

Of the above-cited additive benchmark elements, scPRINT-2 contains: **XPressor, all databases, denoising**, **cluster-based sampling, elastic cell similarity, ZINB+MSE, VAE compressor, variable context with larger context, TF masking, GNN expression encoder, and fine-tuned ESM3** (See Supplementary Figures 5.28, 5.36, 5.44, 5.45)

We now go into some more details about the model:

Encoder / Tokenizer

In scPRINT-2, each gene in a cell is converted to an embedding: It corresponds to the sum of 3 different elements:

1. An embedding representing the gene itself using ESM3 with a fine-tuning adaptor layer (see Methods)
2. An embedding of the gene location in the genome. This helps the model understand that genes with similar locations tend to be regulated by similar regulatory regions[279], a relationship well-known in cellular biology.

We encode the genes' locations using positional encoding. Every gene within 10,000 bp of the next is considered to be in the same location; otherwise, we increment the location by 1. We do this for all genes in the Ensembl database per organism.

We then embed these locations using the Positional Encoding (PE) algorithm of Vaswani et al. [66]. We notice that adding this embedding was important to prevent divergence during training.

3. An embedding of the gene expression in the cell and its neighbor using our **GNN** (see Methods)

Finally, during pre-training, a subset of 3200 genes is used to encode a cell expression profile. If fewer than 3200 genes are expressed in both the cell and its neighbors, we pad

them with randomly sampled unexpressed genes (meaning with an expression value of 0). This approach allows the model to see different patches of the same cell profile during training.

The full set of embeddings of cell i sent to the transformer is the matrix X_i where:

$$X_i = [g_0 + e_{i,0} + l_0, g_1 + e_{i,1} + l_1, ...] \tag{3.16}$$

Where g_j is the gene j encoding, $e_{i,j}$ is the encoding of the expression of gene j in cell i, l_j is the gene j location encoding.

Additionally, the Xpressor layers will receive a set of learnt prototype tokens representing the different class-level cell embeddings.

Trunk

The model "trunk" is a bidirectional encoder similar to BERT[57] with n layers, h attention heads, and a dimension of d. It uses the flashattention2[116] methodology implemented in Triton to compute its attention matrix. It uses the pre-normalization technique[280], with a sped-up layer norm implemented in Triton's tutorial[281]. It uses stochastic depth with increasing dropout probability[282] (see Base for details about small and medium-sized models).

It has a 2-layer MLP with a 4x width increase in its hidden layer and a GELU activation function.

Each Layer or block is composed, in order, of a layer-norm, self-attention, layer-norm, MLP, and layer-norm, cross-attention, layer-norm, MLP, which are only used during the decoding step. It has an additional m Xpressor blocks/layers applied to its 10 latent cell tokens.

The output cell embeddings of the Xpressor layers are then compressed with VAEs with respective latent for the [cell_type, tissue, age, sex, disease, sequencer, ethnicity, organism, cell culture, additional] classes of: 64, 32, 8, 2, 16, 8, 8, 8, 2, None (no VAE)

Class Decoders

The class decoders are MLPs applied to compressed representations of their respective VAEs, with a shape of $[\mu_c, 256, n_c]$ with n_c the number of labels in the class c and μ_c the dimension of this class for the VAE.

Expression Decoder

We had noticed that scPRINT-1 initially produced embeddings that could be biased by the cell-depth token. We thus push scPRINT-2 to be depth-invariant by introducing the sequencing depth information only in the Expression Decoder, ensuring that the output

gene-cell tokens contain little absolute sequencing depth information (see Figure 3.3, see Supplementary Figure 5.28). This debiases cell embedding to depth data and also improves denoising (see Table 3.1).

The expression decoder thus gets applied to the output gene embeddings and also receives the log2p1-transformed sequencing depth (also called total cell expression count) c and is of the form:

$$\mu, \theta, \pi = MLP(e||||c) \tag{3.17}$$

The MLP is a two-layer neural network with dimensions [*d+1, d*, 3], where || denotes the concatenation operation.

The parameters μ, θ, π are the parameters of the ZINB and are used in the ZINB+MSE loss.

3.5.4 Pre-training

The three main tasks in the multi-task pre-training of scPRINT-2 are denoising, classification, and bottleneck learning. While the denoising loss enhances the model' s ability to find meaningful gene-gene connections, the other two try to make the model and its cell embedding representation more robust and cell-type-specific. The tasks are presented below.

Optimization method

Optimization is performed with fused ADAMW and a weight decay of 0.01. We observed a complete inability to learn when using the base ADAM algorithm, which has a similar weight decay schedule. This can be explained by a known inequivalence issue in ADAM[53].

We do not use the stochastic weight averaging[283] method during training.

During pre-training, the hyperparameters are set to a dropout of 0.1, a learning rate (LR) of 1e-4, and the precision is set to 16-mixed with residuals in fp32. We clip gradients to 10 and train over many sub-epochs of 20,000 training and 20,000 validation batches, with a warmup of 2,000 steps. Across epochs, we use a linear LR decrease of 0.6 with a patience of 2, and we stop training after 4 consecutive increases in validation loss. We initialize weights to a normal distribution around 1, biases to 0, and biases for the final layer of the Classifiers to -0.12.

Our batch size is 128, and we use a pre-norm strategy for the transformer with a linearly increasing stochastic depth dropout rate of 0.02 per layer. We use a noise parameter of 70%. We split the cells in the datasets into 98% for training and 2% for validation, and reserve at least 2% of the split datasets for testing. Our reconstruction loss is ZINB+MSE (see the ZINB+MSE section in Methods).

While many pre-training variants can be selected from contrastive learning, classification,

adversarial classification, compression (with XPressor and VAE), masking, biased masking, and imputation, the choice may depend on specific biological assumptions.

scPRINT-2 is trained with denoising an input cell profile, given its nearest neighbor's expression.

Given the same information, it also performs label prediction during pre-training for: cell type, disease, sequencer, age, tissue, ethnicity, sex, cell culture, and organism. The classification task is performed jointly with the denoising task, meaning that labels are predicted from corrupted expression data and from nearest-neighbor expression information. The hierarchical classifier is applied to the VAEs' latent embeddings.

During decoding, it regenerates the expression profile for all input genes, including those dropped during variable context selection. This effectively does gene imputation.

The decoder receives only the gene location and ESM3 embedding and performs cross-attention on cell embeddings. The cell embeddings are the output of the VAEs and Xpressor layers, so the input is:

$$X_i = [g_0 + l_0, g_1 + l_1, ...] \tag{3.18}$$

And cell-embeddings are:

$$C_i = [c_{i0}, c_{i1}, ...] = \cup_j VAE_j(u_{ij})u_{ij} \in U_j \tag{3.19}$$

With U_j the matrix output of Xpressor.

Finally, Embedding independence and Elastic Cell similarity losses are applied to the cell embeddings C_i for all cells i in the minibatch.

Database and sampling

The scPRINT-2 pre-training corpus is composed of all listed databases with weighted random sampling over all predicted labels, together with cluster-weighted sampling to compensate for missing cell-type labels in the Arc's scBasecount database.

Practically, during training, we apply a curriculum learning strategy whereby the S_1 factor slowly increases from 1 to 1000, letting the model initially learn across the diversity of cells and slowly retrieve the true cell state and modality distribution. We also apply depth-weighted cell sampling to each cell group (see the cluster-weighted sampling section in Methods).

Denoising pre-training task

We downsample an expression profile using a zero-inflated Poisson model of the data, following the approach in Kalfon et al. With this formulation, on average, half of the

counts to be dropped are removed by randomly selecting some reads per gene, sampled from a Poisson distribution with a lambda parameter proportional to the gene' s count. The remaining half of the counts to be dropped are dropped by randomly setting some genes to 0, i.e., complete dropout of those genes. It is to be noted that, with this definition of downsampling, the exact average number of counts dropped in both parts depends slightly on the dropout *rrate*. During our pre-training, r is set to 0.7, meaning, on average, 35% of the transcript counts are dropped per cell.

Let x_i be the gene expression vector of cell i with dimensions n_{genes}; we create a downsampled *version* by doing:

$$\widehat{x_i} = max((x_i - p_i) \cdot \pi_i, 0) \tag{3.20}$$

with:

- $m \sim Uniform(0, r)$ the noise level
- $p_i \sim Poisson(x_i \times r \times 0.55)$ a vector of size n_{genes} where the Poisson is sampled for each element x_i of x
- $\pi_i = I(u \geq r \times 0.55)$ a vector of size n_{genes} , the binary mask vector indicating non-dropout genes.
- $u_i \sim Uniform(0, 1)$, a vector of size n_{genes}, of random values drawn from a uniform distribution.
- $\cdot$ denotes the element-wise multiplication.
- r being the dropout amount. We scale it by a tuning hyperparameter of 0.55 instead of 0.5 for numerical reasons.

We uniformly sample a value between 0 and 0.8 for our r, per GPU, during training of scPRINT-2 and other additive models based on denoising, except if noted otherwise.

For the GNN-encoder, we add a second "denoising" step in which we set the noise to 1 and set all expressions to 0 for the center cell. This required the model to predict its expression from the expressions of its neighbors in expression space on the same dataset.

Bottleneck learning pre-training task

During training, we predict gene expression at both the decoder output and the scPRINT-2⟶Expressor⟶scPRINT-2 pipeline outputs, following the XPressor approach in Kalfon et al.

During training, 20% of the time, scPRINT-2 drops between 0 and 2800 genes from its input context per GPU. This pushes the model to learn across a variety of context lengths, it also makes the contrastive loss more robust. Finally, at the output of the decoding step in

the bottleneck learning part, the model always predicts across the full 3200 genes, effectively performing imputation during pre-training.

When cross-GPU training is performed, cell-embedding-level losses are computed across all GPUs.

Classification pre-training task

The Classification task follows the new hierarchical classifier presented in Methods and adds two novel classes: patient age and tissue of origin.

Loss aggregation

The losses are aggregated as follows:

$$L = L_{ZINB+MSE} + L_{class} + 0.2L_{similarity} + 0.3L_{independance} + 0.2L_{contrastive} + 0.001L_{KL} \quad (3.21)$$

The $L_{ZINB+MSE}$ is effectively added 4 times, for the reconstruction post perturbations with denoise of 0.8, 1.0, TF-masking, and post bottleneck learning.

3.5.5 Fine-tuning Task

Our fine-tuning (see Results section 2 and 4) reuses the classification, bottleneck learning, and VAE (KL) loss of our pre-training for 4 epochs with a learning rate of 0.0001. For batch correction and organism integration, we add the MMD loss between samples from batches 1 and 2 within each minibatch[260, 261]. At the same time, an effective MMD loss requires minibatches that are large enough to include a good mix of both label types and cannot accommodate many labels.

With the MMD loss defined as:

$$\mathrm{MMD}^2(X, Y) = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{i'=1}^{m} k(x_i, x_{i'}) + \frac{1}{n^2}\sum_{j=1}^{n}\sum_{j'=1}^{n} k(y_j, y_{j'}) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i, y_j) \quad (3.22)$$

For a finite set of elements from distribution source X and Y, where we use the energy distance kernel:

- $k(x, y) = -|x - y|$

When more than 2 domains exist, we compute MMD between each domain and the remaining domains.

All analyses are defined in the notebook: notebooks/scPRINT-2-repro-notebooks/fine_tuning_cross_species_emb_mmd.ipynb

3.5.6 Classification task

For our classifications tasks (see Results section 2), we use the F1-accuracy as our primary metric. When computing it across hierarchical classes, we consider parental relationships to ensure that even if a more precise cell type is predicted than the ground truth, it remains valid. For example, given a ground truth label of *neuron,* a predicted label of *excitatory neuron* will be considered correct.

If "unknowns" exist in the ground truth or the prediction, they are discarded from the metric.

The cross-organism generalization classification dataset was extracted from the supplementary datasets of the paper titled "Benchmarking cross-organism single-cell RNA-seq data integration methods: towards a cell type tree of life"[243] available at Figshare[3].

The context-increase classification analysis was performed on the "human multiple cortical areas"[246] Smart-seq v4 dataset available at cellxgene[4]. For each NNZ gene level, we used only cells with at least that many genes expressed. We did not apply the same logic to the second version and used a smaller dataset, so the impact of zero-expressed genes in context could be more clearly seen.

All analyses are defined in the notebooks:

- notebooks/scPRINT-2-repro-notebooks/cross_species_embedding.ipynb
- notebooks/scPRINT-2-repro-notebooks/smart_seq_class.ipynb
- notebooks/scPRINT-2-repro-notebooks/unknown_species_classification.ipynb
- notebooks/scPRINT-2-repro-notebooks/large_dataset_anallysis.ipynb
- figures/nice_umap.py
- notebooks/scPRINT-2-repro-notebooks/batch_corr_op ft.ipynb
- notebooks/scPRINT-2-repro-notebooks/batch_corr_op v1.ipynb
- notebooks/scPRINT-2-repro-notebooks/batch_corr_op.ipynb
- notebooks/scPRINT-2-repro-notebooks/plot.ipynb

[3] https://figshare.com/s/6187811b6c3fae02a4d3?file=50608386
[4] https://datasets.cellxgene.cziscience.com/a1d40c84-c81c-406f-bef4-e25edeb651e5.h5ad

Logits refinement (Laplacian smoothing)

We apply logits smoothing at inference by computing the k-nearest neighbors of each cell and their distances, listed in the squared sparse matrix D, and solving for:

$$\tilde{P} = \arg\min_{P} \|P - P_0\|_F^2 + \lambda \mathrm{Tr}(P^T L P) \tag{3.23}$$

Where P_0 are the initial logits, L is the graph Laplacian, and λ controls the strength of regularization where P_0 are the initial logits, L is the graph Laplacian, and λ controls the strength of regularization with default value $\lambda = 0.1$. In our case, we set $K = 6$ and $L = D + D^T - C$, where C is the diagonal degree matrix of $D + D^T$.

The solution has a closed form:

$$\tilde{P} = (I + \lambda L)^{-1} P_0 \tag{3.24}$$

Cluster-aggregation

We compute the per-cluster logits aggregation by first clustering the test dataset and then taking the maximum logits across all cells in each cluster as the label for that cluster. Solving for:

$$p_c = \arg\max_{i} \left(\max_{j} (l_{i,j}) \right) \tag{3.25}$$

where $l_{i,j} \in L_c$ are the logits for class j across all cells i in cluster c, and p_c is the prediction for cluster c. cluster C.

and p_c the prediction for cluster C.

3.5.7 Denoising task

The denoising benchmark (see Results section 3) was performed on eight datasets of varying quality, assessed by the number of non-zero genes (NNZ), sequencing depth, and the distribution of gene counts.

We compute denoising as the dataset-wise percentage improvement in correlation over the 5000 most variable genes, considering only genes that are non-zero in the ground-truth.

Here is the dataset list (all cellxgene datasets available at https://datasets.cellxgene.cziscience.com/):

- retina: 53bd4177-79c6-40c8-b84d-ff300dcf1b5b.h5ad

- kidney: 01bc7039-961f-4c24-b407-d535a2a7ba2c.h5ad
- pancreas: https://figshare.com/ndownloader/files/24539828
- intestine: d9a99b4a-3755-47c4-8eb5-09821ffbde17.h5ad
- glio_smart_highdepth: 6ec440b4-542a-4022-ac01-56f812e25593.h5ad
- lung_smart: 6ebba0e0-a159-406f-8095-451115673a2c.h5ad

human from scbasecount ID: SRX24486462 and SRX22526970

All analyses are defined in the notebook:

notebooks/scPRINT-2-repro-notebooks/denoising_V2.ipynb

3.5.8 Xenium analysis

We apply the Xenium analysis on the FFPE Human Skin Primary Dermal Melanoma with 5K Human Pan Tissue and Pathways Panel found on the 10X genomics platform under:

https://www.10xgenomics.com/datasets/xenium-prime-ffpe-human-skin

Information on the dataset and its preprocessing can be found on the same webpage.

We extract a dense patch that covers 30% of the cells in the dataset, on which we perform all our analyses (see Results section 3).

All analyses are defined in the notebooks:

notebooks/scPRINT-2-repro-notebooks/xenium_analysis.ipynb

3.5.9 Embedding task

We perform the organism-level integration task on the same two datasets listed above from the "Benchmarking cross-organism single-cell RNA-seq data integration methods: towards a cell type tree of life" paper, using the scIB metrics and the same ground truth labels (see Results section 4).

All analyses are defined in the notebooks:

- notebooks/scPRINT-2-repro-notebooks/cross_species_embedding.ipynb
- notebooks/scPRINT-2-repro-notebooks/generative_modelling.ipynb
- notebooks/scPRINT-2-repro-notebooks/batch_corr_op ft.ipynb
- notebooks/scPRINT-2-repro-notebooks/batch_corr_op v1.ipynb
- notebooks/scPRINT-2-repro-notebooks/batch_corr_op.ipynb
- notebooks/scPRINT-2-repro-notebooks/plot.ipynb

3.5.10 Generative task

We perform the generative tasks on two human/mouse datasets extracted from the supplementary datasets of the paper titled "Benchmarking cross-organism single-cell RNA-seq data integration methods: towards a cell type tree of life" (see Results section 4).

We generate cell-embeddings for all mouse cells, giving us a matrix M of size $[10, n_{cell}, d_{emb}]$. We then retrieve an average human organism embedding by using 2000 randomly selected human cells and averaging their organism cell-embedding, resulting in a vector v of size $[d_{emb}]$. We then regenerate an expression profile using the mouse cell-embeddings and the human average organism embedding by replacing it in the matrix like so: $M[:, i, :] = v$.

We then apply the decoder part of scPRINT, which performs cross-attention over the matrix M and takes the human gene embeddings as input tokens.

All analyses are defined in the notebook:

scRNA-seq datasets distances

To compute our distance metric across two scRNA-seq datasets, we first identify the 5000 most variable genes that are also orthologous between the datasets. We use human and mouse data because orthology was readily accessible and well-defined.

We then compute the W2-distance directly on the raw mouse counts, the humanized counts predicted by scPRINT-2, and the human counts. We do not expect a zero or near-zero W2 distance between the humanized mouse data and the human data, as the number of cells, the types of cell, and their composition differ between the two datasets. We perform a similar analysis of the male-to-female conversion.

Over-representation measure

For the overrepresentation analysis and plot, we work on the ordered differential expression gene lists for both human-to-mouse and humanized-mouse-to-mouse, and similarly for male-to-female conversion. We compare the overlap in genes between the two lists at all possible cutoff values from 1 to 5000 to obtain our curve and, therefore, define scores.

3.5.11 Assessment of gene output embeddings

We assess scPRINT-2's gene output embeddings by computing output gene embedding oe a random *vascular lymphangioblast* cell from the glioblastoma Smart-seq-v2 dataset using its 5000 most expressed genes in that cell type. We then cluster it using the Leiden algorithm and, for each clustered group of genes, compute the number of pathways enriched using the "KEGG_2021_Human", "GO_Molecular_Function_2025", "WikiPathways_2024_Human",

and "GO_Cellular_Component_2025" gene set databases. Doing this for both XPressor and non-XPressor architectures, we then compute a t-test between the two sets of numbers.

All analyses are defined in the notebooks:

notebooks/scPRINT-2-repro-notebooks/output_embeddings.ipynb

3.5.12 Extracting meta-cell gene networks from attention matrices

in scPRINT

Transformers compute multiple attention matrices per layer, called attention heads. This is done by splitting the generated K, Q, and V embedding into m sub-embeddings, thus defining m attention heads. Each attention head computes the attention matrix via the equation:

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \tag{3.26}$$

However, we want to aggregate those across multiple cells with similar cell states to increase the signal from a single cell. We are doing so by averaging the Keys and Queries embeddings over the set of cells U passed to the model:

$$\text{softmax}\left(\frac{\text{mean}_U(Q) \cdot \text{mean}_U(K)^T}{\sqrt{d}}\right) \tag{3.27}$$

By doing this, the attention matrix behaves as if each query vector for cell i were "looking" across the key vectors of all the cells in U. The resulting object is a row-wise normalized n^*n matrix, where n is the size of the input context (i.e., the number of genes passed to the model).

in scPRINT-2

In scPRINT-2, we found, after in-depth review, that while the solution from equation (25) allows for faster computation of much larger gene networks from attention matrices, it also decreases accuracy. We thus instead directly took:

$$\text{mean}_U\left(\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)\right) \tag{3.28}$$

However, to prevent adding QK from genes that are not expressed in the given cell, we generate Qs and Ks from forward passes using only the expressed genes in each cell (see "using only expressed genes" in additive benchmark). This has the benefit of biasing the gene network towards genes that are co-expressed in the set of cells we are computing it on.

This means that for a list of n genes, each cell will have a subset of m Qs,Ks. We thus take the average of the set, computing the mean per gene by counting how many times each gene was expressed across the set of cells.

plotting gene sub-networks

To plot a subset of our gene networks, we choose a seed gene and get all its top-K connected nodes. We then overlay the top-N edges in this sub-network, ordered by connection strength. Here K=15 and N=50

3.5.13 Gene network task

We generated gene networks from notebooks: https://figshare.com/s/618...

We used a matched cross-tissue human and mouse dataset from Zhong et al.[243]

We computed the network across all 10 cell types that were common to both human and mouse in the dataset, using the 4000 most variable genes within each cell type, with a maximum of 1024 cells (see Results section 5).

All analyses are defined in the notebooks:

- notebooks/scPRINT-2-repro-notebooks/gene_networks.ipynb
- notebooks/scPRINT-2-repro-notebooks/gene_networks_var_2.ipynb

The Cellmap Ground truth

We used the Cellmap dataset available at https://ndexbio.org under uuid f693137a-d2d7-11ef-8e41-005056ae3c32.

It has a total of 36842 connections across 7543 genes, mainly computed from protein-binding data of AP-MS experiments in the O2US cell line[267].

The Collectri and Omnipath Ground truth

We used the Collectri ground truth from the Decoupler: https://github.com/scverse/decoupler package and the Omnipath ground truth from the Omnipath package[284, 129]: https://github.com/saezlab/omnipath, both accessible with given versions within the BenGRN package: https://github.com/jkobject/benGRN.

The human interactome Ground truth

We use the RF2-PPI predicted network available at https://conglab.swmed.edu/humanPPI/. We set a cutoff of 0.4 for the benchmark and 0.7 for the high-quality (hq) network[285].

3.5.14 Gene network metrics

We use the packages benGRN and GRnnData released with this manuscript to work With Gene networks and perform our benchmarks (see Results section 5).

Our two main metrics are OR and AUPRC. They all take advantage of the fact that the predictions are generated as scores over edges between nodes:

- We have computed the diagnostic odds ratio (OR) as (TP x TN) / (FP x FN) at the cutoff score that yields K positive predictions, where K is the number of positive elements in the ground truth.
  In this context, 1 represents a random prediction, and inf represents a perfect prediction; values below one indicate that inverting the predictor would yield better results.
- Area Under the Precision-Recall Curve (AUPRC) is the area (computed with the composite trapezoidal rule) under the curve defined by the precision ($PR = TP / (TP + FP)$) and recall ($RE = TP / (TP + FN)$), where TP is the number of true positives. FP is the number of false positives. FN is the number of false negatives. This curve is obtained by varying the cutoff from 0 predicted positives to all predicted positives. Here, we compute a version of the AUPRC where the floor of the area is not given by the Precision=0 line but by the prevalence line of the positive class. Moreover, we do not interpolate the curve between the last recall value and the perfect recall: 1. We do this to properly compare AUPRC values across benchmarks and models. Random precision values are given in the supplementary data.

3.5.15 Open Problem benchmarks

We ran all the open-problem benchmark datasets for scPRINT-2 and scPRINT-1 on a local machine, following the instructions at https://openproblems.bio/documentation. We used the same datasets and labels available at: s3://openproblems-data/resources/ (see Results sections 2 and 4). We used the non-transformed count matrices as input. We used the same metrics for classification, the same scIB package version, and the same train-test splits as in the latest run of Open Problems. All other scores displayed are directly copied from that latest run.

On Open-problems, scIB's batch correction score is equal to (avgBatch + 1.5*avgBio)/2.5, which are themselves averages over many scores. Details of each value are available in our package's notebooks[144].

- scIB avgBio is a combination of label-based and label-free metrics, using, for example, the Adjusted Rand Index (ARI)[286] and the Normalized Mutual Information (NMI)[287] on clusters computed from the K-Nearest Neighbor graph. Other scores are used, some based on the conservation of trajectories and cell-cycle variance, others on the conservation of rare-cell populations, the overlap of highly variable genes (see scIB[144]), and more.

- scIB avgBatch is a similar combination of label-based and label-free metrics, using, for example, the average connectivity across clusters of different batches: ASW[288], the graph integration local inverse Simpson's Index: graph iLISI[289], the k-nearest-neighbor Batch Effect Test (kBET)[288], and more.

Finally, we also use two metrics in our classification task:

- Macro-F1: also called macro-average, is the average of the F1 score across each class in a multi-class task, where the F1 score is: $2 \times \frac{PR*RE}{PR+RE}$.

- Accuracy: is computed as $\frac{TP+TN}{TP+TN+FN+FP}$

We did not run on two datasets of Open Problems: immune_cell_atlas & tabula_sapiens, as their sizes were too large for us to run scib on any of our available machines.

Moreover, while we believe it is the same for other foundation models assessed in this benchmark, most of these datasets are part of the pre-training corpus of scPRINT. Therefore, the "zero-shot" performance claims, especially classification, should be viewed in this context.

Finally, Open Problem is a living benchmark. Methods, Results, datasets, and metrics will likely change as the scores are continuously updated. We hereby present our results as they were in the 12th of November 2025.

3.6 Data availability

The model weights are publicly available on HuggingFace under: https://huggingface.co/jkobject. Pre-training logs to assess the model's training are publicly available in weights and biases[5].
The embeddings and classification results over the 350 million cells are available under the public google bucket: gs://scprint2/. The interactive viewer for a subset of these cells is available at https://cantinilab.github.io/scPRINT-2/.
The pre-training dataset is publicly available on CellxGene: https://cellxgene.cziscience.com/, under its census data release version: LTS 2024-07-01, Tahoe and ARC's scBasecount are available on https://github.com/ArcInstitute/arc-virtual-cell-atlas, commit version 68da110. All other datasets used in this work can be downloaded from their respective public databases using the helper scripts in the scPRINT, BenGRN, GRnnData, and scDataLoader packages. Source data is provided with this paper to re-generate the figures. Code to download the input dataset, generate the source data, and the figures are available as a notebook in https://github.com/cantinilab/scPRINT-2. Source data are provided with this paper.

[5] https://wandb.ai/ml4ig/scprint_ablation/reports/scPRINT-2-additive-benchmark--VmlldzoxNTIyOTYwNA

3.7 Code availability

The code and notebooks used to develop the model, perform the analyses, and generate results in this study are publicly available and have been deposited in cantinilab/scPRINT-2 at https://github.com/cantinilab/scPRINT-2 under GPLv3 license. The specific version of the code associated with this publication is archived in the same repository under the tag 1.0.0 and is accessible via https://github.com/cantinilab/scPRINT-2/tree/1.0.0/ and DOI:10.5281/zenodo.

Additional packages for this analysis are defined in the pyproject file and project submodules. Together with packages developed by us:

- GrnnData: https://github.com/cantinilab/GRnnData DOI:10.5281/zenodo.10573141
- BenGRN: https://github.com/jkobject/benGRN DOI:10.5281/zenodo.10573209
- scDataLoader: https://github.com/jkobject/scDataLoader DOI:10.5281/zenodo.10573143

3.8 Acknowledgment

The project leading to this manuscript has received funding from the Inception program (Investissement d'Avenir grant ANR-16-CONV-0005) L.C., and the European Union (ERC StG, MULTIview-CELL, 101115618) L.C.. We acknowledge the help of the HPC Core Facility of the Institut Pasteur and Déborah Philipps for the administrative support. L.C..

The work of G. Peyré was supported by the French government under the management of Agence Nationale de la Recherche as part of the 'Investissements d'avenir' program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute) G.P..
Figure 1B, 1C, 3A, 4D, 4E, 4F, 5G and supplementary Figure S9, S13 used icons by Servier https://smart.servier.com/ is licensed under CC-BY 3.0 Unported https://creativecommons.org/licenses/by/3.0/ NIAID Visual & Medical Arts. RNA. NIAID BIOART Source. bioart.niaid.nih.gov/bioart/452. DBCLS https://togotv.dbcls.jp/en/pics.html is licensed under CC-BY 4.0 International https://creativecommons.org/licenses/by/4.0/. Marcel Tisch https://twitter.com/MarcelTisch is licensed under CC-0 1.0 Universal https://creativecommons.org/publicdomain/zero/1.0/. *Library v1.1. Available via Zenodo (https://zenodo.org/records/17229908).*

3.9 Author Contribution

J.K., L.C., and G.P. designed the study. J.K. developed the tool and performed all the analysis. J.K., and L.C wrote the manuscript. G.P. revised the manuscript.

Discussion and perspectives

By the end of this thesis, we have developed next-generation foundation models for single-cell data and established new methodologies for their evaluation and application. This work began with the goal of improving GRN inference from scRNA-seq data while understanding how single-cell foundation models work. Through three chapters, we systematically addressed fundamental challenges in the field, from architectural design to cross-scale learning and rigorous benchmarking.

scPRINT: Establishing foundations for gene network inference. In our first chapter, we demonstrated that transformer-based foundation models pretrained on large-scale single-cell data can extract meaningful gene regulatory networks without requiring graph-structured inputs. Training on 50 million cells, we showed that several key innovations were critical: protein-based gene encodings using ESM2 embeddings enabled cross-species generalization while reducing parameters; learned expression tokenization via MLPs outperformed hand-crafted binning; and genomic positional encodings captured co-regulation patterns. We learned that the transformer's attention matrices, when properly extracted and filtered via head selection mechanisms, could recover cell-type-specific regulatory interactions that outperformed traditional methods such as GENIE3 on the Omnipath benchmark. [16]

Given a foundation model's pretraining and losses, it cannot be expected to contain much more than co-expression patterns. Comparing to a complex co-expression model like GENIE3, foundation models often indeed underperform them. However, we show that we can reach the best co-expression based gene network inference tools by using better training. Finally, the addition of inductive biases like the gene's sequence and evolutionary similarities can help provide information unavailable in the expression data alone.

But perhaps most importantly, we learned that the lack of standardized evaluation was hindering progress in the field. Our creation of BenGRN and GRnnData addressed this gap, enabling fair comparisons across methods using multiple ground truths, from literature-based networks to cell-type-specific ChIP-seq/perturb-seq intersections and genome-wide perturb-seq data. [34] We showed that scPRINT not only excelled at network inference but also achieved competitive zero-shot performance on orthogonal tasks like cell type annotation, batch correction, and denoising, demonstrating that these capabilities emerge naturally from learning good cellular representations. The application to prostate tissue revealed early TME markers in rare B-cells and differential hub genes in BPH-associated fibroblasts, validating the model's utility for biological discovery. This chapter led to a

publication in Nature Communications.

Xpressor: Learning across biological scales. Our second chapter addressed a fundamental limitation: foundation models at different biological scales (molecules, sequences, cells, tissues) operate in isolation, unable to leverage information flow between scales. We discovered that cross-attention-based compression mechanisms could effectively transform high-dimensional gene-level representations into lower-dimensional cell-state vectors while preserving biological information. The Xpressor architecture, grounded in information bottleneck theory [290], demonstrated that explicit compression and decompression through learned latent tokens outperformed simple pooling strategies, improving cell-type prediction and embedding quality.

More significantly, we showed that lower-scale models could be enriched through fine-tuning on upper-scale tasks. By adding trainable MLP adapters to ESM2 protein embeddings during scPRINT pretraining, we demonstrated that protein sequence representations could be augmented with co-expression information learned from millions of single-cell profiles. This multi-scale fine-tuning improved cell-type prediction and gene network inference compared to frozen embeddings. We learned that each biological scale's vocabulary can be viewed as a compressed representation of the scale below, and that explicit architectural support for this compression—rather than ad hoc concatenation—is essential for effective cross-scale learning. This chapter led to a poster at the ICML 2025 F4MLS workshop and is currently under a second review for publication in Bioinformatics Advances.

scPRINT-2: Systematic validation and next-generation capabilities. In our third chapter, we addressed the field's critical gap in understanding which design decisions matter most. Using an unprecedented additive benchmarking framework that evaluates 42 model variants, some across multiple seeds, we systematically quantified the impact of each architectural choice. We learned that denoising outperforms masking as a pretraining task; un-normalized expression is superior to normalized input; ESM-based gene tokens significantly outperform learned embeddings; genomic location encoding accelerates convergence; and model size correlates strongly with performance on complex tasks like network inference.

Training on 350 million cells from 16 organisms — the largest single-cell corpus assembled to date — we learned that data diversity and quality matter more than sheer quantity. Reducing to 200 human datasets resulted in no change in performance, whereas using only low-diversity datasets led to a significant drop in performance. We showed that cluster-weighted and NNZ-weighted sampling strategies were essential for training efficiently on such heterogeneous data. The incorporation of 12 validated innovations, including the XPressor architecture, GNN-based expression encoders, and criss-cross attention, resulted in state-of-the-art performance in zero-shot cell-type classification on Open Problems (surpassing all existing methods), superior expression denoising across all contexts, and best-in-class batch integration.

Critically, we demonstrated that scPRINT-2 generalizes beyond its training distribution. On Xenium spatial transcriptomics (an unseen modality), it successfully denoised expression, imputed 5,000 unseen genes, and produced meaningful label predictions. On cat and tiger tissues (unseen organisms), it achieved sufficient accuracy across 500 cell types to correct

expert annotations. The XPressor architecture enabled counterfactual generation, allowing us to "humanize" mouse expression profiles with strong enrichment in the true differentially expressed genes. We showed, for the first time, that output gene embeddings were enriched for biological pathways rather than mere expression values. This chapter led to a preprint, currently under review in Nature Methods.

Key learnings and paradigm shifts. Across these three chapters, several overarching insights emerged. First, we learned that GNNs, despite initial promise, were not the optimal approach for single-cell data due to the absence of known graph structures and poor scaling properties. Transformers, which can be viewed as GNNs on fully connected graphs, proved superior when architectural innovations addressed their quadratic computational complexity. Second, we learned that foundation model evaluation in genomics had been inadequate, relying on artificial benchmarks disconnected from biological reality and real-world applicability. Our systematic benchmarking revealed that many claimed capabilities of existing models were overstated or poorly validated. Third, we learned that accessibility and reproducibility are as important as model performance—releasing not just weights but also training code, datasets, tutorials, and documentation was essential for community adoption and scientific progress.

But much work remains at the crossroads of AI and biology. In the following sections, we will discuss the challenges and opportunities we see in this space.

4.10 Collecting data in the wild

The main challenge that remains to be solved for better single-cell foundation models is the lack of data. Indeed, while we have amassed hundreds of millions of cells from public datasets during this thesis, we have identified several limitations along three important axes: diversity and quality.

4.10.1 Genetic diversity

The first issue to address for a better model will be obtaining cell expression data across a much more diverse genetic background, meaning sequencing the genomes of the tissues under consideration, which is rarely done, as genomic data is subject to strict laws governing patient anonymity and public sharing.

Fortunately, we have seen projects beginning on this goal, such as 10K10K, which aims to sequence 10,000 cells from 10,000 people, along with genetic data [291]. The Sanger Institute is also doing similar work with spatial transcriptomics of tens of thousands of samples in development, along with their genomes.

But these remain small-scale projects compared to the diversity of life on Earth. In genomics, scBasecount [230], and other for-profit companies are working on sampling life around the world from barren places to ocean depths, with the stated goal of developing higher quality models that work across the phylogenetic tree of life [292]. Single-cell models

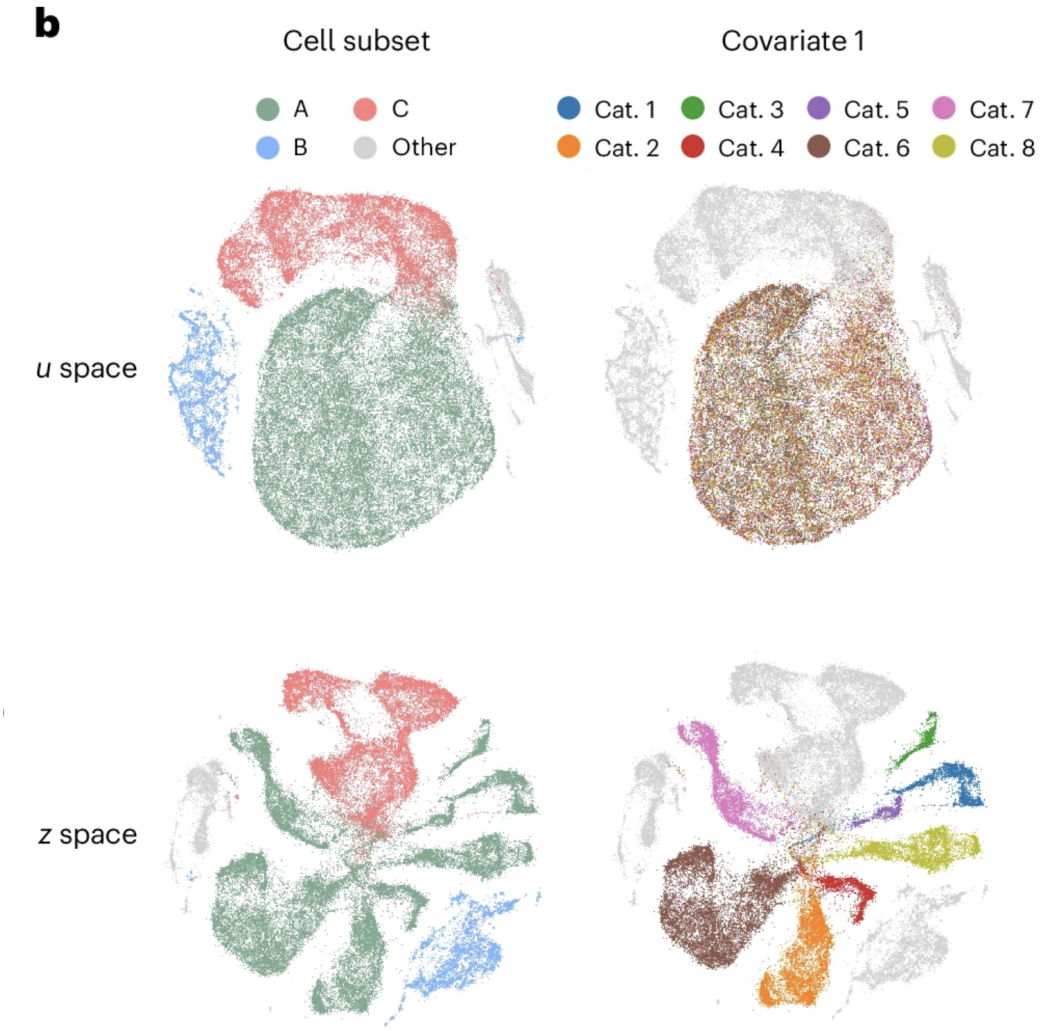


Figure 4.6: The genetic diversity in a single cell dataset highlighted with MR.VI. Adopted from Boyeau et al. [2]

would stand to benefit just as much in their own data modality.

4.10.2 Data quality

The second missing important axis is data quality. Single-cell genomics is plagued by very low-quality, noisy, biased, and poorly labeled datasets. It can be explained by the high cost of sequencing, the complex chemistry of the experiment, and the poor academic incentives driving the creation of other kinds of datasets.

It leads to an unstated Pareto front, where we need both greater depth and greater breadth: more diversity and higher quality.

New technologies might solve this issue. Indeed, we now have technologies like VASA-seq [293], 10X's Flex [294], and smart-seq 3 [295] that promise unparalleled resolution for a given sequencing budget. Indeed, the number of genes they can discover per cell, a measure

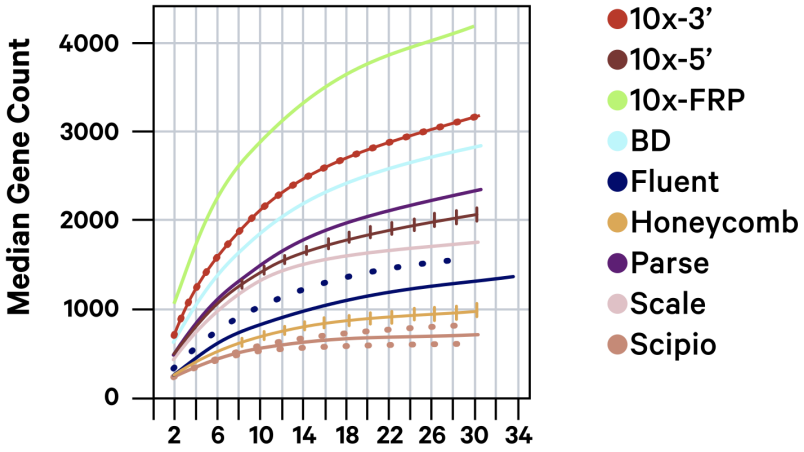


Figure 4.7: The number of genes detected per read for different single-cell technologies. X-axis shows read-depth (x1000). Adopted from Simone et al. [3].

of sequencing quality, increased by 50% compared to 10x v3.

Another issue is the set of cells that can be assessed and their fidelity to what was initially sent for sequencing. 10x's Xenium, BGI's STEREO-seq, and expansion-based in situ methods [210] are promising for sequencing RNAs in their original 2D or even 3D context within sub-cellular locations for millions of cells at once, without manipulating cells or creating droplets, thus addressing many issues of current technologies. But we will also have to be smarter in how we select cells to sequence.

4.11 Multi modality & perturbations

Indeed, these two modalities, spatial and dissociated single-cell, exist within a broad range of other single-cell modalities. Each one uncovers a different aspect of cellular biology, and integrating these diverse data types is crucial to understanding RNA biology itself [296]. One limitation of this thesis was its focus solely on scRNA-seq data. Other common modalities include single-cell ATAC-seq and surface protein sequencing.

Moreover, another axis of information concerns the temporal evolution of cells, with or without interventions such as perturbations. Again, mostly due to a lack of diverse, high-quality data at foundation model scale, we did not focus much on this axis either. But such data will be required for the model to learn causality, especially when assessed at multiple fine-grained timescales and in richer cellular models such as organoids. Currently, only imaging-based methods can achieve high throughput, low cost, and repeat measurement at the single-cell level to obtain information at scale [297]. However, many challenges remain, mostly in terms of indexing of cells across time, perturbations, and modalities.

But even solving for this, the search space remains unfathomable. It is not just the genes one would want to perturb; it is the hundreds of millions of locations that might create a specific effect. It is not just in one cell type, but in all the many possible cell states. It is not just one perturbation, but multiple at a time; not just one time point, but many; not just one

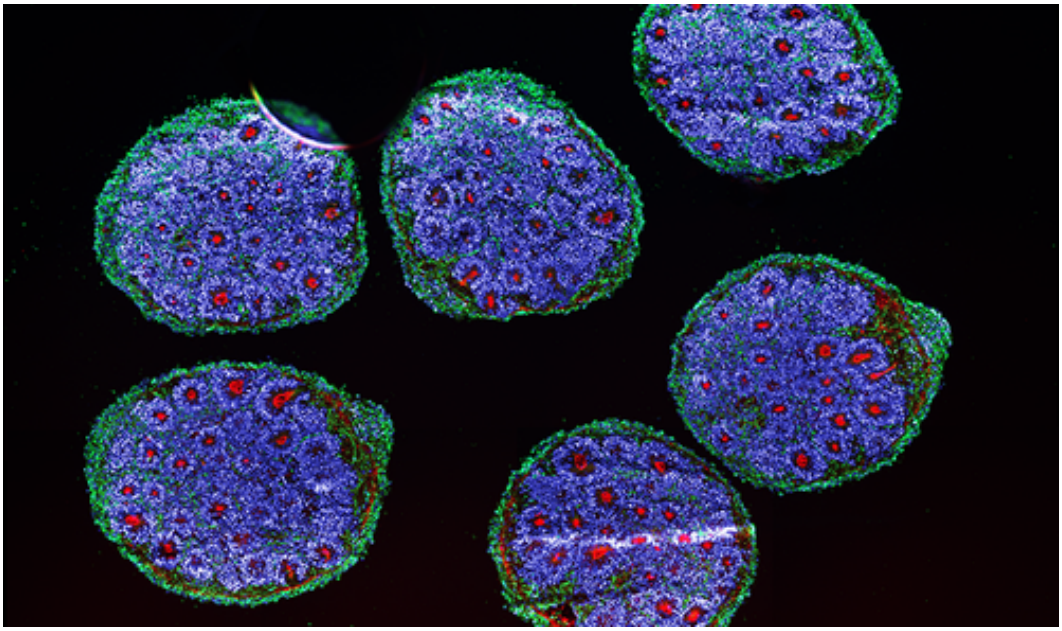


Figure 4.8: Image of brain organoids from the Broad Institute. Adopted from Faravelli et al. [4]

readout. Tools such as digital microfluidics [298] might help us solve some of these problems by indexing the search space. They would provide precise control over which cells receive specific perturbations and obtain particular readouts, rather than pooling experiments and sequencing budgets at random.

If paired with AI models and online active learning, we might someday have a shot at creating a true AI-virtual cell [299].

4.12 The AI virtual cell

We have seen only brief glimpses of the idea of AI-based virtual cell modeling in this thesis. But it is undeniably through the above-mentioned techniques that more work will be needed. Indeed, the goal of a virtual cell is to predict its state over time as it interacts with and is perturbed by its environment. With the advent of CRISPR-screens and drug-seq, the focus has shifted to predicting the effect of a drug on a dissociated cell after a given period of time, often through its survival, morphology, or expression profile. Many big projects have been launched, such as DepMap [300] and Recursion's RxRx [297]. More recently, the Arc Virtual Cell Challenge [5] also highlighted how limited our current technology and benchmarks are, with AI models achieving poor performance, limited usability of cross-lab data, and metrics that were easily gamed.

But most data is and will remain generated through the thousands of labs across the world, and most of the time, siloed. Training on these public noisy, static, and poorly labeled datasets is what we refer to as pretraining; this is what we focused on during this Thesis. The above-mentioned techniques could be seen as reinforcement learning with active feedback (RLAF). Such an AI-VC model would have been pretrained on most available biological data,

using foundation models of single-cell multimodalities, tissues, molecules, and protein-RNA-DNA sequences, pooled together using the kind of approaches we described in Chapter 2. LLMs could allow rich reasoning across these representations, results, and the breadth of written human knowledge [301].

Finally, the hope is that they will drive experiments in the lab and generate hypotheses that can be transferred from cheap in silico predictions to more expensive in vitro experiments on dissociated cells and organoids, and finally to in vivo models. This loop of pretraining, hypothesis generation, and reinforcement learning is what we call the lab-in-the-loop approach to cellular modeling (Figure 4.9).

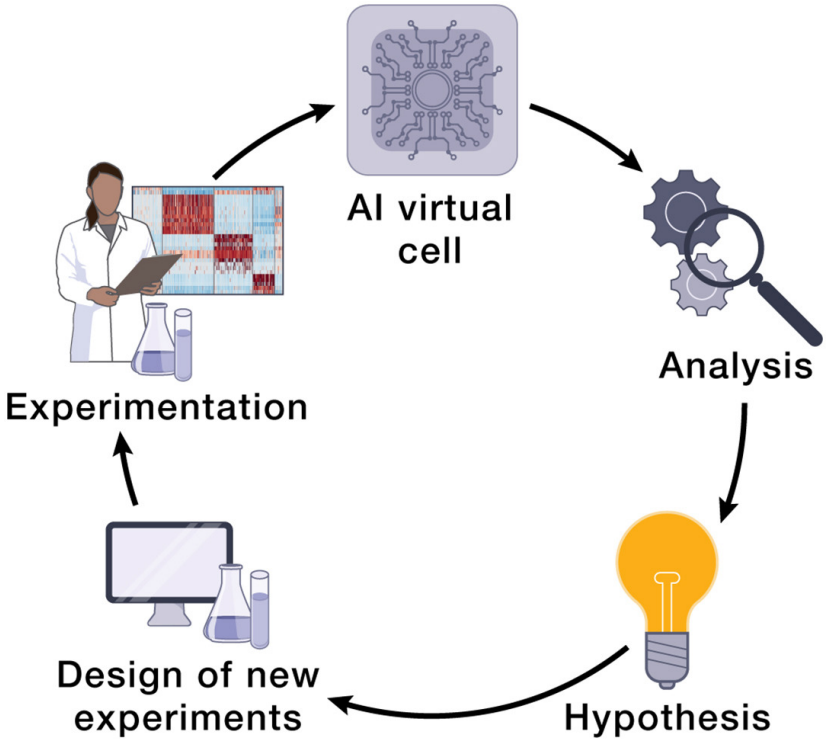


Figure 4.9: Overview of the lab in the loop approach to cellular modeling. Adopted from Bunne et al. [5].

Many challenges remain in bridging fields such as data engineering, machine learning, materials engineering, microelectronics, molecular biology, and cell biology, while improving the interpretability of models [302]. Still, the rewards are tremendous, as many diseases won't be solved by brute-force approaches or by targeting a single gene, whether in cancer or other complex, multicellular aging diseases.

Conclusion

Single-cell Foundation Models, while in their infancy, have the power to change the way we do biology and medicine.

During this Thesis, we have shown that:

- We can use the internal workings of scFMs to predict meaningful gene interactions.
- We can update their training tasks, data, and losses, as well as their architectures, to better capture the cell's underlying biology.
- We can use them to perform a variety of single-cell tasks in a zero-shot or few-shot manner, from cell annotations, denoising, imputation, embeddings generation, batch correction, cross-species integration, and counterfactual reasoning
- We can use multiple techniques at inference and fine-tuning time to improve their performance.
- We can leverage the other foundation models pretrained on other modalities to improve their performance.

In conclusion, the follow-up of these studies should enable broader adoption of single-cell transcriptomics in clinical applications, creating better benchmarks for models and, in turn, better models. It would be to integrate other modalities, such as sequences, epigenetics, proteomics, spatial, and imaging, via a multi-scale architecture and fine-tuning. But also allow these models to reason by integrating them into LLMs. Finally, one will need to gather more data from different species, patient contexts, and across perturbations. We aim to use active learning to guide the experiments and ultimately achieve the grand goal of cellular modeling.

Bibliography

[1] *CIS de l'institut Pasteur*. URL: https://www.pasteur.fr.

[2] Pierre Boyeau et al. "Deep Generative Modeling of Sample-Level Heterogeneity in Single-Cell Genomics". In: *Nature Methods* 22.11 (Nov. 2025), pp. 2264–2274. ISSN: 1548-7105. DOI: 10.1038/s41592-025-02808-x.

[3] Marco De Simone et al. *Comparative Analysis of Commercial Single-Cell RNA Sequencing Technologies*. June 2024. DOI: 10.1101/2024.06.18.599579.

[4] Irene Faravelli et al. "Brain Organoids: Tools for Understanding the Uniqueness and Individual Variability of the Human Brain". In: *Annual Review of Genomics and Human Genetics* 26.1 (Aug. 2025), pp. 299–320. ISSN: 1545-293X. DOI: 10.1146/annurev-genom-111522-014009.

[5] Charlotte Bunne et al. "How to Build the Virtual Cell with Artificial Intelligence: Priorities and Opportunities". In: *Cell* 187.25 (Dec. 2024), pp. 7045–7063. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2024.11.015.

[6] Winfried S. Peters. "The Cells of Robert Hooke: Wombs, Brains and Ammonites". In: *Notes and Records: the Royal Society Journal of the History of Science* (May 2024). ISSN: 0035-9149. DOI: 10.1098/rsnr.2023.0081.

[7] Bruce Alberts et al. "Cells and Genomes". In: *Molecular Biology of the Cell*. 6th ed. New York: Garland Science, 2015. Chap. 1.

[8] Inés Zugasti et al. "CAR-T Cell Therapy for Cancer: Current Challenges and Future Directions". In: *Signal Transduction and Targeted Therapy* 10.1 (July 2025), p. 210. ISSN: 2059-3635. DOI: 10.1038/s41392-025-02269-w.

[9] therese Winslow LLC. *CAR T-cell therapy*. 2017. URL: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/car-t-cell-therapy.

[10] Jamie A. Davies. *Synthetic Biology: A Very Short Introduction*. Oxford University Press, July 2018. ISBN: 978-0-19-880349-2. DOI: 10.1093/actrade/9780198803492.001.0001.

[11] David S. Goodsell. *cross-section through the internode region of a myelinated axon in the central nervous system*. DOI: 10.2210/rcsb_pdb/goodsell-gallery-030. URL: https://www.technologynetworks.com/tn/articles/molecular-science-and-art-with-professor-david-goodsell-368415.

[12] Agnes Ullmann. “Origins of Molecular Biology: A Tribute to Jacques Monod”. In: John Wiley & Sons, Ltd, 2003, pp. i–xxii. ISBN: 978-1-68367-216-6. DOI: 10.1128/9781555817763.fmatter.

[13] *François Jacob, Jacques Monod et André Lwoff discutant de la structure des protéines et de leurs éléments de symétrie, en s'aidant pour cela de cartes à jouer*. 1965.

[14] “The RNA World”. In: *Nature Structural & Molecular Biology* 31.5 (May 2024), pp. 729–729. ISSN: 1545-9985. DOI: 10.1038/s41594-024-01327-1.

[15] Ling-Ling Chen and V. Narry Kim. “Small and Long Non-Coding RNAs: Past, Present, and Future”. In: *Cell* 187.23 (Nov. 2024), pp. 6451–6485. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2024.10.024.

[16] Pau Badia-i-Mompel et al. “Gene Regulatory Network Inference in the Era of Single-Cell Multi-Omics”. In: *Nature Reviews Genetics* 24.11 (Nov. 2023), pp. 739–754. ISSN: 1471-0064. DOI: 10.1038/s41576-023-00618-5.

[17] F. Sanger, S. Nicklen, and A. R. Coulson. “DNA Sequencing with Chain-Terminating Inhibitors”. In: *Proceedings of the National Academy of Sciences of the United States of America* 74.12 (Dec. 1977), pp. 5463–5467. ISSN: 0027-8424. DOI: 10.1073/pnas.74.12.5463.

[18] *dna sequencing*. URL: https://theory.labster.com/dna_sequencing/.

[19] Taishan Hu et al. “Next-Generation Sequencing Technologies: An Overview”. In: *Human Immunology*. Next Generation Sequencing and Its Application to Medical Laboratory Immunology 82.11 (Nov. 2021), pp. 801–811. ISSN: 0198-8859. DOI: 10.1016/j.humimm.2021.02.012.

[20] 1000 Genomes Project Consortium et al. “A Global Reference for Human Genetic Variation”. In: *Nature* 526.7571 (Oct. 2015), pp. 68–74. ISSN: 1476-4687. DOI: 10.1038/nature15393.

[21] Eric S. Lander et al. “Initial Sequencing and Analysis of the Human Genome”. In: *Nature* 409.6822 (Feb. 2001), pp. 860–921. ISSN: 1476-4687. DOI: 10.1038/35057062.

[22] *Infinium Global Screening Array-48 Kit*. URL: https://www.illumina.com/products/by-type/microarray-kits/infinium-global-screening-48.html.

[23] Rory Stark, Marta Grzelak, and James Hadfield. “RNA Sequencing: The Teenage Years”. In: *Nature Reviews Genetics* 20.11 (Nov. 2019), pp. 631–656. ISSN: 1471-0064. DOI: 10.1038/s41576-019-0150-2.

[24] Jason Buenrostro et al. “ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide”. In: *Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.]* 109 (Jan. 2015), pp. 21.29.1–21.29.9. ISSN: 1934-3639. DOI: 10.1002/0471142727.mb2129s109.

[25] Evan Z. Macosko et al. “Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets”. In: *Cell* 161.5 (May 2015), pp. 1202–1214. ISSN: 1097-4172. DOI: 10.1016/j.cell.2015.05.002.

[26] Aviv Regev et al. “The Human Cell Atlas”. In: *eLife* 6 (), e27041. ISSN: 2050-084X. DOI: 10.7554/eLife.27041.

[27] *Visium Spatial*. URL: https://www.10xgenomics.com/support/spatial-gene-expression-ffpe.

[28] Patrik L. Ståhl et al. “Visualization and Analysis of Gene Expression in Tissue Sections by Spatial Transcriptomics”. In: *Science (New York, N.Y.)* 353.6294 (July 2016), pp. 78–82. ISSN: 1095-9203. DOI: 10.1126/science.aaf2403.

[29] Thomas Defard et al. *RNA2seg: A Generalist Model for Cell Segmentation in Image-Based Spatial Transcriptomics*. Mar. 2025. DOI: 10.1101/2025.03.03.641259.

[30] Xiuer Luo et al. “Advances in Protein Sequencing: Techniques, Challenges and Prospects”. In: *TrAC Trends in Analytical Chemistry* 191 (Oct. 2025), p. 118341. ISSN: 0165-9936. DOI: 10.1016/j.trac.2025.118341.

[31] Guillaume Balezo et al. *MIPHEI-ViT: Multiplex Immunofluorescence Prediction from H&E Images Using ViT Foundation Models*. May 2025. DOI: 10.48550/arXiv.2505.10294. arXiv: 2505.10294 [cs].

[32] Diya B. Joseph et al. “Single Cell Analysis of Mouse and Human Prostate Reveals Novel Fibroblasts with Specialized Distribution and Microenvironment Interactions”. In: *The Journal of pathology* 255.2 (Oct. 2021), pp. 141–154. ISSN: 0022-3417. DOI: 10.1002/path.5751.

[33] Lingjia Kong et al. “The Landscape of Immune Dysregulation in Crohn’s Disease Revealed through Single-Cell Transcriptomic Profiling in the Ileum and Colon”. In: *Immunity* 56.2 (Feb. 2023), 444–458.e5. ISSN: 1074-7613. DOI: 10.1016/j.immuni.2023.01.002.

[34] Atray Dixit et al. “Perturb-Seq: Dissecting Molecular Circuits with Scalable Single Cell RNA Profiling of Pooled Genetic Screens”. In: *Cell* 167.7 (Dec. 2016), 1853–1866.e17. ISSN: 0092-8674. DOI: 10.1016/j.cell.2016.11.038.

[35] Britt Adamson et al. “A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response”. In: *Cell* 167.7 (Dec. 2016), 1867–1882.e21. ISSN: 0092-8674. DOI: 10.1016/j.cell.2016.11.048.

[36] Alexander Dobin et al. “STAR: Ultrafast Universal RNA-seq Aligner”. In: *Bioinformatics* 29.1 (Jan. 2013), pp. 15–21. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts635.

[37] Laleh Haghverdi et al. “Batch Effects in Single-Cell RNA-sequencing Data Are Corrected by Matching Mutual Nearest Neighbors”. In: *Nature Biotechnology* 36.5 (May 2018), pp. 421–427. ISSN: 1546-1696. DOI: 10.1038/nbt.4091.

[38] Romain Lopez et al. “Deep Generative Modeling for Single-Cell Transcriptomics”. In: *Nature Methods* 15.12 (Dec. 2018), pp. 1053–1058. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0229-2.

[39] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. "Single-Cell RNA Sequencing Technologies and Bioinformatics Pipelines". In: *Experimental & Molecular Medicine* 50.8 (Aug. 2018), pp. 1–14. ISSN: 2092-6413. DOI: 10.1038/s12276-018-0071-8.

[40] Gökcen Eraslan et al. "Single-Cell RNA-seq Denoising Using a Deep Count Autoencoder". In: *Nature Communications* 10.1 (Jan. 2019), p. 390. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07931-2.

[41] David van Dijk et al. "Recovering Gene Interactions from Single-Cell Data Using Data Diffusion". In: *Cell* 174.3 (July 2018), 716–729.e27. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2018.05.061.

[42] Huidong Chen et al. "STREAM: Single-cell Trajectories Reconstruction, Exploration And Mapping of Omics Data". In: *bioRxiv* (Apr. 18, 2018). DOI: 10.1101/302554. URL: http://biorxiv.org/lookup/doi/10.1101/302554.

[43] Giovanni Palla et al. "Squidpy: A Scalable Framework for Spatial Omics Analysis". In: *Nature Methods* 19.2 (Feb. 2022), pp. 171–178. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01358-2.

[44] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. "scGen Predicts Single-Cell Perturbation Responses". In: *Nature Methods* 16.8 (Aug. 2019), pp. 715–721. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0494-8.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[46] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. Apr. 2022. DOI: 10.48550/arXiv.2112.10752. arXiv: 2112.10752 [cs].

[47] Kai Shen et al. *NaturalSpeech 2: Latent Diffusion Models Are Natural and Zero-Shot Speech and Singing Synthesizers*. May 2023. DOI: 10.48550/arXiv.2304.09116. arXiv: 2304.09116 [eess].

[48] Uriel Singer et al. *Make-A-Video: Text-to-Video Generation without Text-Video Data*. Sept. 2022. DOI: 10.48550/arXiv.2209.14792. arXiv: 2209.14792 [cs].

[49] Remi Lam et al. "Learning Skillful Medium-Range Global Weather Forecasting". In: *Science* 382.6677 (Dec. 2023), pp. 1416–1421. DOI: 10.1126/science.adi2336.

[50] Kaiming He et al. *Deep Residual Learning for Image Recognition*. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. arXiv: 1512.03385 [cs].

[51] John Jumper et al. "Highly Accurate Protein Structure Prediction with AlphaFold". In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2.

[52] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs].

[53] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. Jan. 2019. DOI: 10.48550/arXiv.1711.05101. arXiv: 1711.05101 [cs, math].

[54] Hao Li et al. *Visualizing the Loss Landscape of Neural Nets*. Nov. 2018. DOI: 10.48550/arXiv.1712.09913. arXiv: 1712.09913 [cs].

[55] Jason Wei et al. *Emergent Abilities of Large Language Models*. Oct. 2022. DOI: 10.48550/arXiv.2206.07682. arXiv: 2206.07682 [cs].

[56] Rishi Bommasani et al. *On the Opportunities and Risks of Foundation Models*. July 2022. DOI: 10.48550/arXiv.2108.07258. arXiv: 2108.07258 [cs].

[57] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. May 2019. DOI: 10.48550/arXiv.1810.04805. arXiv: 1810.04805 [cs].

[58] Tom B. Brown et al. *Language Models Are Few-Shot Learners*. July 2020. DOI: 10.48550/arXiv.2005.14165. arXiv: 2005.14165 [cs].

[59] Petar Veličković et al. *Graph Attention Networks*. Feb. 2018. DOI: 10.48550/arXiv.1710.10903. arXiv: 1710.10903 [stat].

[60] Josh Abramson et al. "Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3". In: *Nature* 630.8016 (June 2024), pp. 493–500. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07487-w.

[61] Iz Beltagy, Matthew E. Peters, and Arman Cohan. *Longformer: The Long-Document Transformer*. Dec. 2020. DOI: 10.48550/arXiv.2004.05150. arXiv: 2004.05150 [cs].

[62] Manzil Zaheer et al. *Big Bird: Transformers for Longer Sequences*. Jan. 2021. DOI: 10.48550/arXiv.2007.14062. arXiv: 2007.14062 [cs].

[63] Krzysztof Choromanski et al. *Rethinking Attention with Performers*. Nov. 2022. DOI: 10.48550/arXiv.2009.14794. arXiv: 2009.14794.

[64] Insu Han et al. *HyperAttention: Long-context Attention in Near-Linear Time*. Dec. 2023. DOI: 10.48550/arXiv.2310.05869. arXiv: 2310.05869 [cs].

[65] Nuno Gonçalves, Marcos Treviso, and André F. T. Martins. *AdaSplash: Adaptive Sparse Flash Attention*. June 2025. DOI: 10.48550/arXiv.2502.12082. arXiv: 2502.12082 [cs].

[66] Ashish Vaswani et al. *Attention Is All You Need*. Aug. 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs].

[67] Vân Anh Huynh-Thu et al. "Inferring Regulatory Networks from Expression Data Using Tree-Based Methods". In: *PLOS ONE* 5.9 (Sept. 2010), e12776. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0012776.

[68] Jalil Nourisa et al. *geneRNIB: a living benchmark for gene regulatory network inference*. 2025. DOI: 10.1101/2025.02.25.640181. eprint: 2025.02.25.640181.

[69] Daniele Mercatelli et al. "Gene Regulatory Network Inference Resources: A Practical Overview". In: *Biochimica Et Biophysica Acta. Gene Regulatory Mechanisms* 1863.6 (June 2020), p. 194430. ISSN: 1876-4320. DOI: 10.1016/j.bbagrm.2019.194430.

[70] Roshan Rao et al. *Transformer Protein Language Models Are Unsupervised Structure Learners*. Dec. 2020. DOI: 10.1101/2020.12.15.422761.

[71] Fan Yang et al. “scBERT as a Large-Scale Pretrained Deep Language Model for Cell Type Annotation of Single-Cell RNA-seq Data”. In: *Nature Machine Intelligence* 4.10 (Oct. 2022), pp. 852–866. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00534-z.

[72] Christina V. Theodoris et al. “Transfer Learning Enables Predictions in Network Biology”. In: *Nature* 618.7965 (June 2023), pp. 616–624. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06139-9.

[73] Haotian Cui et al. “scGPT: Toward Building a Foundation Model for Single-Cell Multi-Omics Using Generative AI”. In: *Nature Methods* (Feb. 2024), pp. 1–11. ISSN: 1548-7105. DOI: 10.1038/s41592-024-02201-0.

[74] Rebecca Boiarsky et al. *A Deep Dive into Single-Cell RNA Sequencing Foundation Models*. Oct. 2023. DOI: 10.1101/2023.10.19.563100.

[75] Abdel Rahman Alsabbagh et al. *Foundation Models Meet Imbalanced Single-Cell Data When Learning Cell Type Annotations*. Oct. 2023. DOI: 10.1101/2023.10.24.563625.

[76] Yanay Rosen et al. *Universal Cell Embeddings: A Foundation Model for Cell Biology*. Nov. 2023. DOI: 10.1101/2023.11.28.568918.

[77] Minsheng Hao et al. “Large-Scale Foundation Model on Single-Cell Transcriptomics”. In: *Nature Methods* (June 2024), pp. 1–11. ISSN: 1548-7105. DOI: 10.1038/s41592-024-02305-7.

[78] Simon Batzner et al. “E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials”. In: *Nature Communications* 13.1 (May 2022), p. 2453. ISSN: 2041-1723. DOI: 10.1038/s41467-022-29939-5.

[79] Nicola De Cao and Thomas Kipf. “MolGAN: An Implicit Generative Model for Small Molecular Graphs”. May 30, 2018. arXiv: 1805.11973 [cs, stat]. URL: http://arxiv.org/abs/1805.11973.

[80] Yanglan Gan et al. “Inferring Gene Regulatory Networks from Single-Cell Transcriptomics Based on Graph Embedding”. In: *Bioinformatics* 40.5 (May 2024). DOI: 10.1093/bioinformatics/btae291.

[81] Gabriele Corso et al. “Graph Neural Networks”. In: *Nature Reviews Methods Primers* 4.1 (Mar. 2024), p. 17. ISSN: 2662-8449. DOI: 10.1038/s43586-024-00294-7.

[82] Md Shamim Hussain, Mohammed J. Zaki, and Dharmashankar Subramanian. “Global Self-Attention as a Replacement for Graph Convolution”. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Aug. 2022, pp. 655–665. DOI: 10.1145/3534678.3539296. arXiv: 2108.03348 [cs].

[83] Guangyi Chen and Zhi-Ping Liu. “Graph Attention Network for Link Prediction of Gene Regulations from Single-Cell RNA-sequencing Data”. In: *Bioinformatics* 38.19 (Sept. 2022), pp. 4522–4529. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac559.

[84] Vijay Prakash Dwivedi and Xavier Bresson. *A Generalization of Transformer Networks to Graphs*. Jan. 2021. DOI: 10.48550/arXiv.2012.09699. arXiv: 2012.09699 [cs].

[85] Chaitanya K. Joshi. *Transformers Are Graph Neural Networks*. June 2025. DOI: 10.48550/arXiv.2506.22084. arXiv: 2506.22084 [cs].

[86] Eshaan Nichani, Alex Damian, and Jason D. Lee. *How Transformers Learn Causal Structure with Gradient Descent*. Aug. 2024. DOI: 10.48550/arXiv.2402.14735. arXiv: 2402.14735 [cs].

[87] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. *Self-Attention with Relative Position Representations*. Apr. 2018. DOI: 10.48550/arXiv.1803.02155. arXiv: 1803.02155 [cs].

[88] David Lähnemann et al. "Eleven Grand Challenges in Single-Cell Data Science". In: *Genome Biology* 21.1 (Feb. 2020), p. 31. ISSN: 1474-760X. DOI: 10.1186/s13059-020-1926-6.

[89] Yusuf Roohani, Kexin Huang, and Jure Leskovec. "Predicting Transcriptional Outcomes of Novel Multigene Perturbations with GEARS". In: *Nature Biotechnology* 42.6 (June 2024), pp. 927–935. ISSN: 1546-1696. DOI: 10.1038/s41587-023-01905-6.

[90] Guadalupe Gonzalez et al. "Combinatorial Prediction of Therapeutic Perturbations Using Causally-Inspired Neural Networks". In: *bioRxiv* (Jan. 2024), p. 2024.01.03.573985. DOI: 10.1101/2024.01.03.573985.

[91] Taku Harada et al. "A Distinct Core Regulatory Module Enforces Oncogene Expression in KMT2A-rearranged Leukemia". In: *Genes & Development* 36.5-6 (Mar. 2022), pp. 368–389. ISSN: 0890-9369, 1549-5477. DOI: 10.1101/gad.349284.121.

[92] Taku Harada et al. "Leukemia Core Transcriptional Circuitry Is a Sparsely Interconnected Hierarchy Stabilized by Incoherent Feed-Forward Loops". In: *bioRxiv* (Mar. 2023), p. 2023.03.13.532438. DOI: 10.1101/2023.03.13.532438.

[93] Russell Littman et al. "SCING: Inference of Robust, Interpretable Gene Regulatory Networks from Single Cell and Spatial Transcriptomics". In: *iScience* 26.7 (July 2023), p. 107124. ISSN: 2589-0042. DOI: 10.1016/j.isci.2023.107124.

[94] Sara Aibar et al. "SCENIC: Single-Cell Regulatory Network Inference and Clustering". In: *Nature Methods* 14.11 (Nov. 2017), pp. 1083–1086. ISSN: 1548-7105. DOI: 10.1038/nmeth.4463.

[95] Carmen Bravo González-Blas et al. "SCENIC+: Single-Cell Multiomic Inference of Enhancers and Gene Regulatory Networks". In: *Nature Methods* 20.9 (Sept. 2023), pp. 1355–1367. ISSN: 1548-7105. DOI: 10.1038/s41592-023-01938-4.

[96] Guangxin Su et al. *Inferring Gene Regulatory Networks by Hypergraph Variational Autoencoder*. Apr. 2024. DOI: 10.1101/2024.04.01.586509.

[97] N. Alexia Raharinirina et al. "Inferring Gene Regulatory Networks from Single-Cell RNA-seq Temporal Snapshot Data Requires Higher-Order Moments". In: *Patterns* 2.9 (Sept. 2021), p. 100332. ISSN: 2666-3899. DOI: 10.1016/j.patter.2021.100332.

[98] Lingfei Wang et al. "Dictys: Dynamic Gene Regulatory Network Dissects Developmental Continuum with Single-Cell Multiomics". In: *Nature Methods* 20.9 (Sept. 2023), pp. 1368–1378. ISSN: 1548-7105. DOI: 10.1038/s41592-023-01971-3.

[99] Shilu Zhang et al. “Inference of Cell Type-Specific Gene Regulatory Networks on Cell Lineages from Single Cell Omic Datasets”. In: *Nature Communications* 14.1 (May 2023), p. 3064. ISSN: 2041-1723. DOI: 10.1038/s41467-023-38637-9.

[100] Juexin Wang et al. “Inductive Inference of Gene Regulatory Network Using Supervised and Semi-Supervised Graph Neural Networks”. In: *Computational and Structural Biotechnology Journal* 18 (Nov. 2020), pp. 3335–3343. ISSN: 2001-0370. DOI: 10.1016/j.csbj.2020.10.022.

[101] Kenji Kamimoto et al. “Dissecting Cell Identity via Network Inference and in Silico Gene Perturbation”. In: *Nature* 614.7949 (Feb. 2023), pp. 742–751. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05688-9.

[102] Hantao Shu et al. “Modeling Gene Regulatory Networks Using Neural Network Architectures”. In: *Nature Computational Science* 1.7 (July 2021), pp. 491–501. ISSN: 2662-8457. DOI: 10.1038/s43588-021-00099-8.

[103] Ann Boija et al. “Transcription Factors Activate Genes through the Phase-Separation Capacity of Their Activation Domains”. In: *Cell* 175.7 (Dec. 2018), 1842–1855.e16. ISSN: 0092-8674. DOI: 10.1016/j.cell.2018.10.042.

[104] GRETA FRIAR. *It Takes Three to Tango: Transcription Factors Bind DNA, Protein, and RNA | Whitehead Institute*. https://wi.mit.edu/news/it-takes-three-tango-transcription-factors-bind-dna-protein-and-rna. June 2023.

[105] Ozgur Oksuz et al. “Transcription Factors Interact with RNA to Regulate Genes”. In: *Molecular Cell* 83.14 (July 2023), 2449–2463.e13. ISSN: 1097-2765. DOI: 10.1016/j.molcel.2023.06.012.

[106] Priyanka Dey Talukdar and Urmi Chatterji. “Transcriptional Co-Activators: Emerging Roles in Signaling Pathways and Potential Therapeutic Targets for Diseases”. In: *Signal Transduction and Targeted Therapy* 8.1 (Nov. 2023), pp. 1–41. ISSN: 2059-3635. DOI: 10.1038/s41392-023-01651-w.

[107] Luisa Statello et al. “Gene Regulation by Long Non-Coding RNAs and Its Biological Functions”. In: *Nature Reviews Molecular Cell Biology* 22.2 (Feb. 2021), pp. 96–118. ISSN: 1471-0080. DOI: 10.1038/s41580-020-00315-9.

[108] Peizhuo Wang et al. “Deciphering Driver Regulators of Cell Fate Decisions from Single-Cell Transcriptomics Data with CEFCON”. In: *Nature Communications* 14.1 (Dec. 2023), p. 8459. ISSN: 2041-1723. DOI: 10.1038/s41467-023-44103-3.

[109] Aditya Pratapa et al. “Benchmarking Algorithms for Gene Regulatory Network Inference from Single-Cell Transcriptomic Data”. In: *Nature Methods* 17.2 (Feb. 2020), pp. 147–154. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0690-6.

[110] Sunnie Grace McCalla et al. “Identifying Strengths and Weaknesses of Methods for Computational Network Inference from Single-Cell RNA-seq Data”. In: *G3 Genes|Genomes|Genetics* 13.3 (Mar. 2023), jkad004. ISSN: 2160-1836. DOI: 10.1093/g3journal/jkad004.

[111] Jeremie Kalfon. *Cantinilab/scPRINT*. Machine Learning for Integrative Genomics lab. Feb. 2025.

[112] CZI Single-Cell Biology Program et al. *CZ CELL×GENE Discover: A Single-Cell Data Platform for Scalable Exploration, Analysis and Modeling of Aggregated Data*. Nov. 2023. DOI: 10.1101/2023.10.30.563174.

[113] Jeremie Kalfon. *Jkobject/benGRN: Awesome Benchmark of Gene Regulatory Networks*. Jan. 2025.

[114] Jeremie Kalfon. *Cantinilab/GRnnData*. Machine Learning for Integrative Genomics lab. Jan. 2025.

[115] sergey ribakov jeremie kalfon. *Training Foundation Models on Large Collections of scRNA-seq Data*. https://lamin.ai/blog/arrayloader-benchmarks.

[116] Tri Dao. *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*. July 2023. DOI: 10.48550/arXiv.2307.08691. arXiv: 2307.08691 [cs].

[117] Ana Carolina Leote, Xiaohui Wu, and Andreas Beyer. "Regulatory Network-Based Imputation of Dropouts in Single-Cell RNA Sequencing Data". In: *PLOS Computational Biology* 18.2 (Feb. 2022), e1009849. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1009849.

[118] Zoe Piran et al. "Disentanglement of Single-Cell Data with Biolord". In: *Nature Biotechnology* (Jan. 2024), pp. 1–6. ISSN: 1546-1696. DOI: 10.1038/s41587-023-02079-x.

[119] Alexander Rives et al. "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (Apr. 2021), e2016239118. DOI: 10.1073/pnas.2016239118.

[120] Jun Hu et al. "Improving Protein-Protein Interaction Prediction Using Protein Language Model and Protein Network Features". In: *Analytical Biochemistry* 693 (Oct. 2024), p. 115550. ISSN: 0003-2697. DOI: 10.1016/j.ab.2024.115550.

[121] Samira Abnar and Willem Zuidema. *Quantifying Attention Flow in Transformers*. May 2020. DOI: 10.48550/arXiv.2005.00928. arXiv: 2005.00928 [cs].

[122] Kevin Clark et al. *What Does BERT Look At? An Analysis of BERT's Attention*. June 2019. DOI: 10.48550/arXiv.1906.04341. arXiv: 1906.04341 [cs].

[123] Adrien Bibal et al. "Is Attention Explanation? An Introduction to the Debate". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3889–3900. DOI: 10.18653/v1/2022.acl-long.269.

[124] Vân Anh Huynh-Thu et al. "Inferring Regulatory Networks from Expression Data Using Tree-Based Methods". In: *PLOS ONE* 5.9 (Sept. 2010), e12776. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0012776.

[125] Han Chen et al. *Quantized Multi-Task Learning for Context-Specific Representations of Gene Network Dynamics*. Aug. 2024. DOI: 10.1101/2024.08.16.608180.

[126] Payam Dibaeinia and Saurabh Sinha. "SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks". In: *Cell Systems* 11.3 (Sept. 2020), 252–271.e11. ISSN: 2405-4712. DOI: 10.1016/j.cels.2020.08.003.

[127] Zhi-Ping Liu et al. "RegNetwork: An Integrated Database of Transcriptional and Post-Transcriptional Regulatory Networks in Human and Mouse". In: *Database: The Journal of Biological Databases and Curation* 2015 (Sept. 2015), bav095. DOI: 10.1093/database/bav095.

[128] Aravind Subramanian et al. "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.43 (Oct. 2005), pp. 15545–15550. ISSN: 0027-8424. DOI: 10.1073/pnas.0506580102.

[129] Dénes Türei, Tamás Korcsmáros, and Julio Saez-Rodriguez. "OmniPath: Guidelines and Gateway for Literature-Curated Signaling Pathway Resources". In: *Nature Methods* 13.12 (Dec. 2016), pp. 966–967. ISSN: 1548-7105. DOI: 10.1038/nmeth.4077.

[130] Jamie L. Marshall et al. "High-Resolution Slide-seqV2 Spatial Transcriptomics Enables Discovery of Disease-Specific Cell Neighborhoods and Pathways". In: *iScience* 25.4 (Apr. 2022). ISSN: 2589-0042. DOI: 10.1016/j.isci.2022.104097.

[131] Sean K. Wang et al. "Single-Cell Multiome of the Human Retina and Deep Learning Nominate Causal Variants in Complex Eye Diseases". In: *Cell Genomics* 2.8 (Aug. 2022), p. 100164. ISSN: 2666-979X. DOI: 10.1016/j.xgen.2022.100164.

[132] Eshaan Nichani, Alex Damian, and Jason D. Lee. *How Transformers Learn Causal Structure with Gradient Descent*. Feb. 2024. DOI: 10.48550/arXiv.2402.14735. arXiv: 2402.14735 [cs, math, stat].

[133] Kyuhong Shim et al. *Layer-Wise Pruning of Transformer Attention Heads for Efficient Language Modeling*. Oct. 2021. DOI: 10.48550/arXiv.2110.03252. arXiv: 2110.03252 [cs].

[134] William Fedus, Barret Zoph, and Noam Shazeer. *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. June 2022. DOI: 10.48550/arXiv.2101.03961. arXiv: 2101.03961 [cs].

[135] Oscar Franzén, Li-Ming Gan, and Johan L M Björkegren. "PanglaoDB: A Web Server for Exploration of Mouse and Human Single-Cell RNA Sequencing Data". In: *Database: The Journal of Biological Databases and Curation* 2019 (Apr. 2019), baz046. ISSN: 1758-0463. DOI: 10.1093/database/baz046.

[136] Arthur Liberzon et al. "The Molecular Signatures Database (MSigDB) Hallmark Gene Set Collection". In: *Cell Systems* 1.6 (Dec. 2015), pp. 417–425. ISSN: 2405-4712. DOI: 10.1016/j.cels.2015.12.004.

[137] Peter J. Park. “ChIP–Seq: Advantages and Challenges of a Maturing Technology”. In: *Nature Reviews Genetics* 10.10 (Oct. 2009), pp. 669–680. ISSN: 1471-0064. DOI: 10.1038/nrg2641.

[138] Liying Yan et al. “Single-Cell RNA-Seq Profiling of Human Preimplantation Embryos and Embryonic Stem Cells”. In: *Nature structural & molecular biology* 20.9 (Sept. 2013), pp. 1131–1139. ISSN: 1545-9985. DOI: 10.1038/nsmb.2660.

[139] Benjamin L. Kidder, Gangqing Hu, and Keji Zhao. “ChIP-Seq: Technical Considerations for Obtaining High Quality Data”. In: *Nature immunology* 12.10 (Sept. 2011), pp. 918–922. ISSN: 1529-2908. DOI: 10.1038/ni.2117.

[140] Joseph M. Replogle et al. “Mapping Information-Rich Genotype-Phenotype Landscapes with Genome-Scale Perturb-seq”. In: *Cell* 185.14 (July 2022), 2559–2575.e28. ISSN: 1097-4172. DOI: 10.1016/j.cell.2022.05.013.

[141] Florian Wagner, Yun Yan, and Itai Yanai. *K-Nearest Neighbor Smoothing for High-Throughput Single-Cell RNA-Seq Data*. Apr. 2018. DOI: 10.1101/217737.

[142] Tavé van Zyl et al. “Cell Atlas of the Human Ocular Anterior Segment: Tissue-specific and Shared Cell Types”. In: *Proceedings of the National Academy of Sciences* 119.29 (July 2022), e2200914119. DOI: 10.1073/pnas.2200914119.

[143] Joseph Burclaff et al. “A Proximal-to-Distal Survey of Healthy Adult Human Small Intestine and Colon Epithelium by Single-Cell Transcriptomics”. In: *Cellular and Molecular Gastroenterology and Hepatology* 13.5 (2022), pp. 1554–1589. ISSN: 2352-345X. DOI: 10.1016/j.jcmgh.2022.02.007.

[144] Malte D. Luecken et al. “Benchmarking Atlas-Level Data Integration in Single-Cell Genomics”. In: *Nature Methods* 19.1 (Jan. 2022), pp. 41–50. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01336-8.

[145] *Openproblems-Bio/Openproblems-V2*. Open Problems in Single-Cell Analysis. July 2024.

[146] C. Domínguez Conde et al. “Cross-Tissue Immune Cell Analysis Reveals Tissue-Specific Features in Humans”. In: *Science (New York, N.Y.)* 376.6594 (May 2022), eabl5197. DOI: 10.1126/science.abl5197.

[147] Lisa Sikkema et al. “An Integrated Cell Atlas of the Lung in Health and Disease”. In: *Nature Medicine* 29.6 (June 2023), pp. 1563–1577. ISSN: 1546-170X. DOI: 10.1038/s41591-023-02327-2.

[148] Matthew Amodio et al. “Exploring Single-Cell Data with Deep Multitasking Neural Networks”. In: *Nature Methods* 16.11 (Nov. 2019), pp. 1139–1145. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0576-7.

[149] Jialin Liu et al. “Jointly Defining Cell Types from Multiple Single-Cell Datasets Using LIGER”. In: *Nature Protocols* 15.11 (Nov. 2020), pp. 3632–3662. ISSN: 1750-2799. DOI: 10.1038/s41596-020-0391-8.

[150] Aws Saudi et al. “Immune-Activated B Cells Are Dominant in Prostate Cancer”. In: *Cancers* 15.3 (Feb. 2023), p. 920. ISSN: 2072-6694. DOI: 10.3390/cancers15030920.

[151] Anja Bruchmann et al. “Bcl-2 Associated Athanogene 5 (Bag5) Is Overexpressed in Prostate Cancer and Inhibits ER-stress Induced Apoptosis”. In: *BMC Cancer* 13 (Mar. 2013), p. 96. ISSN: 1471-2407. DOI: 10.1186/1471-2407-13-96.

[152] Vanesa C. Sanchez et al. “Host CLIC4 Expression in the Tumor Microenvironment Is Essential for Breast Cancer Metastatic Competence”. In: *PLOS Genetics* 18.6 (June 2022), e1010271. ISSN: 1553-7404. DOI: 10.1371/journal.pgen.1010271.

[153] Ru Zhu et al. “TAP1, a Potential Immune-Related Prognosis Biomarker with Functional Significance in Uveal Melanoma”. In: *BMC Cancer* 23.1 (Feb. 2023), p. 146. ISSN: 1471-2407. DOI: 10.1186/s12885-023-10527-9.

[154] Xihui Liu et al. “Targeting LIPA Independent of Its Lipase Activity Is a Therapeutic Strategy in Solid Tumors via Induction of Endoplasmic Reticulum Stress”. In: *Nature Cancer* 3.7 (July 2022), pp. 866–884. ISSN: 2662-1347. DOI: 10.1038/s43018-022-00389-8.

[155] Dakai Yang et al. “Cancer-Associated Fibroblasts: From Basic Science to Anticancer Therapy”. In: *Experimental & Molecular Medicine* 55.7 (July 2023), pp. 1322–1332. ISSN: 2092-6413. DOI: 10.1038/s12276-023-01013-0.

[156] Sathyavathi ChallaSivaKanaka et al. “Fibroblast Heterogeneity in Prostate Carcinogenesis”. In: *Cancer Letters* 525 (Jan. 2022), pp. 76–83. ISSN: 1872-7980. DOI: 10.1016/j.canlet.2021.10.028.

[157] Kok Bin Lim. “Epidemiology of Clinical Benign Prostatic Hyperplasia”. In: *Asian Journal of Urology* 4.3 (July 2017), pp. 148–151. ISSN: 2214-3882. DOI: 10.1016/j.ajur.2017.06.004.

[158] J. Kuźnicki et al. “Calcyclin as a Marker of Human Epithelial Cells and Fibroblasts”. In: *Experimental Cell Research* 200.2 (June 1, 1992), pp. 425–430. ISSN: 0014-4827. DOI: 10.1016/0014-4827(92)90191-A. URL: https://www.sciencedirect.com/science/article/pii/001448279290191A.

[159] Yidian Wang et al. “S100A6: Molecular Function and Biomarker Role”. In: *Biomarker Research* 11 (Sept. 2023), p. 78. ISSN: 2050-7771. DOI: 10.1186/s40364-023-00515-3.

[160] *WikiPathways 2024: Next Generation Pathway Database | Nucleic Acids Research | Oxford Academic*. URL: https://academic.oup.com/nar/article/52/D1/D679/7369835.

[161] Iryna R. Vyshnevska et al. “THE ROLE OF BIOMARKER MACROPHAGE MIGRATION INHIBITORY FACTOR IN CARDIAC REMODELING PREDICTION IN PATIENTS WITH ST-SEGMENT ELEVATION MYOCARDIAL INFARCTION”. In: *Wiadomosci Lekarskie (Warsaw, Poland: 1960)* 76.5 pt 1 (2023), pp. 911–919. ISSN: 0043-5147. DOI: 10.36740/WLek202305104.

[162] Rui He et al. “IGFBP7 Promotes Endothelial Cell Repair in the Recovery Phase of Acute Lung Injury”. In: *Clinical Science (London, England: 1979)* 138.13 (July 2024), pp. 797–815. ISSN: 1470-8736. DOI: 10.1042/CS20240179.

[163] Yan Li et al. “The Prostate-Associated Gene 4 (PAGE4) Could Play a Role in the Development of Benign Prostatic Hyperplasia under Oxidative Stress”. In: *Oxidative Medicine and Cellular Longevity* 2022 (2022), p. 7041739. ISSN: 1942-0994. DOI: 10.1155/2022/7041739. pmid: 35633887.

[164] Chengcheng Lv et al. “PAGE4 Promotes Prostate Cancer Cells Survive under Oxidative Stress through Modulating MAPK/JNK/ERK Pathway”. In: *Journal of experimental & clinical cancer research: CR* 38.1 (Jan. 18, 2019), p. 24. ISSN: 1756-9966. DOI: 10.1186/s13046-019-1032-3. pmid: 30658679.

[165] Aline Marques Dias et al. “Downregulation of Metallothionein 2A Reduces Migration, Invasion and Proliferation Activities in Human Squamous Cell Carcinoma Cells”. In: *Molecular Biology Reports* 49.5 (May 2022), pp. 3665–3674. ISSN: 1573-4978. DOI: 10.1007/s11033-022-07206-6. pmid: 35107738.

[166] Jiawen Luo et al. “Mechanism of Prognostic Marker SPOCK3 Affecting Malignant Progression of Prostate Cancer and Construction of Prognostic Model”. In: *BMC Cancer* 23 (Aug. 11, 2023), p. 741. ISSN: 1471-2407. DOI: 10.1186/s12885-023-11151-3. pmid: 37563543. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10416445/.

[167] Gabriela Boufelli de Freitas et al. “The Circulating 70 kDa Heat Shock Protein (HSPA1A) Level Is a Potential Biomarker for Breast Carcinoma and Its Progression”. In: *Scientific Reports* 12 (July 29, 2022), p. 13012. ISSN: 2045-2322. DOI: 10.1038/s41598-022-17414-6. pmid: 35906272. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9338230/.

[168] Michela Pasello, Maria Cristina Manara, and Katia Scotlandi. “CD99 at the Crossroads of Physiology and Pathology”. In: *Journal of Cell Communication and Signaling* 12.1 (Mar. 2018), pp. 55–68. ISSN: 1873-9601. DOI: 10.1007/s12079-017-0445-z.

[169] Marija Milacic et al. “The Reactome Pathway Knowledgebase 2024”. In: *Nucleic Acids Research* 52.D1 (Jan. 5, 2024), pp. D672–D678. ISSN: 0305-1048. DOI: 10.1093/nar/gkad1025. URL: https://doi.org/10.1093/nar/gkad1025.

[170] Mate Maus et al. “Iron Accumulation Drives Fibrosis, Senescence and the Senescence-Associated Secretory Phenotype”. In: *Nature Metabolism* 5.12 (Dec. 2023), pp. 2111–2130. ISSN: 2522-5812. DOI: 10.1038/s42255-023-00928-2. URL: https://www.nature.com/articles/s42255-023-00928-2.

[171] Youliang Qian et al. “Establishment of Cancer-Associated Fibroblasts-Related Subtypes and Prognostic Index for Prostate Cancer through Single-Cell and Bulk RNA Transcriptome”. In: *Scientific Reports* 13.1 (June 3, 2023), p. 9016. ISSN: 2045-2322. DOI: 10.1038/s41598-023-36125-0. URL: https://www.nature.com/articles/s41598-023-36125-0.

[172] Xingguo Li et al. “Accumulation of NCOA1 Dependent on HERC3 Deficiency Transactivates Matrix Metallopeptidases and Promotes Extracellular Matrix Degradation in Intervertebral Disc Degeneration”. In: *Life Sciences* 320 (May 1, 2023), p. 121555. ISSN: 0024-3205. DOI: 10.1016/j.lfs.2023.121555. URL: https://www.sciencedirect.com/science/article/pii/S0024320523001893.

[173] Athina Kladi-Skandali et al. “Expressional Profiling and Clinical Relevance of RNase K in Prostate Cancer: A Novel Indicator of Favorable Progression-Free Survival”. In: *Journal of Cancer Research and Clinical Oncology* 144.10 (Oct. 2018), pp. 2049–2057. ISSN: 1432-1335. DOI: 10.1007/s00432-018-2719-0. pmid: 30054827.

[174] Emmanuelle Carpentier et al. “SELENOPROTEIN T Deficiency Alters Projection Neuron Migration during Corticogenesis in Mice”. In: *Neuroscience* 585 (Oct. 2025), pp. 323–334. ISSN: 1873-7544. DOI: 10.1016/j.neuroscience.2025.09.015.

[175] Eleri M. Jones, Christine A. Cochrane, and Steven L. Percival. “The Effect of pH on the Extracellular Matrix and Biofilms”. In: *Advances in Wound Care* 4.7 (July 2015), pp. 431–439. ISSN: 2162-1918. DOI: 10.1089/wound.2014.0538.

[176] Francesco Colotta et al. “Cancer-Related Inflammation, the Seventh Hallmark of Cancer: Links to Genetic Instability”. In: *Carcinogenesis* 30.7 (July 1, 2009), pp. 1073–1081. ISSN: 0143-3334. DOI: 10.1093/carcin/bgp127. URL: https://doi.org/10.1093/carcin/bgp127.

[177] Douglas Hanahan and Robert A. Weinberg. “Hallmarks of Cancer: The Next Generation”. In: *Cell* 144.5 (Mar. 4, 2011), pp. 646–674. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2011.02.013. pmid: 21376230. URL: https://www.cell.com/cell/abstract/S0092-8674(11)00127-9.

[178] Jérémie Kalfon et al. *scPRINT: Pre-Training on 50 Million Cells Allows Robust Gene Network Predictions*. July 2024. DOI: 10.1101/2024.07.29.605556.

[179] Le Song, Eran Segal, and Eric Xing. *Toward AI-Driven Digital Organism: Multiscale Foundation Models for Predicting, Simulating and Programming Biology at All Levels*. Dec. 9, 2024. DOI: 10.48550/arXiv.2412.06993. arXiv: 2412.06993 [cs]. URL: http://arxiv.org/abs/2412.06993. Pre-published.

[180] Josh Abramson et al. “Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3”. In: *Nature* 630.8016 (June 2024), pp. 493–500. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07487-w.

[181] Oscar Méndez-Lucio, Christos A. Nicolaou, and Berton Earnshaw. “MolE: A Foundation Model for Molecular Graphs Using Disentangled Attention”. In: *Nature Communications* 15.1 (Nov. 2024), p. 9431. ISSN: 2041-1723. DOI: 10.1038/s41467-024-53751-y.

[182] Hugo Dalla-Torre et al. *The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics*. Oct. 2024. DOI: 10.1101/2023.01.11.523679.

[183] Garyk Brixi et al. *Genome Modeling and Design across All Domains of Life with Evo 2*. Feb. 2025. DOI: 10.1101/2025.02.18.638918.

[184] Ihab Bendidi et al. *Benchmarking Transcriptomics Foundation Models for Perturbation Analysis : One PCA Still Rules Them All*. Nov. 2024. DOI: 10.48550/arXiv.2410.13956. arXiv: 2410.13956 [cs].

[185] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. Feb. 2024. DOI: 10.48550/arXiv.2304.07193. arXiv: 2304.07193 [cs].

[186] Xiyue Wang et al. "A Pathology Foundation Model for Cancer Diagnosis and Prognosis Prediction". In: *Nature* 634.8035 (Oct. 2024), pp. 970–978. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07894-z.

[187] Mohammad Lotfollahi et al. "Mapping Single-Cell Data to Reference Atlases by Transfer Learning". In: *Nature Biotechnology* 40.1 (Jan. 2022), pp. 121–130. ISSN: 1546-1696. DOI: 10.1038/s41587-021-01001-7.

[188] Amy X. Lu et al. *Tokenized and Continuous Embedding Compressions of Protein Sequence and Structure*. Nov. 2024. DOI: 10.1101/2024.08.06.606920.

[189] Fabian Mentzer et al. *Finite Scalar Quantization: VQ-VAE Made Simple*. Oct. 2023. DOI: 10.48550/arXiv.2309.15505. arXiv: 2309.15505 [cs].

[190] Chankyu Lee et al. *NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models*. Jan. 2025. DOI: 10.48550/arXiv.2405.17428. arXiv: 2405.17428 [cs].

[191] Maximilian Ilse, Jakub Tomczak, and Max Welling. "Attention-Based Deep Multiple Instance Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. PMLR, July 2018, pp. 2127–2136.

[192] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. Oct. 2021. DOI: 10.48550/arXiv.2106.09685. arXiv: 2106.09685 [cs].

[193] Tim Dettmers et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. May 2023. DOI: 10.48550/arXiv.2305.14314. arXiv: 2305.14314 [cs].

[194] Neil Houlsby et al. *Parameter-Efficient Transfer Learning for NLP*. June 2019. DOI: 10.48550/arXiv.1902.00751. arXiv: 1902.00751 [cs].

[195] Zeming Lin et al. *Language Models of Protein Sequences at the Scale of Evolution Enable Accurate Structure Prediction*. July 2022. DOI: 10.1101/2022.07.20.500902.

[196] Naftali Tishby, Fernando C Pereira, and William Bialek. "The Information Bottleneck Method". In: ().

[197] Qifei Wang et al. *Hierarchical Interpretation of Out-of-Distribution Cells Using Bottlenecked Transformer*. Dec. 2024. DOI: 10.1101/2024.12.17.628533.

[198] C. Domínguez Conde et al. "Cross-Tissue Immune Cell Analysis Reveals Tissue-Specific Features in Humans". In: *Science* 376.6594 (May 2022), eabl5197. DOI: 10.1126/science.abl5197.

[199] Vlad-Raul Constantinescu and Ionel Popescu. *Approximation and Interpolation of Deep Neural Networks*. Apr. 2024. DOI: 10.48550/arXiv.2304.10552. arXiv: 2304.10552 [cs].

[200] Jérémie Kalfon, Gabriel Peyré, and Laura Cantini. *scPRINT-2: Towards the next-Generation of Cell Foundation Models and Benchmarks*. Dec. 2025. DOI: 10.64898/2025.12.11.693702.

[201] Jerret Ross et al. "Large-Scale Chemical Language Representations Capture Molecular Structure and Properties". In: *Nature Machine Intelligence* 4.12 (Dec. 2022), pp. 1256–1264. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00580-7.

[202] Anouar Benali et al. *Pushing the Accuracy Limit of Foundation Neural Network Models with Quantum Monte Carlo Forces and Path Integrals*. 2025. arXiv: 2504.07948 [physics.chem-ph]. URL: https://arxiv.org/abs/2504.07948.

[203] Benjamin Rhodes et al. *Orb-v3: atomistic simulation at scale*. 2025. arXiv: 2504.06231 [cond-mat.mtrl-sci]. URL: https://arxiv.org/abs/2504.06231.

[204] Eric Nguyen et al. *HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution*. 2023. arXiv: 2306.15794 [cs.LG]. URL: https://arxiv.org/abs/2306.15794.

[205] Ning Wang et al. "Multi-Purpose RNA Language Modelling with Motif-Aware Pretraining and Type-Guided Fine-Tuning". In: *Nature Machine Intelligence* 6.5 (May 2024), pp. 548–557. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00836-4.

[206] Andre Cornman et al. *The OMG Dataset: An Open MetaGenomic Corpus for Mixed-Modality Genomic Language Modeling*. Aug. 2024. DOI: 10.1101/2024.08.14.607850.

[207] Yingce Xia et al. *NatureLM: Deciphering the Language of Nature for Scientific Discovery*. Feb. 2025. DOI: 10.48550/arXiv.2502.07527. arXiv: 2502.07527 [cs].

[208] Mark-Anthony Bray et al. "Cell Painting, a High-Content Image-Based Assay for Morphological Profiling Using Multiplexed Fluorescent Dyes". In: *Nature Protocols* 11.9 (Sept. 2016), pp. 1757–1774. ISSN: 1750-2799. DOI: 10.1038/nprot.2016.105.

[209] Johann Wenckstern et al. *AI-powered Virtual Tissues from Spatial Proteomics for Clinical Diagnostics and Biomedical Discovery*. Jan. 2025. DOI: 10.48550/arXiv.2501.06039. arXiv: 2501.06039 [q-bio].

[210] Shahar Alon et al. "Expansion Sequencing: Spatially Precise in Situ Transcriptomics in Intact Biological Systems". In: *Science* 371.6528 (Jan. 2021), eaax2656. DOI: 10.1126/science.aax2656.

[211] Alexander P. Hertle, Benedikt Haberl, and Ralph Bock. "Horizontal Genome Transfer by Cell-to-Cell Travel of Whole Organelles". In: *Science Advances* 7.1 (Jan. 2021), eabd8215. DOI: 10.1126/sciadv.abd8215.

[212] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. Version 2. Jan. 22, 2019. DOI: 10.48550/arXiv.1807.03748. arXiv: 1807.03748 [cs]. URL: http://arxiv.org/abs/1807.03748. Pre-published.

[213] Lei Xiong, Tianyi Chen, and Manolis Kellis. "scCLIP: Multi-modal Single-cell Contrastive Learning Integration Pre-training". In: *NeurIPS 2023 Workshop on Machine Learning for Health*. 2023.

[214] Suyuan Zhao et al. "LangCell: language-cell pre-training for cell identity understanding". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. JMLR.org, 2024, pp. 61159–61185.

[215] Carlo De Donno et al. "Population-level integration of single-cell datasets enables multi-scale analysis across samples". In: *Nature Methods* 20 (2023), pp. 1683–1692. DOI: 10.1038/s41592-023-02035-2.

[216] Xiaodong Yang et al. *GeneCompass: Deciphering Universal Gene Regulatory Mechanisms with Knowledge-Informed Cross-Species Foundation Model*. Sept. 2023. DOI: 10.1101/2023.09.26.559542.

[217] Ding Bai et al. *scLong: A Billion-Parameter Foundation Model for Capturing Long-Range Gene Context in Single-Cell Transcriptomics*. 2024. DOI: 10.1101/2024.11.09.622759. eprint: 2024.11.09.622759.

[218] Yuansong Zeng et al. "CellFM: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells". In: *Nature Communications* 16 (2025), p. 4679. DOI: 10.1038/s41467-025-59749-2.

[219] Alejandro Tejada-Lapuerta et al. "Nicheformer: a foundation model for single-cell and spatial omics". In: *Nature Methods* (2025), pp. 1–14. DOI: 10.1038/s41592-025-02814-z.

[220] Joshua D. Pearce et al. *A Cross-Species Generative Cell Atlas Across 1.5 Billion Years of Evolution: The TranscriptFormer Single-cell Model*. 2025. DOI: 10.1101/2025.04.25.650731. eprint: 2025.04.25.650731.

[221] Jiarui Ding et al. *Toward a privacy-preserving predictive foundation model of single-cell transcriptomics with federated learning and tabular modeling*. 2025. DOI: 10.1101/2025.01.06.631427. eprint: 2025.01.06.631427.

[222] Xikun Fu et al. "A foundation model of transcription across human cell types". In: *Nature* 637 (2025), pp. 965–973. DOI: 10.1038/s41586-024-08391-z.

[223] Rebecca Boiarsky et al. "Deeper Evaluation of a Single-Cell Foundation Model". In: *Nature Machine Intelligence* 6.12 (Dec. 2024), pp. 1443–1446. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00949-w.

[224] Tianyu Liu et al. *Evaluating the Utilities of Foundation Models in Single-cell Data Analysis*. 2024. DOI: 10.1101/2023.09.08.555192. eprint: 2023.09.08.555192.

[225] Chuangyi Han et al. "Reusability Report: Exploring the Transferability of Self-Supervised Learning Models from Single-Cell to Spatial Transcriptomics". In: *Nature Machine Intelligence* 7.9 (Sept. 2025), pp. 1414–1428. ISSN: 2522-5839. DOI: 10.1038/s42256-025-01097-5.

[226] Grant Crowley and Stephen R. Quake. “Benchmarking cell type and gene set annotation by large language models with AnnDictionary”. In: *Nature Communications* 16 (2025), p. 9511. DOI: 10.1038/s41467-025-64840-6.

[227] Till Richter et al. “Delineating the Effective Use of Self-Supervised Learning in Single-Cell Genomics”. In: *Nature Machine Intelligence* 7.1 (Jan. 2025), pp. 68–78. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00934-3.

[228] Shahin Atti and Shankar Subramaniam. *Fundamental Limitations of Foundation Models in Single-Cell Transcriptomics*. 2025. DOI: 10.1101/2025.06.26.661767. eprint: 2025.06.26.661767.

[229] Gilles De Waele, Gerben Menschaert, and Willem Waegeman. *A Systematic Assessment of Single-Cell Language Model Configurations*. bioRxiv. Preprint. 2025. DOI: 10.1101/2025.04.02.646825. URL: https://doi.org/10.1101/2025.04.02.646825.

[230] Nicholas D. Youngblut et al. *scBaseCount: an AI agent-curated, uniformly processed, and autonomously updated single cell data repository*. bioRxiv. Preprint. 2025. DOI: 10.1101/2025.02.27.640494. URL: https://doi.org/10.1101/2025.02.27.640494.

[231] J. Zhang et al. *Tahoe-100M: A Giga-Scale Single-Cell Perturbation Atlas for Context-Dependent Gene Function and Cellular Modeling*. bioRxiv. Preprint. 2025. DOI: 10.1101/2025.02.20.639398. URL: https://doi.org/10.1101/2025.02.20.639398.

[232] James Malone et al. “Modeling sample variables with an Experimental Factor Ontology”. In: *Bioinformatics* 26 (2010), pp. 1112–1118. DOI: 10.1093/bioinformatics/btq099.

[233] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. “SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis”. In: *Genome Biology* 19.1 (Feb. 6, 2018), p. 15. ISSN: 1474-760X. DOI: 10.1186/s13059-017-1382-0. URL: https://doi.org/10.1186/s13059-017-1382-0.

[234] Zaitian Wang et al. *A Comprehensive Survey on Data Augmentation*. Oct. 2025. DOI: 10.48550/arXiv.2405.09591. arXiv: 2405.09591 [cs].

[235] Jérémie Kalfon, Laura Cantini, and Gabriel Peyre. *Towards foundation models that learn across biological scales*. 2025. DOI: 10.1101/2025.05.16.653447. eprint: 2025.05.16.653447.

[236] Zeming Lin et al. *Language models of protein sequences at the scale of evolution enable accurate structure prediction*. 2022. DOI: 10.1101/2022.07.20.500902. eprint: 2022.07.20.500902.

[237] OpenAI et al. *GPT-4 Technical Report*. 2024. DOI: 10.48550/arXiv.2303.08774.

[238] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. DOI: 10.48550/arXiv.2001.08361.

[239] Richard J. Chen et al. “Towards a general-purpose foundation model for computational pathology”. In: *Nature Medicine* 30 (2024), pp. 850–862. DOI: 10.1038/s41591-024-02857-3.

[240] Abhinav K. Adduri et al. *Predicting Cellular Responses to Perturbation across Diverse Contexts with State*. June 2025. DOI: 10.1101/2025.06.26.661135.

[241] Malte D. Luecken et al. "Defining and benchmarking open problems in single-cell analysis". In: *Nature Biotechnology* 43 (2025), pp. 1035–1040. DOI: 10.1038/s41587-025-02582-3.

[242] Lingling Xu et al. *Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment*. 2023. DOI: 10.48550/arXiv.2312.12148.

[243] Haoyu Zhong et al. "Benchmarking cross-species single-cell RNA-seq data integration methods: towards a cell type tree of life". In: *Nucleic Acids Research* 53 (2025), gkae1316. DOI: 10.1093/nar/gkae1316.

[244] Leonard R. Herrmann. "Laplacian-Isoparametric Grid Generation Scheme". In: *Journal of the Engineering Mechanics Division* 102 (1976), pp. 749–756.

[245] Tianle Hu et al. *GRIT: Graph-Regularized Logit Refinement for Zero-shot Cell Type Annotation*. 2025. DOI: 10.48550/arXiv.2508.04747.

[246] Nikolas L. Jorstad et al. "Transcriptomic cytoarchitecture reveals principles of human neocortex organization". In: *Science* 382 (2023), eadf6812. DOI: 10.1126/science.adf6812.

[247] Manzil Zaheer et al. *Deep Sets*. Apr. 2018. DOI: 10.48550/arXiv.1703.06114. arXiv: 1703.06114 [cs].

[248] 10x Genomics. *Preview Data: FFPE Human Skin Primary Dermal Melanoma with 5K Human Pan Tissue and Pathways Panel*. 2024. URL: https://www.10xgenomics.com/datasets/xenium-prime-ffpe-human-skin.

[249] Tommaso Biancalani et al. "Deep Learning and Alignment of Spatially Resolved Single-Cell Transcriptomes with Tangram". In: *Nature Methods* 18.11 (Nov. 2021), pp. 1352–1362. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01264-7.

[250] Chao Zhang et al. "A Single-Cell Analysis Reveals Tumor Heterogeneity and Immune Environment of Acral Melanoma". In: *Nature Communications* 13.1 (Nov. 2022), p. 7250. ISSN: 2041-1723. DOI: 10.1038/s41467-022-34877-3.

[251] Zayd M. K. Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. *Softpick: No Attention Sink, No Massive Activations with Rectified Softmax*. May 2025. DOI: 10.48550/arXiv.2504.20966. arXiv: 2504.20966 [cs].

[252] Juho Lee et al. *Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks*. 2019. DOI: 10.48550/arXiv.1810.00825.

[253] Allan Jabri, David Fleet, and Ting Chen. *Scalable Adaptive Computation for Iterative Generation*. 2023. DOI: 10.48550/arXiv.2212.11972.

[254] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". In: *Nature Methods* 15 (2018), pp. 1053–1058. DOI: 10.1038/s41592-018-0229-2.

[255] Amir Ali Moinfar and Fabian J. Theis. *Unsupervised Deep Disentangled Representation of Single-Cell Omics*. 2025. DOI: 10.1101/2024.11.06.622266. eprint: 2024.11.06.622266.

[256] Chenling Xu et al. "Probabilistic Harmonization and Annotation of Single-cell Transcriptomics Data with Deep Generative Models". In: *Molecular Systems Biology* 17.1 (Jan. 2021), MSB20209620. ISSN: 1744-4292. DOI: 10.15252/msb.20209620.

[257] Andrew Jaegle et al. *Perceiver IO: A General Architecture for Structured Inputs & Outputs*. 2022. DOI: 10.48550/arXiv.2107.14795.

[258] Andrew Jaegle et al. "Perceiver: General Perception with Iterative Attention". In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021, pp. 4651–4664.

[259] João Carreira et al. *HiP: Hierarchical Perceiver*. 2022. DOI: 10.48550/arXiv.2202.10890.

[260] Long Ouyang and Alex Key. *Maximum Mean Discrepancy for Generalization in the Presence of Distribution and Missingness Shift*. 2022. DOI: 10.48550/arXiv.2111.10344.

[261] Cheng Zhang. *Single-Cell Data Analysis Using MMD Variational Autoencoder for a More Informative Latent Representation*. 2019. DOI: 10.1101/613414. eprint: 613414.

[262] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. DOI: 10.48550/arXiv.1802.03426.

[263] Rémi Flamary et al. "POT: Python Optimal Transport". In: *Journal of Machine Learning Research* 22 (2021), pp. 1–8.

[264] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*. Mar. 2020. DOI: 10.48550/arXiv.1803.00567. arXiv: 1803.00567 [stat].

[265] Maochun Wang et al. "Comparative Analysis of Human and Mouse Transcriptomes during Skin Wound Healing". In: *Frontiers in Cell and Developmental Biology* 12 (Oct. 2024). ISSN: 2296-634X. DOI: 10.3389/fcell.2024.1486493.

[266] Meng He and Jürgen Borlak. "A Genomic Perspective of the Aging Human and Mouse Lung with a Focus on Immune Response and Cellular Senescence". In: *Immunity & Ageing* 20.1 (Nov. 2023), p. 58. ISSN: 1742-4933. DOI: 10.1186/s12979-023-00373-5.

[267] Leah V. Schaffer et al. "Multimodal cell maps as a foundation for structural and functional genomics". In: *Nature* 642 (2025), pp. 222–231. DOI: 10.1038/s41586-025-08623-y.

[268] Gielifar Mesquita et al. "H-Ferritin is essential for macrophages' capacity to store or detoxify exogenously added iron". In: *Scientific Reports* 10 (2020), p. 3061. DOI: 10.1038/s41598-020-59898-0.

[269] Karl Walter Bock. "Ah receptor, vitamin B12 and itaconate: how localized decrease of vitamin B12 prevents survival of macrophage-ingested bacteria". In: *Frontiers in Toxicology* 6 (2024), p. 1491184. DOI: 10.3389/ftox.2024.1491184.

[270] Zhuoqing Fang, Xinyuan Liu, and Gary Peltz. "GSEApy: A Comprehensive Package for Performing Gene Set Enrichment Analysis in Python". In: *Bioinformatics* 39.1 (Jan. 2023), btac757. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btac757.

[271] Thomas Hayes et al. "Simulating 500 Million Years of Evolution with a Language Model". In: *Science* 387.6736 (Feb. 2025), pp. 850–858. DOI: 10.1126/science.ads0018.

[272] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". Dec. 20, 2013. arXiv: 1312.6114 [cs, stat]. URL: http://arxiv.org/abs/1312.6114.

[273] Ruochen Jiang et al. "Statistics or Biology: The Zero-Inflation Controversy about scRNA-seq Data". In: *Genome Biology* 23.1 (Jan. 2022), p. 31. ISSN: 1474-760X. DOI: 10.1186/s13059-022-02601-5.

[274] Alexander D. Diehl et al. "The Cell Ontology 2016: Enhanced Content, Modularization, and Ontology Interoperability". In: *Journal of Biomedical Semantics* 7.1 (July 2016), p. 44. ISSN: 2041-1480. DOI: 10.1186/s13326-016-0088-7.

[275] Nicole A. Vasilevsky et al. *Mondo: Unifying Diseases for the World, by the World.* May 2022. DOI: 10.1101/2022.04.13.22273750.

[276] Simona Dalin et al. "Abstract 2710: Associations between Structural Variant Signatures and Drug Sensitivity in Cell Lines". In: *Cancer Research* 82.12_Supplement (June 2022), p. 2710. ISSN: 0008-5472. DOI: 10.1158/1538-7445.AM2022-2710.

[277] Taku Harada et al. "Rapid-Kinetics Degron Benchmarking Reveals off-Target Activities and Mixed Agonism-Antagonism of MYB Inhibitors". In: *bioRxiv* (Apr. 2023), p. 2023.04.07.536032. ISSN: 2692-8205. DOI: 10.1101/2023.04.07.536032.

[278] Sean A. Misek et al. "Germline Variation Contributes to False Negatives in CRISPR-based Experiments with Varying Burden across Ancestries". In: *Nature Communications* 15.1 (June 2024), p. 4892. ISSN: 2041-1723. DOI: 10.1038/s41467-024-48957-z.

[279] François Aguet et al. "Genetic Effects on Gene Expression across Human Tissues". In: *Nature* 550.7675 (Oct. 2017), pp. 204–213. ISSN: 1476-4687. DOI: 10.1038/nature24277.

[280] Toan Q. Nguyen and Julian Salazar. "Transformers without Tears: Improving the Normalization of Self-Attention". In: (Nov. 2019). DOI: 10.5281/zenodo.3525484. arXiv: 1910.05895 [cs, stat].

[281] Philippe Tillet, H. T. Kung, and David Cox. "Triton: An Intermediate Language and Compiler for Tiled Neural Network Computations". In: *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*. MAPL 2019. New York, NY, USA: Association for Computing Machinery, June 2019, pp. 10–19. ISBN: 978-1-4503-6719-6. DOI: 10.1145/3315508.3329973.

[282] Gao Huang et al. *Deep Networks with Stochastic Depth.* July 2016. DOI: 10.48550/arXiv.1603.09382. arXiv: 1603.09382 [cs].

[283] Ben Athiwaratkun et al. *There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average.* Feb. 2019. DOI: 10.48550/arXiv.1806.05594. arXiv: 1806.05594 [cs, stat].

[284] Sophia Müller-Dott et al. “Expanding the Coverage of Regulons from High-Confidence Prior Knowledge for Accurate Estimation of Transcription Factor Activities”. In: *Nucleic Acids Research* 51.20 (Nov. 2023), pp. 10934–10949. ISSN: 0305-1048. DOI: 10.1093/nar/gkad841.

[285] Jing Zhang et al. *Computing the Human Interactome*. Oct. 2024. DOI: 10.1101/2024.10.01.615885.

[286] Lawrence Hubert and Phipps Arabie. “Comparing Partitions”. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/BF01908075.

[287] Fabian Pedregosa et al. “Scikit-Learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928.

[288] Maren Büttner et al. “A Test Metric for Assessing Single-Cell RNA-seq Batch Correction”. In: *Nature Methods* 16.1 (Jan. 2019), pp. 43–49. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0254-1.

[289] Ilya Korsunsky et al. “Fast, Sensitive and Accurate Integration of Single-Cell Data with Harmony”. In: *Nature Methods* 16.12 (Dec. 2019), pp. 1289–1296. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0619-0.

[290] Gal Chechik et al. “Information Bottleneck for Gaussian Variables”. In: (), p. 25.

[291] Anna S. E. Cuomo et al. *Impact of Rare and Common Genetic Variation on Cell Type-Specific Gene Expression*. Mar. 2025. DOI: 10.1101/2025.03.20.25324352.

[292] Luca Nesterenko et al. “Phyloformer: Fast, Accurate, and Versatile Phylogenetic Reconstruction with Deep Neural Networks”. In: *Molecular Biology and Evolution* 42.4 (Apr. 2025). DOI: 10.1093/molbev/msaf051.

[293] Fredrik Salmen et al. “High-Throughput Total RNA Sequencing in Single Cells Using VASA-seq”. In: *Nature Biotechnology* 40.12 (Dec. 2022), pp. 1780–1793. ISSN: 1546-1696. DOI: 10.1038/s41587-022-01361-8.

[294] Oriol Llora-Batlle et al. *10x Genomics Gene Expression Flex Is a Powerful Tool for Single-Cell Transcriptomics of Xenograft Models*. Jan. 2024. DOI: 10.1101/2024.01.25.577066.

[295] Michael Hagemann-Jensen et al. “Single-Cell RNA Counting at Allele and Isoform Resolution Using Smart-seq3”. In: *Nature Biotechnology* 38.6 (June 2020), pp. 708–714. ISSN: 1546-1696. DOI: 10.1038/s41587-020-0497-0.

[296] Jules Samaran, Gabriel Peyré, and Laura Cantini. “scConfluence: Single-Cell Diagonal Integration with Regularized Inverse Optimal Transport on Weakly Connected Features”. In: *Nature Communications* 15.1 (Sept. 2024), p. 7762. ISSN: 2041-1723. DOI: 10.1038/s41467-024-51382-x.

[297] Marta M. Fay et al. *RxRx3: Phenomics Map of Biology*. Feb. 2023. DOI: 10.1101/2023.02.07.527350.

[298] Jun Yu et al. “Field Programmable Digital Microfluidics Chip for High-Throughput Droplet Array Manipulation”. In: *2023 International Electron Devices Meeting (IEDM)*. Dec. 2023, pp. 1–4. DOI: 10.1109/IEDM45741.2023.10413813.

[299] Haotian Cui et al. “Towards Multimodal Foundation Models in Molecular Cell Biology”. In: *Nature* 640.8059 (Apr. 2025), pp. 623–633. ISSN: 1476-4687. DOI: 10.1038/s41586-025-08710-y.

[300] Rand Arafeh et al. “The Present and Future of the Cancer Dependency Map”. In: *Nature Reviews Cancer* 25.1 (Jan. 2025), pp. 59–73. ISSN: 1474-1768. DOI: 10.1038/s41568-024-00763-x.

[301] Xi Wang et al. *Biological Reasoning with Reinforcement Learning through Natural Language Enables Generalizable Zero-Shot Cell Type Annotations*. June 2025. DOI: 10.1101/2025.06.17.659642.

[302] Flavia Pedrocchi et al. *Sparse Autoencoders Reveal Interpretable Features in Single-Cell Foundation Models*. Oct. 2025. DOI: 10.1101/2025.10.22.681631.

Supplementary Materials

5.1 Supplementary Tables for scPRINT

5.1.1 List of novelties in scPRINT and comparison to scGPT and scFoundation

Table 5.2: Comparison of the features and novelties from scPRINT compared to 2 similar published state-of-the-art methods: scGPT and scFoundation.

features	scPRINT	scGPT	scFoundation	Geneformer v2
classification pretraining	v	x	x	x
hierarchical classification	v	x	x	x
denoising pretraining	v	x	v	x
masking pretraining	v	v	v	v
MVC pretraining	v	v	x	x
AE pretraining	v	x	x	x
large cell count GN inference	v	x	x	x
zero-shot classification	v	x	x	x
zero-shot batch correction	v	x	x	x
zero-shot denoising	v	x	x	x
genome-wide GN inference	v	x	x	x
large input context	x	x	v	x
raw count encoding	v	x	v	x
very large model	v	x	x	x
pretraining strategy and dataset	v	x	x	x
low GPU/hours implementation	v	x	x	x
weighted random sampling	v	x	x	x
protein encoding	v	x	x	x
cross-species abilities	v	x	x	x
gene location encoding	v	x	x	x
genome-wide input context	x	x	v	x
xtrimogene architecture	x	x	v	x
train / validate / test strategies	v	x	x	x
flashattention2	v	x	x	x

5.1.2 Model comparison

model name	model size	training time (hours)	training hardware	num cells	num leaf cell type	dimension (d)	layers	heads	token input size	num species	training	attention
scPRINT-small	7M	24	1xA100	41M (91M before QC)	540	128	4	2	2,200	2	denoising (60%) + classification + bottleneck	flashattention2
Geneformer v2	? (~50M)	72	12xV100	30M	?	256	6	4	2,048	1	masked (15%)	normal
scPRINT-medium	20M	72	1xA100	41M (91M before QC)	540	256	8	4	2,200	2	denoising (60%) + classification + bottleneck	flashattention2
scGPT	100M	?	?	33M	?	512	12	8	1,200	1	masked (15%)	flashattention1
scFoundation	100M	?	?	50M	?	768	12+12	12+8	20,000	1	masked (30%) + denoising	xtrimogene
scPRINT	90M	96	4xA100	41M (91M before QC)	540	512	16	8	2,200	2	denoising (60%) + classification + bottleneck	flashattention2
GPT2-small	117M	?	?	300B tokens (~150M cells)	x	768	12	12	1200	x	masked (15%)	normal
UCE	650M	960	24xA100	36M	(~1000 ?) likely <500	1280	33	20	1024	5	masked (20%)	normal
cellFM	700M	?	32xAscend910 NPUs	100M	?	1536	40	48	4096	1	masked (20%)	normal + LORA
scPRINT-vlarge	700M	168	24xA100	41M (91M before QC)	540	1280	20	10	2,200	2	denoising (60%) + classification + bottleneck	flashattention2

Table 5.3: Comparing different model sizes and architectures. Comparing scPRINT to other state-of-the-art methods, as well as GPT2-small and GPT3-large models.

5.1.3 Ablation study and impact on performance across tasks

id	description	denoise/reco2full_vs_noisy2full	emb_lung/ct_class	emb_lung/scib	emb_panc/ct_class	emb_panc/scib	reconstruction loss	classification accuracy	denoising loss	epoch
or46096v	small	0.34	0.31	0.47	0.11	0.41	1.31	0.4	1.16	24
ghqf2hym	medium	0.12	0.58	0.55	0.52	0.51	1.25	0.33	1.125	27
7asy8qpn	large	0.18	0.69	0.56	0.52	0.50	1.23	0.76	1.109	21
24chcp2e	medium-nofreeze	0.15	0.45	0.54	0.52	0.53	1.25	0.33	1.115	23
6o76ew23	medium-2-heads	0.10	0.49	0.55	0.40	0.53	1.25	0.33	1.124	26
lsr3pvnf	medium-MSE	0.21	0.61	0.56	0.51	0.49	1.26	0.33	6.3 (diff)	29
muwj73gx	medium-MVC	0.21	0.51	0.55	0.40	0.47	1.29	0.3	1.132	37
n8jypo8z	medium-noPE	0.09	0.71	0.56	0.35	0.46	1.27	0.33	1.31	23
q0fzpj5g	medium-no-random-weighted	0.17	0.51	0.53	0.19	0.48	1.26	0.26	1.118	27
f5e4qfkr	medium-MLM	0.04	0.53	0.54	0.39	0.46	1.26	0.35	0.999	23

Table 5.4: The table shows the results of the ablation study on denoising, embedding with batch correction, and cell-type classification tasks. Results are displayed for the medium-size scPRINT model. Top to bottom: *small, medium, large:* regular models of various sizes. *medium-nofreeze:* a model trained without freezing gene embedding during pre-training. *medium-2-heads:* a model trained with only two heads per layer instead of 4. *medium-MSE*: a model with Mean Squared Error instead of the ZINB loss. *medium-MVC:* a model trained with scGPT's MVC methodology for the creation of the cell embedding. medium-noPE: a model trained without positional encoding for the gene's location. medium-no-random-weighted: a model trained without weighted random sampling. *medium-MLM*: a model trained with masked language modeling instead of denoising.

5.1.4 Computational speed of various GN inference methods

Table 5.5: The computational speed of running various gene network inference methods on a set of 4000 genes and 1000 cells. It is showing that transformer-based models are far faster than previous methods, owing to their clever use of the GPU and pre-training.

model	speed for 1000 cells	speed for a dataset of 12 cell types	scale to #cells	scale to #genes
DeepSEM	10mn	2 hours	linear	quadratic
GENIE3	50mn	10 hours	quadratic	quadratic
GENIE3 (100 trees)	4mn	1 hour	quadratic	quadratic
Geneformer v2	1mn	15mn	linear	linear
scGPT	1mn	15mn	linear	linear
scPRINT	1mn	15mn	linear	linear

5.1.5 Table S5: Performance of GN inference methods on the Sergio simulated scRNAseq dataset

Table 5.6: We generate a Sergio simulated scRNAseq dataset of 1000 cells for 800 genes from the RegNetwork ground truth network. We here showcase the ability of each model to recover the RegNetwork ground truth from this dataset. It shows how only scPRINT can recover some of RegNetwork's connections.

model	EPR	AUPRC	TF_targ	TF_enr
DeepSEM	0.92601	0.00101	0	FALSE
GENIE3	0.94497	0.00193	5.2	TRUE
Geneformer v2	0.699	0.00409	0	TRUE
scGPT	0.6167	0.00278	10.5	TRUE
scPRINT	1.836	0.00861	13.15	FALSE

5.1.6 Comparison scPRINT model size on performance across tasks and GN inference abilities

id	description	denoise/reco2full_vs_noisy2full	emb_lung/ct_class	emb_lung/scib	emb_panc/ct_class	emb_panc/scib	grn_gwps/auprc	grn_gwps/auprc_omni	grn_gwps/auprc_self	grn_gwps/epr	grn_gwps/epr_omni	grn_gwps/epr_self	grn_omni/auprc	grn_omni/auprc_class	grn_omni/epr	grn_omni/epr_class	grn_omni/tf_enr
or46096v	small	0.34	0.31	0.47	0.11	0.41	0.03804	0.038	0.039	4.375	4.375	4.420	0.00137	0.00134	2.042	2.020	2
ghqf2hym	medium	0.12	0.58	0.55	0.52	0.51	0.03183	0.032	0.035	3.239	3.040	3.514	0.00148	0.00142	2.111	2.293	1
7asy8qpn	large	0.18	0.69	0.56	0.52	0.50	0.03407	0.035	0.038	3.466	3.595	4.126	0.00139	0.00142	2.021	3.257	6

Figure 5.10: Comparison of scPRINT model sizes on performance across tasks and GN inference abilities.

grn_omni/tf_enr_class	grn_omni/tf_targ_enr	grn_omni/tf_targ_enr_class	grn_sroy/auprc	grn_sroy/auprc_omni	grn_sroy/auprc_self	grn_sroy/epr	grn_sroy/epr_omni	grn_sroy/epr_self	mean epr	mean auprc	mean ct_class	mean scib	total weigthed mean	reconstruction loss	classification accuracy	denoising loss	epoch
2	9	4	0.099	0.099	0.106	1.303	1.303	1.538	2.429	0.00459272619	0.21	0.44	**0.8902439048**	1.31	0.4	1.16	24
1	30	21	0.100	0.089	0.088	1.524	0.726	0.540	1.992	0.00407803571	0.55	0.53	**1.023048607**	1.25	0.33	1.125	27
6	32	30	0.100	0.102	0.108	0.589	0.985	1.322	2.252	0.00436602381	0.60	0.53	**1.096761024**	1.23	0.76	1.109	21

Figure 5.11: Comparison of scPRINT model sizes on performance across tasks and GN inference abilities.

5.1.7 Overlap of different GN ground truths

Table 5.7: Comparison of the overlap, expressed as precision and recall, of the three different ground truth networks used: MCalla, Omnipath, and gwps.

comparison	precision	recall	random precision
MCalla et al. vs Omnipath	0.0520	0.0074	0.00154
MCalla et al. - T vs Omnipath	0.0155	0.0022	0.00154
gwps vs Omnipath	0.0015	0.0219	0.00129
gwps -T vs Omnipath	0.0030	0.0426	0.00129

5.1.8 Table S8: Omnipath benchmark results on the genome-wide perturb-seq dataset

Table 5.8: Omnipath network overlap (EPR, AUPRC), as well as transcription factor enrichment, TF target enrichment, and cell type marker enrichment for gene networks generated by the different tools on the genome-wide perturb seq K562 cells at steady state (no perturbations).

tool	EPR	AUPRC	TF target enr.	TF_enr	TF_only	ct_pred	RAND precision
DeepSEM	4.1	0.00192	21.4	FALSE	FALSE	FALSE	0.001633
GENIE3	4.7	0.00188	17.9	TRUE	FALSE	FALSE	0.00163
Geneformer v2	0.2	0.001796	5.9	FALSE	FALSE	FALSE	0.001528
scGPT	1.0	0.00208	14.0	TRUE	FALSE	FALSE	0.00163
scPRINT	2.8	0.00170	8.6	TRUE	FALSE	FALSE	0.00161
scPRINT (omnipath's heads)	4.7	0.00189	3.4	TRUE	FALSE	FALSE	0.00161
scPRINT (gwps' heads)	1.6	0.00190	5.0	TRUE	FALSE	FALSE	0.00161

Omnipath network overlap (EPR, AUPRC), as well as transcription factor enrichment, TF target enrichment, and cell type marker enrichment for gene networks generated by the different tools on the genome-wide perturb seq K562 cells at steady state (no perturbations)

5.1.9 Omnipath benchmark results on the MCalla et al. datasets

tool	dataset	EPR	AUPRC	TF target enr.	TF enr.	cell type enr.
DeepSEM	**Han et. al.**	5.54	0.00029	18.9	FALSE	FALSE
DeepSEM	**Yan et. al.**	0.97	-0.00002	7.5	FALSE	FALSE
GENIE3	**Han et. al.**	1.51	0.00016	11.3	FALSE	TRUE
GENIE3	**Yan et. al.**	1.74	0.00020	0.0	FALSE	TRUE
Geneformer	**Han et. al.**	1.63	0.00010	11.3	FALSE	FALSE
Geneformer	**Yan et. al.**	1.99	0.00011	20.0	FALSE	FALSE
scGPT	**Han et. al.**	0.89	0.00016	17.0	TRUE	FALSE
scGPT	**Yan et. al.**	0.16	0.00007	20.0	FALSE	FALSE
scPRINT	**Han et. al.**	2.03	0.00019	23.6	TRUE	FALSE
scPRINT	**Yan et. al.**	1.76	0.00026	31.1	FALSE	TRUE
scPRINT (omnipath's heads)	**Han et. al.**	5.12	0.00004	3.6	TRUE	FALSE
scPRINT (omnipath's heads)	**Yan et. al.**	3.35	0.00019	13.3	FALSE	TRUE
scPRINT (Han et. al.'s heads)	**Han et. al.**	0.94	0.00030	30.9	TRUE	TRUE
scPRINT (Han et. al.'s heads)	**Yan et. al.**	0.57	-0.00004	6.7	TRUE	TRUE

Omnipath network overlap (EPR, AUPRC), as well as transcription factor enrichment, TF target enrichment, and cell type marker enrichment for gene networks generated by the different tools on the 2 human embryonic stem cell datasets used in Results Section 3 (scPRINT outperforms GENIE3 and scGPT on cell type-specific ground truths).

5.1.10 Denoising results per datasets

tools	denoising (+%) correlation. gNNpgpo6gATjuxTE7CCp	denoising (+%) correlation. R4ZHoQegxXdSFNFY5LGe	denoising (+%) correlation (RElyQZE6OMZm1S3W2Dxi)	denoising (+%) correlation (low cell count: 30). gNNpgpo6gATjuxTE7CCp	denoising (+%) correlation (low cell count: 30). R4ZHoQegxXdSFNFY5LGe	denoising (+%) correlation (low cell count: 30) (RElyQZE6OMZm1S3W2Dxi)	average denoising (+%) correlation	average denoising (+%) correlation (rare cell type)
untrained scPRINT	-16.0	X	X	-16.0	X	X	-16.0	-16.0
scPRINT	19.1	33.9	17.1	22.5	26.6	16.6	23.4	21.9
KNNsmoothing2	21.0	34.9	21.6	17.0	32.0	13.4	25.8	20.8
magic	29.3	34.6	22.7	16.8	24.4	4.6	28.9	15.3
magic (low cell dataset)	X	X	X	11.3	14.0	13.0	X	12.8

This table shows the detail of the denoising results for each of the three datasets for scPRINT-large, KNNsmoothing2, MAGIC, and MAGIC run on only the small cell type cluster. “Random scPRINT model” is the performance of an untrained scPRINT model.

5.1.11 Highlighted B-cell cluster genes in the BPH study

gene	link	in cancer	in b cell	analysis
		prostate	high expr in immune	
MBNL2	link	cancer	tissues	
			high expr in immune	BPH B-cell to normal B-cell diff. expr.
MAGOH	link	cancer	tissues	
RANBP2	link1, link2	B-cell lymphoma	b cell validated	
		prostate	high expr in immune	
CLIC4	link	cancer	tissues	
BAG5	link	prostate cancer	b cell in cancer	
NR4A1	link1, link2	prostate cancer	b cell validated	
BAZ2A	link	prostate cancer	b cell validated	
		tumor suppressor in prostate	b cell	
ZBTB16	link1, link2	cancer	validated b cell	
TAP1	link1, link2	cancer	validated b cell	BPH B-cell to normal B-cell diff. expr. post denoising
TAS2R19	link		validated	
PRDM7	link	cancer		
TSEN54	link	cancer	b cell in cancer b cell	
EHMT2	link		validated	
ERICH6B	link	cancer		
IL10RB	link	cancer	b cell in cancer	

Table of the highlighted genes in the differential expression analysis in BPH vs normal B-cells together with their annotation on their relation to cancer and to b-cells, with sources.

5.1.12 Hub and differential hub genes in the fibroblast GN of the BPH study

TOP 15 hubs in BPH fibroblasts GN	TOP 15 hubs in normal fibroblasts GN	TOP 15 differential hubs in BPH fibroblasts vs normal	TOP 15 eigenvector_centrality differential hubs in BPH fibroblasts vs normal
HSPA1A	S100A6	HLA-A	CD99
MT2A	TGIF2-RAB5IF	MT2A	HLA-A
CREM	MIF	ATP6V0C	HSPA1A
TGIF2-RAB5IF	DNAJB9	DEFA1	LUM
HSPE1	IGFBP7	EIF4A1	ATP6V0C
CALD1	APOD	HSPA1A	CD99
SPOCK3	BRME1	LUM	EIF4A1
HLA-A	SPARCL1	SPOCK3	PAGE4
SPARCL1	TIMP1	nan-99	RYR2
RBP1	DCN	CD99	SERPINF1
C1S	C1S	CPE	C1R
BRME1-1	MGP	THBS1	COL6A2
FABP4	nan-270	LGALS1	HNRNPA0
nan-99	SLC25A6	PYDC2	SERPING1
LUM	BLOC1S5-TXNDC5	SERPING1	SERPINA3

List of the Top-15 elements in different GN analyses. Genes in yellow in the last columns are the new ones found with eigenvector centrality compared to the 3rd columns.

5.1.13 Number of elements predicted per class

Table 5.12: Number of labels predicted by the model for each class. We use hierarchical classification for cell type, disease, assay, and ethnicity.

ethnicity	21
sex	2
organism	2
cell type	424
disease	62
assay	26

5.2 Supplementary figures for scPRINT

5.2.1 visualization of human gene embedding from ESM2

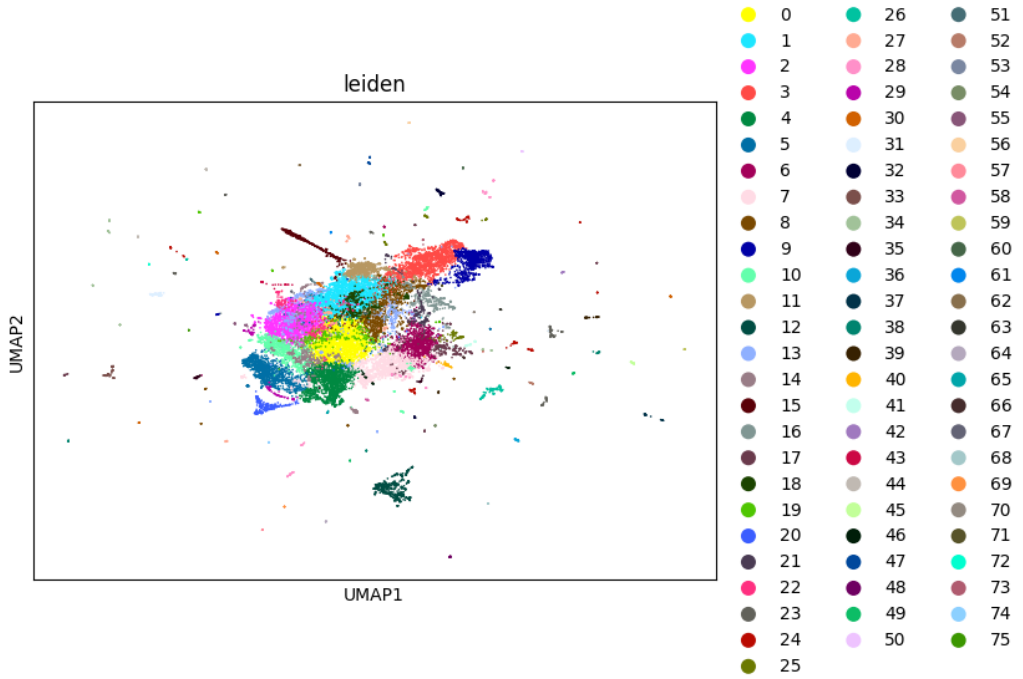


Figure 5.12: Umap of the ESM2 protein embeddings for the most common protein of all protein-coding genes in Ensembl. The PCA variance ratio is 0.856 for the top 50 principal components. We color it using the Louvain clustering of the embedding.

5.2.2 Gene network inference comparison with Omnipath per datasets

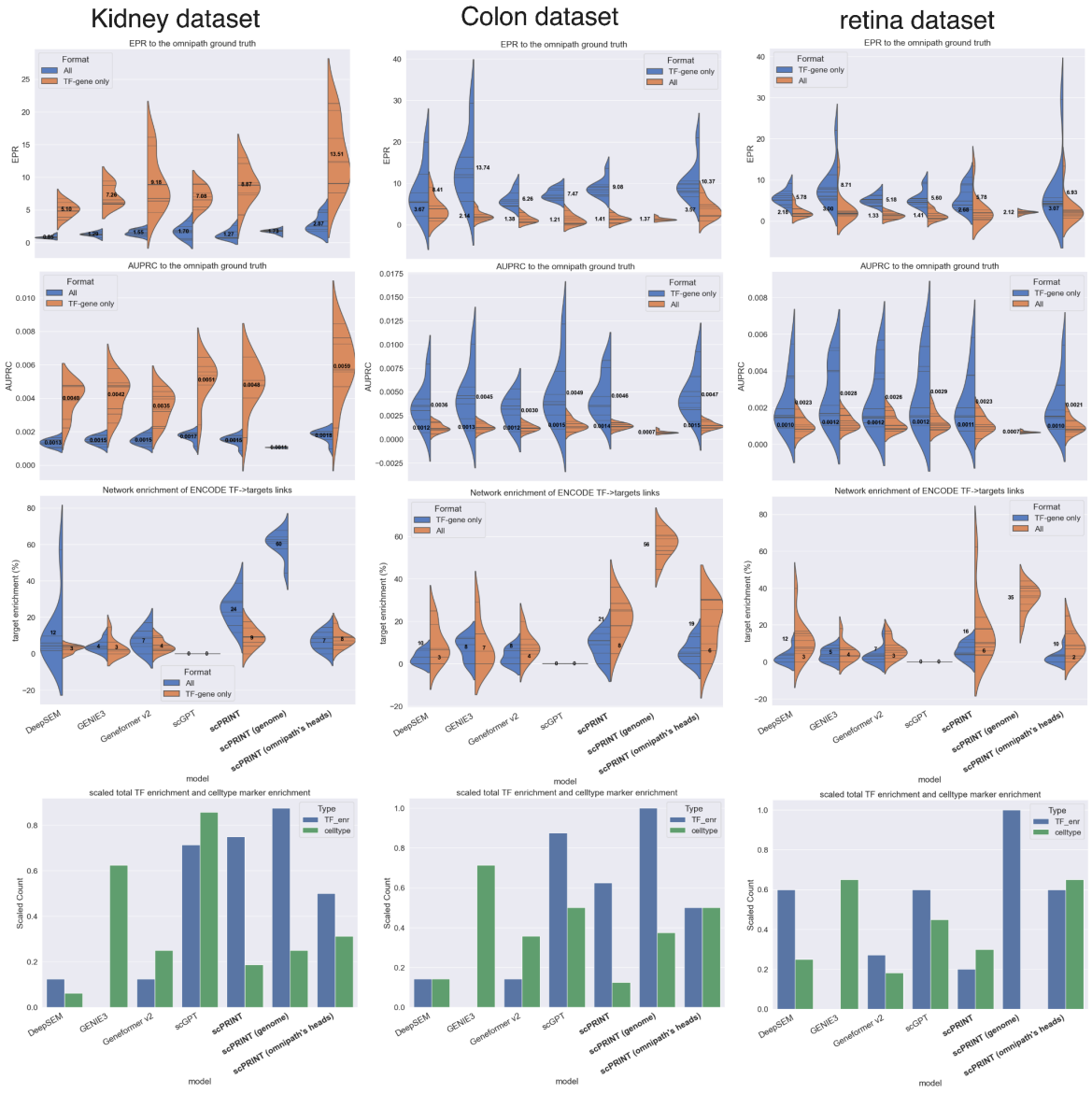


Figure 5.13: The same plots as in Figures 2B, C, and D, showing the Omnipath and enrichment results per dataset for each of the 3 datasets used. Source data are provided as a Source Data file.

5.2.3 Distribution of connection amongst the three ground truths

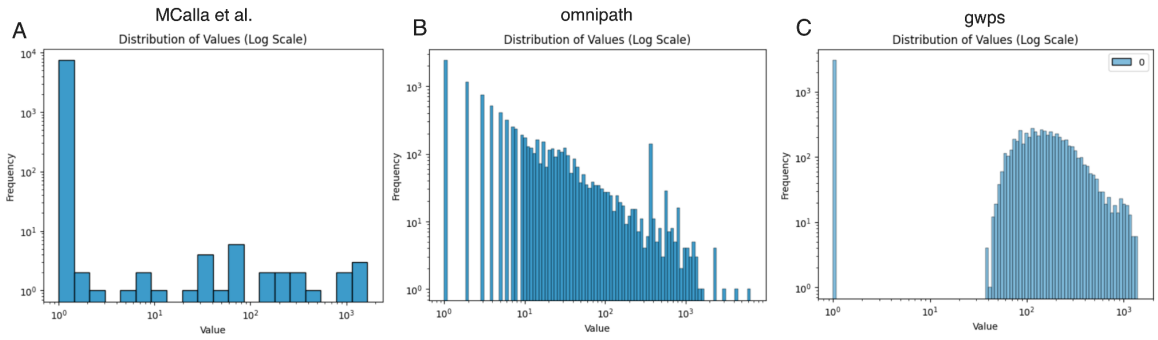


Figure 5.14: Distribution of connection amongst the three ground truths. (a) Barplot of the distribution of the number of connections per edge in the MCalla human ground truth network. Most connections are 0, and there is a roughly uniform distribution of connections otherwise. This means most connections belong to the half a dozen most connected edges. (b) Barplot of the distribution of the number of connections per edge in the Omnipath ground truth network. We can see an almost linear relationship on the log-log scale, suggesting a power law distribution. (c) Barplot of the distribution of the number of connections per edge in the genome-wide perturb-seq ground truth network. We can see a very different distribution where only a few genes have little differentially expressed genes post-knock-out, and this trend increases until reaching around 200 connections. Then, it diminishes in what might be a power law. However, some of it is likely caused by the differential expression method and noise in the scRNAseq methodology.

5.2.4 Performance of each GN inference method on predicting the TF-gene only subset of the GWPS ground truth network

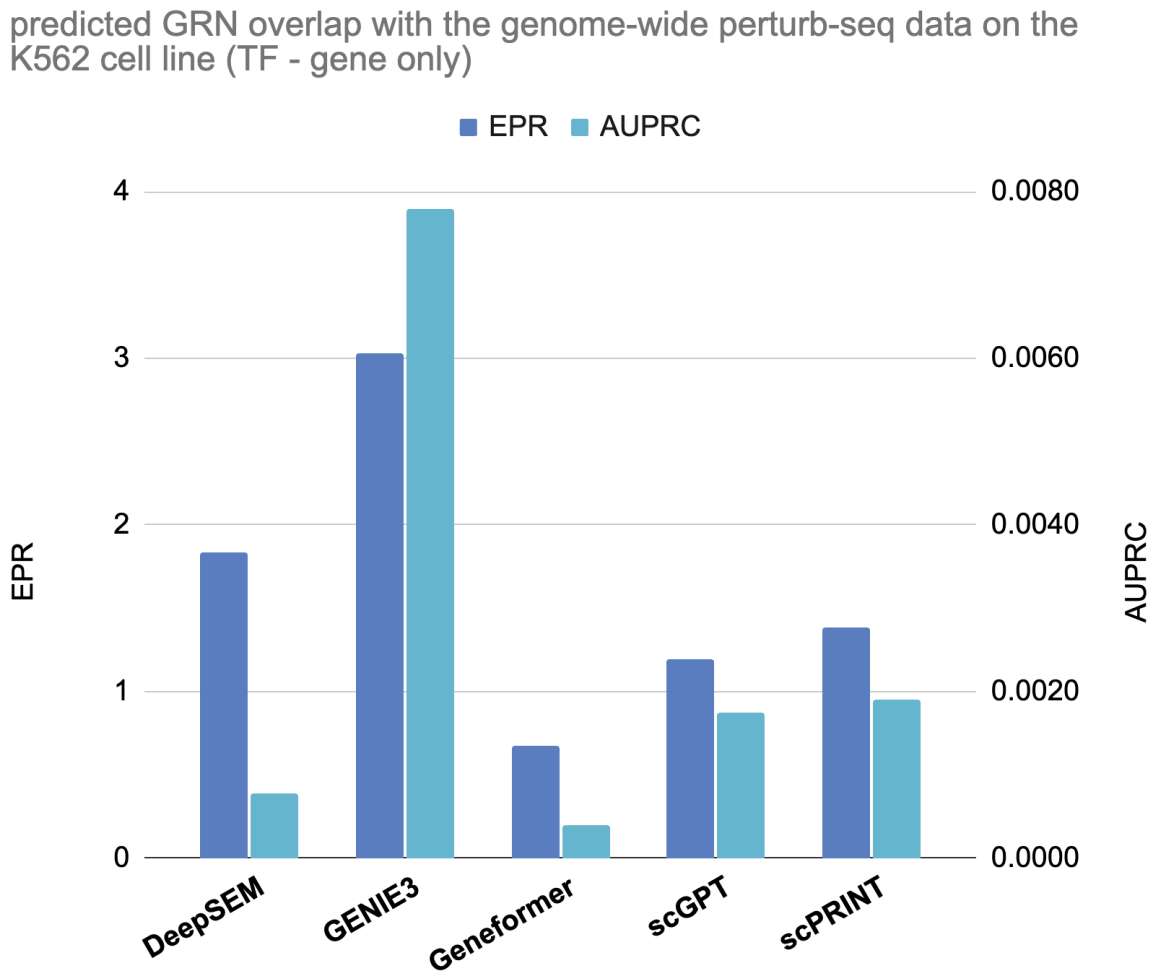


Figure 5.15: Performances of each model's networks on its overlap with the TF-gene-only subset of the genome-wide perturb seq ground truth. It shows that on this task, most foundation models do not perform well. This could be due to the way their attention matrix is normalized. Source data are provided as a Source Data file.

5.2.5 Full denoising results

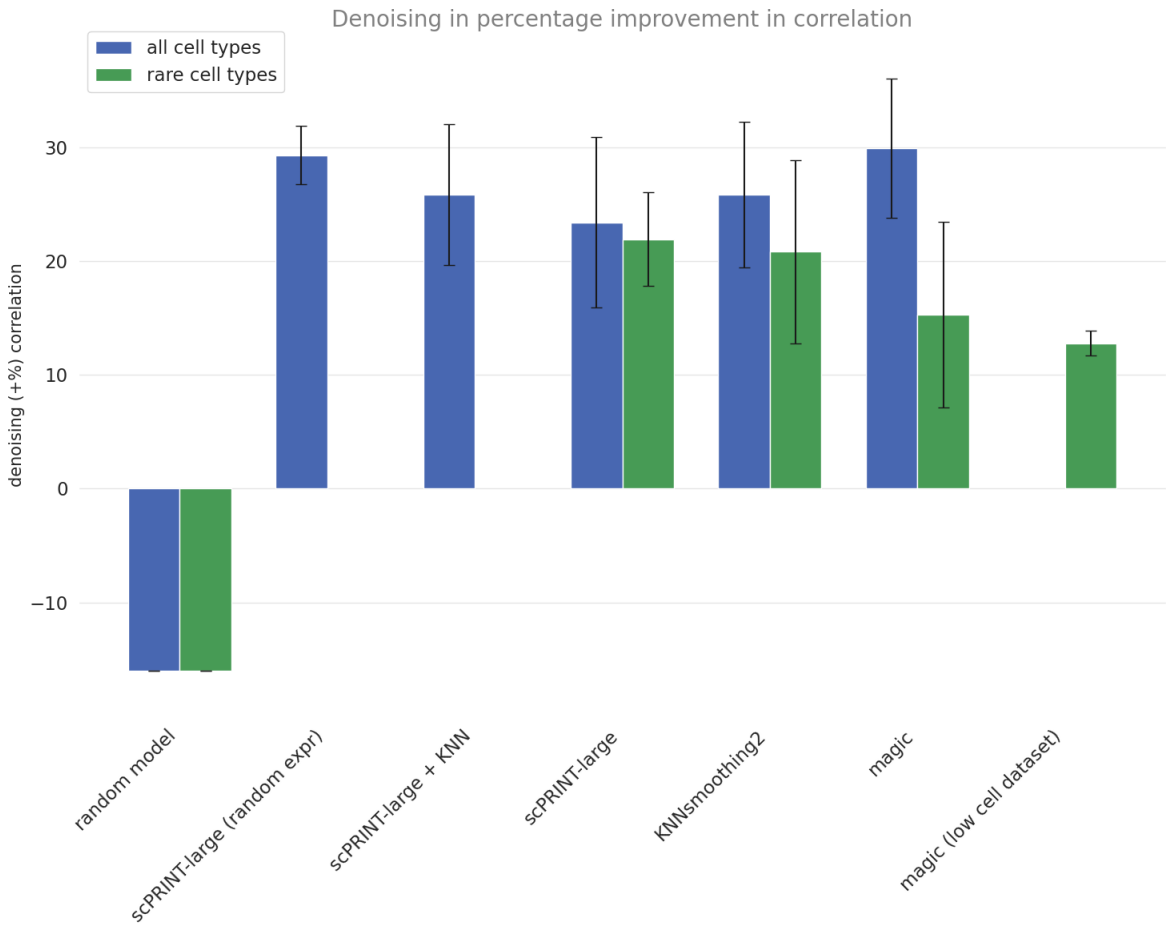


Figure 5.16: Denoising scores, similar to Figure 4A, but over more tools. “Random model” means a scPRINT model without pre-training. “Random expr” means that scPRINT was using a set of 3000 genes in a similar way as done in pre-training: Taking random expressed genes completed with random unexpressed genes if less than 3000 genes are expressed in the cell. “low cell dataset” means that MAGIC was only using the rare cell population for the dataset as presented in the methods. Source data are provided as a Source Data file.

5.2.6 Cell type classification metrics with per-batch split

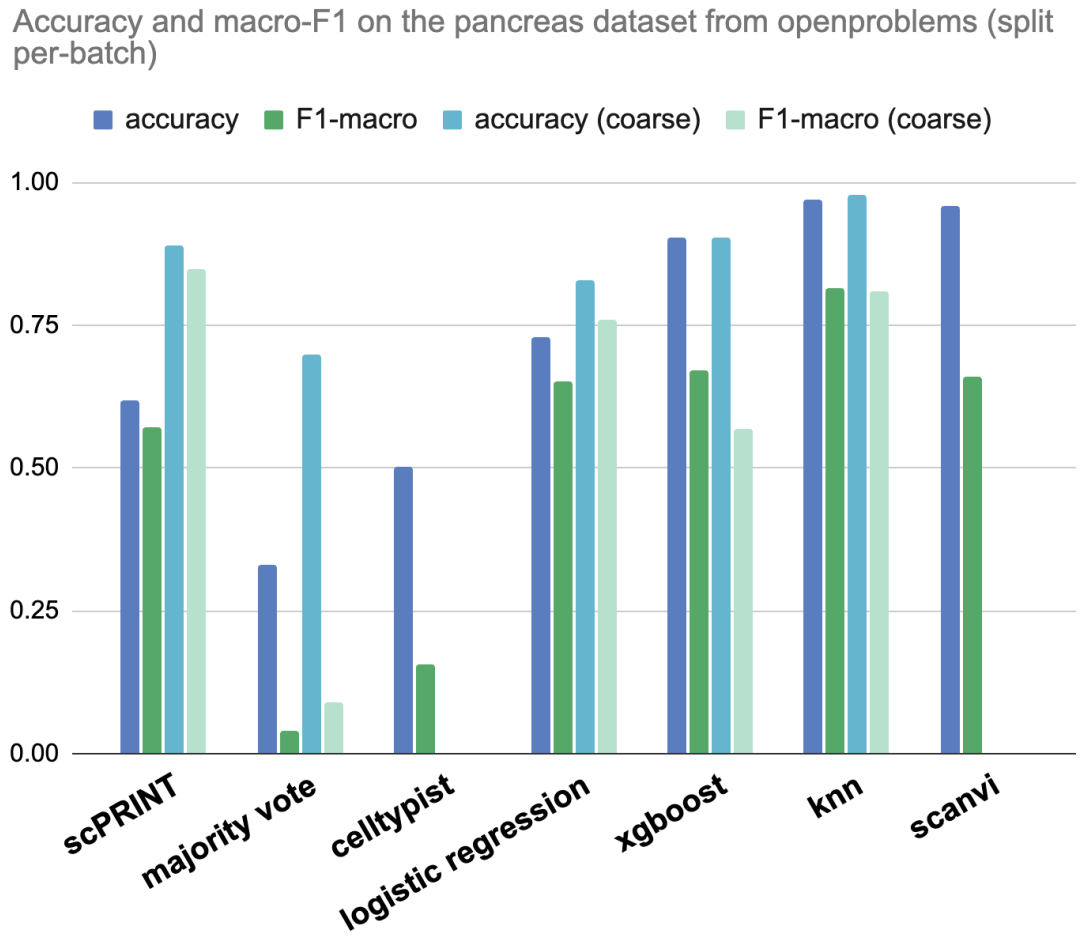


Figure 5.17: Cell type classification scores over the kidney test dataset of openproblems. Same as Figure 4B, but now the trained methods are trained on a subset of the batches representing roughly 70% of the dataset. The performance is lower in this context, and scPRINT, majority voting, and Celltypist's performance are not changing. Source data are provided as a Source Data file.

5.2.7 Full scIB batch correction scores

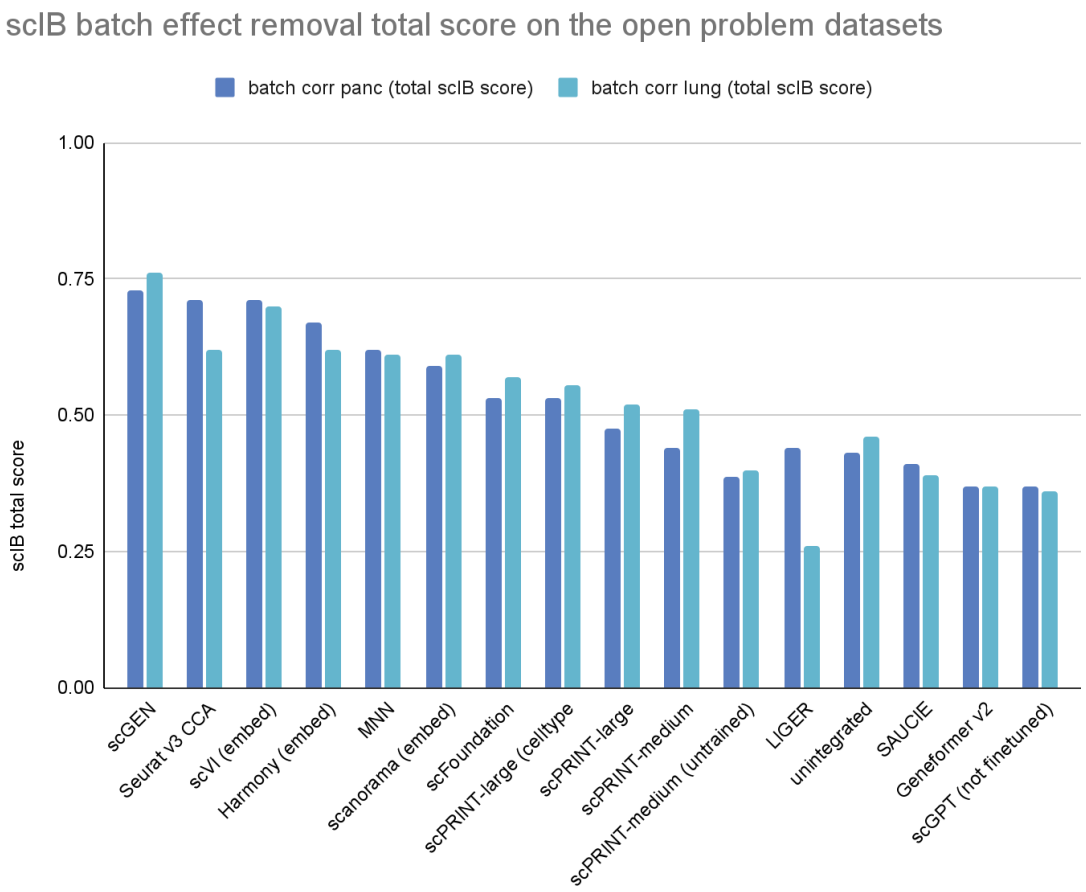


Figure 5.18: scIB benchmarking scores, averaged for the kidney and lung openproblems test datasets. Same as Figure 4C but over more tools. Cell type logits mean that the logits of the cell type classifier have been used as cell embeddings instead of the cell type embedding itself. Source data are provided as a Source Data file.

5.2.8 Full avgBio scores

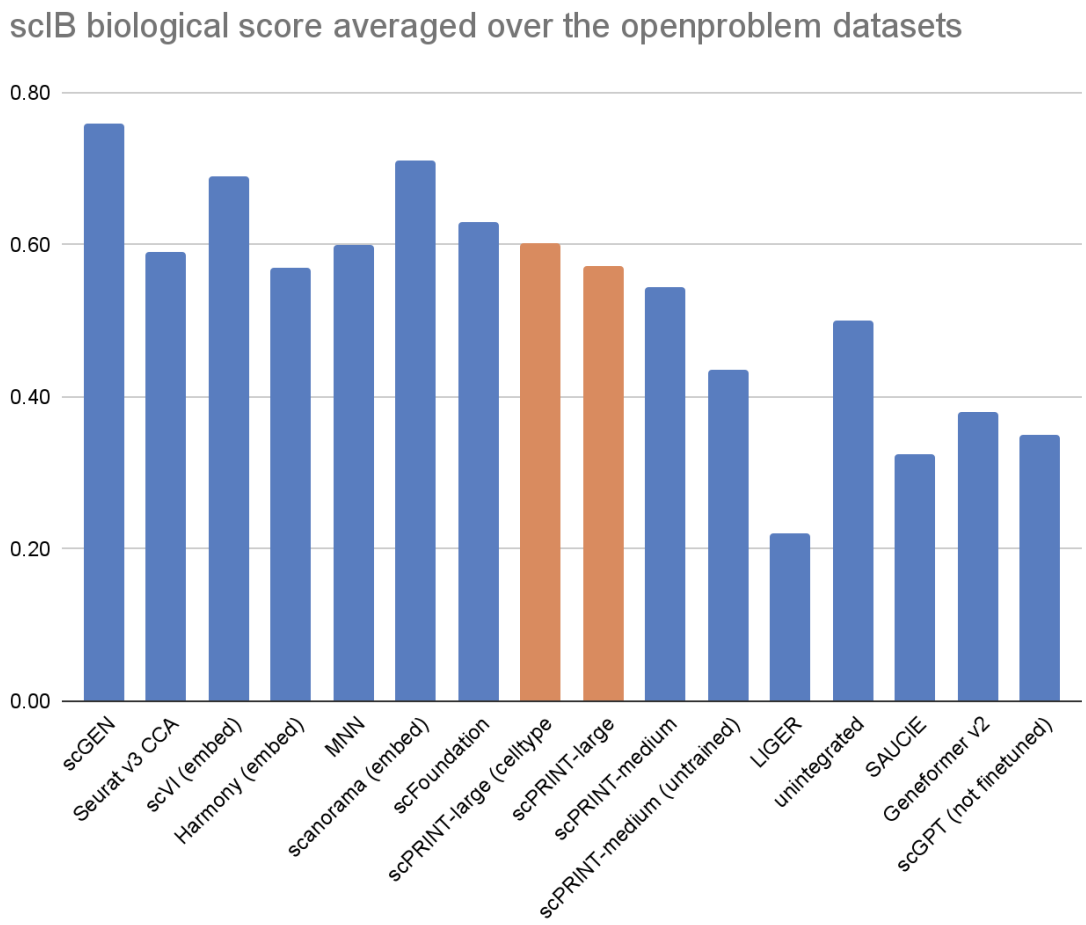


Figure 5.19: The average Biological score of the scIB benchmark averaged over the kidney and lung openproblems test datasets. Same as Figure 4D but over more tools. Cell type logits mean that the logits of the cell type classifier have been used as cell embeddings instead of the cell type embedding itself. Source data are provided as a Source Data file.

5.2.9 In-depth view of the BPH dataset and its scPRINT-predicted annotations

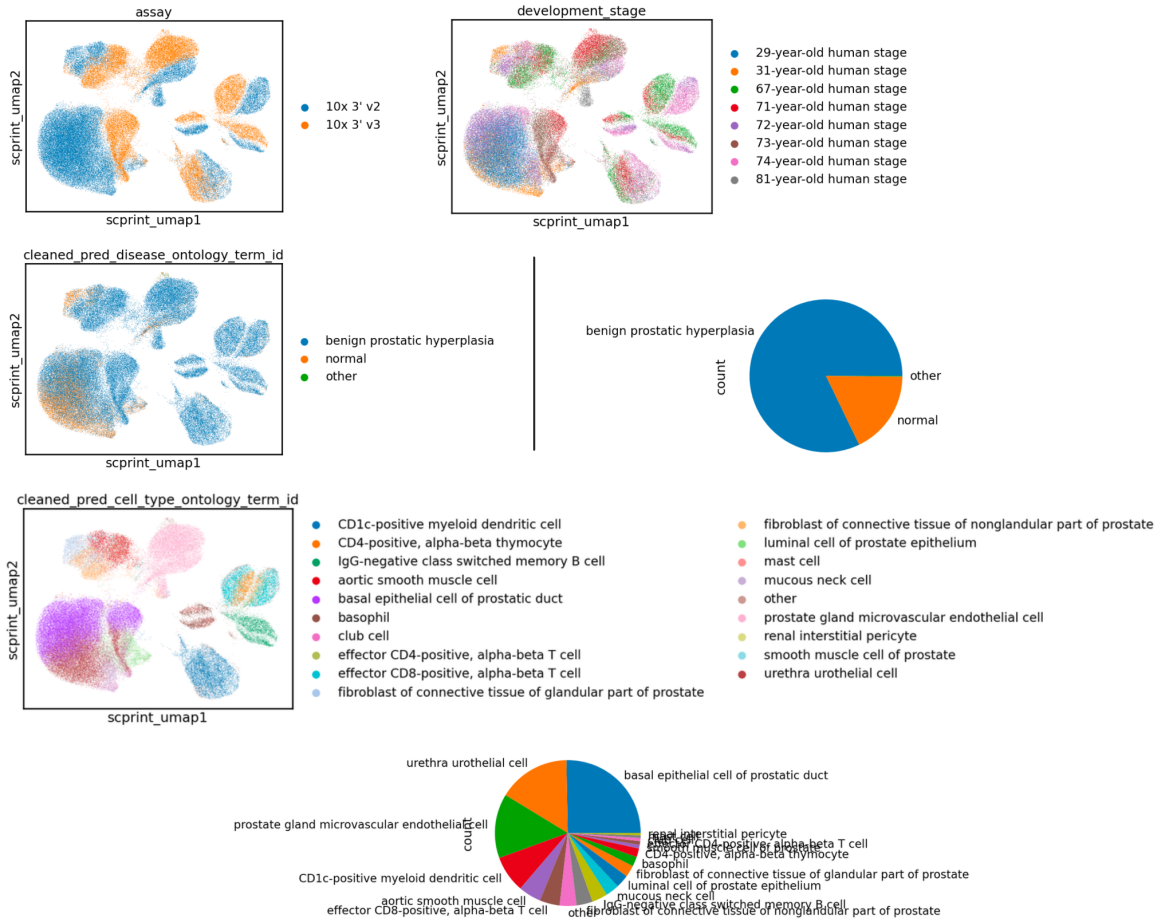


Figure 5.20: Detailed view of the assay, development stage, scPRINT-predicted diseases, and scPRINT-predicted cell types. Predicted diseases and cell types have been "cleaned" following the strategy presented in Figure 5. We also add pie charts of the relative abundance of each predicted label.

5.2.10 Differential expression analysis of the B-cell cluster vs the rest of the cells in the BPH dataset

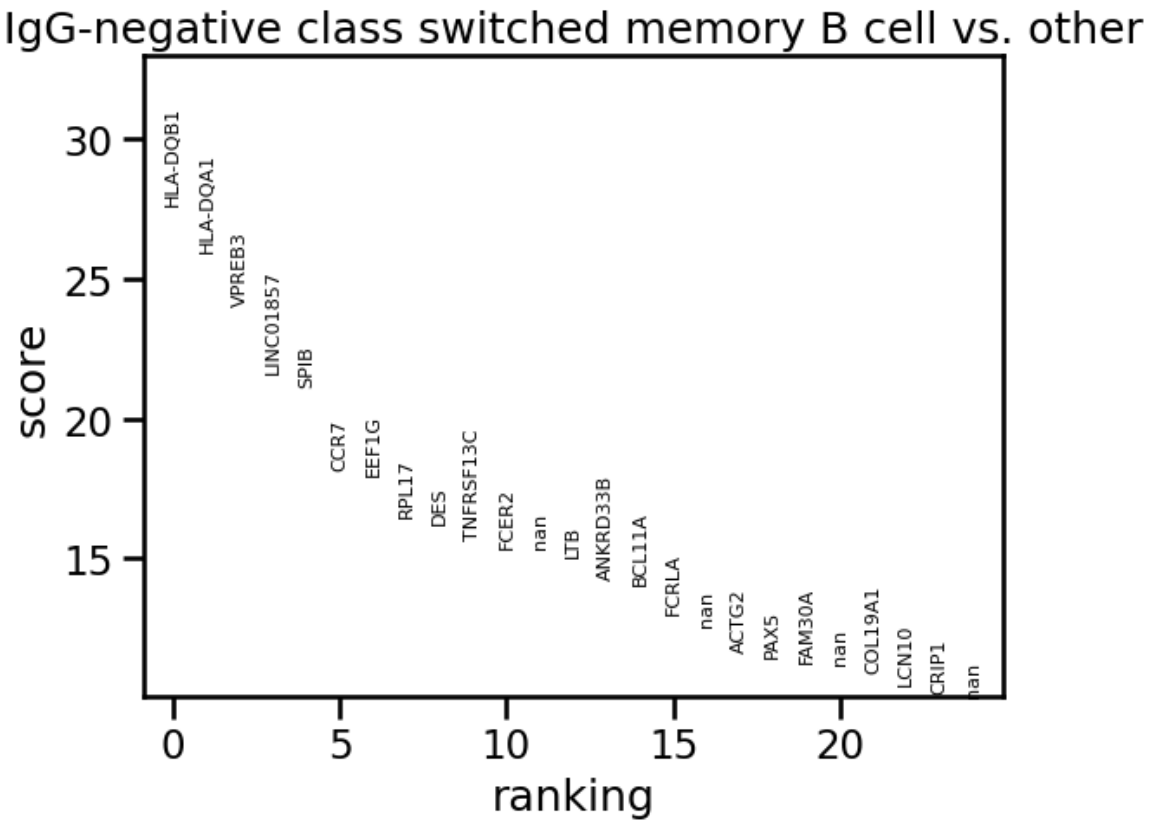


Figure 5.21: Top genes of the differential expression analysis of the scPRINT inferred B-cell cluster in vs the rest of the cells in the BPH dataset.

5.2.11 Gene enrichment comparison in the PAGE4 GN

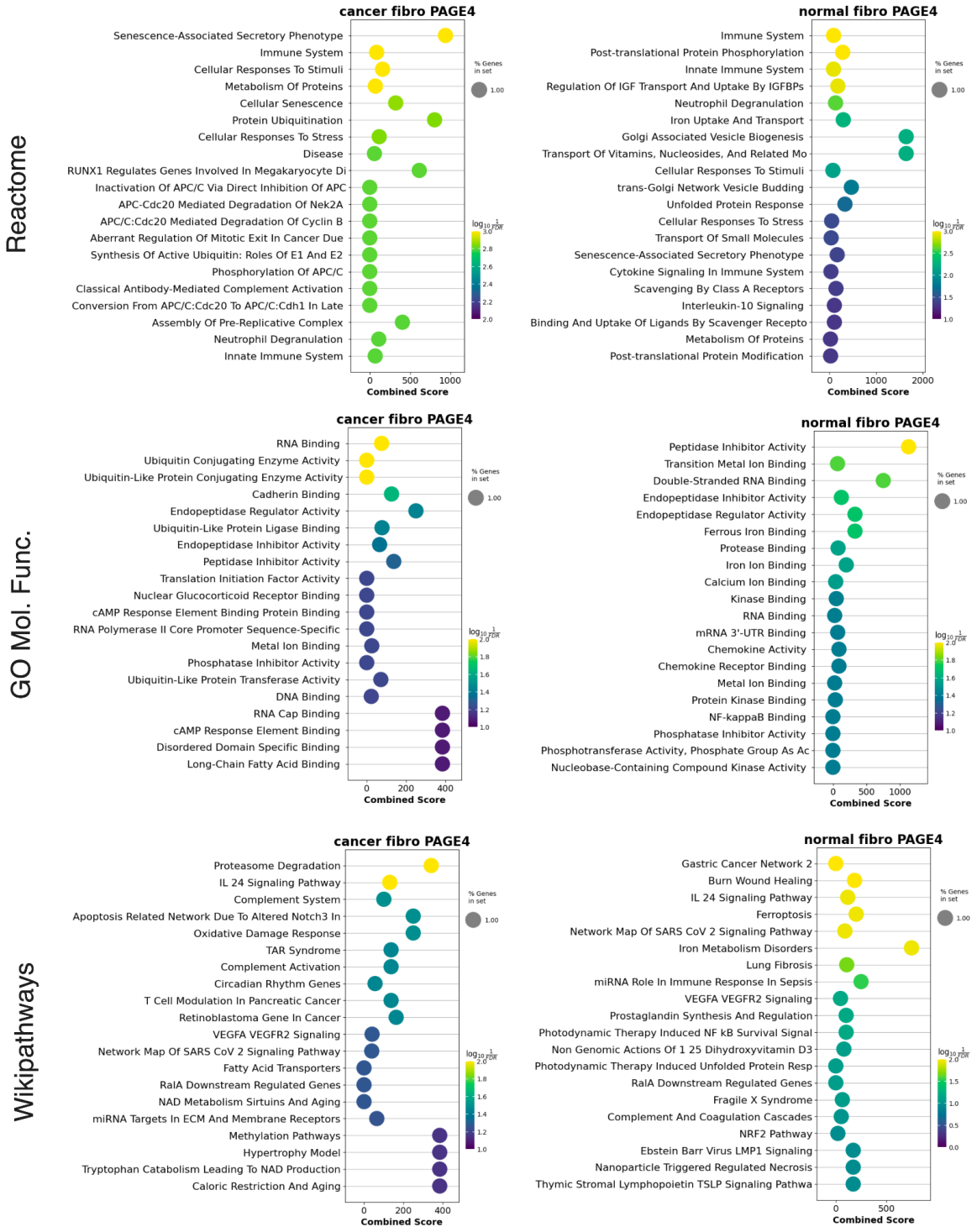


Figure 5.22: Comparison of the top 20 most enriched terms in Wikipathways, GO molecular function, and Reactome for the 40 most connected genes to PAGE4 in both BPH-associated and normal fibroblast GNs inferred by scPRINT.

5.2.12 Gene Network enrichment comparison between the BPH and normal fibroblast on their Louvain communities

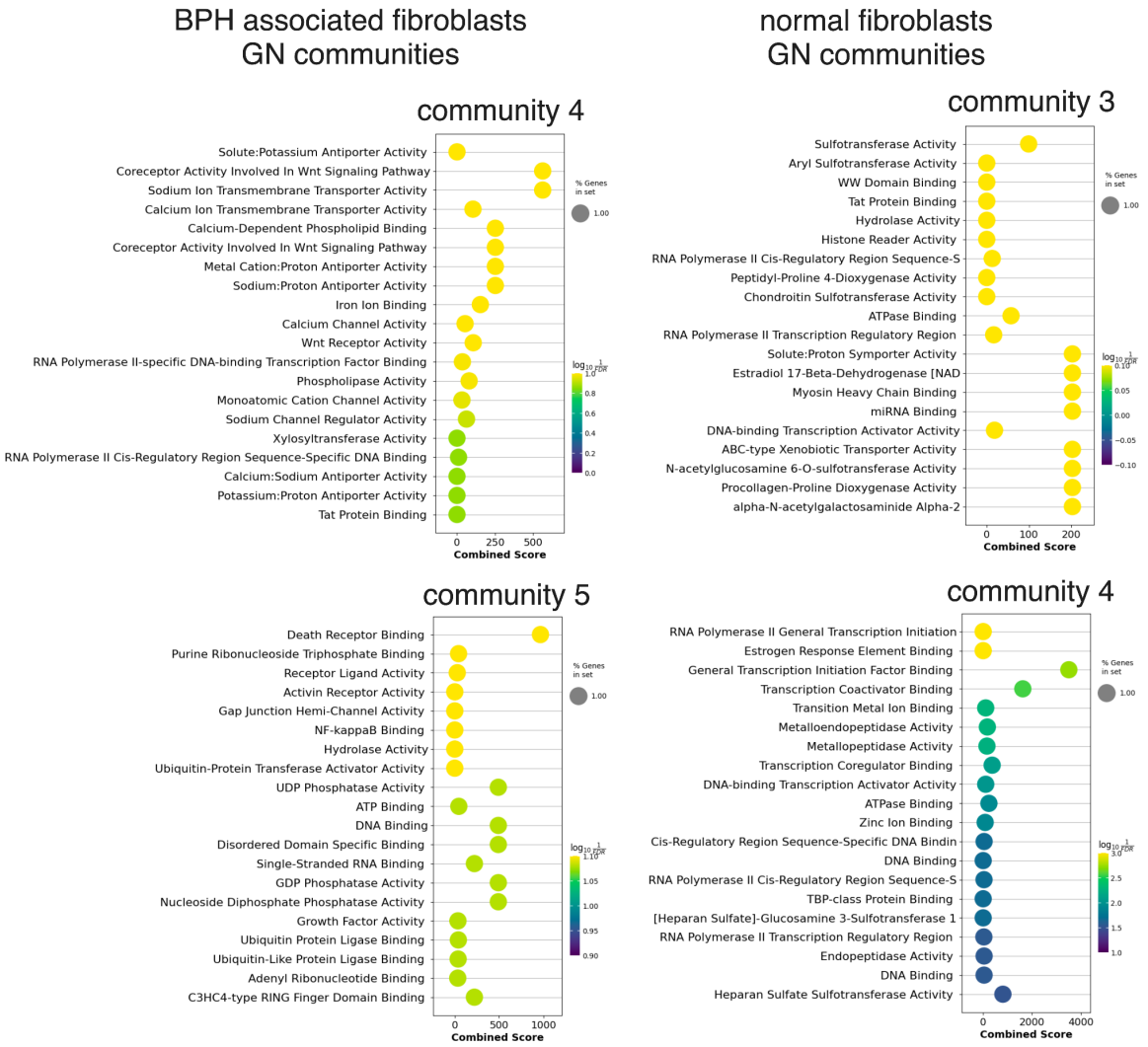


Figure 5.23: Dotplot of the top 20 GO Molecular function gene sets enriched in the Louvain communities of the BPH and normal fibroblast's Gene Networks.

5.2.13 Graphical Model

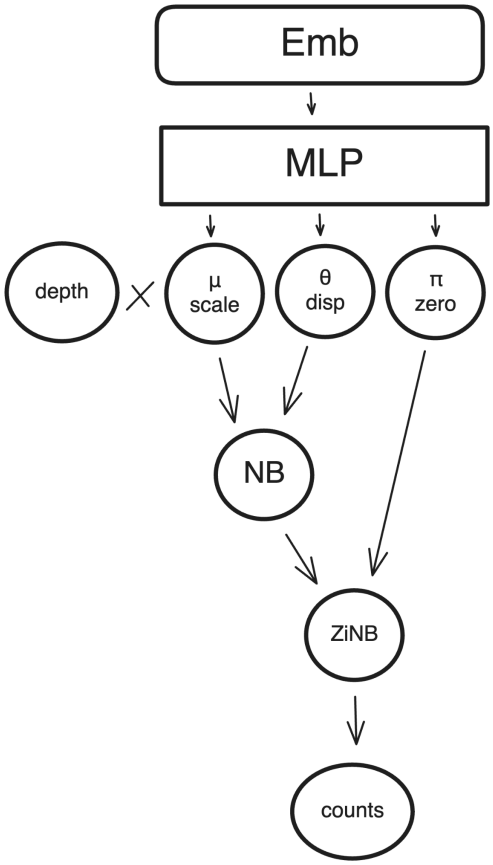


Figure 5.24: Schematic representation of the zero-inflated negative binomial graphical model of the expression decoder. We generate three values μ, θ, π which are used to model a distribution. We also multiply the μ with the depth (or total count) over the cell.

5.2.14 Hierarchical classifier

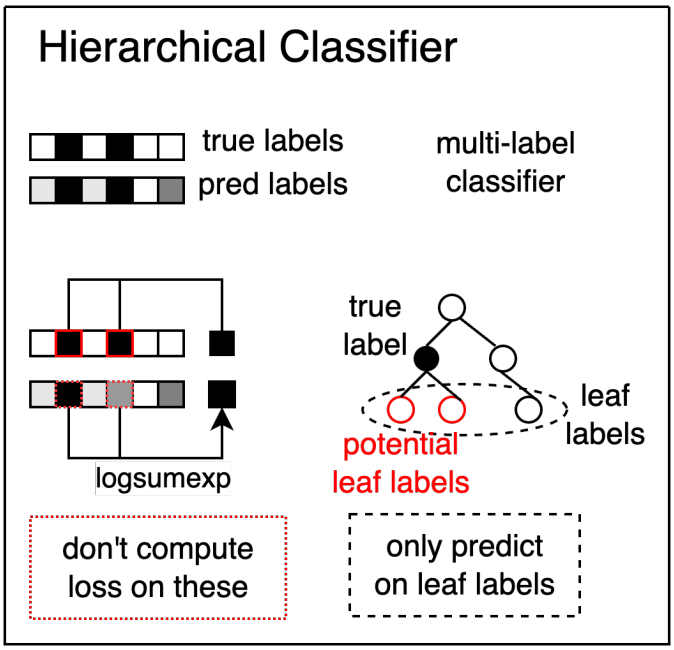


Figure 5.25: Schematic representation of the hierarchical classifier and its behavior during training. We can train on labels not predicted by the classifier as long as they are parent to one of the predicted labels in the ontological tree.

5.2.15 Detailed representation of the bottleneck learning procedure

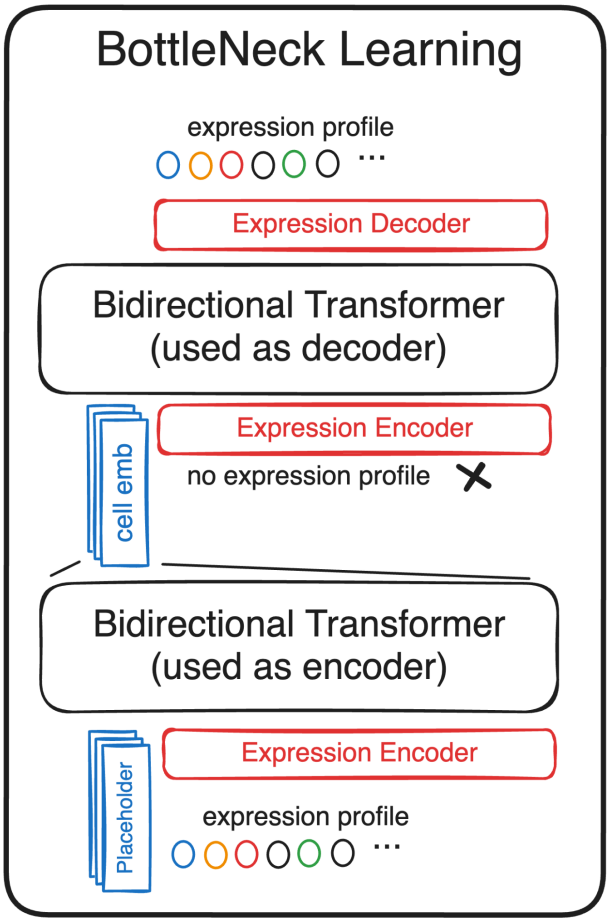


Figure 5.26: Schematic representation of the bottleneck learning procedure where scPRINT's Bidirectional Transformer Encoder is used both as the "Encoder" and "Decoder" of an auto-encoding (AE) bottleneck learning scheme.

5.2.16 Schematic representation of our dataloader

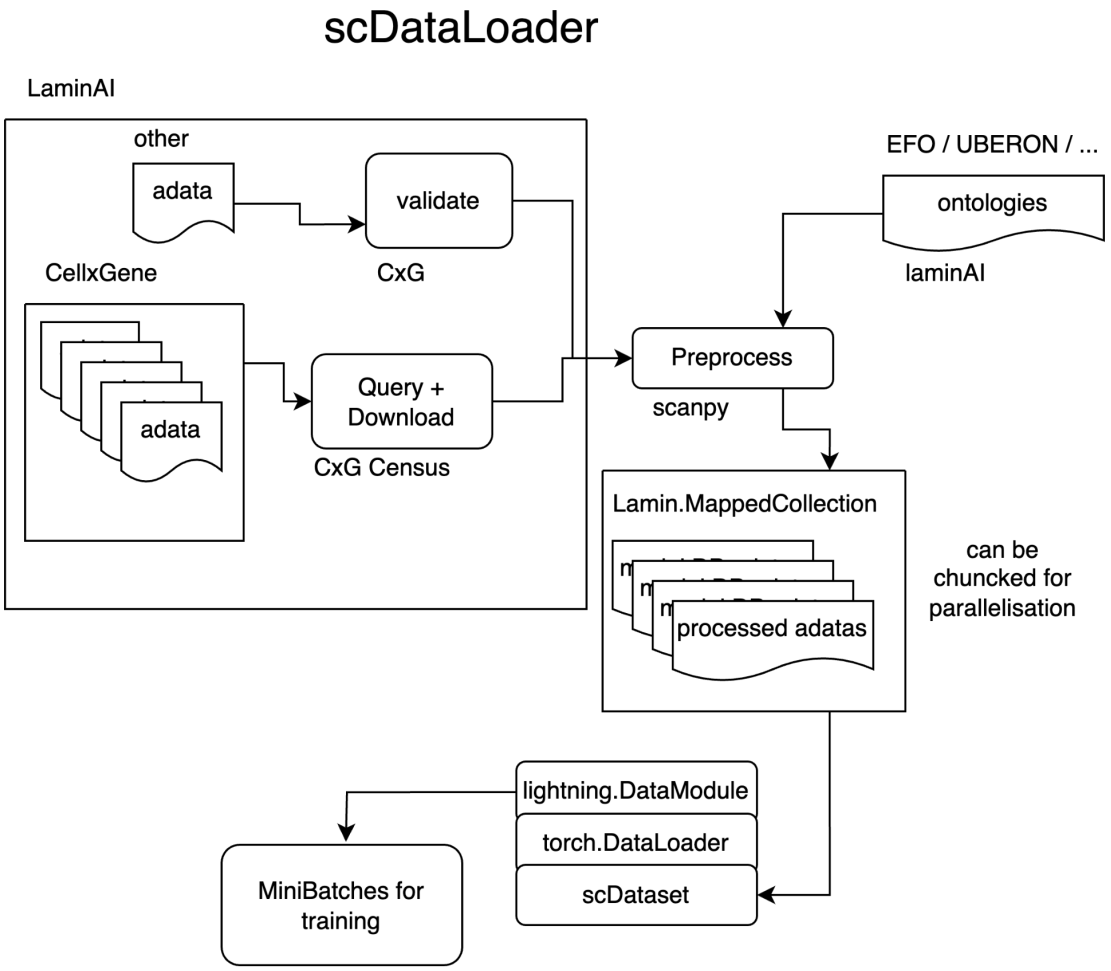


Figure 5.27: Using Lamin.ai, we download and preprocess all cellxgene datasets as AnnDatas. We can also add and validate other expression datasets using lamin.ai. Based on lightning's datamodule framework, torch's dataloaders, our weighted random sampler, and lamin.ai's mapped collection, we can then sample minibatches for pre-training across thousands of datasets and millions of cells with weighted random sampling.

5.3 Supplementary Tables for scPRINT-2

5.3.1 Detailed version of the additive benchmark

	names	GPU time per epoch	denoising score	embed & batch corr. lung	embed & batch corr. pancreas	cell type prediction lung	cell type prediction pancreas	gene regulatory network prediction OR gwps	OR omni.	AUPRC gwps	AUPRC omni.		run id
Base	ross seeds (masking; ZINB loss; + continous expr. emb.; classif. + generative task)	130	2,5 +- 1,5	50.1 +-2.1	42.3 +-2	50 +- 8	45 +- 5	3,8 +-0.7	1,4 +- 0.3	0,044 +-0.002	0,00165 +-0.00015	efwkxesx, te0uwaz1, jsls4j6n, hobjefdj, r18nhnuz	blooming-dew-714, driven-valley-750, summer-deluge-783, silent-poltergeist-843, unraveling-pumpkin-841
	medium model	180	3	50.3	42.8	65.1	60.2	4.7	1.15	0.048	0.00188	p0znio7y	solar-durian-637
	negative control	0	-24	40.2	34.2	0	0	1	0.8	0.021	0.00123		
architecture	no dropout	130	-1	49.9	45.7	47	55.1	3.9	1.5	0.044	0.00157	dipgk9u5	expert-feather-748
	large classifier	130	0	52	43	53	49	4.4	1.8	0.046	0.00148	9q261ics	chocolate-snowball-718
	MVC	130	1	51.7	44.4	54.8	45.8	3.9	1.4	0.043	0.00143	yfvvk4cb	celestial-sun-749
	no decoders / generation	90	2	52.2	44.9	53.2	47.7	4	3.5	0.044	0.00166	z3abxa21	northern-voice-777
	XPressor	160	-4	50.4	45.6	47.3	46.8	4.2	2	0.046	0.00163	dsemm200	crimson-wildflower-791
data	only Tahoe	130	0	40	33	0	0	4.8	0.5	0.0041	0.00104	mxu0p3fs	young-bush-669
	CZI + Tahoe (denoising)	130	16.1	53.9	43.1	52.6	51.1	4.5	1.9	0.046	0.00143	nmc2l1gf	lurking-cat-846
	CZI	130	1	52.3	43.1	47.8	49.1	3.8	2	0.043	0.00162	4u5c4plu	radiant-oath-793
	all databases (denoise)	140	-4	48.6	39	44.9	40.7	3.6	1.3	0.043	0.00157	ujzjjsi3	macabre-apparition-844
	200 human datasets only	130	0	49.3	43.4	40.3	50	3.5	1.8	0.044	0.00166	c60vguww	playful-frost-804
	sampling without replacement	130	0	50	45	36	34	4.5	2.4	0.045	0.0016	lg84geoq	autumn-aardvark-702
	cluster-based sampling only	130	2.7	43.3	42	49.4	40.2	3.5	1.3	0.042	0.00157	s8wwlmxr	rosy-firefly-805
	meta-cell	130	21	52.8	47.7	53.6	51.3	3.4	1.7	0.04	0.00155	gp90j8vn	unique-dawn-806
attention	softpick (larger context)	130	3.1	50	41	53.8	44.6	3.6	1.8	0.042	0.00156	s6alkcvp	eldritch-fang-834
	criss-cross (larger context)	90	5.6	51.2	42.5	42.4	43.7	x	x	x	x	u5udxv4v	uncanny-raven-835
	hyper (denoise, larger context)	160	2	50.1	43.4	42.1	40.6	3.7	0.6	0.04	0.00115	l44og0s3	silver-grass-803
loss	contrastive learning (masking + denoising)	130	21.4	49	41.5	39.5	40.4	4	1.3	0.043	0.00149	wcg8g3hr	northern-frog-797
	elastic cell similarity	130	2.5	52.7	43.1	44.8	34.9	4.3	1.6	0.046	0.00167	qn2tyayf	hopeful-monkey-796
	no embedding ind loss	130	2.2	51.6	43	50	50	4	2.3	0.043	0.00156	4v84b9nm	devoted-wave-795
	ZINB+MSE (denoising)	130	25.5	51.3	48	49.2	42.9	3.4	1.3	0.04	0.00163	bv1r4d3h	firm-silence-747
	MSE	130	-4	54	46	62	43	3.3	1.2	0.042	0.00166	mnk73zbd	generous-dawn-666
	VAE compressor	160	3	51	42	38	27	4.2	1.7	0.044	0.0016	jwkcrxb9	wild-terrain-694
pretraining task	**var. context (larger context)**	170	29.1	53	46	52.9	52.2	3.1	1.2	0.038	0.00146	44p3fv3v	apricot-snowflake-756
	TF masking	130	2.6	49.8	42.8	49.8	42.8	3.7	2.3	0.043	0.00169	8vmjnnsb	winter-meadow-772
	denoising	130	21	52.6	45.1	50.9	54.5	3.6	1.3	0.043	0.0016	bk37305v	efficient-firebrand-753
	no classification	130	3	50	40	0	0	3.9	1.2	0.043	0.00129	oxlxtzim	copper-frost-625
	adv. classifier (+larger classif)	130	1	52	42	48	43	4.1	1.6	0.044	0.0014	2tzkv7m8	sunny-morning-629
input	sum normalization (denoise)	130	12.8	45.6	46.5	21.4	22.9	2.4	1	0.029	0.00136	ldh1fw8d	snowy-galaxy-744
	no random level of denoising	130	19	54.1	45.3	50.7	45.2	3.6	2	0.041	0.00179	0ayw97iw	divine-monkey-798
	binning	130	0	51.8	45.5	58.4	52	4.2	1.3	0.047	0.00162	op7at8xm	twilight-breeze-874
	GNN expression encoder	150	44	48	42	38	35	4	1.4	0.042	0.00128	bv6d9wpl	balmy-totem-727
	using only expressed genes	130	1.2	52.2	42.9	53.2	40.1	3.8	1.3	0.043	0.00157	xz238fyr	sage-snow-873
	without gene location	130	3.4	36.2	35	4	5.9	4.8	1.5	0.048	0.0017	r2n83z4k	youthful-snowflake-792
	learn gene emb (denoising)	130	20.9	51.7	45.1	49.7	46	3.2	1.7	0.041	0.00154	npayct6q	jumping-night-755
	fine-tuned ESM3	130	21.4	51.5	42.8	55.6	44.2	3.7	1.4	0.042	0.00181	fkcgp56s	not-snowflake-755
Main	small model (V2)	1820	44	53	49	46	47	3.5	1.6	0.041	0.0015		honest-vortex-815
	medium model (V2)	5600	x	x	x	x	x	x	x	x	x		x
	medium model (V1)	520	20.9	52.6	45.6	61.8	57.6	3.4	2.2	0.041	0.0017		bewitched-poltergeist-857
	small model (V1)	160	31.7	52.4	50	44.7	44.7	3.6	1.5	0.042	0.00138		dry-smoke-852

Table 5.13: Detailed version of the additive benchmark, listing every value.

5.3.2 Detailed scIB biological conservation scores on the xenium dataset

Bio Metric	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI Bio conservation	**Aggregate score**
X_pca	0.446	**0.255**	**0.036**	0.382	0.955	**0.415**
scprint_emb	0.468	**0.376**	**0.125**	0.383	0.979	**0.466**

Table 5.14: Detailed scIB biological conservation scores on the xenium dataset. scPRINT vs PCA on expression. ScPRINT performs better, likely by denoising the expression.

5.3.3 Detailed scIB scores on the unseen species integration task

	Bio conservation					Batch correction					Aggregate score			Recomputed scores using the same elements		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	BRAS	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	**Total**	Batch correction	Bio conservation	**Total**
scprint_2 ft (cell_type emb)	0.64	0.81	0.75	0.69	1.00	0.69	0.00	0.01	0.91	0.00	0.32	0.78	**0.60**			
scprint_2 ft	0.55	0.72	0.63	0.56	1.00	0.58	0.00	0.00	0.65	0.00	0.25	0.69	**0.51**			
scprint_zeroshot (cell_type emb)	0.57	0.41	0.31	0.53	0.98	0.65	0.00	0.77	0.00	0.28	0.56	0.45	**0.49**			
scprint_zeroshot	0.49	0.00	0.00	0.49	0.68	1.00	0.86	0.00	0.20	1.00	0.61	0.33	**0.44**			
random	0.54	0.23	0.14	0.50	0.97	0.80	0.00	0.00	0.72	0.00	0.30	0.48	**0.41**			
no integration (pca)	0.57	0.21	0.10	0.36	0.99	0.69	0.00	0.00	0.60	0.00	0.26	0.44	**0.37**			
saturn	0.81	0.97	0.49	0.13	1.00	0.79	0.13	0.05	0.91	0.92	0.62	0.92	**0.79**	0.56	0.68	**0.63**
scGen	0.60	0.77	0.49	0.23	0.99	0.88	0.23	0.16	0.91	0.92	0.85	1.00	**0.68**	0.62	0.61	**0.62**
Seurat v4 CCA	0.58	0.57	0.48	0.23	0.97	0.84	0.23	0.13	0.90	0.89	0.73	0.92	**0.50**	0.60	0.57	**0.58**
SAMap		0.62		0.01	0.98	0.91	0.01	0.22	0.74		0.60	1.00	**0.47**	0.47	0.53	**0.51**
scVI	0.51	0.55	0.50	0.23	0.95	0.83	0.23	0.09	0.91	0.98	0.80	0.93	**0.47**	0.61	0.55	**0.57**
BBKNN		0.56		0.11	0.99	0.82	0.11	0.05	0.82		0.31	0.66	**0.41**	0.45	0.55	**0.51**
Scanorama	0.56	0.54	0.49	0.27	0.96	0.76	0.27	0.09	0.84	0.93	0.59	0.79	**0.37**	0.58	0.56	**0.57**
fastMNN	0.52	0.54	0.48	0.09	0.96	0.70	0.09	0.03	0.89	0.86	0.37	0.72	**0.36**	0.51	0.52	**0.52**
Harmony	0.51	0.54	0.44	0.06	0.96	0.70	0.06	0.03	0.86	0.70	0.15	0.70	**0.16**	0.47	0.50	**0.49**

Table 5.15: Full results comparing no integration, random embeddings sampled from the multivariate Gaussian, and different versions of scPRINT zero-shot or fine-tuned, using the merged embeddings or the cell type ones.

5.4 Supplementary figures for scPRINT-2

5.4.1 Illustration of the full scPRINT-2's architecture, input, and output

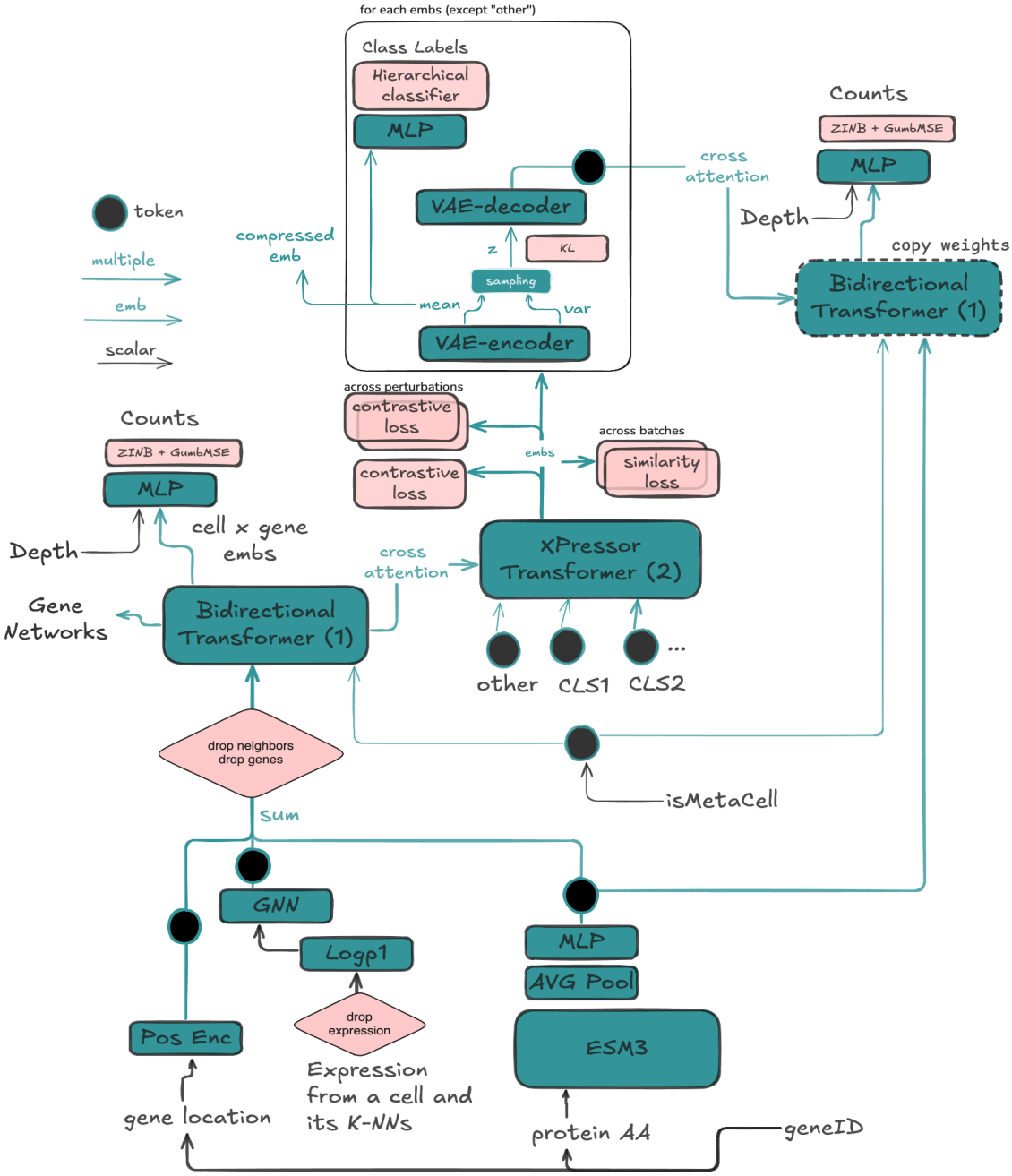


Figure 5.28: Illustration of the full scPRINT-2's architecture, input, and output. In-depth view with all its main different components and the data flow.

5.4.2 Barplot of the F1-macro scores on the label-projection task of the Open Problem benchmark

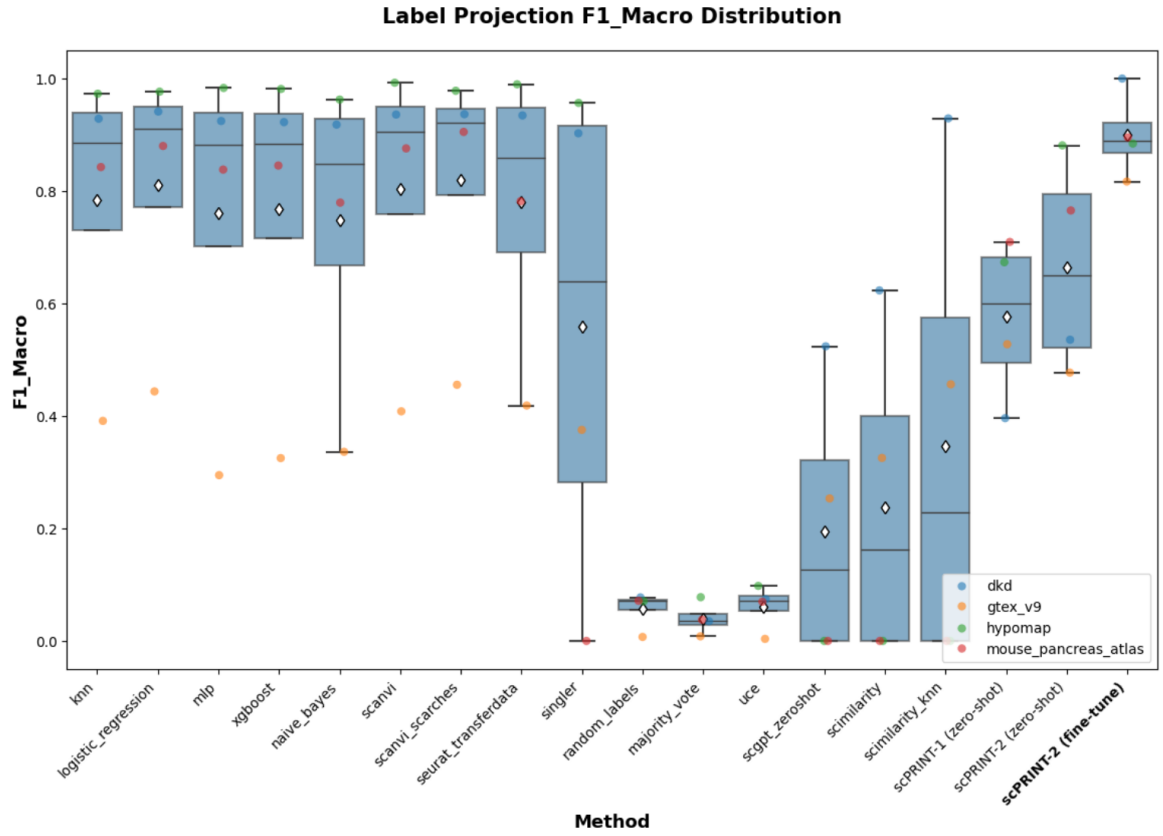


Figure 5.29: Barplot of the F1-macro scores on the label-projection task of the Open Problem benchmark. Comparison of scPRINT-1 and scPRINT-2, zero-shot and finetuned, with all other tested methods.

5.4.3 Heatmap of ethnicity prediction relationship across samples

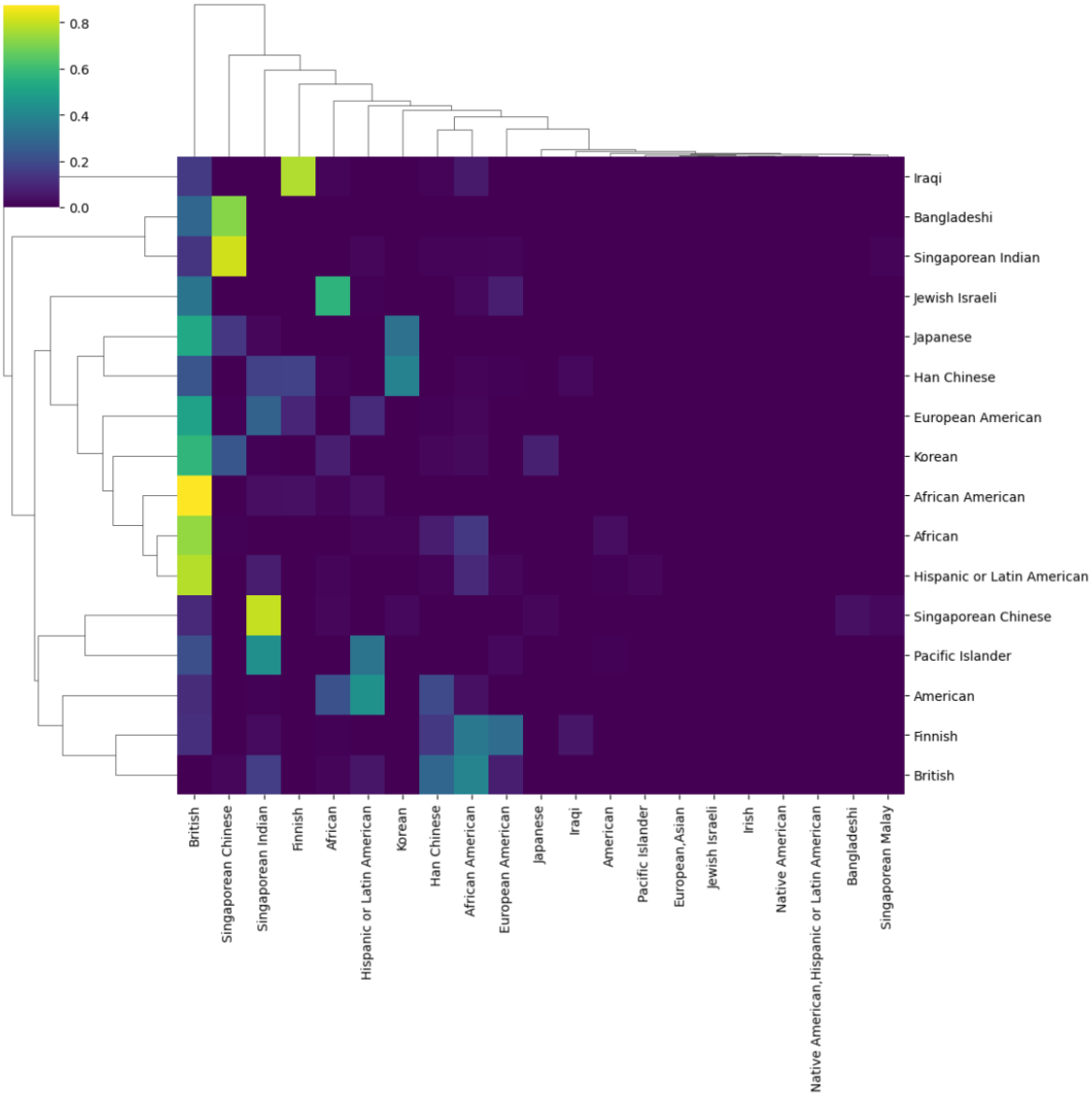


Figure 5.30: Heatmap of ethnicity prediction relationship across samples. Generated using labels predicted as top-1 (x-axis) vs second-best prediction (y-axis) across 10,000 random cells for each predicted label from the scPRINT-2 corpus.

5.4.4 Heatmap of organism prediction relationship across samples

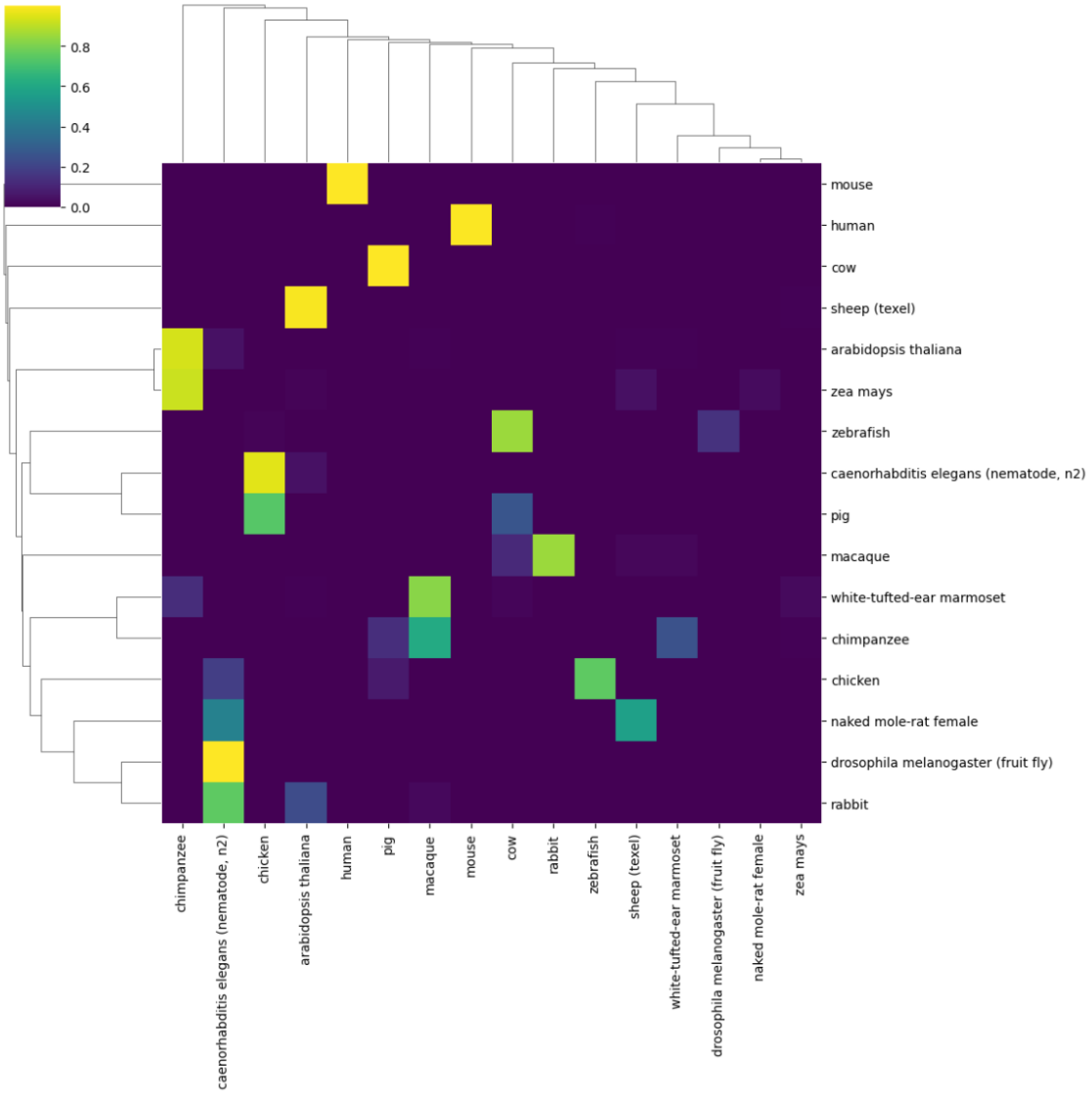


Figure 5.31: Heatmap of organism prediction relationship across samples. Generated using labels predicted as top-1 (x-axis) vs second-best prediction (y-axis) across 10,000 random cells for each predicted label from the scPRINT-2 corpus.

5.4.5 Heatmap of organism prediction relationship using organism embedding similarity across samples

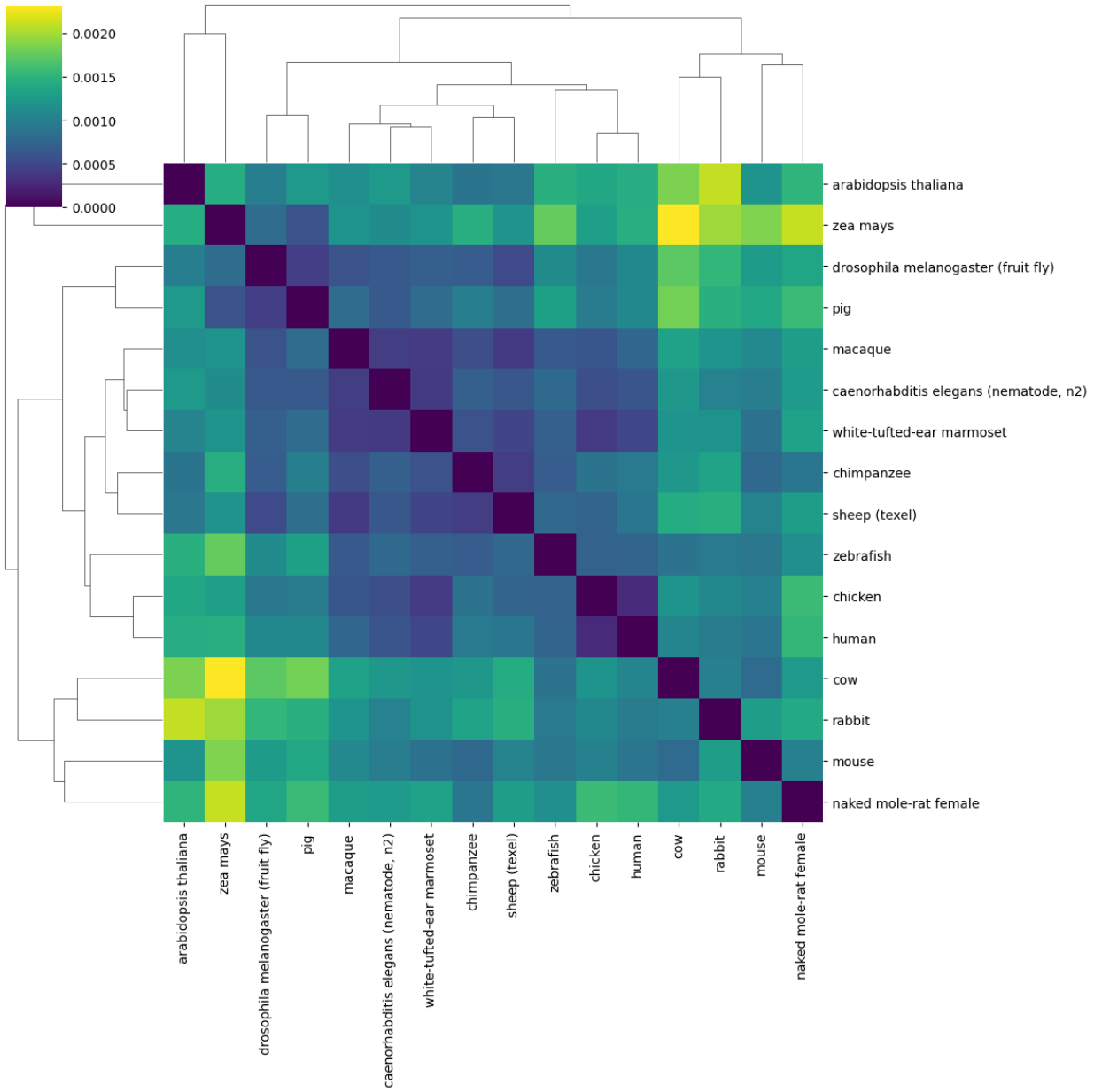


Figure 5.32: Heatmap of organism prediction relationship using organism embedding similarity across samples. Generated by averaging the embeddings for each predicted organism across 10,000 random cells for each predicted label in the scPRINT-2 corpus, and using the L2 distance.

5.4.6 Differential expression plots of the disagreeing cells between scPRINT-2 and ground truth

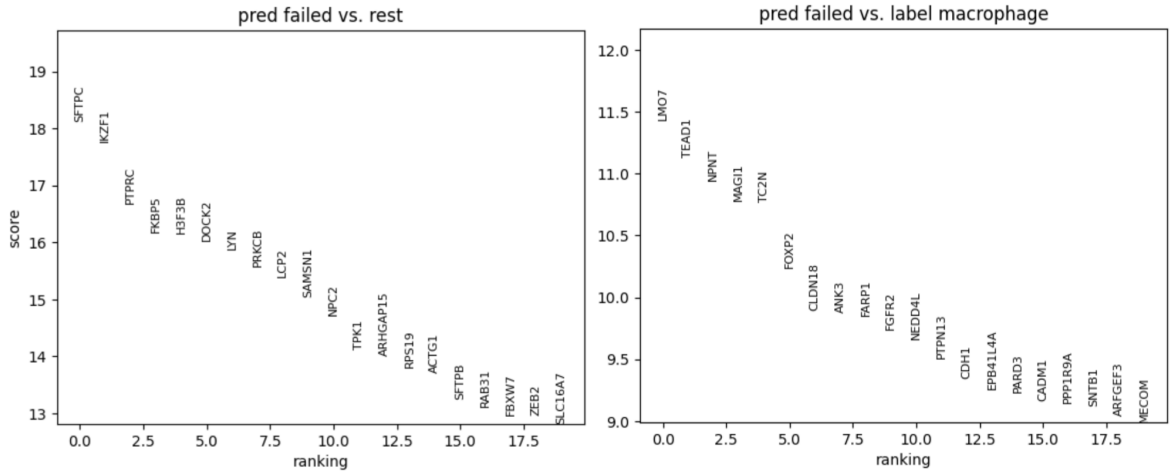


Figure 5.33: Differential expression plots of the disagreeing cells between scPRINT-2 and ground truth. Analysis on the cat/tiger cross-species dataset. “pred failed” is the macrophages labeled as type 2 pneumocytes by scPRINT-2.

5.4.7 Umap of the smart-seq dataset used in the varying context classification task

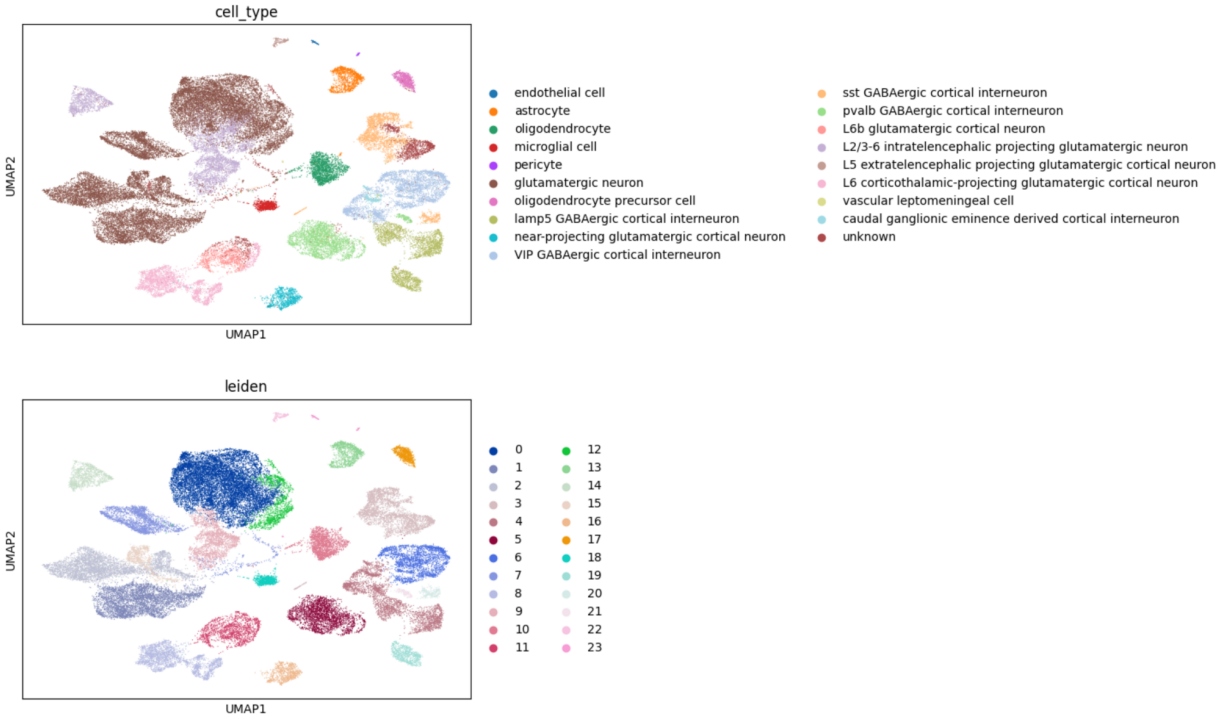


Figure 5.34: Umap of the smart-seq dataset used in the varying context classification task. Cortical areas smart-seq v4 dataset used in results section 2, showing Leiden clusters and ground truth cell types.

5.4.8 Line plot of the classification across varying context length, using the most expressed genes

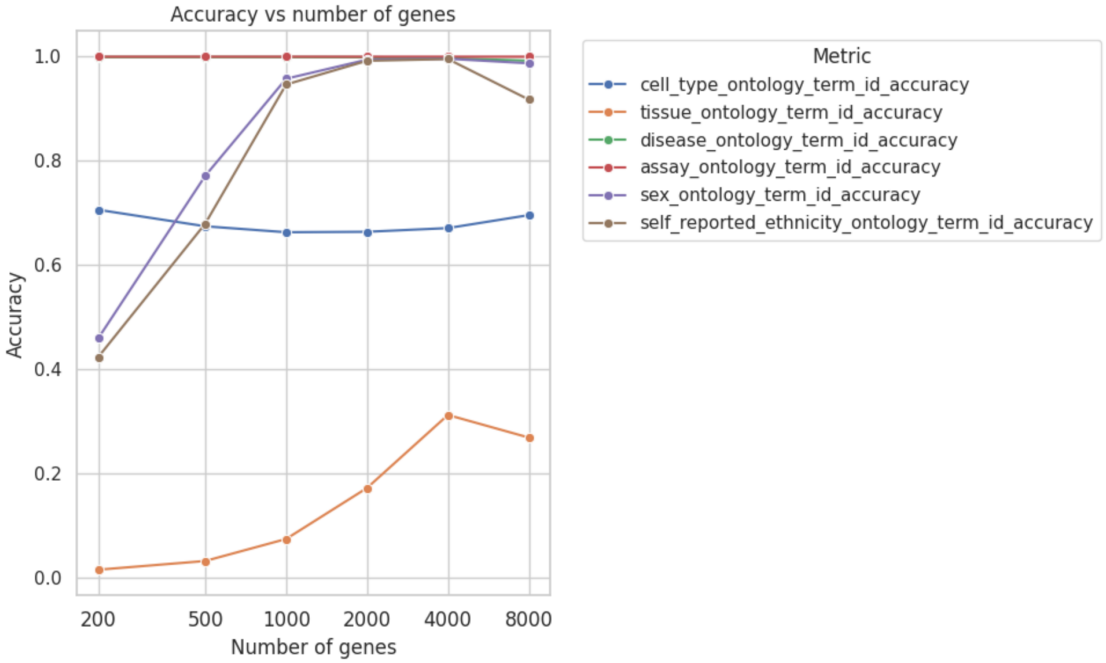


Figure 5.35: Line plot of the classification across varying context length, using the most expressed genes. Each new gene in context is 200 most expressed, then 500, 1000, etc. While cell types are often defined by their most expressed genes, and thus this doesn't change classification accuracy much, other, more complex labels continue increasing in accuracy as context length increases.

5.4.9 Illustration of the multiple perturbations applied to expression data in scPRINT-2

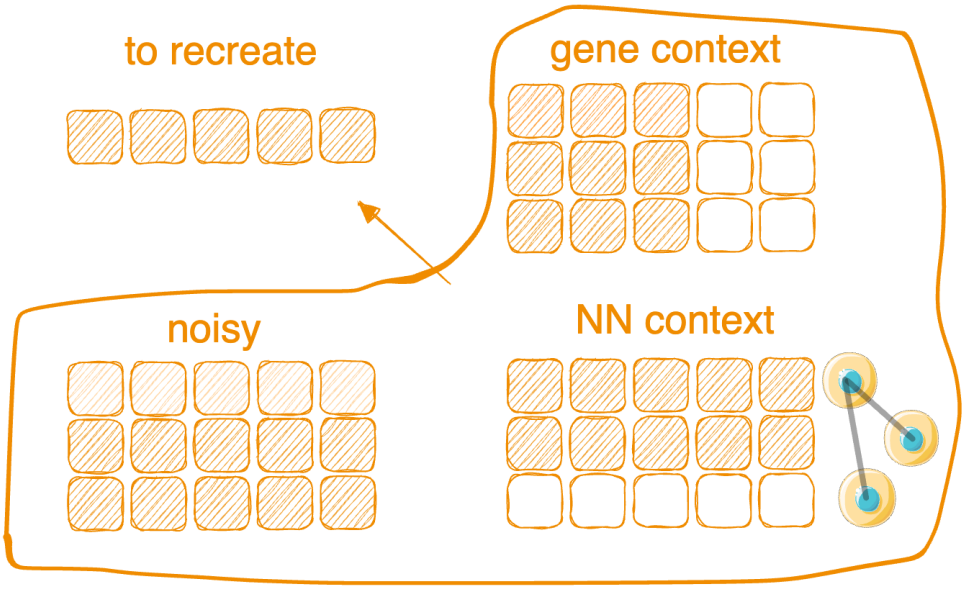


Figure 5.36: Illustration of the multiple perturbations applied to expression data in scPRINT-2. scPRINT can add noise and mask gene expression, modify the number of neighbors, and adjust context lengths.

5.4.10 Distplot of the non-zero count distribution across cells from the three dataset qualities used

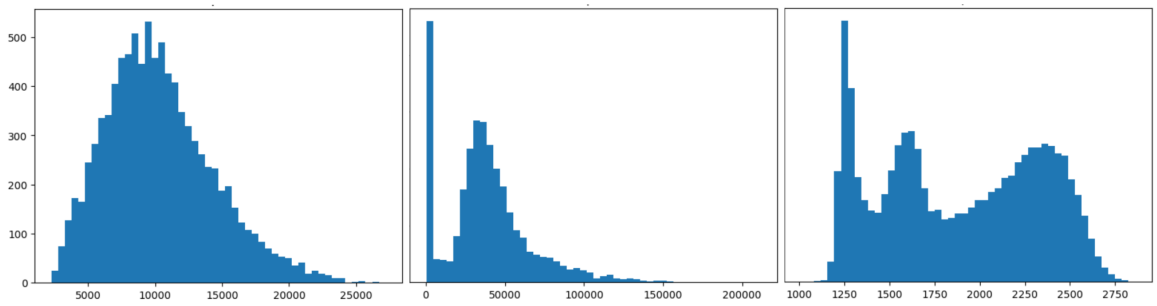


Figure 5.37: From left: good quality; center: excellent quality; right: poor quality datasets used in our denoising benchmark.

5.4.11 Umap over scPRINT-2 and PCA embeddings of the Xenium dataset

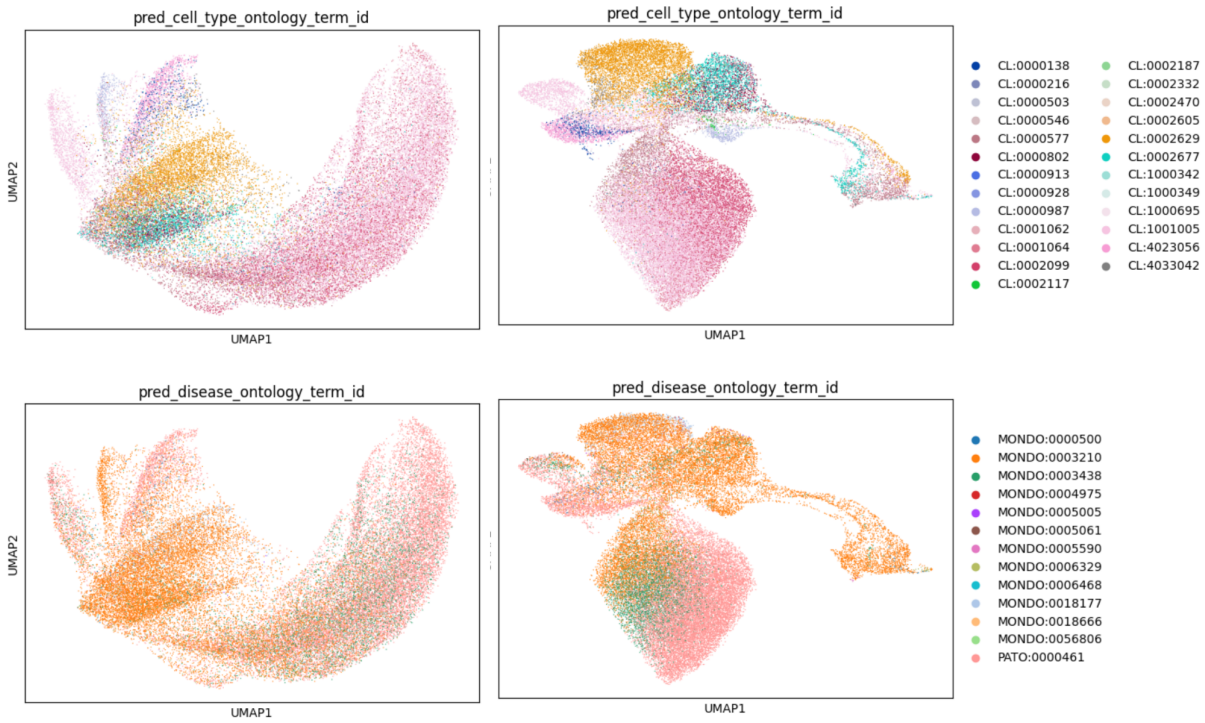


Figure 5.38: Umap over scPRINT-2 and PCA embeddings of the Xenium dataset. Left: raw PCA expression, right: scPRINT-2 embeddings with scPRINT-2 predicted cell types and diseases.

5.4.12 Tangram mapping quality plots

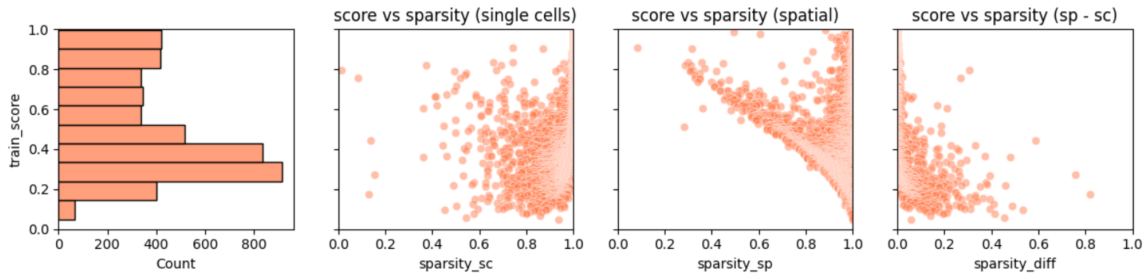


Figure 5.39: Tangram mapping quality plots on the Xenium skin melanoma datasets and 10v3 skin melanoma datasets.

5.4.13 Illustration of scPRINT-2's generative imputation mechanism

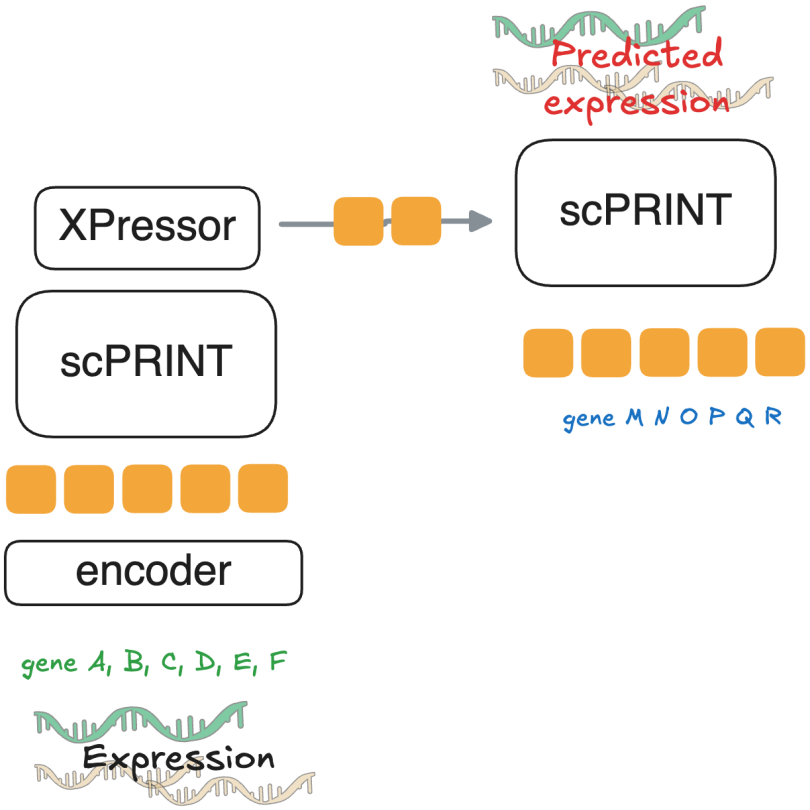


Figure 5.40: Illustration of scPRINT-2's generative imputation mechanism. scPRINT encodes all 5000 measured genes into cell embeddings and decodes them on 5000 different unseen gene embeddings.

5.4.14 Spatial plot of the Xenium melanoma dataset with scPRINT-2 predicted cell labels

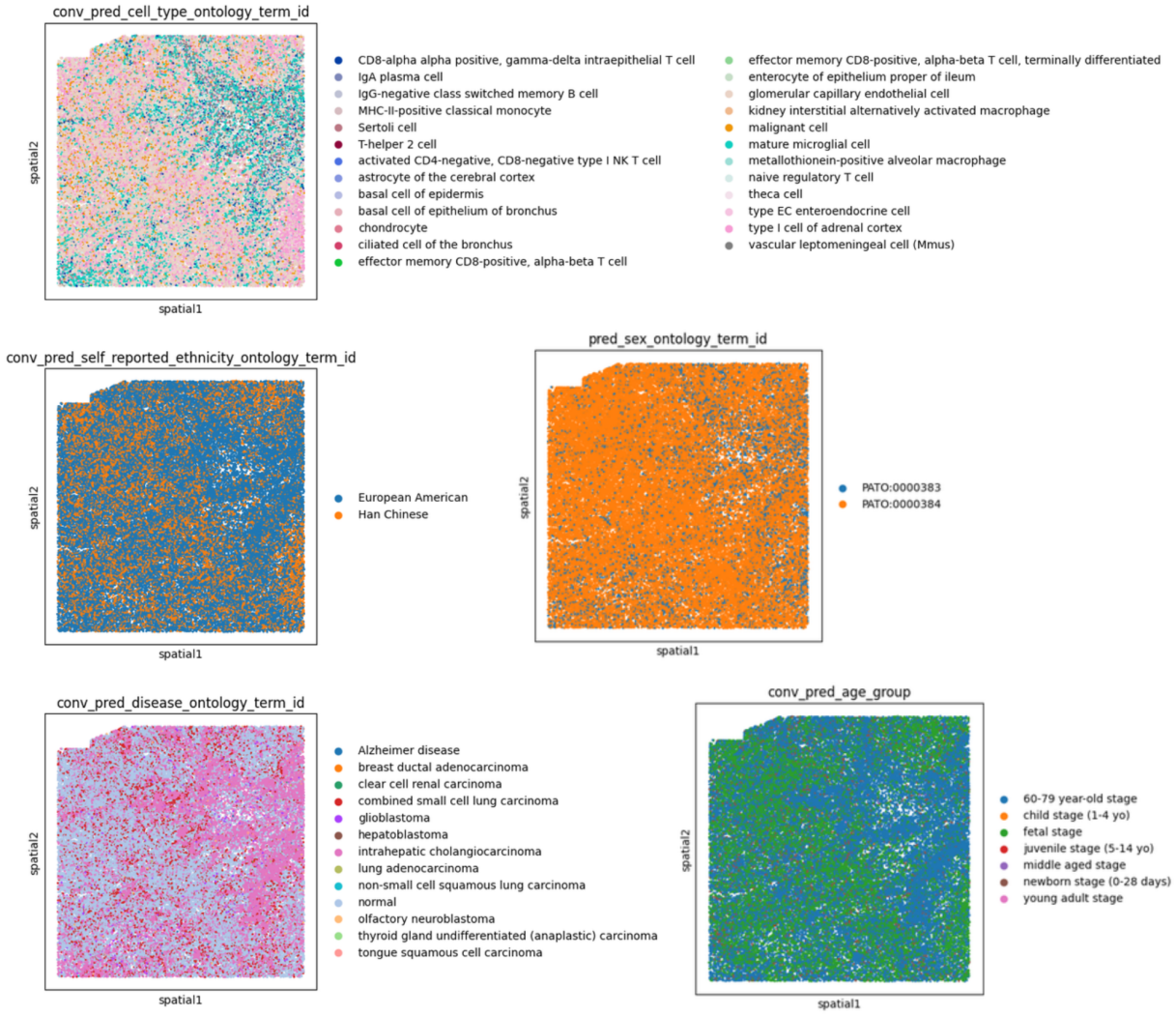


Figure 5.41: Spatial plot of the Xenium melanoma dataset with scPRINT-2 predicted cell labels. The disease, age, ethnicity, sex, and cell type labels are shown on top of the selected Xenium skin melanoma patch.

5.4.15 Violin plot comparison of the gene's expression between predicted malignant vs the rest

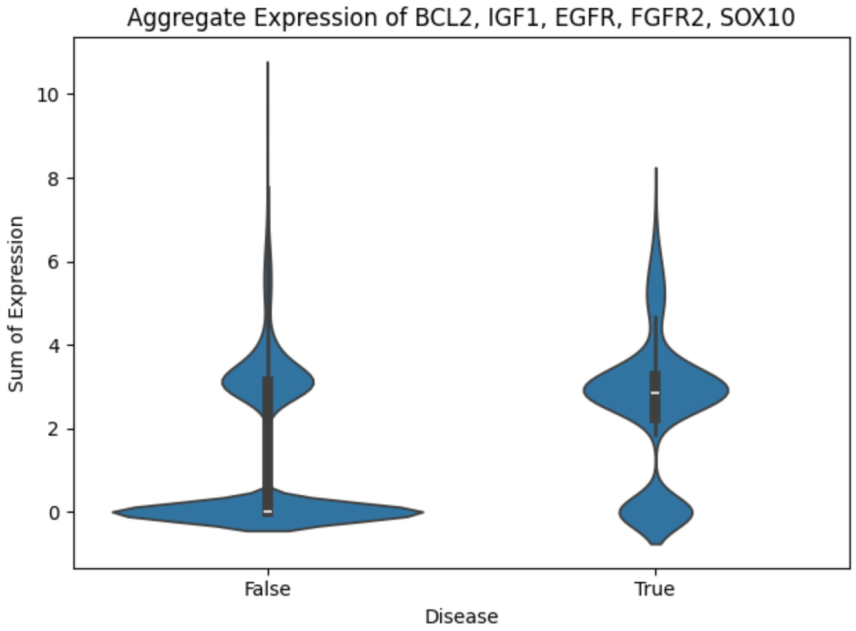


Figure 5.42: Violin plot comparison of the gene's expression between predicted malignant vs the rest. BCL2, IGF1, EGFR, FGFR2, SOX10, key melanoma markers are highly expressed in the malignant cell type label group vs the rest, with a p-value of 10^{-234}

5.4.16 Differential expression plot of “cancer” disease labelled vs rest in the xenium dataset

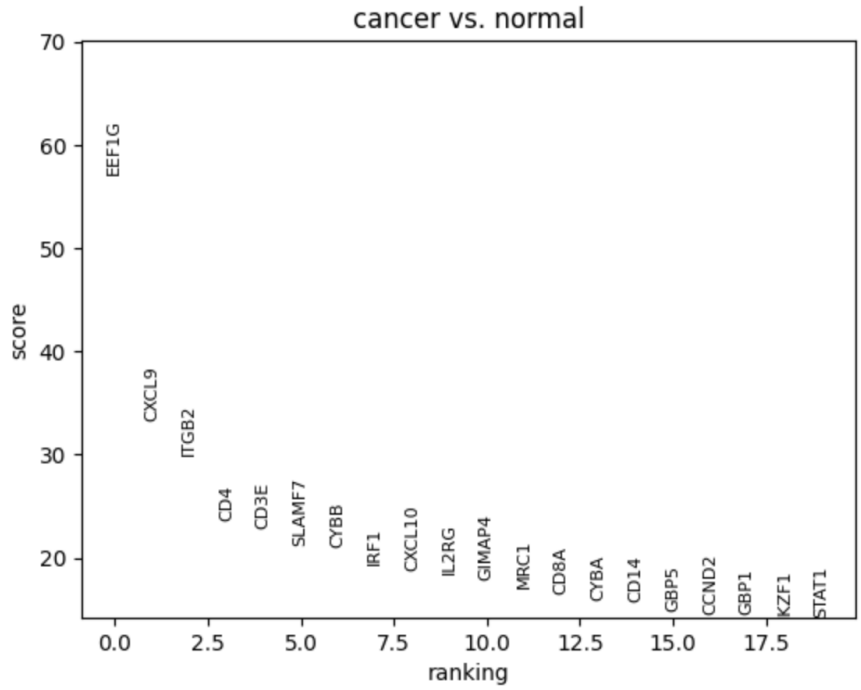


Figure 5.43: Differential expression plot of “cancer” disease labelled vs rest in the xenium dataset. Cells whose disease label is “cancer” vs the rest in the Xenium skin melanoma dataset

5.4.17 Illustration of criss-cross attention

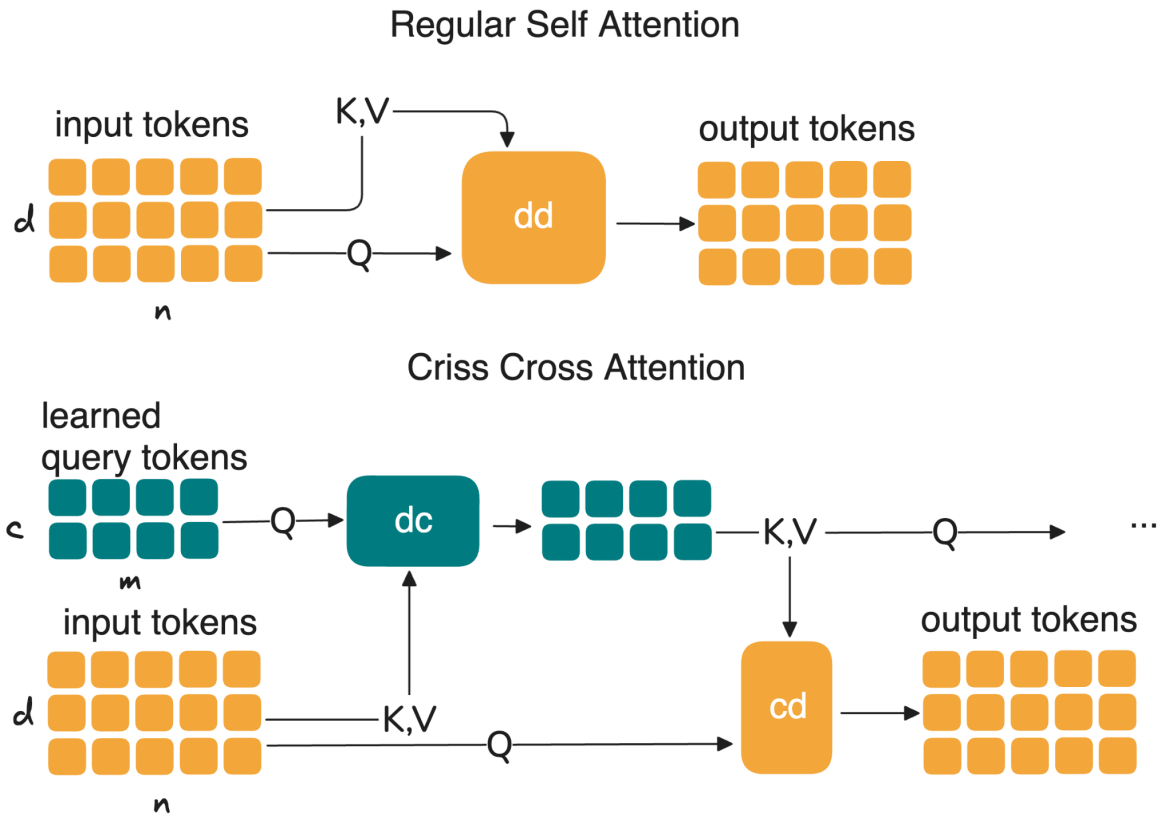


Figure 5.44: Illustration of criss-cross attention mechanism with sub-quadratic complexity

5.4.18 Illustration of the similarity and dissimilarity-based contrastive losses used in scPRINT-2

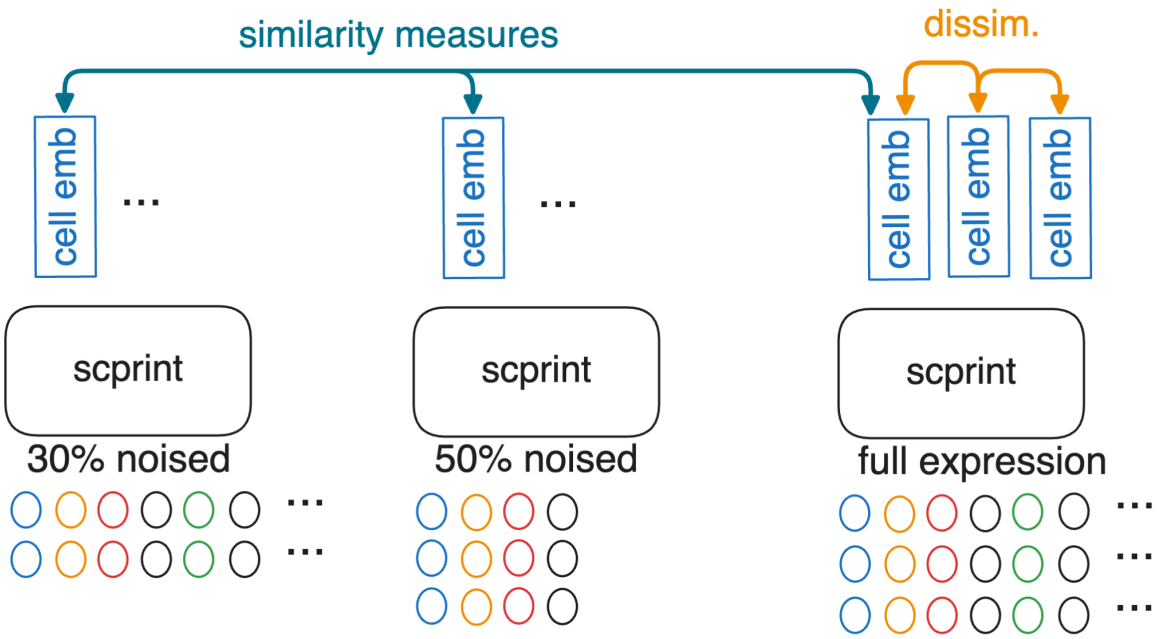


Figure 5.45: Illustration of the similarity and dissimilarity-based contrastive losses used in scPRINT-2. The contrastive losses push embeddings from the same cell at different noise levels to be as similar as possible.

5.4.19 Whisker plot of Open Problems' batch-integration with batch-correction-only scores

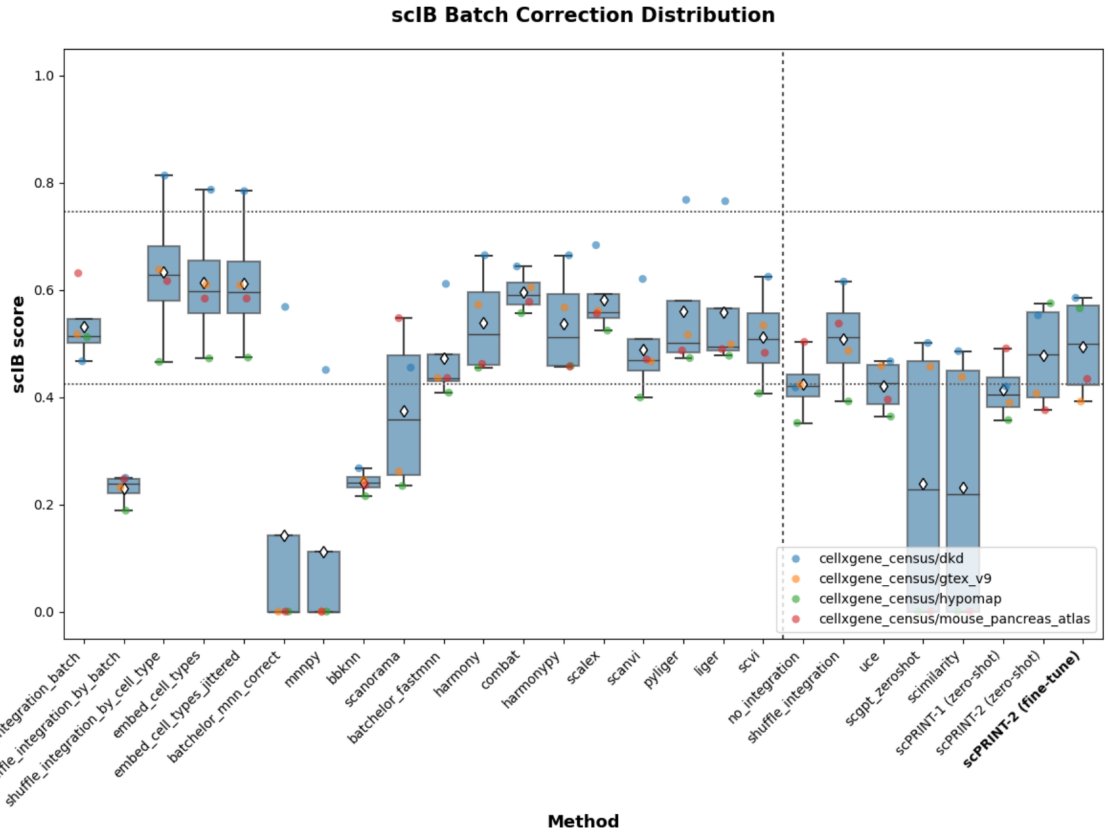


Figure 5.46: Whisker plot of Open Problems' batch-integration with batch-correction-only scores for scPRINT-1 and scPRINT-2 zero-shot, and finetuned, and all other models assessed in open problems.

5.4.20 Whisker plot Open Problems' batch-integration with Bio-conservation-only scores

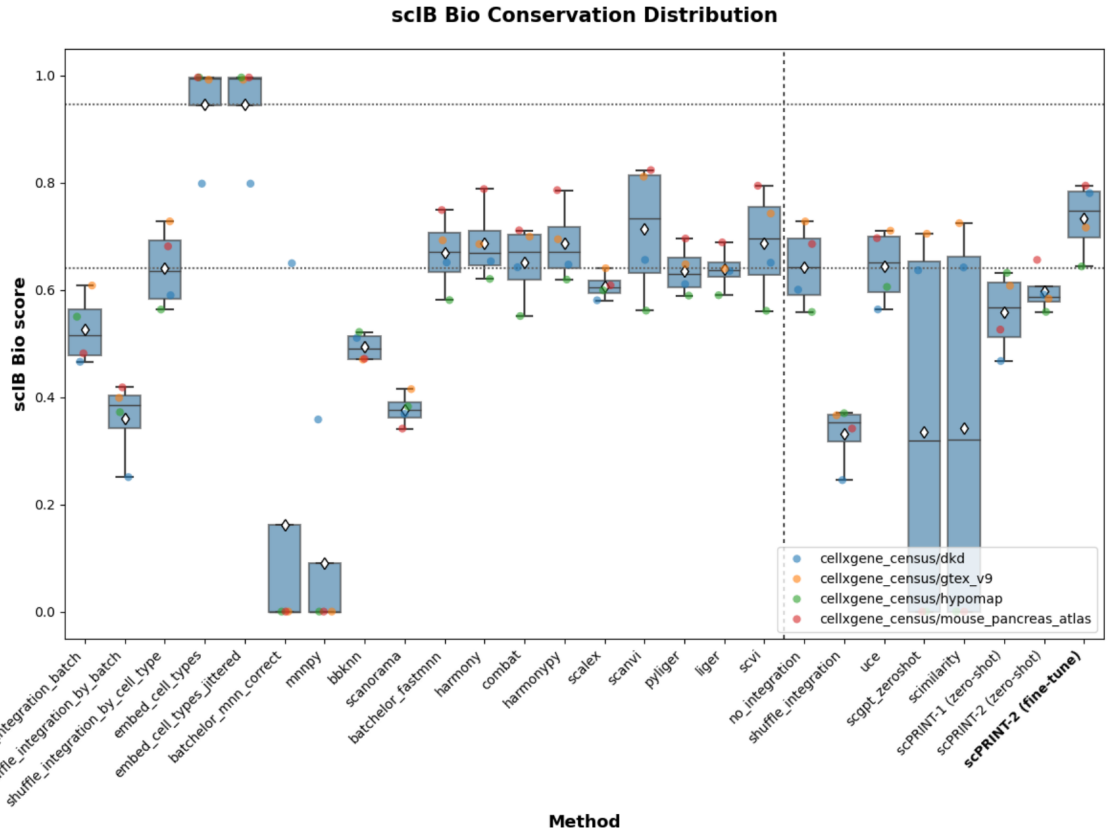


Figure 5.47: Whisker plot of Open Problems' batch-integration with Bio-conservation-only scores for scPRINT-1 and scPRINT-2 zero-shot, and finetuned, and all other models assessed in open problems.

5.4.21 Umap of scPRINT-2's zero-shot multi-species expression embedding using the full cell-embedding

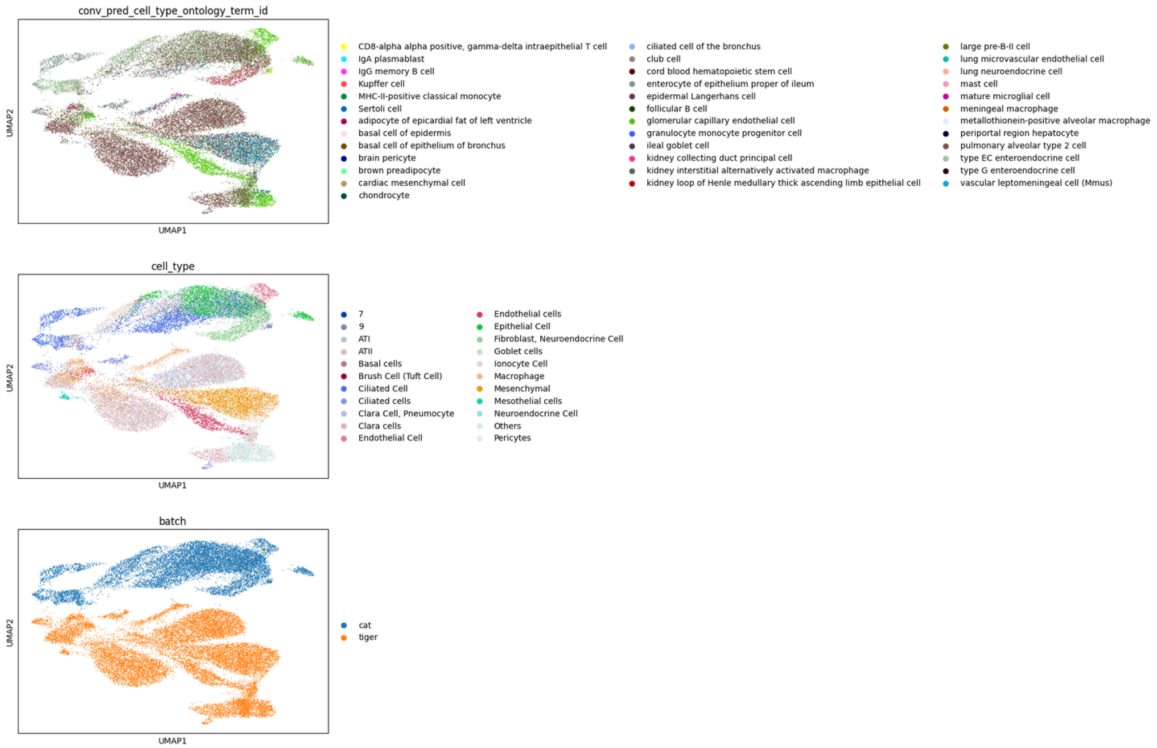


Figure 5.48: Umap of scPRINT-2's zero-shot multi-species expression embedding using the full cell-embedding. From top to bottom, scPRINT-2 predicted cell type labels, ground truth cell type labels, and ground truth organism labels.

5.4.22 Barplot of scIB score on scPRINT-2's multi-species integration

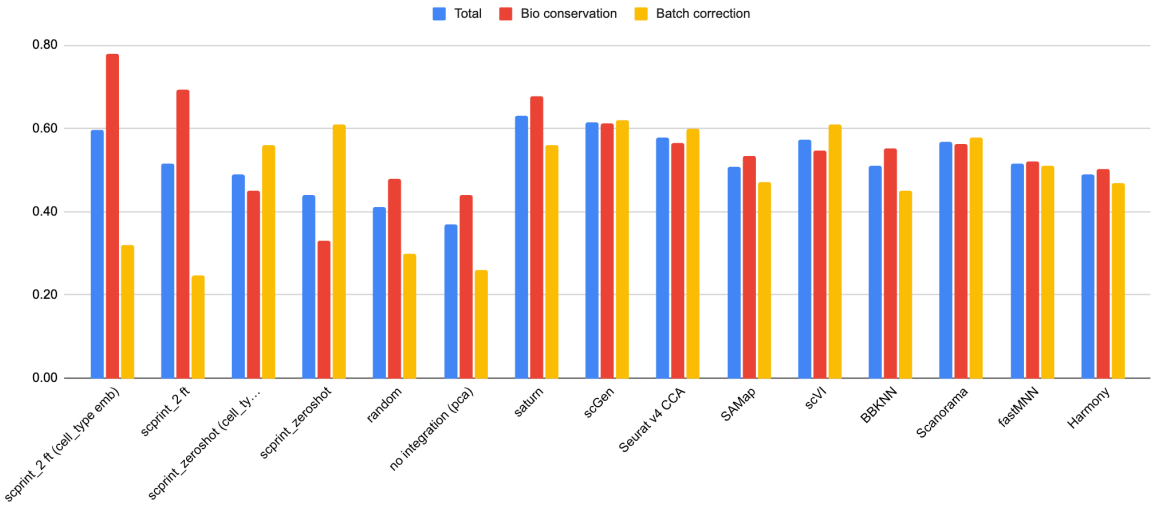


Figure 5.49: Barplot of scIB score on scPRINT-2's multi-species integration showing total, bio conservation, and batch integration across scPRINT-2 zero-shot and fine-tuned versions using both the full cell-embedding and cell-type-only cell-embedding.

5.4.23 Umap of scPRINT-2's zero-shot multi-species expression embedding using the cell-type cell-embedding

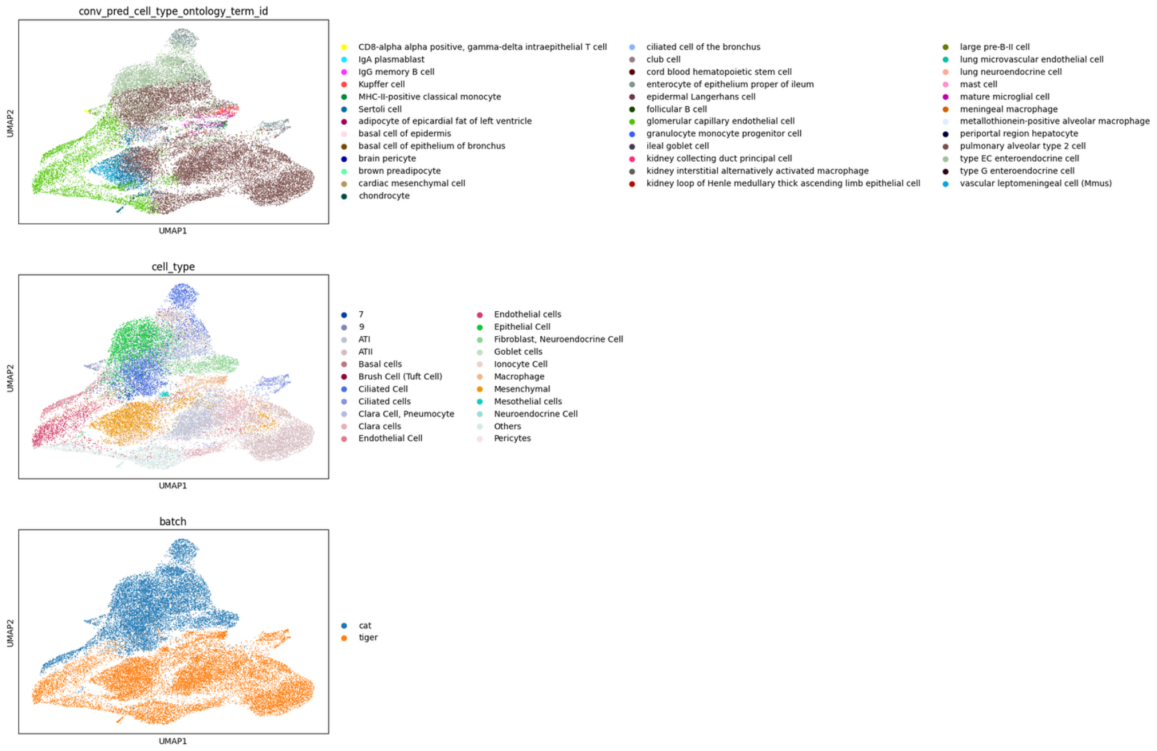


Figure 5.50: Umap of scPRINT-2's zero-shot multi-species expression embedding using the cell-type cell-embedding. From top to bottom, scPRINT-2 predicted cell type labels, ground truth cell type labels, and ground truth organism labels.

5.4.24 Umap of scPRINT-2's multi-species expression embedding post-finetuning using the full cell-embedding

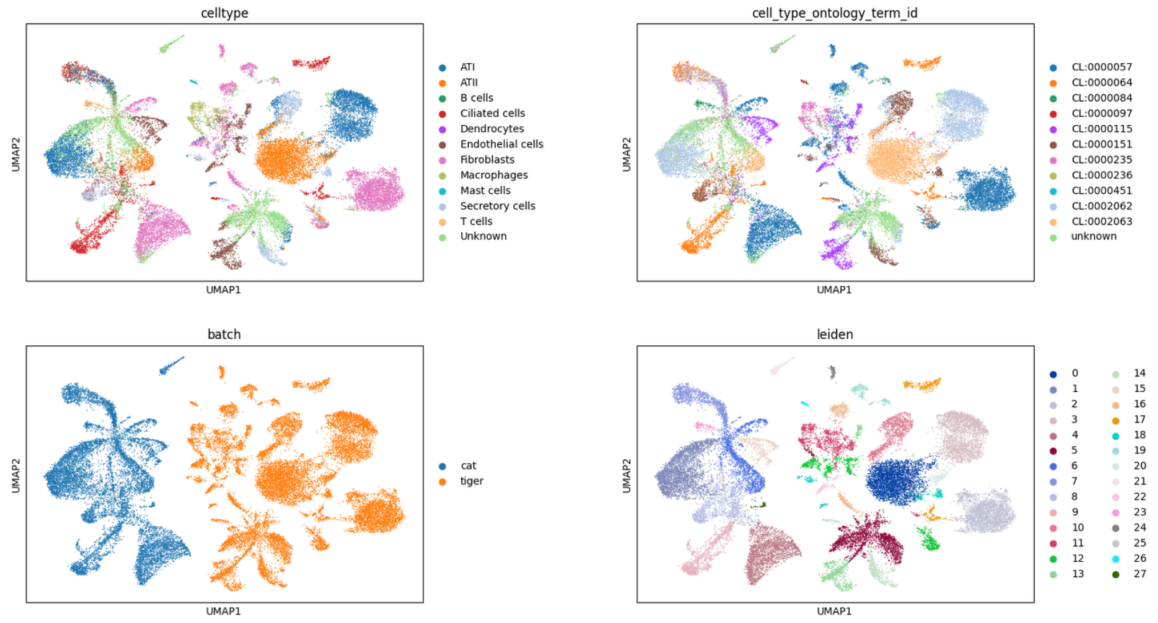


Figure 5.51: Umap of scPRINT-2's multi-species expression embedding post-finetuning using the full cell-embedding. From left to right and top to bottom, ground truth cell type, scPRINT-2 predicted cell type labels, ground truth organism labels, and Leiden clusters.

5.4.25 Differential expression plot of the human vs mouse dataset from section 4

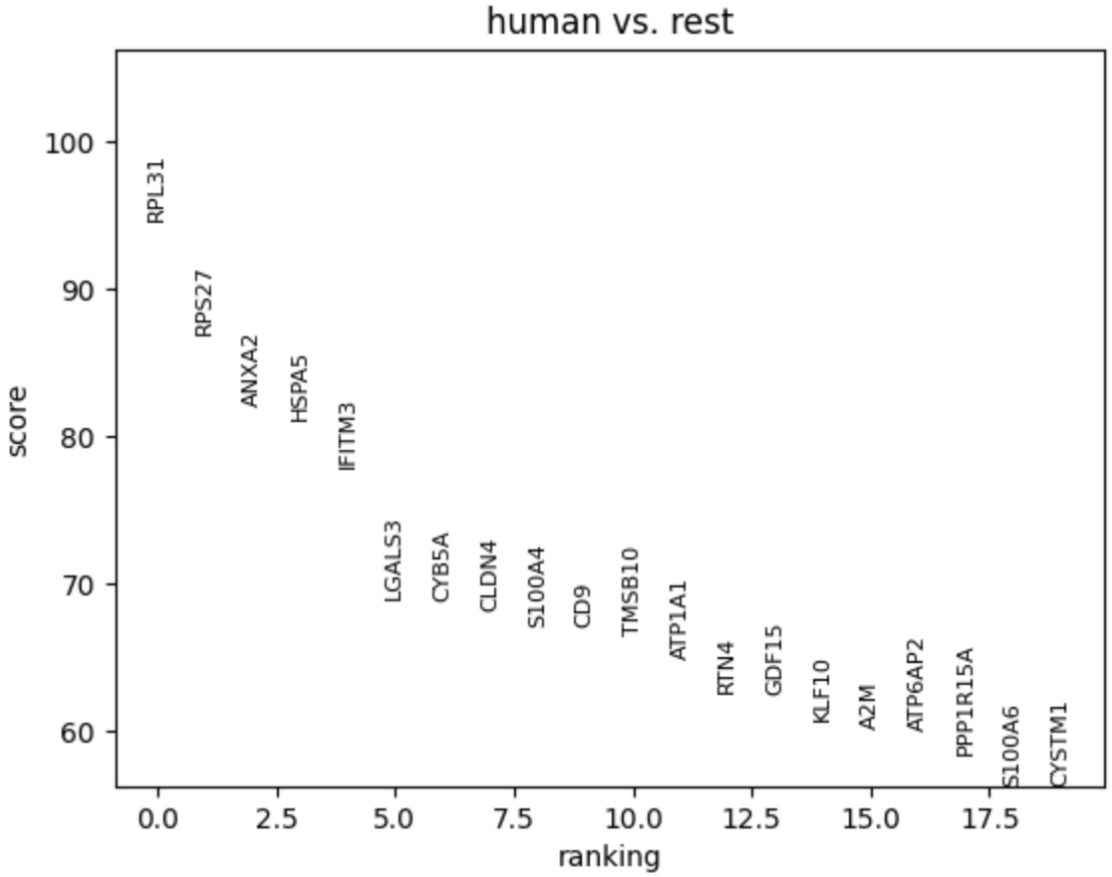


Figure 5.52: Differential expression plot of the human vs mouse dataset from section 4. Rest is mouse here.

5.4.26 Over-representation plot of humanized mouse data vs real mouse data compared to human

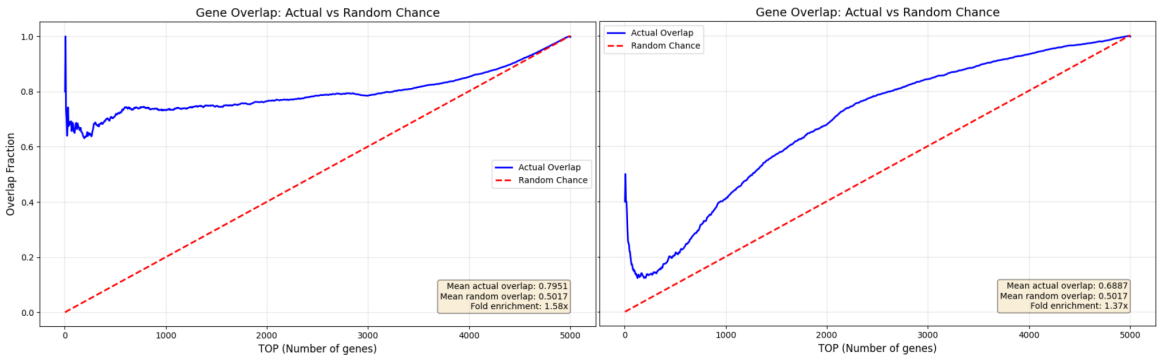


Figure 5.53: Over-representation plot of humanized mouse data vs real mouse data compared to human. Differentially expressed genes in scPRINT-2's humanized mouse data vs real mouse data compared to human.

5.4.27 Over-representation plot of female-like male data vs real female data compared to male

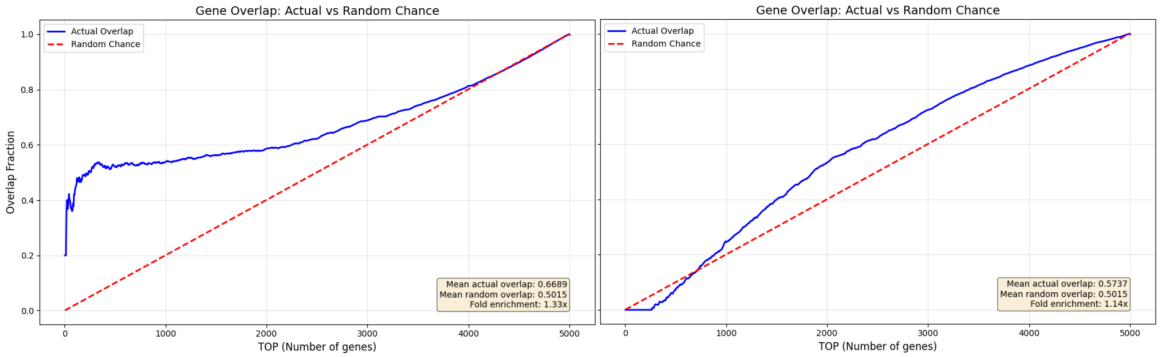


Figure 5.54: Over-representation plot of female-like male data vs real female data compared to male. Top differentially expressed genes in scPRINT-2's female-like male data vs real female data compared to male.

5.4.28 Dot Plot of Gene-set enrichment analysis over the differential expression analysis of section 4

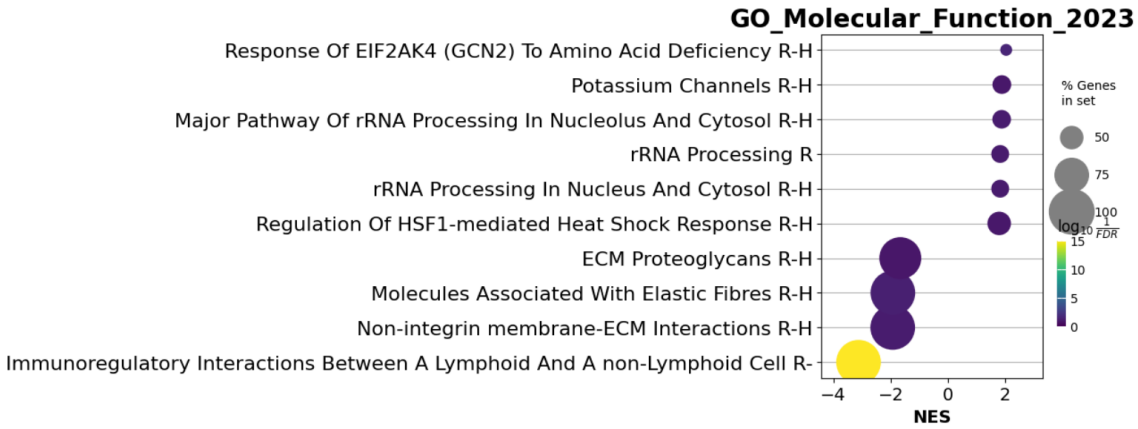


Figure 5.55: Dot Plot of Gene-set enrichment analysis over the differential expression analysis of section 4. Showing the top 10 most enriched gene sets from the GO molecular function 2023 database.

5.4.29 Output gene embedding for a non-fully trained model without XPressor architecture

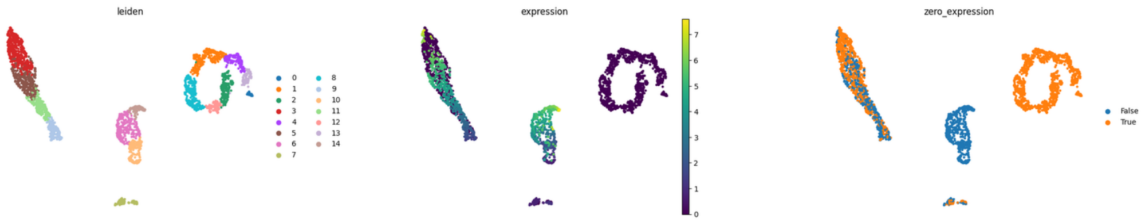


Figure 5.56: Output gene embedding for a non-fully trained model without XPressor architecture. Overlaying in color, from left to right, the Leiden clusters, the expression values, and the zero vs non-zero expression. Despite displaying multiple clusters, the number of enriched pathways in each is still smaller than for a model using XPressor (see Figure 5).

5.4.30 Venn diagram of the different ground truth gene networks

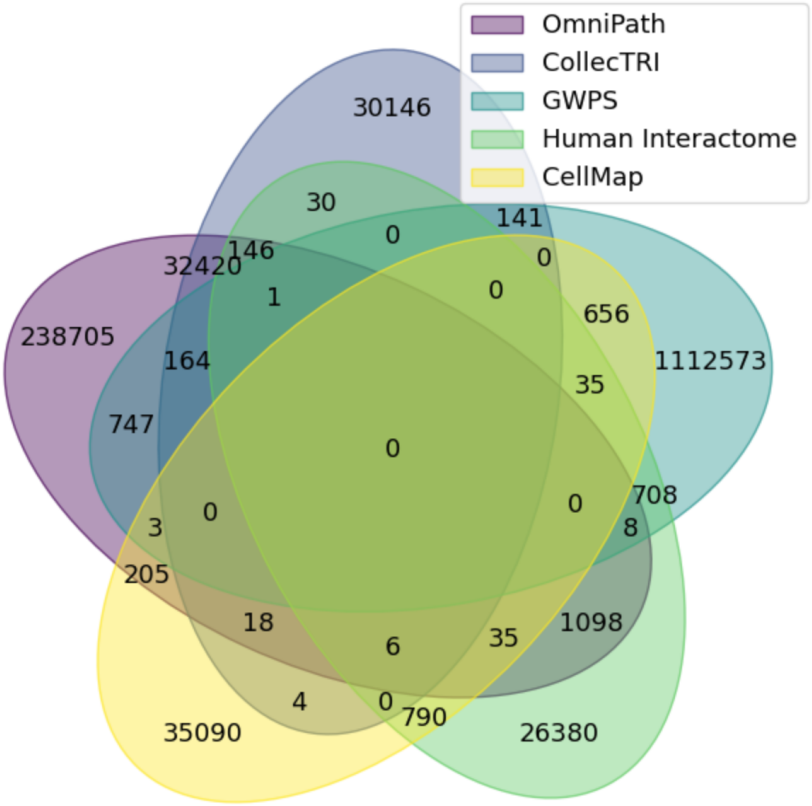


Figure 5.57: Venn diagram of the different ground truth gene networks showing overlap in the edges using gene symbols over the five ground truths used in our benchmark.

5.4.31 Whisker plot of AUPRC-ratio scores for scPRINT-1 and scPRINT-2

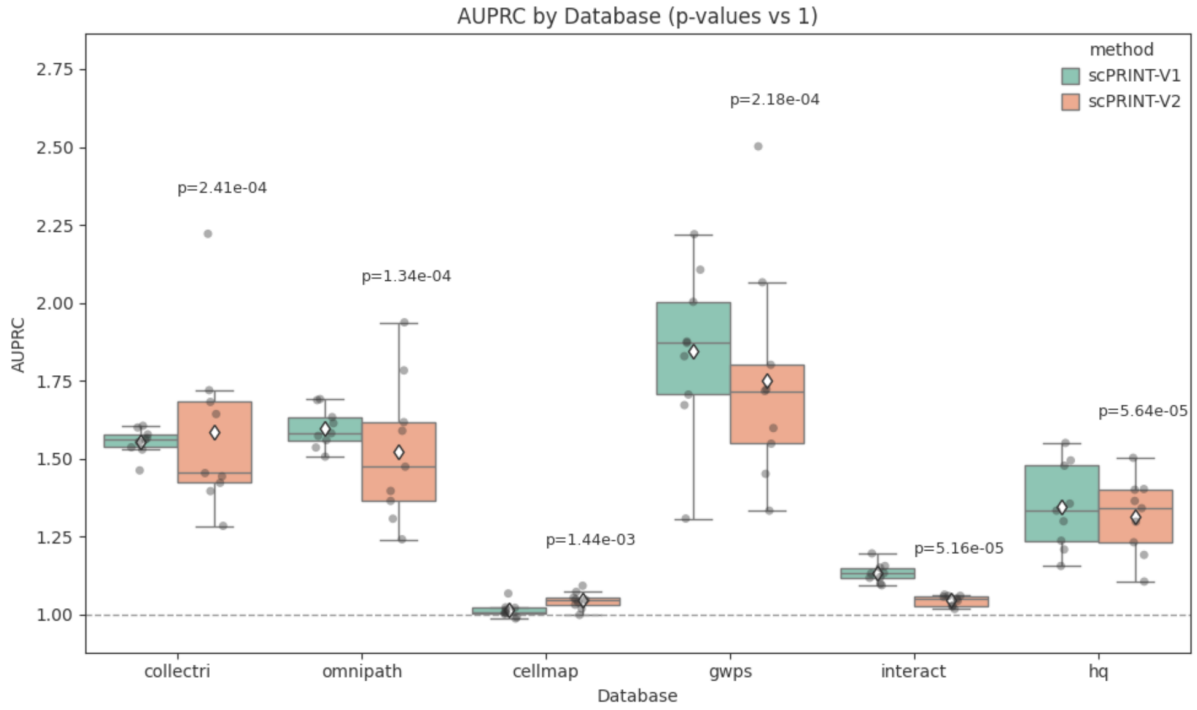


Figure 5.58: Whisker plot of AUPRC-ratio scores for scPRINT-1 and scPRINT-2 using their respective GRN-extraction methods, showing that scPRINT-2 extraction, while highlighting more relevant top connections, remains relatively similar to scPRINT-1 on AUPRC-ratio scores across each of the six ground truth networks.

5.4.32 Additional scPRINT-2 generated gene network computed from CDC45

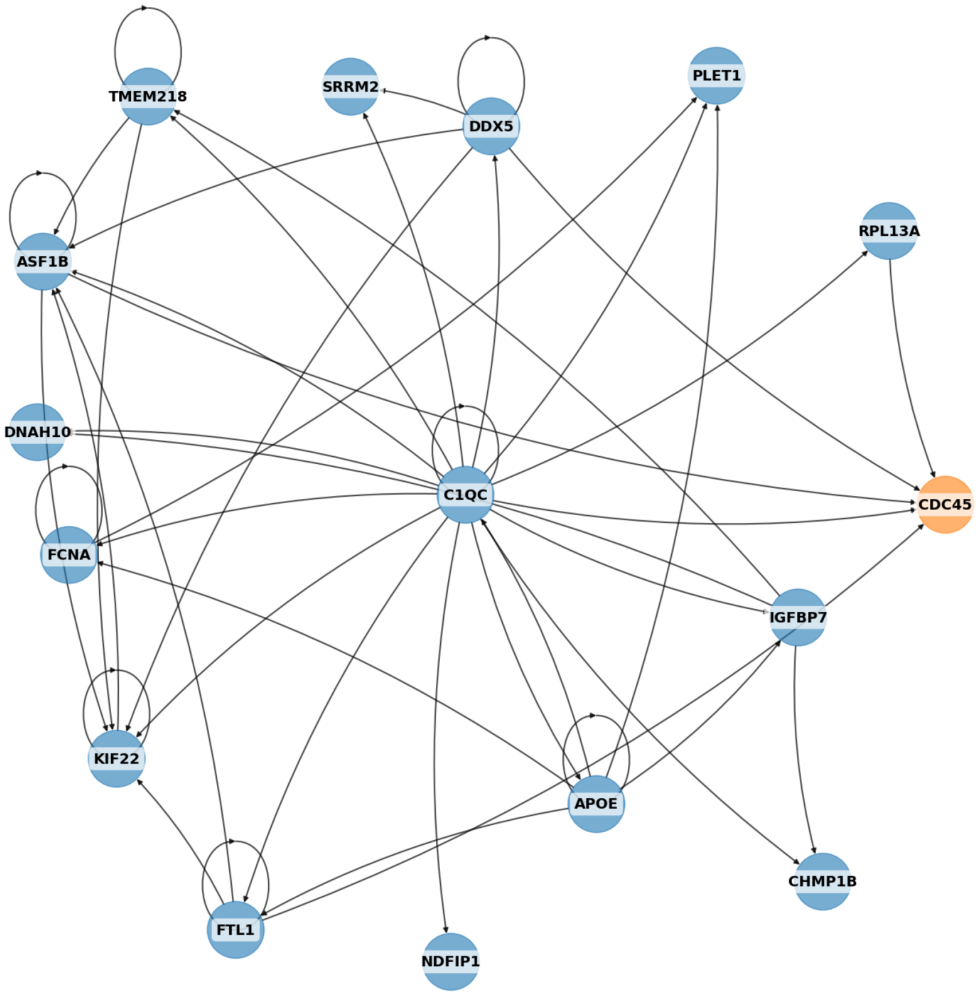


Figure 5.59: Additional scPRINT-2 generated gene network computed from CDC45. Subpart of the gene network using CDC45 as a seed gene and computed on 1024 mouse macrophages, showing how these networks can exhibit complex structures.