UNIVERSITY OF CALIFORNIA

SANTA CRUZ

INTEGRATING PRIOR BIOLOGICAL KNOWLEDGE AND MACHINE LEARNING FOR SINGLE-CELL TRANSCRIPTOMICS ANALYSIS

A dissertation submitted in partial satisfaction of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

BIOMOLECULAR ENGINEERING AND BIOINFORMATICS

by

Lucas Seninge

December 2022

The Dissertation of Lucas Seninge
is approved:

Professor David Haussler, Chair

Professor Joshua Stuart

Professor Benedict Paten

Professor Vanessa Jönsson

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

Abstract

INTEGRATING PRIOR BIOLOGICAL KNOWLEDGE AND MACHINE LEARNING FOR SINGLE-CELL TRANSCRIPTOMICS ANALYSIS

by

Lucas Seninge

Single-cell RNA sequencing (scRNA-Seq) has offered a unique window into studying cellular identity at unprecedented scale and resolution. However, the process of revealing this cellular identity remains challenging. For example, the annotation of each assayed cell with a cell type label indicating its functional identity still relies on manual examination, which is rate-limiting and poses reproducibility issues. Similarly, inferring the activity of gene regulatory pathways specifying cell state relies on methods designed for bulk RNA sequencing data and do not make use of the important amount of data generated by single-cell experiments. Here, I describe my work to combine prior biological knowledge about cellular entities contained in curated databases and machine learning to shed light on the cellular identity of single cells. Specifically, I developed statistical frameworks for the automated annotation of single-cell transcriptomes with cell type labels by integrating prior cell ontology information and cell type-specific marker gene sets. Then, I developed a method to infer pathway activity in single cells by using recent progress in the field of deep generative modeling as well as prior knowledge from gene annotation databases. I discuss potential future direction to design generative model architectures to approach the more ambitious task of modeling targeted perturbation of pathways or transcription

factors to perform *in-silico* experiments and alter cellular state at the single-cell level. Finally, I present collaborative work, notably on generalizing drug response prediction from bulk transcriptomic profiles of cell lines to cancer patients, integrating information about chemical structure in the predictive model. This body of work contributes to the growing literature of methods incorporating prior knowledge about biological systems into complex machine learning frameworks, as well as highlights the challenges met in such integration.

Acknowledgments

I would like to thank Pr. Joshua Stuart for his continuous support and guidance through my thesis. I'd like to believe that we built a relationship that goes beyond simply mentoring.

I would also like to thank Dr. Maximilian Haeussler, and Pr. David Haussler, without whom I wouldn't be in the PhD program. I am forever grateful for your trust and the opportunity you gave me.

I want to highlight the important contribution of Dr. Ioannis Anastopoulos, my great friend and colleague. You continue to inspire me professionally. I also want to thank Dr. Hongxu Ding, who was of tremendous help in my research. I would also like to thank every member of the Stuart Lab (past and present): Rosalyn, Bianca, Verena, David, Chris, Rojin, Alana, Nathan, ... thank you all for being such great colleagues !

I gratefully acknowledge the contribution of every collaborator I had during my thesis: Pr. Sofie Salama who taught me a lot about neurobiology, Gary Mantalas, Stephen Hwang, Harrison Wismer, Lauren Sanders and Allison Cheney. I also thank every member of my thesis committee.

Finally, I want to thank all my friends for their support, here in California and back in France.

This work is dedicated to my family.

Chapter I: Introduction

Cells are the structural and functional units of eukaryotic organisms. The adult human body is composed of roughly 30 trillion cells [Sender et al., 2016], involved in many roles from muscle contraction, nervous system message passing or immune response to an infection. The study of cells' functions and their role in both homeostasis and disease have been confined to molecular and cell biology assays for a long time, from morphological descriptions with microscopy to surface protein characterizations through analytical methods. Since the completion of the first Human Genome sequence in 2003 [International Human Genome Sequencing Consortium, 2004], the development of high-throughput sequencing (HTS) has offered a unique insight on the characterization of gene expression patterns of mixtures of cell populations at the tissue level [Lonsdale et al., 2013]. In more recent years, the advent of single-cell profiling methods have enabled researchers to look at cell populations at an unprecedented resolution. Particularly, the rapid development of single-cell RNA sequencing (scRNA-Seq) protocols have made it possible to explore the transcriptome of an increasing number of individual cells in various biological systems.

scRNA-Seq was introduced in 2009 [Tang et al., 2009], with a very limited throughput at the time, and has since gone through multiple technological improvements. Notably, the development of droplet-based methods and the Unique Molecular Identifier (UMI) technology [Macosko et al., 2015, Islam et al., 2014] have enabled researchers to charac-

terize the transcriptomes of thousands, and even millions of cells simultaneously. This new biological resolution has motivated studies to refine the understanding of species' distinctiveness at the single-cell level, and also to shed light on the role and definition of different cell populations in homeostasis and diseases [Kolodziejczyk et al., 2015].

The novel challenges which came with the advent of single-cell transcriptomics have encouraged the development of efficient computational tools to process and analyze the data, in order to gain insights into the biological system of interest. In fact, a few aspects of scRNA-Seq datasets have motivated the development of new tools and analysis methods. First, the large size of the datasets (from thousands to millions of cells, with 10,000-50,000 genes measured) encouraged the community to develop efficient and scalable tools for analyzing such datasets in a realistic time. Secondly, the noise level due to the low amount of RNA material in a single-cell, stochasticity of gene expression, and technical limitations of amplification techniques result in variation that is not always biologically meaningful. Notably, dropout effects plague singe-cell datasets: transcripts can be missed during the capture and amplification, resulting in a technical zero inflation of the cell-gene count matrix, which hinders the ability to analyze the data [Kharchenko et al., 2014]. Lastly, the differences between experimental plateforms and protocols can lead to huge batch effects that will obscure biological differences [Tung et al., 2017].

As a result of these technical challenges, analysis methods have been developed and adapted to the specific problems of single-cell transcriptomics. A simple analysis pipeline for single-cell transcriptomics can be described as followed: (1) quality control and normalization, which aims at removing genes/cells of poor quality and ensure that

differences in sequencing depth and technical batches are accounted for, (2) dimensionality reduction for summarizing main characteristics about the dataset and visualization, and (3) clustering to group similar cells together. Then, many downstream applications such as differential gene expression [Love et al., 2014, Wang et al., 2019], trajectory inference [Saelens et al., 2019] and functional annotations are possible to further investigate the transcriptome of single-cells.

Despite the maturity of scRNA-Seq analysis procedures, there are a lot of remaining challenges in the field. The first one is the problem of annotating cells with biological entity labels such as cell types. In order to draw biological conclusions on an experiment, it is crucial to know what cell populations are present in a dataset. This task is typically done manually, which is time consuming and pose reproducibility issues. I propose to leverage prior biological knowledge about characteristic marker genes to automate cell type annotation, introducing a novel scoring algorithm called scoreCT (Chapter II). I compare the method to a state-of-the-art enrichment method and show that it is competitive. I further propose to integrate structured information about cell type relationships such as cell ontology to perform annotation and reduce uncertainty of labelling (Chapter I). I show that it is possible to combine these two types of information into a unified framework and highlight challenges when using prior biological knowledge in the annotation task.

The second problem I study in my thesis is the inference of gene regulatory modules activities at the single-cell level by representing them as latent variables (Chapter III). Genes act as coordinated units within programs that define the state of a cell.

These modules can be metabolic modules, specific response to a stimulus, or even gene regulatory networks. As these modules represent functional abstractions and lack direct experimental evidence, their inference through statistical models that infer hidden/latent variables is highly needed. I propose to leverage the recent advances in deep generative modelling to construct an interpretable non-linear encoder-decoder architecture called VEGA (VAE Enhanced by Gene Annotations). I also introduce a Bayesian differential testing procedure to quantify differences in gene module activities across cell populations that is competitive with standard enrichment methods. I discuss how to extend this procedure to control for effect size. Finally, as prior knowledge contained in biological databases can be erroneous and not context-specific enough, I discuss extensions to VEGA to soften assumptions about prior biological knowledge through various regularization strategies, as well as very recent advances in interpretable deep generative models. In the last part of this chapter, I discuss potential strategies to create interpretable models where latent variables can be manipulated to design *in-silico* experiments to study the effect of targeted regulator or gene module perturbation on individual cell populations.

Finally, in the last chapter I introduce collaborative work I performed during my thesis. This notably includes work on a deep learning model for drug response prediction in patients incorporating drug structure information as well as data collected from cell lines. I also contributed to create a novel framework to manipulate large genetics databases such as UKBB for machine learning tasks in an efficient and scalable way.

Together, these projects highlight different strategies and challenges for incorporating prior knowledge into machine learning frameworks.

Chapter II: Using curated and structured prior knowledge to enable fast and interpretable annotation of cell ensembles

2.1 Retrieving cell type identity of cluster fingerprints in single-cell transcriptomics datasets

Annotation of single-cell transcriptomes with biological entity labels is crucial to draw meaningful biological conclusions from a scRNA-Seq dataset. However, it remains a challenging task to automate. In this section, I propose a novel scoring algorithm to label single-cell data using marker gene knowledge. I show that this novel approach is both fast and competitive with state-of-the-art enrichment methods. I provide an efficient Python implementation working within the popular Scanpy environment [Wolf et al., 2018] at `https://github.com/LucasESBS/scoreCT`.

2.1.1 Cell type annotation of scRNA-Seq data

A critical application of computational tools to scRNA-seq data is the annotation of each transcriptome with cell types to summarize the identity and the role of the cells in the studied system, which is crucial to study cell heterogeneity. This step can be challenging as it is often performed manually, which can be time-consuming. Another issue is that

this annotation often relies on biologist knowledge of few relevant features, called "marker genes," which can make the annotation inconsistent and rarely reproducible between labs. Considering these challenges, several attempts have been made to automate cell type annotation in scRNA-seq datasets. A typical approach is to make use of the existing collection of curated scRNA-seq datasets, gathered into an atlas, and try to map the new dataset to annotated partitions of the atlas called clusters [Kiselev et al., 2018]. The methods using this approach often rely on similarity measures on a reduced number of features, but can be intractable when the size of the atlas grows. Also, atlases can introduce reference bias, and it is not clear at the time which atlas should be used for such applications. Other approaches rely on the use of Artificial Neural Network (ANN) to make the annotation a classification problem, where the model is trained on curated datasets and applied to the new dataset to annotate each transcriptome. While providing good results, methods similar to [Ma and Pellegrini, 2020] lack interpretability as to which features motivated the assignment of each individual transcriptome to a particular cell type. Finally, approaches using prior marker genes knowledge have been developed. These models don't introduce a reference dataset bias, and only rely on the prior knowledge of a few marker genes per cell type. Classic enrichment methods such as GSEA [Subramanian et al., 2005] can be repurposed for annotation in such fashion, and more refined probabilistic generative models can also be used [Zhang et al., 2019a]. These methods are closer to the process of manual annotation by biologists, and present the advantage of providing assignments that are directly interpretable in terms of known biology of the system of interest.

2.1.2 Retrieving cell type identity of cluster fingerprints in single-cell transcriptomics datasets

It can be of major interest to propose an annotation framework using a reduced amount of information about the cell gene expression profiles. One way to compress a single-cell dataset along the sample axis is to group similar cells together using clustering algorithms. Most notably, community detection algorithms such as the Louvain method [Blondel et al., 2008] are popular to cluster single-cell dataset and identify sub-populations representing cell types. Another possibility to further compress a single-cell dataset is to only keep the top K differentially expressed genes (DEGs). Those genes are usually picked by comparing groups of cells in a *"one-versus-rest"* type of statistical testing procedure. For example, a Wilcoxon Rank-Sum test [Mann and Whitney, 1947] can be performed to compare the mean gene expression of a query group (cell cluster of interest) to a reference group (the rest of the dataset). This procedure helps to identify biology that is unique to the query group, and hence the top K DGEs (sorted by test statistics or corrected p-values for example) are good candidates to build a cluster "fingerprint" representing the unique identity of the cell cluster and to further compress a single-cell dataset along the feature axis.

In this section, I propose a method named **scoreCT**, that can take as an input these cluster fingerprints and use prior knowledge about marker genes to annotate these compressed datasets with biological labels. The framework aims at formalizing the qualitative procedure that is applied by biologists in a way that is both accurate and does not require to load the full dataset in memory. The code is available at

`https://github.com/LucasESBS/scoreCT`.

2.1.3 The scoreCT procedure

2.1.3.1 Scoring function

The scoreCT algorithm works by attributing a score to each cell type in the reference, for each cluster fingerprint. Based on the DGE ranking of each cluster, scoreCT splits the top K cluster DGEs into m bins, and computes a score based on the weighted intersection with the reference. Formally, the score $S_{i,j}$ for cluster i and cell type j is computed as:

$$S_{i,j} = \sum_{l=1}^{m} w_l \times s_l$$

where w_l is the weight associated with the l-th bin and $s_l = |k_{i,l} \bigcap \mu_j|$, with $k_{i,l}$ being the subset of the top K DGEs for cluster i present in the l-th bin, and μ_j the set of marker genes for cell type j. We note that in the following we focus on the case where the top K DGEs are evenly divided into m bins and $w_l = m - (l - 1)$ (uniformally distributed integer weights), but the scoring function can be extended to other cases.

2.1.3.2 Score significance

Let $M = \{g_1, g_2, ..., g_N\}$ be a set of N background genes, representing the transcriptome of the cells. Let $\mu_j \subset M$ be a set of n genes representing the marker genes for cell type j. To assess if the score $S_{i,j}$ for a given cluster/cell type pair (i, j) is meaningful, I propose to use a permutation test akin to those performed by enrichment methods

[Subramanian et al., 2005]. Formally, P permutations of the full ordered DGE ranking are performed, and the scoreCT score s is recomputed each time on the new randomized gene ranking. If P is large enough, this allows to approximate the null distribution of scores, which can be compared to the original score $S_{i,j}$ (Fig.2.1A). If X is the random variable representing the scores obtained by this method, a p-value is approximated as the probability to observe more extreme values according to the null distribution, as:

$$P(X \geq S_{i,j}) \simeq \frac{\#(s \geq S_{i,j})}{P}$$

We use the p-values derived from our model to assign cell types to cluster fingerprints: we assign the cell type with the smallest p-value to each cluster fingerprint. A threshold p is applied such that any cluster whose best p-value is greater than p is labelled as *'NA'*, meaning that it can't be identified as belonging to any cell type in the reference by the method. p is usually set to 0.1 in our analysis.

This non-parametric approach presents the advantage to be applicable for any variation of the scoreCT scoring function (*eg.* non-uniformally distributed bin weights, non-even binning of top K genes...), but can be relatively slow if P , the number of fingerprints to score and the number of cell types in the reference are very large. As a fast alternative to the permutation null model, I propose an approximation using the sum of multinomial trials.

2.1.3.3 Multinomial sum approximation

The scoring function of scoreCT can be approximated as the sum of the outcome of multinomial trials. This approximation holds relatively well in most cases, discussed in greater length in the next section.

Formally, we formulate the following null hypothesis:

$$H_0 : Y \sim Mult(K, \pi), \quad S_{i,j} = \sum_{k=1}^{K} y_k$$

where K is the number of top DGEs, and π is a probability vector describing the probability of each outcome when selecting a random gene (an outcome being the contribution of the gene to the total score). In the case where the top K are evenly distributed into m bins, we have $\pi = (\frac{N-n}{N}, \overbrace{\frac{1-\frac{N-n}{N}}{m}, \cdots, \frac{1-\frac{N-n}{N}}{m}}^{m})$.

Let X be the random variable representing the scores obtained by the method. To compute the probability $P(X = S_{i,j})$, we can use generating functions [Doubilet et al., 1972] to solve the combinatorial problem. Let us consider the polynomial $f(x)$ such as:

$$f(x) = C_0 + C_1 x^{w_1} + C_2 x^{w_2} + C_3 x^{w_3} + \cdots + C_m x^{w_m}$$

with $(C_0, C_1, \cdots, C_m) = \pi$. The probability of obtaining a certain score $S_{i,j}$ after selecting K random genes is given by the coefficient multiplying the term in x whose exponent is $S_{i,j}$:

$$P(X = S_{i,j}) = [x^{S_{i,j}}]f(x)^K$$

In order to derive a p-value for our score, we can compute the probability that a greater score would be observed under the assumption that the null hypothesis is true (Fig.2.1B).

$$P(X \geq S_{i,j}) = \sum_{s=S_{i,j}}^{S_{max}} [x^s]f(x)^K$$

These p-values are used to assign cell types to cluster fingerprints as previously described. Finally, I demonstrate that for a variety of (m, K) parameters, the multinomial sum approximation is several orders of magnitude faster than the permutation test on simulated data (Fig.2.1C).

I note that this approach can be easily generalized to the case where the top K genes are not evenly distributed into m bins, which only affects π and thus the coefficients of the generating function. More worthy of note is the generalization to bin weighting beyond the case $w_l = m - (l - 1)$, which can lead to polynomial expansions with non-integer exponents to derive p-values. We leave this case for further research, as this is a non-trivial issue.

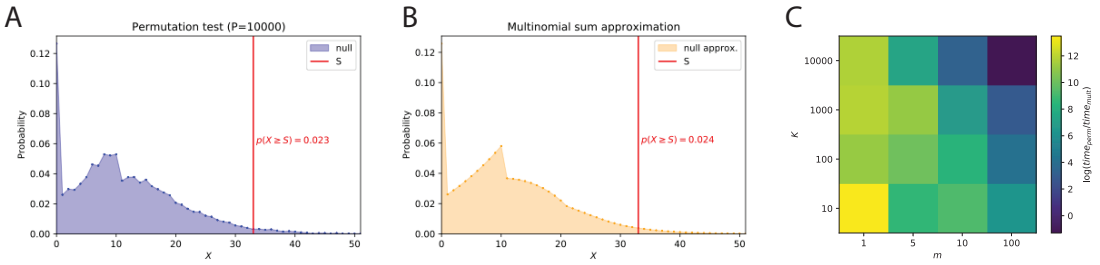

Figure 2.1: **Proposed null models to compute scoreCT's score significance.** Null model derived from synthetic data with $n = 100$ marker genes and $N = 48284$ total genes with (A) a permutation test and (B) the multinomial sum approximation. (C) Time improvement of the multinomial sum approximation compared to the permutation null model for a grid of (m, K) parameters. Permutation number is kept constant at $P = 1000$. Time ratio is shown on a log scale.

2.1.3.4 Relation to other tests

This method is strongly related to enrichment tests such as GSEA [Subramanian et al., 2005] or Fischer's exact test. We note that although enrichment tests have been studied extensively and perform well in general, they do not allow to store compressed fingerprints as they require to be run on the full transcriptome to be meaningful. On the other hand, Fischer's exact test can be used on a reduced amount of features similarly to scoreCT, but does not make use of the ranking information of the top DGEs, therefore not differentiating between edge cases where the same amount of marker genes are placed at the top or bottom of the top K DGEs.

2.1.4 Validity of the multinomial sum approximation

In this section I discuss the validity of the multinomial sum approximation. To this end, I used synthetic data generated as followed. The genes of the human transcriptome are randomly ranked and K top genes are used as the synthetic cluster fingerprint.

For n marker genes, we select $k = \lfloor n * p \rfloor$ genes from the K top as being part of the marker gene list, where $p \sim \mathcal{U}(0.33, 0.95)$ represents the proportion of total marker genes present in the fingerprint. The $n - k$ rest of marker genes are selected at random in the non-fingerprint genes. This ensures that in practice, there are always a proportion p of marker genes in the top K selected genes of the fingerprint.

From the formulation of the multinomial sum approximation, we notice intuitively that since the process for drawing marker genes is "with replacement", when the number of marker genes n becomes large we are likely to inflate the null model for larger scores. We first validated this intuition qualitatively comparing the 2 null models. We see that for a total number of genes $N = 48284$ and $n \in \{10, 100\}$ marker genes, the 2 null distribution overlap really well (Fig.2.2A,B). However when the number of marker genes becomes much larger ($n = 1000$), the distribution obtained from the multinomial sum approximation is shifted towards larger scores (Fig.2.2C). This null model is therefore likely to underappreciate scores obtained from scoreCT.

To validate this trend quantitatively and derive a rule of thumb for using the multinomial sum approximation, I studied the evolution of the (log) Kullblack-Leibler divergence (KLD) between the permutation distribution and the multinomial sum approximation distribution as a function of the ratio of marker genes to total genes, $\frac{n}{N}$ (Fig.2.2D). As expected, for small ratio $\frac{n}{N}$, the two distribution are very close. However, the log-KLD increases rapidly when $\frac{n}{N} > 5\text{e}{-3}$. Using the elbow of this plot, we decide on a rule of thumb to use the multinomial sum approximation model when $\frac{n}{N} < 3\text{e}{-3}$. This encompasses most of use cases since marker genes are usually in the range of a few to a

few dozen, while the transcriptome is usually in the order of a few tens of thousands genes.

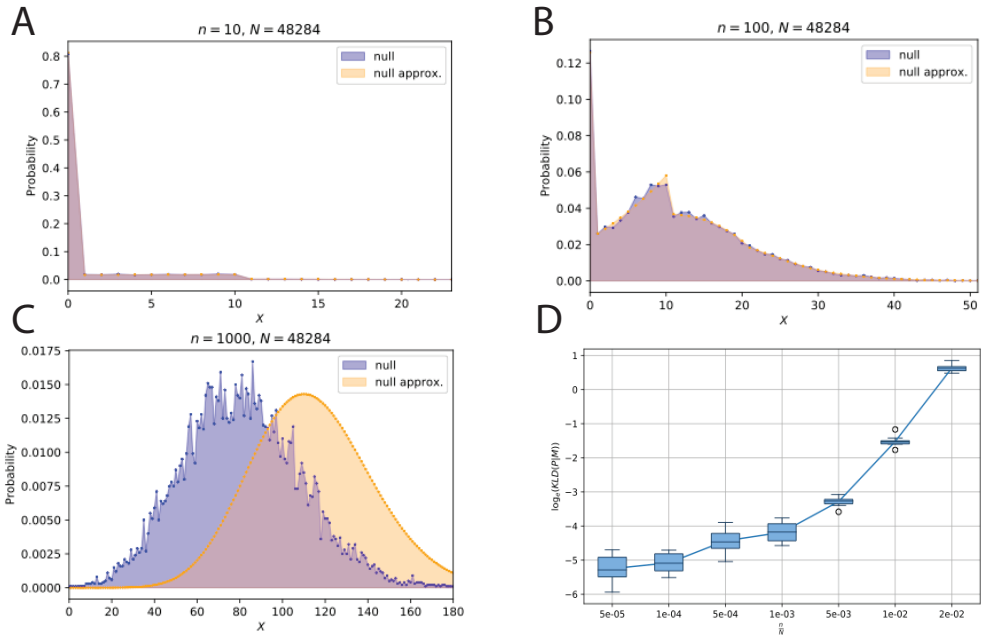

Figure 2.2: **Condition of validity for the multinomial sum approximation null model.** Comparison of the 2 null models for different number of marker genes n: (A) $n = 10$, (B) $n = 100$, (C) $n = 1000$. (D) Evolution of the log-Kullback-Leibler divergence between the 2 null distributions as a function of the fraction of marker genes in the reference (10 random datasets per fraction). For this figure, parameters K and m were kept constant respectively at 1000 and 10.

2.1.5 Results on a cortical organoid dataset

As a proof of concept of scoreCT ability to annotate clusters with cell types, I performed a re-analysis of the week 2 cortical organoid dataset from [Field et al., 2019]. An independent clustering of the cells was performed using the louvain algorithm [Blondel et al., 2008]. I used a Wilcoxon rank-sum test to rank DGEs for each cluster in a 'one-vs-rest' comparison. ScoreCT is able to correctly assign cell types to most of the clusters

identified independently of the clustering solution provided by the original authors. ScoreCT was able to correctly identified the clusters belonging to the three major cell types present in the dataset (Neuroepithelium, Radial Glia and Cajal-Retzius neurons). ScoreCT also rejected a population of cells labelled as doublets by the author, assigning to $'NA'$ as none of the cell type present in the reference passed the p-value threshold of 0.1 for the combination of parameter ($m = 5, K = 300$) used for the annotation for this cluster.

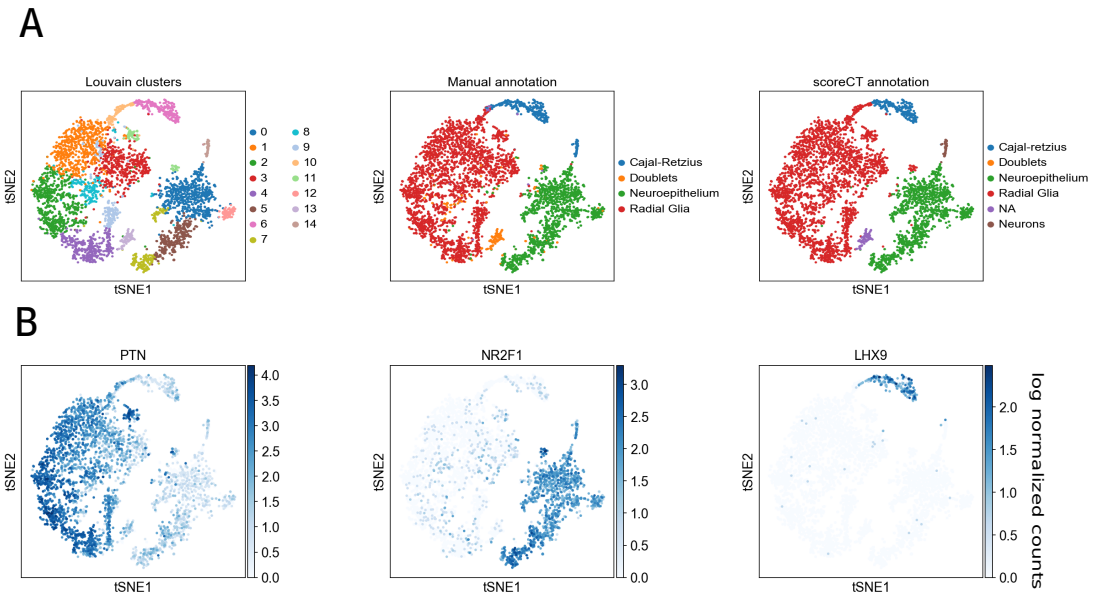

Figure 2.3: **scoreCT accurately identifies the main cell types in week 2 cortical organoid dataset.** (A) scoreCT annotation recapitulates the cell type identity of individual cluster on a t-SNE plot. Cells annotated as doublets were found to not be included in any cell type population (NA). (B) t-SNE plots of the expression of known cell type markers *PTN* (Radial glia cells), *NR2F1* (Neuroepithelium) and *LHX9* (Cajal-Retzius neurons). Levels correspond to log-normalized counts.

2.1.6 Comparison with a state-of-the-art enrichment method

Authors from [Diaz-Mejia et al., 2019] evaluated 5 enrichment methods making use of prior marker gene knowledge to assign cell types to cluster centroids. Over all evaluated

datasets, GSVA [Hänzelmann et al., 2013] globally performed the best in annotating the cluster centroids. We evaluated scoreCT against GSVA on 4 selected gold standard datasets provided by the authors: a liver dataset [MacParland et al., 2018], a retinal neurons dataset [Shekhar et al., 2016], the Tabula Muris atlas [Schaum et al., 2018], and a PBMC dataset [Zheng et al., 2017]. We used the gene sets collected by the authors in order to evaluate our method in similar conditions. Overall, scoreCT achieved similar performance to GSVA, being only slightly better with a mean score of 0.58 (against 0.57 for GSVA). However, we didn't need to include more than the top 1000 expressed genes to achieve similar performance, showing the possibility of only keeping a set of K genes to maintain relative performance when assessing cell type identity of clusters. This demonstrates the ability of scoreCT to be used on stored cluster fingerprints to perform fast scoring of reference cell types.

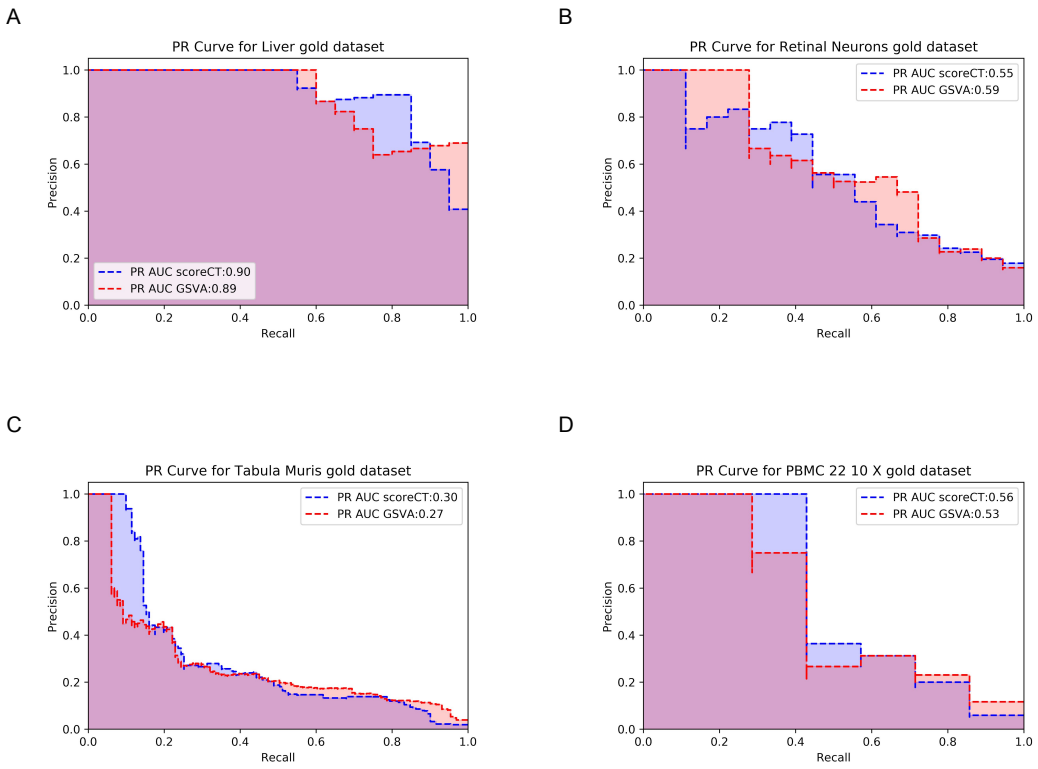

Figure 2.4: **scoreCT comparison with GSVA on selected scRNA-seq datasets.** We compared scoreCT predictions to GSVA, a top performing method evaluated in [Diaz-Mejia et al., 2019] and show precision-recall curves for the (A) liver dataset from [MacParland et al., 2018], (B) retinal neurons dataset from [Shekhar et al., 2016] (C) Tabula Muris atlas [Schaum et al., 2018] and (D) PBMC dataset from [Zheng et al., 2017] .

2.1.7 Stability of parameter choice

To understand the stability of scoreCT results and p-values, we compared the $-log_{10}(p$-values) for different (m, K) combinations and for the main cell type of 3 different datasets: Hepatocytes (liver), BC1A (retinal) and CD8+ T-cells (PBMCs). I found that scoreCT results are robust for a large range of K values, while the number of bins m has less influence once more than 1 bin is chosen. As a rule of thumb, we recommend the user to

use $K = 1000, m = 5$, which works well in most studied cases.

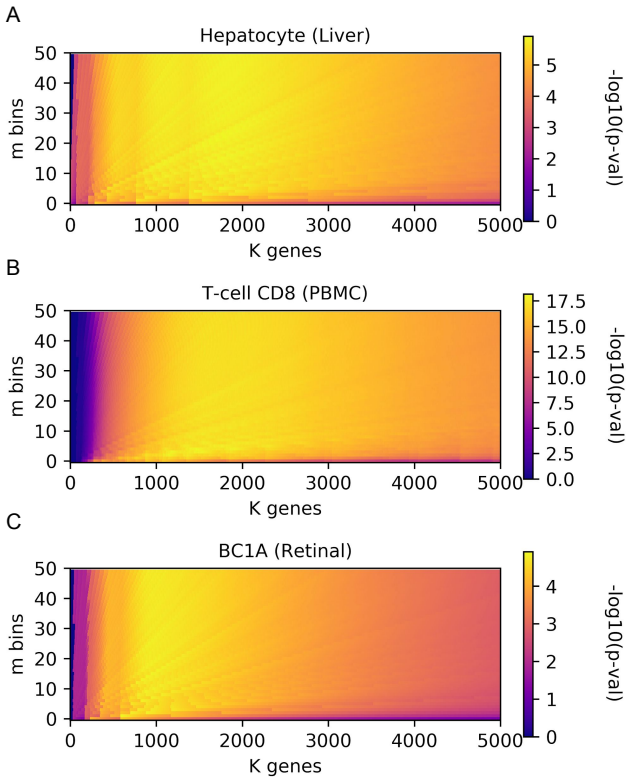

Figure 2.5: **Exploring parameter space for scoreCT.** We ran scoreCT over clusters of known cell types from three different datasets: (A) hepatocyte cluster from a liver dataset, (B) T-cell CD8 cluster from a PBMC dataset, (C) BC1A cluster from a retinal neuron dataset, and reported the $-log10(pvalue)$ associated with the correct cell type as reported by scoreCT, for various combinations of parameters (m, K).

2.2 Ontology-based annotation of cell ensembles

In the previous section, I introduced a simple statistical tool to automate the annotation of single-cell clusters based on a prior knowledge about marker genes and a reduced set of DGEs which I refered to as "cluster fingerprint". In this section, I approach the cell type annotation problem by using ontologies as natural reference structures for mapping

single-cell data to discrete categories representing cell types and marker gene knowledge. Ontologies capture the relationships between cell types and can help lowering uncertainty of annotating cells by mapping to broader categories when annotation is uncertain.

2.2.1 Rationale and preliminary model

Ontologies are natural structures to represent relationships of biological entities such as gene functions or diseases. Cell types, the functional entities of pluricellular organisms, can be seen as an analogous system where the hierarchy can represent an ontology (hierarchical categorization of cell types) or lineages (developmental hierarchies). Ontologies are typically represented as Directed Acyclic Graphs (DAGs) and most often with a single source node. The nature of ontologies provides an interesting property regarding uncertainty, that is that a parent category is a broader definition encapsulating its descendants. This property is of particular interest when we attempt to assign labels to a set of points such as clusters generated by single-cell transcriptomics, since uncertainty about labeling for a given set of potential labels may be resolved by labeling with a broader category of the ontology.

Knowledge about individual cell type is often recapitulated through marker gene sets, which are sets of genes that are characteristic and specific of a given cell type. They have been successfully used in cell type annotation tasks [Zhang et al., 2019a, Pliner et al., 2019] and provide a compact representation of cell types. Therefore, I propose to continue using them as guides for the cell type annotation task.

2.2.1.1 Mathematical formulation

We first formalize the problem. Let $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ be a single-cell dataset with n cells and m gene features. Let $C = \{C_1, C_2, ..., C_k\}$ be a set of k clusters partitioning $\boldsymbol{X}$. Let $G = \{V, E\}$ be a DAG representing our knowledge about the hierarchy of individual cell types in the dataset. Each node $j \in V$ is equipped with a marker gene set μ_j representing our knowledge about the cell type of node j. Because of the hierarchical nature of the problem, a natural required property of the marker sets in G is the *inheritance* of marker genes: since the predecessor nodes of j are broader categories, marker genes are inherited from predecessors up to the single source node of G. We denote the set of predecessor nodes to j as $\Pi(j)$. Specifically:

$$\mu_j = \{ \bigcup_{k \in \{\Pi(j), j\}} \mu_k \}$$

Mapping single-cell clusters to nodes in G

We propose a simple probabilistic mapping of clusters to G. Let $\Phi_{i,j}$ denote the mapping of cluster C_i to node j. We propose to evaluate the following likelihood function:

$$\Phi^*_{i,j} = \underset{V_j \in G}{\operatorname{argmax}} \mathcal{L}(\Phi_{i,j}),$$

$$\mathcal{L}(\Phi_{i,j}) = P(C_i | \mu_j)$$

We further define the quantity $P(C_i|\mu_j)$ in terms of the individual cells contained in C_i,

as followed:

$$P(C_i|\mu_j) = \prod_{c \in C_i} P(c|\mu_j)$$

However, this probability might be hard to evaluate given the amount of noise present in single-cell datasets. We redefine the problem as evaluating the probability ratio of each individual gene expression value in cells of cluster C_i under two different models: (1) the expression value was generated from a marker gene, (2) the expression was generated by a background expression profile $\mathcal{B}$. This formulation is notably used in analyzing DNA motifs in genomics. We therefore introduce 2 empirical probability density functions, $h_{marker}(x)$ and $h_{background}(x)$ which associate a density to a given expression value in a single cell. We now re-define the score to be evaluated in the mapping problem as :

$$\begin{aligned} S_{i,j} &= \log\left(P(C_i|\mu_j)\right) - \log\left(P(C_i|\mathcal{B})\right) \\ &= \sum_{c \in C_i} \sum_{g \in \mu_j} \log(h_{marker}(x_{c,g})) - \log(h_{background}(x_{c,g})) \end{aligned}$$

Here, we would like to note that the score S can also be penalized according to different source of information, such as prior information on the marker set. This is equivalent to putting a prior on $P(\mu_j)$ and $P(\mathcal{B})$. Although not discussed in the main document, we provide a note on this in appendix A.0.1.

2.2.2 Simulating single-cell data derived from DAG structures

In order to evaluate whether this model is suitable for annotating single-cell data, I propose to create synthetic simulated data using a probabilistic model. A first generative

model will produce a DAG representing an arbitrary cell type hierarchy with associated marker genes, and sink nodes (nodes with out-degree 0) of the DAG will be used as a second generative model to generate single-cell count data.

2.2.2.1 A simple probabilistic cell type DAG generator

To generate a synthetic cell type ontology, we propose the following model. The DAG depth d is sampled from a Poisson distribution with parameter λ_d, which can be seen as the average longest path length in our generative model. The number of successors c per node is also sampled from a Poisson distribution with parameter λ_c, which represents the average number of successors per node in the DAG. As it is, the generative mode produces a specific type of DAG: directed trees. In order to transform the tree into a more general DAG, we add a probability p_{DAG} to create a directed edge between a predecessor node and a newly created node. This ensure that the graphs that are generated by the model are single source DAGs.

Finally, we need to include marker genes for each node in the DAG, with the inheritance property described before. Marker genes are randomly (uniformly) sampled without replacement from a list of genes representing the whole transcriptome (*eg.* from a GTF file), and added to each node, including the marker from its predecessors. The number of marker genes m is sampled from another Poisson distribution with parameter λ_m. Taken together, the model is summarized as followed:

$$d \sim Poisson(\lambda_d)$$

$$c \sim Poisson(\lambda_c)$$

$$m \sim Poisson(\lambda_m)$$

2.2.2.2 Generative model for synthetic single-cell count data

Dataset generated by scRNA-Seq using the UMI technology are count data. However, these data are often over-dispersed and zero-inflated because of the phenomenon of dropout. Therefore, we draw inspiration from the literature [Zappia et al., 2017, Delaney et al., 2019] and propose a simple count data generative model.

Each sink node of the DAG uses a noisy Gamma-Poisson mixture to generate over-dispersed count data. The Gamma component has a different shape hyperparameter α if the mean gene count is generated for a marker gene (α_m, "more expression") or a background gene (α_0, "less expression"). The rate β is kept the same in both cases. The true mean count λ_g is sampled from this Gamma distribution and used to sample the true gene count Y_g^0 from a Poisson distribution. This count value is then renoised as followed: A noise level L is sampled from a Uniform distribution with range $[1-e, 1+e]$, and is multiplied with Y_g^0 to give a renoised mean count λ_g^*. This serves as the mean of a new Poisson distribution from which we sample the renoised observed count Y_g^*.

Finally, because single-cell count data are zero-inflated, we add a probability π to dropout the count data. Since dropout effects have been shown to be stronger in less expressed genes [Kharchenko et al., 2014], we set the probability as a sigmoid function of Y_g^*:

$\pi_g = \frac{1}{1+e^{-k(\log(Y_g^*)-x_0)}}$. We summarize the generative model as followed:

$$\alpha_g = \alpha_m \text{ if } g \in \mu_j \ , \ \alpha_0 \text{ else}$$

$$\lambda_g \sim \Gamma(\alpha_g, \beta)$$

$$Y_g^0 \sim Poisson(\lambda_g)$$

$$L \sim U(1-e, 1+e)$$

$$\lambda_g^* = L \times Y_g^0$$

$$Y_g^* \sim Poisson(\lambda_g^*)$$

$$\pi_g = \frac{1}{1+e^{-k(\log(Y_g^*)-x_0)}}$$

$$D \sim Bernoulli(\pi_g)$$

$$Y_g = D \times Y_g^*$$

In the evaluation, we used the following hyperparameters: $\alpha_0 = 2^4$, $\alpha_m = 2^5$, $\beta = 0.1$, $e = 0.2$, $k = 2.5$, $x_0 = 5$. The sampling is repeated for all genes, for N cells, and for the K different sink nodes (cluster-generating nodes).

2.2.2.3 Applying the framework to synthetic data

As a proof of concept, we generated DAGs with average depth $\lambda_d = 3$, average number of successors $\lambda_c = 3$ and $p_{DAG} = 0.2$. We used *Mus musculus* genes with an average number of unique marker genes per node of $\lambda_m = 4$ (setting a threshold of a minimum of 2 unique marker genes per nodes). Sink nodes were used as generative units for cell type clusters, with 500 synthetic cells per cluster. We applied the ontology mapping framework

described previously to the simulated dataset, with the goal to recover the true node labels for each simulated group. We present the result in Fig.2.6. The proposed ontology mapping framework correctly recovers the true label of the node used to generate the single-cell data. We note that more distinct cell types (branching out earlier in the ontology) are easier to recover, which is expected since their expression profile is more unique. Equipped with these preliminary results, we can now apply the framework to real world data.

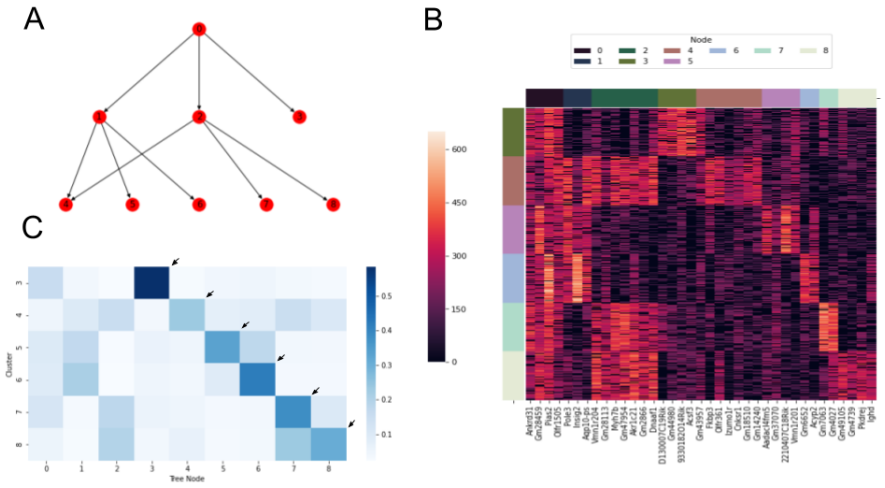

Figure 2.6: **Synthetic ontology and associated simulated single-cell RNA-seq count data.** (A): An example of simulated DAG representing a synthetic cell-type ontology. Sink nodes are used to generate single-cell data corresponding to individual cell types. (B) Heatmap of simulated single-cell data showing the counts of marker genes selected during the generative process. Column colors correspond to the nodes in the ontology, and row colors correspond to the label of the true node used to generate the data. (C) Confusion matrix of assignment probabilities from the proposed ontology mapping framework. The model correctly assigned each cluster to the true node that generated its data.

2.2.3 Applying the annotation framework to real world data

2.2.3.1 Constructing a reference ontology from existing databases

The use of an existing ontology with marker gene information is critical to our method. As it can be challenging for biologists to come up with a specific annotated ontology representing the expected biology of their dataset, we turn to existing databases. Cell Ontology (CO) [Diehl et al., 2016] provides a defined vocabulary for cell types and their relationship, which we can use for the topology of our reference. In order to populate this topology with marker genes (the other core components of our reference), we use the CellMarker database [Zhang et al., 2019b]. This database is composed of manually curated marker genes from the literature for various cell types annotated by tissue of origin, and present the advantage to also use CO identity tags. Therefore, it is simple to link these two existing databases to construct a reference for our framework. We propose a simple method to construct the reference ontology and restrict it to meaningful entities for the target dataset to analyze.

First, we restrict cell types from CellMarkerDB to the tissue(s) of interest. We use these cell type entities to restrict the CO graph to a sub-graph containing entities related to the selected cell types from CellMarkerDB. We then prune nodes from the graph for which we don't have marker gene information (whether the marker set is empty in CellMarkerDB or simply does not exist because this node is an intermediate entity not present in CellMarkerDB). All predecessors of a pruned node are linked to all successors of that node, ensuring that the graph is still a single component. Finally, marker genes are inherited from predecessor nodes. This results in a subset of CO, annotated with

marker genes from CellMarkerDB.

2.2.3.2 Application to the annotation of Peripheral Mononuclear Blood Cells

We applied our ontology mapping framework to a demonstration Peripheral Blood Mononuclear Cells (PBMCs) scRNA-Seq dataset from 10X Genomics. Transcriptomes were normalized and PCA was performed, retaining the first 50 principal components (PCs). We used clusters generated by the Louvain community detection algorithm as input to our method (12 clusters). We constructed the reference using the peripheral blood tissue subset of cells in CellMarkerDB and CellOntology. We compared the results to manual annotation based on a few well-characterized immune cell markers: CST3 for Monocytes, MS4A1 for B-cells, CD3E/D for T-cells, CD8A/B for CD8+ T-cells, GZMB for Natural Killer cells. The results are shown in Fig.2.7. The ontology mapping framework is able to correctly map all the clusters to the correct 5 major cell types of the dataset (Fig.2.7A,B). Despite the ontology containing 29 different immune cell types (Fig.2.7C), the ontology mapping algorithm was able to correctly use the marker information to label clusters. When studying the mean expression of canonical immune markers in each predicted group, we can see that the algorithm was able to contrast the expression of the specific markers with the background to correctly annotate clusters (Fig.2.7D).

However, the task of labelling PBMC major cell types is quite easy. The markers for PBMCs are well-characterized and strongly expressed by the individual populations, making it easy for our algorithm to correctly label each group of cells. To further validate

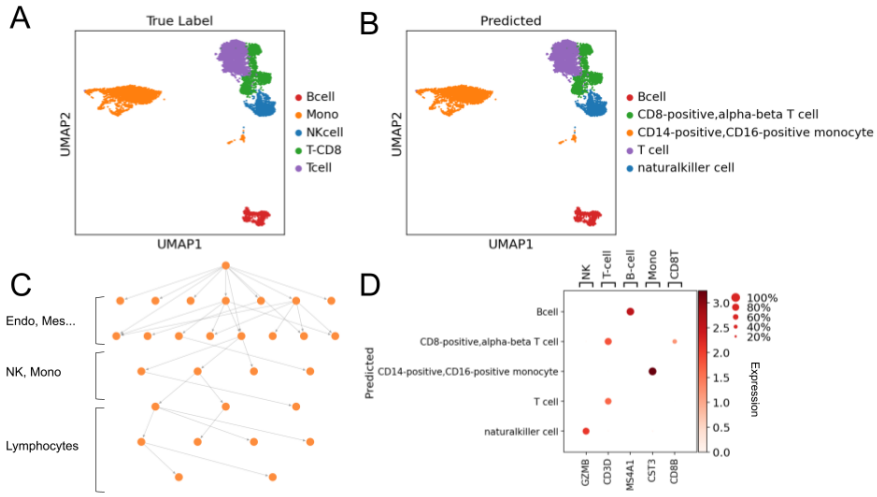

Figure 2.7: **Application of the proposed method to a PBMC dataset** (A): UMAP plot with manually annotated cell type labels. (B) UMAP plot with predicted labels (Maximum Likelihood). (C) Topology of the ontology used for the annotation. The ontology is composed of many immune cell types, many of which are not in the studied dataset. (D) Mean expression of canonical markers for the different cell types of the dataset (the size of each dot represents the percentage of cell expressing the gene).

our method, we need to apply our framework to a more challenging dataset with more diverse cell populations.

2.2.3.3 Application to a liver dataset

We gathered data from a human liver dataset [MacParland et al., 2018]. Transcriptomes were normalized, and we use thed authors original clustering solutions, which comprises 20 distinct clusters. We also gathered their annotation of the cell types, describing an heterogeneous landscape of 8 major cell types. As a reference, we gathered all cell types annotated as part of the liver in CellMarkerDB and constructed our ontology reference from there. Once again, we passed the clusters to our ontology mapping algorithm and compared the predicted labels with the original labels. The results are shown in Fig.2.8.

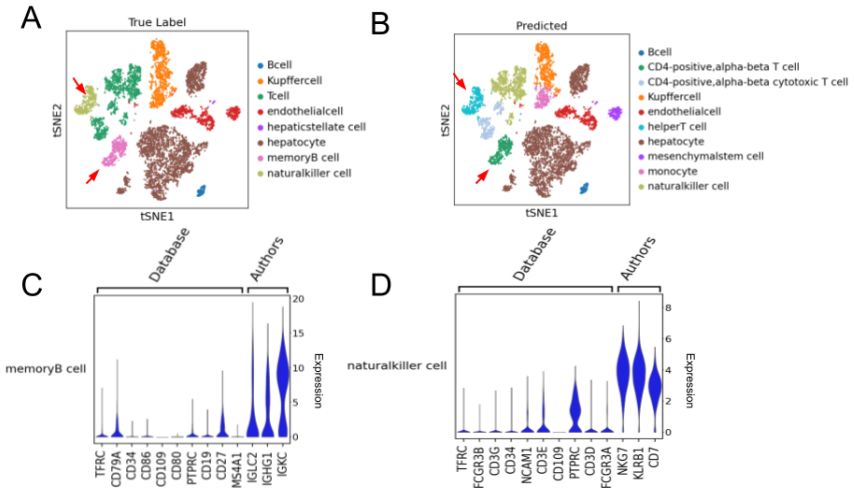

Figure 2.8: **Non-expressed annotated marker genes can lead to incorrect labels** (A): UMAP plot with manually annotated cell type labels. (B) UMAP plot with predicted labels (Maximum Likelihood). (C) Violin plot of marker expressions from the database and those used by the original authors for the annotation in memory B cell cluster. (D) Violin plot of marker expressions from the database and those used by the original authors for the annotation in the natural killer cells cluster.

While some clusters are correctly labeled (notably hepatocytes clusters), an important part of them are incorrectly labeled by the model (Fig.2.8A,B). Notably, most of the immune cell types get incorrectly labeled. We demonstrate why this is the case using the example of the memory B cells and Natural Killer cells (red arrows in Fig.2.8A). When investigating the discrepancies between the markers for those cell types in CellMarkerDB and those used by the authors for the annotation in [MacParland et al., 2018], we see that markers originating from CellMarkerDB have zero or very limited expression in their respective groups. In contrast, markers used by the original study are highly expressed. This showcase an example where the knowledge in CellMarkerDB is incomplete, or not specific enough for scRNA-Seq datasets. We discuss these issues and propose potential solutions in the following section.

2.2.4 Challenges met when using prior marker genes information

2.2.4.1 Incomplete knowledge about markers in CellMarkerDB

As we showed in the results on the human liver dataset, incorrect labelling can come from incomplete or erroneous knowledge about marker genes in the reference. In fact, CellMarkerDB was built using manual curation of marker genes from the literature. An important portion of these markers most likely come from assays relying on protein (immunostaining, blots...). mRNA level of expression does not always correlate to the corresponding protein level. This poses a major challenge when using marker genes to annotate single-cell data, since the signal for certain markers may be weaker at the mRNA level. Secondly, the knowledge about marker genes in this database might be incomplete, in the sense that some important markers might not be present for certain cell types (*eg.* NKG7 for Natural Killer cells). Together, these problems hinder our ability to readily use CellMarkerDB for annotation purposes. We note that, in the literature, none of the methods using prior marker gene knowledge for annotation use CellMarkerDB as a source of marker, but rather custom made sets that are hand curated [Zhang et al., 2019a, Pliner et al., 2019]. This illustrates the need for further work on the curation of marker gene databases.

2.2.4.2 Using pre-annotated datasets to further curate CellMarkerDB and improve annotation

Following these preliminary results, the need for a method that could distinguish poor and powerful markers from CellMarkerDB is clear. I supervised the work from a master

student (Alex Pearson) on building a Naive Bayes classifier using both pre-annotated scRNA-seq datasets (training data) and marker gene knowledge from CellMarkerDB for classification purposes. The trained classifier outputs "optimally binarized" data that can be used to decide if a marker gene from CellMarkerDB was helpful in the classification of a particular dataset. This obviously provides a framework to classify new datasets, but also help to determine whether a marker from CellMarkerDB needs to be removed from the reference or not. We demonstrated that the Naive Bayes classifier has competitive performance with other type of models on predicting cell type labels, but also emphasized how it could be use to study the predictive power of individual marker genes from CellMarkerDB. The details of the method are available in Alex Pearson's master thesis [Pearson, Alexander, 2020].

Chapter III: Inferring gene modules activity at the single-cell level using sparse Variational Autoencoders

Genes are expressed as coordinated units in gene modules. Studying these gene modules is crucial to understand the role of each individual cell in a larger biological context. In this section, I propose a novel deep generative architecture for Variational Autoencoders (VAE) called VEGA, which incorporates prior knowledge about gene modules (such as pathways, Gene Regulatory Networks (GRNs) or cell type marker sets) to achieve interpretability over the model's latent space and infer the activity of various gene programs in single cell populations. Finally, I introduce a Bayesian testing procedure for scoring differentially activated programs (DAPs) and show that it suffers less bias than enrichment tests such as GSEA. The model is implemented in the Scanpy and scvi-tools ecosystems [Gayoso et al., 2022], and available at `https://github.com/LucasESBS/vega`. VEGA is published in *Nature Communications* [Seninge et al., 2021].

3.1 Background

3.1.1 Modularity of gene expression

One of the key insights of modern biology is that genes are often co-regulated and transcribed as coordinated units, with those gene modules corresponding to key functions of the cell [Zhang and Zhang, 2013]. For example, genes involved in important metabolic functions such as oxydative phosphorylation have been shown to be co-regulated, both in human and mouse [van Waveren and Moraes, 2008]. Notably, *PGC-1α/NRF1* transcription factors have been shown to control the transcription of both nuclear and mitochondrial genes involved in the cell respiratory chain, such as ATP synthase components (*eg. ATP5G2*) or parts of the respiratory complexes (*eg. NDUFA38*) [Satoh et al., 2013, Hood et al., 2015]. Those types of core functions are often conserved across organisms [Stuart et al., 2003], but the particular interplay between gene modules and specific cell populations is yet to be fully elucidated. Particularly, external stimuli can lead to drastic changes in gene expression programs of core cell functions, and studying these changes is of major interest to link those stimuli to functional responses of specific cell types. scRNA-seq provides an unprecedented insight on gene co-expression and co-regulation at the single-cell level, and therefore enables researchers to study functional gene module behaviours at the single-cell level.

To computationally study the activity of these gene modules, several efforts have been made. The development of databases such as the Molecular Signature Database (MSigDB) [Subramanian et al., 2005] to gather curated gene sets corresponding to core cellular

processes or functions have been particularly import. Enrichment score methods such as the popular GSEA [Subramanian et al., 2005] have been developed to study differences at the gene module level in RNA-seq samples. Briefly, a differential expression test is performed between two groups of interest and used to rank genes (according to test statistic or fold-change). Then, a rank-based statistic is used to assess the significance of the enrichment of different modules, often using reference gene sets from curated databases. A variant of GSEA, single-sample GSEA (ssGSEA) was later developed to infer gene module activity within each individual sample rather than the difference between two groups [Barbie et al., 2009].

Principal Component Analysis (PCA) [Jolliffe, 1986] has also been used to summarized the main source of variation in the data. Briefly, PCA decomposes a gene expression matrix into "metagenes", which are formed from linear combination of the original gene features. Those metagenes can be regarded as core source of variation such as co-expressed genes acting together into a module. However, relating those metagenes to known biological functions can be challenging and requires further investigation of the PCA loadings. Integrating prior knowledge about gene modules directly into factor analysis models would be highly desirable to study the activity of known biological functions.

3.1.2 Autoencoders and Variational Autoencoders

3.1.2.1 Autoencoder architectures

Autoencoders [Hinton, 2006] are popular neural network architectures used to efficiently learn how to reconstruct data through a latent code, robust to noise. A simple mathematical description of such model can be formulated as such. Let $f_\phi : \mathcal{X} \to \mathcal{Z}$ and $g_\theta : \mathcal{Z} \to \mathcal{X}$ be respectively the encoding and decoding functions, where $\{\phi, \theta\}$ is a set of learnable parameters. In its simplest formulation, the goal is to learn $\{\phi, \theta\}$ minimizing the following criterion:

$$\mathcal{L}(\mathbf{X}) = \|\mathbf{X} - [g_\theta \circ f_\phi](\mathbf{X})\|_2^2 \tag{3.1}$$

which we will refer to as the *reconstruction error* (RE). Note that minimizing this criterion is analogous to minimizing the negative log-likelihood (NLL) in Maximum Likelihood Estimate (MLE) procedures under the assumption that the data $\mathbf{X}$ are generated from a Gaussian distribution.

This kind of network present the advantage of allowing to efficiently encode a datum $\boldsymbol{x} \in \mathbb{R}^g$ into a latent code $\boldsymbol{z} \in \mathbb{R}^h$, allowing for data compression (when $h < g$) and learning useful data properties.

3.1.2.2 Variational Inference and Variational Autoencoders

We introduced autoencoders, which allows to map a dataset $\mathbf{X}$ onto a latent code $\mathbf{z}$. It might be desirable to see $\mathbf{z}$ as a set of latent random variables that encodes some properties about the data, and from which new examples can be sampled. We thus reframe the autoencoder framework through an *inference* network (encoder) with same parameters ϕ, and a *generative* network (decoder) with parameter θ. The names of *inference* and *generative* networks are chosen on purpose: the encoder can be seen as *inferring* the parameters for the set of random variables $\mathbf{z}$ from the data, while the decoder *generates* new examples from samples of the multivariate distribution $\mathbf{z}$.

In this setting, we are interested in computing the true posterior distribution $p(\mathbf{z}|\mathbf{X})$ during training. However, this is often intractable because it involves computing a costly integral. To solve this, we can use Variational Inference (VI) [Jordan et al., 1998], a powerful alternative to Markov Chain Monte Carlo (MCMC) for posterior approximation when the amount of data is large. The goal of VI is to treat the problem as an *optimization* problem, where we try to find a distribution $q(\mathbf{z}|\mathbf{X})$ member of a family of densities $\mathcal{Q}$ that best approximates the true posterior $p(\mathbf{z}|\mathbf{X})$ by minimizing the Kullblack-Leibler (KL) divergence with the posterior, *ie*:

$$q^*(\mathbf{z}|\mathbf{X}) = \underset{q(\mathbf{z}|\mathbf{X})\in\mathcal{Q}}{\operatorname{argmin}} KL(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z}|\mathbf{X}))$$

In this new formulation, the generative model described here is called a Variational

Autoencoder (VAE) [Kingma and Welling, 2014]. We want to find the parameters ϕ for the inference network such that

$$q_\phi^*(\mathbf{z}|\mathbf{X}) = \underset{\phi}{\operatorname{argmin}}\, KL(q_\phi(\mathbf{z}|\mathbf{X})||p(\mathbf{z}|\mathbf{X}))$$

From there, we can derive the Evidence of Lower Bound (ELBO), which is used as an objective to be maximized to train the VAE:

$$ELBO(\mathbf{X}, \phi, \theta) = \mathbb{E}_{q_\phi(\mathbf{z}|X)}\left[\log\left(p_\theta(\mathbf{X}|\mathbf{z})\right)\right] - KL(q_\phi(\mathbf{z}|\mathbf{X})||p(\mathbf{z})) \tag{3.2}$$

Intuitevely, this can be seen as maximizing the log-likelihood of the reconstructed data with respect to samples from the variational posterior, while minimizing the KL-divergence between our variational posterior and $p(\mathbf{z})$. It is worth noting that the negative ELBO is often used as an objective to minimize, since it can be implemented as a reconstruction loss and a regularization by the KL-divergence.

A multivariate Gaussian distribution with diagonal covariance is often chosen as the variational posterior $q(\mathbf{z}|\mathbf{X})$, which leads to an inference network parametrized with $\{\mu_\phi, \Sigma_\phi\}$. This allows a closed-form computation of the KL term in (3.2) (since the prior $p(\mathbf{z})$ is often set to a standard normal distribution $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$). However, because the first term in (3.2) has no closed-form solution, we draw Monte Carlo samples from $q_\phi(\mathbf{z}|\mathbf{X})$ to approximate the expectation in the ELBO and make the VAE optimization tractable.

While it is impossible to use standard backpropagation through stochastic operation like sampling, we can use the reparametrization trick [Kingma and Welling, 2014] to sample from the variational posterior, while maintaining the gradient with respects to the weights of the inference network: $\mathbf{z} = \mu_\phi(\mathbf{X}) + \Sigma_\phi^{\frac{1}{2}}(\mathbf{X}) * \epsilon$, with $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

3.2 Incorporating prior biological knowledge about gene modules into a VAE architecture: VEGA

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-26017-0 OPEN

VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics

Lucas Seninge[1], Ioannis Anastopoulos[1], Hongxu Ding[1✉] & Joshua Stuart[1✉]

Deep learning architectures such as variational autoencoders have revolutionized the analysis of transcriptomics data. However, the latent space of these variational autoencoders offers little to no interpretability. To provide further biological insights, we introduce a novel sparse Variational Autoencoder architecture, VEGA (VAE Enhanced by Gene Annotations), whose decoder wiring mirrors user-provided gene modules, providing direct interpretability to the latent variables. We demonstrate the performance of VEGA in diverse biological contexts using pathways, gene regulatory networks and cell type identities as the gene modules that define its latent space. VEGA successfully recapitulates the mechanism of cellular-specific response to treatments, the status of master regulators as well as jointly revealing the cell type and cellular state identity in developing cells. We envision the approach could serve as an explanatory biological model for development and drug treatment experiments.

[1] Department of Biomolecular Engineering and Genomics Institute, University of California, Santa Cruz, CA, USA. ✉email: hding16@ucsc.edu; jstuart@ucsc.edu

Recent advances in single-cell RNA sequencing (scRNA-Seq) technologies have enabled the characterization of cellular states at an unprecedented scale and resolution[1]. Among the many widely-used frameworks for analyzing complex transcriptomic patterns in single cells, artificial neural networks (ANNs) such as autoencoders (AEs)[2] have emerged as powerful tools. AEs are neural networks that transform an input dataset into a decoded representation while minimizing the information loss[3]. The diversity in their architectural design makes AEs suitable to tackle various important challenges of scRNA-Seq analysis, such as dimensionality reduction[4], clustering[5], and data denoising[6].

More recently, deep generative models such as variational autoencoders[7] (VAEs) have proven to be extremely useful for the probabilistic modeling of single-cell transcriptomes, such as scVI and scGen[8–10]. While standard AEs learn to reconstruct an input dataset, deep generative architectures explicitly model and learn the true data distribution, which allows a broader set of queries to be addressed. While deep generative models have shown impressive performance for their dedicated modeling tasks, they often lack interpretability thus cannot offer a biologically meaningful latent representation of transcriptomes. For example, latent perturbation vectors extracted with scGen cannot be directly related to gene module variations[10].

Integration of prior knowledge about gene modules to aid interpretability has already been successfully applied to transcriptomics data. DCell[11] is a deep neural network integrating the hierarchical information about the molecular subsystems involved in cellular processes to guide supervised learning tasks, such as predicting growth in yeast. Such a model yields an informative biological interpretation of predictions by investigating the activation of the different subsystems embedded in the model's architecture. However, this model only works in a supervised learning setting where the goal is to predict a phenotypic outcome. On the other hand, f-scLVM[12] is a Bayesian hierarchical model with explicit prior biological knowledge specification to infer the activity of latent factors as a priori characterized gene modules. While this approach enables the modeling of single-cell transcriptomes in an interpretable manner, the computational cost of the inference algorithm, as well as the absence of inference for out-of-sample data, make the development of more efficient approaches highly desirable.

Here we propose VEGA (VAE enhanced by gene annotations), a VAE with a sparse linear decoder informed by biological networks. VEGA offers an interpretable latent space to represent various biological information, e.g., the status of biological pathways or the activity of transcriptional regulators. Specifically, the scope of VEGA is twofold, (1) encoding data over an interpretable latent space and (2) inferring gene module activities for out-of-sample data.

Results

Architectural design of VEGA. To create a readily interpretable VAE, we propose a novel architecture we refer to as VEGA (VAE enhanced by gene annotations) where the decoder (generative part) connections of the neural network are guided by gene module membership as recorded in gene annotation databases (e.g., Gene Ontology, PANTHER, MolSigDB, or Reactome) (Fig. 1a). In many standard VAE implementations, the information bottleneck of the encoder-decoder architecture often represents latent variables modeled as a multivariate normal distribution. Despite providing highly informative representations of the input data, VAE latent variables are in general hard to interpret. Svensson et al.[13] proposed using a linear decoder which directly connects latent variables to genes, providing interpretability similar to that offered by standard factor models such as PCA. Although providing valuable insights, such an approach requires further statistical enrichment tests on the weights of the decoder to infer biological processes contributing to the single-cell expression dataset.

In contrast to previous approaches, VEGA implements a sparse architecture that explicitly reflects knowledge about gene regulation. In the service of biological pathways, genes work together in gene modules, regulated by common transcription factors that often produce correlated expression. Thus, if a given scRNA-Seq dataset X reflects the patterns of known gene modules, then it is possible for a VAE to learn a compact representation of the data by incorporating those modules as latent variables Z. VAEs use multiple layers to approximate the latent variable distribution and produce a low dimensional, nonlinear representation of the original feature data. Importantly, the first and last layers directly connect to the input or predicted features and so can be fashioned to depict intuitive groupings. Standard VAEs use a fully connected layer for both the encoding first layer and the decoding final layer (SFig. 1aiv). Instead, VEGA uses a gene membership mask $\mathbf{M}$ to select a subset of trainable weights in the decoder layer that are determined by a given set of gene modules (see Methods). The mask is applied to the weights that connect to the predicted output features to yield an interpretation of the latent variable layer where each latent variable is viewed as a specific gene module, henceforth referred to as a gene module variable (GMV). Specifically, the generative part of VEGA (decoder) maintains a link from a GMV to an output gene only if this gene is annotated to be a member of this specific gene module. The two main advantages of this design are (1) the latent variables are directly interpretable as the activity of biological modules and (2) the flexibility in the gene module specification allows it to generalize to different biological abstractions (such as pathways, gene regulatory networks (GRNs), or even cell types) and can be taken from any of several curated databases of gene sets (such as MSigDB[14], Reactome pathways[15], inferred GRNs[16]). Additionally, VEGA incorporates information about covariates such as technical replicates in its latent space. This can be used to alleviate batch effects, as it has been demonstrated in previous deep generative models for single-cell data[9] (Fig. 1a and SFig. 2)

Note that it is possible to implement gene module sparseness in the encoder half of the neural network (inference part), in addition to (or in place of) the decoder half (generative part), which gives three possible VAE architectures that we considered for single-cell RNA-seq analysis (SFig. 1ai–iii). As expected, we found that the GMV-guided designs resulted in decent although slightly worse performance compared to the full architecture (SFig. 1c). Among these options, we chose the sparse decoding architecture over the others for its improved separation of known cellular states and types in the Kang et al. PBMC data[17] (SFig. 1b). Intuitively, using a deep encoder maintains a full VAE's inference capacity to capture a potentially complex latent space while together with a sparse decoder approximates the posterior distribution of GMV activities $p(Z|X)$ to provide interpretation over gene modules. Additionally, we found that VEGA benefits from having a trainable, sparse decoder to adequately capture the biological signal of a dataset compared to simpler pathway transformations (SFig. 3).

Recapitulating biological information over an interpretable latent space. We asked if VEGA could recapitulate the status of biological pathways by applying it to a published and well-studied peripheral blood mononuclear cells (PBMCs) dataset stimulated with the chemokine interferon-β[17] (Methods). We first found that VEGA is able to capture cell types and stimulation status using the Reactome collection of processes and pathways[15] in the GMV decoding layer

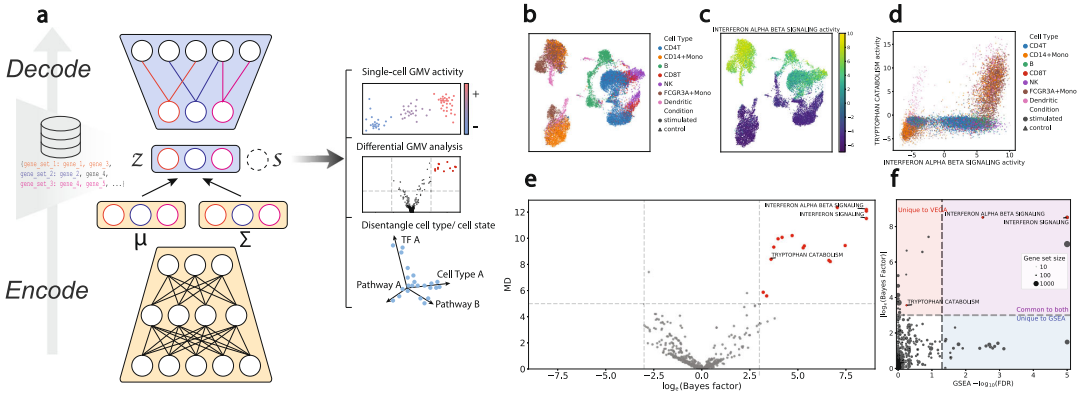

Fig. 1 Designing a novel VAE architecture with interpretable latent space. a Overview of the VEGA model. Composed of a deep nonlinear encoder (μ, Σ) and a masked linear decoder, VEGA represents single-cell transcriptomics data into a lower-dimensional interpretable latent space z that approximates a set of user-supplied gene modules (GMV). Additionally, VEGA can integrate batch information as another variable s to condition its generative process on batch labels. **b** UMAP embedding of the latent space of VEGA retains the biological signal of the Kang et al. PBMCs dataset[17]. **c** Inferred interferon-alpha/beta signaling pathway activity segregates stimulated cells from the control population. **d** Bivariate GMV plot showing the ability of the model to recover the tryptophan catabolism activity, an innate (Dendritic cells, FCGR3A+monocytes, CD14+monocytes) immune cell-specific response to the perturbation. **e** Volcano plot showing differentially active GMVs between stimulated and control innate immune cells. The red dots indicate GMVs with $|\log_e(\text{Bayes Factor})| > 3$ and a mean absolute difference (MD) in the latent space of at least 5. **f** Comparison of VEGA Bayes Factor with GSEA $-\log_{10}$(FDR). The size of the dots indicates the gene set size. The red, blue, and purple quadrants correspond respectively to significant hits unique to our model, unique to GSEA, and common to both.

(Fig. 1b). Specifically, we found that the interferon-α/β signaling GMV activity segregates stimulated and naive cells, confirming the ability of VEGA to capture pathway activity in its latent space (Fig. 1c, d). We further examined other known biological pathways involved in interferon-induced immune cell activation and found cell-type-specific activation of certain cellular processes. For example, tryptophan catabolism response to interferon separates innate immune cells (Dendritic cells, FCGR3A+monocytes, and CD14+monocytes) from adaptive immune cells (NK cells, T-cell CD8, T-cell CD4, and B cells) (Fig. 1d), as previously investigated[18,19]. Together, these results suggest that VEGA's GMV's reflect the expected major biological pathways in PBMCs and therefore may be useful for other datasets to project cells into an interpretable space, allowing investigation of cell-type-specific patterns at the cellular process level.

We next asked whether the differential activities of the GMVs accurately contrast pathway states as a function of a specific, experimentally controlled context.

For this purpose, we propose a similar Bayesian hypothesis testing procedure as introduced by Lopez et al.[9] to study the difference in GMV activities. As VEGA models the posterior distribution of each GMV, we can formulate mutually exclusive hypotheses similar to differential gene expression tests (i.e., GMVs are activated at different levels). We can approximate the posterior probability of these hypotheses through Monte Carlo sampling of VEGA's latent variable distribution. The ratio of hypothesis probabilities corresponds to the Bayes Factor[20] (BF, see Methods).

When applied to innate immune cells in the stimulated vs control groups of the Kang et al.[17] dataset, the BF analysis found GMVs that correspond to pathways expected to be activated in the stimulated groups (interferon signaling, tryptophan catabolism; $|\log_e(\text{BF})| > 3$, Fig. 1e). We compared the GMV BFs with the false discovery rate (FDR) values of the standard GSEA toolkit (Methods, Fig. 1f). While both methods found the expected activation of the interferon-α/β signaling pathway GMV in the stimulated groups, GSEA missed the tryptophan catabolism activation in innate immune cells (Fig. 1f). Overall, VEGA seems more robust than GSEA to gene set size bias (Fig. 1f and SFig. 4), suggesting it may emphasize more context-relevant pathways. Additionally, the differential GMV activity test can be applied in a cell-type-specific fashion (similar to one-vs-rest differential gene expression analyses). We found that such a procedure yields informative results in terms of cell type-specific biological processes activated independently of perturbation status (SFig. 5 and Supplementary Data 1).

Large-scale investigation of biological responses to drug treatments in cell lines. Next, we investigated whether VEGA could detect patterns of drug responses in large-scale experiments over cancer cell lines, such as the data introduced in recent experimental protocols like MIX-Seq[21]. To this end, we gathered single-cell data for 97 cancer cell lines under five different conditions: 24 h DMSO treatment (control), 24 h Trametinib treatment (MEK inhibitor), 24 h Dabrafenib treatment (Mutated BRAF inhibitor), 24 h Navitoclax treatment (Bcl-2 inhibitor), and 24 h BRD3379 treatment (tool compound with unknown mode of action, MoA) (Methods). We trained one model for each different drug treatment (four models in total) by combining the drug treatment dataset and the control group (DMSO dataset), initializing the GMVs of VEGA with the hallmark gene sets from MSigDB[22] to focus on core cellular processes. Overall, each model was able to separate cell lines and treatment conditions in the GMV space (Fig. 2a, and SFig. 6). For Trametinib notably, the important change in G2M checkpoint GMV activity (decrease in the treated condition) agrees with the expected MoA of a MEK inhibitor[23,24] (Fig. 2b). Next, we sought to investigate whether we could recapitulate the pattern of biological responses between control and treated conditions for each cell line/drug treatment pair. For each pair, we computed GMV BFs to approximate differential pathway activities between the two conditions. The resulting heatmap can be used to understand and interpret patterns of response over all experimental conditions (Fig. 2c). As found when visually investigating the low dimensional

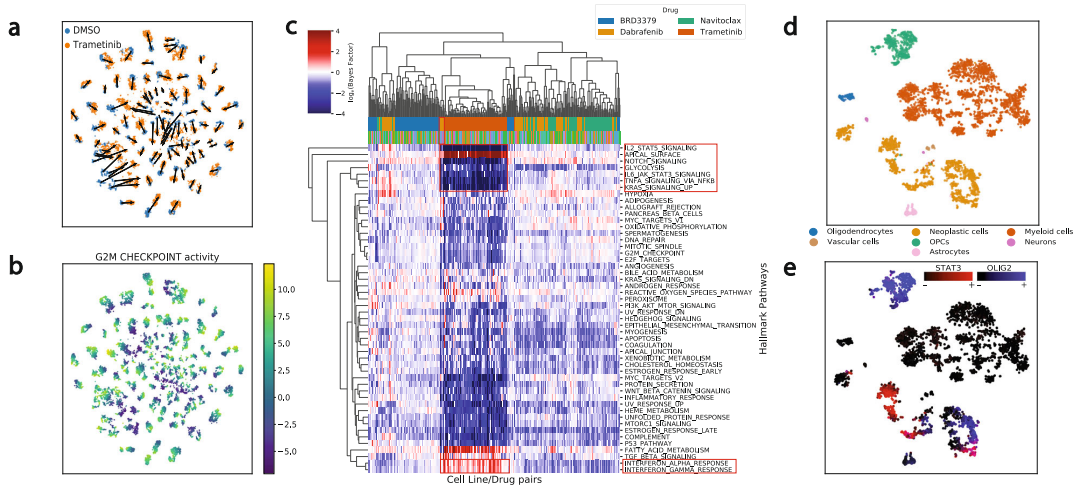

Fig. 2 The flexibility in the latent space specification sheds light on the activity of core cellular processes and transcription factors. a tSNE embedding of the latent space of VEGA for the MIX-Seq data[21]. The color indicates the treatment condition, and the arrow indicates the median shift in coordinates of each cell line between the two conditions. **b** Inferred G2M checkpoint activity of each cells, showing a decreased activity in the treated condition, as expected from the MoA of Trametinib. **c** Heatmap with hierarchical clustering showing the average $\log_e$(Bayes Factor) of each pathway for each cell line/drug treatment pair (test between DMSO and treatment condition). Each row corresponds to a hallmark gene set and each column to a different cell line/drug pair. The first row of color indicates the drug, and the second row of color indicates the tissue identity (Tissue legend available in SFig. 2). Highlighted cell lines correspond to BRAF-mutant melanoma. Highlighted activities correspond to Trametinib-specific responses. **d** tSNE embedding of the latent space of the model for the glioblastoma dataset[29], colored by cell type or **e** Inferred activity of the master regulators STAT3 and OLIG2.

embedding of each dataset (Fig. 2a and SFig. 6a–c), Trametinib resulted in the strongest transcriptional response of all studied drugs. Notably, the Trametinib-specific interferon-α and interferon-γ response was correctly recapitulated in VEGA's latent space, consistent with previous experimental work[25] and the findings reported by the original MIX-Seq authors[21]. Furthermore, we found that Dabrafenib-treated BRAF-mutant melanoma cell lines exhibited larger $|\log_e(BF)|$ than other Dabrafenib-treated cell lines (average $|\log_e(BF)|$ of 0.763 vs 0.668 for other cell lines), clustering with the Trametinib-treated cell lines as reported in the MIX-Seq study (Fig. 2c and SFig. 6d). Overall, the results presented here agree with the previous gene set analysis results on this dataset, and demonstrate VEGA's GMVs can recapitulate patterns of drug response in large-scale experiments.

Gene regulatory analysis of glioblastoma reveals stratification of neoplastic cells. As previously mentioned, one of VEGA's strengths is the flexibility in the specification of the GMV connectivity, as any gene module can be used in the decoder. Transcription factors often exert tight regulation of gene expression in many biological contexts[26]. Analyzing the activity of transcriptional regulators is important in understanding biological states like cell types or diseases, as dysregulation in their activity can have a dramatic impact on gene expression programs and phenotypes[27,28]. To this end, we investigated whether using master transcriptional regulators as the GMVs could help understand the underlying GRNs in the context of a single-cell glioblastoma (GBM) dataset[29]. We used the GBM ARACNe[16] network reported in Carro et al.[28] to guide the structural design of our model. Specifically, VEGA's GMVs were set to the reported transcription factors and the connectivity matrix **M**, defining the GMVs decoding architecture, was created from the set of predicted target genes of each transcription factor. After training, we found that the pre-annotated cell types were well-separated in the latent space (Fig. 2d). We examined the activity of STAT3 and OLIG2, two well-known master regulators of the mesenchymal (MES) and proneural (PN) GBM subtypes, respectively. We confirmed that their GMV activity was largely anticorrelated in neoplastic cells (Fig. 2e). Additionally, OLIG2, a known master regulator of oligodendrocytes differentiation[30], was inferred as activated in oligodendrocyte precursor cells (OPCs). These results demonstrate that VEGA is able to home-in on the relevant transcriptional regulators when the decoder wiring is extended to model known factor-to-target relationships.

Combining cell type and cellular state representations refines cortical organoid development analysis. A great challenge of modern cellular biology is to identify and define cell types and cellular states, at the level of individual cells, in order to systematically study homeostasis and disease development under a common vocabulary. In a typical single-cell study, a few "marker sets" will be known, each containing a list of genes having expected expression patterns for some of the cell types of interest. Leveraging such marker sets often provides clues and helps orient data analysis. We asked whether the information recorded in such marker sets could be used in VEGA to produce a disentangled representation of cell types and cellular states. To this end, we added a GMV z_t, with appropriate entries in **M**, for each latent cell type t in addition to the Reactome pathway GMVs already in VEGA's model.

We applied VEGA to a dataset of cells assayed during the early development of cortical organoids from Field et al.[31], including all of the major cell types defined in the study as GMVs (Fig. 3a). After training, we found that the activity of each marker set GMV was able to correctly segregate its corresponding cell type as annotated by the original authors (Fig. 3b–d). Moreover, in a one-vs-rest differential GMV analysis setting for each cell type population, the activity of the corresponding marker set GMV showed significant enrichment ($|\log_e(BF)| > 3$), which suggests using GMV BFs could help annotate the cell types of unknown clusters (Fig. 3e). We further noted that the

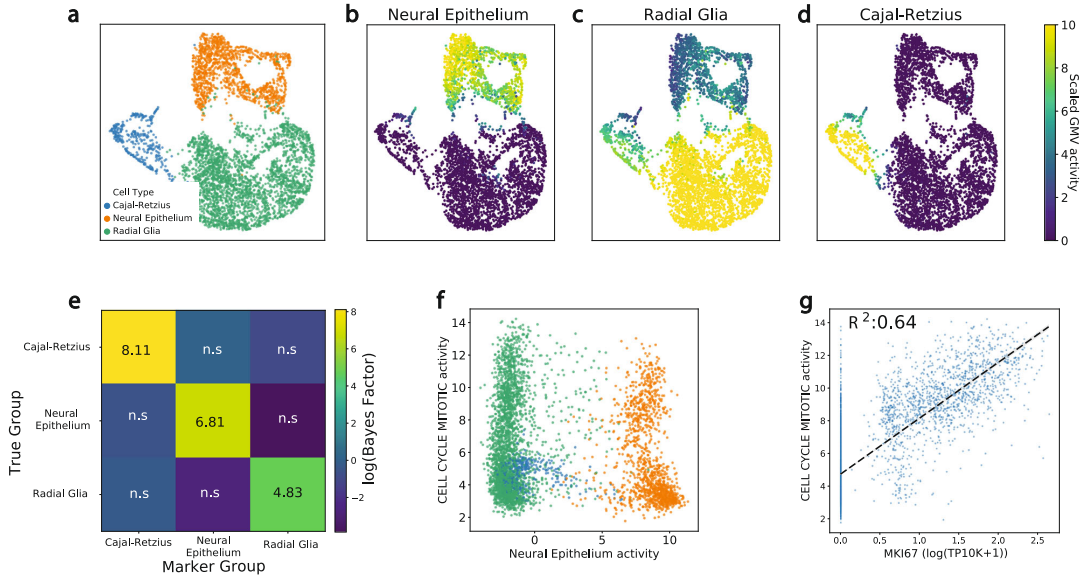

Fig. 3 Disentangling cellular states and cell types in the early development of cortical organoids. a UMAP embedding of the latent space of our model for the week 2 cortical organoid dataset[31]. The cell type annotation corresponds to the original paper annotation. **b**, **c**, **d** The inferred activity of each cell type GMVs (as defined by marker genes) correctly identifies the three main subpopulations of cells. **e** One-vs-rest differential GMV analysis of each cell type population provides a statistical significance for each cell type signature. The significance threshold for positive enrichment was set to $\log_e(BF) > 3$. **f** Identification of dividing and quiescent subpopulations of neural progenitors using pathway and cell-type activity projection. **g** CELL_CYCLE_MITOTIC pathway activity correctly identifies dividing cells as reported by its correlation with MKI67 gene expression (an external canonical marker of dividing cells).

most differentially activated GMVs were coherent in the context of early brain development (SFig. 7 and Supplementary Data 2). To study whether VEGA could separate cell type identity from cellular states such as dividing vs quiescent cell populations, we projected the dataset into two components: (1) the cell type GMV representing the neural epithelium marker set (a type of early brain progenitor) and (2) the cell state GMV representing the cell cycle mitotic pathway activity (Fig. 3f). As discussed previously, the activity of the neural epithelium GMV separated the neural epithelium cells from the rest of the dataset, while the activity of the cell cycle mitotic pathway GMV separated quiescent from actively dividing cells in the two progenitors populations (radial glia cells and neural epithelium). To validate that the cells identified as dividing were proliferating, we studied the correlation between the cell cycle mitotic pathway GMV activity and the expression of the MKI67 gene, a canonical marker of proliferation (external validator not present in the cell cycle mitotic pathway set) (Fig. 3g). Overall, the expression of MKI67 correlates well with the inferred activity of the cell cycle mitotic pathway GMV ($R^2 = 0.64$). Together, these results demonstrate VEGA's potential use to jointly infer cell type and state for different populations of cells, as combining different sources of information (pathways, master regulators, and cell type markers) in the latent space can shed light on different aspects of the identity of a single-cell.

Generalization of the inference process to out-of-sample data. We next asked whether VEGA could generalize to correctly infer an interpretable latent representation of data unseen at the time of training (out-of-sample data). To this end, we evaluated VEGA in two settings. In the first case, we measured the biological generalization of VEGA's inference by holding out (cell type, condition) pairs during training. Specifically, we investigated whether the inferred GMV activities for held-out cells were conveying the same biological information as to when this population is seen at the time of training. To this end, we removed one cell type of the stimulated condition during training, and then inferred the GMV activities for that held-out population (out-of-sample) and compared them to the GMV activities learned from the fully trained model. The experiment was conducted using the Kang et al.[17] PBMC dataset. In the second case, we estimated the "technical generalization" of VEGA's inference by training on one dataset (study A) and then evaluating on a second dataset (study B) that contains only control cells. We used the Kang et al.[17] PBMC dataset as study A and the Zheng et al.[32] dataset as study B.

For the biological generalization test, we first checked that the distribution of the interferon-α/β signaling pathway GMV activity in the out-of-sample stimulated CD4 T cells matched the inferred activity in the in-sample CD4 T cells (Fig. 4a). To perform a more systematic comparison of the inferred latent space between out-of-sample and in-sample cells, we used the differential BF procedure (Methods) between (1) stimulated in-sample cells and control cells for a given cell type (model trained with the whole dataset) and (2) stimulated out-of-sample cells and control cells for the same cell type (model trained with one cell type/condition pair left out), and checked the amount of overlaps in the top 50 differentially activated GMVs (Fig. 4b). The results suggested consistency between the in-sample and out-of-sample differentially activated GMVs, with an average 72% overlap. To further evaluate the capacity of data reconstruction, we measured the R^2 between the original and decoded data in the in-sample and out-of-sample settings (Fig. 4c). We found that the R^2 decreases only marginally in the out-of-sample setting, confirming the ability of the model to generalize to unseen data produced in a similar experimental setting.

For the technical generalization test, we again checked that the interferon-α/β signaling pathway GMV activity distribution of study B encoded control CD4 T cells matched that of study A

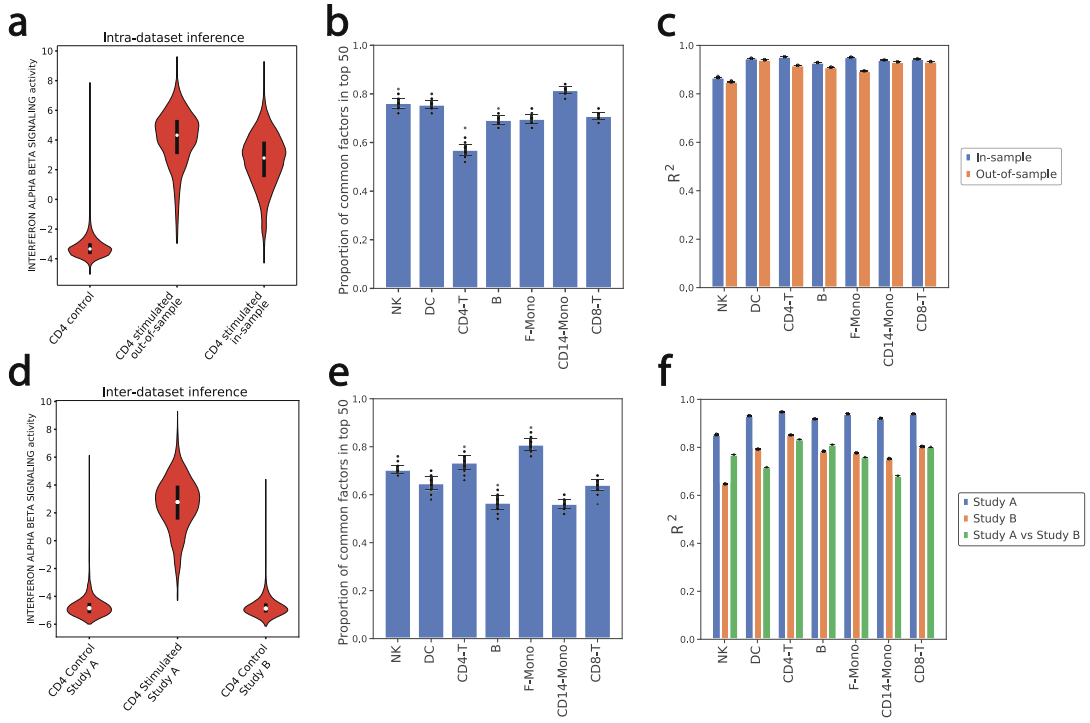

Fig. 4 Generalization of VEGA architecture to out-of-sample data. a Violin plot ($n = 10{,}000$ randomly sampled cells per condition) representing the distribution of the interferon-α/β pathway activity in control CD4-T cells, stimulated CD4 T cells unseen at the time of training (out-of-sample), and stimulated CD4-T cells when included in the training procedure (in-sample). Boxes inside the violins represent the median of the distribution bounded by the first and third quartile. Violin limits correspond to data extrema. **b** Proportion of overlap in the top 50 differentially activated GMVs in the in-sample and out-of-sample settings with stimulated vs control differential procedures for the seven main cell types in the study. Data were presented as mean values ± standard deviation over 100 random sampling. **c** R^2 between the mean expression of real and reconstructed cells in the in-sample and out-of-sample settings for the seven main cell types of the study. Data were presented as mean values ± standard deviation over 100 random samplings. **d** Violin plot ($n = 2000$ randomly sampled cells per condition) of distribution of the interferon-α/β pathway activity in control CD4-T cells of study A (Kang et al.[17]), stimulated CD4-T cells of study A and control CD4-T cells of study B (Zheng et al.[32]). Boxes inside the violins represent the median of the distribution bounded by the first and third quartile. Violin limits correspond to data extrema. **e** Proportion of overlap in the top 50 differentially activated GMVs of each study with one-vs-rest differential procedures for the control cells of the seven main PBMC cell types. Data were presented as mean values ± standard deviation over 100 random samplings. **f** R^2 between the mean expression of real and reconstructed cells of study A (Study A), mean expression of real and reconstructed cells of study B (Study B), and mean expression of real cells of study A and real cells of study B (Study A vs Study B). Data were presented as mean values ± standard deviation over 100 random samplings.

control CD4 T cells (Fig. 4d). We also investigated whether the top 50 differential GMVs of each cell type in a "one-vs-rest" differential setting for the control cells of study A overlapped with a similar procedure performed on the control cells of study B (Fig. 4e). We found that on average 67% of the top 50 differential GMVs for study A overlap with those of study B, showing that the model can generalize across studies unseen at the time of training. We then asked whether the model can use the inferred latent space to accurately reconstruct the original expression profiles of both studies. We found that the R^2 between original and reconstructed cells of study B, although lower than those for study A, improves upon the baseline correlation between the expression profiles of study A vs study B for most of the cell types (Fig. 4f).

Discussion

In this study, we introduced VEGA, a novel VAE architecture with a decoder inspired by known biology to infer the activity of various gene modules at the level of individual cells. By encoding single-cell transcriptomics data into an interpretable latent space specified a priori, our method provides a fast and efficient way of analyzing the activity of various biological abstractions in different contexts. In contrast, previous approaches used a posteriori interpretations of the latent variables to infer modules. VEGA's flexibility in the specification of the latent space paves the way for analyzing the activity of biological modules such as pathways, transcriptional regulators, and cell type-specific modules. We illustrated how VEGA could be used to simultaneously investigate both cell type and cell state of cell subpopulations, in both control and experimentally perturbed conditions. Additionally, the weights of decoder connections provide direct interpretability of the relationship between the latent variables and the original features. For example, the decoder's weights could be used to contrast interaction confidence in inferred GRNs or to rank genes by their importance in a certain biological module in a data-driven way. We further note that it was possible to modify VEGA's architecture, following the same rationale as widely-used scVI[9] and linear scVI[13], such that it could handle count data in place of normalized expression profiles (SFig. 8).

The clear limitations of the current architecture resides in the sparse, single-layer decoder of the model. In fact, such an architectural design prevents the further improvement of generalizability and robustness. As a consequence, the generative capacity of VEGA is limited. For example, while VEGA theoretically could be used for interpretable response prediction using latent vector arithmetics in a similar fashion to scGen[10], VEGA's limited generative capacity sacrifices predictive performance for biological interpretability of the latent space. We believe advanced insights in network biology, e.g., multi-layer GRNs that can describe regulatory machinery more comprehensively, could alleviate these limitations. This would open the possibility to perform targeted, in-silico activation, and repression of biological programs on specific cell populations to study its effect on development or disease progression. On the other hand, hard-coded connections of the linear decoder do not leave any room for correcting prior knowledge about gene modules when the context requires it, as is the case in other latent variable models such as f-scLVM[12]. In fact, prior biological knowledge obtained from existing databases like MSigDB can be incomplete or not context-specific, as additional unannotated genes can play an important role in certain gene modules. In parallel to our work on VEGA, Rybakov et al.[33] introduced a regularization procedure to incorporate prior knowledge from gene annotation databases via a penalty term on the weights of the linear decoder. We demonstrated that VEGA performs comparatively to their interpretable autoencoder (SFig. 9), and that their approach is complementary to the unique attributes of VEGA and can be used to recover missing gene-GMV links in a data-driven fashion (SFig. 10).

In summary, we found VEGA useful for understanding the response of specific cell type populations to different perturbations, providing interpretable insights on biological module activity. The variational aspect of VEGA provides an advantage for addressing queries about samples, or sample groups, that are not possible with a regular AE. We illustrated how the latent multivariate Gaussian distribution of the VAE, which approximates the posterior probability of every GMV, enables a new kind of differential test to be performed. The BF reflects the likelihood of how active a gene module is in one condition compared to another, providing a straightforward method to perform differential activity analysis using the RNA-Seq data similar to the approach described by Lopez et al.[9]. Other types of queries are possible, for example, to automate the annotation of unsupervised clusters or modules that dynamically change across the branches of an inferred cellular trajectory. We envision VEGA could also be useful to prioritize drugs based on pathway expression in cancer, as studying the response of specific cell populations may inform drug sensitivity and resistance. Integrating drug response prediction models with such explanatory models could benefit designing novel therapeutic strategies.

Methods

The VEGA architecture. VEGA is a deep generative VAE that aims at maximizing the likelihood of a single-cell dataset X under a generative process[7,10] described as:

$$p(X|\theta) = \int p(X|Z,\theta)p(Z|\theta)dZ, \tag{1}$$

with θ being the learnable parameters of a neural network. VEGA uses a set of latent variables Z that explicitly represent sets of genes (gene modules), such as pathways, GRNs, or cell type marker sets. To enforce the VAE to interpret a dataset from the viewpoint of a set of gene modules, VEGA's decoder part is made up of a single, masked, linear layer. Specifically, the connection of this layer, between latent node $z^{(j)}$ and gene features, are specified using a binary mask $\mathbf{M}$ in which $M_{i,j}$ is true if gene i is a member of gene module j and false otherwise. We refer to each latent variable $z^{(j)}$ as a GMV since each provides a view of the data constrained to the subset of genes for a distinct gene module j. During training, gradients associated with masked (false) weights are "zeroed out" such that backpropagation only applies to weights originating from a user-supplied given gene set. Additionally, the weights of the decoder are constrained to be nonnegative ($w \geq 0$) to maintain interpretability as to the directionality of gene module activity.

Having explicitly specified the connections between genes and latent variables in the decoder of VEGA (generative part), we incentivize that the latent space represents a biological module activity interpretation of the data. We choose to model the GMVs as a multivariate normal distribution, parametrized by our inference network with learnable parameters ϕ As such, the distribution of the Z latent variables can be expressed as:

$$q(Z|X,\phi) = \mathcal{N}(\mu_\phi(X), \Sigma_\phi(X)) \tag{2}$$

This choice of variational distribution is common and has proven to work well in previous single-cell studies[9,10]. Following similar standard VAE implementations[7,10], the objective to be maximized during training is the evidence of lower bound (ELBO):

$$\mathcal{L}(X) = \mathrm{E}_{q(Z|X,\phi)}\left[\log p(X|Z,\theta)\right] - KL(q(Z|X,\phi)||p(Z|\theta)) \tag{3}$$

where the expectation over the variational distribution can be approximated using Monte Carlo integration over a minibatch of data, and the Kullblack–Leibler divergence term has a closed-form solution as we set the prior to:

$$p(Z|\theta) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{4}$$

The reparametrization trick[7] is used when sampling VEGA's variational distribution to allow standard backpropagation to be applied when training the model.

To retain information of genes that are not present in our pre-annotated biological networks, we add additional fully connected nodes to the latent space of our model. This has two effects: (1) it allows VEGA to model the expression of unannotated genes, which could be crucial for a good reconstruction of the data during training, and (2) it can help capture additional variance of the data that is unexplained by the provided gene modules, considerably improving the training of the model. The number of additional fully connected nodes can be determined based on a trade-off between model performances and the loss of information encoded by pre-annotated GMV nodes. As a rule of thumb, we recommend picking 16 or fewer extra FC nodes to preserve the biological signals encoded by GMV nodes (SFig. 11).

Additionally, the diagonal covariance prior used in the latent space modeling discourages GMVs from being correlated. Thus, the VAE may be forced to choose an arbitrary gene set among many equally informative but overlapping sets and could fail to reveal a key annotation. To address this issue, we add a dropout layer to the latent space of the model. This has been shown to force the VAE to preserve redundancy between latent variables[34], which is applicable when the gene annotation database used to initialize VEGA's latent space contains overlapping gene sets (SFig. 12).

Finally, batch information or other categorical covariates can be encoded via extra nodes in the latent space, conditioning the generative process of VEGA on this additional covariate information (SFig. 2).

Measuring differential GMVs activity of the latent space with Bayes Factor (BF). The difference in the activity of genes and/or pathways is often of interest when contrasting two different groups of cells. To this end, we draw inspiration from the Bayesian differential gene expression procedure introduced in Lopez et al.[9] and propose a similar differential GMV analysis procedure. We follow a similar notation as Lopez et al. For a given GMV k, a pair of cells (x_a, x_b) and their respective group ID (s_a, s_b) (e.g., two different treatment conditions), our two mutually exclusive hypotheses are:

$$\mathcal{H}_0^k := \mathrm{E}_s\left[z_a^k\right] > \mathrm{E}_s\left[z_b^k\right] \text{ vs. } \mathcal{H}_1^k := \mathrm{E}_s\left[z_a^k\right] \leq \mathrm{E}_s\left[z_b^k\right] \tag{5}$$

This can intuitively be seen as testing whether a cell has a higher mean GMV activation than another, the expectation representing empirical frequency. We evaluate the most probable hypothesis by studying the log-Bayes factor K defined as:

$$K = \log_e \frac{p(\mathcal{H}_0^k|x_a,x_b)}{p(\mathcal{H}_1^k|x_a,x_b)} \tag{6}$$

Here, the sign of K tells us which hypothesis is more likely, and the magnitude of K encodes a significance level. Having access to the conditional posterior distribution q(Z|X) over the GMVs activation (the encoding part of VEGA), we can approximate each hypothesis' probability distribution as:

$$p\left(\mathcal{H}_0^k|x_a,x_b\right) \approx \sum_s p(s) \int_{\sup.(z_a)} \int_{\sup.(z_b)} p\left(z_a^k > z_b^k\right) dq\left(z_a^k|x_a\right) dq\left(z_b^k|x_b\right) \tag{7}$$

where $p(s)$ is the relative abundance of cells in group s, and the integrals are approximated with direct Monte Carlo sampling.

Similarly to Lopez et al.[9], assuming cells are independent, we can compute the average Bayes factor across many cell pairs randomly sampled from each group respectively. This helps us decide whether a GMV is activated at a higher frequency in one group or the other. Through the paper, we consider GMVs to be significantly differentially activated if the absolute value of K is greater than 3 (equivalent to an odds ratio of ≈20)[9,20].

Datasets and preprocessing

Kang et al. dataset. The Kang et al.[17] dataset consisted of two groups of PBMCs, one control and one stimulated with interferon-β. We chose to use the same preprocessing steps as described by scGen authors[10], using the Scanpy package[35]. Briefly, cells were annotated using the maximum correlation to one of the eight original cell type clusters identified, using an average of the top 20 cluster genes. Megakaryocytes were removed due to uncertainty about their annotation. Then data were filtered to remove cells with less than 500 genes expressed and genes expressed in five or less cells, using the `scanpy.pp.filter_genes()` and `scanpy.pp.filter_cells()` functions. Count per cells were then normalized and log-transformed using the `scanpy.pp.normalize_per_cell()` and `scanpy.pp.log1p()` functions, and we selected the top 6998 highly variable genes with `scanpy.pp.highly_variable_genes()`, resulting in a final dataset of 18,868 cells. Raw data is available at GSE96583. We used the same preprocessing functions for the rest of the datasets unless specified otherwise.

Zheng et al. dataset. The Zheng et al.[32] dataset consists of 3K PBMCs from a healthy donor. After filtering the cells, the count per cells were normalized and log-transformed. We then subset the genes to use the same 6998 genes of the Kang et al. PBMC dataset. The final dataset has 2623 cells and 6998 genes. Raw data are available at https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k.

MIX-seq dataset. The MIX-seq[21] datasets were obtained from https://figshare.com/s/139f64b495dea9d88c70, and we used the data from experiment 3 to have enough cells to carry a smooth training of our model. For the five available datasets (97 cell lines treated with respectively DMSO, Trametinib, Dabrafenib, Navitoclax, and BRD3379), we removed cells with 200 or less expressed genes, and genes expressed in less than three cells. We then normalized the number of counts per cell, and log-transformed the data. Finally, each dataset that was a drug treatment experiment was combined with a copy of the control dataset (DMSO treatment), and we extracted the top 5000 highly variable genes. This resulted in final datasets of size (16,732 cells and 4999 genes) for the Trametinib+DMSO data, (16,942 cells and 5000 genes) for the Dabrafenib+DMSO data, (14,507 cells and 5000 genes) for the Navitoclax+DMSO data, and (15,304 cells and 5000 genes) for the BRD3379+DMSO data.

Darmanis et al. dataset. The raw GBM data from Darmanis et al.[29] were obtained from http://www.gbmseq.org/ and preprocessed as followed: we removed cells with 200 or less expressed genes, and genes expressed in three or less cells. Count per cells were normalized and data were then log-transformed. Finally, we restricted the transcriptome to the top 6999 highly variable genes. The final dataset had a total of 3566 cells. Raw data is available at GSE84465.

Field et al. dataset. The cortical organoid data from Field et al.[31] was processed similarly to the GBM dataset. After normalization and highly variable genes selection, the dataset had a total of 4378 cells, with 6999 genes. Raw data is available at GSE106245.

Shekhar et al. dataset. The mouse retina dataset from Shekhar et al.[36] was processed as described (see https://github.com/broadinstitute/BipolarCell2016). Briefly, we removed cells with more than 10% mitochondrial transcripts. Then, cells with less than 500 genes were removed, and genes expressed in less than 30 cells and with less than 60 transcripts across all cells were removed. To be able to use human versions of gene modules from the Reactome database, we performed one-to-one ortholog mapping of mouse transcripts to human transcripts using BioMart from the Ensembl project[37]. Genes without human orthologs were removed. We saved a version of the dataset with the raw count data for the selected genes/cells, and further processed the data by normalizing and log-transforming the libraries. Finally, we restricted the transcriptome to the top 4000 highly variable genes. The same highly variable genes were used to subset the raw QC count matrix. The final datasets (for both count and log-normalized versions) had a total of 27,499 cells, coming from two technical batches. We used the annotation with 15 cell types from the original authors. Raw data is available at GSE81904.

Choice of gene annotations for the latent space of VEGA. When initializing the latent space of our model, we chose to use pre-annotated gene sets from the Molecular Signature Database (MSigDB, at https://www.gsea-msigdb.org/gsea/msigdb/collections.jsp#C2)[14]. In particular, we chose to use the hallmark gene sets annotation (50 gene sets) or the Reactome database (674 gene sets). Reactome was used for the stimulated PBMCs analysis, and MSigDB's Hallmark gene sets were used in the MIX-Seq analysis part of this study. For the gene regulatory network analysis of GBM cells, we derived an ARACNe[16,38] network from bulk RNA-Seq samples of GBM. Specifically, this network was obtained from a previously published paper[39] and repurposed for the study of GBM single-cell transcriptomics profiles.

For the cell type marker genes in the cortical organoid analysis, we contacted the authors to obtain relevant genes used in annotating those cell types. The GMT file including these marker genes can be found along with the reproducibility code at https://github.com/LucasESBS/vega-reproducibility.

Dimensionality reduction for visualization. For visualizing datasets, we used the UMAP algorithm[40] as implemented in the Scanpy[35] python package, using `scanpy.pp.neighbors()` for the k-NN computation with `n_neighbors=15`, and `scanpy.tl.umap()` for the actual dimensionality reduction. We used default parameters except for the `min_dist` parameter that we set to 0.5. We also used tSNE[41] implemented as `sklearn.manifold.TSNE()` in the sklearn python package[42], with default parameters.

Comparison with GSEA. We ran Gene Set Enrichment Analysis https://www.zotero.org/google-docs/?grfpAv[14] (GSEA) using the `prerank` function from the gseapy package in Python. Briefly, we calculated differential expression scores for each gene between the control and treatment group using a Wilcoxon rank-sum test, as implemented in the `scanpy.tl.rank_genes_groups()` functionality of the Scanpy package https://www.zotero.org/google-docs/?fKytT7[35]. We ranked genes according to their test statistics, and ran GSEA using the gseapy package function `gseapy.prerank()` with the following settings: a minimum gene set size `min_size=5`, a maximum gene set size `max_size=1000`, and a number of permutations `permutation_num=1000`. We ranked gene sets according to their FDR and considered significant hits when FDR ≤0.05. When the FDR returned by GSEA was equal to 0, we replaced it with 1e-5 (to avoid math error when taking the logarithm).

Batch correction comparison. To assess batch information integration in VEGA's latent space, we compared the average silhouette scores on batch labels from the Shekhar et al. retina dataset of (1) PCA with 50 principal components (computed using `scanpy.tl.pca()` function), (2) linear scVI[13] as implemented in the `scvi-tools` package ran on the count version of the dataset with following parameters: AnnData object setup with `batch_key=Batch`, model initialized with `n_hidden=800`, `n_layers=2`, `dropout_rate=0.2`, `n_latent=677`, training performed with `max_epochs=300`, `early_stopping=True`, `lr=5e-4`, `train_size=0.8`, `early_stopping_patience=20`, and (3) VEGA with following parameters: AnnData object setup with `batch_key=atch`, model initialized using the REACTOME pathway database with three extra FC nodes to initialize the latent space and the same training hyperparameters as linear scVI.

Evaluation metrics. Silhouette scores were calculated to evaluate the separation of cell types and states in the latent space of our model. We used Euclidean distance in the latent space to compute the silhouette coefficient of each cell i defined as :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{8}$$

where $a(i)$ and $b(i)$ are respectively the mean intra-cluster distance and the mean nearest-cluster distance for cell i. We used either the stimulation or cell type labels from Kang et al.[17] to assess the biological relevance of the latent space of our model. The sklearn package[17] `silhouette_score()` implementation was used for computation. For computing correlations throughout the paper, we used the function `numpy.corrcoef()` from the Numpy package[43].

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

All of the datasets analyzed in this manuscript are publicly available. Please see the section Datasets and preprocessing of Methods for details. These datasets are also downloadable at https://github.com/LucasESBS/vega-reproducibility.

Code availability

The package and API for VEGA is available at https://github.com/LucasESBS/vega/tree/vega_dev[44]. The code and data to reproduce the results of this manuscript is available at https://github.com/LucasESBS/vega-reproducibility.

Received: 11 January 2021; Accepted: 13 September 2021;
Published online: 28 September 2021

References

1. Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C. & Teichmann, S. A. The technology and biology of single-cell RNA sequencing. *Mol. Cell* **58**, 610–620 (2015).
2. Hinton, G. E. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).
3. Baldi, P. Autoencoders, unsupervised learning, and deep architectures. *Proc. Mach. Learn. Res.* **27**, 37_49 (2012).
4. Wang, D. & Gu, J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. *Genomics, Proteomics Bioinformatics* **16**, 320–331 (2018).

5. Geddes, T. A. et al. Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis. *BMC Bioinformatics* **20**, 660 (2019).
6. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
7. Kingma, D.P. & Welling, M. Auto-encoding variational Bayes. Preprint at *arXiv*:1312.6114 [cs, stat] (2014).
8. Way, G. P. & Greene, C. S. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pac. Symp. Biocomput.* **23**, 80–91 (2018).
9. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
10. Lotfollahi, M., Wolf, A. F. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods* **16**, 715–721 (2019).
11. Ma, J. et al. Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* **15**, 290–298 (2018).
12. Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C. & Stegle, O. f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol.* **18**, 212 (2017).
13. Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics* **36**, 3418–3421 (2020).
14. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA* **102**, 15545–15550 (2005).
15. Jassal, B. et al. The reactome pathway knowledgebase. *Nucleic Acids Res.* **48**, D498–D503 (2020).
16. Margolin, A. A. et al. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7**, S7 (2006).
17. Kang, H. M. et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36**(January), 89–94 (2018). Number: 1 Publisher: Nature Publishing Group.
18. Mellor, A. L., Lemos, H. & Huang, L. Indoleamine 2,3-dioxygenase and tolerance: where are we now? *Front. Immunol.* **8**, 1360 (2017).
19. Sorgdrager, F. J. H., Naudé, P. J. W., Kema, I. P., Nollen, E. A. &d De Deyn, P. P. Tryptophan metabolism in inflammaging: from biomarker to therapeutic target. *Front. Immunol.* **10**, 2565 (2019).
20. Kass, R. E. & Raftery, A. E. Bayes factors. *J. Am. Stat. Assoc.* **90**, 773–795 (1995).
21. McFarland, J. M. et al. Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nat. Commun.* **11**, 4296 (2020).
22. Liberzon, A. et al. The molecular signatures database hallmark gene set collection. *Cell Syst.* **1**, 417–425 (2015).
23. Kurata, K. et al. Growth arrest by activated BRAF and MEK inhibition in human anaplastic thyroid cancer cells. *Int. J. Oncol.* **49**, 2303–2308 (2016).
24. Joshi, M., Rice, S. J., Liu, X., Miller, B. & Belani, C. P. Trametinib with or without vemurafenib in BRAF mutated non-small cell lung cancer. *PLoS ONE.* **10**, e0118210 (2015).
25. Lulli, D., Carbone, M. L. & Pastore, S. The MEK inhibitors trametinib and cobimetinib induce a type I interferon response in human keratinocytes. *Int. J. Mol. Sci.* **18**, 2227 (2017).
26. Spitz, F. & Furlong, E. E. M. Transcription factors: from enhancer binding to developmental control. *Nat. Rev. Genet.* **13**, 613–626 (2012).
27. Lee, T. I. & Young, R. A. Transcriptional regulation and its misregulation in disease. *Cell* **152**, 1237–1251 (2013).
28. Carro, M. S. et al. The transcriptional network for mesenchymal transformation of brain tumours. *Nature* **463**, 318–325 (2010).
29. Darmanis, S. et al. Single-cell RNA-seq analysis of infiltrating neoplastic cells at the migrating front of human glioblastoma. *Cell Reports* **21**, 1399–1410 (2017).
30. Lu, R. Q. et al. Common developmental requirement for olig function indicates a motor neuron/oligodendrocyte connection. *Cell* **109**, 75–86 (2002).
31. Field, A. R. et al. Structurally conserved primate LncRNAs are transiently expressed during human cortical differentiation and influence cell-type-specific genes. *Stem Cell Reports* **12**, 245–257 (2019).
32. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 1–12 (2017).
33. Rybakov, S., Lotfollahi, M., Theis, F. J. & Wolf, A. F. Learning interpretable latent autoencoder representations with annotations of feature sets. Preprint at *bioRxiv* https://doi.org/10.1101/2020.12.02.401182 (2020).
34. Yeung, S., Kannan, A., Dauphin, Y. & Fei-Fei, L. Tackling over-pruning in variational autoencoders. Preprint at *arXiv*: 1706.03643 (2017).
35. Wolf, A. F., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
36. Shekhar, K. et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell* **166**, 1308–1323.e30 (2016).
37. Howe, K. L. et al. Ensembl 2021. *Nucleic Acids Res.* **49**, D884–D891 (2021).
38. Lachmann, A., Giorgi, F. M., Lopez, G. & Califano, A. ARACNe-AP: gene network reverse engineering through adaptive partitioning inference of mutual information. *Bioinformatics* **32**, 2233–2235 (2016).
39. Ding, H. et al. Quantitative assessment of protein activity in orphan tissues and single cells using the metaVIPER algorithm. *Nat. Commun.* **9**, 1471 (2018).
40. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. Preprint at *arXiv*: 1802.03426 (2020).
41. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
42. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
43. Harris, C. R. et al. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
44. Seninge, L., Anastopoulos, I., Ding, H. & Stuart, J. VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *vega*. https://doi.org/10.5281/zenodo.5338892 (2021).

Acknowledgements

L.S. was supported by the Schmidt Futures Foundation SF 857 and by the National Institute Of Mental Health of the National Institutes of Health award R01MH120295. J.M.S. was supported by a grant 5R01GM109031 from the NIGMS. J.S. and H.D. were supported by a grant from the Chan-Zuckerberg Initiative's Human Cell Atlas portals project. H.D. was supported by a gift from Seagate Technology. J.S. was supported by grant GC1R-06673-C from the California Institute for Regenerative Medicine's Center of Excellence for Stem Cell Genomics. The authors would like to thank Dr. David Haussler and Dr. Sofie Salama for their support. L.S. would also like to thank David Parks for the useful feedback during the early development of the method. We also would like to thank Dr. Maximilian Haeussler for the feedback on the manuscript.

Author contributions

L.S. and I.A. conceived the idea. L.S. implemented the method and gathered the data. L.S. and I.A. performed the analysis. H.D. and J.S. supervised the research. All authors contributed to the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-26017-0.

Correspondence and requests for materials should be addressed to Hongxu Ding or Joshua Stuart.

Peer review information *Nature Communications* thanks Gokcen Eraslan and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

3.3 Extensions of VEGA's inference process to other tasks

3.3.1 Controlling for effect size in detecting differentially activated programs

In the VEGA section, I introduced a Bayesian testing procedure for detecting differentially activated (DA) GMVs inspired by previous work on differential gene expression (DGE) in the context of VAEs [Lopez et al., 2018]. However, this testing procedure can lead to detect irrelevant gene modules as being significantly activated, since it does not control for effect size (in this context, an arbitrary measure of how different the activity of a given GMV is between 2 groups of cells). Controlling for effect size has been a standard procedure in differential gene expression tests for a long time [Costa-Silva et al., 2017]: the most popular method, DESeq2 [Love et al., 2014] offers a composite null hypothesis aimed at detecting genes whose (absolute) log-fold change (LFC) is greater than a threshold. Recently, this type of null hypothesis implicitly controlling for effect size in DGE has been incorporated into deep generative modelling framework such as scVI [Lopez et al., 2018, Boyeau et al., 2019]. I once again draw inspiration from this work to propose an alternative null hypothesis controlling for effect size in the context of detecting DA GMVs.

3.3.1.1 Formalization

In this section, I use the same notation as [Boyeau et al., 2019] with application to GMVs instead of genes. Let a and b be two cells, and let $r^k_{a,b} = f(z^k_a, z^k_b)$ be a random variable representing the cells difference in activity of the GMV k. Here f represents a

function capturing some biological difference between its input. In DGE analysis, f is set to the LFC defined as $f(x_{ag}, x_{bg}) = \log_2(x_{ag}) - \log_2(x_{bg})$. However, in the context of our interpretable VAE the activity of GMVs is represented as normally distributed latent variables, meaning that LFCs are not readily available as a measure to capture biological changes in the magnitude of GMV activities. I propose to use a simpler measure of difference as $f(z_a^k, z_b^k) = z_a^k - z_b^k$. I discuss the implication of such a choice in a later section.

We formulate the two mutually exclusive hypotheses:

$$H_0^k := |r_{a,b}^k| \leq \delta \quad \text{versus} \quad H_1^k := |r_{a,b}^k| > \delta$$

where δ is an arbitrary threshold set to detect GMVs whose activity shift is large enough to be biologically meaningful. Similarly to DGE, this design implicitly excludes programs whose shift might be significant but not large enough to be interesting in practice.

The posterior distribution for the difference measure is given by:

$$p(r_{a,b}^k | x_a, x_b) = \iint p\left(r_{a,b}^k | z_a^k, z_b^k\right) dp\left(z_a^k | x_a\right) dp\left(z_b^k | x_b\right)$$

Similarly to the previous Bayesian hypothesis testing procedure, the posterior probabilities $p(H_0^k | x_a, x_b)$ and $p(H_1^k | x_a, x_b)$ can be approximated using Monte Carlo sampling of the variational posterior $q(z_n | x_n)$, and generalized for groups of cells rather than pairs. For short in the following we will note $p(H_1^k | x_a, x_b)$ as p^k.

3.3.1.2 Decision rule for calling gene module variables as differentially activated

In the work on DGE inspiring this procedure [Boyeau et al., 2019], the authors describe 2 ways to decide whether to call a gene as DE or not. The first one is to consider Bayesian decision theory [Berger, 1985] and call a gene g as differentially expressed when:

$$p^g \geq \alpha, \text{ where } \alpha = \frac{K_1}{K_1 + K_0}$$

with K_0 being the cost of a false negative and K_1 the cost of a false positive. This can be extended without loss of generality to calling a GMV k as differentially activated, replacing p^g with p^k.

Another strategy is to control the false discovery rate (FDR) [Muller et al., 2006, Cui et al., 2015]. Specifically, as explicited in [Cui et al., 2015] we rank GMVs in decreasing order of p^k. We write the ordered estimated posterior probabilities as $p^{(1)} > p^{(2)} > \ > p^{(M)}$, where M is the number of GMVs in the model. If we call "differentially activated" the set GMVs such that $p^k \geq p^{(d)}$, for each $d = 1, ..., M$, then the corresponding posterior expected FDR is defined:

$$\widehat{\text{FDR}}_d = \frac{\sum_{i=1}^{d}(1 - p^{(i)})}{d}$$

We then call GMVs as differentially activated when $\widehat{\text{FDR}}_d < q_0$. In practice we set $q_0 = 0.1$.

3.3.1.3 Implications of the choice for measuring latent variable differences

One implication of the choice $f(z_a^k, z_b^k) = z_a^k - z_b^k$ is that it is not as readily interpretable as LFC in terms of meaningful difference in activities, and therefore the threshold δ can be hard to set in practice for users. The properties of VEGA's latent space (continuous support over $\mathbb{R}$) prevent the use of a definition of f involving the log function. A potential future direction would be to study variation of VEGA with a log-normally distributed latent space. This would amount to embedding each cell into a M-dimensional simplex, with the interesting property of $z_n^k \geq 0$ for all cell n and all GMV k. Such parametrization of the variational posterior in single-cell VAEs has been studied and shown to work well in practice [Svensson et al., 2020], providing support for extending the use of LFC to differential activity testing in interpretable deep generative models such as VEGA.

3.4 Future work on interpretable deep generative models in single-cell analysis

3.4.1 Improving the integration of prior biological knowledge in deep generative models

Following the publication of VEGA, different strategies have been proposed to improve the integration of prior knowledge in deep generative models. I demonstrated (concurrently to [Rybakov et al., 2020]) that it is possible to soften the prior knowledge constraints on the linear decoder of the VAE by applying L1 regularization on terms for which prior knowledge is not available [Seninge et al., 2021]. I also implemented other regularization

methods such as GelNet [Sokolov et al., 2016] in VEGA to incorporate information about gene-gene interaction networks. [Lotfollahi et al., 2022b] proposed to solve the issue of potentially correlated latent variables (because of potential redundancy in gene sets) by deactivating irrelevant terms using an L2 or L1 regularization term (analogous to an ARD prior in the case of L2 [Neal, 1996], or group LASSO [Yuan and Lin, 2007] in the case of L1). Further work done on linearly decoded VAEs implemented a closed-form solution to the calculation of Bayes Factors in the differential activity test [Lotfollahi et al., 2022b]. This implementation also combined ideas of architecture surgery [Lotfollahi et al., 2022a] and an Hilbert-Schmidt independence criterion loss in order to integrate new dataset to a pre-trained model and discover novel biological programs specific to the newly integrated dataset [Lotfollahi et al., 2022b]. On the other hand, the pmVAE approach does not use a linear decoder, but models each pathway as its own VAE module using annotated genes as input/output variables, and combines the individual losses into a global loss [Gut et al., 2021]. More recently, the PAUSE approach implemented recent advances *post-hoc* analyses of deep learning models using pathway and gene attributions to build an interpretable deep generative model [Janizek et al., 2022]. This active development and improvement of interpretable deep generative models demonstrate that it is a popular area of research in the single-cell field, and is likely to remain one in the near future.

3.4.2 Modelling transcription factor activities in deep generative models for *in-silico* perturbation experiments

3.4.2.1 Gene regulatory mechanics

Most cellular functions are carried through the controlled expression of certain genes acting together to achieve said function. In order to control when a certain function is carried, whether it is as a function of time (eg. cell cycle, circadian rythm) or as a response to an external stimuli (eg. stress response), the expression of most gene products is regulated. While this can be done at different stages of the gene expression process (chromatin accessibility, RNA stability, translation, post-translational modifications...), one of the main regulatory mechanism is the control of the transcription level by molecules called transcription factors (TFs). TFs can bind to specific DNA motifs in order to activate or repress the transcription of certain genes into RNA molecules. This modulation of transcription results in potentially large-scale changes in the cell, affecting its state or even its fate. TFs play an important role in the development of diseases [Lee and Young, 2013], and as such are primary targets for new therapeutics. As they play a crucial role in cell differentiation into committed cell types, they are also targeted in stem cell differentiation protocols [Oh and Jang, 2019]. Modulating the TF activity is therefore of main interest in potentially treating certain diseases or improving stem cell differentiation protocols. However, the combinatorial nature of TF modulating experiments in order to find which combination of activations/repressions achieve a cellular phenotype of interest makes such experiments time-consuming and expensive. It is therefore highly desirable to provide a model achieving high performance

when predicting how the modulation of a set of TFs will affect the transcriptome of a cell. This would provide an *in-silico* experimental platform for researchers to prioritize combinations of TFs to perturbed in order to achieve a cellular phenotype of interest.

3.4.2.2 Incorporating transcription factor information in deep generative models

I showed in the previous sections of this chapter that it is possible to infer transcription factor activities from their targets in VEGA. However, the linear decoder of VEGA reduces the generative capacity of the model and it's ability to model complex regulatory behaviours between genes. A potential strategy would be to directly model transcription factor activities in the latent space by using available information about transcription factor expression directly. Such model would enforce interpretability in the latent space by using a prior on transcription factor expression for each latent variable, rather than constraining the architecture in the way VEGA or other linearly decoded VAE do.

The model could be formalized this way. Let $\mathbf{X} \in \mathbb{R}^{N \times G}$ be a single-cell experiment count matrix with N cells and G genes. Let $\mathcal{T} = \{\mathrm{TF}_1, \ldots, \mathrm{TF}_k\}$ represent the subset of k genes known *a-priori* to be transcription factors. We describe the generative process from which each observed expression vector x_n is drawn:

$$z_n \sim \log \mathrm{normal}\left(\tau_n, \tau_\sigma^2\right)$$

$$x_n = f_w\left(z_n, s_n\right)$$

s_n represents some batch ID for cell n and f_w is a neural network. The main modification from a standard VAE is that z_n is a k-dimensional log-normal distribution representing the activity of TFs for cell n, with a prior mean vector τ_n with each entry representing the corresponding observed (empirical) TF log-normalized expression (directly computed from the count matrix), and the prior covariance is a diagonal matrix with $\tau_\sigma^{2(i)}$ being the i-th entry computed as the observed TF-specific variance (variance of TF i log-normalized expression over all cells). Explicitly:

$$\tau_n^{(i)} = \log(\mathrm{TF}^{(i)} + 1)$$
$$\tau_\sigma^{2(i)} = \frac{\sum_n^N \tau_n^{(i)} - \bar{\tau^{(i)}}}{N-1}$$

Intuitively, we are using observed TF expression as a prior for the latent TF activity, which is used by the generative model to produce expression values. I note that the prior covariance matrix can be computed over the whole dataset, per batch (reflecting batch-specific variance of each TF) or per cell type label if available (reflecting cell type-specific variance of each TF). The goal is for the generative model to learn the regulatory mechanics which map transcription factor activities to an observed transcriptome. The TF-VAE model is trained using mini-batch optimization: the variational posterior distribution of the TF-VAE is mean-field and the ELBO is :

$$ELBO(x,s) = \mathbb{E}_{q(z|x,s)} \log p\left(x|z,s\right) - \mathrm{KL}\left(q(z|x,s) \parallel p(z)\right)$$

3.4.2.3 Perturbing cells *in-silico* by affecting transcription factor activities

If such generative model is successful in producing faithful transcriptomes using prior information about TFs expression, arithmetics operations on the latent space are possible in order to perturb cells by pushing the activity of a given set of TFs in a direction or another (down-regulation or up-regulation). Latent vector arithmetics have been shown to work in a classic VAE setting with models such as scGen [Lotfollahi et al., 2019], but the lack of interpretability as to what each latent variable represents has limited applications for *in-silico* screens of possible perturbations. A VAE incorporating information about TFs into latent variables would enable such experiments.

I obtained preliminary results suggesting that this task is possible using scRNA-Seq data. Dr. Hongxu Ding and I collaborated with the Oro lab at the Stanford University on screening potential combinations of TF perturbations to improve stem cell differentiation protocols in the context of esophageal basal cell differentiation. The preliminary results (Fig.3.1) suggest that the perturbation strategies put forward by the model are successful in improving stem cell differentiation towards a target cell fate, upon optimization of TF modulator concentrations. Further biological and computational validation is underway and will be crucial to validate that such models are successful to act as *in-silico*

perturbation screening platforms to prioritize experiments in the lab.

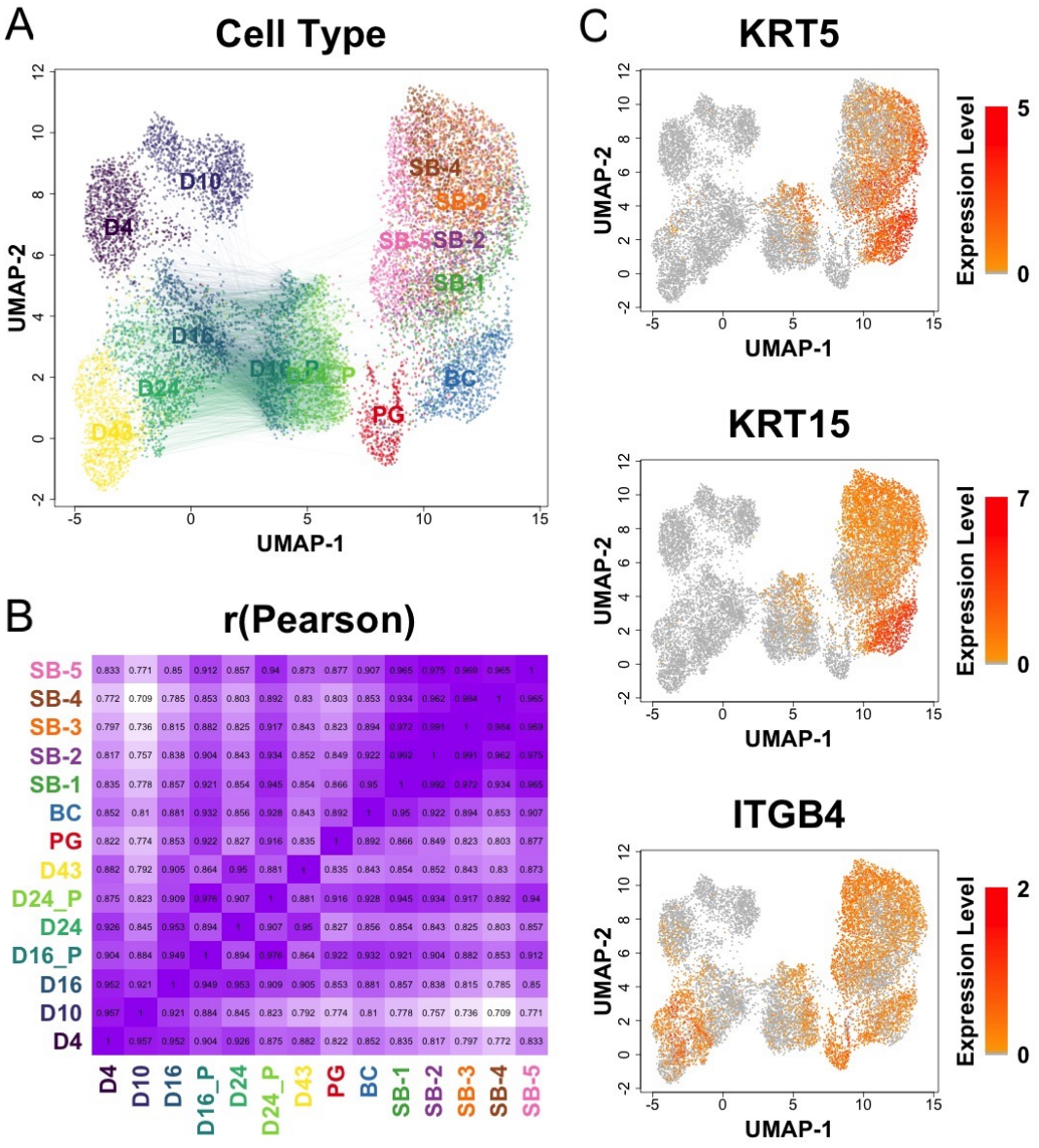

Figure 3.1: **Preliminary results of *in-silico* perturbation strategies to produce basal cells from stem cell cultures** (A) UMAP of stem cells (D4-43), perturbed stem cells (D16_P, D24_P) and *in-vivo* basal cells (PG,BC, SB1-5). (B) Heatmap of pairwise Pearson R from cell group expression centroids. (C) Expression of known basal cell marker genes. Perturbed stem cells display expression of those 3 marker genes.

3.4.2.4 Embedding more complex regulatory logic in deep generative models

The model discussed in the previous section is quite simple in its design, and rely on the assumption that a neural network can learn the regulatory mechanics mapping transcription factor activities to an expression profile. However, it does not address key concepts of gene regulation. First, similarly to most VAE implementations, the variational distribution is chosen to have a diagonal covariance, assuming independence among the latent variables representing transcription factor activities. This assumption is too simplistic, as TFs can be assembled in TF regulatory networks breaking the independence assumption. As one example, hematopoietic differentiation is driven by the interplay of several TFs [Goode et al., 2016] In addition, it can be important to model other key members of GRNs, such as kinases or transcription co-factors. However, this requires knowledge about the GRN structure.

In the past, Probabilistic Graphical Models (PGMs) and more notably Bayesian Networks (BNs) have been used to infer cellular networks from expression data [Friedman, 2004]. Such models are attractive because the graph stucture can be interpreted to determine causal relationships (for example regulation) between genes. But learning regulatory networks from expression data is challenging as it is an NP-hard problem [Friedman, 2004]. Using knowledge about which genes are likely to be regulators can simplify the learning procedure to infer GRNs [Pe'er et al., 2002]. Similarly, genes can be grouped *a priori* into modules of co-regulated genes (*eg.* in pathways) to facilitate the inference task of learning a module network [Segal et al., 2003]. Once the network structure and the

conditional probability distribution has been learned, BNs offer the possibility to simulate interventions using the "Do calculus" framework [Pearl, 1995]. This enables us to interrogate the effect of the knock-down of a set of regulators on the transcriptome of a cell.

Similarly, dynamical models such as ordinary differential equation (ODE) [Gardner et al., 2003] models can be used together with experiments to infer GRNs and the effect of perturbation on cellular signaling. Boolean network models can be used to encode regulatory logic in a *a priori* fashion between a few key regulators, as well as simulating perturbation *in-silico* to produce knock-out phenotypes [Krumsiek et al., 2011]. However, those models are limited by the number of genes that can be modeled, as well as requiring some prior knowledge about the GRN structure.

The combination of well-studied mechanistic models such as BNs and ODE dynamic models with novel efficient parameter optimization engines such as those used in deep learning offers interesting tracks to improve the ability to perform *in-silico* perturbation experiments. For example, module BNs with VAEs have been combined to model clinical data and simulate counterfactual scenarios in virtual patients [Gootjes-Dreesbach et al., 2020]. A similar approach could be envisioned for single-cell data, where a BN is learned simultaneously to training a VAE model, allowing to simulate perturbations either at the module level (if the BN is learned as a module network) or at the regulator level (if the BN is learned by restricting the set of potential regulators using litterature). Such model would present the advantage to incorporate mechanistic information (the BN

structure, as well as potentially enforcing prior knowledge on the BN structure during inference) while retaining a powerful generative capacity with the VAE. On the other hand, recent work on simulating perturbation though dynamical systems coupled to efficient optimization through deep learning techniques has found great success. This has been done either by directly modelling the dynamics of the system though molecular interaction terms between genes and perturbators like in CellBox [Yuan et al., 2021], or by modelling cellular differentiation has a diffusion process via stochastic differential equations and inferring a potential function like in PRESCIENT [Yeo et al., 2021] These approaches showcase the strength of dynamical models as powerful generative models that are likely to become even more popular in the future.

In summary, I envision that deep generative models integrating mechanistic information and prior biological knowledge will become powerful tools to model biological processes and provide opportunities for *in-silico* experiments at the single-cell level. I believe that combining the high-throughput of single-cell experiments, the existing knowledge about gene regulation and the recent advances in generative models will greatly accelerate biological discoveries.

Chapter IV: Collaborative work

4.1 Improving drug response prediction in patients using cell line and drug structure information

During my PhD I also collaborated with Ioannis Anastopoulos on improving drug response prediction in cancer patients. Large efforts have been made to screen drugs on cancer cell lines and gather RNA-seq data to determine the transcriptional profile of cancer types responding to certain compounds, such as the CCLE database [Ghandi et al., 2019]. However, the environment in which cell lines are grown is vastly different from the environment of primary tumors. Similarly, the mutational landscape of those cell lines differs from primary tumors. Therefore, translating sensitivity of cell lines to similar tumor types in patients is a non-trivial task. Sequencing of tumors in large consortia such as The Cancer Genome Atlas (TCGA) have shed light on the expression profiles of a large number of tumor types. However, drug response data for such large cohort of patients is not yet readily available. There is an unmet need for computational methods able to translate cell line sensitivity information to real patients. To that end, Dr. Anastopoulos lead work to design a novel deep learning architecture called the PACE framework (Patient Adapted with Chemical Embedding) leveraging from 2 recent fields of machine learning: (1) graph embedding with Graph Convolutional Networks (GCNs) [Kipf and Welling, 2017] and (2) domain adaptation [Farahani et al., 2020].

The former aims at embedding drug structure information as well as atomic features in a data-driven way to aid in the drug response prediction tasks. The latter aims at bridging the inherent differences between cell line transcriptional profiles and primary tumors, using Maximum Mean Discrepancy distance (MMD) to align the information from the CCLE expression neural network and the TCGA expression neural network. I contributed to designing the domain adaptation strategy and analyzing the results to improve the model's performance. The full work is described in the following manuscript, which has been submitted on *BioRxiv* and to the *International Journal of Environmental Research and Public Health*.

Patient Informed Domain Adaptation Improves Clinical Drug Response Prediction

Ioannis Anastopoulos[1,2], Lucas Seninge[1,2], Hongxu Ding[1,2], Joshua Stuart[1,2]

[1]Department of Biomolecular Engineering, UC Santa Cruz, Santa Cruz, California, USA.
[2]UC Santa Cruz Genomics Institute, Santa Cruz, California, USA.
[*]Correspondence should be addressed to I.A. (ianastop@ucsc.edu), or J.S. (jstuart@ucsc.edu).

ABSTRACT

In-silico modeling of patient clinical drug response (CDR) promises to revolutionize personalized cancer treatment. State-of-the-art CDR predictions are usually based on cancer cell line drug perturbation profiles. However, prediction performance is limited due to the inherent differences between cancer cell lines and primary tumors. In addition, current computational models generally do not leverage both chemical information of a drug and a gene expression profile of a patient during training, which could boost prediction performance. Here we develop a Patient Adapted with Chemical Embedding (PACE) dual convergence deep learning framework that a) integrates gene expression along with drug chemical structures, and b) is adapted in an unsupervised fashion by primary tumor gene expression. We show that PACE achieves better discrimination between sensitive and resistant patients compared to the state-of-the-art linear regularized method (9/12 VS 3/12 drugs with available clinical outcomes) and alternative methods.

GLOSSARY: ***GCN***, Graph Convolutional Network. ***MorganFP***, Morgan Fingerprint. ***SMILES***, Simplified Molecular Input Line Entry System for annotating chemical structures using character strings. ***ML/DL***, machine learning/deep learning, ***CDR***, Clinician Drug Response. ***CDI*** Cell-line-Drug-IC_{50}. ***EM*** Expression Module. ***DM*** Drug Module. ***PM*** Prediction Module. ***OOD*** Out of Distribution. ***CL*** Cell Line.

INTRODUCTION

INTRODUCTION

Precision medicine promises to revolutionize cancer treatment by improving clinical drug response (CDR) prediction. CDR prediction could be greatly facilitated by cutting-edge high-throughput sequencing technologies, which provide comprehensive and individualized omics profiles. Based on these omics profiles, several CDR prediction approaches have been proposed. For instance, Tissue-Guided Lasso TG-LASSO[1] integrates tissue-of-origin information with gene expression profiles for CDR prediction. DeepDR[2], on the other hand, predicts CDR from mutation and expression profiles.

However, as another crucial component for CDR prediction, the chemical properties of drugs have been under-utilized. Although the traditional Morgan Fingerprint molecular representation[3] has been used to integrate drug chemical information for CDR prediction, it cannot adaptively learn alternative representations of drug chemical properties as it is a static representation of the molecule and does not dynamically extract features for the desired prediction task. For example, CDRscan, similarly to TG-LASSO, does not take advantage of the

diverse patient RNA-Seq profiles that are published on The Cancer Tumor Atlas (TCGA), and is not evaluated to address if the model can be applied for drug response prediction to patients, which is what such a model would be used for in practice. In addition, CDRscan uses Morgan Fingerprints to represent key molecular substructures using an explicitly defined featurization. A limitation of this specific methodology is its inability to adaptively learn alternative representations that may be beneficial to the particular task in hand[4]. DrugCell also uses Morgan Fingerprints to represent drugs along with an Visible Neural Network (VNN) embedded in Gene Ontology (GO) terms, which provides interpretable results[5], but it is also not evaluated on patients.

Graph Convolutional Network (GCN)[6] representations emerge as a powerful alternative for encoding drug chemical properties. GCN adaptively learns chemical information by generalizing the convolution operation from a grid of pixels to a graph, where each node can have a variable number of neighbors. GCNs have been used to explore drug-target interactions and side effect predictions - the two most important factors for developing a new drug. For instance, Decagon uses GCN to predict potential side effects of a drug [7]. Such methodological advances provide novel insights in incorporating drug chemical properties during CDR prediction.

The majority of CDR prediction algorithms are trained with cancer cell line (CL) drug preturbation profiles. CLs have long served as models to study molecular mechanisms of cancer, because they maintain valuable molecular information of the primary tumor from which they were derived. CLs offer the advantages of being easily grown, relatively inexpensive, and amenable to high-throughput assays. Data generated from CLs can then be used to link cellular drug response to molecular features, where the ultimate goal is to build predictive signatures of patient outcomes [8]. Various models have been developed to predict patient CDR from the molecular profiles of CLs [9–11]. However, these models only show limited success in certain drugs [12 13]. Therefore, developing a model based on CL molecular features to predict CDR in patients for most drugs remains challenging [14]. One major difficulty for such cross-domain CDR prediction is the prominent differences between cell lines and primary tumors [15–19]. Recent advances in domain adaptation aim at aligning domains to tackle domain alignment problems, such as batch effect correction to reconcile differences across laboratories and studies [20]. Mean Maximum Discrepancy (MMD) [21] has shown promising results in aligning domains in an unsupervised manner [22]. Such a technique could be used to align CL and patient tumors in developing drug response models that are more clinically focused.

Inspired by the advanced GCN-based drug chemical information encoding, as well as the MMD-based domain adaptation, we develop a drug response predictor using Patient Adaptation and Chemical Embedding (PACE). This deep learning framework uses a GCN to dynamically learn chemical information of each compound, and is adapted to implicitly align the CL expression representation with that of a patient sample of the same tissue of origin. Thus, the model does not assume that CL and patient samples are drawn from the same distribution. We achieve this by using Mean Maximum Discrepancy (MMD) to align the latent spaces produced by CL expression and patient expression vectors. Such a technique has been successfully applied in previous studies [23–25] in transferring generalizable features across domains [26]. We trained our model on 142,351 of CL-drug-IC_{50} (CDI) pairs where each CL vector was paired with a random patient expression vector of the same tissue of origin. We used a Graph Convolutional

Network (GCN) to encode drug information and pair it with the patient adapted expression information in order to predict IC_{50}. To evaluate our model, we collected a curated CDR dataset with patient outcome information on response to 12 drugs in total. Our model achieved superior performance (9/12 drugs) compared to alternative PACE models and the state-of-the-art linear method (3/12 drugs) in significantly discriminating between sensitive and resistant patients.

RESULTS

Overview of PACE and evaluation strategy

The PACE deep learning framework consists of three modules: an Expression Module (EM), Drug Module (DM), and Prediction Module (PM). As shown in Figure 1A, the EM is composed of fully connected layers and learns highly informative features from gene expression vectors. The DM is composed of a GCN and learns highly informative features for each atom from the graph representation of chemical compounds. Atom-level features are then aggregated to represent information about the compound as a whole (see METHODS). Given the success GCNs have had in computational chemistry and biology applications [27–29], we posited that the DM could learn a general graph embedding that would extend to drugs unseen during training. The PM is composed of a fully connected layer and takes the information learned from the EM and DM as input to predict $\log(IC_{50})$. We included gene expression information, the drug used, and the associated IC_{50} value from the Genomics of Drug Sensitivity in Cancer (GDSC) project [30]. The model was trained with "CDI tuples" -- Cell line, Drug SMILES, IC_{50} -- indicating which drug was applied to a particular cell line and what that cell line's response was to the drug.

Our goal is to extrapolate drug response from cell lines to patients. Hence, the model needs to generate an out-of-distribution (OOD) embedding space (for patient samples) representing a distribution not present in the training data (of cell lines). Inspired by [23], and recent advances in the field of domain adaptation[26], we used maximum mean discrepancy (MMD) to adapt the latent distribution produced by the EM so that cell line gene expression is aligned to patient gene expression. Each CL was paired with a TCGA tumor sample's gene expression vector of the same tissue of origin (see Supplemental Table 1/2), which has been shown to play a key role in a tumor's treatment and progression. Restricting cell lines to match the tissue of origin of primary tumors resulted in 531 cell lines treated across 310 drugs, amounting for 142,351 CDI pairs. Cell lines and tumors that did not have a matching tissue of origin were not included in training. Each cell line was paired with a random primary tumor sample of the same tissue with the goal of creating a general enough adaptation of EM's latent space.

To test the efficacy of adapting the EM with patient gene expression via MMD, we constructed a non-adapted version of PACE for comparison purposes. In addition, we also compared our model to one in which the DM uses Morgan Fingerprints (MorganFP), representing a more conventional molecular encoding. Altogether, we created three alternative models closely related to PACE -- PACE-Morgan, noPACE, and noPACE-Morgan (see Table 1). Alternative models with an adapted EM should provide a poorer fit to the cell line data, yielding poorer performance in cell lines, compared to non-adapted models because the adapted

models attempt to fit the distribution of both cell lines and patients. On the other hand, the adapted models should perform better in the patient setting.

Table 1. All alternative models used and their specifications

EM	DM	Name
Adapted	GCN	PACE
Adapted	MorganFP	PACE-Morgan
Not adapted	GCN	noPACE
Not adapted	MorganFP	noPACE-Morgan

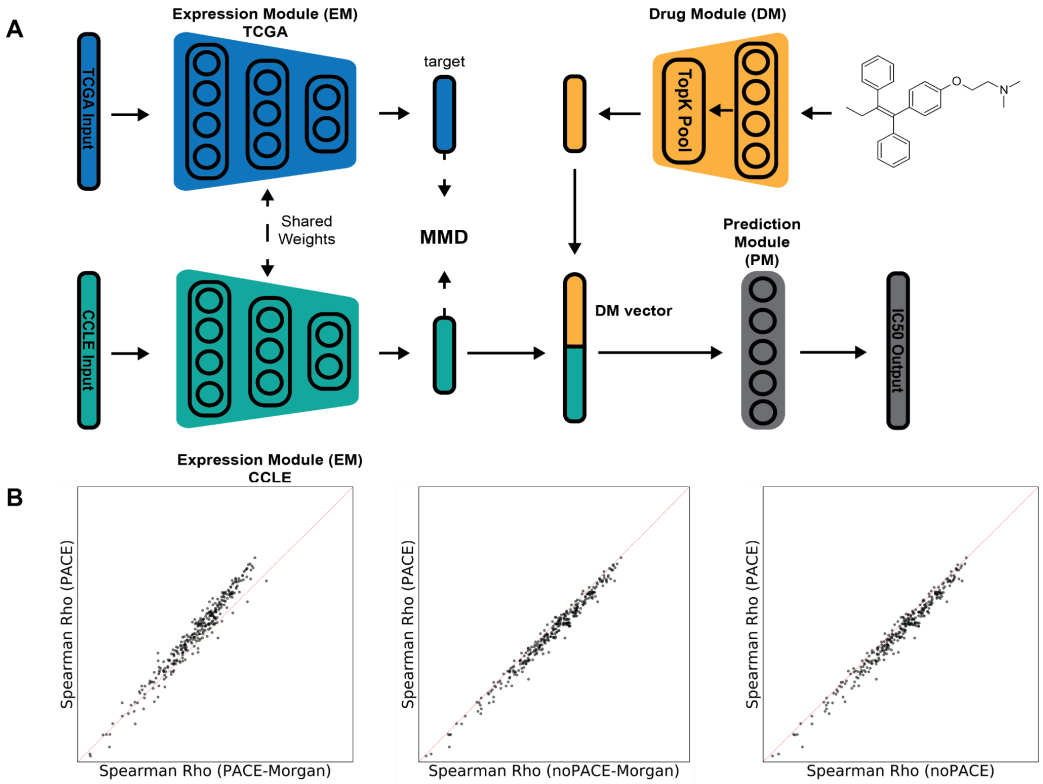

Figure 1. Deep learning model architecture. (A) Graphic overview of the proposed deep learning framework. Expression Module (EM) extracts highly informative features for the input expression vectors for both CCLE and TCGA via shared weights. These two compact expression representations are compared with each other via Mean Maximum Discrepancy (MMD) to diminish the distance between them, thereby aligning the two representations. The Drug Module (DM) encodes the molecule and pools the most informative nodes (atoms) to also create a highly informative compact representation. Finally, the CCLE expression representation and the drug representation are

concatenated together and passed to the Prediction Module (PM) that makes the final log(IC_{50}) prediction for each CDI pair. (B) Per drug predictive performance, which is quantified by theSpearman Rho between actual and predicted log(IC_{50}) across all perturbed cell lines. GCN-MMD is compared against MORGAN-MMD (left), GCN-NoMMD (middle), MORGAN-NoMMD (right). Each point represents a drug. Points above the diagonal represent better performance by GCN-MMD.

Combining domain adaptation and graph encoding preserves performance in cell lines while increasing the accuracy of predicting patient response for more drugs.

We compared the Spearman Rho achieved by our proposed PACE model to all the other variations for cell lines treated by each drug in our dataset (Figure 1B). We found that although PACE does better compared to PACE-Morgan for the majority of the drugs (points above the diagonal), the non adapted versions (noPACE and noPACE-Morgan) achieve a higher correlation (points below the diagonal). Nevertheless, when comparing across all 142,351 CDI pairs, the proposed model and the alternatives achieved comparable results (Supplementary Fig. 1). This suggests that MMD adaptation preserves the prediction performance of drug response in CL while yielding superior discrimination performance in the patient setting (shown next).

In addition to the models described in the previous section, we compared PACE to the state-of-the-art TG-LASSO[1] model, which is a linear regularized method to predict *clinical drug response* (CDR). To evaluate all of the models in the patient setting, we followed the same evaluation presented in the TG-LASSO study[1]. We used the same curated CDR dataset consisting of 531 patients treated across 24 drugs labeled with the type of response indicated for each patient. The majority of patients in this dataset (70%) were treated with a single drug, while the rest were given two or more. Patients with stable disease and clinical progressive disease were labeled as resistant (R), whereas those with partial or complete response were labeled as sensitive (S). After following the same filtering steps and retaining samples for which we had expression information, 506 patients across 12 drugs remained. To measure the performance of the methods, we asked if their predicted log(IC_{50}) drug response (a continuous measure) correlated to the drug response labels (R/S) (a categorical measure) in the CDR dataset (see METHODS). Specifically, a one sided Mann-Whitney U test was used to determine whether the predicted log(IC_{50}) for the true resistant (R) patients is significantly larger than that of true sensitive (S) patients.

As shown in Figure 2A, MMD adaptation produces an embedding that can discriminate between resistant and sensitive patients across more drugs compared to all other models that lack such adaptation. The combination of patient information adaptation with MMD and GCN for drug embedding had better correlation to patient response than all the alternative methods examined (Figure 2A, Supplementary Fig 2). Specifically, PACE showed significant discrimination between resistant and sensitive patients ($p<0.05$) for nine out of the twelve drugs compared to six by noPACE (Figure 2A). Similarly, PACE-Morgan predicted six drugs , compared to five by noPACE-Morgan (Supplementary Fig. 2). TG-LASSO was the worst performing method with three drugs predicted significantly.

We also observed that regardless of the method used, cisplatin, etoposide, and paclitaxel were predicted correctly. In contrast, bicalutamide, bleomycin, and docetaxel were not predicted correctly by any of the methods. We further observe that CDR prediction using MMD

adaptation improved CDR prediction for cisplatin, doxorubicin, gemcitabine, paclitaxel, tamoxifen, temozolomide, and vinorelbine suggesting that tissue of origin may play a crucial role for these drugs. However, for gemcitabine, tamoxifen and vinorelbine, using the MorganFP drug encoding did not achieve significant CDR prediction in patients, even with MMD adaptation, further suggesting that the appropriate drug embedding is needed for such a task (Supplementary Fig. 2).

Lastly, we found that for the hard to predict bicalutamide PACE and PACE-Morgan produced predicted IC_{50} values with the correct direction (predicted IC_{50} for resistant patients should be higher than that of sensitive patients) (Figure 2B, Supplementary Figure 3A). The non-adapted variants of PACE (noPACE, noPACE-Morgan) produced predicted IC_{50} values with the incorrect direction (Supplementary Figure 3B/C). This is also evident by the direction of the difference between the median of sensitive predicted IC_{50} and sensitive predicted IC_{50}, which we term as ΔIC_{50} (Supplementary Table 3). The ΔIC_{50} for bleomycin was also observed to be the most negative in the PACE models compared to the noPACE models. PACE-Morgan produced the most negative ΔIC_{50} for docetaxel compared to the noPACE models, however for this hard to predict drug PACE produced a ΔIC_{50} in the wrong direction.

Taken together, these results suggest that the combination of patient adaptation via MMD and a combination of the chemical embedding learned from GCN produced a highly informative model that can be extended to the patient setting.

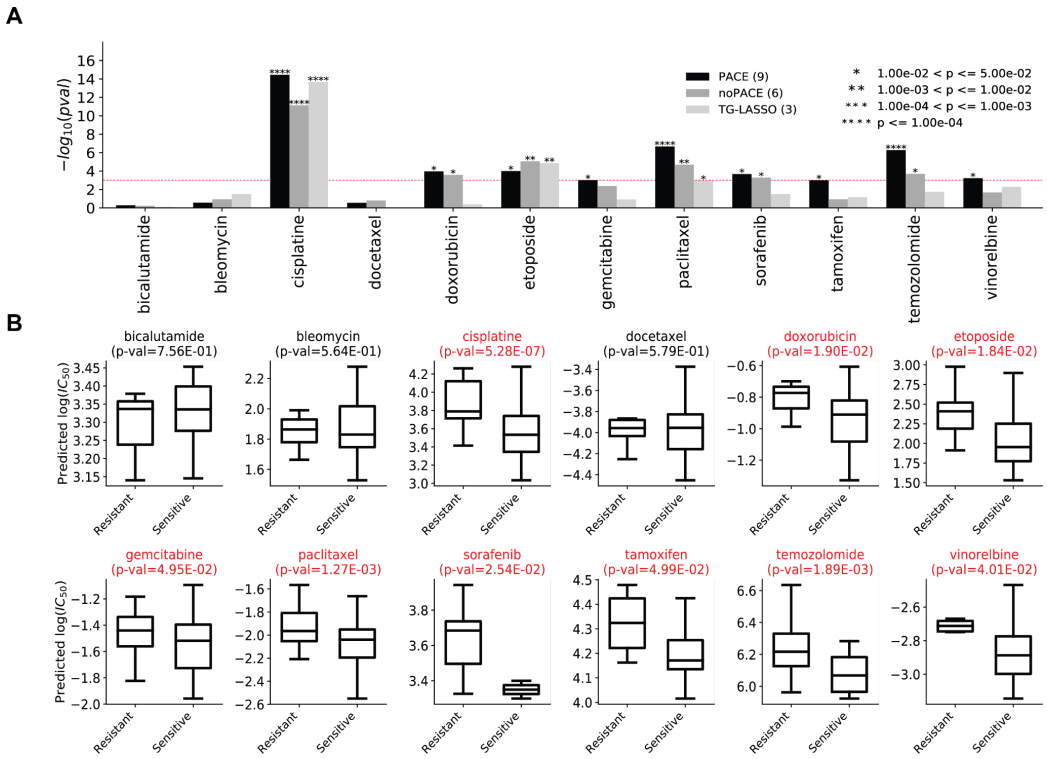

Figure 2. TCGA CDR performance comparison with framework variants and with TG-LASSO. We compare our model with the unregularized alternative and with the state-of-the-art TG-LASSO linear method. GCN-MMD yields the best resistant-sensitive discrimination performance in the TCGA CDR dataset (A) Bar plots showcasing p-value, corresponding to the one-sided Mann-Whitney U test, determined by averaging 10 independent predictions made by each model. (B) Box plots reflecting the distribution of estimated log(IC_{50}) values using GCN-MMD for resistant or sensitive patients. The p-values here also correspond to a one-sided Mann-Whitney U test.

Cell line diversity is more important than drug diversity for patient CDR prediction.

Next, we asked if gene expression information or drug information has a bigger impact in predicting drug response in patients. To this end, we created two different drop out experiments -- one where all the CDI pairs for a *cell line* were withheld and another in which all the CDI pairs for a *drug* were withheld. For the cell line dropout experiment, we created training sets with 20%, 40%, 60%, and 80% of the total cell lines (531). For the drug dropout experiment, we measured the performance of a method in predicting the twelve CDR drugs having never seen those same drugs during training. To this end, we removed the 12 drugs present in the CDR dataset and then created training sets in cell lines that included 20%, 40%, 60%, and 80% of the remaining 298 drugs. For each of the training sets, the model was trained ten independent times on each fold. The ten independent cell line-trained models were then applied to the patient CDR dataset, and the average predicted log(IC_{50}) was computed. The Mann-Whitney U test was used to evaluate the discrimination between labeled resistant and sensitive patients (see METHODS).

As summarized in Figure 3A, lack of gene expression information had a bigger impact compared to lack of drug information across all drugs in our CDR dataset. This is likely caused by the vast difference in complexity and variance between the gene expression profiles and the compound structures. The robustness (measured by the variance of the p-value across 10 fold cross validation) of the model suffers more with 20% of the CLs included in training compared to the same percentage of drugs included in training (Supplementary Figure 4A/B). Addition of more CLs in the training set drastically improves robustness of the model as shown by the decreasing variance of the p-value across all 10 folds, indicative of the crucial role expression information plays in predicting drug response (Supplementary Figure 4A). This result also suggests that the GCN needs a small amount of graph examples in the training set to be able to generalize well to new graphs not seen in the training set. Additional graph examples improved the robustness of the prediction of all but five drugs: bicalutamide, bleomycin, gemcitabine, sorafenib, and tamoxifen in the CDR dataset (Supplementary Figure 4B) with the variance in p-value decreasing slightly. We conducted an additional experiment, where only the 12 CDR drugs were removed from training and reported minimal reduction in CDR prediction performance (Figure 3B). It is crucial to note that the small improvements in CDR prediction from additional training molecules can be explained by the fact that most of the CL have been treated by most of the drugs (median number of CL = 498.5). This means that most of the variability in gene expression has been seen by the model even when 20% of the drugs are included in training (Supplementary Figure 4C), which leads to robust results in the CDR dataset and is consistent with what we observed in the CL dropout experiment.

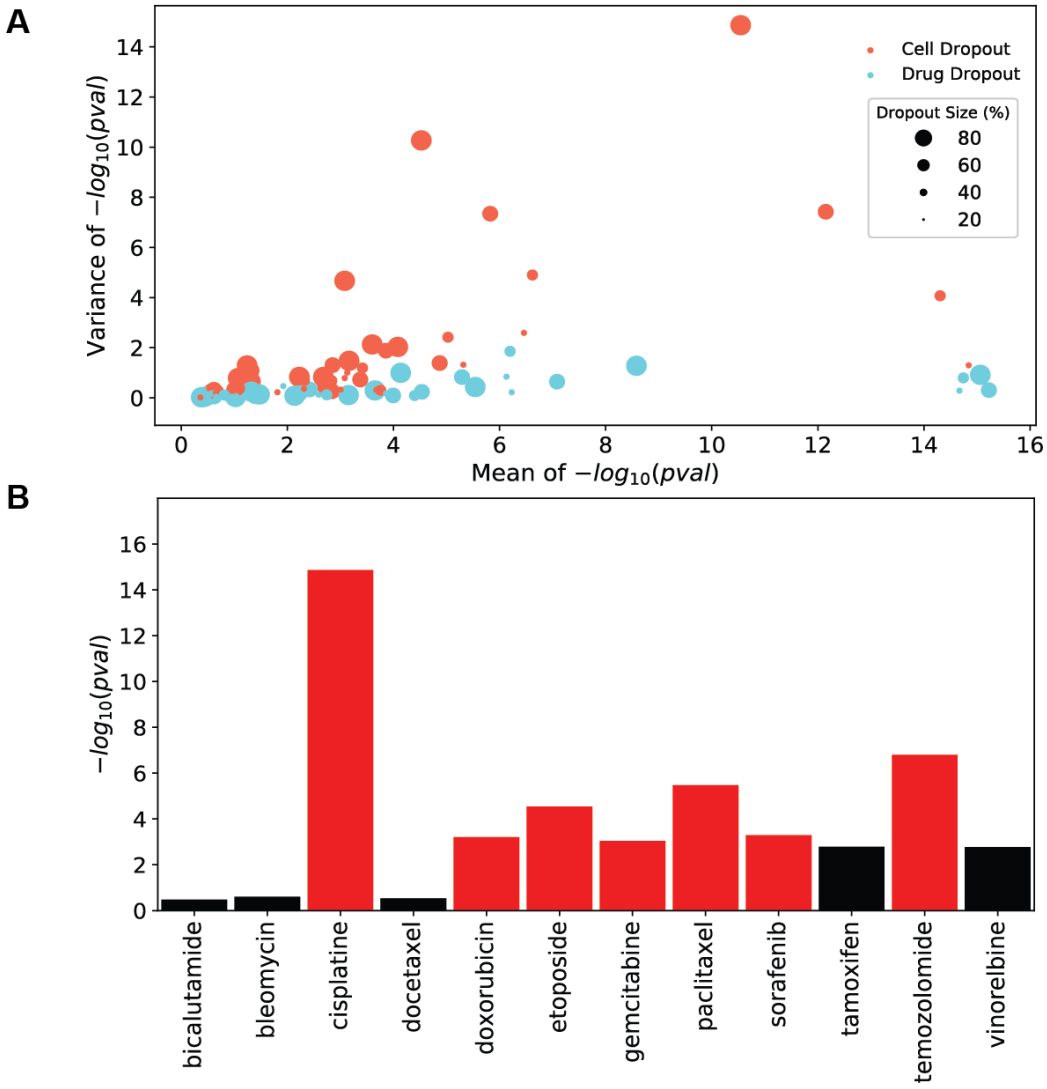

Figure 3. Drug/Cell Line 10 fold cross validation dropout experiment. Dropping data, drug-wise and CL-wise to test the limits of the model's OOD inference ability in a 10-fold cross validation fashion. For each fold the training was repeated 10 independent times. (A) Showing the variance of the -log(p-val) determined by the Mann-Whitney U test for the difference between the predicted log(IC_{50}) between resistant and sensitive patients across 4 conditions: 20% of CL retained in training, 40% of CL retained in training, 60% of CL retained in training, and 80% of CL retained in training. (B) Dropping only the 12 drugs in the CDR dataset. Variance results are displayed for all 12 drugs in the CDR dataset.

Top Predictions Recapitulate Knowledge on Targeted Therapy

Most of the drugs with CDR in patients that were tested here can be classified as chemotherapy agents, with the exception of sorafenib (VEGFR inhibitor) and tamoxifen (*ESR1* inhibitor). To assess the performance on targeted agents for well characterized cohorts, we carried out an *in-silico* analysis on drugs with known biomarkers of response. We used mutations as biomarkers of response for the TCGA cohorts where expression and mutation information were available. For the cohorts where these were not available we used expression as a biomarker of response to confirm that the model learns biologically meaningful information. The idea here is that as a target gene's expression increases, the drug's predicted IC_{50} should decrease accordingly, indicating an increase in sensitivity.

We collected mutation information from TCGA breast cancer (BRCA), melanoma (SKCM) samples and LUSC/LUAD cohorts (combined and abbreviated as LUNG). We used mutation information as a biomarker of sensitivity. We tested trametinib, olaparib, dabrafenib and gefitinib on all of the aforementioned cohorts. Trametinib is a MEK inhibitor and used to treat SKCM. Olaparib is a PARP1 inhibitor and used to treat *BRCA1*- or *BRCA2*-mutated breast and ovarian (OV) cancers. Next, we examined the correlation between a drug's predicted $\log(IC_{50})$ and its target gene's expression (after Z-score transformation of the gene expression values). We specifically examined OV in this way due to the fact that we could not collect sufficient OV samples with predicted *BRCA1* or *BRCA2* mutations, and thus could not use *BRCA1/2* mutation as a biomarker for OV and olaparib.

As expected, *BRCA1* mutant samples were predicted to be more significantly sensitive to olaparib compared to *BRCA1* WT samples (Figure 4A), *NRAS* and *MAP2K1* SKCM mutants were predicted to be significantly more sensitive to trametinib compared to SKCM WT samples (Figure 4B/E). *NRAS* SKCM mutants were additionally predicted to be more sensitive to dabrafenib compared to *NRAS* WT samples (Figure 4C). Olaparib has been previously shown to be effective in *ATM* mutated BRCA patients. Although our model was not able to predict this association correctly, SKCM *ATM* mutated samples were predicted to be sensitive to olaparib (Figure 4D). Lastly, LUNG cancer *NRAS* mutated samples were predicted to be more sensitive to dabrafenib (Figure 4F).

Studies have pointed to *PARP* expression as a promising biomarker of olaparib response[31]. When we examined the correlation between *PARP1* z-score and the predicted $\log(IC_{50})$ per disease, OV had a significantly negative Spearman Rho (rho=-0.48) (Supplementary Figure 5A). Testicular cancer (TGCT) showed the strongest negative correlation between *PARP1* expression and predicted IC_{50} for olaparib, which has recently been in clinical trials in combination with chemotherapy for TGCT[32].

We further examined the predicted response of lapatinib and the correlation with the expression of its target genes, *EGFR* and *ERBB2* (gene expression first transformed . *In-vitro* studies have previously shown that lapatinib inhibits cell proliferation and migration of breast cancer cell lines expressing different levels of *EGFR* and *ERBB2*, and that cells overexpressing *ERBB2* were more sensitive[33]. Interestingly, our model predicted *EGFR* expression as a stronger biomarker (Supplementary Fig. 5B) in BRCA patients compared to *ERBB2* expression (Supplementary Fig. 5C). Taken together, these results suggest that our model can recapitulate the relationship of well characterized drugs with the appropriate biomarkers, and their applicability in equally well-characterized cohorts.

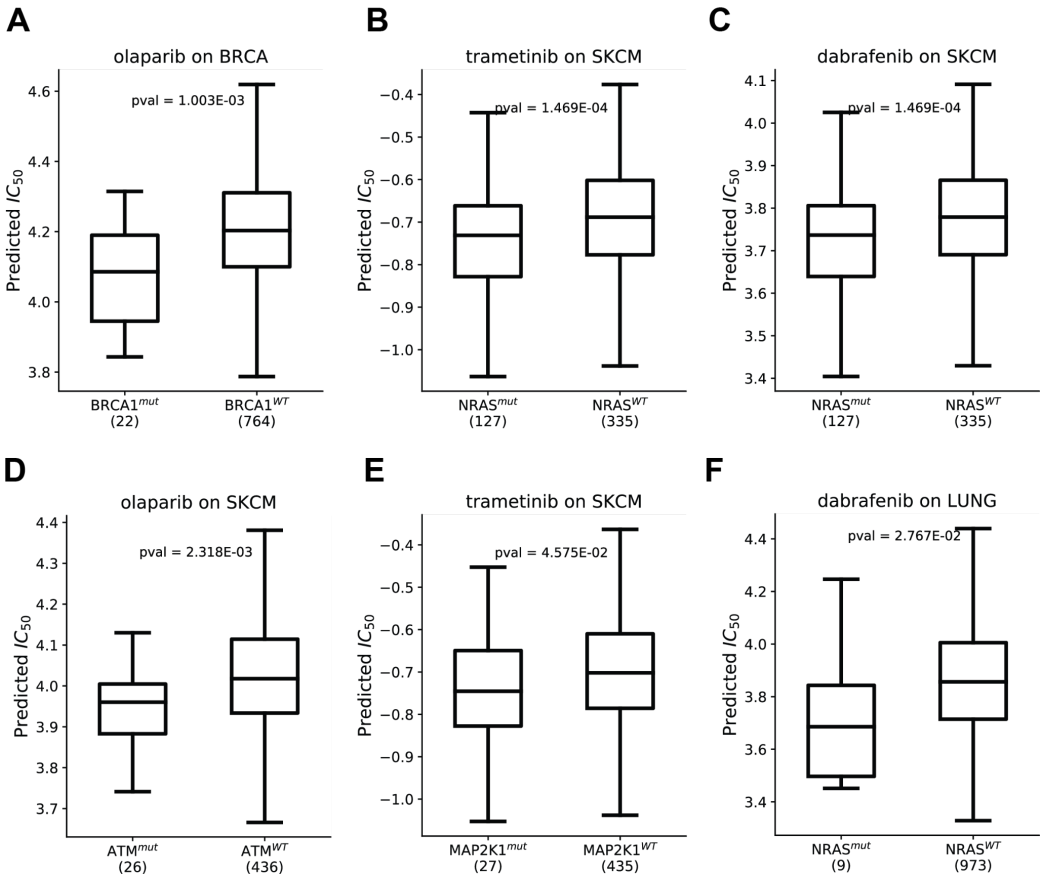

Figure 4. Functional Analysis. (A) Predicted log(IC50) for BRCA1 mutant and wild-type (WT) breast cancer (BRCA) samples in-silico treated with olaparib. P-value corresponds to the one-sided Mann Whitney U test discriminating between mutant and WT predicted log(IC50). (B) Predicted log(IC50) for NRAS mutant and wild-type (WT) melanoma (SKCM) samples in-silico treated with trametinib. (C) Predicted log(IC50) for NRAS mutant and wild-type (WT) melanoma (SKCM) samples in-silico treated with dabrafenib. (D) Predicted log(IC50) for ATM mutant and wild-type (WT) melanoma (SKCM) samples in-silico treated with olaparib. (E) Predicted log(IC50) for MAP2K1 mutant and wild-type (WT) melanoma (SKCM) samples in-silico treated with trametinib. (F) Predicted log(IC50) for NRAS mutant and wild-type (WT) LUSC/LUAD (LUNG) samples in-silico treated with dabrafenib.

DISCUSSION

In this study, we presented a new deep learning framework that uses both a graph convolutional network (GCN) as a general encoding for drug information together with patient information to aid in out-of-dataset prediction. During training, the method aligns cell-line and patient gene expression domains using implicit tissue-driven adaptation together with drug information to derive highly informative features for drug response prediction.

We showed that adapting tumor information with maximum mean discrepancy (MMD) preserves performance in cell lines while improving the prediction of clinical drug response (CDR) in patients regardless of the drug encoding used. We found that GCN's embedding extends to drugs that have not been seen in training. These results suggest that a combination of implicit tissue-driven adaptation and a highly flexible drug encoding lead to improved prediction performance in patient samples. Interestingly, we note that the drug dropout experiments revealed that only 20% (298) of drugs are needed to yield robust generalization performance. On the other hand, the cell line dropout experiments showed that a lack of cell line diversity during training greatly impacts generalization of drug response in patients. We further examined if our model can recapitulate some of the well known therapeutics for melanoma, breast cancer, and lung cancer. We found that the model was able to predict *MEK2* mutant melanomas as significantly more sensitive to trametinib, a MEK inhibitor, compared to the WT cohort. Similarly, *BRCA1* mutants in breast cancer were significantly more sensitive to olaparib, a first-line treatment to patients with such a mutation, compared to the WT cohort. #

Ideally, the type of models studied here should be trained on patient samples rather than surrogates such as cell lines. However, at this time, an adequate amount of patient data is lacking for any particular drug of interest as most patients receive the standard of care based on the tissue of origin. For example, we attribute the poor performance in predicting sensitivity to bicalutamide, bleomycin, and docetaxel to lack of adequate training data (Figure 2A). In the coming years, single cell sequencing should improve the performance of predictive models. For example, promising results have been published by the MIX-seq study in which the sequencing of cell lines before and after drug treatment has detailed the heterogeneity in response across individual cancer cells[36]. Together with single-cell sequencing, human-derived xenografts and 3D human organoids should complement cell line studies to add needed realism for classifier training; e.g. by including contributions from the microenvironment. #

The framework we presented can be extended to incorporate additional diverse biological data as it becomes available. As expected, we found that accuracy depended heavily on the presence of an appropriate cell line of a matching tissue type in the training data. Beyond extending the training data to cover more cell lines, which will increase the diversity of patients to which the method can be applied, other data types may also provide a boost in performance. For example, the current work focuses on gene expression and does not consider genomic alterations, such as mutations and structural variants, and the vulnerabilities that these may introduce. Genetic dependency data generated from the ACHILLES project [37] for example are now available for many of the same cell lines that our model was trained on. In theory, incorporating synthetic lethality prediction into the model should improve drug response prediction, as drugs that target synthetically lethal pairs should have a substantial impact on drug response. Incorporating protein level information could also lead to improvements in performance as many of the drugs target specific proteins whose expression may or may not be

correlated with the gene's RNA. The ongoing CPTAC project [38] is systematically quantifying protein levels and phosphorylation states in cancer patients from TCGA. In addition, it has been shown that proteome-level characterization of cell lines can aid in drug response prediction [39]. It is therefore evident that addition of proteomic data to our model could have a significant impact on the prediction of drug response.

Lastly, increasing the interpretability of our model would be of great value. It would be very informative to developers of new drugs if they could predict the pathways affected by administration of a new treatment. Recent advances in developing more interpretable biological models[40,41] should help models like ours in providing generalizable and interpretable results. Lastly, the GCN of our model uses only atom features for drug encoding. Other types of GCN, such as GINEConv from this study [42], are more expressive and use both atom and bond features, which could potentially create an even more generalizable drug embedding. We leave the exploration of the most appropriate GCN for this task and the inclusion of an interpretable EM to future studies.

METHODS

Overall Framework. Our model is an adapted dual convergence architecture that integrates gene expression information with drug structure aimed at generalizing clinical drug response (CDR) prediction in patients. It consists of three modules: Expression Module (EM), Drug Module (DM) and Prediction Module(PM). Highly informative representations of gene expression and drug structure are generated by the EM and DM, respectively. These representations are jointly passed to the PM where the log(IC_{50}) prediction is made. The model takes as input a cell line (CL) expression vector ($\boldsymbol{x_c}$), a primary tumor expression vector ($\boldsymbol{x_t}$), and the compound that was applied on the CL. The way the compound is presented as input to the model is explained in the ***Morgan Fingerprint (MorganFP) Representation of Drugs*** and ***Graph Representation of Drugs*** sections.

Expression Module (EM). The EM consists of 2 fully connected layers of 1024, and 100 nodes with Rectified Linear Unit (ReLU) activation. BatchNormalization and Dropout of 0.35 are applied on each layer. During training, the EM produces latent representations for both CL and primary tumors via weight sharing as follows:

$$f_{EM}(x_c;\ \theta_{EM}) = z_c, \text{ and}$$

$$f_{EM}(x_t;\ \theta_{EM}) = z_t,$$

where z_c and z_t represent the latent vectors of CL and primary tumor, respectively.

Inspired by the field of domain adaptation, and driven by the need to generalize drug response prediction to patients, we used a domain alignment method called Mean Maximum Discrepancy

(MMD) [15]. Specifically, the model tries to align z_c to z_t with the goal of making the cell line latent space more similar to the primary tumor latent space by minimizing the following loss:

$$L_{MMD}(z_c, z_t) = k(z_c, z_c) + k(z_t, z_t) - k(z_t, z_c) - k(z_c, z_t),$$

where k denotes the universal Gaussian kernel. Each z_c and z_t represent the same tissue-of-origin during training. Thereby, the model implicitly aligns CL and primary tumors in a tissue driven manner.

MorganFP Representation of Drugs. We used the python library RDKit to generate Simplified Molecular Input Line Entry System (SMILES) strings, which describe the structure of a molecule using a single line of text, and compute MorganFP for each molecule in our datasets [43]. SMILES strings are simple string annotations that describe the structure of the molecule. MorganFP is part of the Extended-Connectivity Fingerprints (ECFPs) family and are generated using the Morgan algorithm [44,45]. These fingerprints represent molecular structures and the presence of substructures by means of circular atom neighborhoods (bond radius). In this study we used radius 2 and constructed a 2048 long bit vector for each molecule. A radius of 2 takes into account neighbors up to two atoms away when constructing the bit vector (fingerprint) of the molecule.

Graph Representation of Drugs. We used RDKit to generate SMILES strings for each drug. Next, we represented the SMILES string for each compound $\{c_j\} \in C$ as a graph $G = \{V, X\}$, where $V = \{v_j\}$ represents the set of nodes (nodes here are atoms on the molecule). An adjacency matrix A represents the topological structure of each molecule with $A_{i,j} = 1$ denoting a bond between two atoms, otherwise $A_{i,j} = 0$. $x_i \in X$ indicates the vector of features for each atom v_i on the compound. The features (189 in total) used for each compound can be found in Table 2.

z

Table 2. Description of Atomic Features

Atom feature	Size	Description

Atom symbol	19	[As, B, Br, C, Cl, F, Hg, I, K, N, Na, O, P, Pt, S, Sb, Se, V, Zn] (one-hot)
Atomic Number	119	Atomic number of each atom (one-hot)
Chirality type	4	[UNSPECIFIED, R, S, OTHER]
Degree	11	Number of covalent bonds (one-hot)
Formal Charge	12	Electrical charge (one-hot)
Hydrogens	9	Number of connected hydrogens (one-hot)
Radical Electrons	5	Number of radical electrons (one-hot)
Hybridization	8	[UNSPECIFIED, sp, sp2, sp3, sp3d, sp3d2, OTHER] (one-hot)
Aromatic	2	Atom is an aromatic ring (one-hot)
Total	**189**	**Total number of features**

Drug Module (DM). The DM of the model aims at extracting highly informative features from each molecule. This is done via either the MorganFP representation of the molecule, or the graph representation of the molecule. For the former, the DM consists of one fully connected layer, ReLU, BatchNormalization and Dropout. For the latter, we used the python library PyTorch Geometric to produce data-driven molecular features using GCN [46]. In particular, we used the GCN architecture from [47]. That architecture learns substructures of a given graph, and relationships between graphs, which is crucial in this study as we aim to generate a general embedding space for structurally diverse molecules presented in the drug response dataset. This type of GCN falls under the spatial GCN category, which can generalize the learned embedding to heterogeneous graphs [48],. We used one layer, followed by a pooling layer, which aggregates highly informative nodes on the molecular graph [49]. The DM consists of one layer due to the small average size of the molecules (34 nodes). ReLU, BatchNormalization and Dropout were applied here as well.

The purpose of the GCN is to map each $v_i \in V$ to low dimensional vectors $z_i \in R$. The formal mapping is as follows:

$$f_{DM}(A, X; \theta_{DM}) \rightarrow Z ,$$

with $Z \in R^{n \times d}$ for compound c_j, where n is the number of atoms, and d is the dimension of the latent space produced by the GCN.

Furthermore, to obtain a latent representation z_d for graph c_j, we computed both average and maximal features across Z and concatenated them with the following operation:

$$z_d = \left(\frac{1}{n} \sum_{i=1}^{n} Z_{i,j} \parallel max_{1 \le i \le n} Z_{i,j} \right)$$

, where $z_d \in R^{n \times 2d}$ and $\|$ denotes the concatenation operator. The dimensionality is doubled due to concatenation of both average and maximal features for each graph.

Prediction Module (PM). The PM of the model consists of one fully connected layer, and aims at predicting log(IC_{50}) using highly informative features derived from the EM and DM. As such, the operation carried out by the PM is the following:

$$f_{PM} = \left(z_c \| z_d ; \theta_{PM} \right)$$

Our model updates the weights of EM, DM, and PM by minimizing the mean squared error (MSE),

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \widehat{y_i} \right)^2 ,$$

where N is the number of samples,

between the observed and predicted log(IC_{50}), denoted by y and $\widehat{y}$, respectively, and L_{MMD}. Hence, the overall loss minimized by the model is:

$$L_{PACE} = L_{MSE} + \lambda \cdot L_{MMD}$$

where λ controls the tradeoff between the goals of aligning the CL latent space with the primary tumor latent space, and achieving an accurate predicted log(IC_{50}).

Training Procedure and Tuning. Our model was implemented in Python with the PyTorch API[50] using the Adam optimizer [51] for gradient descent optimization. The training was allowed to proceed for a maximum of 200 epochs. To control for overfitting EarlyStopping was used to monitor the training loss for overfitting. Training was terminated after 10 epochs if the training loss was not further minimized after 10 consecutive epochs, with a delta of 0.05. Dropout was applied on a random 35% of nodes to further prevent overfitting. We used the Adam[51] optimizer for gradient descent optimization with a learning rate of 1E-4. Given the stochasticity of the training procedure, and that we wanted to achieve considerable robustness with our model when predicting CDR of patients, we repeated the training 10 independent times.

Due to the computational expense of training, the number of layers for the DM and PM were fixed to one, and the number of layers of the EM were fixed to 2. We experimented with the λ, and with the number of drug nodes for the DM. We found that $\lambda = 0.01$ and 200 drug nodes were the best parameters for distinguishing sensitive from resistant patients in the CDR dataset

CDR prediction in TCGA patients. We obtained the clinical drug response (CDR) of 531 TCGA patients across 24 drugs from this study [17]. Following the same filtering steps as Huang et al. resulted in 12 drugs. Finally, after filtering for patients for which we had gene expression information resulted in 506 patients. Patients with “clinical progressive disease” or “stable disease” were labeled as resistant (**R**). Those with “partial response” or “complete response” were labeled sensitive (**S**). These are categorical variables, whereas our model predicts $\log(IC_{50})$ which is a continuous variable. To test how well our model can be extended to OOD samples, we grouped the predicted $\log(IC_{50})$ of each patient in the corresponding R or S bin. Then, we tested if the predicted $\log(IC_{50})$ of the R patients was significantly larger than that of the S patients by performing a one-sided nonparametric Mann Whitney U test. A summary of the number of R and S patients for each drug is shown in Table 3.

Table 3. Number of Resistant and Sensitive Patients in TCGA CDR dataset

Drug	Num Resistance	Num Sensitive	N of CL in Training	Mode of Action
bicalutamide	3	14	525	Androgen receptor antagonist
bleomycin	4	46	470	DNA synthesis inhibitor
cisplatine	25	108	524	DNA synthesis inhibitor
docetaxel	17	55	524	Tubulin polymerization inhibitor
doxorubicin	7	52	479	Topoisomerase inhibitor
etoposide	10	71	484	Topoisomerase inhibitor
gemcitabine	43	37	509	Ribonucleotide reductase inhibitor
paclitaxel	27	66	470	Tubulin polymerization inhibitor
sorafenib	13	2	470	FLT3 inhibitor
tamoxifen	4	14	520	ESR1 inhibitor
temozolomide	83	10	520	DNA alkylating agent
vinorelbine	6	23	513	Tubulin polymerization inhibitor

Drug/CL exclusion experiment. For dropout analysis, we created random train splits in a 10-fold cross validation. After training on each fold 10 independent times, we tested the generalizability potential of our model in the CDR dataset for each fold, thereby producing 10 p-values (see ***CDR prediction in TCGA patients)***. For drug-centered dropout analysis, we created train sets by first removing all 12 CDR drugs (bicalutamide, bleomycin, cisplatine, docetaxel, doxorubicin, etoposide, gemcitabine, paclitaxel, sorafenib, tamoxifen, temozolomide, and vinorelbine), and then retaining random 20%, 40%, 60%, and 80% of the remaining 298 drugs (310 drugs in total). Similar to the drug-centered dropout analysis, the CL-centered dropout was carried out in a similar manner without removing the 12 CDR drugs.

Expression Datasets. We downloaded gene expression data of 1376 cell lines of the Cancer Cell Line Encyclopedia (CCLE) project, along with their metadata [52], and 10,536 TCGA pan-cancer tumors from the DepMap project [53] and UCSC Xena browser [54], respectively. All expression values were represented as $\log_2$(TPM+1), where TPM denoted transcripts per million reads of each gene in each sample. The gene space was intersected resulting in 31,501 common genes.

Drug Response Datasets. We downloaded release 8.1 of the GDSC project containing drug response measured by the half maximal inhibitory concentration (IC_{50}) from the DepMap project, which has harmonized cell lines and drug names[32,55]. In total 974 cell lines tested across 398 drugs are included in this dataset, amounting to 387,626 cell line-drug-IC_{50} pairs (CDI pairs). After intersecting for cell lines included in the CCLE RNA-seq compendium, selecting drugs for which we could obtain SMILES string, removing CDI pairs representing combination therapies and pairs with missing values for either drug name or IC_{50}, 692 cell lines tested on 310 drugs remained, amounting to 185,186 CDI pairs. All IC_{50} values were transformed to log scale $\log_{10}(IC_{50})$. After selecting for cell lines that represent the same tissue of origin as the TCGA dataset (25 tumor types), 531 cell lines tested on 310 drugs amounting to 142,351 CDI pairs.

AUTHOR CONTRIBUTIONS

I.A. conceived the idea, performed deep learning framework modeling, optimization and analysis. L.S. contributed in developing ideas on how to align cell lines and patients. H.D., J.S. supervised the project. All authors prepared the manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

CODE AVAILABILITY

The package and API for PACE is available at https://github.com/ioannisa92/PACE. The code and data to train and deploy the model of this manuscript are available on the github.

BIBLIOGRAPHY

1. Verma, M. Personalized Medicine and Cancer. *Journal of Personalized Medicine* vol. 2 1–14 (2012).
2. Goodspeed, A., Heiser, L. M., Gray, J. W. & Costello, J. C. Tumor-Derived Cell Lines as Molecular Models of Cancer Pharmacogenomics. *Mol. Cancer Res.* **14**, 3–13 (2016).
3. Costello, J. C. *et al.* A community effort to assess and improve drug sensitivity prediction algorithms. *Nat. Biotechnol.* **32**, 1202–1212 (2014).
4. Jiang, P., Sellers, W. R. & Liu, X. S. Big Data Approaches for Modeling Response and Resistance to Cancer Drugs. *Annu Rev Biomed Data Sci* **1**, 1–27 (2018).
5. Ali, M. & Aittokallio, T. Machine learning and feature selection for drug response prediction in precision oncology applications. *Biophys. Rev.* (2018) doi:10.1007/s12551-018-0446-z.
6. Geeleher, P., Cox, N. J. & Huang, R. S. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome Biol.* **15**, R47 (2014).
7. Falgreen, S. *et al.* Predicting response to multidrug regimens in cancer patients

using cell line experiments and regularised regression models. *BMC Cancer* **15**, 235 (2015).

8. Geeleher, P. *et al.* Discovering novel pharmacogenomic biomarkers by imputing drug response in cancer patients from large genomics studies. *Genome Res.* **27**, 1743–1751 (2017).
9. Yu, K. *et al.* Comprehensive transcriptomic analysis of cell lines as models of primary tumors across 22 tumor types. *Nature Communications* vol. 10 (2019).
10. Domcke, S., Sinha, R., Levine, D. A., Sander, C. & Schultz, N. Evaluating cell lines as tumour models by comparison of genomic profiles. *Nat. Commun.* **4**, 2126 (2013).
11. Chen, B., Sirota, M., Fan-Minogue, H., Hadley, D. & Butte, A. J. Relating hepatocellular carcinoma tumor samples and cell lines using gene expression data in translational research. *BMC Medical Genomics* vol. 8 (2015).
12. Jiang, G. *et al.* Comprehensive comparison of molecular portraits between cell lines and tumors in breast cancer. *BMC Genomics* **17 Suppl 7**, 525 (2016).
13. Vincent, K. M., Findlay, S. D. & Postovit, L. M. Assessing breast cancer cell lines as tumour models by comparison of mRNA expression profiles. *Breast Cancer Res.* **17**, 114 (2015).
14. Ge, S., Wang, H., Alavi, A., Xing, E. & Bar-Joseph, Z. Supervised Adversarial Alignment of Single-Cell RNA-seq Data. doi:10.1101/2020.01.06.896621.
15. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **13**, 723–773 (2012).
16. Louizos, C., Swersky, K., Li, Y., Welling, M. & Zemel, R. The Variational Fair

Autoencoder. *arXiv [stat.ML]* (2015).

17. Huang, E. W., Bhope, A., Lim, J., Sinha, S. & Emad, A. Tissue-guided LASSO for prediction of clinical drug response using preclinical samples. *PLoS Comput. Biol.* **16**, e1007607 (2020).
18. Yap, C. W. PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry* vol. 32 1466–1474 (2011).
19. Chang, Y. *et al.* Cancer Drug Response Profile scan (CDRscan): A Deep Learning Model That Predicts Drug Effectiveness from Cancer Genomic Signature. *Sci. Rep.* **8**, 8857 (2018).
20. Mayr, A., Klambauer, G., Unterthiner, T. & Hochreiter, S. DeepTox: Toxicity Prediction using Deep Learning. *Front. Environ. Sci. Eng. China* **3**, 80 (2016).
21. Kuenzi, B. M. *et al.* Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells. *Cancer Cell* **38**, 672–684.e6 (2020).
22. Zhavoronkov, A. *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **37**, 1038–1040 (2019).
23. Zitnik, M., Agrawal, M. & Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34**, i457–i466 (2018).
24. Finlayson, S. G., McDermott, M. B. A., Pickering, A. V., Lipnick, S. L. & Kohane, I. S. Cross-modal representation alignment of molecular structure and perturbation-induced transcriptional profiles. in *Biocomputing 2021* 273–284 (WORLD SCIENTIFIC, 2020).
25. Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional

out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics* **36**, i610–i617 (2020).

26. Long, M., Cao, Y., Wang, J. & Jordan, M. Learning Transferable Features with Deep Adaptation Networks. in *Proceedings of the 32nd International Conference on Machine Learning* (eds. Bach, F. & Blei, D.) vol. 37 97–105 (PMLR, 2015).
27. Zhang, X., Yu, F. X., Chang, S.-F. & Wang, S. Deep Transfer Network: Unsupervised Domain Adaptation. *arXiv [cs.CV]* (2015).
28. Farahani, A., Voghoei, S., Rasheed, K. & Arabnia, H. R. A Brief Review of Domain Adaptation. *arXiv [cs.LG]* (2020).
29. Pham, T.-H., Qiu, Y., Zeng, J., Xie, L. & Zhang, P. A deep learning framework for high-throughput mechanism-driven phenotype compound screening. doi:10.1101/2020.07.19.211235.
30. Stokes, J. M. *et al.* A Deep Learning Approach to Antibiotic Discovery. *Cell* **181**, 475–483 (2020).
31. Keshavarzi Arshadi, A., Salem, M., Collins, J., Yuan, J. S. & Chakrabarti, D. DeepMalaria: Artificial Intelligence Driven Discovery of Potent Antiplasmodials. *Front. Pharmacol.* **10**, 1526 (2019).
32. Iorio, F. *et al.* A Landscape of Pharmacogenomic Interactions in Cancer. *Cell* **166**, 740–754 (2016).
33. Oplustilova, L. *et al.* Evaluation of candidate biomarkers to predict cancer cell sensitivity or resistance to PARP-1 inhibitor treatment. *Cell Cycle* **11**, 3837–3850 (2012).
34. Yi, M. *et al.* Advances and perspectives of PARP inhibitors. *Exp. Hematol. Oncol.* **8**,

29 (2019).

35. Gril, B. *et al.* Effect of lapatinib on the outgrowth of metastatic breast cancer cells to the brain. *J. Natl. Cancer Inst.* **100**, 1092–1103 (2008).
36. McFarland, J. M. *et al.* Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nat. Commun.* **11**, 4296 (2020).
37. McFarland, J. M. *et al.* Improved estimation of cancer dependencies from large-scale RNAi screens using model-based normalization and data integration. *Nat. Commun.* **9**, 4610 (2018).
38. Rodriguez, H., Zenklusen, J. C., Staudt, L. M., Doroshow, J. H. & Lowy, D. R. The next horizon in precision oncology: Proteogenomics to inform cancer diagnosis and treatment. *Cell* **184**, 1661–1670 (2021).
39. Frejno, M. *et al.* Proteome activity landscapes of tumor cell lines determine drug responses. *Nat. Commun.* **11**, 3639 (2020).
40. Seninge, L., Anastopoulos, I., Ding, H. & Stuart, J. Biological network-inspired interpretable variational autoencoder. *Cold Spring Harbor Laboratory* 2020.12.17.423310 (2020) doi:10.1101/2020.12.17.423310.
41. Ma, J. *et al.* Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* **15**, 290–298 (2018).
42. Hu, W. *et al.* Strategies for Pre-training Graph Neural Networks. *arXiv [cs.LG]* (2019).
43. Landrum, G. & Others. RDKit: Open-source cheminformatics. (2006).
44. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**,

742–754 (2010).

45. Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **5**, 107–113 (1965).
46. Fey, M. & Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv [cs.LG]* (2019).
47. Morris, C. *et al.* Weisfeiler and leman go neural: Higher-order graph neural networks. in *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 33 4602–4609 (2019).
48. Such, F. P. *et al.* Robust Spatial Filtering With Graph Convolutional Neural Networks. *IEEE Journal of Selected Topics in Signal Processing* vol. 11 884–896 (2017).
49. Gao, H. & Ji, S. Graph U-Nets. in *Proceedings of the 36th International Conference on Machine Learning* (eds. Chaudhuri, K. & Salakhutdinov, R.) vol. 97 2083–2092 (PMLR, 2019).
50. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv [cs.LG]* (2019).
51. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2014).
52. Ghandi, M. *et al.* Next-generation characterization of the Cancer Cell Line Encyclopedia. *Nature* **569**, 503–508 (2019).
53. Tsherniak, A. *et al.* Defining a Cancer Dependency Map. *Cell* **170**, 564–576.e16 (2017).

54. Goldman, M. J. *et al.* Visualizing and interpreting cancer genomics data via the Xena platform. *Nat. Biotechnol.* **38**, 675–678 (2020).

55. Picco, G. *et al.* Functional linkage of gene fusions to cancer cell fitness assessed by pharmacological and CRISPR-Cas9 screening. *Nat. Commun.* **10**, 2198 (2019).

4.2 Sparse data storage to support petabyte scale genomics data analysis

During an internship at Coral Genomics Inc, I worked with Andres Manas and Atray Dixit to develop an efficient data format in order to perform deep learning tasks using petabyte scale genomics data. Large-scale genomics databases such as the UK BioBank (UKBB) store millions of SNPs for thousands of individuals with several associated phenotypes [Bycroft et al., 2018]. This source of information is promising in helping to predict risk of developing certain diseases or even potential response to treatment using one's genetic information. Such task is called Polygenic Risk Score (PRS) prediction. However, the scale of such database makes computational tasks challenging as it does not readily fit into memory. We took advantage of the sparsity of such dataset and developped a new genetics format on top of Tensorflow's TFRecords to enable fast and efficient machine learning applications on UKBB. I contributed to designing the codebase for converting genetics data from other formats to our adaptation of TFRecords, as well as testing the framework in tasks such as GWAS or PRS prediction using deep learning models in Keras. This work has been recently submitted to *BioRxiv*.

DNARecords: An extensible sparse format for petabyte scale genomics analysis

Andres Manas[1], Lucas Seninge[1,2], Atray Dixit[1]

Abstract:

Recent growth in population scale sequencing initiatives involve both cohort scale and proportion of genome surveyed, with a transition from genotyping arrays to broader genome sequencing approaches. The resulting datasets can be challenging to analyze. Here we introduce DNARecords a novel sparse-compatible format for large scale genetic data. The structure enables integration of complex data types such as medical images and drug structures towards the development of machine learning methods to predict disease risk and drug response. We demonstrate its speed and memory advantages for various genetics analyses. These performance advantages will become more pronounced as it becomes feasible to analyze variants of lower population allele frequencies. Finally, we provide an open-source software plugin, built on top of Hail, to allow researchers to write and read such records as well as a set of examples for how to use them.

[1] Coral Genomics, Inc., 953 Indiana St., San Francisco, CA 94107, USA

[2] Biomolecular Engineering, University of California, Santa Cruz, CA 95064, USA

Correspondence to: atray@coralgenomics.com

Introduction:

Modern population scale sequencing projects involve sequencing hundreds of thousands to millions of individuals in which genetics data can be correlated with a broad assortment of healthcare data. The number of variants that can be analyzed has grown substantially from initial efforts with genotyping arrays to recent releases of whole genome sequencing (WGS) data from both the UK Biobank [1] and the All of Us initiative [2]. As the size of these datasets has grown, new methods have been developed to facilitate their analysis. Tools such as Hail [3] and REGENIE [4] are useful for performing genetic analysis, such as fixed or random effects GWAS efficiently on large patient cohorts in a distributed framework that can be deployed on the cloud. Other frameworks have been developed to enable GPU based operations on genetics datasets [5], including QTL analysis [6], for improved speed, but are oriented for variant-by-variant analysis.

One goal of large population sequencing initiatives is the generation of novel precision medicine solutions to help stratify patients with respect to disease risk (including polygenic risk scores) and therapy selection. Currently, genetics informed clinical predictors are approaching clinical utility for certain complex diseases, including predictors of cardiac risk [7]. Healthcare outcomes are a result of genetic (G), environmental (E), sociodemographic (S), and random effects. There are additional clinical variables that can be informative. Most current approaches for predicting healthcare outcomes using genetic data take into account simple numerical covariates (S and E) such as age or lipid levels. A modeling approach that can take into account other important, but harder to model data types (such as medical images or drug structure) is likely to result in improved predictors for critical outcomes such as hospitalization risk [8], [9]. The integration of these complex non-tabular datatypes is made possible by modern deep learning methods (including forms of convolutional neural network architectures), which operate

on datasets oriented in a sample-wise fashion. Lastly, there is significant evidence that there is a small, but significant non-additive component to the heritability of complex human traits [10], [11]. Modeling these nonlinearities is facilitated by data structures organized by sample. As such, there is a need for analysis frameworks that restructure genetics datasets in a machine learning compatible format (i.e. the transpose of traditional genetics data storage formats).

Here we introduce DNARecords, a new genetics data format that has significant advantages for speed and novel machine learning applications. The format leverages sparsity in genetics datasets for faster computation and can be stored in a sample-wise manner for integrations of complex data types. Moreover, by working on top of existing frameworks including Tensorflow's TFRecords and Apache Parquet, datasets can be distributed in the cloud for analysis as well as across GPUs or TPUs [12].

Results and Discussion:

DNARecords stores genetic data in both sample-wise and variant-wise data structures in which sparsity is leveraged (**Figure 1A**). Specifically, entries where the genotypes are homozygous for the reference allele (or relatedly when genotype dosage is below a certain threshold) can be stored implicitly. We show that when using this format to perform genetics analysis like GWAS there are negligible losses in sensitivity or specificity to detect significant variants when the dosage threshold is less than 0.1 (**Figure 1C**). The performance advantages of DNARecords in memory footprint for several example datasets are described below (**Table 1**). These advantages become pronounced with larger WES and WGS datasets, in which progressively rare variants with lower population allele frequencies are considered for analysis.

Dataset	Number of samples	Number of variants	Average allele frequency	HWE estimated sparsity factor[3]
Genotyping array[4]	3,000	693,158	0.1434	3.8
Imputed genotypes AF>0.001[3]	490,030	13,000,000	0.081	6.4
Imputed genotypes[5]	490,030	100,000,000	0.021	24
WGS[4]	76,156	759,302,267	0.0043	116
WES[6]	125,748	17,201,296	0.0016	312

Table 1: Footprint comparison of DNARecords to other genetics datatypes as a function of genotyping assay.

These dataset size advantages are propagated into faster analysis for operations like GWAS and PCA (see **Table 2**).

Method	GWAS	PCA	2-layer model with images
Hail	1.11	23.73	N/A
DNARecords (+1 P100 GPU)	0.55	1.16	1.15

Table 2: CPU hours. Speed comparisons for basic genetics operations for 2.5M variant dataset with the 1,000 genomes dataset. Numbers extrapolated based on the UK Biobank dataset.

[3] $\frac{1}{2AF-AF^2}$
[4] based on 1,000 genomes
[5] based on UK Biobank
[6] based on gnomAD

Methods:

The basic procedures for data generation are described here: https://dnarecords.readthedocs.io/.

A python package for generating DNARecords for variant-wise analysis or sample wise machine learning is available here: https://pypi.org/project/dnarecords/. Open source code is available in Github here: https://github.com/amanas/dnarecords. DNARecords datasets for the 1kg dataset are hosted in Google Cloud storage (in both TFRecords and Parquet format) here: gs://dnarecords/1kg

Acknowledgements

This work was supported by an NIH NHGRI 2R44HG010445-02.

Author contributions

A.M. developed the data pipeline to transform standard genetics formats into DNARecords. L.S., A.M., and A.D. performed computational analysis. A.D. drafted the original manuscript. A.M., L.S., and A.D. revised the manuscript.

Competing interests

A.D. is an equity holder in Coral Genomics, Inc.

Figures

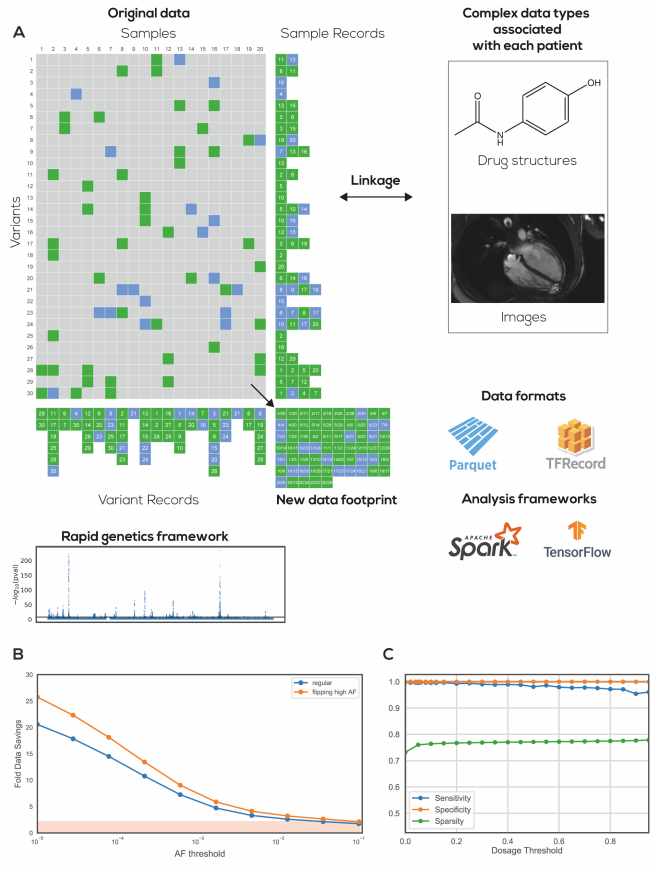

Figure 1: (**A**) DNARecords data format stores variant data separately in sparse sample oriented and variant oriented files to allow for efficient parallelism on standard genetics operations as well as new machine learning approaches. Sample-wise records can be linked to complex patient specific datatypes such as graph-encoded chemical structures and medical images (**B**) Data savings associated with thresholding variants above different allele frequency cutoffs. (**C**) Relationship between increasing sparsity (green) by adjusting the threshold for dosage and sensitivity (blue) /specificity (green) in a GWAS for height in the UK Biobank on chromosome 22.

References:

[1] C. Bycroft *et al*., "The UK Biobank resource with deep phenotyping and genomic data," *Nature*, vol. 562, no. 7726, pp. 203–209, Oct. 2018.

[2] All of Us Research Investigators, "The 'All of Us' Research Program," *N. Engl. J. Med.*, vol. 381, no. 7, pp. 668–676, Aug. 2019.

[3] Hail Team, "Hail."

[4] J. Mbatchou *et al*., "Computationally efficient whole-genome regression for quantitative and binary traits," *Nat. Genet.*, vol. 53, no. 7, pp. 1097–1103, Jul. 2021.

[5] J. Freudenthal, M. Ankenbrand, D. Grimm, and A. Korte, "GWAS-Flow: A GPU accelerated framework for efficient permutation based genome-wide association studies," *bioRxiv*, 2019.

[6] A. Taylor-Weiner *et al*., "Scaling computational genomics to millions of individuals with GPUs," *Genome Biol.*, vol. 20, no. 1, p. 228, Dec. 2019.

[7] A. V. Khera *et al*., "Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations," *Nat. Genet.*, vol. 50, no. 9, pp. 1219–1224, Sep. 2018.

[8] I. Anastopoulos, C. Herczeg, K. Davis, and A. Dixit, "Multi-Drug Featurization and Deep Learning Improve Patient-Specific Predictions of Adverse Events," *Int. J. Environ. Res. Public Health*, vol. 18, no. 5, p. 2600, Mar. 2021.

[9] M. E. Haas *et al*., "Machine learning enables new insights into genetic contributions to liver fat accumulation," *Cell Genomics*, vol. 1, no. 3, p. 100066, Dec. 2021.

[10] B. Sheppard, N. Rappoport, P.-R. Loh, S. J. Sanders, N. Zaitlen, and A. Dahl, "A model and test for coordinated polygenic epistasis in complex traits," *Proc. Natl. Acad. Sci.*, vol.

118, no. 15, Apr. 2021.

[11] A. I. Young and R. Durbin, "Estimation of Epistatic Variance Components and Heritability in Founder Populations and Crosses," *Genetics*, vol. 198, no. 4, pp. 1405–1416, Dec. 2014.

[12] M. Abadi *et al*., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016.

4.3 Minor contributions

I also had minor contributions to other work during my PhD work. I performed single-cell analysis for [Sanders et al., 2020] and [Robinson et al., 2022]. I also contributed to some of the codebase of [Speir et al., 2021].

Appendix

A.0.1 Penalty in the ontology mapping score function

Here, we give a sense on how to regularize the ontology mapping score we developed in the first chapter.

A.0.1.1 Bayesian approach to mapping

One simple way to regularize the score $S_{i,j}$ is to simply put a prior on the μ_j. We are now interested in the "posterior mapping score":

$$\begin{aligned} S_{i,j}^{post} &= \log P(\mu_j|C_i) - \log P(\mathcal{B}|C_i) \\ &= \log P(C_i|\mu_j) - \log P(C_i|\mathcal{B}) + \log \frac{P(\mu_j)}{P(\mathcal{B})} \end{aligned}$$

Note that when we consider a uniform prior on the nodes of the ontology, the posterior mapping score is similar to the non-regularized score.

A.0.1.2 Regularization via parent information

Another desirable property of a penalty term is to regularize for parent information. Intuitively, mapping to a successor should only happen if the unique successor marker genes bring enough additional information compared to its parents. Formally, the regularized score has the form:

$$S_{i,j}^{\Omega} = S_{i,j} + \Omega(\sum_{k \in \pi(j)} w_k S_{i,k})$$

where $S_{i,j}^{\Omega}$ is the parent-regularized score, Ω is a regularization function, and $S_{i,\pi(j)}$ are the scores of the parent nodes for node j.

We leave the exploration of such regularization approaches to further studies.

References

[Barbie et al., 2009] Barbie, D. A., Tamayo, P., Boehm, J. S., Kim, S. Y., Moody, S. E., Dunn, I. F., Schinzel, A. C., Sandy, P., Meylan, E., Scholl, C., Fröhling, S., Chan, E. M., Sos, M. L., Michel, K., Mermel, C., Silver, S. J., Weir, B. A., Reiling, J. H., Sheng, Q., Gupta, P. B., Wadlow, R. C., Le, H., Hoersch, S., Wittner, B. S., Ramaswamy, S., Livingston, D. M., Sabatini, D. M., Meyerson, M., Thomas, R. K., Lander, E. S., Mesirov, J. P., Root, D. E., Gilliland, D. G., Jacks, T., and Hahn, W. C. (2009). Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1. *Nature*, 462(7269):108–112.

[Berger, 1985] Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer, New York, NY.

[Blondel et al., 2008] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008. arXiv:0803.0476 [cond-mat, physics:physics].

[Boyeau et al., 2019] Boyeau, P., Lopez, R., Regier, J., Gayoso, A., Jordan, M. I., and Yosef, N. (2019). Deep Generative Models for Detecting Differential Expression in Single Cells. preprint, Cell Biology.

[Bycroft et al., 2018] Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., Cortes, A., Welsh, S., Young, A., Effingham, M., McVean, G., Leslie, S., Allen, N., Donnelly, P., and Marchini, J. (2018). The UK Biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209. Number: 7726 Publisher: Nature Publishing Group.

[Costa-Silva et al., 2017] Costa-Silva, J., Domingues, D., and Lopes, F. M. (2017). RNA-Seq differential expression analysis: An extended review and a software tool. *PLOS ONE*, 12(12):e0190152. Publisher: Public Library of Science.

[Cui et al., 2015] Cui, S., Guha, S., Ferreira, M. A. R., and Tegge, A. N. (2015). hmmSeq: A hidden Markov model for detecting differentially expressed genes from RNA-seq data. *The Annals of Applied Statistics*, 9(2).

[Delaney et al., 2019] Delaney, C., Schnell, A., Cammarata, L. V., Yao-Smith, A., Regev, A., Kuchroo, V. K., and Singer, M. (2019). Combinatorial prediction of marker panels from single-cell transcriptomic data. *Molecular Systems Biology*, 15(10):e9005. Publisher: John Wiley & Sons, Ltd.

[Diaz-Mejia et al., 2019] Diaz-Mejia, J. J., Meng, E. C., Pico, A. R., MacParland, S. A., Ketela, T., Pugh, T. J., Bader, G. D., and Morris, J. H. (2019). Evaluation of methods to assign cell type labels to cell clusters from single-cell RNA-sequencing data. Technical Report 8:296, F1000Research. Type: article.

[Diehl et al., 2016] Diehl, A. D., Meehan, T. F., Bradford, Y. M., Brush, M. H., Dahdul,

W. M., Dougall, D. S., He, Y., Osumi-Sutherland, D., Ruttenberg, A., Sarntivijai, S., Van Slyke, C. E., Vasilevsky, N. A., Haendel, M. A., Blake, J. A., and Mungall, C. J. (2016). The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. *Journal of Biomedical Semantics*, 7(1):44.

[Doubilet et al., 1972] Doubilet, P., Rota, G.-C., and Stanley, R. (1972). On the foundations of combinatorial theory. VI. The idea of generating function. *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, 6.2:267–319. Publisher: University of California Press.

[Farahani et al., 2020] Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. (2020). A Brief Review of Domain Adaptation. arXiv:2010.03978 [cs].

[Field et al., 2019] Field, A. R., Jacobs, F. M., Fiddes, I. T., Phillips, A. P., Reyes-Ortiz, A. M., LaMontagne, E., Whitehead, L., Meng, V., Rosenkrantz, J. L., Olsen, M., Hauessler, M., Katzman, S., Salama, S. R., and Haussler, D. (2019). Structurally Conserved Primate LncRNAs Are Transiently Expressed during Human Cortical Differentiation and Influence Cell-Type-Specific Genes. *Stem Cell Reports*, 12(2):245–257.

[Friedman, 2004] Friedman, N. (2004). Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, 303(5659):799–805.

[Gardner et al., 2003] Gardner, T. S., di Bernardo, D., Lorenz, D., and Collins, J. J. (2003). Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science*, 301(5629):102–105. Publisher: American Association for the Advancement of Science.

[Gayoso et al., 2022] Gayoso, A., Lopez, R., Xing, G., Boyeau, P., Valiollah Pour Amiri, V., Hong, J., Wu, K., Jayasuriya, M., Mehlman, E., Langevin, M., Liu, Y., Samaran, J., Misrachi, G., Nazaret, A., Clivio, O., Xu, C., Ashuach, T., Gabitto, M., Lotfollahi, M., Svensson, V., da Veiga Beltrame, E., Kleshchevnikov, V., Talavera-López, C., Pachter, L., Theis, F. J., Streets, A., Jordan, M. I., Regier, J., and Yosef, N. (2022). A Python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, 40(2):163–166. Number: 2 Publisher: Nature Publishing Group.

[Ghandi et al., 2019] Ghandi, M., Huang, F. W., Jané-Valbuena, J., Kryukov, G. V., Lo, C. C., McDonald, E. R., Barretina, J., Gelfand, E. T., Bielski, C. M., Li, H., Hu, K., Andreev-Drakhlin, A. Y., Kim, J., Hess, J. M., Haas, B. J., Aguet, F., Weir, B. A., Rothberg, M. V., Paolella, B. R., Lawrence, M. S., Akbani, R., Lu, Y., Tiv, H. L., Gokhale, P. C., Weck, A. d., Mansour, A. A., Oh, C., Shih, J., Hadi, K., Rosen, Y., Bistline, J., Venkatesan, K., Reddy, A., Sonkin, D., Liu, M., Lehar, J., Korn, J. M., Porter, D. A., Jones, M. D., Golji, J., Caponigro, G., Taylor, J. E., Dunning, C. M., Creech, A. L., Warren, A. C., McFarland, J. M., Zamanighomi, M., Kauffmann, A., Stransky, N., Imielinski, M., Maruvka, Y. E., Cherniack, A. D., Tsherniak, A., Vazquez, F., Jaffe, J. D., Lane, A. A., Weinstock, D. M., Johannessen, C. M., Morrissey, M. P., Stegmeier, F., Schlegel, R., Hahn, W. C., Getz, G., Mills, G. B., Boehm, J. S., Golub,

T. R., Garraway, L. A., and Sellers, W. R. (2019). Next-generation characterization of the Cancer Cell Line Encyclopedia. *Nature*, 569(7757):503–508. Number: 7757 Publisher: Nature Publishing Group.

[Goode et al., 2016] Goode, D., Obier, N., Vijayabaskar, M., Lie-A-Ling, M., Lilly, A., Hannah, R., Lichtinger, M., Batta, K., Florkowska, M., Patel, R., Challinor, M., Wallace, K., Gilmour, J., Assi, S., Cauchy, P., Hoogenkamp, M., Westhead, D., Lacaud, G., Kouskoff, V., Göttgens, B., and Bonifer, C. (2016). Dynamic Gene Regulatory Networks Drive Hematopoietic Specification and Differentiation. *Developmental Cell*, 36(5):572–587.

[Gootjes-Dreesbach et al., 2020] Gootjes-Dreesbach, L., Sood, M., Sahay, A., Hofmann-Apitius, M., and Fröhlich, H. (2020). Variational Autoencoder Modular Bayesian Networks for Simulation of Heterogeneous Clinical Study Data. *Frontiers in Big Data*, 3:16.

[Gut et al., 2021] Gut, G., Stark, S. G., Rätsch, G., and Davidson, N. R. (2021). pmVAE: Learning Interpretable Single-Cell Representations with Pathway Modules. preprint, Bioinformatics.

[Hinton, 2006] Hinton, G. E. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507.

[Hood et al., 2015] Hood, D. A., Tryon, L. D., Vainshtein, A., Memme, J., Chen, C., Pauly, M., Crilly, M. J., and Carter, H. (2015). Chapter Five - Exercise and the Regulation of Mitochondrial Turnover. In Bouchard, C., editor, *Progress in Molecular Biology and Translational Science*, volume 135 of *Molecular and Cellular Regulation of Adaptation to Exercise*, pages 99–127. Academic Press.

[Hänzelmann et al., 2013] Hänzelmann, S., Castelo, R., and Guinney, J. (2013). GSVA: gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14(1):7.

[International Human Genome Sequencing Consortium, 2004] International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945. Number: 7011 Publisher: Nature Publishing Group.

[Islam et al., 2014] Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P., and Linnarsson, S. (2014). Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166.

[Janizek et al., 2022] Janizek, J. D., Spiro, A., Celik, S., Blue, B. W., Russell, J. C., Lee, T.-I., Kaeberlin, M., and Lee, S.-I. (2022). Principled feature attribution for unsupervised gene expression analysis. preprint, Bioinformatics.

[Jolliffe, 1986] Jolliffe, I. T. (1986). Principal Component Analysis and Factor Analysis.

In Jolliffe, I. T., editor, *Principal Component Analysis*, Springer Series in Statistics, pages 115–128. Springer, New York, NY.

[Jordan et al., 1998] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1998). An Introduction to Variational Methods for Graphical Models. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 105–161. Springer Netherlands, Dordrecht.

[Kharchenko et al., 2014] Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742. Number: 7 Publisher: Nature Publishing Group.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*. arXiv: 1312.6114.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs, stat].

[Kiselev et al., 2018] Kiselev, V. Y., Yiu, A., and Hemberg, M. (2018). scmap: projection of single-cell RNA-seq data across data sets. *Nature Methods*, 15(5):359–362. Number: 5 Publisher: Nature Publishing Group.

[Kolodziejczyk et al., 2015] Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., and Teichmann, S. A. (2015). The technology and biology of single-cell RNA sequencing. *Molecular Cell*, 58(4):610–620.

[Krumsiek et al., 2011] Krumsiek, J., Marr, C., Schroeder, T., and Theis, F. J. (2011). Hierarchical Differentiation of Myeloid Progenitors Is Encoded in the Transcription Factor Network. *PLOS ONE*, 6(8):e22649. Publisher: Public Library of Science.

[Lee and Young, 2013] Lee, T. and Young, R. (2013). Transcriptional Regulation and Its Misregulation in Disease. *Cell*, 152(6):1237–1251.

[Lonsdale et al., 2013] Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., Hasz, R., Walters, G., Garcia, F., Young, N., Foster, B., Moser, M., Karasik, E., Gillard, B., Ramsey, K., Sullivan, S., Bridge, J., Magazine, H., Syron, J., Fleming, J., Siminoff, L., Traino, H., Mosavel, M., Barker, L., Jewell, S., Rohrer, D., Maxim, D., Filkins, D., Harbach, P., Cortadillo, E., Berghuis, B., Turner, L., Hudson, E., Feenstra, K., Sobin, L., Robb, J., Branton, P., Korzeniewski, G., Shive, C., Tabor, D., Qi, L., Groch, K., Nampally, S., Buia, S., Zimmerman, A., Smith, A., Burges, R., Robinson, K., Valentino, K., Bradbury, D., Cosentino, M., Diaz-Mayoral, N., Kennedy, M., Engel, T., Williams, P., Erickson, K., Ardlie, K., Winckler, W., Getz, G., DeLuca, D., MacArthur, D., Kellis, M., Thomson, A., Young, T., Gelfand, E., Donovan, M., Meng, Y., Grant, G., Mash, D., Marcus, Y., Basile, M., Liu, J., Zhu, J., Tu, Z., Cox, N. J., Nicolae, D. L., Gamazon, E. R., Im, H. K., Konkashbaev, A., Pritchard, J., Stevens, M., Flutre, T., Wen, X., Dermitzakis, E. T., Lappalainen, T., Guigo, R., Monlong, J., Sammeth, M., Koller, D., Battle, A., Mostafavi, S., McCarthy, M., Rivas, M., Maller, J., Rusyn, I., Nobel, A., Wright, F., Shabalin, A., Feolo, M.,

Sharopova, N., Sturcke, A., Paschal, J., Anderson, J. M., Wilder, E. L., Derr, L. K., Green, E. D., Struewing, J. P., Temple, G., Volpi, S., Boyer, J. T., Thomson, E. J., Guyer, M. S., Ng, C., Abdallah, A., Colantuoni, D., Insel, T. R., Koester, S. E., Little, A. R., Bender, P. K., Lehner, T., Yao, Y., Compton, C. C., Vaught, J. B., Sawyer, S., Lockhart, N. C., Demchok, J., and Moore, H. F. (2013). The Genotype-Tissue Expression (GTEx) project. *Nature Genetics*, 45(6):580–585. Number: 6 Publisher: Nature Publishing Group.

[Lopez et al., 2018] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058. Number: 12 Publisher: Nature Publishing Group.

[Lotfollahi et al., 2022a] Lotfollahi, M., Naghipourfar, M., Luecken, M. D., Khajavi, M., Büttner, M., Wagenstetter, M., Avsec, , Gayoso, A., Yosef, N., Interlandi, M., Rybakov, S., Misharin, A. V., and Theis, F. J. (2022a). Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40(1):121–130. Number: 1 Publisher: Nature Publishing Group.

[Lotfollahi et al., 2022b] Lotfollahi, M., Rybakov, S., Hrovatin, K., Hediyeh-zadeh, S., Talavera-López, C., Misharin, A. V., and Theis, F. J. (2022b). Biologically informed deep learning to infer gene program activity in single cells. preprint, Bioinformatics.

[Lotfollahi et al., 2019] Lotfollahi, M., Wolf, F. A., and Theis, F. J. (2019). scGen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721. Number: 8 Publisher: Nature Publishing Group.

[Love et al., 2014] Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550.

[Ma and Pellegrini, 2020] Ma, F. and Pellegrini, M. (2020). ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics*, 36(2):533–538.

[Macosko et al., 2015] Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A., and McCarroll, S. A. (2015). Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, 161(5):1202–1214. Publisher: Elsevier.

[MacParland et al., 2018] MacParland, S. A., Liu, J. C., Ma, X.-Z., Innes, B. T., Bartczak, A. M., Gage, B. K., Manuel, J., Khuu, N., Echeverri, J., Linares, I., Gupta, R., Cheng, M. L., Liu, L. Y., Camat, D., Chung, S. W., Seliga, R. K., Shao, Z., Lee, E., Ogawa, S., Ogawa, M., Wilson, M. D., Fish, J. E., Selzner, M., Ghanekar, A., Grant, D., Greig, P., Sapisochin, G., Selzner, N., Winegarden, N., Adeyi, O., Keller, G., Bader, G. D., and McGilvray, I. D. (2018). Single cell RNA sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nature Communications*, 9(1):4383. Number: 1 Publisher: Nature Publishing Group.

[Mann and Whitney, 1947] Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60. Publisher: Institute of Mathematical Statistics.

[Muller et al., 2006] Muller, P., Parmigiani, G., and Rice, K. (2006). FDR and Bayesian Multiple Comparisons Rules. *Johns Hopkins University, Dept. of Biostatistics Working Papers*.

[Neal, 1996] Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, NY.

[Oh and Jang, 2019] Oh, Y. and Jang, J. (2019). Directed Differentiation of Pluripotent Stem Cells by Transcription Factors. *Molecules and Cells*, 42(3):200–209.

[Pearl, 1995] Pearl, J. (1995). Causal Diagrams for Empirical Research. *Biometrika*, 82(4):669–688. Publisher: [Oxford University Press, Biometrika Trust].

[Pearson, Alexander, 2020] Pearson, Alexander (2020). Spam Filters Meet Cell Annotators: Creating and Contextualizing Cell Annotations with the Cell Ontology and Marker Databases.

[Pe'er et al., 2002] Pe'er, D., Regev, A., and Tanay, A. (2002). Minreg: Inferring an active regulator set. *Bioinformatics*, 18(suppl_1):S258–S267.

[Pliner et al., 2019] Pliner, H. A., Shendure, J., and Trapnell, C. (2019). Supervised classification enables rapid annotation of cell atlases. *Nature Methods*, 16(10):983–986. Number: 10 Publisher: Nature Publishing Group.

[Robinson et al., 2022] Robinson, E. K., Worthington, A., Poscablo, D., Shapleigh, B., Salih, M. M., Halasz, H., Seninge, L., Mosqueira, B., Smaliy, V., Forsberg, E. C., and Carpenter, S. (2022). *lincRNA-Cox2* Functions to Regulate Inflammation in Alveolar Macrophages during Acute Lung Injury. *The Journal of Immunology*, 208(8):1886–1900.

[Rybakov et al., 2020] Rybakov, S., Lotfollahi, M., Theis, F. J., and Wolf, F. A. (2020). Learning interpretable latent autoencoder representations with annotations of feature sets. preprint, Bioinformatics.

[Saelens et al., 2019] Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554. Number: 5 Publisher: Nature Publishing Group.

[Sanders et al., 2020] Sanders, L. M., Cheney, A., Seninge, L., van den Bout, A., Chen, M., Beale, H. C., Kephart, E. T., Pfeil, J., Learned, K., Lyle, A. G., Bjork, I., Haussler, D., Salama, S. R., and Vaske, O. M. (2020). Identification of a differentiation stall in epithelial mesenchymal transition in histone H3–mutant diffuse midline glioma. *GigaScience*, 9(12):giaa136.

[Satoh et al., 2013] Satoh, J.-i., Kawana, N., and Yamamoto, Y. (2013). Pathway Analysis of ChIP-Seq-Based NRF1 Target Genes Suggests a Logical Hypothesis of their Involvement in the Pathogenesis of Neurodegenerative Diseases. *Gene Regulation and Systems Biology*, 7:139–152.

[Schaum et al., 2018] Schaum, N., Karkanias, J., Neff, N. F., May, A. P., Quake, S. R., Wyss-Coray, T., Darmanis, S., Batson, J., Botvinnik, O., Chen, M. B., Chen, S., Green, F., Jones, R. C., Maynard, A., Penland, L., Pisco, A. O., Sit, R. V., Stanley, G. M., Webber, J. T., Zanini, F., Baghel, A. S., Bakerman, I., Bansal, I., Berdnik, D., Bilen, B., Brownfield, D., Cain, C., Chen, M. B., Chen, S., Cho, M., Cirolia, G., Conley, S. D., Darmanis, S., Demers, A., Demir, K., de Morree, A., Divita, T., du Bois, H., Dulgeroff, L. B. T., Ebadi, H., Espinoza, F. H., Fish, M., Gan, Q., George, B. M., Gillich, A., Green, F., Genetiano, G., Gu, X., Gulati, G. S., Hang, Y., Hosseinzadeh, S., Huang, A., Iram, T., Isobe, T., Ives, F., Jones, R. C., Kao, K. S., Karnam, G., Kershner, A. M., Kiss, B. M., Kong, W., Kumar, M. E., Lam, J. Y., Lee, D. P., Lee, S. E., Li, G., Li, Q., Liu, L., Lo, A., Lu, W.-J., Manjunath, A., May, A. P., May, K. L., May, O. L., Maynard, A., McKay, M., Metzger, R. J., Mignardi, M., Min, D., Nabhan, A. N., Neff, N. F., Ng, K. M., Noh, J., Patkar, R., Peng, W. C., Penland, L., Puccinelli, R., Rulifson, E. J., Schaum, N., Sikandar, S. S., Sinha, R., Sit, R. V., Szade, K., Tan, W., Tato, C., Tellez, K., Travaglini, K. J., Tropini, C., Waldburger, L., van Weele, L. J., Wosczyna, M. N., Xiang, J., Xue, S., Youngyunpipatkul, J., Zanini, F., Zardeneta, M. E., Zhang, F., Zhou, L., Bansal, I., Chen, S., Cho, M., Cirolia, G., Darmanis, S., Demers, A., Divita, T., Ebadi, H., Genetiano, G., Green, F., Hosseinzadeh, S., Ives, F., Lo, A., May, A. P., Maynard, A., McKay, M., Neff, N. F., Penland, L., Sit, R. V., Tan, W., Waldburger, L., Youngyunpipatkul, J., Batson, J., Botvinnik, O., Castro, P., Croote, D., Darmanis, S., DeRisi, J. L., Karkanias, J., Pisco, A. O., Stanley, G. M., Webber, J. T., Zanini, F., Baghel, A. S., Bakerman, I., Batson, J., Bilen, B., Botvinnik, O., Brownfield, D., Chen, M. B., Darmanis, S., Demir, K., de Morree, A., Ebadi, H., Espinoza, F. H., Fish, M., Gan, Q., George, B. M., Gillich, A., Gu, X., Gulati, G. S., Hang, Y., Huang, A., Iram, T., Isobe, T., Karnam, G., Kershner, A. M., Kiss, B. M., Kong, W., Kuo, C. S., Lam, J. Y., Lehallier, B., Li, G., Li, Q., Liu, L., Lu, W.-J., Min, D., Nabhan, A. N., Ng, K. M., Nguyen, P. K., Patkar, R., Peng, W. C., Penland, L., Rulifson, E. J., Schaum, N., Sikandar, S. S., Sinha, R., Szade, K., Tan, S. Y., Tellez, K., Travaglini, K. J., Tropini, C., van Weele, L. J., Wang, B. M., Wosczyna, M. N., Xiang, J., Yousef, H., Zhou, L., Batson, J., Botvinnik, O., Chen, S., Darmanis, S., Green, F., May, A. P., Maynard, A., Pisco, A. O., Quake, S. R., Schaum, N., Stanley, G. M., Webber, J. T., Wyss-Coray, T., Zanini, F., Beachy, P. A., Chan, C. K. F., de Morree, A., George, B. M., Gulati, G. S., Hang, Y., Huang, K. C., Iram, T., Isobe, T., Kershner, A. M., Kiss, B. M., Kong, W., Li, G., Li, Q., Liu, L., Lu, W.-J., Nabhan, A. N., Ng, K. M., Nguyen, P. K., Peng, W. C., Rulifson, E. J., Schaum, N., Sikandar, S. S., Sinha, R., Szade, K., Travaglini, K. J., Tropini, C., Wang, B. M., Weinberg, K., Wosczyna, M. N., Wu, S. M., Yousef, H., Barres, B. A., Beachy, P. A., Chan, C. K. F., Clarke, M. F., Darmanis, S., Huang, K. C., Karkanias, J., Kim, S. K., Krasnow, M. A., Kumar, M. E., Kuo, C. S., May, A. P., Metzger, R. J., Neff, N. F., Nusse, R., Nguyen, P. K., Rando, T. A., Sonnenburg, J., Wang, B. M.,

Weinberg, K., Weissman, I. L., Wu, S. M., Quake, S. R., Wyss-Coray, T., The Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection and processing, Library preparation and sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group, and Principal investigators (2018). Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562(7727):367–372. Number: 7727 Publisher: Nature Publishing Group.

[Segal et al., 2003] Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176. Number: 2 Publisher: Nature Publishing Group.

[Sender et al., 2016] Sender, R., Fuchs, S., and Milo, R. (2016). Revised Estimates for the Number of Human and Bacteria Cells in the Body. *PLOS Biology*, 14(8):e1002533. Publisher: Public Library of Science.

[Seninge et al., 2021] Seninge, L., Anastopoulos, I., Ding, H., and Stuart, J. (2021). VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature Communications*, 12(1):5684. Number: 1 Publisher: Nature Publishing Group.

[Shekhar et al., 2016] Shekhar, K., Lapan, S. W., Whitney, I. E., Tran, N. M., Macosko, E. Z., Kowalczyk, M., Adiconis, X., Levin, J. Z., Nemesh, J., Goldman, M., McCarroll, S. A., Cepko, C. L., Regev, A., and Sanes, J. R. (2016). Comprehensive Classification of Retinal Bipolar Neurons by Single-Cell Transcriptomics. *Cell*, 166(5):1308–1323.e30. Publisher: Elsevier.

[Sokolov et al., 2016] Sokolov, A., Carlin, D. E., Paull, E. O., Baertsch, R., and Stuart, J. M. (2016). Pathway-Based Genomics Prediction using Generalized Elastic Net. *PLOS Computational Biology*, 12(3):e1004790. Publisher: Public Library of Science.

[Speir et al., 2021] Speir, M. L., Bhaduri, A., Markov, N. S., Moreno, P., Nowakowski, T. J., Papatheodorou, I., Pollen, A. A., Raney, B. J., Seninge, L., Kent, W. J., and Haeussler, M. (2021). UCSC Cell Browser: visualize your single-cell data. *Bioinformatics*, 37(23):4578–4580.

[Stuart et al., 2003] Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science*, 302(5643):249–255. Publisher: American Association for the Advancement of Science Section: Research Article.

[Subramanian et al., 2005] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

[Svensson et al., 2020] Svensson, V., Gayoso, A., Yosef, N., and Pachter, L. (2020). Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics*, 36(11):3418–3421. Publisher: Oxford Academic.

[Tang et al., 2009] Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., and Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382. Number: 5 Publisher: Nature Publishing Group.

[Tung et al., 2017] Tung, P.-Y., Blischak, J. D., Hsiao, C. J., Knowles, D. A., Burnett, J. E., Pritchard, J. K., and Gilad, Y. (2017). Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports*, 7(1):39921. Number: 1 Publisher: Nature Publishing Group.

[van Waveren and Moraes, 2008] van Waveren, C. and Moraes, C. T. (2008). Transcriptional co-expression and co-regulation of genes coding for components of the oxidative phosphorylation system. *BMC Genomics*, 9(1):18.

[Wang et al., 2019] Wang, T., Li, B., Nelson, C. E., and Nabavi, S. (2019). Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinformatics*, 20(1):40.

[Wolf et al., 2018] Wolf, F. A., Angerer, P., and Theis, F. J. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15.

[Yeo et al., 2021] Yeo, G. H. T., Saksena, S. D., and Gifford, D. K. (2021). Generative modeling of single-cell time series with PRESCIENT enables prediction of cell trajectories with interventions. *Nature Communications*, 12(1):3222. Number: 1 Publisher: Nature Publishing Group.

[Yuan et al., 2021] Yuan, B., Shen, C., Luna, A., Korkut, A., Marks, D. S., Ingraham, J., and Sander, C. (2021). CellBox: Interpretable Machine Learning for Perturbation Biology with Application to the Design of Cancer Combination Therapy. *Cell Systems*, 12(2):128–140.e4.

[Yuan and Lin, 2007] Yuan, M. and Lin, Y. (2007). Model Selection and Estimation in the Gaussian Graphical Model. *Biometrika*, 94(1):19–35. Publisher: [Oxford University Press, Biometrika Trust].

[Zappia et al., 2017] Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, 18(1):174.

[Zhang et al., 2019a] Zhang, A. W., O'Flanagan, C., Chavez, E. A., Lim, J. L. P., Ceglia, N., McPherson, A., Wiens, M., Walters, P., Chan, T., Hewitson, B., Lai, D., Mottok, A., Sarkozy, C., Chong, L., Aoki, T., Wang, X., Weng, A. P., McAlpine, J. N., Aparicio, S., Steidl, C., Campbell, K. R., and Shah, S. P. (2019a). Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nature Methods*, 16(10):1007–1015. Number: 10 Publisher: Nature Publishing Group.

[Zhang and Zhang, 2013] Zhang, J. and Zhang, S. (2013). Modular Organization of Gene Regulatory Networks. In Dubitzky, W., Wolkenhauer, O., Cho, K.-H., and Yokota, H., editors, *Encyclopedia of Systems Biology*, pages 1437–1441. Springer, New York, NY.

[Zhang et al., 2019b] Zhang, X., Lan, Y., Xu, J., Quan, F., Zhao, E., Deng, C., Luo, T., Xu, L., Liao, G., Yan, M., Ping, Y., Li, F., Shi, A., Bai, J., Zhao, T., Li, X., and Xiao, Y. (2019b). CellMarker: a manually curated resource of cell markers in human and mouse. *Nucleic Acids Research*, 47(D1):D721–D728.

[Zheng et al., 2017] Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J., and Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):14049. Number: 1 Publisher: Nature Publishing Group.