Modeling single-cell perturbations using deep learning

Mohammad Lotfollahi

September 2021

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Life Sciences

Modeling single-cell perturbations using deep learning

Mohammad Lotfollahi

Vollständiger Abdruck der von der TUM School of Life Sciences der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Mathias Wilhelm

Prüfer*innen der Dissertation:

1. Prof. Dr. Dr. Fabian J. Theis
2. Prof. Dr. Dana Pe'er
3. Prof. Dr. Maria Colomé-Tatché

Die Dissertation wurde am 25.10.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Life Sciences am 15.03.2022 angenommen.

Acknowledgments

I would like to thank Fabian Theis for teaching me how to research and find exciting problems to make beautiful figures. His trust and mentorship led me to learn leadership, management, and supervision skills. I appreciate all his efforts and support for me as an immigrant researcher. It was simply not possible without your support and mentorship. Arigatōgozaimashita Fabian Sensei!

I am also grateful for Anna Sacher's support for all her efforts and kindness in helping through the complicated non-linear manifolds of VISA and contracts. I am also thankful to Sabine Kunz for all her non-stop support in almost everything during these four years.

I would like to thank all my other collaborators and mentors from whom I also learned a lot, specifically David Lopez-Paz, Sasha Misharin, and Alex Wolf.

I would also express my gratitude to all members of Theis lab for being such incredible, supportive, and intelligent people. Specifically, my office mates Volker, Marius, and Subarna for four years of joyful moments we had together. I would also thank my dear friends Adriana, David Muramatsu, Farzin, Setareh (Jalali-iii), Hamid, Babak, Nina, Ramin. You have helped me a lot, and I am not sure how to thank you for everything you have done for me.

I have to thank Soroor, Simon, Loen, Yang, and Nacho for proofreading my thesis. I also appreciate Maren's help to proofread the German abstract of the thesis.

Finally, I would like to thank my baba, Maman, and malmali (khva Ghazad) for their unconditional support and Monir e khoshgele man, the love of my life, for keeping my keshtie be gel nesheste always balanced!

Abstract

Single-cell genomics has revolutionized the understanding of heterogeneity in both health and disease and empowered profiling millions of cells across different tissues to construct *reference atlases*. The ultimate goal of a single-cell reference atlas is to facilitate understanding cellular perturbations by comparing them to a healthy reference. Perturbation is defined as any intervention changing the cell state from normal to perturbed state. The intervention can be caused by disease or a treatment such as drugs. In this cumulative thesis, the goal was to develop deep learning algorithms to analyze single-cell perturbation studies.

To pursue this, we first need to map the newly acquired datasets (i.e., query) like perturbation studies into healthy reference atlases built by consortia such as Human Cell Atlas (HCA). However, the usability of public reference atlases to analyze the query data is hindered by technical variations between the query and reference atlas, computational complexities, resource limitations, and raw data sharing policies. To address these issues, I developed a deep learning algorithm called single-cell architecture surgery (scArches). scArches allows fast, efficient, and accurate integration of perturbation datasets into the reference atlas while preserving the perturbation heterogeneity enabling the discovery of novel cell states.

The integration of the perturbation dataset into the reference atlas transforms it into a *perturbation atlas*. Yet, the space of potential outcomes is vast and experimentally infeasible to measure all possible perturbations such as drugs or gene knockouts. Therefore, computational tools are required to predict the response to the stimuli for unseen phenomena not observed in the initial atlas for an efficient experimental design and novel biological discovery. This motivates the second aim of this thesis, which is to design models to predict the transcriptomic responses to a perturbation at the single-cell level. To accomplish this goal, I developed deep learning algorithms to learn and predict perturbation responses. These methods demonstrated the ability to predict the response to drugs, genetic knock-outs, and diseases. I envision the strategies presented in this thesis would facilitate efficient experimental design and thus hypothesis generation using single-cell genomics.

Zusammenfassung

Die Einzelzellgenomik hat das Verständnis der Heterogenität in Gesundheit und Krankheit revolutioniert und die Erstellung von Profilen von Millionen von Zellen in verschiedenen Geweben ermöglicht, um Referenzatlanten zu erstellen. Das ultimative Ziel eines Einzelzell-Referenzatlasses ist es, das Verständnis zellulärer Störungen zu erleichtern, indem sie mit einer gesunden Referenz verglichen werden. Als Störung wird jeder Stimulus definiert, der den Zellzustand von einem normalen zu einem gestörten Zustand verändert. Die Störung kann durch eine Krankheit oder eine Behandlung, z. B. durch Medikamente, verursacht werden. In dieser kumulativen Arbeit war das Ziel, Deep-Learning-Algorithmen zur Analyse von Einzelzell-Perturbationsstudien zu entwickeln.

Um dies zu erreichen, müssen wir zunächst die neu erworbenen Datensätze, wie z. B. Perturbationsstudien, in gesunde Referenzatlanten, die von Konsortien wie dem Human Cell Atlas (HCA) erstellt wurden, integrieren. Die Verwendbarkeit öffentlicher Referenzatlanten für die Analyse neuer Abfragedaten wird jedoch durch technische Unterschiede zwischen Abfrage und Referenzatlas, rechnerische Komplexität, Ressourcenbeschränkungen und Richtlinien zur Freigabe von Rohdaten behindert. Um diese Probleme zu lösen, habe ich einen Deep-Learning-Algorithmus namens Single-Cell Architecture Surgery (scArches) entwickelt. scArches ermöglicht eine schnelle, effiziente und genaue Integration von Störungsdatensätzen in den Referenzatlas, wobei die Heterogenität der Störung erhalten bleibt und die Entdeckung neuer zellulärer Zustände ermöglicht wird.

Die Integration von Störungsdatensätzen in den Referenzatlas verwandelt diesen in einen *Störungsatlas*. Der Raum möglicher Ergebnisse ist jedoch immens und es ist experimentell nicht realisierbar, alle möglichen Störungen wie Medikamente oder Gen-Knockouts zu messen. Daher werden computergestützte Werkzeuge benötigt, um die Reaktion auf die Stimuli für unbekannte Phänomene vorherzusagen, die im ursprünglichen Atlas nicht beobachtet wurden. Dadurch werden ein effizientes experimentelles Design und neue biologische Entdeckungen ermöglicht. Dies motiviert das zweite Ziel dieser Arbeit, nämlich die Entwicklung von Modellen zur Vorhersage der transkriptomischen Reaktionen auf eine Störung auf Einzelzellebene. Um dieses Ziel zu erreichen, habe ich Deep-Learning-

Algorithmen entwickelt, um die Reaktionen auf eine Störung zu erlernen und vorherzusagen. Diese Methoden haben gezeigt, dass sie in der Lage sind, die Reaktion auf Medikamente, genetische Knock-outs und Krankheiten vorherzusagen. Ich postuliere, dass die in dieser Arbeit vorgestellten Strategien ein effizientes experimentelles Design und damit Hypothesengenerierung mit Hilfe der Einzelzellgenomik erleichtern werden.

List of contributed articles

Publication in the context of my doctoral thesis

(i) Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Ziga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin, Fabian J. Theis. **"Mapping single-cell data to reference atlases by transfer learning."** Nature Biotechnology, August, 1–10 (2021).

(ii) Yuge Ji, **Mohammad Lotfollahi**, F. Alexander Wolf, and Fabian J. Theis. **"Machine learning for perturbational single-cell omics."** Cell Systems 12 (6) (2021): 522–37.

(iii)Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. **"scGen predicts single-cell perturbation responses."** Nature methods 16.8 (2019): 715-721.

(iv) Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J. Theis, and F. Alexander Wolf. **Conditional out-of-distribution generation for unpaired data using transfer VAE.** Bioinformatics 36, no. $\text{Supplement}_2(2020)$: $i610 - i617$.

(v) Mohammad Lotfollahi*, Anna Klimovskaia*, Carlo De Donno, Yuge Ji, Ignacio L. Ibarra, F. Alexander Wolf, Nafissa Yakubova, Fabian J. Theis, and David Lopez-Paz. **Learning interpretable cellular responses to complex perturbations in high-throughput screens.** BioRxiv (2021) (in-revision in Nature Biotechnology).

Further publications

I also contributed in further collaborations during my PhD studies leading to the following research papers:

(vi) Squidpy: a scalable framework for spatial single cell analysis. Giovanni Palla, Hannah Spitzer, Michal Klein, David Fischer, Anna Christina Schaar, Louis Benedikt Kuemmerle, Sergei Rybakov, Ignacio L Ibarra, Olle

Holmberg, Isaac Virshup, **Mohammad Lotfollahi**, Sabrina Richter, Fabian J Theis. Accepted in Nature Methods (2021).

(vii) Learning interpretable latent autoencoder representations with annotations of feature sets. Sergei Rybakov, **Mohammad Lotfollahi**, Fabian J Theis, F Alexander Wolf. Machine Learning in Computational Biology conference (2020).

(viii) Out-of-distribution prediction with disentangled representations for single-cell RNA sequencing data. Mohammad Lotfollahi*, Leander Dony*, Harshita Agarwala*, Fabian Theis. ICML 2020 Workshop on Computational Biology (WCB) Proceedings Paper.

(ix) Jointly learning T-cell receptor and transcriptomic information to decipher the immune response. Yang An, Felix Drost, Fabian Theis, Benjamin Schubert, **Mohammad Lotfollahi**. ICML 2021 Workshop on Computational Biology (WCB) Proceedings Paper.

(x) Multigrate: single-cell multi-omic data integration. Mohammad Lotfollahi* Anastasia Litinetskaya*, Fabian Theis. ICML 2021 Workshop on Computational Biology (WCB) Proceedings Paper.

(xi) scvi-tools: a library for deep probabilistic analysis of single-cell omics data. Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, Yining Liu, Jules Samaran, Gabriel Misrachi, Achille Nazaret, Oscar Clivio, Chenling Xu, Tal Ashuach, **Mohammad Lotfollahi**, Valentine Svensson, Eduardo da Veiga Beltrame, Carlos Talavera-López, Lior Pachter, Fabian J Theis, Aaron Streets, Michael I Jordan, Jeffrey Regier, Nir Yosef. BioRxiv (2021) (in-revision in Nature Biotechnology)

$*$ = Equal contributions

Contents

Chapter 1

Introduction

In this chapter, the characteristics of the single-cell datasets and existing methods to analyze them are described. I further present the motivation of using machine learning solutions to analyze such data.

1.1 Single-cell sequencing technology

The first usage of single-cell transcriptomics was reported by profiling seven cells [1]. The technology has evolved since then, allowing to profile the gene expression for millions of cells (see **Figure 1.1**)).

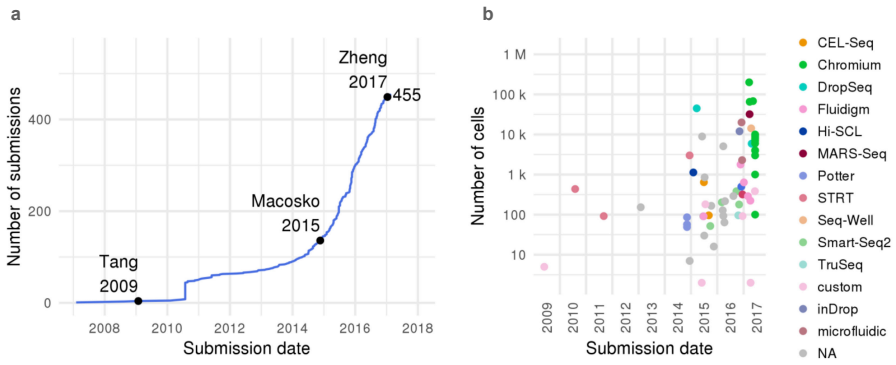


Figure 1.1: **The increasing availability and evolution of single-cell datasets. (a)** The number of submitted articles related to single-cell sequencing. **(b)** The dataset size of submitted articles. The figure is adapted with courtesy from Angerer et al. [2].

The single-cell sequencing pipeline starts by isolating and lysis of cells. Following that step, messenger RNAs (mRNAs) are reverse transcribed into a complementary DNA strand (cDNAs). Next, the second stranded completing cDNA is synthesized and amplified to increase detectable molecules. Finally, the amplified sequences are fed into a sequencing machine and will be mapped to a reference genome (see **Figure 1.2**).

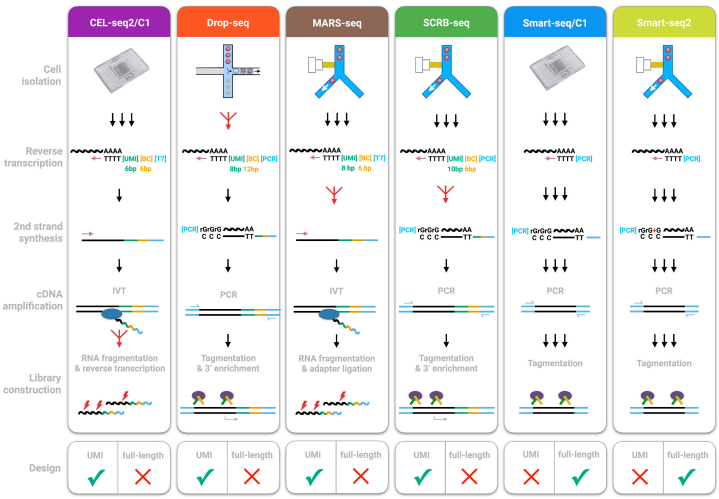


Figure 1.2: **Library preparation steps for different single-cell RNA-seq profiling technologies.** Demonstration of overal scRNA-seq library preparation steps across different methods. The figure is adapted with courtesy from Ziegenhain et al. [3].

An essential step in the outlined pipeline is the unique labeling of amplified sequences to prevent recounting the same molecule multiple times. The unique amplification step is performed using unique molecular identifiers (UMIs) [4] by assigning unique bar codes to detect amplified transcripts originating from the same sequence. The UMI-based methods can tag up to $\approx 10^6$ unique transcript molecules depending on UMI length.

Another critical factor in the scRNA-seq pipeline and data quality is the sensitivity of different methods to detect RNA molecules. In comparison to Bulk sequencing, scRNA-seq requires higher sensitivity to compensate for a lower number of cells for each sample. However, increasing the sensitivity comes with the cost of losing variability leading to sparse or zero reads for most of the genes, also known as dropout [5].

1.2 Single-cell perturbation datasets and their analysis

The advent of single-cell RNA sequencing (scRNA-seq) [6] has enabled analyzing heterogeneity of cells at an unprecedented resolution across tissues [7], species [8], developmental phases, [9] and conditions [10, 11]. An example is HCA, a worldwide consortium aiming to provide a map of all cell-types across different tissues in the Human body. Recently, such technologies have been adapted to conduct large-scale perturbation studies similar to high throughput screens (HTS) performed on bulk samples [12]. Most of these techniques exploit cell barcoding approaches, also known as cellular hashing [13–16]. Cellular hashing allows allocating unique labels to each cell to mix cells from different samples (e.g., perturbations) before scRNA-seq. Such approaches enable profiling millions of cells perturbed with thousands of unique perturbations.

1.2.1 Preprocessing steps for scRNA-seq

Once the scRNA-seq data is obtained, the traditional pipeline [6] of single-cell data analysis can be applied to explore the generated readouts. This pipeline comprises two major steps, starting from pre-processing and continuing to downstream analysis as depicted in **Figure 1.3**. Given the importance of preprocessing steps such as data normalization and quality control, multiple approaches [17–22] have attempted to address finding an optimum solution ranging from linear models [17,21] to neural networks (NN) [20]. However, data integration [6, 23] and correction has been the area of research attracting the most attention from machine learning (ML) and computational biologists. We discuss this area in detail later in the thesis.

Similar to any big data analysis, visualization is an essential step to understand the data better. Thus, an additional step in analyzing single-cell data is using dimension reduction algorithms that are developed for visualizing high-dimensional data. Two widely used methods are t-distributed stochastic neighbor embedding (t-SNE) [24] and UMAP: Uniform Manifold Approximation and Projection [25] and their algorithmic advancements and variations [26–28]. Overall, the goal of these methods is to preserve local similarities between very

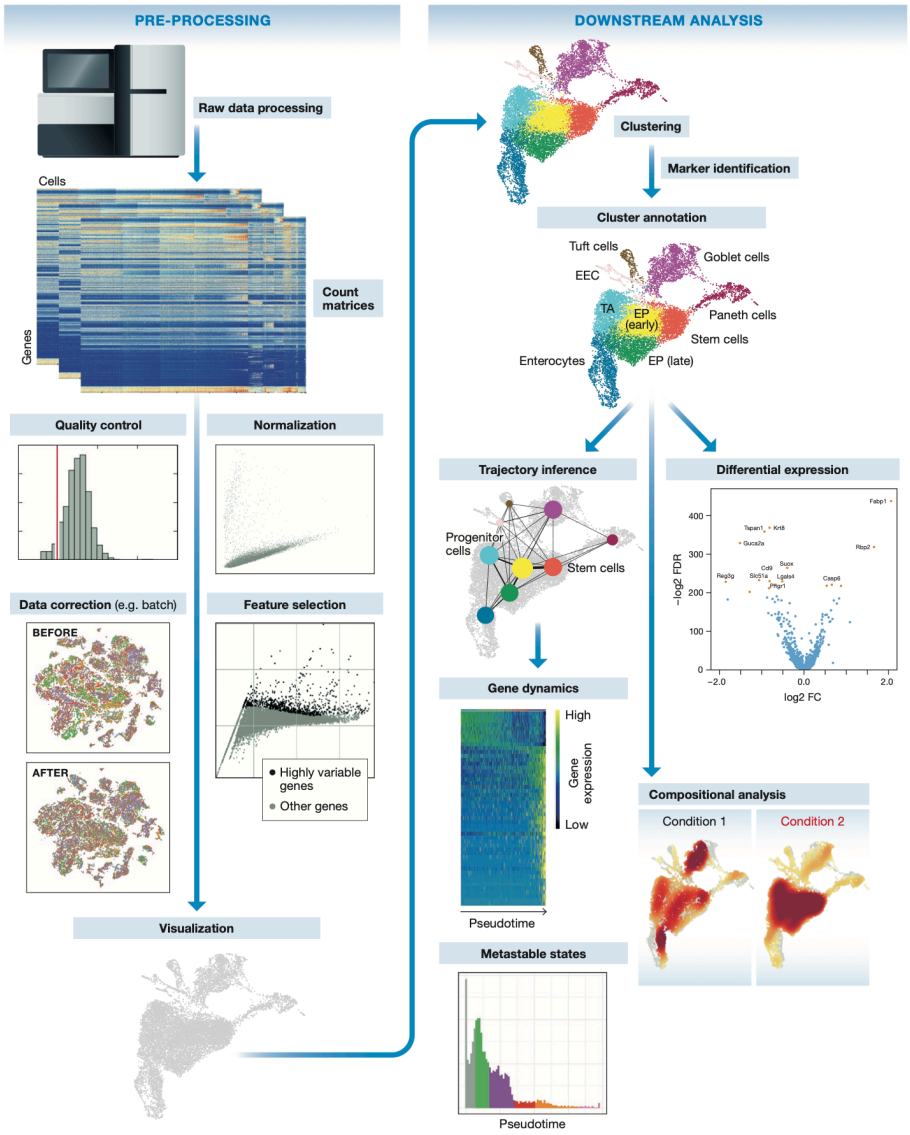


Figure 1.3: **Single-cell data analysis pipeline.** The raw data obtained from sequencing machines are quantified to count matrices to start the preprocessing step. The preprocessed data is used to perform downstream analyses such as clustering, trajectory analysis, differential expression testing, and meta-analysis. Figure adapted with courtesy from Luecken *et al.* [6].

close data points while capturing the global structure of the data. This is important since interpreting global distances between cell clusters could potentially lead to spurious conclusions. This has been recently discussed in a paper by Chari *et al.* [29] examining the loss of biological signals in all-in-one visualiza-

tion methods while proposing an alternative NN-based semi-supervised algorithm leveraging predefined cell-type labels for accelerating hypothesis-driven biological discovery. In addition to Chari *et al.* [29], unsupervised NN-based algorithms leveraging deep generative models (DGM) [20, 30, 31] or hyperbolic geometry inspired [30, 32] methods to capture continuous differentiation's trajectories.

1.2.2 Downstream analysis for scRNA-seq

The next step in single-cell data analysis, which is tightly connected to visualizations, is cell-type annotation and clustering. Clustering coupled with differential testing is a crucial tool to assign cell-type identity. The process usually starts by constructing a K-nearest-neighbour graph (kNN) from the high-dimensional single-cell gene expression. Following that, depending on the choice of the user, an unsupervised graph-based clustering algorithm such as Leiden [6,33] or Louvain [34], which were proposed to detect communities in large graphs such as social networks, is applied. In a single-cell context, those communities are single-cell clusters representing different cell types. Today, these algorithms are widely used and well accepted as the main approach which can quickly scale to datasets with millions of cells [35]. Following the clustering step, differential testing is performed. The test methods vary from a simple t-test [35] to more sophisticated approaches such as Limma [36] and Mast [37] accounting for multiple covariates and advanced design matrices. The outcome of differential testing methods is a set of discriminative genes between clusters found by unsupervised clustering algorithms. The cell-type annotation step is an iterative process by which clustering and differential analysis are repeated to find ever more refined clusters. Thus, such a pipeline can get tedious and time-consuming when the number of cells and data complexity increases. An alternative approach is using a well-annotated cell atlas as a reference to classify new datasets, also know as reference-based cell-type annotation methods [38]. The prerequisite to building a reference atlas is data integration and correction, which is necessary to construct a comprehensive reference atlas powerful enough to annotate a new query dataset. The reference-based analysis is not only limited to cell-type annotation [39,40], but is overall applicable to other scenarios and covers a vast range of

different applications such as disease and perturbation contextualization within a healthy reference atlas [41] or missing modality prediction [41, 42].
In the following section, I discuss the reference-based analysis of single-cell data and the computational problems and challenges around it.

1.3 Reference-based analysis for single-cell genomics

As previously discussed, manual annotation and clustering of single-cell datasets can become very time-consuming. This is more prominent in the presence of large cell numbers, and complex experimental designs such as the whole organism like Tabula Senis-Muris (TSM) [43] for Mice or the Human Cell Landscape (HCL) [8] wherein manual annotation and analysis become impossible. On the other hand, the emergence of large and well-annotated single-cell atlases provides an opportunity to leverage such knowledge to analyze upcoming single-cell data. The idea of reference-based analysis for single-cell genomics is conceptually analogous to the work on DNA reference assembly [44] and also tools for mapping new sequences to the reference genome [44]. The reference-based analysis is comprised of two steps: 1) how to build the reference cell atlas and 2) how to make use of it (**Figure 1.4**). The first step is tightly connected with data integration problems [23] both for uni-modal and multi-modal [45] single-cell datasets. In the following, we review the computational problem for reference building and proposed solutions and later discuss using such a reference.

1.3.1 Data integration to construct reference cell atlases

Single-cell reference atlases [46] have to be sufficiently large and diverse to capture cell-types, developmental trajectories, and other biological events of interest. However, most individual single-cell studies are too limited concerning the size or biological diversity to be used as a comprehensive cell atlas. We need to build the reference atlas from multiple independent studies generated from different laboratories to address this limitation. Since these studies were produced at different times and use other experimental protocols and technologies, measurement differences exist, also known as 'batch effects', which hinder the joint analysis of multiple datasets.

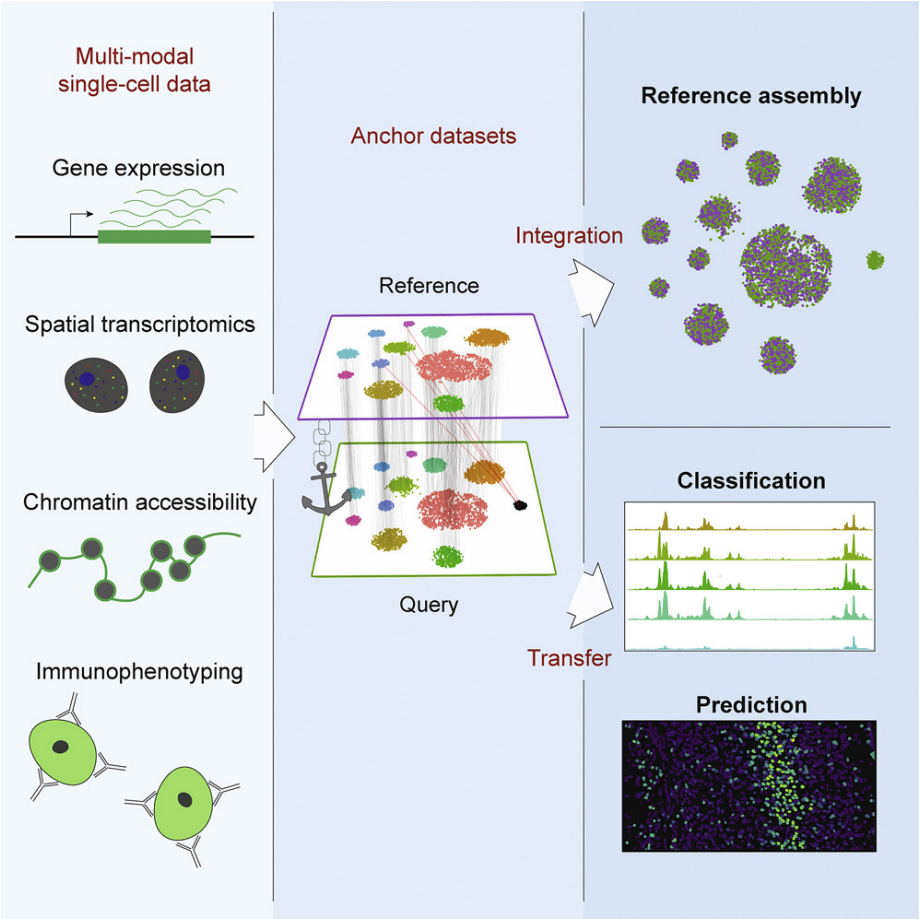


Figure 1.4: **Reference-based analysis for single-cell genomics.** The first step is to construct a shared latent space integrating different modalities from gene expression to immunophenotyping into one reference atlas. The second step is to make the reference atlas usable to analyze a new query dataset for downstream analysis. Figure adapted with courtesy from Stuart *et al.* [39].

An example can be seen in **Figure 1.5** in which we can observe that similar cell-types are separated because they originated from different experimental protocols and are therefore convoluted with batch effects [47–50]. The batch corrected data aligns cell-types from different studies enabling joint analysis of the whole data and constructing the reference atlas.

Since the early emergence of single-cell technologies, batch correction [23, 51] has been the center of attention for computational biologists and computer scientists [52]. The earliest methods leveraged linear models to address the batch effects [53]. While efficient in terms of computational time, linear models

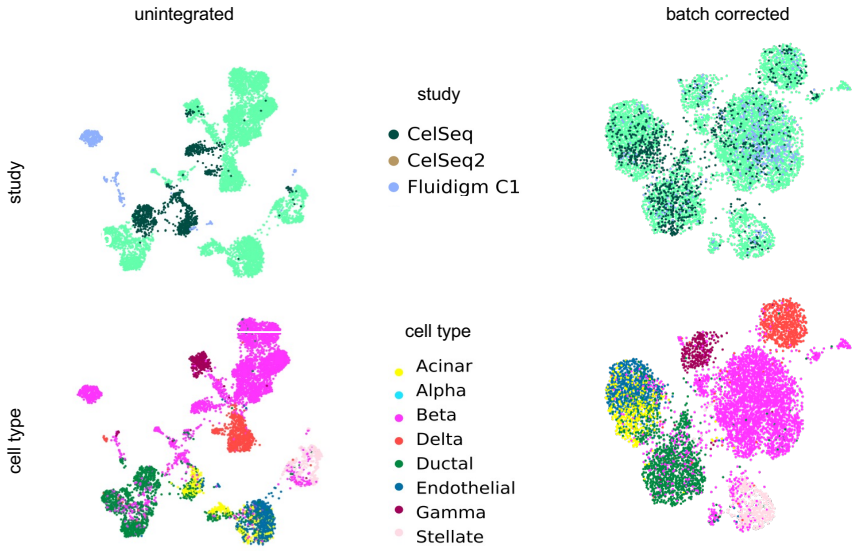


Figure 1.5: **Constructing a Pancreas reference atlas.** UMAP representation of three unintegrated datasets, demonstrating that batch-effects are a major axis of variation if datasets are not properly integrated. The second column shows an integrated representation for the same data where the batch effects have been removed.

were not successful in correcting nested and complex batch effects in complicated studies [23]. This motivated researchers to develop more sophisticated methods. The most broadly used methods in the single-cell community can be categorized into two broad categories, mutual nearest neighbors (MNNs) based methods and NN based models. The MNN based methods first introduced by Haghverdi [54] and its variations [39, 55, 56] aim to find cells having similar gene expression profiles across different batches assumed to represent a unique cell-type or population. MNN based methods assume that the batch effect and biological signals are almost orthogonal [54]. Thus, they can calculate the difference between MNNs in high-dimensional gene expression space and subtract those vectors from nearest neighbors to a set of MNNs. MNNs based methods have been shown to effectively remove batch effects [23] while scaling to hundreds of thousands of cells [55, 56].

The second line of methods for batch effect correction tries to harness the power

of NNs to learn a low-dimensional latent representation [57]. The most promising NN-based data integration method leverages conditional variational autoencoders (CVAE) [58] branded as single-cell variational inference (scVI) [20] and its variations [59, 60]. These models work by learning a conditional distribution of gene-expression conditioned of experimental variable (batch labels). The learned representation would be free of batch variations, and thus similar cell-types from different studies will be aligned. These models are pretty flexible in the sense that they can count for both continuous or discrete covariates. While NN-based models have been shown to be among top-performers [6, 51], they also scale linearly with the number of cells [23] making it ideal for larger datasets. These models have been extended to multi-modal datasets to correct the batch effect between studies while integrating multiple modalities. An example is totalVI [61] which integrates single-cell RNA expression and surface proteins, also known as CITE-seq [62]. Other examples have also demonstrated the integration of transcriptome with open chromatin accessibility measurements (e.g. Assay for Transposase-Accessible Chromatin using sequencing (ATAC-seq)) [63, 64], or T-cell receptor sequences [65].

While all previously outlined methods have been proven accurate according to a few metrics, data integration is, in essence, a trade-off between mixing studies and preserving biological variations. A recent benchmarking study [23] to evaluate the data integration method showed that some strategies are only optimizing for a few scores while being worse in other metrics. Therefore, it is crucial to have a comprehensive set of metrics to evaluate integration quality.

Next, I discuss how we can use that to analyze new query data once we have an integrated reference atlas. This will introduce the next computational problem and second step required for reference-based analysis for single-cell genomics.

1.3.2 Using integrated reference atlases

Assuming we have an integrated reference atlas using one of the above approaches, the next challenge is how to leverage this to analyze a completely new single-cell dataset. The first naive approach is to rerun the data integration algorithm by combining the reference datasets, and the new dataset referred to as "query" hereafter. However, this will become computationally expensive

as newer datasets arrive in time, prohibiting further usage of such a model. Additionally, every new dataset requires extra hyperparameter tuning requiring expertise in machine learning and algorithms.

To address these problems and further democratizing the usage of reference atlases, new approaches such as single-cell architecture surgery (scArches) [41] or weighted-nearest neighbor (WNN) [66] were developed. Both WNN and scArches build upon a class of integration algorithms that were previously published. Therefore they do not propose a new integration algorithm rather a way to integrate new query data at a minimal computational cost without rerunning integration algorithms. We refer to such methods as "reference mapping" methods.

Reference mapping allows users to rapidly integrate their data into reference atlases and efficiently annotate the query datasets. An example is Azimuth [1], a web application that uses reference mapping to automate the processing, analysis, and interpretation of a new single-cell dataset.

Analyzing single-cell perturbation using reference mapping

The ultimate goal of large consortia such as the Human Cell Atlas is to understand cellular functions in healthy and disease states. To understand the mechanism of disease as a perturbation, one needs to build a healthy comprehensive reference atlas. Once a healthy reference atlas has been constructed, we can query and compare perturbed states like disease or specific treatments. Successful integration of a new cohort including both healthy and perturbed states on the top of a healthy reference atlas has to satisfy three criteria: (1) the biological variations of the healthy cells has to be preserved, (2) integration of matching cell-types between healthy and perturbed states and, (3) preserving novel disease states not present in the healthy reference. Therefore, once those three criteria are met, the healthy reference atlas will be updated with the disease states, referred to as "perturbation atlas".

Running de-novo data integration methods to integrate healthy and disease states can potentially remove disease variations by mixing these states. Further in the thesis, we outline scArches [41] as a contribution of this doctoral thesis

[1] `https://satijalab.org/azimuth/`

allowing reference-based analysis of perturbation data.

In the next section, I will discuss that once the perturbation atlas is built, we can leverage it for predictive tasks such as predicting the response of a single-cells to specific stimuli for phenomena which were not observed in the perturbation atlas also known as "out-of-distribution" (OOD).

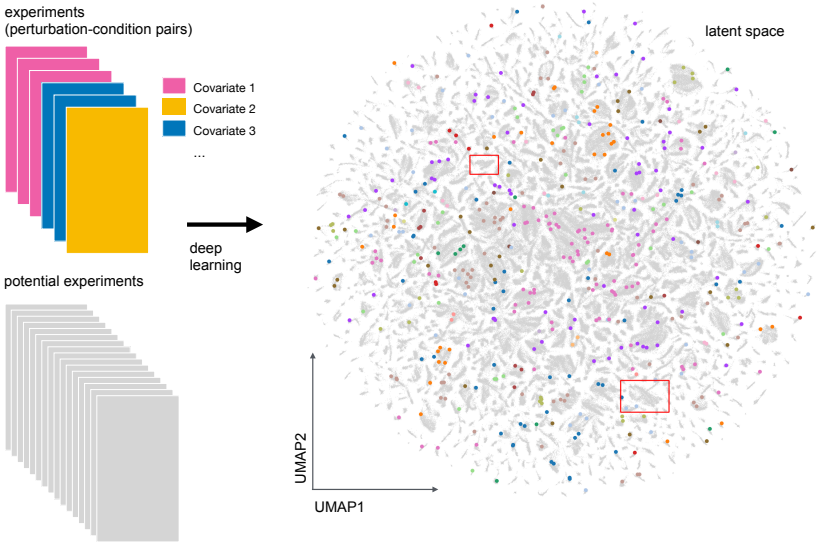


Figure 1.6: **Example schematic of a perturbation atlas.** Perturbation atlases are constructed from multiple perturbations and covariates. Colored dots are existing perturbation experiments and may be connected to other dots via covariates such as cell-types or other labels such as dose or time. Red squares show regions in the perturbation space where no real experiment exists or is sparse. Figure adapted with courtesy from Ji, Lotfollahi *et al.* [67]

1.4 Single-cell perturbation response modeling

Single-cell perturbation atlases allow modeling and predicting of perturbation effects including cellular response to stimuli such as drugs, and the impact of genetic knock-downs [67]. Perturbation experiments try to stimulate the basal (i.e., control) cells with a set of stimuli. An example of single-cell perturbation is a recent work by Sirverstan *et al.* [15] on massively multiplex HTS at single-

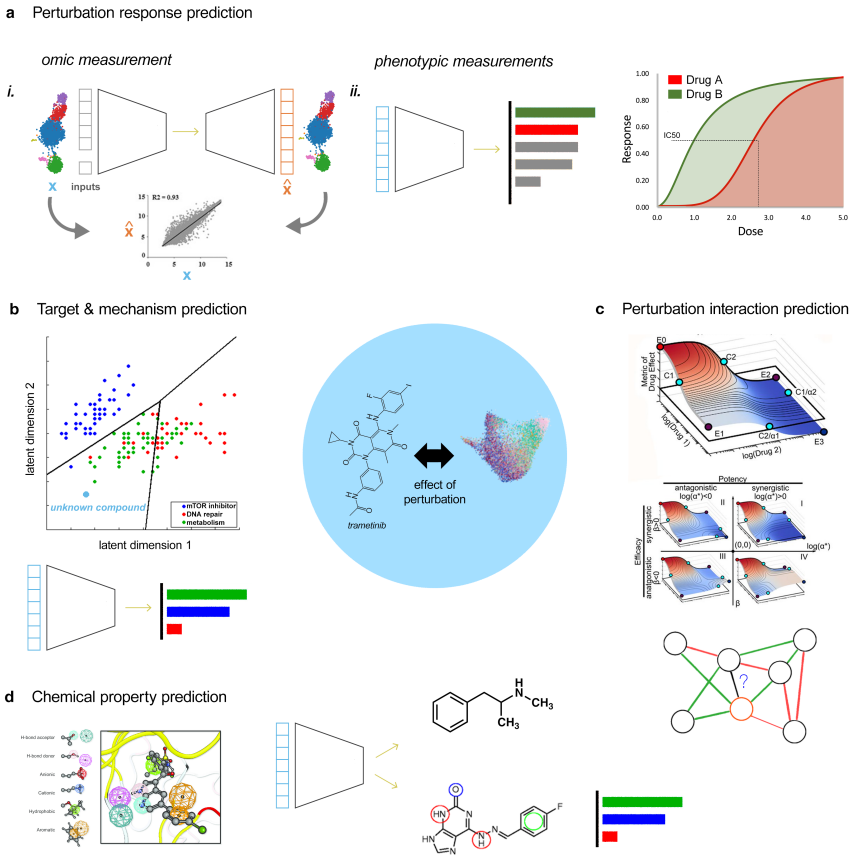


Figure 1.7: **Perturbation modeling tasks and aims.** **(a)** Perturbation response prediction task: (i) predicting the missing omic measurements in which the input is the unperturbed controls and the model is asked to predict the perturbed version of that measurements. (ii) Predicting phenotypic low-dimensional responses such as IC_{50}, area under the curve (AUC). **(b)** The model is given an omic measurement, and the task is to predict a proxy for the compound targets like the mechanism of action (MoA) even for the compound not seen during the training. **(c)** Predicting the combinatorial perturbation effects to elucidate the mechanism of interactions such as synergistic or antagonistic. **(d)** Given an omic measurement, the model has to predict the chemical properties contained in the compound. The figure is adapted with courtesy from Ji, Lotfollahi *et al.* [67]

cell resolution. The authors evaluated the effect of 188 compounds in different dosages across three cancer cell lines. Overall, 5,000 conditions across $\approx$ 0.5M cells were profiled, making it one of the most extensive perturbation experiments at the single-cell level. However, large-scale drug discovery efforts equivalent to

studies like connectivity map [68] are not possible even at single-modality due to the extensive cost of such experiments. Therefore, computational approaches are required to explore perturbation space allowing to aid experiment design and predict unseen or sparsely sampled regions (**Figure 1.6**).
The generation of a perturbation atlas that includes a wide variety of different cell-states will be a significant step towards understanding fundamental biology and drug discovery. In addition, the scale and complexities of the perturbation atlas make them ideal for training deep learning (DL) models, a category of machine learning algorithms using NNs, to solve multiple downstream tasks, outlined in the following section.

1.4.1 Perturbation response modeling

Perturbation modeling objectives can be categorized into four global categories formulated as individual tasks to be solved and benchmarked using ML algorithm. These tasks include reconstructing and quantifying cellular responses and predicting targets, interactions, and chemical properties for drugs and genetic perturbations. [67] (**Figure** 1.7).

1.5 The aims of the thesis

The complexity and scale of perturbation atlases require the development of methods capable of analyzing such data. Within the scope of this cumulative thesis, the following aims were pursued:

- How to leverage existing published uni-modal or multi-modal single-cell reference atlases to facilitate the analysis of newly generated single-cell datasets?
- How to integrate a perturbation dataset (e.g., disease) into a healthy reference atlas while preserving the disease heterogeneity?
- How to in-silico predict the response of a single-cell to a specific perturbation while that cell was never measured under that perturbation?

The results presented in this thesis aim to address previous questions using DL. Specifically, we have developed deep learning-powered algorithms resulting in

interpretable and reusable representations. The overview and contributions of this thesis are presented in **Figure 1.8**.

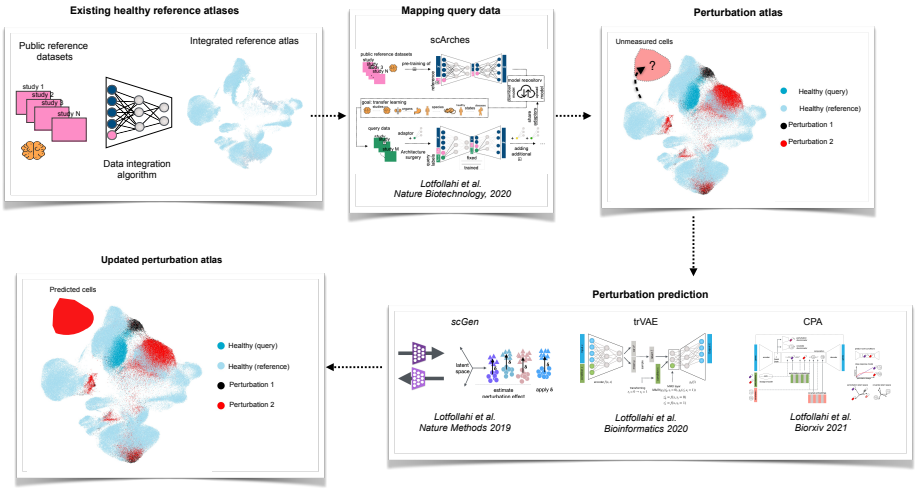


Figure 1.8: **Overview of research questions and contributions of this cumulative doctoral thesis.**

The series of papers presented in this thesis leverage deep representation learning methods to analyze and predict perturbation responses. scArches uses conditional models to integrate novel query studies into reference atlas while preserving unseen perturbation variations. Single-cell generator (scGen) uses generative models and vector arithmetics [69] to perturb control cells towards the desired perturbation. We further proposed an end-to-end alternative for scGen using maximum mean discrepancy (MMD) regularised CVAE called transfer VAE (trVAE) [59] extending scGen to handle multiple perturbations at the same time. Finally, the last contribution of this thesis is to model single-cell perturbations by compositional perturbation autoencoders (CPA) [70]. CPA is an interpretable model that scales to large-scale high-throughput screens (HTS) by explicitly modeling dose, time, and discrete covariates such as cell-types, patients, and species. CPA can also predict combinatorial perturbation effects.

Chapter 2

Materials and Methods

In this section, I overview the fundamental machine learning and deep learning concepts used in this thesis. I first define what does it mean to learn from data. Next, I discuss two types of learning, supervised and unsupervised learning. Furthermore, I review a specific class of unsupervised learning algorithms called autoencoders. Finally, I introduce variational autoencoders, a class of unsupervised methods seeking to estimate data distribution instead of just reconstructing it.

2.1 Supervised learning from data using machine learning

Machine learning is a field of science seeking to find reoccurring patterns in the data tensor X to solve a downstream task T according to criteria p [71]. The data is usually given in the form of (X_{train}, Y_{train}) including of N set of input data (x_i) and labels (y_i). The input data is usually a tensor of real numbers ($x_i \in \mathbb{R}^D$). At the same time, labels could be a tensor of real numbers (regression tasks) or categorical variables (classification tasks). The aim is to learn a function f_w that receives x_i as input and predicts y_i as output. The function f is characterised with set of trainable parameters (weights) w which are optimized using (X_{train}, Y_{train}) in a procedure called training. The objective of the training is to find the best possible parameters using training data to

provide accurate predictions on separate data (X_{test}, Y_{test}) not seen during the training. The trained model is assessed using evaluation criteria p, which assesses the algorithm's accuracy in predicting the target.

The setting described above is also known as *supervised learning*. The parameters (w) of the supervised algorithm f are obtained by defining a differentiable loss function $l(f_w(x_i), y_i)$. The loss function measures the difference between the prediction $f_w(x_i)$ and target y_i. Overall, supervised learning algorithm seeks to minimize the following objection function:

$$J(w) = \sum_{i=1}^{n} l(f_w(x_i), y_i) \tag{2.1}$$

Traditionally, an optimization algorithm such as Gradient descent [71] is employed to find optimal parameters w minimizing the objective function. In the case of neural networks, those parameters are the weights for different layers in the network.

A simple supervised algorithm is logistic regression [72]. Logistic regression network receives an input x_i and applies the following transformation:

$$\hat{y}_i = \sigma(w^T \cdot x_i) \tag{2.2}$$

where w are the weights of single layer network and $\sigma(x) = \frac{1}{1+e^{-x}}$. Thus the $\hat{y}_i$ can be interpreted as the probability of assigning x_i to class 1 in a binary classification problem. Additional hidden layers (i.e., extra weight layers) can be added to a logistic regression algorithm to transform it to a multi-layer neural network. The hidden layers empower the model to solve classification problems not possible to solve using logistic regression (**Figure2.1a-b**).

2.1.1 On the depth and parameter size of deep neural networks

The number of layers in a NN is not limited to two, modern architectures such as residual networks (resNets) [74] have successfully achieved state-of-the-art (SOTA) results in classification tasks with up to 1,000 layers. However, archiving such depth comes with the cost of gradient vanishing [71,75]. Gradient vanishing refers to the gradient signal becoming weaker when it reaches early layers closer

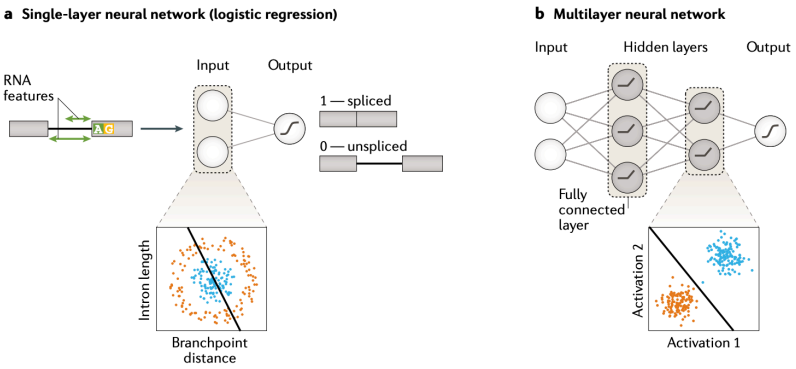


Figure 2.1: **Single and multi-layer neural networks for supervised learning. (a)** A logistic regression algorithm trained on two RNA features to predict if an intron is spliced out (class 1) or unspliced (class 0). The model fails to classify to non-linearly separable classes. **(b)** A trained multi-layer neural network can successfully separate two classes as opposed to the logistic regression. The figure is taken from [73] with courtesy.

to input. Thus, this problem prevents proper training for early layers. The resNet and similar [76,77] architectures solve this problem by proposing a bypass gate known as identity shortcut connection allowing to bypass one or multiple hidden layers by feeding in the input instead of the hidden layer representation.

While deepening the representations could potentially increase the model capacity resulting in better accuracy and performance in downstream tasks, the number of parameters also increases. For example, bidirectional encoder representations from Transformers (BERT) [78], an attention-based [79], SOTA model in natural language processing (NLP) has about 345 million parameters. The model has been demonstrated to perform better on the small downstream task than the smaller 110 million versions of the same model. The size and parameters of such models hinder the usage of such models for user low computational resources.

2.2 Unsupervised learning

Unsupervised learning aims to learn the underlying data structure of the data without direct supervision and labels [71]. Classic examples of unsupervised learning include K-means , singular value decomposition (SVD), PCA, UMAP, and tSNE. All these algorithms seek to discover structures in the form of data clusters or independent axis of variations. NN-based models can find similar structures in the data using specific network architectures called autoencoder (AE). AEs encodes the data into a low-dimensional latent representation called bottleneck and reconstruct the original data from the low-dimensional bottleneck. The reconstruction will not be perfect since the bottleneck layer's dimension is much smaller than the input layer. Therefore, the encoder is encouraged to solely learn essential features of the data, which can be used to reconstruct the output (**Figure 2.2**). In practice, an AE with a single fully connected layer trained with a squared error loss can span a similar subspace as PCA while being not identical to PCA loadings [80].

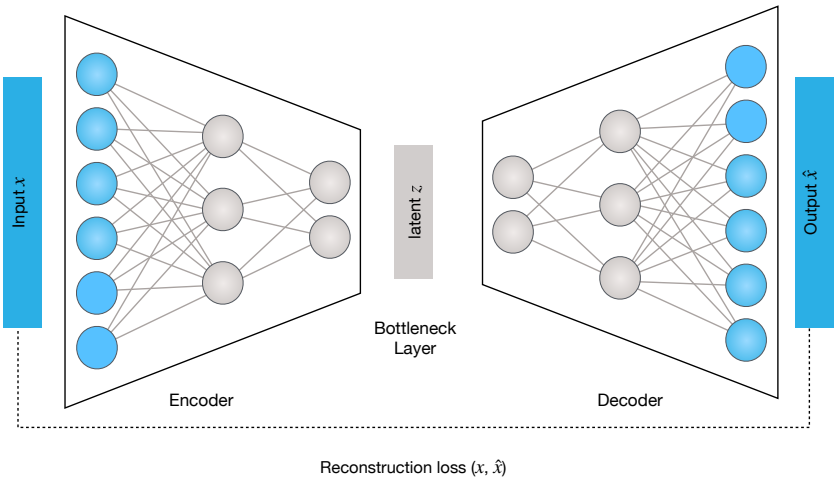


Figure 2.2: **Unsupervised learning using autoencoders.** An AE is comprised of two parts, the encoder and the decoder. The encoder reduces the dimensionality of the data by compressing it into a low-dimensional representation, the so-called bottleneck layer. While the decoder seeks to reconstruct the input from the bottleneck layer. The network's weights are learned to minimize the reconstruction loss between input (x) and the output ($\hat{x}$.)

Stacking multiple non-linear layers in autoencoders can convert them into non-

linear dimensionality reduction methods like tSNE and UMAP. Autoencoders have been exploited for data integration [20, 60], imputation [20, 81], dimensionality reduction [30, 82], out-of-distribution prediction [59, 61, 69, 70]. This framework is also flexible to incorporate prior knowledge [83] in the form of gene-modules and Gene Ontology (GO) terms [84]. This allows interpreting each latent dimension of the bottleneck layer. At the same time, this is not normally feasible in other vanilla autoencoders since the bottleneck layer is a mix of all input features after multiple non-linear functions.

While autoencoders have been shown useful in representation learning, they can only learn a fixed and deterministic function of input, not the distribution of the data. Learning the distribution of the data is essential to measure the likelihood of the data [58, 85], and generate new data points from the learned distribution [86, 87]. Neural networks are now the main drivers in the field of machine learning to learn the data generating process, the so-called generative modeling. The two most popular approaches are variational autoencoders [86], and generative adversarial networks [87]. In the following section, we will describe two widely used variational autoencoder architectures in this thesis, namely, vanilla VAE [86] and conditional VAE [88].

2.3 Generative modeling using variational autoencoders

2.3.1 Probabilistic modeling and Variational Inference

Two significant fields in machine learning are generative modeling and discriminative modeling [85]. The goal of discriminative modeling is to learn a predictor using data observations. In contrast, generative modeling seeks to learn a more general problem of a joint distribution of all variables. Therefore, generative modeling learns the data generation process, which is the central goal in many branches of science. Generative models are a class of probabilistic models where we are interested to learn a joint distribution characterizing all correlations and dependencies between random variables explaining the model.

Formally, Given a vector x, we would like to learn a joint distribution over the

set of all observed random variables. Here and in the following, we adapt the notations and presentation by Kingma *et al.*. Given the data originates from an unknown data-generating distribution $p^*(x)$, we aim to learn an approximate distribution using a model $p_\theta(x)$ with parameters θ such that:

$$x \sim p_\theta(x) \tag{2.3}$$

In this case, the learning algorithm searches in the parameters space to find optimal θ such that the resulting probability distribution learned by the model predicts similar probabilities for each sample similar to $p^*(x)$

$$p_\theta(x) \approx p^*(x) \tag{2.4}$$

The model $p_\theta(x)$ should be flexible and have enough capacity to model the data. In addition, we would like to consider the information we know about the data distribution in learning the model.

In particular, the variational autoencoders (VAEs) have emerged as one of the most popular approaches to unsupervised learning of complicated distributions.

VAEs are generative models that assume the latent space or so-called bottleneck layer is governed by a specific distribution. A VAE comprises two independently parameterized modules called encoder (recognition model) and decoder (generative model). These two models are coupled together and support each other. The encoder model provides an approximation to its posterior estimation for the decoder to update its parameter within an iterative learning framework [85].

Given a data point x, the encoder network parameterized with ϕ learns an approximated posterior distribution $q_\phi(z|x)$ (variational distribution) to estimate the real posterior distribution $p_\theta(z|x)$ such that:

$$q_\phi(z|x) \approx p_\theta(z|x) \tag{2.5}$$

By assuming an isotropic multivariate Gaussian distribution for the posterior distribution, the output of the encoder model is as follows [86]:

$$(\mu, \log \sigma^2) = \text{Encoder}_\phi(x) \tag{2.6}$$

Thus, the encoder outputs a parameterized Normal distribution as follows:

$$q_\phi(z|x) = \mathcal{N}(z; \mu, diag(\sigma^2)) \tag{2.7}$$

To make the backpropagation of the gradient possible during the sampling of the random variable z from $q_\phi(z|x)$, the reparameterization trick is used, which is simply a change of variable and it can be expressed as:

$$z = \mu + \sigma^2 \cdot \epsilon \tag{2.8}$$

where ϵ is sampled from $\mathcal{N}(0, I)$. The overall VAE architecture can be observed in **Figure 2.3**:

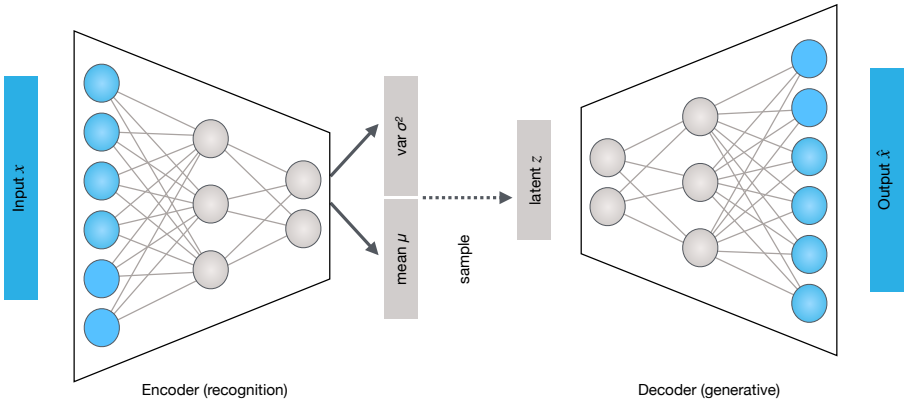


Figure 2.3: **Schematic of a VAE architecture.** VAE is comprised of two independently parameterized yet coupled parts, the encoder (recognition) model and the decoder (generative) model. The encoder model encodes the input data and produces the parameters of a Gaussian distribution. The latent variable z is sampled using the reparameterization trick and is fed as input to the decoder model.

The overall loss function for VAE, the evidence lower bound (ELBO), is comprised of two terms: a log-likelihood (reconstruction loss), and a distributional distance known as Kullback-Leibler (KL) divergence between the variational posterior and true posterior [86]. The ELBO is the lower bound on the marginal log-likelihood of the data [85], therefore maximizing it results in tightening the bound on real marginal likelihood:

$$\mathcal{L}_{VAE}(X; \phi, \theta) = \log p_\theta(X) - \alpha * D_{KL}(q_\phi(Z|X)||p_\theta(Z|X)), \tag{2.9}$$

where $\mathcal{L}_{VAE}(X;\phi,\theta)$ is equal to:

$$\mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X \mid Z)] - \alpha * D_{KL}(q_\phi(Z|X)||p_\theta(Z))$$

(2.10)

where α is the regularization weights to tune the effect of KL loss. The larger values of α increase the stochasticity of the model while the small α drives the model toward deterministic AEs. Furthermore, it has been shown that a higher α encourages a disentangled latent representation [89] also known as β-VAE, where each dimension of the latent space potentially encodes one axis of variation in the data. This means that by changing the value of that dimension, only one high-level feature of the data will change. For example, in the context of single-cell genomics, β-VAEs are helpful to manipulate specific covariates such as perturbations in the data [90, 91].

2.3.2 Non-Gaussian priors for VAE

The original VAE model [86] introduced isotropic Gaussian due to analytical solutions to computing the KL term, and computational efficiency originated from that. However, using Gaussian priors leads to problems related to empty or useless latent dimensions [92]. An example of non-isotropic Gaussian solutions is Variational Mixture of Posteriors (VampPrior) [92]. VampPrior is comprised of a mixture of multiple distributions (e.g., Gaussians) and has been reported to achieve better results than vanilla single Gaussian priors. Another example is imposing inductive biases about the data in the form of prior distributing such as hyperspherical structure [93] or Poincaré ball [94] for hierarchical structures. Similar approaches have been applied to model hierarchical structures in developmental single-cell studies [30, 32]. The benefit from such models is that the pressure to be around a center of mass such as the mean of Gaussian is lifted, and the data could be projected into a uniformly distributed space or a disk. Such projections allow the discovery of cell hierarchies, branched developmental trajectories [30].

Next, we will describe another variation of VAEs, known as conditional VAE (CVAE). CVAEs are broadly used in the context of single-cell genomics [20, 41]. Importantly, they are the main class of models used in the two articles [41, 59] presented in this thesis.

2.4 Conditional variational autoencoders

CVAE is a VAE in which the posterior distribution is conditioned on a single or multiple variables of interest. Mathematically, CVAE loss function is similar to a VAE but modified as following:

$$\mathbb{E}_{q_\phi(Z|X,S)}[\log p_\theta(X \mid Z, S)] - \alpha * D_{KL}(q_\phi(Z|X,S)||p_\theta(Z \mid S)), \tag{2.11}$$

where S is the condition vector for each data point X. The condition vector could be a one-hot encoded vector representing a specific categorical variable in the data. Contextualizing this in the field of single-cell genomics, the condition variables could be experimental protocols, cell-types, or perturbation labels. Once a condition is provided, the latent space of the CVAE would be free from potential variation explained by the conditional variable. This is the basis of all CVAE based data integration methods such as scVI [20] in which the experimental labels such as batch are provided as input of the model; thus, the model aims to learn a latent representation that is batch-free. An example of the conceptual difference in the latent space after applying a VAE on a single-cell dataset compared to a CVAE model used on the same data (**Figure 2.4**).
As observed, the CVAE model provided with condition labels has removed the variation separating the control and the stimulated condition and aligned similar cell-types irrespective of their condition. In contrast, the VAE model preserved the condition variation resulting in two different clusters for each cell-type. The overall architecture schematics for a CVAE model are depicted in **Figure 2.5**. The only practical difference compared to VAE is the concatenation of condition vectors with the input of the encoder and the decoder. However, in some implementations, the condition vector is only concatenated with the decoder input. The condition vector can be both in the form of a one-hot vectored data,

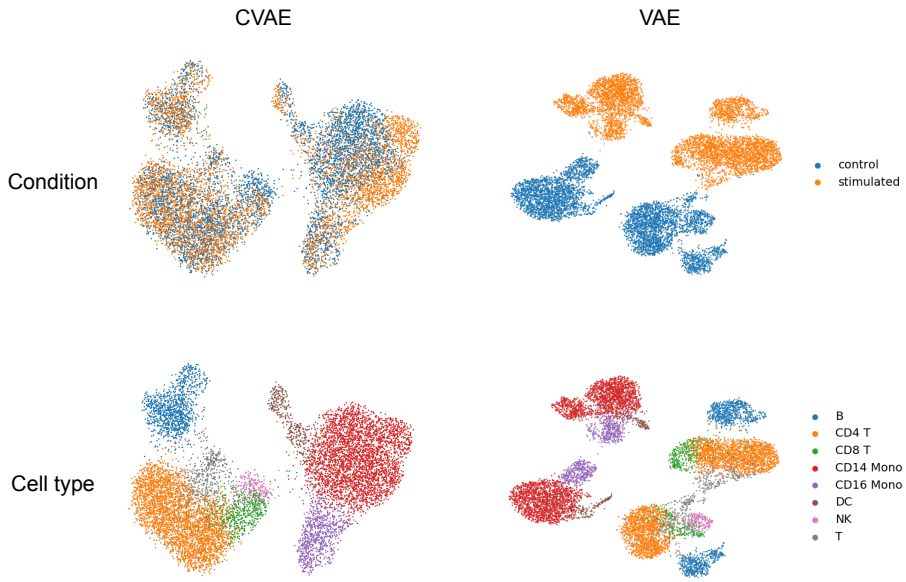


Figure 2.4: **Comparison of VAE and CVAE latent spaces.** The first column demonstrates applying a CVAE on a single-cell dataset [95] while providing the condition labels as the input to the model. The second column shows applying a VAE model on the same data [59]

continuous covariate [20], or representation of another network [59].

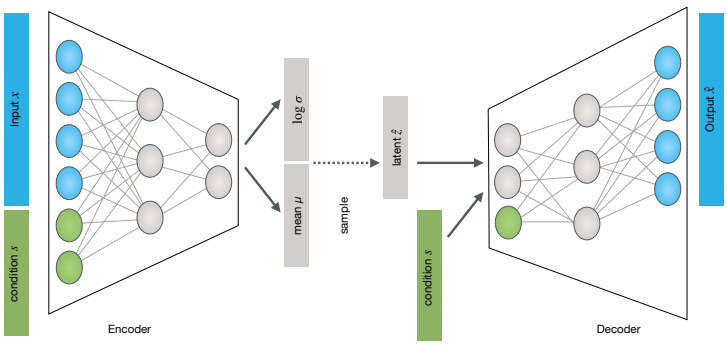


Figure 2.5: **Schematic of a CAVE architecture.** VAE is comprised of two independently parameterized yet coupled parts, the encoder (recognition) model and the decoder (generative) model. The encoder model encodes the input data and the conditions and outputs the parameters of a Gaussian distribution. The latent variable z is sampled using the reparameterization trick, concatenated with condition vector, which is then fed as the input to the decoder model.

Chapter 3

Summary of contributed articles

This chapter provides a summary of my contributed articles as part of my doctoral studies. The publications are sorted according to the sequence outlined in the Introduction section. The first publication addresses the question of building and mapping into single-cell atlases, facilitating the generation of perturbation atlases. The later publications are built upon perturbation atlases to predict single-cell responses to unseen phenomena not observed in the atlases.

(i) Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Ziga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin, Fabian J. Theis. **"mapping single-cell data to reference atlases by transfer learning."** Nature Biotechnology, August, 1–10 (2021).

Summary: In this article, we describe a deep learning framework to enable using large single-cell reference atlases to analyze new query data. While batch-correction methods have partially addressed the integration of multiple datasets, learning from reference atlases requires rerunning computationally expensive algorithms and thus prohibitive for large-scale collaboration due to their centralized nature. We developed single-cell architecture surgery (scArches) to address these problems, democratizing and facilitating the usage of single-cell refer-

ence atlases. scArches build upon existing integration methods using CVAEs [20, 57, 59–61].

Given a reference model, scArches appends new conditional weights corresponding to a new query dataset and optimizes those few weights, called adaptors, while keeping the rest of the network weights frozen. Therefore, the user can download a reference model and quickly integrate a new query dataset without further hyperparameter turning and data sharing. Furthermore, since the internal weights are fixed, the reference representation remains fixed and interpretable. Therefore, the reference can be queried and updated as many times as required. Finally, since the query training optimizes the source of variations originating from technical effects, the perturbation variation such as disease and treatments will be preserved in the query data, making it an ideal approach to analyze perturbation effects.

Contribution: I conceived the project idea with some feedbacks from Ziga Avsec and Fabian Theis and implemented the method together with Mohsen Naghipourfar. I designed all case studies, figure plans, and scenarios. All visualization ideas originated from me, and I developed and applied all necessary steps. Finally, I created and interpreted all result figures and wrote the first complete draft of the publication, which I then finalized based on comments from other collaborators.

(ii) Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. **"scGen predicts single-cell perturbation responses."** Nature Methods 16.8 (2019): 715-721.

Summary: This is the first work in the trilogy for perturbation response modeling. Given a reference perturbation atlas including both control and perturbed cells, the goal was to learn a transformation to transport a control cell to the desired perturbed state. To pursue that, we first discovered a low-dimensional latent representation of the data. Then, we postulated that the non-linear perturbation effect could be modeled in a linear fashion once the data is in the latent space. Thus, we calculated the average difference vector between the desired perturbation and control states. Therefore, the latent vector of each control cell could be transported to the perturbed state by simply adding the

difference vector to its latent vector. This approach was inspired by work done in the field of computer vision [96] and natural language processing [97], in which a difference vector manipulates the representation of a word or an image by simple addition and subtraction operators known as latent space arithmetics (LSA).

We first demonstrated our method outperformed other non-linear approaches such as cycleGAN and CVAE. We further showed that LSA in high-dimensional gene expression or PCA space fails to predict highly non-linear effects like cell-type-specific responses to a perturbation while successfully predicting global responses present in all cell-types. This result also elucidated the necessity of using non-linear methods for this problem.

We demonstrated the generalizability of the method by applying it to predict the response across studies and disease response. Additionally, the technique was successful in predicting the perturbation response across species. Finally, we showed that LSA in the latent space could remove the batch effect given the availability of cell-type labels. Our method ranked among top performers in the recent data integration algorithms benchmark [23] and has been applied for biological discovery by the single-cell community [98].

Finally, we also explored the model's limitations by showing that scGen can not predict completely unseen and new perturbation responses if cells with similar responses are absent from training data.

Contribution: The original idea of using generative models for single-cell genomics and perturbation modeling came from me. Alex Wolf further polished and directed the idea toward specific applications with comments and supervision from Fabian Theis. I performed the research, implemented all methods. I generated all figures and visualization in the paper. I also completed all the data analyses for all datasets in the paper. Finally, I wrote the article's first draft, which Alex Wolf and Fabian Theis later revised.

(iii) Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J. Theis, and F. Alexander Wolf. **"Conditional out-of-distribution generation for unpaired data using transfer VAE."** Bioinformatics 36, no. Supplement_2 (2020): i610-i617.

Summary: This paper aimed to address limitations in scGen. The first limita-

tion was the restriction to model only one perturbation, making it not applicable to more extensive perturbation studies. The second limitation was that the perturbation effects were not learnable and separate from the representation learning step. Thus, the model could not learn all stages from initial input to output results, known as end-to-end training.

To address previous limitations, we formulated perturbation response modeling as a distribution matching problem by learning a transformation to match and thus transport the distribution of control cells to the desired perturbation distribution. The problem is also known as style transfer in the machine learning community [99, 100]. This was achieved by designing a CVAE coupled with a distribution matching regularizer using maximum mean discrepancy (MMD).

We first demonstrated that transfer VAE (trVAE) could manipulate images attributes such as transforming a non-smiling face image to a smiling one while preserving other image attributes. Similarly, we showed that the model could manipulate handwritten digit images by thickening or thinning them while keeping their identity. Next, we applied trVAE to the single-cell perturbation modeling problem. We showed that our model outperforms six other state-of-the-art deep learning methods across two different datasets. Specifically, we showed trVAE also learned cell-type-specific responses and improved the accuracy of predicting most cell-type-specific genes by 65% compared to scGen.

Finally, trVAE allows modeling multiple perturbations simultaneously, enabling applying it to highly multiplexed single-cell experiments. We specifically showed that the model successfully predicted the response for unseen cell-types after Salmonella or Heligmosomoides polygyrus (H. poly) infections.

Contribution: The original idea of an End-to-End model is from me. Mohsen Naghipourfar and I implemented the model. The figure ideas, visualizations, and case studies were all designed by me. I wrote the first draft of the paper with comments and revisions from Alex Wolf and Fabian Theis.

(iv) Mohammad Lotfollahi*, Anna Klimovskaia*, Carlo De Donno, Yuge Ji, Ignacio L. Ibarra, F. Alexander Wolf, Nafissa Yakubova, Fabian J. Theis, and David Lopez-Paz. **"Learning interpretable cellular responses to complex perturbations in high-throughput screens."** bioRxiv (2021).

Summary: In this article, we sought to improve the interpretability of previous perturbation models by modeling data variation as a compositional process in a low-dimensional latent space termed as compositional perturbation autoencoder (CPA). The compositional process decomposes a single-cell into a basal state vector (cell representation free from covariate and perturbation information), covariate vectors (e.g., cell-type, species, patients), and perturbation vector (e.g., disease, drug, genetic knockouts). The perturbation vectors are scaled using non-linear functions applied on continuous covariates such as dose and time. All perturbations and covariate vectors (embeddings) are learned by removing those effects using adversarial training [87] and learning adding them back to reconstruct gene expression effects. The learned embeddings recapitulate the similarity of perturbations in gene expression, useful for drug-repurposing efforts. We specifically showed that our model could infer a latent drug space from a massively multiplexed [101] experiment revealing drug response similarity for 188 different drugs. We additionally showed CPA differentiates responsive and not responsive drugs on cancer cell lines. The results revealed that the dose-response prediction accuracy is robust if the model has seen other dosages of the same drug on that cell line. However, the prediction would deteriorate or fail if the drug was never observed at any dosage in that cell line.

The other primary goal was to predict combinatorial interventions, be it genetic perturbations (e.g., CRISPR experiments) or drug cocktails. The compositional formulation allows learning an arbitrary number of perturbations as additive interventions, thus enabling the prediction of unseen combinations. In particular, we showed CPA could predict all missing genetic combinations from a single-cell perturb-seq experiment while providing uncertainty values for the predicted combination. However, the model fails to predict the combined effect if contributing single perturbations are never seen in combination with other perturbations. This necessitates the careful analysis of the results with strong attention to uncertainty values, preventing spurious conclusions. Finally, CPA is not designed to replace experiments, rather facilitating the design of further experiments.

Contribution: The original idea of designing a model for predicting combinatorial single-cell perturbations is from me. I, David Lopez-Pax, and Anna

Klimovskaia designed the method. Then, I implemented the first version of the method. I researched and selected datasets for case studies in the paper together with Fabian Theis and Anna Klimovskaia. Finally, I wrote the first draft of the paper with contributions from other co-authors.

$*$ = Equal contributions

Chapter 4

Discussion

This chapter will discuss potential future directions and ideas for reference mapping and perturbation modeling methods for single-cell genomics.

4.1 single-cell reference mapping

In this thesis, I have developed algorithms to analyze single-cell reference perturbation atlases. I first discussed that existing integration algorithms are not optimal and secure to integrate novel query datasets into existing reference atlases. Then, I devised single-cell architecture surgery to address those challenges. scArches is a general framework applicable to existing integration algorithms that use CVAEs [88]. Finally, I demonstrated that perturbation datasets could be integrated into a healthy reference converting them to perturbation atlas. In the following, I will discuss further applications and extensions for future research in single-cell reference mapping.

4.1.1 Towards leveraging multi-modal reference atlases

Recent advances in single-cell multi-omics have enabled measuring multiple measurements from the same single-cell [45, 65, 102] providing a holistic view into cellular heterogeneity. I envision a reference mapping method such as scArches would be the canonical way to use reference atlases. However, the existing multi-modal integration methods are imperfect, thus the bottleneck for further

development of multi-modal powered reference mapping methods. Therefore, I believe that reference mapping should be a central goal when designing multi-modal reference integration methods.

4.1.2 Clinical applications of reference mapping using Multi-Instance Learning

The availability of large-scale single-cell multi-omics datasets with hundreds of individual patient samples in both healthy and disease states [103] have enabled the construction of patient-level multi-omic reference atlases. Such reference atlases can be combined with multi-instance learning (MIL) [104–106] to predict disease severity. Additionally, recent attention-based MIL [105] allows attributing the phenotype of interest to a specific cell-type or state that is important to determine the phenotype. Therefore, once the reference atlas has been built with a trained MIL model, it can be deployed in clinical settings for further diagnosis applications.

4.1.3 Querying gene modules against a reference atlas

While reference mapping methods enable querying a completely new query dataset, they do not allow to query of a set of genes or gene modules related in the context of a reference atlas. A recent method [83] sought to partially address this by learning an interpretable latent space, where each latent dimension encodes a specific gene module, therefore allowing to analyze the queries and reference within that specific gene module. However, this problem remains an interesting future direction to pursue.

4.2 Single-cell perturbation response modeling

The existing perturbation atlases are partially complete due to the extensive cost of experiments and technology limitations [101] hindering the profiling of all potential perturbations. This hinders in-depth analysis and application of single-cell technologies in real-world, large-scale drug discovery efforts. The thesis's second contribution and research direction aimed to address this prob-

lem by proposing computational approaches for in silico prediction of unseen perturbation to complete perturbation atlases.

In our work on scGen [69], I demonstrated that VAEs coupled with vector arithmetics could predict single-cell responses across cell-types, studies, and species. Importantly, I demonstrated that scGen could capture cell-type-specific effects for cell-types not present in the dataset. However, this depends on cell-types similar to the ones of interest that I would like to predict. Otherwise, the model will fail to make accurate predictions (see Supplementary figure 7 in [69]). While the differential perturbation vector is calculated linearly, it can predict the cell-type-specific effect. I postulate that this emerges due to the complex nonlinear dimension reduction performed by the VAE.

I further improved scGen by formulating the perturbation response prediction as a distribution matching problem solved by trVAE. I demonstrated that trVAE is a general framework, which works with any data modality ranging from images to single-cell gene expression profiles. I further showed that it could handle multiple perturbations simultaneously while also improving the performance in cell-type-specific prediction. Finally, with CPA, I developed a general interpretable model that incorporates both discrete and continuous covariates. Having large-scale perturbation atlases at hand, CPA can infer a perturbation space capturing the similarity of gene expression response. CPA is also able to predict combinatorial perturbation effects such as genetic knock-out or drug combinations.

In the following, I will discuss future work related to perturbation response modeling.

4.2.1 Combining perturbation modeling with structural molecular information

While current models can predict the effects of small molecules [101], the predictions are limited to drugs that are already included and measured in the dataset, and prediction of out-of-library drugs is not possible. This prohibits the usage of such models for large-scale in-silico screening for potential experimental design or repurposing efforts. To address these problems, such models should be extended to predict the response to out-of-library drugs. This would be possi-

ble by encoding the molecular structure, such as a simplified molecular-input line-entry system (SMILES) using graph neural networks [107]. It would also be interesting to inspect the perturbation space since encoding gene expression response and molecular information would lead to different latent perturbation spaces.

4.2.2 Multi-modal perturbation modeling

The works presented in this thesis were all focused on the prediction at the transcript level. However, two recently published papers [108, 109] have demonstrated the feasibility of combining CRISPR-compatible CITE-seq, which combines pooled CRISPR screens with single-cell mRNA and surface protein profiling. Therefore, a potential future direction would be to adapt the current response prediction models to predict perturbation response at multiple levels. This will require a joint representation learning for all modalities related to reference mapping and integration.

Bibliography

[1] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, *et al.*, "mrna-seq whole-transcriptome analysis of a single cell," *Nature methods*, vol. 6, no. 5, pp. 377–382, 2009.

[2] P. Angerer, L. Simon, S. Tritschler, F. A. Wolf, D. Fischer, and F. J. Theis, "Single cells make big data: New challenges and opportunities in transcriptomics," *Current Opinion in Systems Biology*, vol. 4, pp. 85–91, 2017.

[3] C. Ziegenhain, B. Vieth, S. Parekh, B. Reinius, A. Guillaumet-Adkins, M. Smets, H. Leonhardt, H. Heyn, I. Hellmann, and W. Enard, "Comparative analysis of single-cell rna sequencing methods," *Molecular cell*, vol. 65, no. 4, pp. 631–643, 2017.

[4] S. Islam, A. Zeisel, S. Joost, G. La Manno, P. Zajac, M. Kasper, P. Lönnerberg, and S. Linnarsson, "Quantitative single-cell rna-seq with unique molecular identifiers," *Nature methods*, vol. 11, no. 2, pp. 163–166, 2014.

[5] V. Svensson, K. N. Natarajan, L.-H. Ly, R. J. Miragaia, C. Labalette, I. C. Macaulay, A. Cvejic, and S. A. Teichmann, "Power analysis of single-cell rna-sequencing experiments," *Nature methods*, vol. 14, no. 4, pp. 381–387, 2017.

[6] M. D. Luecken and F. J. Theis, "Current best practices in single-cell RNA-seq analysis: a tutorial," *Molecular Systems Biology*, vol. 15, June 2019. Publisher: John Wiley & Sons, Ltd.

[7] T. M. Consortium *et al.*, "Single-cell transcriptomics of 20 mouse organs creates a tabula muris.," *Nature*, vol. 562, no. 7727, p. 367, 2018.

[8] X. Han, Z. Zhou, L. Fei, H. Sun, R. Wang, Y. Chen, H. Chen, J. Wang, H. Tang, W. Ge, *et al.*, "Construction of a human cell landscape at single-cell level," *Nature*, vol. 581, no. 7808, pp. 303–309, 2020.

[9] N. Almanzar, J. Antony, A. S. Baghel, I. Bakerman, I. Bansal, B. A. Barres, P. A. Beachy, D. Berdnik, B. Bilen, D. Brownfield, C. Cain, C. K. F. Chan, M. B. Chen, M. F. Clarke, S. D. Conley, S. Darmanis, A. Demers, K. Demir, A. de Morree, T. Divita, H. du Bois, H. Ebadi, F. H. Espinoza, M. Fish, Q. Gan, B. M. George, A. Gillich, R. Gòmez-Sjöberg, F. Green, G. Genetiano, X. Gu, G. S. Gulati, O. Hahn, M. S. Haney, Y. Hang, L. Harris, M. He, S. Hosseinzadeh, A. Huang, K. C. Huang, T. Iram, T. Isobe, F. Ives, R. Jones, K. S. Kao, J. Karkanias, G. Karnam, A. Keller, A. M. Kershner, N. Khoury, S. K. Kim, B. M. Kiss, W. Kong, M. A. Krasnow, M. E. Kumar, C. S. Kuo, J. Lam, D. P. Lee, S. E. Lee, B. Lehallier, O. Leventhal, G. Li, Q. Li, L. Liu, A. Lo, W.-J. Lu, M. F. Lugo-Fagundo, A. Manjunath, A. P. May, A. Maynard, A. McGeever, M. McKay, M. W. McNerney, B. Merrill, R. J. Metzger, M. Mignardi, D. Min, A. N. Nabhan, N. F. Neff, K. M. Ng, P. K. Nguyen, J. Noh, R. Nusse, R. Pálovics, R. Patkar, W. C. Peng, L. Penland, A. O. Pisco, K. Pollard, R. Puccinelli, Z. Qi, S. R. Quake, T. A. Rando, E. J. Rulifson, N. Schaum, J. M. Segal, S. S. Sikandar, R. Sinha, R. V. Sit, J. Sonnenburg, D. Staehli, K. Szade, M. Tan, W. Tan, C. Tato, K. Tellez, L. B. T. Dulgeroff, K. J. Travaglini, C. Tropini, M. Tsui, L. Waldburger, B. M. Wang, L. J. van Weele, K. Weinberg, I. L. Weissman, M. N. Wosczyna, S. M. Wu, T. Wyss-Coray, J. Xiang, S. Xue, K. A. Yamauchi, A. C. Yang, L. P. Yerra, J. Youngyunpipatkul, B. Yu, F. Zanini, M. E. Zardeneta, A. Zee, C. Zhao, F. Zhang, H. Zhang, M. J. Zhang, L. Zhou, J. Zou, and The Tabula Muris Consortium, "A single-cell transcriptomic atlas characterizes ageing tissues in the mouse," *Nature*, vol. 583, pp. 590–595, July 2020. Number: 7817 Publisher: Nature Publishing Group.

[10] R. A. Grant, L. Morales-Nebreda, N. S. Markov, S. Swaminathan, M. Querrey, E. R. Guzman, D. A. Abbott, H. K. Donnelly, A. Donayre, I. A. Goldberg, Z. M. Klug, N. Borkowski, Z. Lu, H. Kihshen, Y. Politanska, L. Sichizya, M. Kang, A. Shilatifard, C. Qi, J. W. Lomasney, A. C.

Argento, J. M. Kruser, E. S. Malsin, C. O. Pickens, S. B. Smith, J. M. Walter, A. E. Pawlowski, D. Schneider, P. Nannapaneni, H. Abdala-Valencia, A. Bharat, C. J. Gottardi, G. R. S. Budinger, A. V. Misharin, B. D. Singer, and R. G. Wunderink, "Circuits between infected macrophages and T cells in SARS-CoV-2 pneumonia," *Nature*, pp. 1–8, Jan. 2021. Publisher: Nature Publishing Group.

[11] J.-Y. Zhang, X.-M. Wang, X. Xing, Z. Xu, C. Zhang, J.-W. Song, X. Fan, P. Xia, J.-L. Fu, S.-Y. Wang, R.-N. Xu, X.-P. Dai, L. Shi, L. Huang, T.-J. Jiang, M. Shi, Y. Zhang, A. Zumla, M. Maeurer, F. Bai, and F.-S. Wang, "Single-cell landscape of immunological responses in patients with COVID-19," *Nature Immunology*, pp. 1–12, Aug. 2020. Publisher: Nature Publishing Group.

[12] C. Ye, D. J. Ho, M. Neri, C. Yang, T. Kulkarni, R. Randhawa, M. Henault, N. Mostacci, P. Farmer, S. Renner, *et al.*, "Drug-seq for miniaturized high-throughput transcriptome profiling in drug discovery," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[13] P. Datlinger, A. F. Rendeiro, C. Schmidl, T. Krausgruber, P. Traxler, J. Klughammer, L. C. Schuster, A. Kuchler, D. Alpar, and C. Bock, "Pooled crispr screening with single-cell transcriptome readout," *Nature methods*, vol. 14, no. 3, pp. 297–301, 2017.

[14] P. Datlinger, A. F. Rendeiro, T. Boenke, M. Senekowitsch, T. Krausgruber, D. Barreca, and C. Bock, "Ultra-high-throughput single-cell RNA sequencing and perturbation screening with combinatorial fluidic indexing," *Nature Methods*, pp. 1–8, May 2021. Publisher: Nature Publishing Group.

[15] S. R. Srivatsan, J. L. McFaline-Figueroa, V. Ramani, L. Saunders, J. Cao, J. Packer, H. A. Pliner, D. L. Jackson, R. M. Daza, L. Christiansen, *et al.*, "Massively multiplex chemical transcriptomics at single-cell resolution," *Science*, vol. 367, no. 6473, pp. 45–51, 2020.

[16] J. Gehring, J. Hwee Park, S. Chen, M. Thomson, and L. Pachter, "Highly multiplexed single-cell RNA-seq by DNA oligonucleotide tagging of cellu-

lar proteins," *Nature Biotechnology*, vol. 38, pp. 35–38, Jan. 2020. tex.ids: gehringHighlyMultiplexedSinglecell2020a Number: 1 Publisher: Nature Publishing Group.

[17] C. Hafemeister and R. Satija, "Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression," *Genome biology*, vol. 20, no. 1, pp. 1–15, 2019.

[18] T. S. Andrews, V. Y. Kiselev, D. McCarthy, and M. Hemberg, "Tutorial: guidelines for the computational analysis of single-cell rna sequencing data," *Nature protocols*, vol. 16, no. 1, pp. 1–9, 2021.

[19] R. Bacher, L.-F. Chu, N. Leng, A. P. Gasch, J. A. Thomson, R. M. Stewart, M. Newton, and C. Kendziorski, "Scnorm: robust normalization of single-cell rna-seq data," *Nature methods*, vol. 14, no. 6, pp. 584–586, 2017.

[20] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," *Nature methods*, vol. 15, no. 12, pp. 1053–1058, 2018.

[21] D. Risso, "Normalization of single-cell rna-seq data," in *RNA Bioinformatics*, pp. 303–329, Springer, 2021.

[22] S. Choudhary and R. Satija, "Comparison and evaluation of statistical error models for scrna-seq," *bioRxiv*, 2021.

[23] M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, *et al.*, "Benchmarking atlas-level data integration in single-cell genomics," *BioRxiv*, 2020.

[24] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[25] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[26] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger, "Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data," *Nature methods*, vol. 16, no. 3, pp. 243–245, 2019.

[27] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell, "Dimensionality reduction for visualizing single-cell data using umap," *Nature biotechnology*, vol. 37, no. 1, pp. 38–44, 2019.

[28] S. Canzar *et al.*, "A generalization of t-sne and umap to single-cell multimodal omics," *Genome Biology*, vol. 22, no. 1, pp. 1–9, 2021.

[29] T. Chari, J. Banerjee, and L. Pachter, "The specious art of single-cell genomics," *bioRxiv*, 2021.

[30] J. Ding and A. Regev, "Deep generative model embedding of single-cell rna-seq profiles on hyperspheres and hyperbolic spaces," *Nature communications*, vol. 12, no. 1, pp. 1–17, 2021.

[31] J. M. Graving and I. D. Couzin, "Vae-sne: a deep generative model for simultaneous dimensionality reduction and clustering," *BioRxiv*, 2020.

[32] A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel, "Poincaré maps for analyzing complex hierarchies in single-cell data," *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.

[33] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[34] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[35] F. A. Wolf, P. Angerer, and F. J. Theis, "Scanpy: large-scale single-cell gene expression data analysis," *Genome biology*, vol. 19, no. 1, pp. 1–5, 2018.

[36] C. W. Law, Y. Chen, W. Shi, and G. K. Smyth, "voom: Precision weights unlock linear model analysis tools for rna-seq read counts," *Genome biology*, vol. 15, no. 2, pp. 1–17, 2014.

[37] G. Finak, A. McDavid, M. Yajima, J. Deng, V. Gersuk, A. K. Shalek, C. K. Slichter, H. W. Miller, M. J. McElrath, M. Prlic, *et al.*, "Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data," *Genome biology*, vol. 16, no. 1, pp. 1–13, 2015.

[38] J. T. H. Lee and M. Hemberg, "Supervised clustering for single-cell analysis," *Nature methods*, vol. 16, no. 10, pp. 965–966, 2019.

[39] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, "Comprehensive integration of single-cell data," *Cell*, vol. 177, no. 7, pp. 1888–1902, 2019.

[40] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, *et al.*, "Sc3: consensus clustering of single-cell rna-seq data," *Nature methods*, vol. 14, no. 5, pp. 483–486, 2017.

[41] M. Lotfollahi, M. Naghipourfar, M. D. Luecken, M. Khajavi, M. Büttner, M. Wagenstetter, Ž. Avsec, A. Gayoso, N. Yosef, M. Interlandi, S. Rybakov, A. V. Misharin, and F. J. Theis, "Mapping single-cell data to reference atlases by transfer learning," *Nat. Biotechnol.*, pp. 1–10, Aug. 2021.

[42] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, *et al.*, "Integrated analysis of multimodal single-cell data," *Cell*, 2021.

[43] T. M. Consortium *et al.*, "A single cell transcriptomic atlas characterizes aging tissues in the mouse," *Nature*, vol. 583, no. 7817, p. 590, 2020.

[44] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, *et al.*, "De novo assembly of human genomes with mas-

sively parallel short read sequencing," *Genome research*, vol. 20, no. 2, pp. 265–272, 2010.

[45] S. Teichmann and M. Efremova, "Method of the year 2019: single-cell multimodal omics," *Nat. Methods*, vol. 17, no. 1, 2020.

[46] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, *et al.*, "Science forum: the human cell atlas," *elife*, vol. 6, p. e27041, 2017.

[47] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, and I. Yanai, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure," *Cell Systems*, vol. 3, pp. 346–360.e4, Oct. 2016.

[48] M. J. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. A. Engelse, F. Carlotti, E. J. de Koning, and A. van Oudenaarden, "A single-cell transcriptome atlas of the human pancreas," *Cell Systems*, vol. 3, pp. 385–394.e3, Oct. 2016.

[49] Å. Segerstolpe, A. Palasantza, P. Eliasson, E.-M. Andersson, A.-C. Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. K. Bjursell, D. M. Smith, M. Kasper, C. Ämmälä, and R. Sandberg, "Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes," *Cell Metabolism*, vol. 24, pp. 593–607, Oct. 2016.

[50] N. Lawlor, J. George, M. Bolisetty, R. Kursawe, L. Sun, V. Sivakamasundari, I. Kycia, P. Robson, and M. L. Stitzel, "Single-cell transcriptomes identify human islet cell signatures and reveal cell-type–specific expression changes in type 2 diabetes," *Genome Research*, vol. 27, pp. 208–222, Nov. 2016.

[51] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, "A benchmark of batch-effect correction methods for single-cell rna sequencing data," *Genome biology*, vol. 21, no. 1, pp. 1–32, 2020.

[52] L. Zappia and F. J. Theis, "Over 1000 tools reveal trends in the single-cell rna-seq analysis landscape," *bioRxiv*, 2021.

[53] M. Büttner, Z. Miao, F. A. Wolf, S. A. Teichmann, and F. J. Theis, "A test metric for assessing single-cell rna-seq batch correction," *Nature methods*, vol. 16, no. 1, pp. 43–49, 2019.

[54] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni, "Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors," *Nature biotechnology*, vol. 36, no. 5, pp. 421–427, 2018.

[55] K. Polański, M. D. Young, Z. Miao, K. B. Meyer, S. A. Teichmann, and J.-E. Park, "Bbknn: fast batch alignment of single cell transcriptomes," *Bioinformatics*, vol. 36, no. 3, pp. 964–965, 2020.

[56] B. Hie, B. Bryson, and B. Berger, "Efficient integration of heterogeneous single-cell transcriptomes using scanorama," *Nature biotechnology*, vol. 37, no. 6, pp. 685–691, 2019.

[57] R. Lopez, A. Gayoso, and N. Yosef, "Enhancing scientific discoveries in molecular biology with deep generative models," *Molecular Systems Biology*, vol. 16, no. 9, p. e9198, 2020.

[58] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[59] M. Lotfollahi, M. Naghipourfar, F. J. Theis, and F. A. Wolf, "Conditional out-of-distribution generation for unpaired data using transfer vae," *Bioinformatics*, vol. 36, no. Supplement_2, pp. i610–i617, 2020.

[60] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, and N. Yosef, "Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models," *Molecular systems biology*, vol. 17, no. 1, p. e9620, 2021.

[61] A. Gayoso, Z. Steier, R. Lopez, J. Regier, K. L. Nazor, A. Streets, and N. Yosef, "Joint probabilistic modeling of single-cell multi-omic data with totalvi," *Nature Methods*, vol. 18, no. 3, pp. 272–282, 2021.

[62] M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert, "Simultaneous

epitope and transcriptome measurement in single cells," *Nature methods*, vol. 14, no. 9, pp. 865–868, 2017.

[63] J. D. Buenrostro, B. Wu, H. Y. Chang, and W. J. Greenleaf, "Atac-seq: a method for assaying chromatin accessibility genome-wide," *Current protocols in molecular biology*, vol. 109, no. 1, pp. 21–29, 2015.

[64] T. Ashuach, D. A. Reidenbach, A. Gayoso, and N. Yosef, "Peakvi: A deep generative model for single cell chromatin accessibility analysis," *bioRxiv*, 2021.

[65] Y. An, F. Drost, F. Theis, B. Schubert, and M. Lotfollahi, "Jointly learning t-cell receptor and transcriptomic information to decipher the immune response," *bioRxiv*, 2021.

[66] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, *et al.*, "Integrated analysis of multimodal single-cell data," *Cell*, 2021.

[67] Y. Ji, M. Lotfollahi, F. Alexander Wolf, and F. J. Theis, "Machine learning for perturbational single-cell omics," *cels*, vol. 12, pp. 522–537, June 2021.

[68] J. Lamb, E. D. Crawford, D. Peck, J. W. Modell, I. C. Blat, M. J. Wrobel, J. Lerner, J.-P. Brunet, A. Subramanian, K. N. Ross, *et al.*, "The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease," *science*, vol. 313, no. 5795, pp. 1929–1935, 2006.

[69] M. Lotfollahi, F. A. Wolf, and F. J. Theis, "scgen predicts single-cell perturbation responses," *Nature methods*, vol. 16, no. 8, pp. 715–721, 2019.

[70] M. Lotfollahi, A. Klimovskaia, C. De Donno, Y. Ji, I. L. Ibarra, F. A. Wolf, N. Yakubova, F. J. Theis, and D. Lopez-Paz, "Learning interpretable cellular responses to complex perturbations in high-throughput screens," *bioRxiv*, 2021.

[71] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[72] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The journal of educational research*, vol. 96, no. 1, pp. 3–14, 2002.

[73] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, "Deep learning: new computational modelling techniques for genomics," *Nature Reviews Genetics*, vol. 20, no. 7, pp. 389–403, 2019.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[75] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, pp. 1310–1318, PMLR, 2013.

[76] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[77] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*, pp. 646–661, Springer, 2016.

[78] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[80] E. Plaut, "From principal subspaces to principal components with linear autoencoders," *arXiv preprint arXiv:1804.10253*, 2018.

[81] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell rna-seq denoising using a deep count autoencoder," *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.

[82] A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel, "Poincaré maps for analyzing complex hierarchies in single-cell data," *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.

[83] S. Rybakov, M. Lotfollahi, F. J. Theis, and F. A. Wolf, "Learning interpretable latent autoencoder representations with annotations of feature sets," *bioRxiv*, 2020.

[84] G. O. Consortium, "Gene ontology consortium: going forward," *Nucleic acids research*, vol. 43, no. D1, pp. D1049–D1056, 2015.

[85] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

[86] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[87] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[88] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.

[89] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[90] M. Lotfollahi, H. Agarwala, F. J. Theis, *et al.*, "Out-of-distribution prediction with disentangled representations for single-cell rna sequencing data," *bioRxiv*, 2021.

[91] H. Yu and J. D. Welch, "Michigan: sampling from disentangled representations of single-cell data using generative adversarial networks," *Genome biology*, vol. 22, no. 1, pp. 1–26, 2021.

[92] J. Tomczak and M. Welling, "Vae with a vampprior," in *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223, PMLR, 2018.

[93] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," *arXiv preprint arXiv:1804.00891*, 2018.

[94] E. Mathieu, C. L. Lan, C. J. Maddison, R. Tomioka, and Y. W. Teh, "Continuous hierarchical representations with poincar\'e variational auto-encoders," *arXiv preprint arXiv:1901.06033*, 2019.

[95] H. M. Kang, M. Subramaniam, S. Targ, M. Nguyen, L. Maliskova, E. McCarthy, E. Wan, S. Wong, L. Byrnes, C. M. Lanata, *et al.*, "Multiplexed droplet single-cell rna-sequencing using natural genetic variation," *Nature biotechnology*, vol. 36, no. 1, pp. 89–94, 2018.

[96] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[97] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[98] M. Litviňuková, C. Talavera-López, H. Maatz, D. Reichart, C. L. Worth, E. L. Lindberg, M. Kanda, K. Polanski, M. Heinig, M. Lee, *et al.*, "Cells of the adult human heart," *Nature*, vol. 588, no. 7838, pp. 466–472, 2020.

[99] J. He, X. Wang, G. Neubig, and T. Berg-Kirkpatrick, "A probabilistic formulation of unsupervised text style transfer," *arXiv preprint arXiv:2002.03912*, 2020.

[100] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review. arxiv," *arXiv preprint arXiv:1705.04058*, 2017.

[101] S. R. Srivatsan, J. L. McFaline-Figueroa, V. Ramani, L. Saunders, J. Cao, J. Packer, H. A. Pliner, D. L. Jackson, R. M. Daza, L. Christiansen, *et al.*, "Massively multiplex chemical transcriptomics at single-cell resolution," *Science*, vol. 367, no. 6473, pp. 45–51, 2020.

[102] E. Swanson, C. Lord, J. Reading, A. T. Heubeck, P. C. Genge, Z. Thomson, M. D. Weiss, X.-j. Li, A. K. Savage, R. R. Green, *et al.*, "Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using tea-seq," *Elife*, vol. 10, p. e63632, 2021.

[103] E. Stephenson, G. Reynolds, R. A. Botting, F. J. Calero-Nieto, M. D. Morgan, Z. K. Tuong, K. Bach, W. Sungnak, K. B. Worlock, M. Yoshida, *et al.*, "Single-cell multi-omics analysis of the immune response in covid-19," *Nature medicine*, vol. 27, no. 5, pp. 904–916, 2021.

[104] M. Liu, J. Zhang, E. Adeli, and D. Shen, "Landmark-based deep multi-instance learning for brain disease diagnosis," *Medical image analysis*, vol. 43, pp. 157–168, 2018.

[105] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International conference on machine learning*, pp. 2127–2136, PMLR, 2018.

[106] G. Campanella, M. G. Hanna, L. Geneslaw, A. Miraflor, V. W. K. Silva, K. J. Busam, E. Brogi, V. E. Reuter, D. S. Klimstra, and T. J. Fuchs, "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images," *Nature medicine*, vol. 25, no. 8, pp. 1301–1309, 2019.

[107] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods andapplications," *arXiv preprint arXiv: 1812.08434*, 2018.

[108] E. Papalexi, E. P. Mimitou, A. W. Butler, S. Foster, B. Bracken, W. M. Mauck, H.-H. Wessels, Y. Hao, B. Z. Yeung, P. Smibert, *et al.*, "Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens," *Nature Genetics*, vol. 53, no. 3, pp. 322–331, 2021.

[109] C. J. Frangieh, J. C. Melms, P. I. Thakore, K. R. Geiger-Schuller, P. Ho, A. M. Luoma, B. Cleary, L. Jerby-Arnon, S. Malu, M. S. Cuoco, *et al.*, "Multimodal pooled perturb-cite-seq screens in patient models define

mechanisms of cancer immune evasion," *Nature genetics*, vol. 53, no. 3, pp. 332–341, 2021.

Appendices

Appendix A

Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology (2021).*

This is a published version of the article Nature Biotechnology following peer review. The article is open-access thus the published version is inserted here.

(i) Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Ziga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin, Fabian J. Theis. **"Mapping Single-Cell Data to Reference Atlases by Transfer Learning."** Nature Biotechnology, August, 1–10 (2021).
The article is also available online at:
`https://www.nature.com/articles/s41587-021-01001-7`

nature biotechnology

ANALYSIS

https://doi.org/10.1038/s41587-021-01001-7

Check for updates

OPEN

Mapping single-cell data to reference atlases by transfer learning

Mohammad Lotfollahi[1,2], Mohsen Naghipourfar[1], Malte D. Luecken[1], Matin Khajavi[1], Maren Büttner[1], Marco Wagenstetter[1], Žiga Avsec[3], Adam Gayoso[4], Nir Yosef[4,5,6,7], Marta Interlandi[8], Sergei Rybakov[1,9], Alexander V. Misharin[10] and Fabian J. Theis[1,2,9] ✉

Large single-cell atlases are now routinely generated to serve as references for analysis of smaller-scale studies. Yet learning from reference data is complicated by batch effects between datasets, limited availability of computational resources and sharing restrictions on raw data. Here we introduce a deep learning strategy for mapping query datasets on top of a reference called single-cell architectural surgery (scArches). scArches uses transfer learning and parameter optimization to enable efficient, decentralized, iterative reference building and contextualization of new datasets with existing references without sharing raw data. Using examples from mouse brain, pancreas, immune and whole-organism atlases, we show that scArches preserves biological state information while removing batch effects, despite using four orders of magnitude fewer parameters than de novo integration. scArches generalizes to multimodal reference mapping, allowing imputation of missing modalities. Finally, scArches retains coronavirus disease 2019 (COVID-19) disease variation when mapping to a healthy reference, enabling the discovery of disease-specific cell states. scArches will facilitate collaborative projects by enabling iterative construction, updating, sharing and efficient use of reference atlases.

Large single-cell reference atlases[1–4] comprising millions[5] of cells across tissues, organs, developmental stages and conditions are now routinely generated by consortia such as the Human Cell Atlas[6]. These references help to understand the cellular heterogeneity that constitutes natural and inter-individual variation, aging, environmental influences and disease. Reference atlases provide an opportunity to radically change how we currently analyze single-cell datasets: by learning from the appropriate reference, we could automate annotation of new datasets and easily perform comparative analyses across tissues, species and disease conditions.

Learning from a reference atlas requires mapping a query dataset to this reference to generate a joint embedding. Yet query datasets and reference atlases typically comprise data generated in different laboratories with different experimental protocols and thus contain batch effects. Data-integration methods are typically used to overcome these batch effects in reference construction[7]. This requires access to all relevant datasets, which can be hindered by legal restrictions on data sharing. Furthermore, contextualizing a single dataset requires rerunning the full integration pipeline, presupposing both computational expertise and resources. Finally, traditional data-integration methods consider any perturbation between datasets that affects most cells as a technical batch effect, but biological perturbations may also affect most cells. Thus, conventional approaches are insufficient for mapping query data onto references across biological conditions.

Exploiting large reference datasets is a well-established approach in Computer Vision[8] and Natural Language Processing[9]. In these fields, commonly used deep learning approaches typically require a large number of training samples, which are not always available. By leveraging weights learned from large reference datasets to enhance learning on a target or query dataset[10], transfer-learning (TL) models such as ImageNet[11] and BERT[12] have revolutionized analysis approaches[8,9]: TL has improved method performance with small datasets (for example, clustering[13], classification and/or annotation[14]) and enabled model sharing[15–18]. Recently, TL has been applied to single-cell RNA-seq (scRNA-seq) data for denoising[19], variance decomposition[20] and cell type classification[21,22]. However, current TL approaches in genomics do not account for technical effects within and between the reference and query[19] and lack of systematic retraining with query data[20–23]. These limitations can lead to spurious predictions on query data with no or small overlap in cell types, tissues or species[24,25]. Nonetheless, deep learning models for data integration in single-cell genomics demonstrated superior performance[7,26–28]. We propose a TL and fine-tuning strategy to leverage existing conditional neural network models and transfer them to new datasets, called 'architecture surgery', as implemented in the scArches pipeline. scArches is a fast and scalable tool for updating, sharing and using reference atlases trained with a variety of neural network models. Specifically, given a basic reference atlas, scArches enables users to share this reference as a trained network with other users, who can in turn update the reference using query-to-reference mapping and partial weight optimization without sharing their data. Thus, users can build their own extended reference models or perform stepwise analysis of datasets as they are collected,

[1]Helmholtz Center Munich—German Research Center for Environmental Health, Institute of Computational Biology, Neuherberg, Germany. [2]School of Life Sciences Weihenstephan, Technical University of Munich, Munich, Germany. [3]Department of Computer Science, Technical University of Munich, Munich, Germany. [4]Center for Computational Biology, University of California, Berkeley, Berkeley, CA, USA. [5]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA. [6]Chan Zuckerberg Biohub, San Francisco, CA, USA. [7]Ragon Institute of MGH, MIT and Harvard, Cambridge, MA, USA. [8]Institute of Medical Informatics, University of Münster, Münster, Germany. [9]Department of Mathematics, Technical University of Munich, Munich, Germany. [10]Division of Pulmonary and Critical Care Medicine, Feinberg School of Medicine, Northwestern University, Chicago, IL, USA. ✉e-mail: fabian.theis@helmholtz-muenchen.de

which is often crucial for emerging clinical datasets. Furthermore, scArches allows users to learn from reference data by contextualizing new (for example, disease) data with a healthy reference in a shared representation. Due to the flexible choice of the underlying core model that is transferred using scArches, we can learn references with various base models but also train on multimodal data. We demonstrate the features of scArches using single-cell datasets ranging from pancreas to whole-mouse atlases and immune cells from patients with COVID-19. scArches is able to iteratively update a pancreas reference, transfer labels or unmeasured data modalities between reference atlases and query data and map COVID-19 data onto a healthy reference while preserving disease-specific variation.

Results

scArches enables mapping query data to reference. Consider the scenario with N 'reference' scRNA-seq datasets of a particular tissue or organism. A common approach to integrate such datasets is to use a conditional variational autoencoder (CVAE) (for example, single-cell variational inference (scVI)[29], transfer variational autoencoder (trVAE)[30]) that assigns a categorical label S_i to each dataset that corresponds to the study label. These study labels may index traditional batch IDs (that is, samples, experiments across laboratories or sequencing technologies), biological batches (that is, organs or species when used over the set of orthologous genes), perturbations such as disease or a combination of these categorical variables. Training a CVAE model with reference studies $S_{1:N}$ (Fig. 1a) results in a latent space where the effects of condition labels (that is, batch or technology) are regressed out. Thus, we can use this embedding for further downstream analysis such as visualization or identification of cell clusters or subpopulations.

Architectural surgery is a TL approach that takes existing reference models and adapts these to enable query-to-reference mapping. After training an existing autoencoder model on multiple reference datasets, architectural surgery is the process of transferring these trained weights with only minor weight adaptation (fine tuning) and adding a condition node to map a new study into this reference. While this approach is broadly applicable on any deep conditional model, here we apply scArches to three unsupervised models (CVAEs, trVAE, scVI), a semi-supervised (single-cell annotation using variational inference (scANVI))[31] algorithm and a multimodal (total variational inference (totalVI))[32] algorithm (Methods).

To facilitate model sharing, we adapted existing reference-building methods to incorporate them into our scArches package as 'base models'. Reference models built within scArches can be uploaded to a model repository via our built-in application programming interface for Zenodo (Methods). To enable users to map new datasets on top of custom reference atlases, we propose sharing model weights, which one can download from the model repository and fine tune with new query data. This fine tuning extends the model by adding a set of trainable weights per query dataset called 'adaptors'. In classical conditional neural networks, a study corresponds to an input neuron. As a trained network has a rigid architecture, it does not allow for adding new studies within the given network. To overcome this, we implement the architecture surgery approach to incorporate new study labels as new input nodes (Methods). These new input nodes with trainable weights are the aforementioned adaptors. Importantly, adaptors are shareable, allowing users to further customize shared reference models by downloading a reference atlas, choosing a set of available adaptors for that reference and finally incorporating the user's own data by training query adaptors (Fig. 1b). Trainable parameters of the query model are restricted to a small subset of weights for query study labels. Depending on the size of this subset, this restriction functions as an inductive bias to prevent the model from strongly adapting its parameters to the query studies. Thus, query data update the reference atlas.

To illustrate the feasibility of this approach, we applied scArches with trVAE, scVI and scANVI (see Supplementary Tables 1–7 for detailed parameters) to consecutively integrate two studies into a pancreas reference atlas comprising three studies (Fig. 1c). To additionally simulate the scenario in which query data contain a new cell type absent in the reference, we removed all alpha cells in the training reference data. We first trained different existing reference models within the scArches framework to integrate training data and construct a reference atlas (Fig. 1d,e and Supplementary Fig. 1, first column). Once the reference atlas was constructed, we fine tuned the reference model with the first query data (SMART-seq2 (SS2)) and iteratively updated the reference atlas with this study (Fig. 1d,e, second column) and the second query data (CelSeq2, Fig. 1d,e, third column). After each update, our model overlays data from all shared cell types present in both query and reference while yielding a separate and well-mixed cluster of alpha cells in the query datasets (black dashed circles in Fig. 1d,e). To further assess the robustness of the approach, we held out two cell types (alpha cells and gamma cells) in the reference data while keeping both in the query datasets. Here our model robustly integrated query data while placing unseen cell types into distinct clusters (Supplementary Fig. 2). Additional testing using simulated data showed that scArches is also robust to simultaneously updating the reference atlas with several query studies at a time (Supplementary Fig. 3).

Overall, TL with architectural surgery enables users to update learnt reference models by integrating query data while accounting for differences in cell type composition.

Minimal fine tuning performs best for model update. To determine the number of weights to optimize during reference mapping, we evaluated the performance of different fine-tuning strategies. Reference mapping performance was assessed using ten metrics recently established to evaluate data-integration performance[7] in terms of removal of batch effects and preservation of biological variation. Batch-effect removal was measured via principal-component regression, entropy of batch mixing, k-nearest neighbor (kNN) graph connectivity and average silhouette width (ASW). Biological conservation was assessed with global cluster matching (adjusted Rand index (ARI), normalized mutual information (NMI)), local neighborhood conservation (kNN accuracy), cell type ASW and rare cell type metrics (isolated label scores). An accurate reference mapping integration should result in both high conservation of biological variation and high batch-removal scores.

Next to fine tuning only the weights connecting newly added studies as proposed above (adaptors), we also considered (1) training input layers in both encoder and decoder while the rest of the weights were frozen and (2) fine tuning all weights in the model. We trained a reference model for each base model using 250,000 cells from two mouse brain studies[33,34]. Next, we compared the integration performance of candidate fine-tuning strategies when mapping two query datasets[1,35] onto the reference data. Applying scArches trVAE to the brain atlas, the model with the fewest parameters performed competitively with other approaches in integrating different batches while preserving distinctions between different cell types (Fig. 2a–c). Notably, the strongly regularized scArches reduced trainable parameters by four to five orders of magnitude (Fig. 2d). Overall, evaluating integration accuracy for different base models demonstrates the optimal time and integration performance trade-off of using adaptors to incorporate new query datasets compared to that of other approaches (Fig. 2e).

Architectural surgery allows for efficient data integration. To use scArches, one requires a reference atlas model. The quality of reference mapping performed by scArches relies on the parameterization and architecture chosen for the base model as well as the quality and quantity of reference data. To determine the sensitivity of scArches

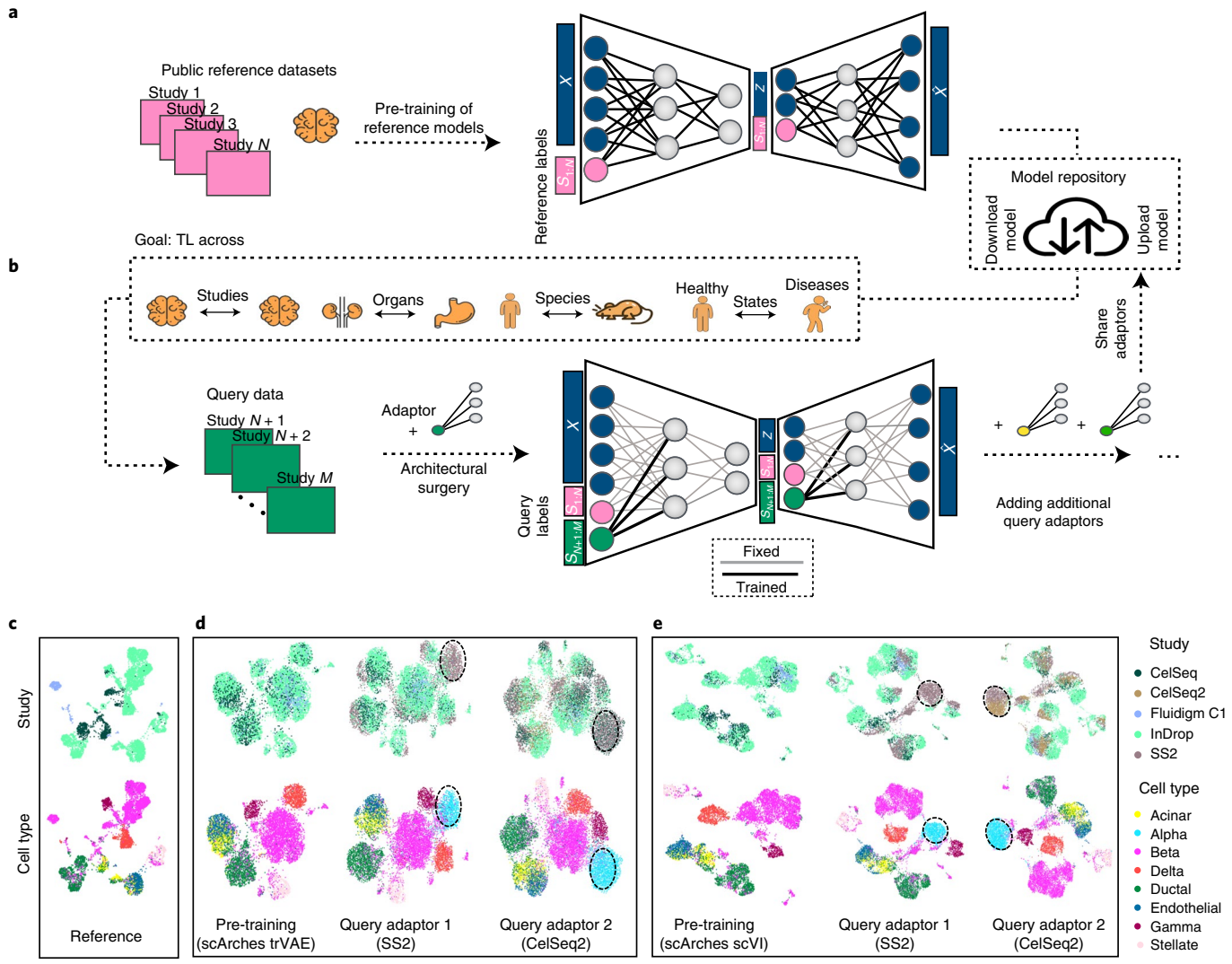


Fig. 1 | scArches enables iterative query-to-reference single-cell integration. a, Pre-training of a latent representation using public reference datasets and corresponding reference labels. **b**, Decentralized model building: users download parameters for the atlas of interest, fine tune the model and optionally upload their updated model for other users. **c**–**e**, Illustration of this workflow for a human pancreas atlas across different scArches base models. Training a reference atlas across three human pancreas datasets (CelSeq, InDrop, Fluidigm C1), uniform manifold approximation and projection (UMAP) embedding for the original (**c**) and the integrated reference for pre-trained reference models (**d**,**e**, first column). Second column in **d**,**e**, querying a new SS2 dataset to the integrated reference. Updating the cell atlas with a fifth dataset (CelSeq2). Third column in **d**,**e**, black dashed circles represent cells absent in the reference data. UMAP plots are based on the model embedding.

reference mapping to the reference model used, we investigated how much reference data are needed to enable successful reference mapping. Therefore, we leveraged a human immune cell dataset composed of bone marrow[36] and peripheral blood mononuclear cells (PBMCs)[37–39]. We built reference models of increasing quality by incrementally including more studies in reference building while using the rest of the studies as query data. To further challenge the model, we included a unique cell type for each study while removing it from the rest of the studies. In our experiments, the reference mapping accuracy of scArches scANVI substantially increased until at least 50% (~10,000 cells) of the data were used as reference (Fig. 3a–c). Specifically, we observed distinct clusters of megakaryocyte progenitors, human pluripotent stem cells, CD10$^+$ B cells and erythroid progenitors only in higher reference ratios (Fig. 3b,c), while these were mixed in the lowest reference fraction (Fig. 3a). This observation held true across other base models (Fig. 3d and Supplementary Fig. 4). We repeated similar experiments on brain and pancreas datasets (Supplementary Figs. 5 and 6). Overall, while performance is both model and data dependent, we observed a robust performance when at least 50% of the data, including multiple study batches, are used in reference training (Fig. 3d and Supplementary Figs. 7–10).

Reference mapping is designed to generate an integrated dataset without sharing raw data and with limited computational resources. Thus, it must be evaluated against the gold standard of de novo data integration, for which these restrictions are not present. To assess this, we performed scArches reference mapping using a reference model containing approximately two-thirds of batches and compared this to existing full integration autoencoder methods and other existing approaches[22,40–44]. The overall score for the scArches reference mapping model is similar to that of de novo integration performance (Fig. 3e and Supplementary Figs. 13–15).

We also evaluated the speed of scArches reference mapping compared to full integration strategies. In an scArches pipeline, the reference model must either be built once and can be shared or it can be downloaded directly to map query datasets. Therefore, we

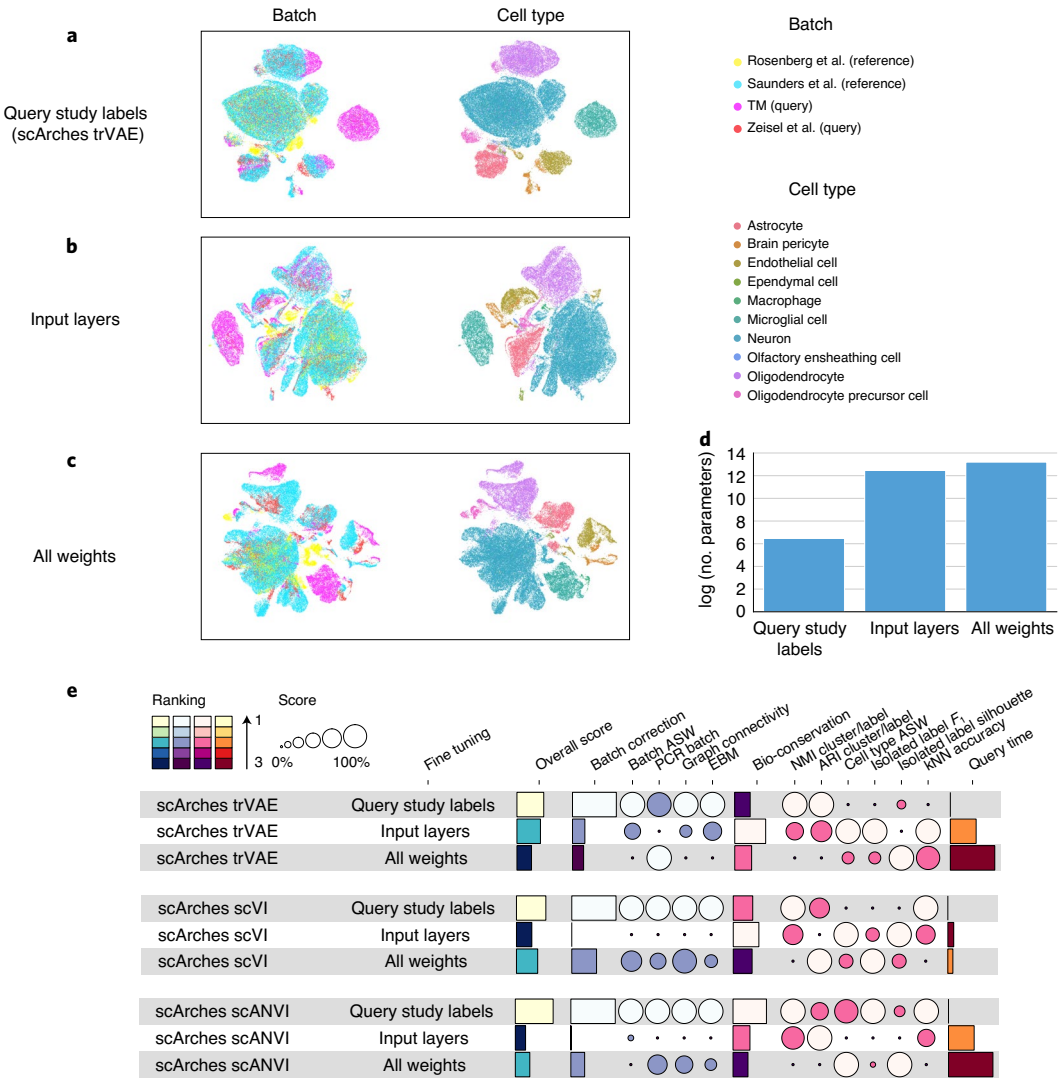


Fig. 2 | TL and architecture surgery allow fast and accurate reference mapping. a–c, Comparing different granularity levels in the proposed TL strategy by mapping data from two brain studies to a reference brain atlas. The reference model was trained on a subset of 250,000 cells from two brain studies and then updated with data from Zeisel et al.[35] and the TM brain subset. Fine-tuning strategies vary from training a few query study label weights (**a**) to input layers of both the encoder and decoder (**b**) and retraining the full network (**c**). **d**, Number of trained weights across these three granularity levels. **e**, Comparison of integration accuracy for different fine-tuning strategies on mapping data from two query studies on a brain reference atlas across various base models. Individual scores are minimum–maximum scaled between 0 and 1. Overall scores were computed using a 40:60-weighted mean of batch correction and bio-conservation scores, respectively (see Methods for further visualization details). EBM, entropy of batch mixing; PCR, principal-component regression.

consider the time spent by the user to map query datasets as the relevant basis for our comparisons. The running time is also dependent on the base model type. For example, trVAE was much slower than other base models due to the maximum mean discrepancy term, while scVI and scANVI were the fastest (Supplementary Fig. 11a). Overall, scArches can offer a speed-up of up to approximately fivefold and eightfold for scVI and scANVI compared to that of running a de novo autoencoder-based integration for these methods (Supplementary Fig. 16a). This allows mapping of 1 million query cells in less than 1 h (Supplementary Fig. 11b).

scArches is sensitive to nuanced cell states. We further evaluated scArches under a series of challenging cases. A particular challenge for deep learning methods with many trainable parameters is the small data regime. Thus, we first tested the ability of scArches to map rare cell types. For this purpose, we subsampled a specific cell type in our pancreas and immune integration tasks (delta cells and CD16$^+$ monocytes, respectively), such that this population constituted between ~0.1% and ~1.0% of the whole data. Next, we integrated one study as query data and evaluated the quality of reference mapping for the rare cell type. While in all cases the query cells are integrated with reference cells, rare cluster cells can be mixed with other cell types when the fraction is smaller than ~0.5%, and we only observed a distinct cluster for higher fractions (Supplementary Fig. 12).

Second, we evaluated our method on data with continuous trajectories. We trained a reference model using a pancreatic endocrinogenesis dataset[45] from three early time points (embryonic day (E)12.5, E13.5 and E14.5). We integrated the latest time point (E15.5) as query data. Here query data integrated well with reference data, and our velocity[46] analysis on the integrated data confirmed the known differentiation trajectory toward major alpha, beta, delta and epsilon fates (Supplementary Fig. 13).

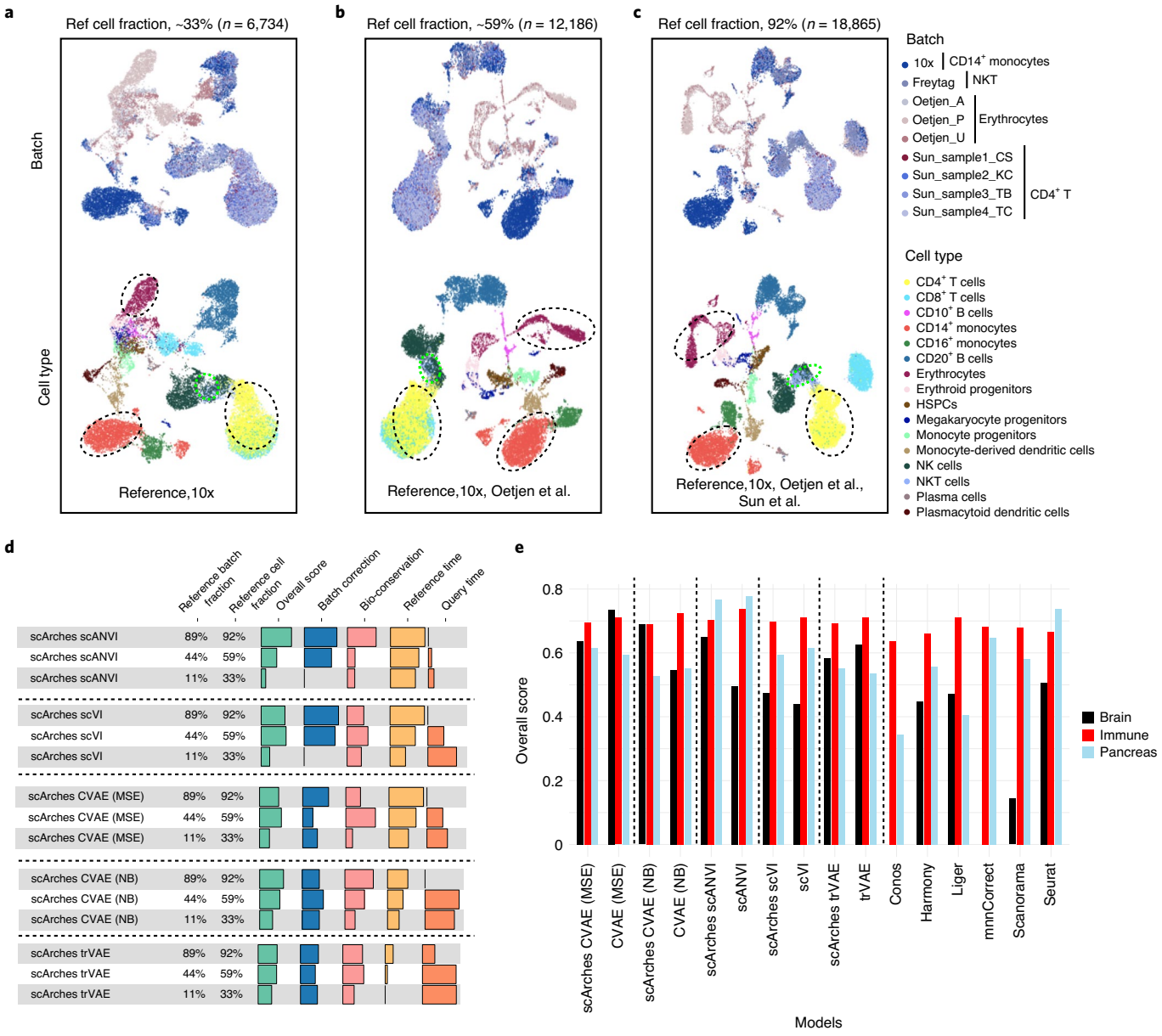


Fig. 3 | scArches enables efficient reference mapping compared to full integration workflow with existing data-integration methods. a–d, Evaluating the effect of the reference (ref) size for immune data ($n = 20{,}522$) on the quality of reference mapping. **a–c**, UMAP plots show the latent representation of the integrated query and reference data together for scArches scANVI. Cell types highlighted with dashed circles represent cells unique to a specific study denoted by the batch legend. Reference ratio refers to the fraction of cells in the reference compared to all data. The studies used as reference are indicated at the bottom of each panel. HSPCs, hematopoietic stem and progenitor cells. **d**, Quantitative metrics for the performance of different base models across various reference ratios in immune data. **e**, Comparison of different scArches base models trained with a reference dataset with ~66% of batches in the whole data against de novo full integration methods across immune ($n = 20{,}522$), pancreas ($n = 15{,}681$) and brain ($n = 332{,}129$) datasets. Conos and mnnCorrect were not able to integrate brain data due to excessive memory usage and time requirements, respectively. MSE, mean squared error.

Finally, we evaluated how well scArches resolves nuanced, transcriptionally similar cell types in the query. We therefore trained a reference model excluding natural killer (NK) cells, while the reference data contained highly similar NKT cells. Integrated query and reference cells resulted in a separate NK cluster in proximity to NKT cells (Supplementary Fig. 14a). Repeating a similar experiment with both NK and NKT cells absent in the reference reproduced distinct clusters for both populations in the vicinity of each other (Supplementary Fig. 14b).

scArches enables knowledge transfer from reference to query. The ultimate goal of query-to-reference mapping is to leverage and transfer information from the reference. This knowledge transfer can be transformative for analyzing new query datasets by transferring discrete cell type labels that facilitate annotation of query data[47,48] or by imputing continuous information such as unmeasured modalities that are present in reference but absent from query measurements[32,48,49]

We first studied transferring discrete information (for example, cell type labels) to query data. We used the recently published Tabula Senis[3] as our reference, which includes 155 distinct cell types across 23 tissues and five age groups ranging from 1 month to 30 months from plate-based (SS2) and droplet-based (10x Genomics) assays. As query data, we used cells from the 3-month time point (equivalent to Tabula Muris (TM)).

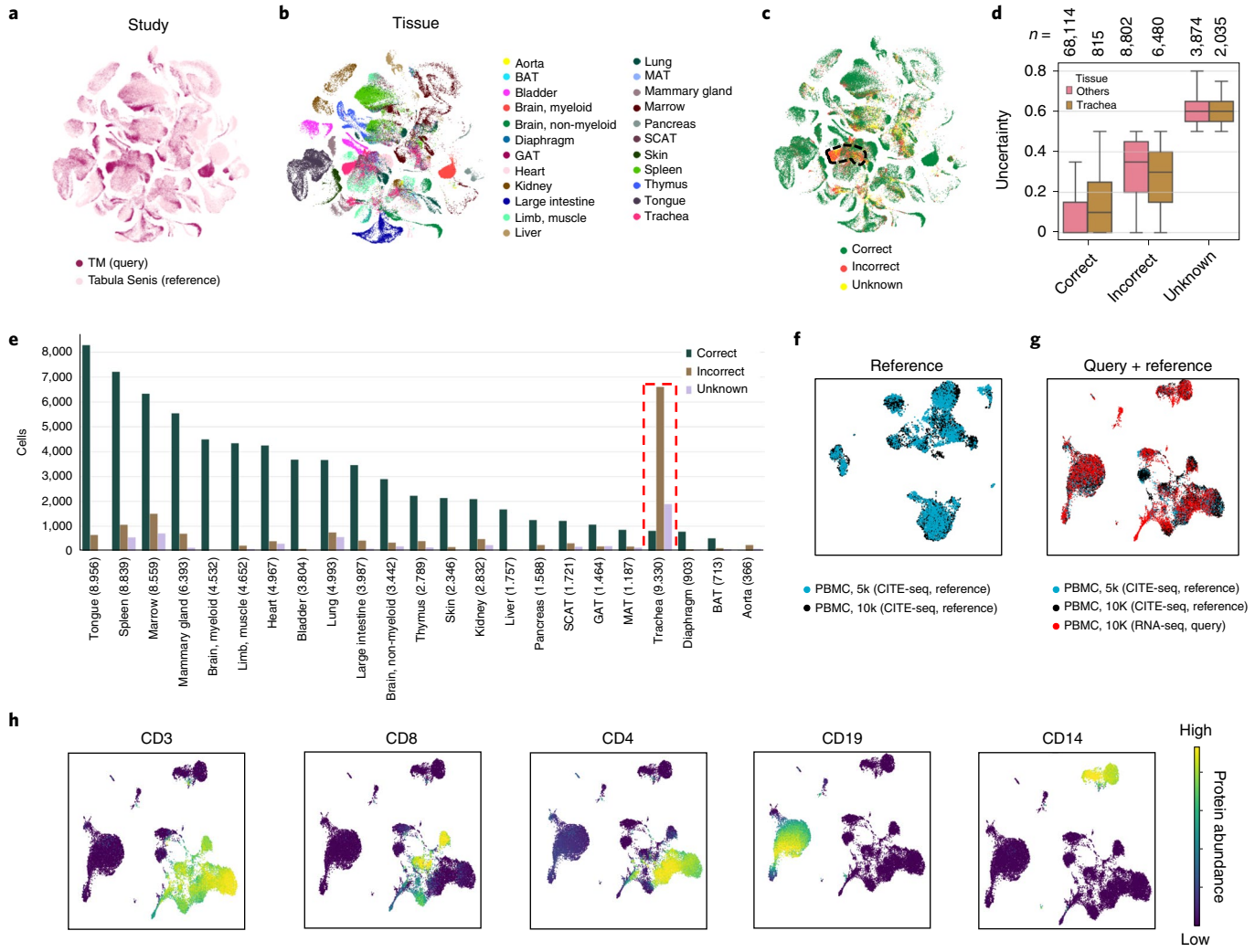


Fig. 4 | scArches successfully transfers knowledge from reference to query. a,**b**, Querying TM (n = 90,120) to the larger reference atlas Tabula Senis (n = 264,287) using scArches trVAE (**a**) across different tissues (**b**). Tissues were correctly grouped across the two datasets (**a**,**b**). **c**, Location of misclassified and unknown cells after transferring labels from the reference to the query data. The highlighted tissue represents tracheal cells, which we removed from the reference data. **d**, Reported uncertainty of the transferred labels, which was low in correctly classified cells and high in the incorrect and unknown ones, particularly in the trachea. Box plots indicate the median (center lines) and interquartile range (hinges), and whiskers represents minimum and maximum values. Numbers of cells (n) are denoted above each box plot. **e**, Numbers of correct, incorrect and unknown cells across different tissues. The red dashed line represents tracheal cells only present in TM. **f**, Construction of a reference CITE-seq atlas using two PBMC datasets (n = 10,849 cells). **g**, Integration of scRNA-seq data (n = 10,315) into the CITE-seq reference. **h**, Imputation of missing proteins for the query dataset using the reference. BAT, brown adipose tissue; GAT, gonadal adipose tissue; MAT, mesenteric adipose tissue; SCAT, subcutaneous adipose tissue.

The query data consists of 90,120 cells from 24 tissues including a previously unseen tissue, trachea, which we excluded from the reference data. scArches trVAE accurately integrates query and reference data across time points and sequencing technologies and creates a distinct cluster of tracheal cells (n = 9,330) (Fig. 4a,b and Supplementary Fig. 15a; see Supplementary Fig. 16 for tissue-level data).

We then investigated the transfer of cell type labels from the reference dataset. Each cell in the query TM was annotated using its closest neighbors in the reference dataset. Additionally, our classification pipeline provides an uncertainty score for each cell while reporting cells with more than 50% uncertainty as unknown (Methods). scArches achieved ~84% accuracy across all tissues (Fig. 4c). Moreover, most of the misclassified cells and cells from the unseen tissue received high uncertainty scores (Fig. 4d and Supplementary Fig. 15b). Overall, classification results across tissues indicated a robust prediction accuracy across most tissues (Fig. 4e), while highlighting cells that were not mappable to the reference. Therefore, scArches can successfully merge large and complex query datasets into reference atlases. Notably, we used scArches to map a large query (the mouse cell atlas[2]) onto TM and further onto a recently published human cell landscape (HCL)[4] reference, demonstrating applicability to study similarity of cell types across species (Supplementary Note 1 and Supplementary Figs. 18–21). Overall, scArches-based label projection performs competitively when compared with state-of-the-art methods such as SVM rejection[47,50], Seurat version 3 (ref. [22]) and logistic regression classifiers[50] (Supplementary Fig. 17).

In addition to the label transfer, one can use reference atlases to impute continuous information in the query data such as missing antibody panels in RNA-seq-only assays. Indeed, one can combine scArches with existing multimodal integration architectures such as totalVI[32], a model for joint modeling of RNA expression and surface protein abundance in single cells. Leveraging scArches totalVI, we

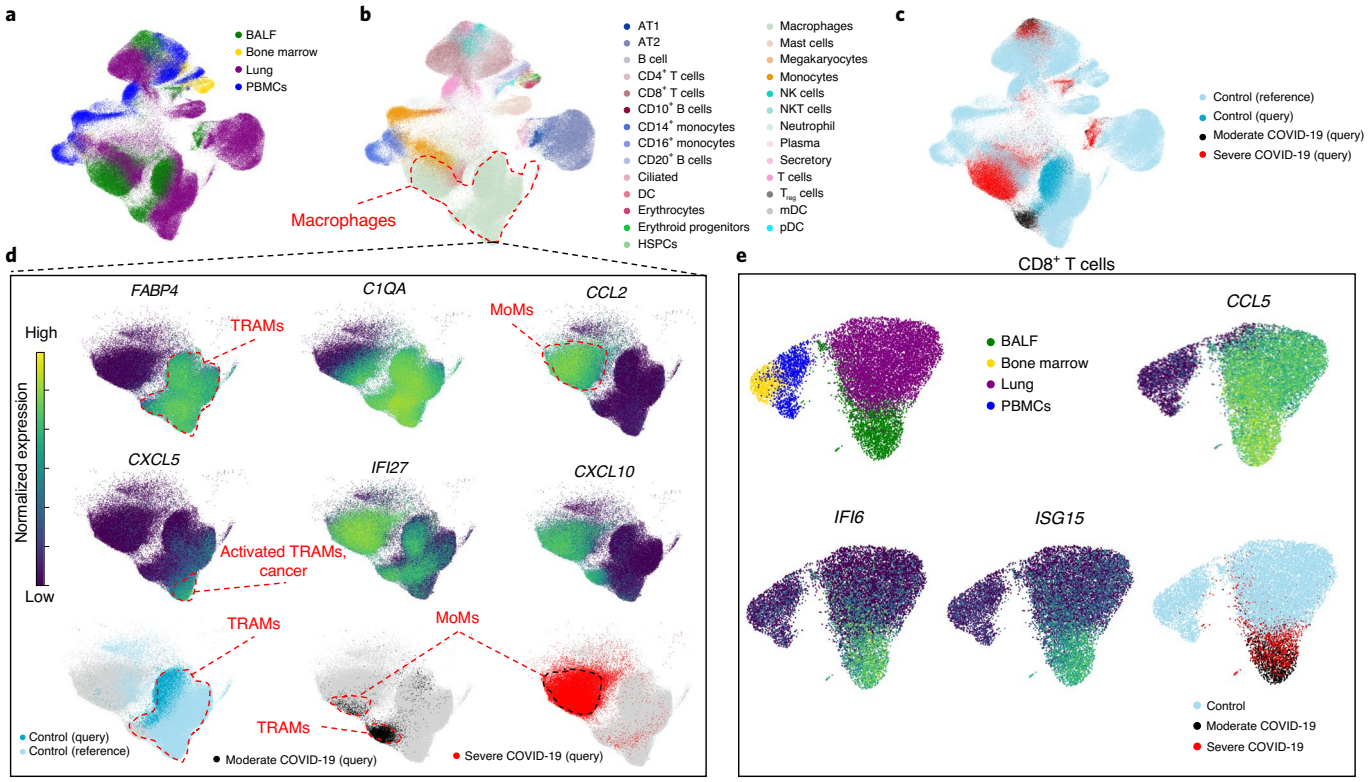


Fig. 5 | scArches resolves severity in COVID-19 query data mapped to a healthy reference and reveals emergent cell states. a–c, Integration of query data from immune and epithelial cells from patients with COVID-19 on top of a healthy immune atlas across multiple tissues (**a**), cell types (**b**) and cell states (**c**). BALF, bronchoalveolar lavage fluid; DC, dendritic cell; T_{reg} cells, regulatory T cells. **d**, Comparison of various macrophage subpopulations across both healthy and COVID-19 states. Top, TRAMs are characterized by expression of *FABP4*, while monocyte-derived inflammatory macrophages (MoMs) are characterized by expression of *CCL2*. Upregulation of *C1QA* illustrates maturation of MoMs as they differentiate from monocytes to macrophages. Middle, *CXCL5*, *IFI27* and *CXCL10* illustrate context-dependent activation of TRAMs. Bottom, scArches correctly maps TRAMs from query to TRAMs from reference, while preserving MoMs, unseen in the reference, as a distinct cell type. **e**, Separation of activated query CD8$^+$ T cells from patients with COVID-19 from the rest of CD8$^+$ T cells in the reference. AT, alveolar type; mDC, myeloid dendritic cells; pDC, plasmacytoid dendritic cells.

built a cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq)[51] reference using two publicly available PBMC datasets (Fig. 4f). Next, we integrated query scRNA-seq data into the reference atlas (Fig. 4g) and used the multimodal reference atlas to impute missing protein data for the query dataset. Using imputed protein abundances, we can distinguish the observed major populations such as T cells (CD3$^+$, CD4$^+$ and CD8$^+$), B cells (CD19$^+$) and monocytes (CD14$^+$) (Fig. 4h) (see Supplementary Fig. 22 for all proteins).

Preserving COVID-19 cell states after reference mapping. In the study of disease, contextualization with healthy reference data is essential. A successful disease-to-healthy data integration should satisfy three criteria: (1) preservation of biological variation of healthy cell states; (2) integration of matching cell types between healthy reference and disease query; and (3) preservation of distinct disease variation, such as the emergence of new cell types that are unseen during healthy reference building. To showcase how one can perform disease contextualization with scArches, we created a reference aggregated from bone marrow[36], PBMCs[37–39] and normal lung tissue[52–54] ($n = 154{,}723$; Fig. 5a–c) and then mapped onto it a dataset containing alveolar macrophages and other immune cells collected via bronchoalveolar lavage from (1) healthy controls and patients with (2) moderate and (3) severe COVID-19 ($n = 62{,}469$)[55]. As described by Liao and colleagues, this dataset contains immune cells found in the normal lung (for example, tissue-resident alveolar macrophages, TRAMs) as well as unique populations that are absent in the normal lung and emerge only during inflammation (for example, monocyte-derived alveolar macrophages, MoAMs)[55]. We used a negative binomial (NB) CVAE base model for this experiment (Methods).

We first evaluated the integration of query batches in the reference. scArches successfully integrated alveolar macrophages from different datasets and preserved biological variability between them, although some ambient RNA signals remained (Supplementary Note 2 and Supplementary Fig. 23). For example, activated TRAMs (*FABP4*$^+$*IL1B*$^+$*CXCL5*$^+$) that originate from a single individual (donor 2 in the Travaglini et al.[52] dataset) formed a distinct subcluster within TRAMs (Fig. 5a–d). We then evaluated the projection of COVID-19 query data onto the reference model. The dataset from Liao and colleagues contains the following cell types: airway epithelial cells, plasma cells and B cells, CD4$^+$ and CD8$^+$ T cells, NK cells, neutrophils, mast cells, dendritic cells, monocytes and alveolar macrophages (Fig. 5b,c and Supplementary Fig. 24)[55]. Within the macrophage cluster (characterized by the expression of *C1QA*), two distinct populations dominated the structure of the embedding (Fig. 5c,d): TRAMs (*FABP4*$^+$*C1Q*$^+$*CCL2*$^-$) and inflammatory MoAMs (*FABP4*$^-$*C1Q*$^+$*CCL2*$^+$). As expected, query TRAMs from healthy controls integrated well with TRAMs from the reference dataset. While TRAMs from patients with moderate COVID-19 integrated with TRAMs from control lung tissue, they did not mix with normal TRAMs completely, as they were activated and characterized by increased expression of *IFI27* and *CXCL10*. MoAMs are predominantly found in samples from patients with severe COVID-19

and to a lesser extent in samples from patients with moderate COVID-19. MoAMs originate from monocytes that are recruited to sites of infection (as illustrated by the gradient of *C1QA* expression) and thus do not appear in healthy reference tissue. Indeed, MoAMs were embedded in closer proximity to monocytes than to TRAMs in our embedding, reflecting their ontological relationship (see Supplementary Fig. 25 for partition-based graph abstraction[56] proximity analysis).

We then evaluated $CD8^+$ T cells. While the reference bone marrow and blood cells predominantly contained naive $CD8^+$ T cells (*CCL5*$^-$), lung and bronchoalveolar lavage fluid contained cytotoxic memory $CD8^+$ T cells (*GZMA*$^+$*GZMH*$^+$; Fig. 5e and Supplementary Fig. 26). Moreover, cytotoxic memory $CD8^+$ T cells from patients with COVID-19 were characterized by the expression of interferon-response genes *ISG15*, *MX1* and others, which is in agreement with a recent report that the interferon response is a feature separating severe acute respiratory syndrome coronavirus 2 pneumonia from other viral and non-viral pneumonias[57,58] (Fig. 5e, Supplementary Note 2 and Supplementary Fig. 26).

Overall, the scArches joint embedding was dominated by nuanced biological variation, for example, macrophage subtypes, even when these subtypes were not annotated in reference datasets (for example, activated TRAMs from patients with moderate COVID-19 or a patient with a lung tumor). Although disease states were absent in the reference data, scArches separated these states from the healthy reference and even preserved biological variation patterns. Hence, disease-to-healthy integration with scArches met all three criteria for successful integration.

Discussion

We introduced architectural surgery, an easy-to-implement approach for TL, reusing neural network models by adding input nodes and weights (adaptors) for new studies and then fine tuning only those parameters. Architectural surgery can extend any conditional neural network-based data-integration method to enable decentralized reference updating, facilitate model reuse and provide a framework for learning from reference data.

In applications, we demonstrated how integration of whole-species atlases enables the transfer of cell type annotations from a reference to a query atlas. We further showed that COVID-19 query data can be mapped on top of a healthy reference while retaining variation among both disease and healthy states, which we promote in scArches by avoiding showing the method the disease effect during training. In general, different effects such as disease states are assumed to be orthogonal in high-dimensional space[43]; thus, if a batch-confounded effect (for example, any donor-level covariate when donor is used as batch) is not seen during training, we would not expect it to be removed. We observe this phenomenon in our COVID-19 example and in multiple experiments: biologically meaningful variations from held-out alpha cells in the pancreas (Fig. 1d,e) or unseen nuanced cell identities in immune cell data (Supplementary Fig. 14) are mapped to a new location when they are unseen during training.

The reduction in model training complexity by training adaptors moreover leads to an increase in speed while preserving integration accuracy when compared to full integration methods. It also improves usability and interpretability, because mapping a query dataset to a reference requires no further hyperparameter optimization and keeps reference representation intact. Adaptors only impact the first network layer and therefore 'commute': application order is irrelevant for iteratively expanding a reference, arriving always at the same result due to the frozen nature of the network and independence of adaptor weights. With scArches, one can therefore use pre-trained neural network models without computational expertise or graphics processing unit power to map, for example, disease data onto stored reference networks prepared from independent atlases. We make use of these features by providing a model database on Zenodo (Methods).

Model sharing in combination with reference mapping via scArches allows users to create custom reference atlases by updating public ones and paves the way for automated and standardized analyses of single-cell studies. Especially for human data, sharing expression profiles is often difficult due to data-protection regulations, size, complexity and other organizational hurdles. With scArches, users can obtain an overview of the whole dataset to validate harmonized cell type annotation. By sharing a pre-trained neural network model that can be locally updated, international consortia can generate a joint embedding without requiring access to the full gene sets. In turn, users can quickly build upon this by mapping their own typically much smaller data into the reference, acquiring robust latent spaces, cell type annotation and identification of subtle state-specific differences with respect to the reference.

scArches is a tool that leverages existing conditional autoencoder models to perform reference mapping. Thus, by design, it inherits both benefits and limitations of the underlying base models. For example, a limitation of these models is that the integrated output is a low-dimensional latent space instead of a corrected feature matrix as provided by mnnCorrect or Scanorama. While generating a batch-corrected input matrix is possible[30], this may lead to spurious and false signals similar to denoising methods[59]. Similarly, imputation of modalities not measured in query data (for example, via scArches totalVI) performs better for more abundant features, which has already been outlined in the original totalVI publication[32]. A further limitation is the need for a sufficiently large and diverse set of samples for reference building. Deep learning models typically have more trainable parameters than other integration methods and thus often require more data. This constraint translates directly to the performance of scArches reference mapping (Fig. 3a–d): using a small reference along with a low number of studies leads to poor integration of query data while removing biological variation such as nuanced cell types. Furthermore, even with equal training data, reference model performance will differ, affecting reference mapping via scArches. As robust and scalable reference building is still ongoing research in the scRNA-seq field[7], the choice of reference model is a central challenge when using scArches. Yet we demonstrate that even imperfect reference models (Supplementary Note 3) can be used for meaningful analyses as demonstrated by our data analysis of patients with COVID-19. Finally, one must consider the limitations of the base model on batch-effect removal during reference mapping, in which it is unlikely to remove stronger batch effects than those seen in the training data. In our cross-species experiments, reference mapping performs well mostly in immune cell populations, which appear to contain the smallest batch effect across species (Supplementary Figs. 18, 20 and 21).

While scArches is applicable in many scenarios, it is best suited when the query data consists of cell types and experimental protocols similar to the reference data. Then, the query data may easily contain new cell types or states such as disease or other kinds of perturbations, which are preserved after mapping. Additionally, we advise against using scArches for integrating query data with a reference created out of a single study and recommend integration with full sample access instead. Further, the number of overlapping genes between query and reference data can also influence integration quality. We generally recommend using a larger set of highly variable genes (HVGs) in the reference-building step to guarantee a bigger feature overlap between reference and query, which increases the robustness of reference mapping in the presence of missing genes (Methods and Supplementary Fig. 28).

We envision two major directions for further applications and development. First, scArches can be applied to generate context-specific large-scale disease atlases. Large disease reference datasets are increasingly becoming available[60–62]. By mapping

between disease references, we can assess the similarity of these diseases at the single-cell level and thus inform for finding mechanisms, reverting disease state or studying perturbations, for example, for drug repurposing. The suitability of model organisms for disease research can be directly translated into the human context: for example, projecting mouse single-cell tumor data on a reference human patient tumor atlas may help to identify accurate tumor models that include desired molecular and cellular properties of a patient's microenvironment. Incorporating additional covariates as conditional neurons in the reference model will allow modeling of treatment response with a certain perturbation or drug[63,64]. Secondly, we envision assembling multimodal single-cell reference atlases to include epigenomic[65], chromosome conformation[66], proteome[51] and spatially resolved measurements.

In summary, with the availability of reference atlases, we expect scArches to accelerate the use of these atlases to analyze query datasets.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-021-01001-7.

Received: 30 July 2020; Accepted: 28 June 2021;
Published online: 30 August 2021

References

1. Schaum, N., Karkanias, J., Neff, N. & Pisco, A. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* **562**, 367–372 (2018).
2. Han, X. et al. Mapping the mouse cell atlas by Microwell-seq. *Cell* **172**, 1091–1107 (2018).
3. The Tabula Muris Consortium et al. A single cell transcriptomic atlas characterizes aging tissues in the mouse. Preprint at *bioRxiv* https://doi.org/10.1101/661728 (2020).
4. Han, X. et al. Construction of a human cell landscape at single-cell level. *Nature* **581**, 303–309 (2020).
5. 10x Genomics. 10x Datasets Single Cell Gene Expression, Official 10x Genomics Support. https://www.10xgenomics.com/resources/datasets/
6. Regev, A. et al. Science forum: the human cell atlas. *eLife* **6**, e27041 (2017).
7. Luecken, M. D. et al. Benchmarking atlas-level data integration in single-cell genomics. Preprint at *bioRxiv* https://doi.org/10.1101/2020.05.22.111161 (2020).
8. Zheng, H. et al. Cross-domain fault diagnosis using knowledge transfer strategy: a review. *IEEE Access* **7**, 129260–129290 (2019).
9. Ruder, S., Peters, M. E., Swayamdipta, S. & Wolf, T. Transfer learning in natural language processing. in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* 15–18 (ACL, 2019).
10. Yang, L., Hanneke, S. & Carbonell, J. A theory of transfer learning with applications to active learning. *Mach. Learn.* **90**, 161–189 (2013).
11. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. in *Proceedings of the 25th International Conference on Neural Information Processing Systems* 1097–1105 (NIPS, 2012).
12. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint at https://arxiv.org/abs/1810.04805v2 (2018).
13. Hsu, Y.-C., Lv, Z. & Kira, Z. Learning to cluster in order to transfer across domains and tasks. Preprint at https://arxiv.org/abs/1711.10125 (2017).
14. Shin, H.-C. et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **35**, 1285–1298 (2016).
15. Dahl, G. E., Yu, D., Deng, L. & Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **20**, 30–42 (2011).
16. Ker, J., Wang, L., Rao, J. & Lim, T. Deep learning applications in medical image analysis. *IEEE Access* **6**, 9375–9389 (2017).
17. Avsec, Ž. et al. The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nat. Biotechnol.* **37**, 592–600 (2019).
18. Gayoso, A. et al. scvi-tools: a library for deep probabilistic analysis of single-cell omics data. Preprint at *bioRxiv* https://doi.org/10.1101/2021.04.28.441833 (2021).
19. Wang, J. et al. Data denoising with transfer learning in single-cell transcriptomics. *Nat. Methods* **16**, 875–878 (2019).
20. Stein-O'Brien, G. L. et al. Decomposing cell identity for transfer learning across cellular measurements, platforms, tissues, and species. *Cell Syst.* **8**, 395–411 (2019).
21. Lieberman, Y., Rokach, L. & Shay, T. CaSTLe—classification of single cells by transfer learning: harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments. *PLoS ONE* **13**, e0205499 (2018).
22. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019).
23. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 (2020).
24. Wang, X., Huang, T.-K. & Schneider, J. Active transfer learning under model shift. in *Proceedings of the 31st International Conference on Machine Learning* 1305–1313 (PMLR, 2014).
25. Arjovsky, M., Bottou, L., Gulrajani, I. & Lopez-Paz, D. Invariant risk minimization. Preprint at https://arxiv.org/abs/1907.02893 (2019).
26. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods* **16**, 715–721 (2019).
27. 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).
28. Litvinukova, M. et al. Cells and gene expression programs in the adult human heart. Preprint at *bioRxiv* https://doi.org/10.1101/2020.04.03.024075 (2020).
29. Lopez, R., Regier, J., Jordan, M. I. & Yosef, N. Information constraints on auto-encoding variational Bayes. in *Advances in Neural Information Processing Systems* 6114–6125 (NIPS, 2018).
30. 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).
31. Xu, C. et al. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol. Syst. Biol.* **17**, e9620 (2021).
32. Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nat. Methods* **18**, 272–282 (2021).
33. Saunders, A. et al. Molecular diversity and specializations among the cells of the adult mouse brain. *Cell* **174**, 1015–1030 (2018).
34. Rosenberg, A. B. et al. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360**, 176–182 (2018).
35. Zeisel, A. et al. Molecular architecture of the mouse nervous system. *Cell* **174**, 999–1014 (2018).
36. Oetjen, K. A. et al. Human bone marrow assessment by single-cell RNA sequencing, mass cytometry, and flow cytometry. *JCI Insight* **3**, e124928 (2018).
37. Freytag, S., Tian, L., Lönnstedt, I., Ng, M. & Bahlo, M. Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. *F1000Res.* **7**, 1297 (2018).
38. Sun, Z. et al. A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. *Nat. Commun.* **10**, 1649 (2019).
39. 10x Genomics. 10x Datasets Single Cell Gene Expression, Official 10x Genomics Support https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3
40. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).
41. Welch, J. D. et al. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177**, 1873–1887 (2019).
42. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat. Biotechnol.* **37**, 685–691 (2019).
43. Haghverdi, L., Lun, A. T., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).
44. Barkas, N. et al. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nat. Methods* **16**, 695–698 (2019).
45. Bastidas-Ponce, A. et al. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development* **146**, dev173849 (2019).
46. Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.* **38**, 1408–1414 (2020).
47. Abdelall, T. et al. A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biol.* **20**, 194 (2019).
48. Stuart, T. et al. Comprehensive integration of single cell data. *Cell* **177**, 1888–1902 (2019).
49. Zhou, Z., Ye, C., Wang, J. & Zhang, N. R. Surface protein imputation from single cell transcriptomes by deep neural networks. *Nat. Commun.* **11**, 651 (2020).
50. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

51. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
52. Travaglini, K. J. et al. A molecular cell atlas of the human lung from single cell RNA sequencing. *Nature* **587**, 619–625 (2020).
53. Reyfman, P. A. et al. Single-cell transcriptomic analysis of human lung provides insights into the pathobiology of pulmonary fibrosis. *Am. J. Respir. Crit. Care Med.* **199**, 1517–1536 (2019).
54. Madissoon, E. et al. scRNA-seq assessment of the human lung, spleen, and esophagus tissue stability after cold preservation. *Genome Biol.* **21**, 1 (2020).
55. Liao, M. et al. Single-cell landscape of bronchoalveolar immune cells in patients with COVID-19. *Nat. Med.* **26**, 842–844 (2020).
56. Wolf, F. A. et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 59 (2019).
57. Grant, R. A. et al. Circuits between infected macrophages and T cells in SARS-CoV-2 pneumonia. *Nature* **590**, 635–641 (2021).
58. Muus, C. et al. Integrated analyses of single-cell atlases reveal age, gender, and smoking status associations with cell type-specific expression of mediators of SARS-CoV-2 viral entry and highlights inflammatory programs in putative target cells. Preprint at *bioRxiv* https://doi.org/10.1101/2020.04.19.049254 (2020).
59. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Res.* **7**, 1740 (2019).
60. Schulte-Schrepping, J. et al. Severe COVID-19 is marked by a dysregulated myeloid cell compartment. *Cell* **182**, 1419–1440 (2020).
61. Wen, W. et al. Immune cell profiling of COVID-19 patients in the recovery stage by single-cell sequencing. *Cell Discov.* **6**, 31 (2020).
62. Wilk, A. J. et al. A single-cell atlas of the peripheral immune response in patients with severe COVID-19. *Nat. Med.* **26**, 1070–1076 (2020).
63. Lotfollahi, M. et al. Compositional perturbation autoencoder for single-cell response modeling. Preprint at *bioRxiv* https://doi.org/10.1101/2021.04.14.439903 (2021).
64. Lotfollahi, M., Dony, L., Agarwala, H. & Theis, F. Out-of-distribution prediction with disentangled representations for single-cell RNA sequencing data. in *ICML 2020 Workshop on Computational Biology* **37** (ICML, 2020).
65. Kelsey, G., Stegle, O. & Reik, W. Single-cell epigenomics: recording the past and predicting the future. *Science* **358**, 69–75 (2017).
66. Qiu, X. et al. Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* **14**, 979–982 (2017).

Methods

Architecture surgery. Our method relies on a concept known as TL. TL is an approach in which weights from a model trained on one task are taken and used as weight initialization or fine tuning for another task. We introduce an architecture surgery, a strategy to apply TL in the context of conditional generative models and single-cell data. Our proposed method is general and can be used to perform TL on both CVAEs and conditional generative adversarial nets[67].

Let us assume that we want to train a reference CVAE model with a d-dimensional dataset ($x \epsilon R^d$) from n different studies ($s \epsilon R^n$), where R denotes real number space. We further assume that the bottleneck z with layer size is $k(z \epsilon R^k)$. Then, an input for a single cell i will be $x' = x \cdot s$, where x and s are the d-dimensional gene expression profile and n-dimensional one-hot encoding of study labels, respectively. The $\cdot$ symbol denotes the row-wise concatenation operation. Therefore, the model receives $(d+n)$-dimensional and $(k+n)$-dimensional vectors as inputs for encoder and decoder, respectively. Assuming m query datasets, the target model will be initialized with all the parameters from the reference model. To incorporate m new study labels, we add m new dimensions to s in both encoder and decoder networks. We refer to these new added study labels as s'. Next, m new randomly initialized weight vectors are also added to the first layer of the encoder and decoder. Finally, we fine tune the new model by only training the weights connected to the last m dimensions of x' that correspond to the condition labels. Let us assume that p and q are the number of neurons in the first layer of the encoder and decoder; then, during the fine tuning, only $(m) \times (p+q)$ parameters will be trained. Let us parameterize the first layer of the encoder and decoder part of the scArches as f_1 and g_1, respectively. Let us further assume that ReLU activations are used in the layers. Therefore the equations for f_1 and g_1 are

$$f_1(x, s, s\prime; \phi_x, \phi_s, \phi_{s\prime}) = \max(0, \phi_x^T x + \phi_s^T s + \phi_{s\prime}^T s\prime)$$
$$g_1(z, s, s\prime; \theta_z, \theta_s, \theta_{s\prime}) = \max(0, \theta_z^T z + \theta_s^T s + \theta_{s\prime}^T s\prime),$$

where ϕ and θ are parameters of encoder and decoder, and T denotes transpose operation. Therefore, the gradients of f and g with respect to $\phi_{s'}$ and $\theta_{s'}$ are

$$\nabla_{\phi_{s\prime}} f_1 = \begin{cases} 0 & \text{if } \phi_x^T x + \phi_s^T s + \phi_{s\prime}^T s\prime \leq 0 \\ s\prime & \text{otherwise} \end{cases}$$
$$\nabla_{\theta_{s\prime}} g_1 = \begin{cases} 0 & \text{if } \theta_z^T z + \theta_s^T s + \theta_{s\prime}^T s\prime \leq 0 \\ s\prime & \text{otherwise} \end{cases}$$

Finally, because all other weights except $\phi_{s'}$ and $\theta_{s'}$ are frozen, we only compute the gradient of scArches' cost function with respect to $\phi_{s'}$ and $\theta_{s'}$:

$$\nabla_{\phi_{s\prime}} L_{\text{scArches}}(x, s, s\prime; \theta, \phi) = \nabla_{f_1} L_{\text{scArches}}(x, s, s\prime; \phi) \cdot \nabla_{\phi_{s\prime}} f_1(x, s, s\prime; \phi_x, \phi_s, \phi_{s\prime})$$
$$\nabla_{\theta_{s\prime}} L_{\text{scArches}}(x, s, s\prime; \theta, \phi) = \nabla_{g_1} L_{\text{scArches}}(z, s, s\prime; \theta, \phi) \cdot \nabla_{\theta_{s\prime}} g_1(x, s, s\prime; \theta_z, \theta_s, \theta_{s\prime}).$$

scArches base models. *Conditional variational autoencoders.* Variational autoencoders (VAEs)[68] were shown to learn the underlying complex structure of data. VAEs were proposed for generative modeling of the underlying data leveraging variational inference and neural networks to maximize the following equation:

$$p_\theta(X \mid S) = \int p_\theta(X \mid Z, S) p_\theta(Z \mid S) dZ,$$

where X is a random variable representing the model's input, S is a random variable indicating various conditions, θ is the neural network parameters, and $p_\theta(X \mid Z, S)$ is the output distribution that we sample Z to reconstruct X. In the following equation, we exploit notations from ref. [29] and a tutorial from ref. [69]. We approximate the posterior distribution $p_\theta(Z|X, S)$ using the variational distribution $q_\phi(Z|X, S)$ that is approximated by a deep neural network parameterized with ϕ:

$$L_{\text{CVAE}}(X, S; \phi, \theta) = \log p_\theta(X \mid S) - \alpha \cdot D_{\text{KL}}(q_\phi(Z|X, S) || p_\theta(Z|X, S)) =$$
$$= \mathbb{E}_{q_\phi(Z|X,S)}[\log p_\theta(X \mid Z, S)] - \alpha \cdot D_{\text{KL}}(q_\phi(Z|X, S) || p_\theta(Z|S)),$$

where $\theta = \{\theta\prime, \theta_z, \theta_s\}$ and $\phi = \{\phi\prime, \phi_x, \phi_s\}$ are parameters of decoder and encoder, respectively, $\mathbb{E}$ is the expectation and D_{KL} is the Kullback-Leibler divergence scaled by parameter α. On the left-hand side, we have the log likelihood of the data and an error term that depends on the capacity of the model. The right-hand side of the above equation is also known as the evidence lower bound. CVAE[70] is an extension of VAE framework in which $S \neq \emptyset$.

scArches trVAE. trVAE[30] builds upon VAE[68] with an extra regularization to further match the distribution between conditions. Following the method proposed by Lotfollahi et al.[30], we use the representation of the first layer in the decoder, which is regularized by maximum mean discrepancy[71]. For implementation, we use multi-scale radial basis function (RBF) kernels defined as

$$k(x, x\prime) = \sum_{i=1}^{l} k(x, x\prime, \gamma_i),$$

where $k(x, x\prime, \gamma_i) = e^{-\gamma_i |*x - *x\prime|^2}$, γ_i is a hyperparameter, and l denotes maximum number of RBF kernels.

We will parameterize the encoder and decoder part of scArches as f_ϕ and g_θ, respectively. So the networks f_ϕ and g_θ will accept inputs x, s and z, s, respectively. Let us distinguish the first ($g^{(1)}_{\theta_z,\theta_s}$) and the remaining layers ($g^{(2)}_{\theta\prime}$) of the decoder network $g_\theta = g^{(2)}_{\theta\prime} \circ g^{(1)}_{\theta_z,\theta_s}$. Therefore, we can define the following maximum mean discrepancy (MMD) cost function:

$$L_{\text{MMD}}(X, S; \phi, \theta_z, \theta_s) = \sum_{i \neq j}^{\text{No. studies}} l_{\text{MMD}}(g^{(1)}_{\theta_z,\theta_s}(f_\phi(X_{S=i}, i), i), g^{(1)}_{\theta_z,\theta_s}(f_\phi(X_{S=j}, j), j)),$$

where

$$l_{\text{MMD}}(X, X') = \frac{1}{N_0^2} \sum_{n=1}^{N_0} \sum_{m=1}^{N_0} k(x_n, x_m) + \frac{1}{N_1^2} \sum_{n=1}^{N_1} \sum_{m=1}^{N_1} k(x'_n, x'_m) - \frac{2}{N_0 N_1} \sum_{n=1}^{N_0} \sum_{m=0}^{N_1} k(x_n, x'_m).$$

We used the notation $X_{S=i}$ for samples drawn from ith study distribution in the training data. Finally, the trVAE's cost function is

$$L_{\text{trVAE}}(X, S; \phi, \theta) = L_{\text{CVAE}}(X, S; \phi, \theta) - \beta \cdot L_{\text{MMD}}(X, S; \phi, \theta_z, \theta_s),$$

where β is a regularization scale parameter. The gradients of trVAE's cost function with respect to ϕ_s and θ_s are

$$\nabla_{\phi_s} L_{\text{trVAE}}(X, S; \theta, \phi) = \nabla_{\phi_s} L_{\text{CVAE}}(X, S; \theta, \phi) - \beta \cdot \nabla_{\phi_s} L_{\text{MMD}}(X, S; \phi, \theta_z, \theta_s),$$
$$\nabla_{\theta_s} L_{\text{trVAE}}(X, S; \theta, \phi) = \nabla_{\theta_s} L_{\text{CVAE}}(X, S; \theta, \phi) - \beta \cdot \nabla_{\theta_s} L_{\text{MMD}}(X, S; \phi, \theta_z, \theta_s).$$

Therefore L_{trVAE} can be optimized using stochastic gradient ascent with respect to ϕ_s and θ_s as all the other parameters are frozen.

scArches scVI. Lopez et al.[27] developed a fully probabilistic approach, called scVI, for normalization and analysis of scRNA-seq data. scVI is also based on a CVAE, described in detail above. But, in contrast to the trVAE architecture, the decoder assumes a zero-inflated negative binomial (ZINB) distribution; and therefore the reconstruction loss differs to the MSE loss of trVAE. Another major difference is that scVI explicitly models the library size, which is needed for the ZINB loss calculation with another shallow neural network called the library encoder. Therefore, with similar notation as above, we have the output distribution $p(X|Z, S, L)$, where L is the scaling factor that is sampled by the outputs of the library encoder, namely the empirical mean L_μ and the variance L_σ of the log library per batch:

$$L \sim \text{lognormal}(L_\mu, L_\sigma^2).$$

When we now separate the outputs of the decoder g_θ into g^x_θ, the decoded mean proportion of the expression data, and g^d_θ, the decoded dropout effects, we can write the ZINB mass function for $p(X|Z, S, L)$ in the following closed form:

$$\begin{cases} p(X = 0|Z, S, L) = \\ \quad g^d_\theta(Z, S) + (1 - g^d_\theta(Z, S)) \left(\frac{\Sigma}{\Sigma + L \cdot g^x_\theta(Z,S)}\right)^\Sigma \\ p(X = Y|Z, S, L) = \\ \quad (1 - g^d_\theta(Z, S)) \frac{\Gamma(Y+\Sigma)}{\Gamma(Y+1)\Gamma(\Sigma)} \left(\frac{\Sigma}{\Sigma + L \cdot g^x_\theta(Z,S)}\right)^\Sigma \left(\frac{g^x_\theta(Z,S)}{\Sigma + L \cdot g^x_\theta(Z,S)}\right)^Y, \end{cases}$$

where Σ is the gene-specific inverse dispersion, Γ is the gamma function, and Y represents non-zero entries drawn from a ZINB distribution. Because the evidence lower bound and therefore the optimization objective can be calculated by applying the reparameterization trick and supposing Gaussians, which is possible here because of the proposed ZINB distribution, we can write the scVI cost function as follows:

$$L_{\text{scVI}}(X, S; \phi, \theta) = L_{\text{CVAE}}(X, S; \phi, \theta) - \alpha \cdot D_{\text{KL}}(q_\phi(L|X, S) || p_\theta(L)).$$

Furthermore, because of the applied reparameterization trick, an automatic differentiation operator can be used, and the cost function can be optimized by applying stochastic gradient descent. For the application in scArches, we removed

the library encoder and computed the library size for each batch in a closed form by summing up the counts. This does not decrease the performance of the model and accelerates the surgery step. The resulting network can then be used similarly to the trVAE network by simply retraining only the condition weights corresponding to the new batch annotations in S.

scArches scANVI. scANVI is a semi-supervised method that builds up on the scVI model and was proposed in detail by Xu et al.[31]. By constructing a mixture model, it is able to make use of any cell type annotations during autoencoder training to improve latent representation of the data. In addition to this, scANVI is capable of labeling datasets with only some marker gene labels as well as transferring labels from a labeled dataset to an unlabeled dataset. For the training of scANVI, the authors proposed an alternating optimization of the cost function $L_{\text{scANVI}}(X, S;\phi, \theta)$ and the classification loss C, which results from a shallow neural network that serves as a classifier with a cross-entropy loss after the last softmax layer. In more detail, the cost function can be formulated in the following manner:

$$L_{\text{scANVI}}(X, S;\phi, \theta) = L_{\text{labeled}}(X, S, C;\phi, \theta) + L_{\text{unlabeled}}(X, S;\phi, \theta),$$

where C is the cell types in the annotated datasets, and both cost function summands L_{labeled} and $L_{\text{unlabeled}}$ are obtained by similar calculations as in the case of scVI. The major difference here, however, is that the Kullback–Leibler divergence is applied to an additional latent encoder that takes cell type annotations into account. For the unlabeled case, each sample is broadcasted into every available cell type. As scANVI builds up on scVI, we use the same adjustments here to apply surgery. On top of that, we also freeze the classifier even for semi-supervised query data, because we want an unchanging reference performance for building a cell atlas and also to force cells in the query data with the same cell type annotation to be near to the corresponding reference cells in the latent representation.

scArches totalVI. For the purpose of combining paired measurement of RNA and surface proteins from the same cells, such as for CITE-seq data, Gayoso et al.[32] presented a deep generative model called totalVI. totalVI learns a joint low-dimensional probabilistic representation of RNA and protein measurements. For the RNA portion of the data, totalVI uses an architecture similar to that of scVI, which we discussed in detail above; but, for proteins, a new model is introduced that separates protein information into background and foreground components. With the surgery functionality of scArches added to totalVI, it is now possible to learn a joint latent space of RNA and protein data on a CITE-seq reference dataset and do surgery on a query dataset with only RNA data to impute protein data for that query dataset as well. To accomplish this goal, we again only retrain the weights that correspond to the new batch labels.

CVAEs for single-cell genomics. CVAEs were first applied to scRNA-seq data in scVI[29] for data integration and differential testing. Here we focus on how CVAEs perform data integration and potential pitfalls. These models receive a matrix of gene expression profile for cells (X) and label (condition) matrix (S). The condition matrix comprises a nuisance variable, which we want to regress out from the data. Labels can encode batch, technologies, disease state or other discrete variables. The CVAE model seeks to infer a low-dimensional latent space (Z) for the cell that would be free of variations explained by the label variable. For example, if the labels are the experimental batches, then similar cell type separated by batch effect in the original gene expression space will be aligned together. Importantly, variation attributed to the labels will be merely regressed in the latent space while still present in the output of the CVAE. Therefore, the reconstructed output will still contain batch effects. Additionally, while autoencoder-based data-integration methods were shown to perform best when outputting integrated embeddings, these methods can also output corrected expression matrices. This is achieved by forcing all batches to be transformed to a specific batch as previously shown in scGen.

scArches builds upon existing CVAEs. The results of the integration heavily depend on the type of labels used as batch covariates for condition inputs. If the dataset is the batch covariate, within-dataset donor effects will not be removed, but donors become more comparable across datasets. In our COVID-19 example, the disease is used as a query and thus is not captured fully in the encoder, which is trained on data from healthy individuals. Adaptor training removes the donor- and/or dataset-specific batch effect from a disease sample but does not remove variation unseen in network training. Thus, choice of training data and choice of batch covariate are crucial to assess whether variation from disease is removed in training or not.

Overall, the choice and design of the label matrix is a crucial step for optimal outcome. The label matrix can encode one covariate (for example, batch), multiple covariates (for example, technology, cell types, disease, species,…) or a combination of covariates (for example, technology and species). However, the interpretability of the latent space will be challenging in the presence of complex label design and will require extra caution.

Model sharing. We currently support an application programming interface to upload and download model weights and data (if available) using Zenodo. Zenodo is a general-purpose open-access repository developed to enable researchers to share datasets and software. We have provided step-by-step guides for the whole pipeline from training and uploading models to downloading, updating and further sharing models. These tutorials can be found in the scArches GitHub repository (https://github.com/theislab/scarches).

Feature overlap between reference and query. An important practical challenge for reference mapping using scArches is the number of features (genes) that are shared between the query and the reference model and/or dataset. It is important to note that, with the current pipeline, the query data must have the same gene set as the reference model. Therefore, the user has to replace missing reference genes in the query with zeros. We investigated the effect of zero filling and observed that integration performance was robust when 10% (of 2,000 genes) were missing from query data. However, the performance will deteriorate with larger differences between query and reference (Supplementary Fig. 28a). We further observed good integration with 4,000 HVGs, even when 25% of genes were missing from the query data, conveying that the model would be robust if the overall number of shared genes is large (for example, 4,000 HVGs, Supplementary Fig. 28b).

Evaluation metrics. Evaluation metrics and their definitions in the current paper were taken from work by Luecken et al.[7], unless specifically stated otherwise.

Entropy of batch mixing. This metric[43] works by constructing a fix similarity matrix for cells. The entropy of mixing in a region of cells with c batches is defined as

$$E = \sum_{i=1}^{c} p_i \log_c(p_i),$$

where $_{pi}$ is defined below as

$$p_i = \frac{\text{no. cells with batch } i \text{ in the region}}{\text{no. cells in the region}}.$$

Next, we define U, a uniform random variable on the cell population. Let B_U be the frequencies of 15 nearest neighbors for the cell U in batch x. We report the entropy of this variable and then average across $T = 100$ measurements of U. To normalize the entropy of the batch mixing score between 0 and 1, we set the base of the logarithm to the number of batches c.

Average silhouette width. Silhouette width measures the relationship between within-cluster distances of a cell and between-cluster distances of that cell to the closest cluster. In general, an ASW score of 1 implies clusters that are well separated, an ASW score of 0 implies overlapping clusters, and an ASW score of −1 implies strong misclassification. When we use the ASW score as a measure of biological variance, we calculate it on cell types in the following manner:

$$\text{ASW}_c = \frac{\text{ASW} + 1}{2},$$

where the final score is already scaled between 0 and 1. Therefore larger values correspond to denser clusters. In contrast to the ASW_c score, we also calculate an ASW score on batches within cell clusters to obtain a measure for batch-effect removal. In this case, we again scale but also invert the ASW score to have a consistent metric comparison:

$$\text{ASW}_b = 1 - \text{abs}(\text{ASW}).$$

A higher final score here implies better mixing and therefore a better batch-removal effect.

Normalized mutual information. We use NMI to compare the overlap of two different cell type clusterings. In detail, we computed a Louvain clustering on the latent representation of the data and compared it to the latent representation itself in a cell type-wise manner. To obtain scores between 0 and 1, the overlap was scaled using the mean of entropy terms for cell type and cluster labels. Therefore an NMI score of 1 corresponds to a perfect match and good conservation of biological variance, whereas an NMI score of 0 corresponds to uncorrelated clustering.

Adjusted Rand index. This metric considers correct clustering overlaps as well as counting correct disagreements between two clusterings. Again, similar to NMI, cell type labels in the integrated dataset are compared with Louvain clustering. The adjusted Rand index score is normalized between 0 and 1, where 1 corresponds to good conservation of biological variance and 0 corresponds to random labeling.

Principal-component regression. In contrast to principal-component analysis (PCA), we calculate a linear regression R with respect to the batch label onto each principal component. The total variance (Var) explained by the batch variable can then be formulated as follows:

$$\text{Var}(X|B) = \sum_{i=1}^{N} \text{Var}(X|\text{PC}_i) \cdot R^2(\text{PC}_i|B),$$

where X is the data matrix, B is the batch label, and N is the number of principal components (PC).

Graph connectivity. For this metric, we calculate a subset kNN graph $G(N_c, E_c)$ for each cell type label c, such that each subset only contains cells from the given label. The total graph connectivity score can then be calculated as follows:

$$gc = \frac{1}{|C|}\sum_{c\in C}\frac{|\mathrm{LCC}(G(N_c, E_c))|}{|N_c|},$$

where C is the set of cell type labels, $|\mathrm{LCC}()|$ is the number of nodes in the largest connected component of the graph, and $|N_c|$ is the number of nodes with the given cell type label. This means that we check if the graph representation of the latent representation connects all cells with the same cell type label. Therefore, a score of 1 would imply that all cells with the same cell type label are connected, which would further indicate good batch mixing. A graph in which no cells are connected would result in a score of 0.

Isolated label F_1. We defined isolated labels as cell type labels that are present in the least number of batches. If there are multiple isolated labels, we simply take the mean of each score. To determine how well those cell types are separated from other cell types in the latent representation, we first determine the cluster with the largest number of an isolated label. Subsequently, an F_1 score of the isolated label against all other labels within that cluster is computed, where the F_1 score is defined as follows:

$$F_1 = 2\frac{\mathrm{precision}\cdot\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}.$$

This results in a score between 0 and 1 once again, where 1 implies that all cells with the isolated label are captured in the cluster.

Isolated label silhouette. For this metric, we use ASW_c, defined above, but only on the isolated label subset of the latent representation. Scaling and meaning of the score are the same as described for ASW. If there are multiple isolated labels, we average over each score similar to the isolated labeled F_1 score.

kNN accuracy. We first compute the 15 nearest neighbors of each cell in the data. We then compute the ratio of the correct cell type annotations inside those 15 neighbors. This cell-wise score is then averaged over all cell types separately and then averaged over all remaining scores again to obtain a single kNN-accuracy score between 0 and 1. A higher kNN-accuracy score corresponds to better preservation of local cell type purity. This metric was inspired by a similar metric used in scANVI.

Visualization of integration scores. To compare performances of different models, we designed an overview table (inspired by Saelens et al.[72]) that displays individual integration scores as circles and aggregated scores as bars. Each individual score is minimum–maximum scaled to improve visual comparison of different models and then averaged into aggregated scores by category (batch correction and biological conservation). Finally, an overall score is calculated as a weighted sum of batch correction and bio-conservation, considering a ratio of 40:60, respectively. When shown, reference and query times are not considered in the calculation of aggregated scores. Moreover, these time values are scaled together to allow direct comparison. The overall ranking of each model, for each score, is represented by the color scheme.

Datasets. All cell type labels and metadata were obtained from original publications unless specifically stated otherwise below.

Brain data. The mouse brain dataset is a collection of four publicly available scRNA-seq mouse brain studies[1,33–35], for which additional information on cerebral regions was provided. We obtained the raw count matrix from Rosenberg et al.[34] under GEO accession ID GSE110823, the annotated count matrix from Zeisel et al.[35] from http://mousebrain.org (file name L5_all.loom, downloaded on 9 September 2019) and count matrices per cell type from Saunders et al.[33] from http://dropviz.org (DGE by region section, downloaded on 30 August 2019). Data from mouse brain tissue sorted by flow cytometry (myeloid and non-myeloid cells, including the annotation file annotations_FACS.CSV) from TM were obtained from https://figshare.com (retrieved 14 February 2019). We harmonized cluster labels via fuzzy string matching and attempted to preserve the original annotation as far as possible. Specifically, we annotated ten major cell types (neuron, astrocyte, oligodendrocyte, oligodendrocyte precursor cell, endothelial cell, brain pericyte, ependymal cell, olfactory ensheathing cell, macrophage and microglia). In the case of Saunders et al.[33], we facilitated the additional annotation data table for 585 reported cell types (annotation.BrainCellAtlasSaundersversion2018.04.01.TXT retrieved from http://dropviz.org on 30 August 2019. Among these, some cell types were annotated as 'endothelial tip', 'endothelial stalk' and 'mural'. We examined the subset of the Saunders et al.[33] dataset as follows: we used Louvain clustering (default resolution parameter, 1.0) to cluster, followed by gene expression profiling via the rankgenesgroups function in scanpy. Using marker gene expression, we assigned microglia (*C1qa*), oligodendrocytes (*Plp1*), astrocytes (*Gfap*, *Clu*) and endothelial cells (*Flt1*) to the subset. Finally, we applied scran[73] normalization and log(counts + 1) to transform count matrices. In total, the dataset consists of 978,734 cells.

Pancreas. Five publicly available pancreatic islet datasets[74–78], with a total of 15,681 cells in raw count matrix format were obtained from the Scanorama[42] dataset, which has already assigned its cell types using batch-corrected gene expression by Scanorama. The Scanorama dataset was downloaded from http://scanorama.csail.mit.edu/data.tar.gz. In the preprocessing step, raw count datasets were normalized and log transformed by scanpy preprocessing methods. Preprocessed data were used directly for the pipeline of scArches. One thousand HVGs were selected for training the model.

The human cell landscape. The HCL dataset was obtained from https://figshare.com/articles/HCL_DGE_Data/7235471. Raw count matrix data for all tissues were aggregated. A total of 277,909 cells were selected and processed using the scanpy Python package. Data were normalized using size factor normalization such that every cell had 10,000 counts and then log transformed. Finally, 5,000 HVGs were selected as per their average expression and dispersion. We used processed data directly for training scArches at the pre-training phase.

The mouse cell atlas. The mouse cell atlas dataset was obtained from https://figshare.com/articles/HCL_DGE_Data/7235471. Raw count matrix data for all tissues were aggregated together. A total of 150,126 cells were selected and processed using the scanpy Python package. Homologous genes were selected using BioMart 100 before merging with HCL data. Data were normalized together with HCL as explained before.

Immune data. The immune dataset consists of ten human samples from two different tissues: bone marrow and peripheral blood. Data from bone marrow samples were retrieved from Oetjen et al.[36], while data from peripheral blood samples were obtained from 10x Genomics (https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3), Freytag et al.[37], Sun et al.[38] and Villani et al.[79]. Details on the retrieval location of datasets, the different protocols used and ways in which samples were chosen for analysis can be found in Luecken et al.[7]. We performed quality control separately for each sample but adopted a common strategy for normalization: all samples for which count data were available were individually normalized by scran pooling[73]. This excludes data from Villani et al.[79], which included only TPM values. All datasets were log+1 transformed in scanpy[80]. Cell type labels were harmonized starting from existing annotations (Oetjen et al.[36]) to create a consistent set of cell identities. Well-known markers of cell types were collected and used to extend annotation to samples for which they were not previously available. When necessary, subclustering was performed to derive more precise labeling. Finally, cell populations were removed if no label could be assigned. Four thousand HVGs were selected for training.

Endocrine pancreas. The raw dataset of pancreatic endocrinogenesis ($n = 22{,}163$)[45] is available at the GEO under accession number GSE132188. We considered a subset of 2,000 HVGs for training. Cell type labels were obtained from an adata object provided by the authors of scVelo[46].

CITE-seq. We obtained three publicly available datasets from 10x Genomics, already curated and preprocessed as described in the totalVI study[32]. These data include '10k PBMCs from a Healthy Donor—Gene Expression and Cell Surface Protein' (PBMC, 10k (CITE-seq)[81]), '5k PBMCs from a healthy donor with cell surface proteins (v3 chemistry)' (PBMC, 5k (CITE-seq)[82]) and '10k PBMCs from a Healthy Donor (v3 chemistry)' (PBMC, 10k (RNA-seq)[57,83,84]). Reference data included 14 proteins, and 4,000 HVGs were selected for training.

COVID-19. The COVID-19 dataset along with its metadata was downloaded from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE1459261 and https://github.com/zhangzlab/covid_balf. The dataset that was used in this paper includes $n = 62{,}469$ cells. Data from lungs[52–54], PBMCs[37–39] and bone marrow[36] were later merged with those from COVID-19 samples. Data were normalized using scanpy, and 2,000 HVGs were selected for training the model. Cell type labels were obtained from the original study.

Tabula Muris Senis. The TM Senis dataset with GEO accession number GSE132042 is publicly available at https://figshare.com/projects/Tabula_Muris_Senis/64982. The dataset contains 356,213 cells with cell type, tissue and method annotation. We normalized the data using size factor normalization with 10,000 counts for each cell. Next, we log+1 transformed the dataset and selected 5,000 HVGs according to their average expression and dispersion. All preprocessing steps were carried out using the scanpy Python package. In this study, we used a combination of sequencing technology and time point as batch covariates.

Benchmarks. *Full integration methods.* We ran PCA with 20 principal components on the final results from Seurat, Scanorama and mnnCorrect to be comparable (similar approach as described in ref. [31]) when computing metrics to deep learning models, which had a latent representation of size 10–20.

- Harmony: we used the Harmony Matrix function from the Harmony package. We provided the function with a PCA matrix with 20 principal components on the gene expression matrix.
- Scanorama: we used the correct_scanpy function from the Scanorama package with default parameters.
- Seurat: we applied Seurat as in the walkthrough (https://satijalab.org/seurat/v3.1/integration.html) with default parameters.
- Liger: we used the Liger method as in the walkthrough (https://github.com/welch-lab/liger/blob/master/vignettes/walkthrough_pbmc.pdf). We used $k=20$, $\lambda=5$ and resolution = 0.4 with other default parameters. We only scaled data as we had already preprocessed data.
- Conos: we followed the Conos tutorial at https://htmlpreview.github.io/?https://raw.githubusercontent.com/kharchenkolab/conos/master/doc/walkthrough.html. Unlike the tutorial, we used our own preprocessed data for better comparisons. We used PCA space with parameters $k=30$, k.self = 5, ncomps = 30, matching.method = 'mNN' and metric = 'angular' to build the graph. We set the resolution to 1 to find communities. Finally, we saved the corrected pseudo-PCA space with 20 components.
- mnnCorrect: we used the mnnCorrect function from the scran package with default parameters.

Cell type-classification methods.

- Seurat: we followed the walkthrough (https://satijalab.org/seurat/v3.1/integration.html) and used reciprocal PCA for dimension reduction. As described in the original publication[48], we examined projection scores and assigned cells with the lowest 20% of values to be 'unknown'.
- SVM: we fitted an SVM model from the scikit-learn library to the reference data and classified query cells. We assigned cells with uncertainty probability greater than 0.7 as 'unknown'.
- Logistic regression: we fitted logistic regression from the scikit-learn library to the reference data and predicted query labels.

All these methods were tested on a machine with one eight-core Intel i7-9700KQ CPU addressing 32 GB RAM and one Nvidia GTX 1080 ti (12 GB) addressing 12 GB VRAM.

Model output. Throughout this paper, all low-dimensional representations were obtained using the latent space of scArches models. The output of scArches models will be confounded with condition variables not fit for data-integration applications but best for imputation or denoising scenarios.

Cell type annotation. To classify labels for the query dataset, we trained a weighted kNN classifier on the latent-space representation of the reference dataset. For each query cell c, we extracted its kNNs (N_c). We computed the standard deviation of the nearest distances:

$$\text{s.d.}_{c,N_c} = \sqrt{\frac{\sum_{n \in N_c} (\text{dist}(c,n))^2}{k}},$$

where dist(c,n) is the Euclidean distance of the query cell c and its neighbors n in the latent space. Next, we applied the Gaussian kernel to distances using

$$D_{c,n,N_c} = e^{-\frac{\text{dist}(c,n)}{(2\text{s.d.}_{c,N_c})^2}}.$$

Next, we computed the probability of assigning each label y to the query cell c by normalizing across all adjusted distances using

$$p(Y = y|X = c, N_c) = \frac{\sum_{i \in N_c} I(y^{(i)} = y) \cdot D_{c,n_i,N_c}}{\sum_{j \in N_c} D_{c,n_j,N_c}},$$

where $y^{(i)}$ is the label of ith nearest neighbor and I is the indicator function. Finally, we calculated the uncertainty u for each cell c in the query dataset using its set of closest neighbors in the reference dataset (N_c). We defined the uncertainty u_{c,y,N_c} for a query cell c with label y and N_c as its set of nearest neighbors as

$$u_{c,y,N_c} = 1 - p(Y = y|X = c, N_c).$$

We reported cells with more than 50% uncertainty as unknown to detect out-of-distribution cells with new labels, which do not exist in the training data. Therefore, we labeled each cell c in the query dataset as follows:

$$\hat{y}'_c = \text{argmin}_y \, u_{c,y,N_c}$$

$$\hat{y}_c = \begin{cases} \hat{y}'_c & \text{if } u_{c,\hat{y}'_c,N_c} \leq 0.5 \\ \text{unknown} & \text{o.w.} \end{cases}$$

Protein imputation. For scArches totalVI, missing proteins for RNA-seq-only data were imputed by conditioning query cells as being in the other batches in the reference with protein data. It is possible to impute based on a specific batch or average across all batches. In the example in the paper, the average version was used.

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

Data availability

All datasets used in the paper are public, referenced and downloadable at https://github.com/theislab/scarches-reproducibility.

Code availability

Software is available at https://github.com/theislab/scarches. The code to reproduce the results is available at https://github.com/theislab/scarches-reproducibility.

References

67. Mirza, M. & Osindero, S. Conditional generative adversarial nets. Preprint at https://arxiv.org/abs/1411.1784 (2014).
68. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. Preprint at http://arxiv.org/abs/1312.6114 (2013).
69. Doersch, C. Tutorial on variational autoencoders. Preprint at https://arxiv.org/abs/1606.05908 (2016).
70. Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. in *Advances in Neural Information Processing Systems* (eds. Cortes, C. et al.) **28**, 3483–3491 (Curran Associates, 2015).
71. 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).
72. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).
73. Lun, A. T. L., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17**, 75 (2016).
74. Baron, M. et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.* **3**, 346–360 (2016).
75. Muraro, M. J. et al. A single-cell transcriptome atlas of the human pancreas. *Cell Syst.* **3**, 385–394 (2016).
76. Segerstolpe, Å. et al. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab.* **24**, 593–607 (2016).
77. Lawlor, N. et al. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. *Genome Res.* **27**, 208–222 (2016).
78. Grün, D. et al. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell* **19**, 266–277 (2016).
79. Villani, A.-C. et al. Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* **356**, eaah4573 (2017).
80. Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
81. 10x Genomics. 10k PBMCs from a Healthy Donor, Gene Expression and Cell Surface Protein https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_protein_v3 (2018).
82. 10x Genomics. 5k Peripheral Blood Mononuclear Cells (PBMCs) from a Healthy Donor with Cell Surface Proteins (v3 Chemistry) https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.2/5k_pbmc_protein_v3? (2019).
83. 10x Genomics. 10k PBMCs from a Healthy Donor (v3 Chemistry) https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3?
84. Mould, K. J. et al. Airspace macrophages and monocytes exist in transcriptionally distinct subsets in healthy adults. *Am. J. Respir. Crit. Care Med.* **203**, 946–956 (2020).

Acknowledgements

We are grateful to all members of the Theis laboratory. M.L. is grateful for valuable feedback from A. Wolf and financial support from the Joachim Herz Stiftung. This work was supported by the BMBF (01IS18036A and 01IS18036B), by the European Union's Horizon 2020 research and innovation program (grant 874656) and by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI (ZT-I-PF-5-01) and sparse2big (ZT-I-0007) and Discovair (grant 874656), all to F.J.T. For the purpose of open access, the authors have applied a CC BY public copyright licence to any author accepted manuscript version arising from this submission.

Author contributions

M.L. conceived the project with contributions from F.J.T. and Z.A. M.L., M.N., M.W., S.R. and M.K. implemented models and analyzed data. M.B. curated the mouse brain dataset.

M.I. designed visualizations and curated the immune dataset. M.L. and M.D.L. analyzed the COVID-19 dataset with help from A.V.M. A.G. and N.Y. contributed by adapting scArches with scvi-tools. F.J.T. supervised the research. All authors wrote the manuscript.

Competing interests

F.J.T. reports ownership interest in Cellarity. N.Y. is an advisor and/or has equity in Celsius Therapeutics and Rheos Medicines. The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-021-01001-7.

Correspondence and requests for materials should be addressed to F.J.T.

Peer review information *Nature Biotechnology* thanks Dana Pe'er and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature research

Corresponding author(s): Fabian Theis

Last updated by author(s): Jun 15, 2021

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	The software is downloadable from pip and is available at https://github.com/theislab/scarches. The guides for using software are available https://scarches.readthedocs.io/en/latest/
Data analysis	https://github.com/theislab/scarches-reproducibility

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

All of the datasets analyzed in this manuscript are public and published in other papers. We referenced them in the manuscript and they are downloadable at https://github.com/theislab/scarches-reproducibility

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No experiments in study
Data exclusions	No experiments in study
Replication	No experiments in study
Randomization	No experiments in study
Blinding	No experiments in study

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Appendix B

scGen predicts single-cell perturbation responses.

Nature Methods (2019).

This is a pre-proof version of the article Nature Methods following peer review.

(iii) Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. **"scGen predicts single-cell perturbation responses."** Nature methods 16.8 (2019): 715-721.1).

The article is also available online at:

`https://www.nature.com/articles/s41592-019-0494-8`

Generative modeling and latent space arithmetics predict single-cell perturbation response across cell types, studies, and species

M. Lotfollahi[1], F. Alexander Wolf[1†] & Fabian J. Theis[1,2‡]
1 Helmholtz Center Munich – German Research Center for Environmental Health, Institute of Computational Biology, Neuherberg, Munich, Germany.
2 Department of Mathematics, Technische Universität München, Munich, Germany.
† alex.wolf@helmholtz-muenchen.de ‡ fabian.theis@helmholtz-muenchen.de

Abstract

Accurately modeling cellular response to perturbations is a central goal of computational biology. While such modeling has been proposed based on statistical, mechanistic and machine learning models in specific settings, no generalization of predictions to phenomena absent from training data (out-of-sample) has yet been demonstrated. Here, we present scGen, a model combining variational autoencoders and latent space vector arithmetics for high-dimensional single-cell gene expression data. In benchmarks across a broad range of examples, we show that scGen accurately models dose and infection response of cells across cell types, studies and species. In particular, we demonstrate that scGen learns cell type and species specific response implying that it captures features that distinguish responding from non-responding genes and cells. With the upcoming availability of large-scale atlases of organs in healthy state, we envision scGen to become a tool for experimental design through *in silico* screening of perturbation response in the context of disease and drug treatment.

Introduction

Single-cell transcriptomics has become an established tool for unbiased profiling of complex and heterogeneous systems [1, 2]. The generated datasets are typically used for explaining phenotypes through cellular composition and dynamics. Of particular interest are the dynamics of single cells in response to perturbations, be it to dose [3], treatment [4, 5] or knockout of genes [6–8]. Although advances in single-cell differential expression analysis [9, 10] have enabled the identification of genes associated with a perturbation, generative modeling of perturbation response takes a step further in that it enables the generation of data *in silico*. The ability to generate data that cover phenomena not seen during training is particularly challenging and referred to as out-of-sample prediction.

While dynamic mechanistic models have been suggested for predicting low-dimensional quantities that characterize cellular response [11, 12], such as a scalar measure of proliferation, they face fundamental problems. These models cannot easily be formulated in a data-driven way and require temporal resolution of the experimental data. Due to the typically small number of time points available, parameters are often hard to identify. Resorting to linear statistical models for modeling perturbation response [6, 8], by contrast, leads to low predictive power for the complicated nonlinear effects that single-cell data display. In contrast, neural network models do not face these limits.

Recently, such models have been suggested for the analysis of single-cell RNA-seq data [13–17]. In particular, generative adversarial networks (GANs) have been proposed for simulating single cell differentiation through so-called latent space interpolation [16]. While providing an interesting alternative to established pseudotemporal ordering algorithms [18], this analysis does not demonstrate the capability of GANs for out-of-sample prediction. The use of GANs for the harder task of out-of-sample prediction is hindered by fundamental difficulties: (1) GANs are hard to train for structured high-dimensional data, leading to high-variance predictions with large errors in extrapolation, and

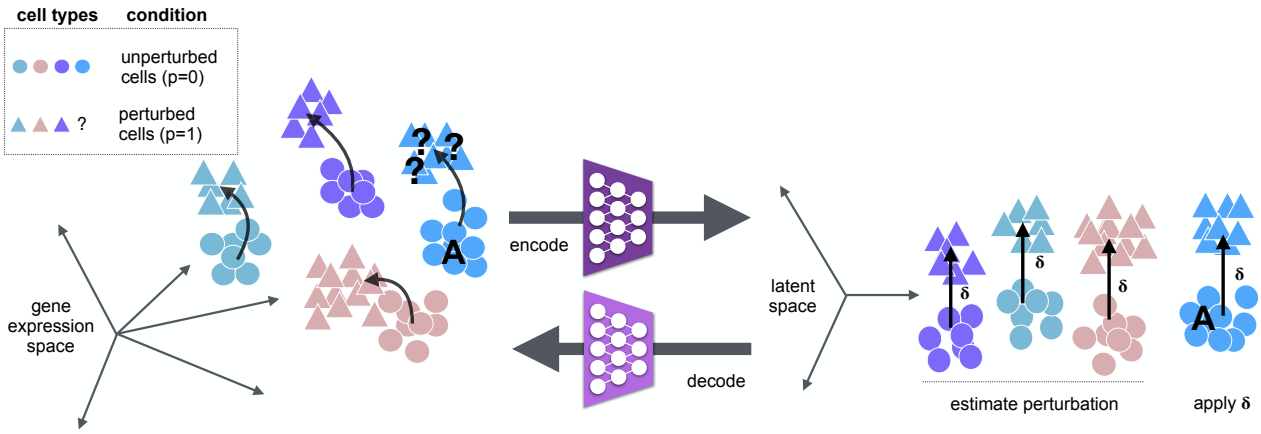


Figure 1 | scGen, a method to predict single-cell perturbation response. Given a set of observed cell types in control and stimulation, we aim to predict the perturbation response of a new cell type A (blue) by training a model that learns to generalize the response of the cells in the training set. Within scGen, the model is a variational autoencoder, and the predictions are obtained using vector arithmetics in the autoencoder's latent space. Specifically, we project gene expression measurements into a latent space using an encoder network and obtain a vector δ that represents the difference between perturbed and unperturbed cells from the training set in latent space. Using δ, unperturbed cells of type A are linearly extrapolated in latent space. The decoder network then maps the linear latent space predictions to highly nonlinear predictions in gene expression space.

(2) GANs do not allow for the direct mapping of a gene expression vector x on a latent space vector z, making it difficult or impossible to generate a cell with a set of desired properties. In addition, for structured data, GANs have not yet shown advantages over the simpler variational autoencoders (VAE) [19] (Supplemental Note 1.1).

To overcome the problems inherent to GANs, we built scGen, which is based on a VAE combined with vector arithmetics, with an architecture adapted for single-cell RNA-seq data. For the first time, scGen enables predictions of dose and infection response of cells for phenomena absent from training data across cell types, studies, and species. In a broad benchmark, it outperforms other potential modeling approaches, such as linear methods, conditional variational autoencoders (CVAE) [20] and style-transfer GANs. The benchmark of several generative neural network models should present a valuable resource for the community showing opportunities and limitations for such models when applied to scRNA-seq data. scGen is based on Tensorflow [21] and on the single-cell analysis toolkit Scanpy [22].

Results

scGen accurately predicts single-cell perturbation response out-of-sample

High-dimensional scRNA-seq data is typically assumed to be well parametrized by a low-dimensional manifold arising from the constraints of the underlying gene regulatory networks. Current algorithms mostly focus on characterizing the manifold using graph-based techniques [25, 26] in the space spanned by a few principal components. More recently, the manifold has been modeled using neural networks [13–17]. As in other application fields [27, 28], in the latent spaces of these models, the manifolds display astonishingly simple properties, such as approximately linear axes of variation for latent variables explaining a major part of the variability in the data. Hence, linear extrapolations of the low-dimensional manifold could in principle capture variability related to perturbation and other covariates (Supplemental Note 1.2, Supplemental Figure 1).

Let every cell i with expression profile x_i be characterized by a variable p_i, which represents a discrete attribute across the whole manifold, such as perturbation, species, or batch. To start with, we assume only two conditions 0 (unperturbed) and 1 (perturbed). Let us further consider the

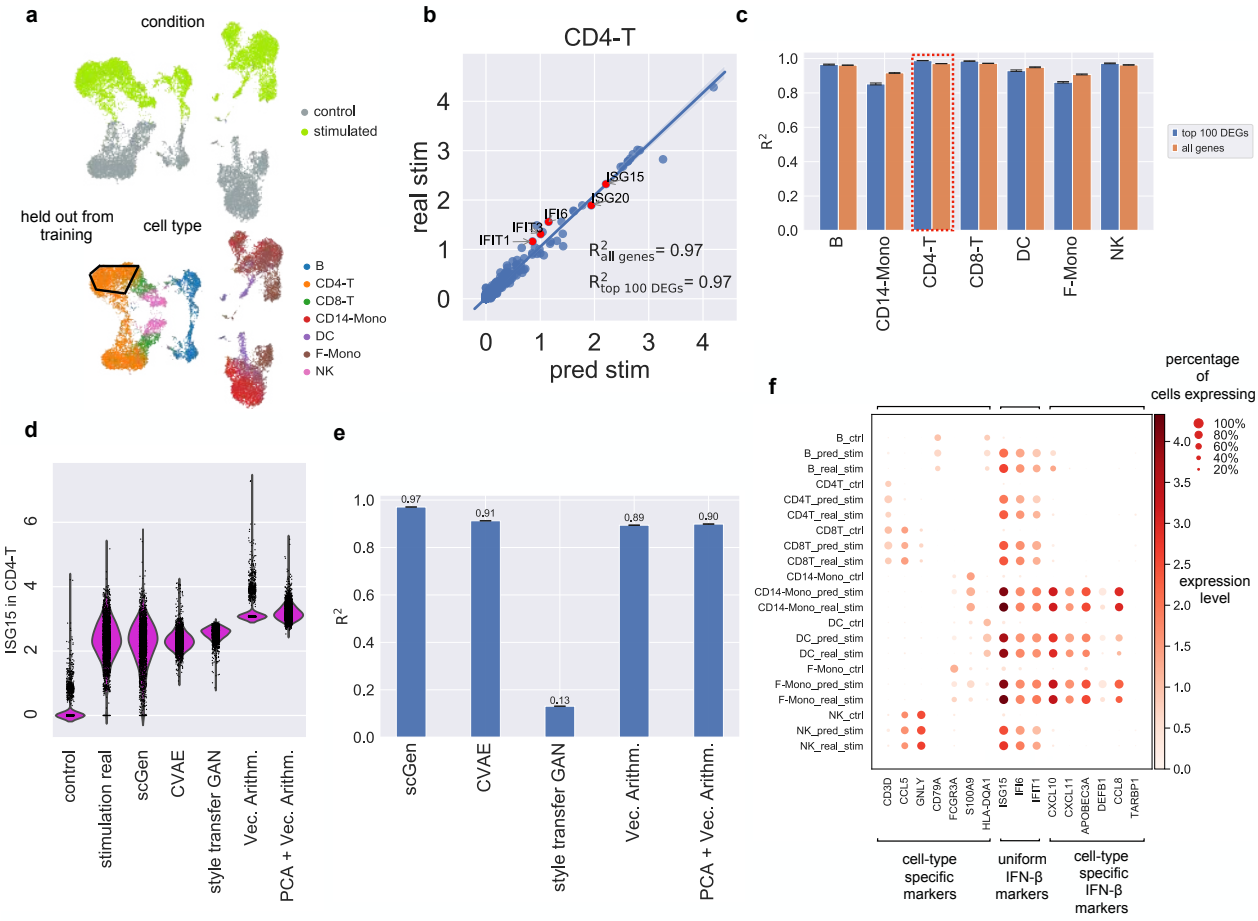


Figure 2 | scGen accurately predicts single-cell perturbation response out-of-sample. **(a)** UMAP visualization [23] of the distributions of condition, cell type, and data split for the prediction of IFN-β stimulated CD4-T cells from PBMCs from Kang *et al.* [3] (n=18,868). **(b)** Mean gene expression of 6,998 genes between scGen predicted and real stimulated CD4-T cells together with top five upregulated differentially expressed genes. **(c)** Comparison of R^2 values for mean gene expression between real and predicted cells for the seven different cell types of the study. **(d)** Distribution of *ISG15*: the top uniform marker (response) gene to IFN-β [24] between control, predicted and real stimulated cells of scGen when compared with other potential prediction models. **(e)** Similar comparison of R^2 values to predict unseen CD4-T stimulated cells. **(f)** Dot plot for comparing control, real, and predicted stimulation when predicting on seven cell types from Kang *et al.*

conditional distribution $P(x_i|z_i, p_i)$, which assumes that each cell x_i comes from a low-dimensional representation z_i in condition p_i. We use a VAE to model $P(x_i|z_i, p_i)$ in its dependence on z_i and vector arithmetics in the VAE's latent space to model the dependence on p_i (Figure 1).

Equipped with this, consider a typical extrapolation problem. Assume cell type A exists in the training data only in the unperturbed ($p = 0$) condition. From that, we predict the latent representation of perturbed cells ($p = 1$) of cell type A using $\hat{z}_{i,A,p=1} = z_{i,A,p=0} + \delta$, where $z_{i,A,p=0}$ and $\hat{z}_{i,A,p=1}$ denote the latent representation of cells with cell type A in conditions $p = 0$ and $p = 1$, respectively, and δ is the difference vector of means between cells in the training set in condition 0 and 1 (Supplemental Note 1.3). From the latent space, scGen maps predicted cells to high-dimensional gene expression space using the generator network estimated while training the VAE.

To demonstrate the performance of scGen, we apply it to published human peripheral blood mononuclear cells (PBMCs) stimulated with interferon (IFN-β) [3] (Supplemental Note 2). As a first test, we study the predictions for stimulated CD4-T cells that are held out during training (Figure 2a). Comparing with the real data, scGen's prediction of mean expression correlates well with the ground-truth across all genes (Figure 2b), in particular, those strongly responding to IFN-β and hence most

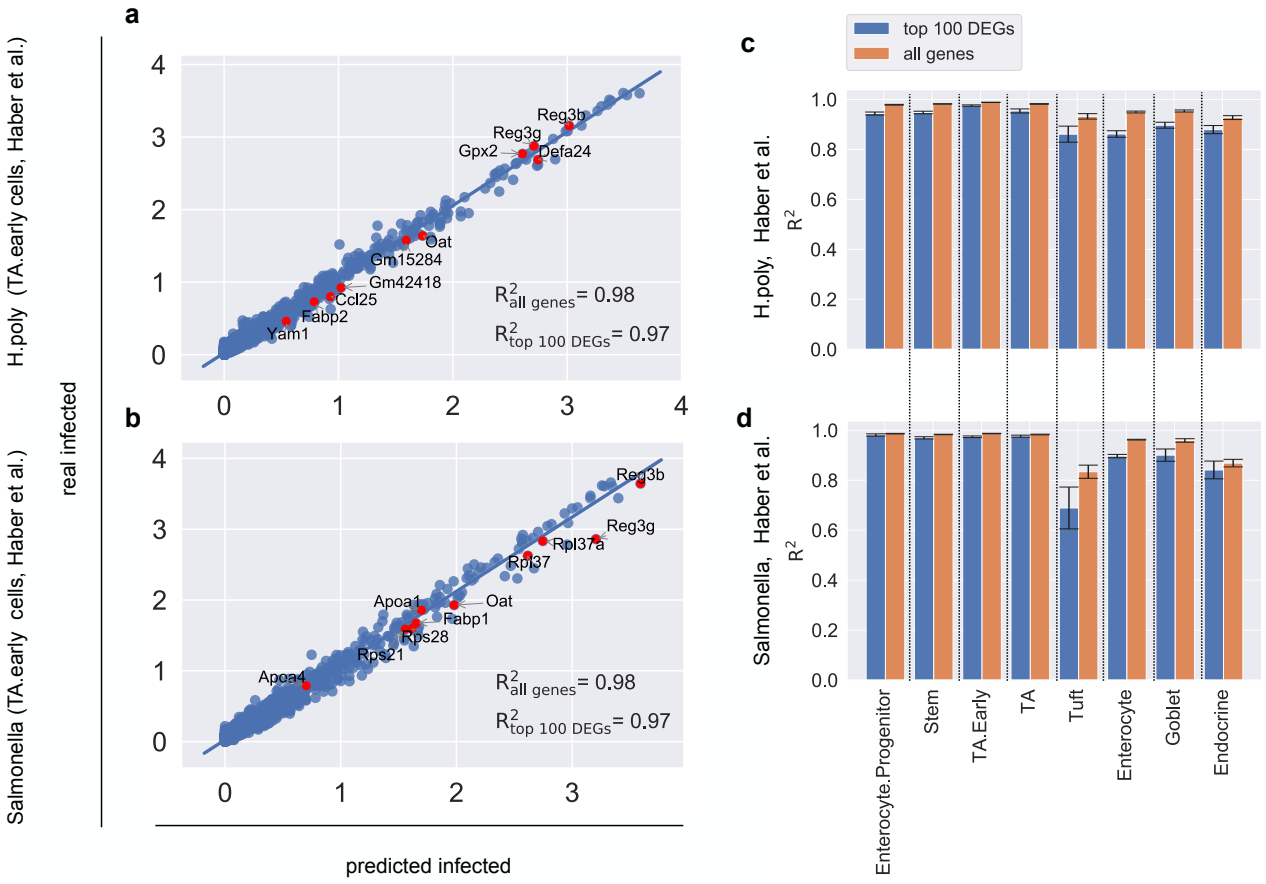


Figure 3 | scGen models infection response in two datasets of intestinal epithelial cells. (a-b) Prediction of early transit-amplifying (TA.early) cells from two different small intestine datasets from Haber *et al.* [4] infected with *Salmonella* (n=5,010) and helminth *Heligmosomoides polygyrus (H.poly)* (n=5,951) after 2 and 10 days, respectively. The mean gene expression of 7,000 genes between infected and predicted cells for different cell types shows how scGen transforms control to predicted perturbed cells in a way that the expression of the top five upregulated and downregulated differentially expressed genes are similar to real infected cells. **(c-d)** Comparison of R^2 values for mean gene expression between real and predicted cells for all the cell types in two different datasets illustrates that scGen performs well for all cell types in different scenarios.

differentially expressed (labeled genes in Figure 2b and inset "top 100 DEGs"). To evaluate generality, we trained six other models holding out each of the six major cell types present in the study. Figure 2c shows that our model accurately predicts all other cell types (average $R^2 = 0.948$ and $R^2 = 0.936$ for all and top 100 differentially expressed genes (DEGs), respectively). Moreover, the distribution of the strongest regulated IFN-β response gene *ISG15* as predicted by scGen not only provides a good estimate for the mean but well predicts the full distribution (Figure 2d, all genes in Supplemental Figure 2a).

scGen outperforms alternative modeling approaches

Aside from scGen, we studied further natural candidates for modeling a conditional distribution that is able to capture perturbation response. We benchmark scGen against four of these candidates, including two generative neural networks and two linear models. The first of these models is the conditional variational autoencoder (CVAE) (Supplemental Note 3, Supplemental Figure 3a, [20]), which has recently been adapted to preprocessing, batch-correcting and differential testing of single-cell data [13]. However, it has not been shown to be a viable approach for out-of-sample predictions, even though, formally, it readily admits the generation of samples from different conditions. The second class of models are style transfer GAN (Supplemental Note 4, Supplemental Figure 3b), which are commonly used for unsupervised image-to-image translation [29, 30]. In our implementation,

such a model is directly trained for the task of transferring cells from one condition to another. The adversarial training is highly flexible and does not require an assumption of linearity in a latent space. In contrast to other propositions for mapping biological manifolds using GANs [31], style transfer GANs are able to handle unpaired data, a necessity for their applicability to scRNA-seq data. We also tested ordinary GANs combined with vector arithmetics similar to Ghahramani *et al.* [16]. However, for the fundamental problems outlined above, we were not able to produce any meaningful out-of-sample predictions using this setup. In addition to the nonlinear generative models, we tested simpler linear approaches based on vector arithmetics in gene expression space and the latent space of principal component analyses (PCA). Applying the competing models to the PBMC dataset, we observe that all other models fail to predict the distribution of *ISG15* (all genes in Supplemental Figure 2), in stark contrast to scGen's performance (Figure 2d). The predictions from the CVAE and the style transfer GAN are less accurate compared to scGen's predictions and linear models even yield incorrect negative values (Figure 2e, Supplemental Figure 2, Supplemental Note 5).

A likely reason for why CVAE fails to provide more accurate out-of-sample predictions, is that it disentangles perturbation information from its latent space representation z in the bottleneck layer. Hence, the layer does not capture non-trivial patterns linking perturbation to cell type. A likely reason for that the style transfer GAN is incapable for achieving the task is its attempt to match two high-dimensional distributions, with much more complex models involved than in the case of scGen, which are notoriously more difficult to train. Some of these arguments can be better understood when inspecting the latent space distribution embeddings of the generative models. As the CVAE completely strips off all perturbation-variation, its latent space embedding does not allow to distinguish perturbed from unperturbed cells (Supplemental Figure 4a). In contrast to CVAE representations, the scGen (VAE) latent space representation captures both information for condition and cell type (Supplemental Figure 4c), reflecting that non-trivial patterns across condition and cell type variability are stored in the bottleneck layer. Hyperparameters (Supplemental Note 6) and architectures are reported in Supplemental Tables 1 (scGen), 2 (style transfer GAN), and 3 (CVAE).

scGen predicts both response shared among cell types and cell type-specific response

Depending on shared or individual receptors, signaling pathways, and regulatory networks, the perturbation response of a group of cells may result in expression-level changes that are shared across all cell types or unique to only some. Predicting both types of responses is essential for understanding mechanisms involved in disease progression as well as adequate drug dose predictions [32, 33]. scGen is able to capture both types of responses after stimulation by IFN-β when any of the cell types in the data is held out during training and subsequently predicted (Figure 2f). For this, we use previously reported marker genes [24] of three different kinds: cell type-specific markers independent of the perturbation such as *CD79A* for B cells, perturbation-response specific genes like *ISG15*, *IFI6*, *IFIT1* expressed in all cell types, and genes of cell type specific responses to the perturbation such as *APOBEC3A* for DC cells. Across the seven different held out perturbed cell types present in the data of Kang *et al.*, scGen consistently makes good predictions not only of unperturbed and shared perturbation effects but also for cell type specific ones. These findings not only hold for these few selected marker genes, but for the top 10 most cell-type specific responding genes and to the top 500 DEGs between stimulated and control cells (Supplemental Figure 5 a-b). The linear model, by contrast, fails capturing cell type-specific differential expression patterns (Supplemental Figure 5 c-d).

scGen robustly predicts response of intestinal epithelial cells to infection

We evaluate scGen's predictive performance for two datasets from Haber *et al.* [4] (Supplemental Note 2) using the same network architecture as for the data of Kang *et al.*. These datasets consist of intestinal epithelial cells after *Salmonella* or *Heligmosomoides polygyrus (H.poly)* infections, respectively. scGen shows good performance for early transit-amplifying (TA.early) cells after in-

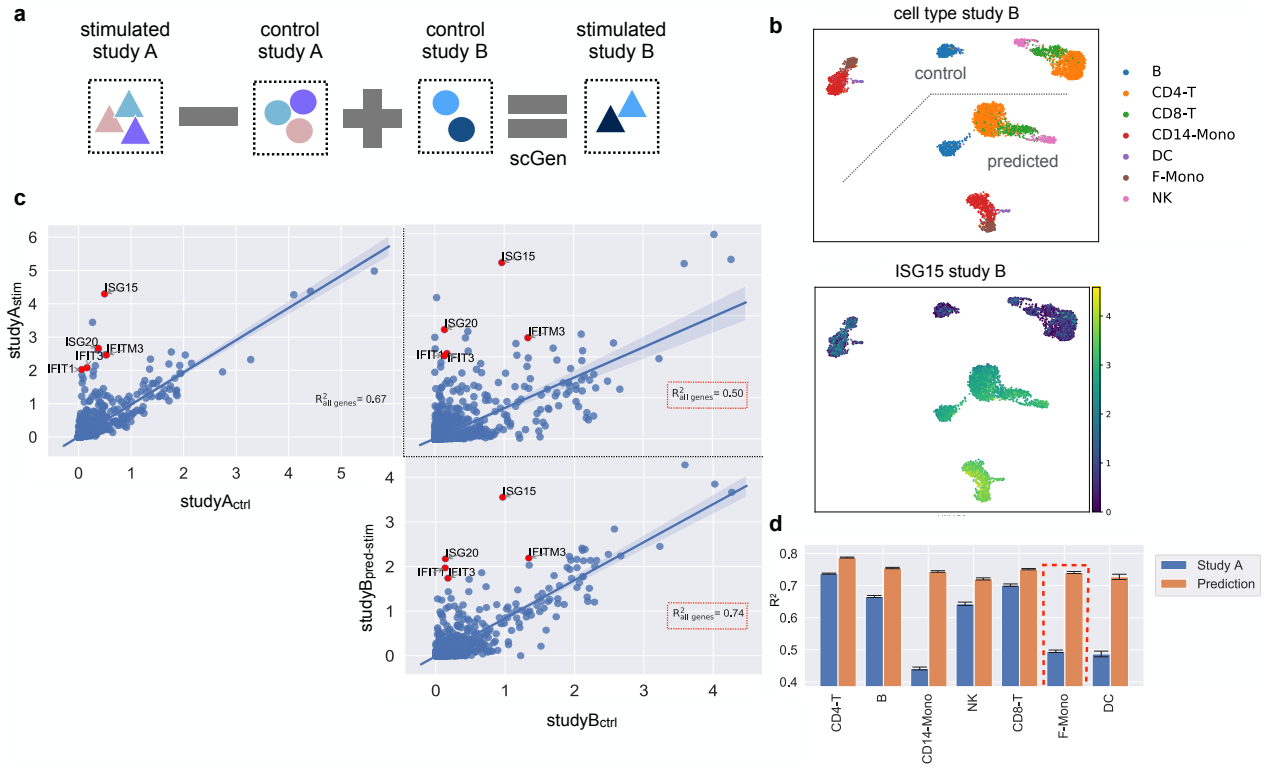


Figure 4 | scGen accurately predicts single-cell perturbation across different studies. (a) scGen can be used to translate the effect of stimulation trained in study A to how stimulated cells in study B would look, given a control sample set. **(b)** Cell types for control and predicted stimulated cells for study B (Zheng *et al.* [34]) in two conditions where ISG15, the top IFN-β response gene, is only expressed in stimulated cells. **(c)** Average expression between: control and stimulated F-Mono cells from study A (upper left), control from study B and stimulated cells from study A (upper right), and control from study B and predicted stimulated cells for study B (lower right). Red points denote top five differentially expressed genes for F-Mono cells after stimulation in study A. **(d)** Comparison of R^2 values highlighted in panel c for F-Mono and all other cell types.

fection with *H.poly* and *Salmonella* (Figure 3a,b), predicting each condition with high precision ($R^2_{\text{all genes}} = 0.98$ and $R^2_{\text{all genes}} = 0.98$, respectively). Figure 3c,d depicts similar analyses for both datasets and all occurring cell types — as before, the predicted ones being held out during training — indicating that scGen's prediction accuracy is robust across most cell types. Again, we show that these results generalize to the top 10 most cell-type specific responding genes out of 500 DEGs (Supplemental Figure 6).

To understand when scGen starts to fail at making meaningful predictions, we trained it on the PBMC data of Kang *et al.*, but now with more than one cell type held out. This study shows that scGen's predictions are robust when holding out several dissimilar cell types (Supplemental Figure 7a-b) but start failing when training on data that only contains information about the response of one highly dissimilar cell type (see CD4-T predictions in Supplemental Figure 7c).

Finally, similar to what has been shown by [16] for differentiation of epidermal cells, we cannot only generate fully responding cell populations, but also intermediary cell states between two conditions. Here, we do so for the IFN-β stimulation and the *Salmonella* infection (Supplemental Note 8, Supplemental Figure 8).

scGen enables cross-study predictions

In order to be applicable to broad cell atlases such as the Human Cell Atlas [35], scGen needs to be robust against batch effects and generalize across different studies. For this, we consider a scenario with two studies: study A, where cells have been observed in two biological conditions, e.g., control and stimulation, and study B with the same setting as study A but only in the control condition.

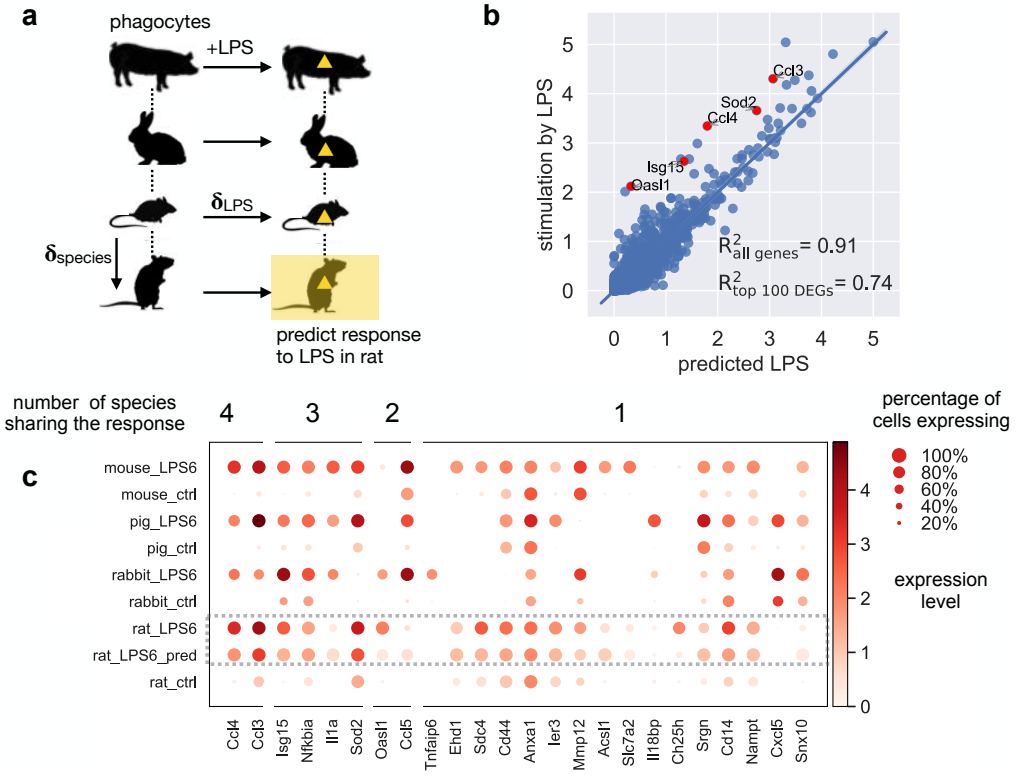


Figure 5 | scGen predicts perturbation response across different species. (a) Prediction of unseen rat LPS phagocytes, on control and stimulated scRNA-seq from mouse, rabbit, and pig by Hagai *et al.* [5] (n=77,642). **(b)** Mean gene expression of 6,619 one-to-one orthologs between species for predicted LPS rat cells plotted against real LPS, whereas highlighted points represent top five differentially expressed genes after LPS stimulation in the real data. **(c)** Dot plot of top 10 differentially expressed genes after LPS stimulation in each species, with numbers indicating how many species have those responsive genes among their top 10 differentially expressed genes.

By jointly encoding the two datasets, scGen provides a model for predicting the perturbation for study B (Figure 4a) by estimating the study effect as the linear perturbation in the latent space. To demonstrate this, we use as study A the PBMC dataset from Kang *et al.* and as study B another PBMC study consisting of 2,623 cells that are available only in the control condition (Zheng *et al.* [34]). After training the model on data from study A, we use the trained model to predict how the PBMCs in study B would respond to stimulation with IFN-β.

As a first sanity check, we show that *ISG15* is also expressed in the prediction of stimulated cells based on the Zheng *et al.* (Figure 4b). This observation holds for all other differential genes associated with the stimulation, which we show for *FCGR3A+*-Monocytes (F-Mono) (Figure 4c): The predicted stimulated F-Mono cells correlate more strongly with the control cells in their study than with stimulated cells from study A while still expressing differentially expressed genes known from study A. Similarly, predictions for other cell types yield a higher correlation than the direct comparison with study A (Figure 4d).

scGen predicts single-cell perturbation across species

In addition to learning the variation between two conditions, e.g. health and disease for a species, scGen can be used to predict across species. We trained a model on a scRNA-seq dataset by Hagai *et al.* [5] comprised of bone marrow-derived mononuclear phagocytes from mouse, rat, rabbit, and pig perturbed with lipopolysaccharide (LPS) after six hours. Similar to what we did previously, we held out the rat LPS cells from the training data (Figure 5a).

In contrast to previous scenarios, now, two global axes of variation exist in the latent space associated with species and stimulation, respectively.

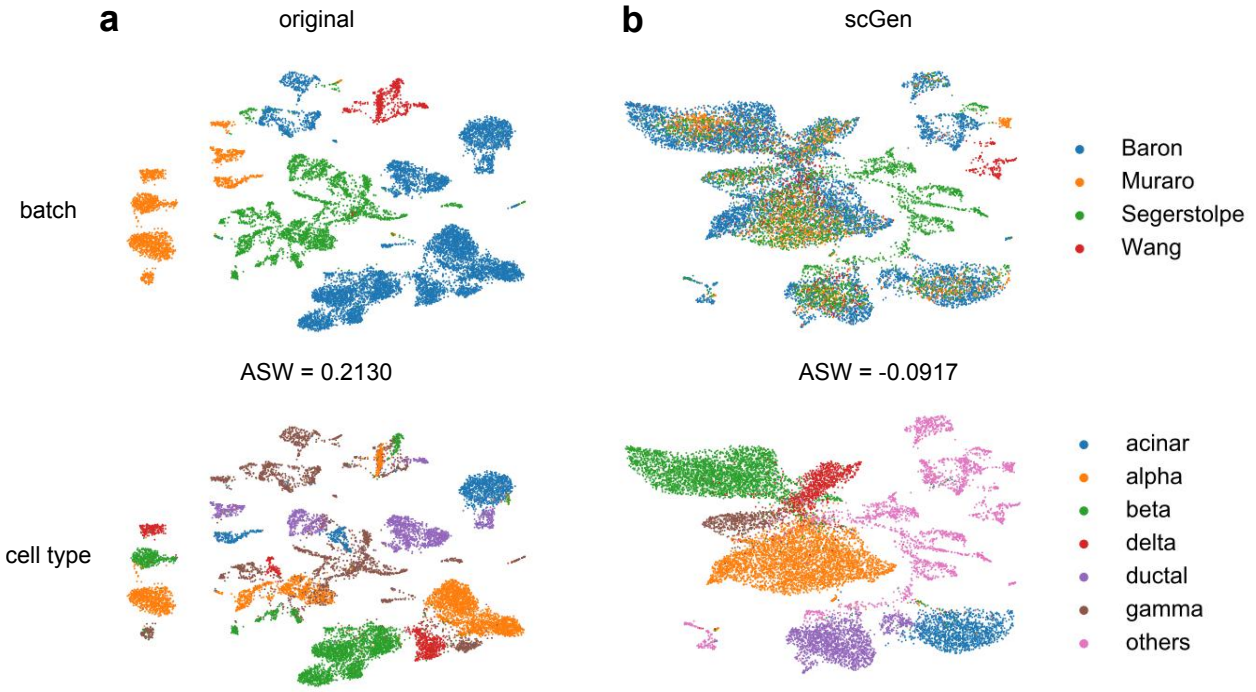


Figure 6 | scGen removes batch effects. (a) UMAP visualization of four technically diverse pancreatic datasets (n=14,693) with their corresponding batch and cell types. We report average silhouette width (ASW) for batches in the original data (ASW = 0.2130, lower is better for batch-effect evaluation). **(b)** Data corrected by scGen mixes shared cell types from different studies while preserving study specific cell types as independent (ASW = -0.0917).

Based on this, we have two latent difference vectors: δ_{LPS}, which encodes the variation between control and LPS cells, and $\delta_{species}$, which accounts for differences between species. Next, we predict rat LPS cells using $z_{i,\text{rat,LPS}} = \frac{1}{2}(z_{i,\text{mouse,LPS}} + \delta_{\text{species}} + z_{i,\text{rat,control}} + \delta_{\text{LPS}})$ (Figure 5b). This equation takes an average of the two alternative ways of reaching rat LPS cells (Figure 5a). All other predictions along the major linear axes of variation also yield plausible results for stimulated rat cells (Supplemental Figure 9).

In addition to the species-conserved response of a few upregulated genes, e.g. *Ccl3* and *Ccl4*, cells also display species specific responses. For example, *Il1a* is highly upregulated in all species except rat. Strikingly, scGen correctly identifies the rat cells as non-responding with this gene. Only the fraction of cells expressing *Il1a* increases at a low expression level (Figure 5c). Based on these early demonstrations, we hope to predict cellular response to treatment in human based on data from untreated humans and different treated animal models.

scGen removes batch effects

Let us now show that scGen is able to efficiently correct for batch effects. To evaluate scGen's batch correction capability, we merged four pancreatic datasets [36–39] (Figure 6a). We train scGen on these data and define a source and destination batch and compute a difference vector δ_{batch} between the source and the destination batch. To remove the batch effects from the destination batch, we add the learned δ_{batch} to the latent representation of the cells in the destination batch (Figure 6b). Using the cell type labels from the studies we observe a homogeneous overlap. A comparison with four existing batch removal methods (Supplemental Figure 10) shows that scGen performs as well as the other methods [24, 40–42]. To further evaluate batch removal ability of our model on a larger dataset, we merged eight different mouse single cell atlases comprised of 114,600 cells from different organs [43–50]. As expected, the homogeneity of the data increased after batch correction (Supplemental Figure 11).

Discussion

By adequately encoding the original expression space in a latent space, scGen achieves simple, near-to-linear mappings for highly non-linear sources of variation in the original data, which explain a large portion of the variability in the data, associated with, for instance, perturbation, species, or batch. This allows to use scGen in several contexts including perturbation prediction response for unseen phenomena across cell types, study and species, for interpolating cells between conditions and for batch effect removal.

While we showed proof-of-concept for *in silico* predictions of cell type- and species-specific cellular response, in the present work, scGen has been trained on relatively small datasets, which only reflect subsets of biological and transcriptional variability. While we demonstrated scGen's predictive power in these settings, a trained model cannot be expected to be predictive beyond the domain of the training data. To gain confidence in predictions, one needs to make realistic estimates for prediction errors by holding out parts of the data with known ground truth that are representative for the task. It is important to realize that such a procedure arises naturally when applying scGen in an alternating iteration of experiments, retraining based on new data and *in silico* prediction. By design, such strategies are expected to yield highly performing models for specific systems and perturbations of interest. It is evident that such strategies could readily exploit the upcoming availability of large-scale atlases of organs in healthy state, such as the Human Cell Atlas [35].

We demonstrated that scGen is able to learn cell type and species specific response. To be able to do so, the model needs to capture features that distinguish weakly from strongly responding genes and cells. Building biological interpretations of these features, for instance, along the lines of Ghahramani *et al.* [16] or Way and Greene [51], could help in understanding the differences between cells that respond to certain drugs and cells that do not respond, which is often crucial for understanding patient response to drugs [52].

Code availability

Code is available at `https://github.com/theislab/scGen`.

Data availability

All data is available from the original publications and linked on `https://github.com/theislab/scGen`.

Author Contributions

M.L. performed the research, implemented the models and analyzed the data. F.A.W. conceived the project with contributions from M.L. and F.J.T.. F.A.W. and F.J.T. supervised the research. All authors wrote the manuscript.

Acknowledgments

We are grateful to all members of the Theis lab, in particular, D.S. Fischer for early comments on predicting across species. M.L. is grateful for valuable feedback of L. Haghverdi regarding batch-effect removal. F.A.W. acknowledges discussions with N. Stranski on responding and non-responding cells and support by the Helmholtz Postdoc Programme, Initiative and Networking Fund of the Helmholtz Association. F.J.T. gratefully acknowledges support by the Helmholtz Association within the project "Sparse2Big" and by the German Research Foundation (DFG) within the Collaborative Research Centre 1243, Subproject A17.

During the work on the project, we became aware of reference [51], which suggests to study differences between cancer subtypes in the latent space of a VAE trained on bulk RNA-seq data from the Cancer Genome Atlas. The authors also demonstrate biological interpretability of these differences. In the weeks before submission of the manuscript, we became aware of the preprint [53], which addresses out-of-sample prediction in its revised version, but not in the context of single cell RNA-seq data.

Supplemental Notes

Contents

Supplemental Note 1: Models and theoretical background

Supplemental Note 1.1: Variational autoencoders

A variational autoencoder is a neural network consisting of an encoder and a decoder similar to classical autoencoders. Unlike classical autoencoders, however, VAEs are able to generate new data points. The mathematics underlying VAEs aslo differs from that of classical autoencoders. The difference is that the model maximizes the likelihood of each sample x_i (more accurately, maximizes the log evidence (sum of log likelihoods of all x_i)) in the training set under a generative process as formulated in Equation (1):

$$P(x_i|\theta) = \int P(x_i|z_i;\theta)P(z_i|\theta)dz_i. \tag{1}$$

where θ is a model parameter which in our model corresponds to a neural network with its learnable parameters, and z_i is a latent variable. The key idea of a VAE is to sample latent variables z_i that are likely to produce x_i and using those to compute $P(x_i|\theta)$ [54]. We approximate the posterior distribution $P(z_i|x_i,\theta)$ using the variational distribution $Q(z_i|x_i,\phi)$ which is modeled by a neural network with parameter ϕ, called the inference network (the encoder). Next, we need a distance measure between the true posterior $P(z_i|x_i,\theta)$ and the variational distribution. To compute such a distance we use the Kullback–Leibler ($\mathbb{KL}$) divergence between $Q(z_i|x_i,\phi)$ and $P(z_i|x_i,\theta)$, which yields:

$$\mathbb{KL}(Q(z_i|x_i,\phi)||P(z_i|x_i,\theta)) = \mathbb{E}_{Q(z_i|x_i,\phi)}[\log Q(z_i|x_i,\phi) - \log P(z_i|x_i,\theta)]. \tag{2}$$

Now, we can derive both $P(x_i|\theta)$ and $P(x_i|z_i,\theta)$ by applying Bayes' rule to $P(z_i|x_i,\theta)$, which results in:

$$\mathbb{KL}(Q(z_i|x_i,\phi)||P(z_i|x_i,\theta)) = \mathbb{E}_{Q(z_i|x_i,\phi)}[\log Q(z_i|x_i,\phi) - \log P(z_i|\theta) - \log P(x_i|z_i,\theta)] + \log P(x_i|\theta). \tag{3}$$

Finally, by rearranging some terms and exploiting the definition of KL divergence we have :

$$\log P(x_i|\theta) - \mathbb{KL}(Q(z_i|x_i,\phi)||P(z_i|x_i,\theta)) = \mathbb{E}_{Q(z_i|x_i,\phi)}[\log P(x_i|z_i,\theta)] - \mathbb{KL}[Q(z_i|x_i,\phi)||P(z_i|\theta)]. \tag{4}$$

On the left hand side of Equation (4), we have the log-likelihood of the data denoted by $\log P(x_i|\theta)$ and an error term which depends on the capacity of the model. This term ensures that Q is as complex as P and assuming a high capacity model for $Q(z_i|x_i,\phi)$, this term will be zero [54]. Therefore, we will directly optimize $\log P(x_i|\theta)$:

$$\mathbb{E}_{Q(z_i|x_i,\phi)}[\log P(x_i|z_i,\theta)] - \mathbb{KL}[Q(z_i|x_i,\phi)||P(z_i|\theta)]. \tag{5}$$

Equation (4) and (5) are also known as the evidence lower bound (ELBO). In order to maximize the Equation (5), we choose the variational distribution $Q(z_i|x_i,\phi)$ to be a multivariate Gaussian $Q(z_i|x_i) = \mathcal{N}(z_i;\mu_\phi(x_i),\Sigma_\phi(x_i))$ where μ_ϕ and Σ_ϕ are implemented with the encoder neural network and Σ_ϕ is constrained to be a diagonal matrix. The $\mathbb{KL}$ term in Equation (5) can be computed analytically since both prior $(P(z_i|\theta))$ and posterior $(Q(z_i|x_i,\phi))$ are multivariate Gaussian distributions. The integration for the first term in (5) has no closed-form and we need Monte Carlo integration to estimate it. We can sample $Q(z_i|x_i,\phi)$ L times and directly use stochastic gradient descent to optimize Equation (6) as the loss function for every training point x_i from dataset D:

$$Loss(x_i) = \frac{1}{L}\sum_{l=1}^{L} \log P(x_i|z_{i,l},\theta) - \alpha\mathbb{KL}[Q(z_i|x_i,\phi)||P(z_i|\theta)]. \tag{6}$$

Where the hyperparameter (α) controls how much the KL divergence loss contributes to learning. However, the first term in Equation (6) only depends on the parameters of P, without reference to the parameters of variational distribution Q. Therefore, it has no gradient with respect to ϕ to be backpropagated. In order to address this, the *reparameterization trick* [19] has been proposed. This trick works by first sampling from $\epsilon \sim \mathcal{N}(0,I)$ and then computing $z_i = \mu_\phi(x_i) + \Sigma_\phi^{\frac{1}{2}}(x_i) \times \epsilon$. Thus, we can use gradient-based algorithms to optimize Equation (6).

Supplemental Note 1.2: Linearity of the latent space

scGen exploits vector arithmetics in the latent space of VAEs, a technique which assumes the shift (response) induced by stimuli can be modeled linearly. Similar to what has been shown by [55], we empirically demonstrate the linearity of the latent space with respect to biological conditions. In pursuit of this, we design a simple linear classifier based on the difference vector (δ) between two conditions in the latent space. We hypothesize that the δ vector points toward a direction in the latent space where condition 1 increases. Therefore, by moving along the direction of δ we are moving from condition 0 to condition 1. A high-level intuition for this is that the difference vector manipulates cells by adding and removing information to them. Suppose, for example, a dimension of the latent vector corresponds to the degree of the infection in a cell. Increasing that attribute would be as easy as adding the δ vector corresponding to that attribute. In consequence, the dot product of the cells from condition 1 in δ will be approximately greater than zero (or a constant positive value) indicating high similarity. Similarly, the dot product of cells in condition 0 would yield negative values showing low similarity (Supplemental Figure 1a). After finding the difference vector for each condition, including IFN-β from Kang *et al.* [3] and *H.poly* and *Salmonella* infections from Haber *et al.* [4], we demonstrate the histogram of dot product results for the latent representation of all cells with their corresponding difference vectors (Supplemental Figure 1b).

We did another test, calculating $\delta_{\text{stim-k}}$ denoting the difference between stimulated and control cells for cell type k. We also calculated another set of difference vectors, $\delta_{\text{celltype-ij}}$, representing the difference between each of the seven cell types present in the Kang *et al.* dataset, irrespective of condition. Next, we calculated the cosine similarity for each set of previous vectors with δ. Supplemental Figure 1c shows that vectors in the $\delta_{\text{stim-k}}$ set have very high cosine similarity with δ, indicating that they are both oriented toward the same direction with a small angle. However, most of the $\delta_{\text{celltype-ij}}$ vectors have cosine similarity close to zero, which indicates the cell type and condition vectors are different and nearly orthogonal. In order to get an intuition for how unlikely it is to get a high cosine similarity in 100-dimensional vector space, we randomly drew 1000 samples from a 100-dimensional standard normal distribution and calculated pairwise cosine similarity between them (Supplemental Figure 1c, random).

Supplemental Note 1.3: δ vector estimation

In order to estimate δ, first, we extracted all cells for each condition. Next, for each cell type, we upsampled the cell type sizes to be equal to the maximum cell type size for that condition. To further remove the population size bias, we randomly downsampled the condition with a higher sample size to match the sample size of the other condition. Finally, we estimated the difference vector by calculating $\delta = \text{avg}(z_{\text{condition}=1}) - \text{avg}(z_{\text{condition}=0})$, where $z_{\text{condition}=1}$ and $z_{\text{condition}=0}$ denote the latent representation of cells in each condition, respectively.

Supplemental Note 2: Datasets

The Kang *et al.* [3] included two groups of control and stimulated peripheral blood mononuclear cells (PBMCs). We annotated cell types by extracting an average of the top 20 cluster genes from each of eight identified cell types in PMBCs from [34]. Next, the Spearman correlation between each single cell and all 8 cluster averages was calculated, and each cell was assigned to the cell type for which it had a maximum correlation (similar to [3]). After identifying cell types, megakaryocyte cells were removed from the dataset due to the high uncertainty of the assigned labels. Next, the dataset was filtered for cells with minimum 500 expressed genes and genes which were expressed at least in 5 cells. Moreover, we normalized counts per–cell, and the top 6,998 highly variable genes were selected. Finally, we log-transformed the data to facilitate a smoother training procedure. The final data include 18,868 cells.

The Haber *et al.* dataset was comprised of epithelial cell responses to pathogen infection from Haber *et al.* [4]. In this dataset, the responses of intestinal epithelial cells to *Salmonella* and parasitic helminth *Heligmosomoides polygyrus (H.poly)* were investigated. These data included three different conditions: 1,770 *Salmonella*–infected cells; 2,711 cells 10 days after *H.poly* infection and a group of 3,240 control cells. Each data was normalized per–cel, and log-transformed and top 7,000 highly variable genes were selected.

The PBMC dataset from Zheng *et al.* [34] was obtained from `http://cf.10xgenomics.com/samples/cell-exp/1.1.0/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz`. After filtering cells, the data were merged with filtered PBMCs from Kang *et al.* [3]. The megakaryocyte cells were removed from the smaller dataset. Next, the data were normalized, and we selected top 7,000 highly variable genes. The merged dataset was log-transformed and cells from Kang *et al.* (n=16,893) were used for training the model. The remaining 2,623 cells from Zheng *et al.* were used for the prediction.

Pancreatic datasets (n=14,693) were downloaded from `ftp://ngs.sanger.ac.uk/production/teichmann/BBKNN/objects-pancreas.zip`. Comparisons to other batch correction methods were performed similar to [41] with 50 principal components. The data were already preprocessed and directly used for training the model.

Mouse cell atlases data (n=114,600) were obtained from ftp://ngs.sanger.ac.uk/production/teichmann/BBKNN/MouseAtlas.zip. The data were already preprocessed and directly used for training the model.

The LPS dataset [5] was obtained from https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-6754/?query=tzachi+hagai. The data were further filtered for cells, normalized and log-transformed. We used BiomaRt (v84) [56] to find ENSEMBL IDs of the one-to-one orthologs in the other three species with the mouse. In total 6,619 genes were selected from all species for training the model. The final data include 77,642 cells.

Supplemental Note 3: Conditional variational autoencoder

The conditional variational autoencoder (CVAE) [20] is also based on the variational inference framework. In the CVAE setting one can train a model conditioned on two existing biological conditions. We concatenated the condition of every cell with its input (x_i) and latent variable (z_i). At test time, we fed the model with cells in condition 0 and the label of condition 1 (inverse label) to transform the cells to the same cell type as in condition 1 (Supplemental Figure 3**a**).

Supplemental Note 4: Style transfer GAN

The original style transfer model [30] learns to transform images in one visual domain (e.g., domain of all horses) to another domain (e.g., the domain of all zebras). We adapted this to the single cell domain by training a network that receives single cells in condition 0 and transforms them to similar cells with the same cell type but in condition 1. This can be achieved in an adversarial training fashion (Supplemental Figure 3**b**). As shown in Supplemental Figure 3**b**, the model transforms cells in condition 0 to cells in condition 1 via G_{0-1} and then transforms them back to condition 1 using G_{1-0}. There exists a second line of networks which learns to transform cells from condition 1 to 0 and reconstruct them back to condition 0. These two pipelines must work in a way that they can fool two discriminators (one for each condition) which are trained to detect real cells from generated (fake) cells. In order to make the problem setting more constrained, the reconstructions should not highly deviate from the real data according to a distance metric. Moreover, similar networks in both lines share parameters. At test time, one can feed the gene expression profile of all target cells in condition 0 to transform them to condition 1.

Supplemental Note 5: Model comparison

We compare the distribution-matching capability of each model on their estimate of variance and mean for every individual gene. Our model yields the most accurate mean ($R^2_{\text{all genes}} = 0.97$, Supplemental Figure 2a). Notably, applying vector arithmetics in gene expression and PCA space leads the mean of some genes to take invalid negative values and vector arithmetics in gene expression space leaves the variance of top 100 DEGS intact as it was in the real control cells (Supplemental Figure 2d,e). Furthermore, scGen also shows reasonable performance in variance estimation of top 100 DEGs ($R^2_{\text{top 100 DEGs}} = 0.78$) and outperforms all other models (Supplemental Figure 2a).

Supplemental Note 6: Hyperparameters

Here, in this section, we present all of the hyperparameters of the proposed scGen, CVAE, and style transfer GAN used in the paper. These parameters are stated in Tables 1, 2, and 3. We used early-stopping criterion for scGen and CVAE: the training stopped after 20 consecutive epochs with improvement less than a threshold on the validation loss. All architectures use dropout [57]

regularization, batch normalization [58], and Adam optimizer [59]. We used a L2 regularization for pancreas and mouse atlases datasets using a scale of 0.1. In the following, to have more compact tables, we have used some abbreviations, namely, FC stands for "Fully-connected", NoF stands for "Number of Features", BN stands for "Batch Normalization". We used scikit-learn's PCA estimator [60] with 100 principal components for vector arithmetics in PCA space.

Supplemental Note 7: Gene specificity score

In order to bin genes based on diffentrial pattern specificity , we use the following metric for each to score each gene:

$$\text{score}_i = max_{\text{celltype}_j} \mid \overline{\text{lfc}}_{ij} - \text{median}(\overline{\text{lfc}}_{ij}) \mid \tag{7}$$

where $\overline{\text{lfc}}_{ij} = \mid \overline{\text{stim}}_{ij} - \overline{\text{ctrl}}_{ij} \mid$ for cell type j and gene i. We also define $\overline{\text{stim}}_{ij}$ and $\overline{\text{ctrl}}_{ij}$ as the average of all cells in cell type j in gene i for stimulated and control cells, respectively and median($\overline{\text{lfc}}_{ij}$) is the median of all $\overline{\text{lfc}}_{ij}$ for gene i.

Supplemental Note 8: Latent space interpolation

We exemplify the latent space interpolation ability of our model by generating 2,000 intermediary TA (*Salmonella*, Haber *et al.*) and CD4-T (IFN-β, Kang *et al.*) cells. First, we project average control and predicted cells into the latent space and then linearly interpolate 2,000 intermediary points between them. Next, by using the generator network we map back latent intermediary cells into high-dimensional gene expression space (Supplemental Figure 8a-b). One can observe a smooth change of the top five upregulated and downregulated *Salmonella* response genes as we traverse the cell manifold from control toward *Salmonella* cells (Supplemental Figure8c). Similarly, we can see the upregulation of top 10 IFN-β response genes (Supplemental Figure 8d).

Supplemental Note 9: Biased sampling effect

Usually, the condition sizes are not equal leading to a biased δ vector estimation. Moreover, White [55] discovered that by removing the smile vector from female faces, the male attribute was also added. This originates from the sampling bias induced by unequal size of smiling male and female samples. In order to prevent a similar problem, as previously described we balanced cell type and condition sizes before estimating δ. Supplemental Figure 12 depicts the effects of using biased and unbiased δ vector for the prediction of stimulated CD4-T from the Kang *et al.* dataset using PCA and vector arithmetics.

Supplemental Note 10: Evaluations and tests

Silhouette width. We calculated the silhouette width based on the first 50 PCs of the corrected data or the latent space of the algorithm if it did not return corrected data. The Silhouette coefficient for for cell i is defined as:

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

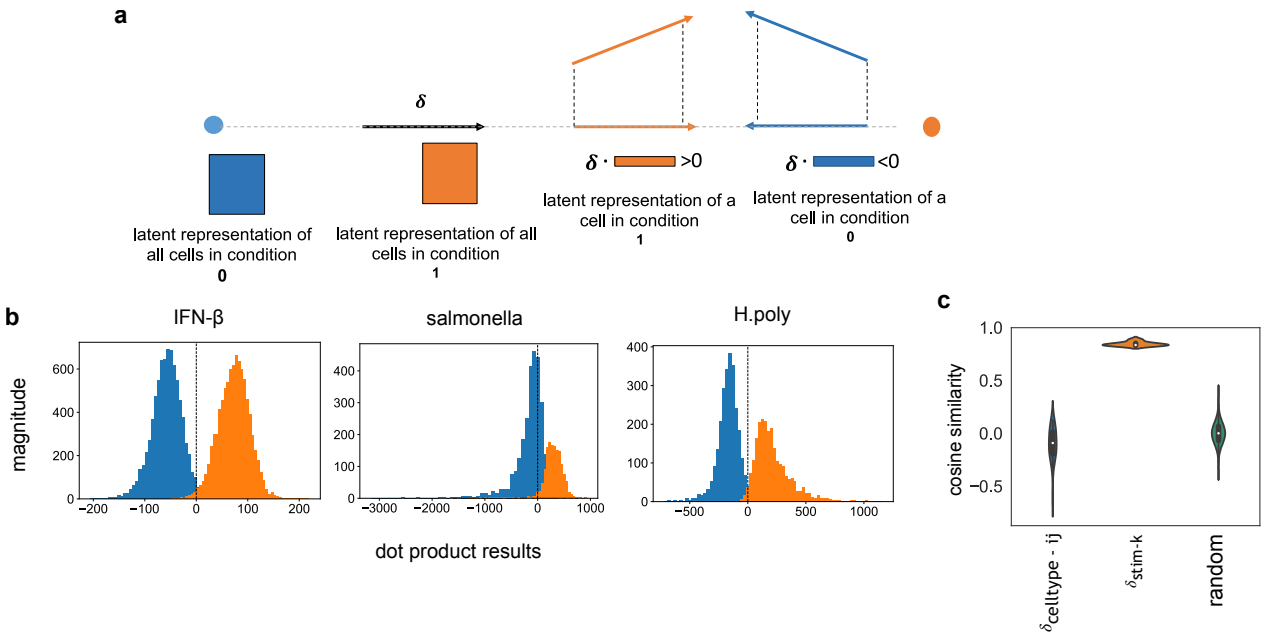
where $a(i)$ and $b(i)$ indicate the mean intra-cluster distance and the mean nearest-cluster distance for sample i, respectively. Instead of cluster labels one can use batch labels to assess batch correction methods. We used the *silhouette_score* function from scikit-learn [60] to calculate the average Silhouette width over all samples.

Error bars. These were computed by randomly resampling 80% of the data with replacement 100 times and recomputing R^2 for each resampled data. The interval represents the mean of R^2 values plus/minus the standard deviation of those 100 R^2 values. We used the mean of 100 R^2 values for the magnitude of each bar. All the R^2 values were calculated by squaring the *rvalue* output of *scipy.stats.linregress* function.

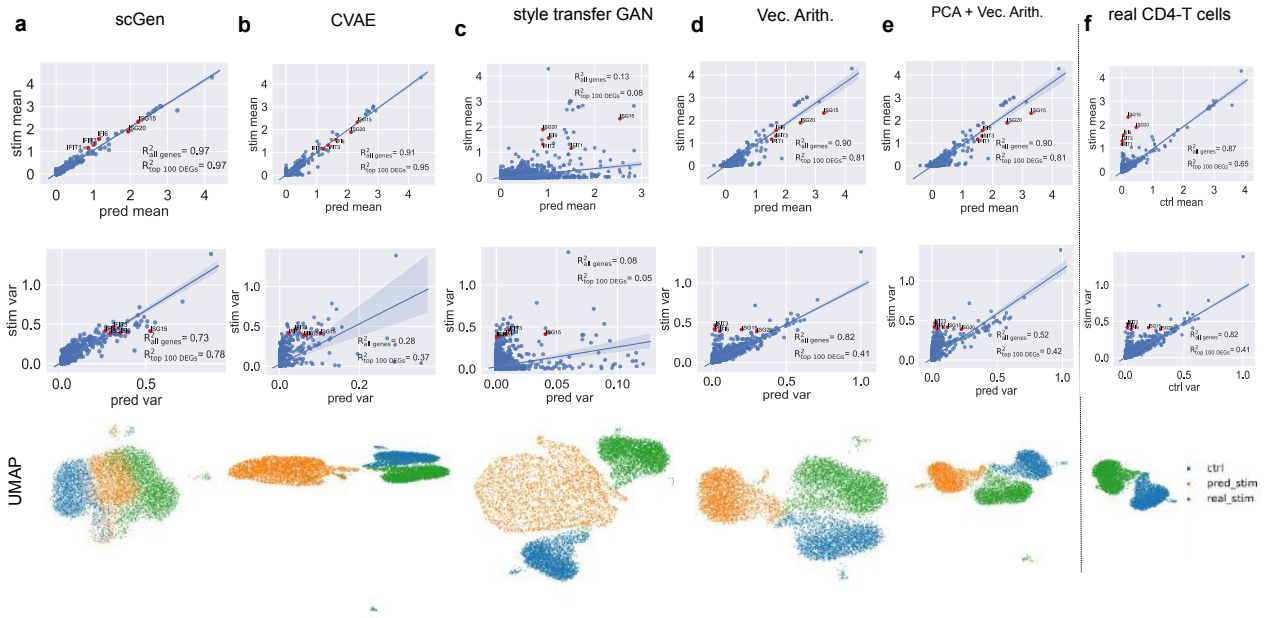
Cosine similarity. The *cosine_similarity* function from scikit-learn was used to compute cosine similarity. This function computes the similarity as the normalized dot product of X and Y defined as:

$$cosine\_similarity(X, Y) = \frac{<X,Y>}{||X||||Y||}$$

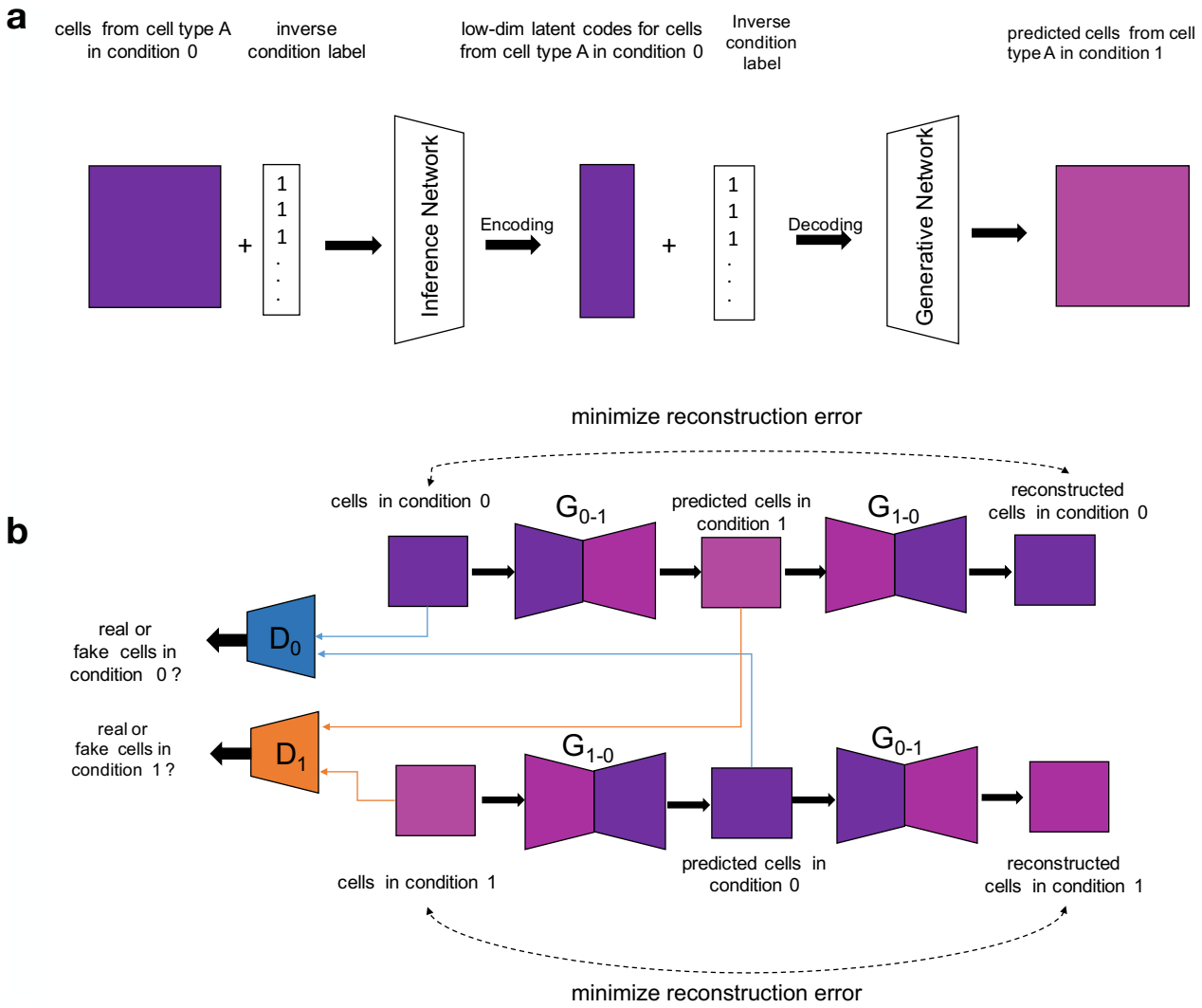
Differential tests. All the differential tests to extract DEGs were performed using scanpy's *rank_genes_groups* function with wilcoxon as the method parameter.



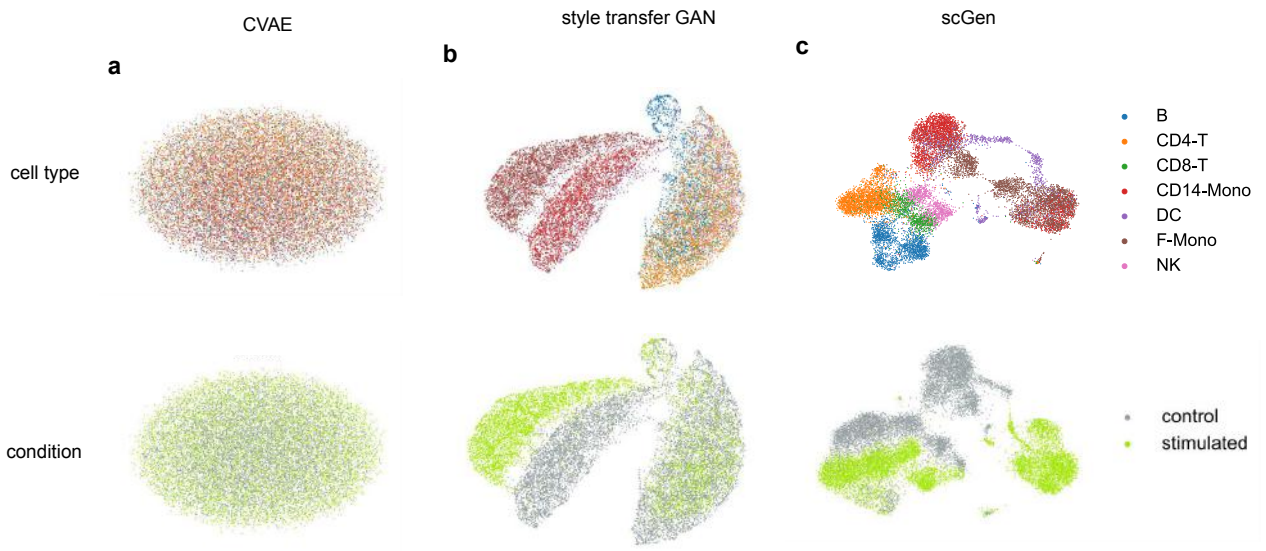
Supplemental Figure 1 | Linearity of the latent space. **(a)** Building a linear classifier based on the dot product between the difference vector (δ) and the latent representation of each cell. **(b)** Dot product results between latent representation of all cells with their corresponding difference vector (δ) for each condition show that two conditions are approximately linearly separable using dot product classifier. **(c)** Cosine similarity of $\delta_{\text{stim-k}}$, $\delta_{\text{celltype-ij}}$ with δ where $\delta_{\text{celltype-ij}} = \text{avg}(z_{\text{celltype}=i}) - \text{avg}(z_{\text{celltype}=j})$ and $\delta_{\text{stim-k}} = \text{avg}(z_{\text{stim, celltype}=\text{k}}) - \text{avg}(z_{\text{ctrl, celltype}=\text{k}})$ for all seven cell types present in the Kang *et al.* dataset (z denotes the latent representation of all cells with the corresponding label). The third violin plot shows pairwise cosine similarity for a set of 1000 random samples from 100–dimensional standard normal distribution.



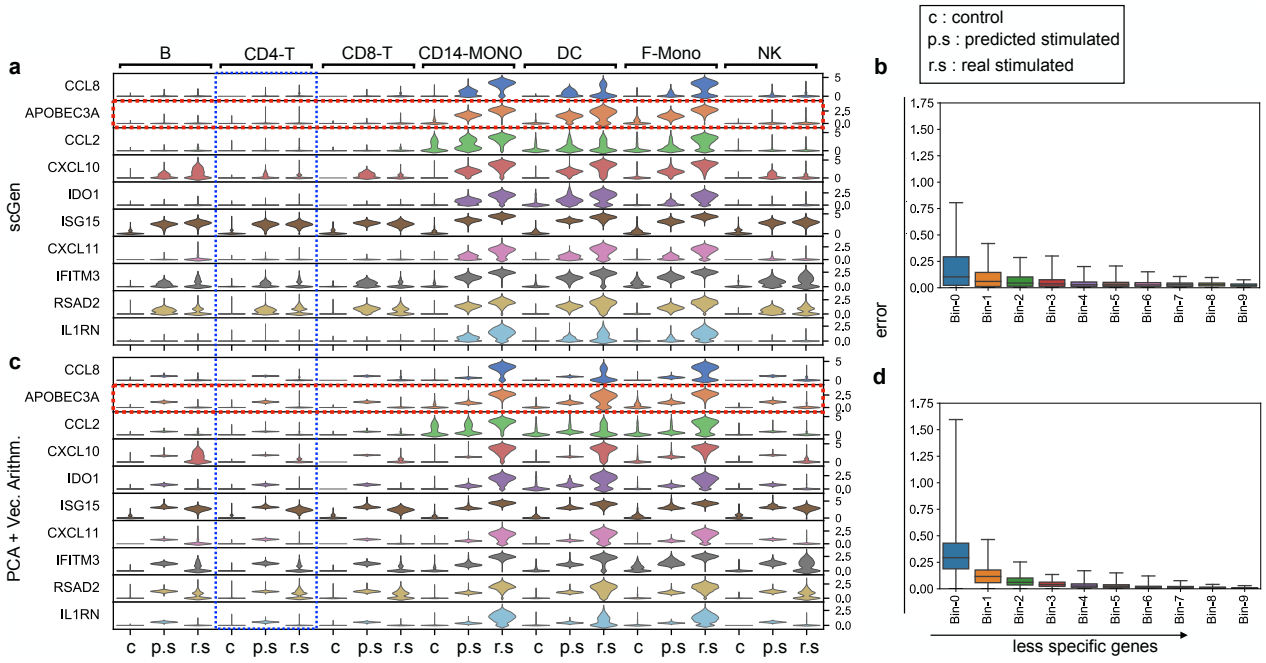
Supplemental Figure 2 | Distribution-matching comparison between different models. **(a-e)** Mean and variance matching comparison between scGen and four alternative models for CD4-T cells shows scGen outperforms other models. Similarly, by comparing UMAP visualizations, one can see the predictions by scGen have more overlap with ground truth cells whereas predictions from other models lie far from real stimulated cells. **(f)** Ground truth mean and variance between control and stimulated CD4-T cells.



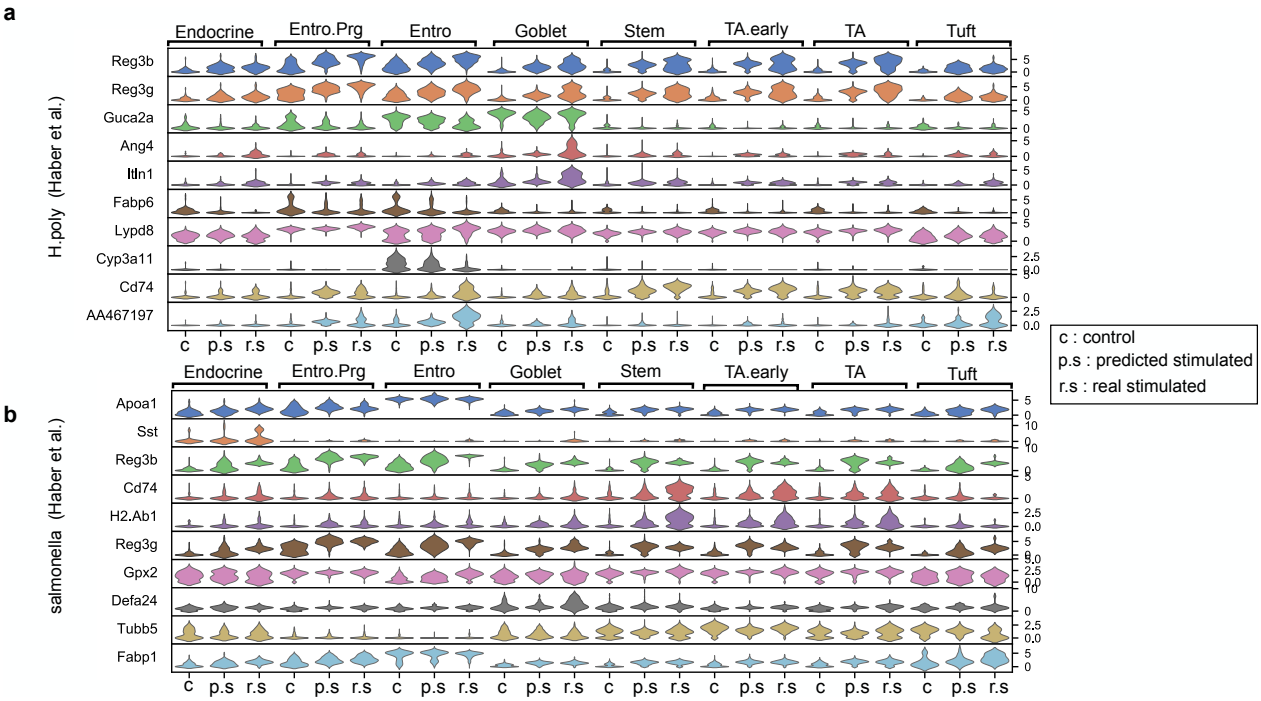
Supplemental Figure 3 | Graphical pipeline of two alternative approaches to predict unseen single–cell perturbations. (a) CVAE pipeline at test time to predict unseen condition. In order to predict cells in condition 1, we feed all cells present in condition 0 with inverse label 1 concatenated (shown with + symbol) to the data matrix. This informs the model that these cells are from condition 1. Therefore, the model changes the condition of input cells from 0 to 1. **(b)** The style transfer GAN to transform one condition to another. This would be possible by learning a joint two-way mapping in an adversarial learning setting. There exist two generators: G_{0-1} which transforms cells from condition 0 to 1, and G_{1-0}, which does the same task but in the reverse direction. Two discriminators, denoted by D_0 and D_1, are trained to detect real from fake cells generated by G_{1-0} and G_{0-1}, respectively.



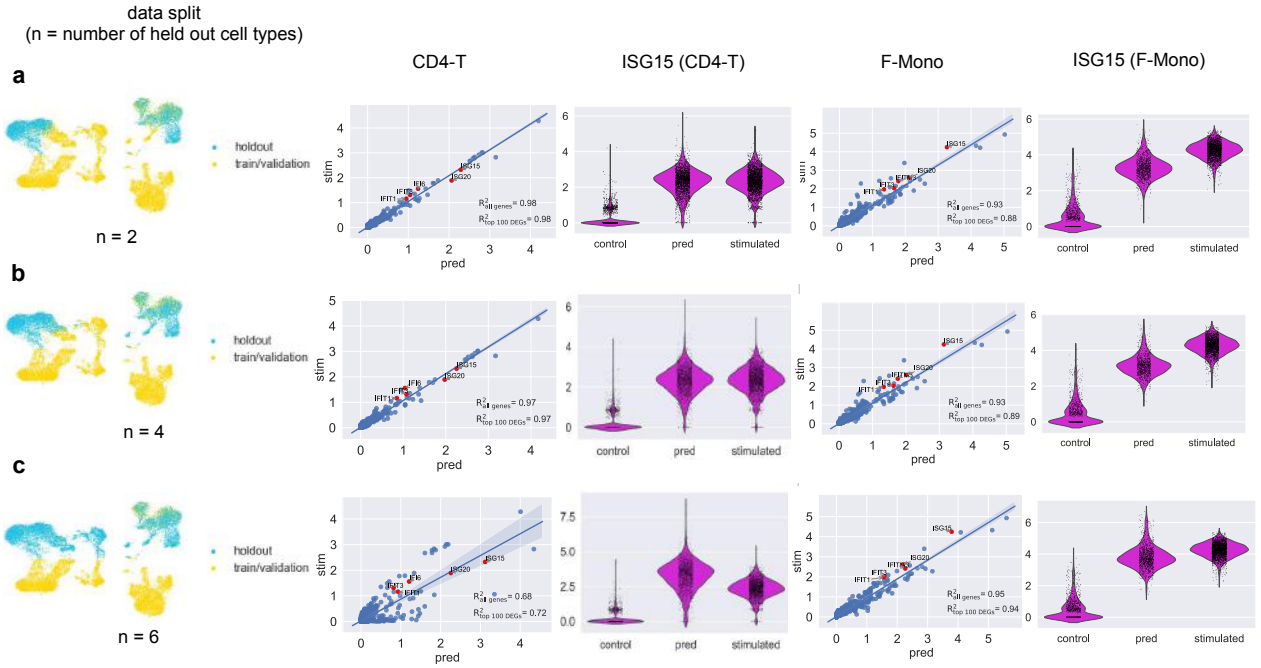
Supplemental Figure 4 | Latent space comparison. (a-c) UMAP visualization of latent space representation for PBMCs from the Kang *et al.* dataset. For scGen (VAE) and CVAE we used the bottleneck layer but for the style transfer GAN we used the discriminator's penultimate output as the input for UMAP algorithm.



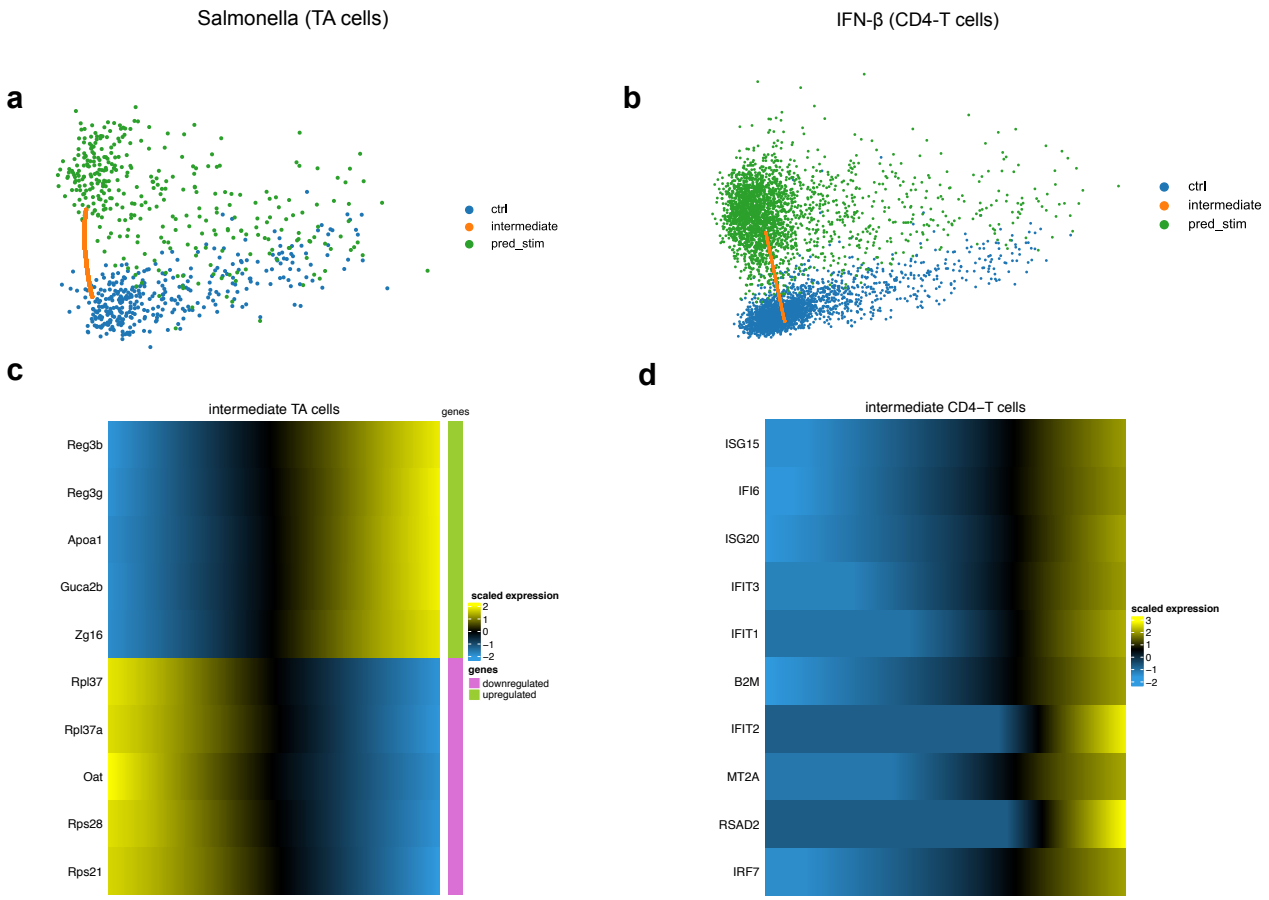
Supplemental Figure 5 | scGen captures cell type specific response patterns. (a) Violin plot for top 10 specific response genes from Kang *et al.* out of 500 DEGs according to the gene specificity score (Supplemental Note 7) across control (c), real stimulated (r.s), and predicted stimulated (p.s) for different cell types. **(b)** Box plots of top 500 DEGs ordered by the gene specificity score. Each bin is composed of 50 genes and each point in the bin shows the error between average expression of that gene within a cell type and average prediction by scGen for that cell type. In total each bin contains 50 (number of genes) × 7 (number of cell types) points and the error is $\frac{|x-x_{\text{pred}}|}{max(x,1)}$. **(c)** Predictions using linear PCA + Vec. Arithm. shows how this linear model fails to capture specific responses. Note how the model increases *APOBEC3A* in all cell types whereas scGen upregulates it only in responsive cell types. Similar phenomenon happens in CD4-T cells, in which scGen only upregulates responsive genes whereas the linear model upregulates all genes. **(d)** Similar box plot as **b** depicts how a linear model yields larger error in bin-0 which includes top 50 genes with cell type specific differential expression patterns.



Supplemental Figure 6 | scGen captures cell type specific responses patterns in two datasets of intestinal epithelial cells. Violin plot for top 10 specific response genes out of 500 (top 250 upregulated and top 250 downregulated) DEGs according to gene specificity score (Supplemental Note 7) for *H.poly* **(a)** and *Salmonella* **(b)** datasets from Haber *et al.* across control (c), real stimulated (r.s), and predicted stimulated (p.s) for different cell types.



Supplemental Figure 7 | scGen performs robustly when holding out more than one cell type. (a-c) Predicting IFN-β stimulated CD4-T and F-Mono cells form the Kang *et al.* dataset in different scenarios with different numbers of held out cell types. First panel shows UMAP visualization for the position of held out cells. Other panels show mean gene expression of all genes and violin plot for ISG15, the top response gene after stimulation with IFN-β for CD4-T and F-Mono cells.

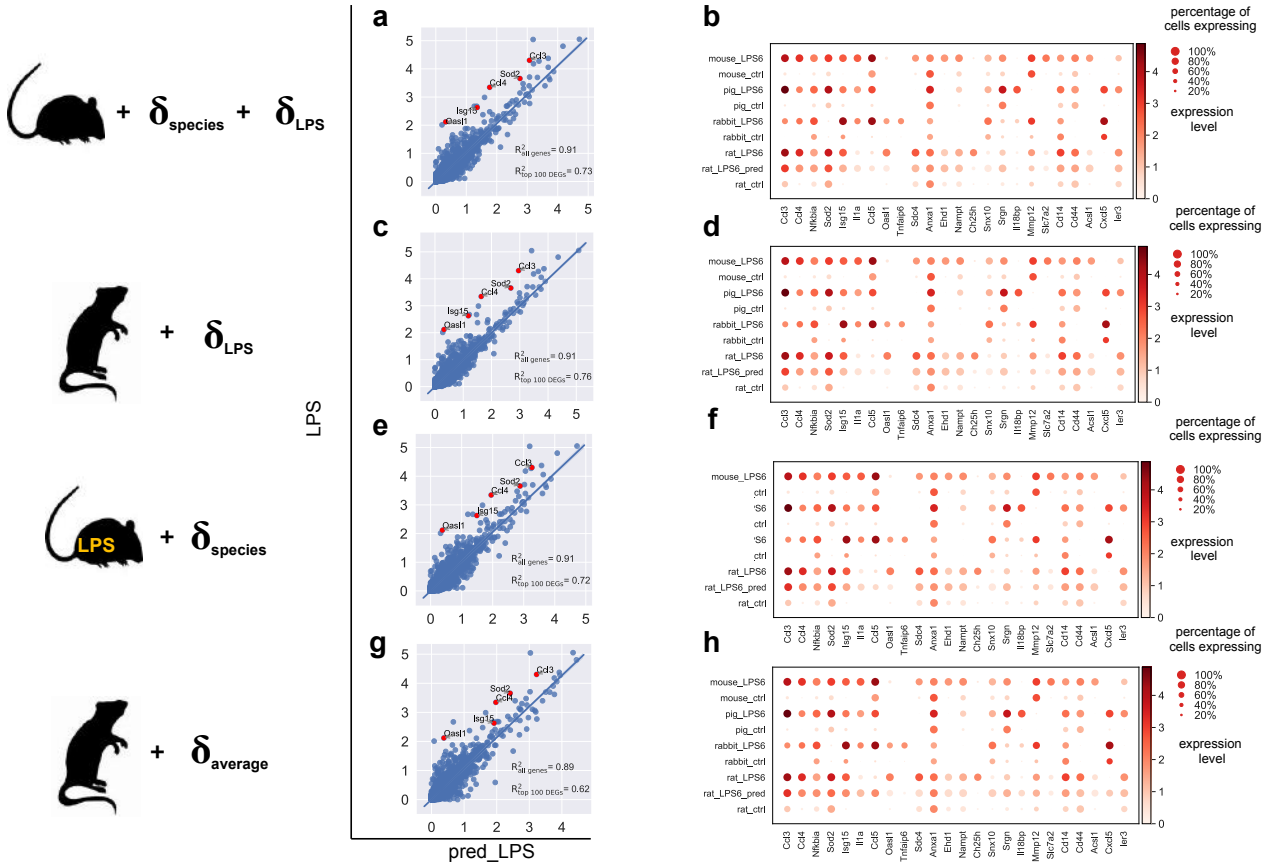


Supplemental Figure 8 | scGen enables the generation of intermediary cells between two conditions. (a-b) PCA visualization of 2,000 generated intermediary TA (Haber *et al.*) and CD4-T (Kang *et al.*) cells between control and predicted cells. **(c)** Change in top five upregulated and downregulated genes as we move from control to *Salmonella* infected cells. **(d)** Similarly, variation of top 10 IFN-β marker genes while transitioning from control to predicted IFN-β stimulated cells.

Name	Operation	NoF	Dropout	BN	Activation	Input
input	-	input_dim	×	×	-	-
FC-1	FC	800	0.2	√	Leaky ReLU	input
FC-2	FC	800	0.2	√	Leaky ReLU	FC-1
mean	FC	100	×	×	Linear	FC-2
var	FC	100	×	×	Linear	FC-2
z-sample	FC	100	×	×	Linear	[mean, var]
FC-3	FC	800	0.2	√	Leaky ReLU	z-sample
FC-4	FC	800	0.2	√	Leaky ReLU	FC-3
output	FC	input_dim	×	×	ReLU	FC-4

Optimizer	Adam
Learning Rate	0.001
Leaky ReLU slope	0.2
Batch Size	32
# of Epochs	300
α	0.00005

Supplemental Table 1 | scGen detailed architecture. We used the same architecture for all the examples in the paper. The input_dim parameter for each dataset is: IFN-β (6,998), H.poly (7,000), salmonella (7,000), cross species (7,000), LPS (6,619), pancreas (2,448), and mouse atlases (2,797).

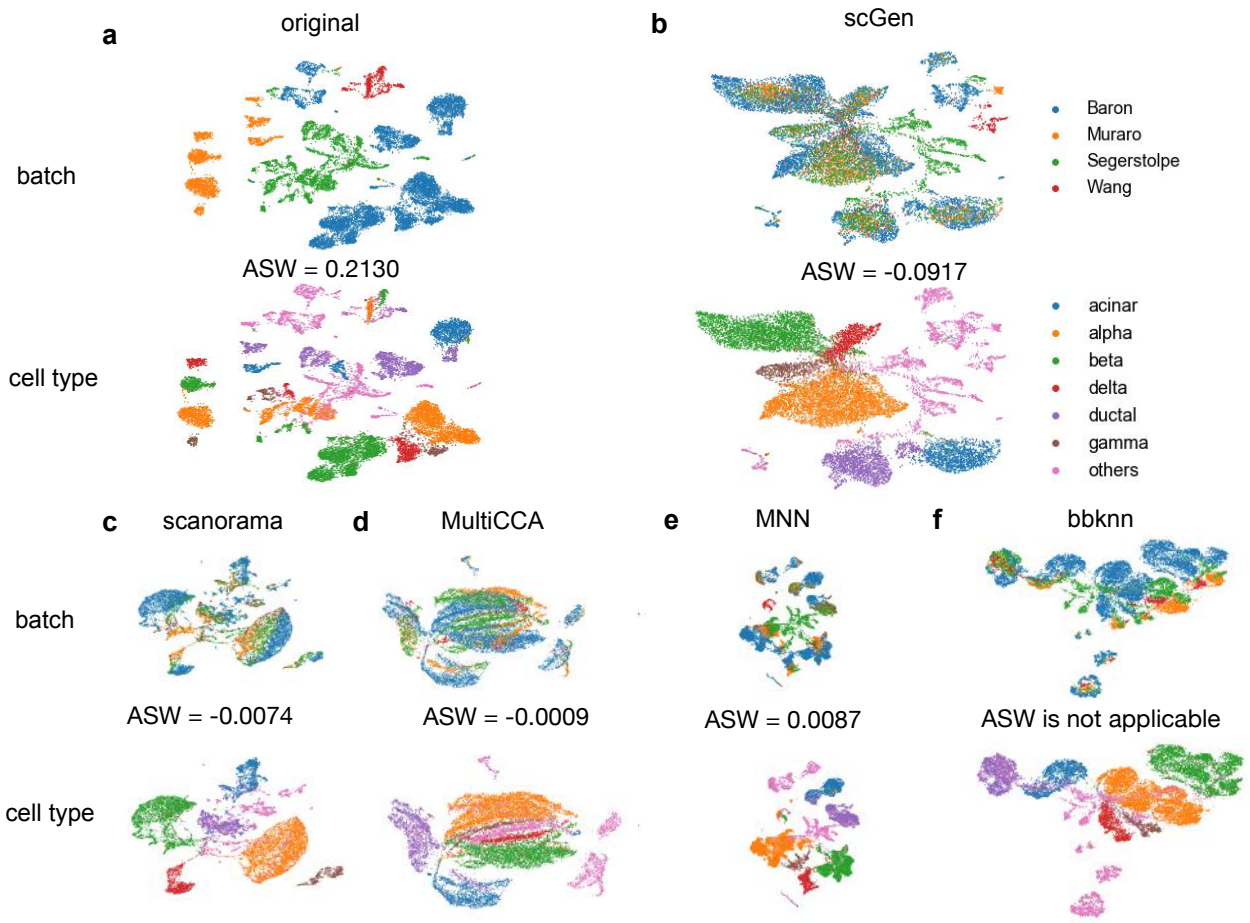


Supplemental Figure 9 | Alternative vector arithmetics for cross–species prediction. **(a-f)** Prediction of rat_{LPS} by adding difference vectors estimated using rat and mouse where $\delta_{\text{LPS}} = \text{mouse}_{\text{LPS}} - \text{mouse}_{\text{control}}$ and $\delta_{\text{species}} = \text{rat}_{\text{control}} - \text{mouse}_{\text{control}}$. **(g-h)** Prediction of rat_{LPS} by adding δ_{average} to $\text{rat}_{\text{control}}$ where $\delta_{\text{average}} = \text{avg}(z_{\text{LPS, all species}}) - \text{avg}(z_{\text{control, all species}})$.

Name	Operation	NoF	Dropout	BN	Activation	Input
input	-	6,998	×	×	-	-
FC-1	FC	700	0.5	√	Leaky ReLU	input
FC-2	FC	100	0.5	√	Leaky ReLU	FC-1
FC-3	FC	50	0.5	√	Leaky ReLU	FC-2
FC-4	FC	100	0.5	√	Leaky ReLU	FC-3
FC-5	FC	700	0.5	√	Leaky ReLU	FC-4
generator_out	FC	6,998	×	√	ReLU	FC-5
FC-6	FC	700	0.5	√	Leaky ReLU	generator_out
FC-7	FC	100	0.5	√	Leaky ReLU	FC-6
discriminator_out	FC	1	×	×	Sigmoid	FC-7

Generator Optimizer	Adam
Discriminator Optimizer	Adam
Learning Rate	0.001
Leaky ReLU slope	0.2
# of Epochs	1000

Supplemental Table 2 | Style transfer GAN detailed architecture.

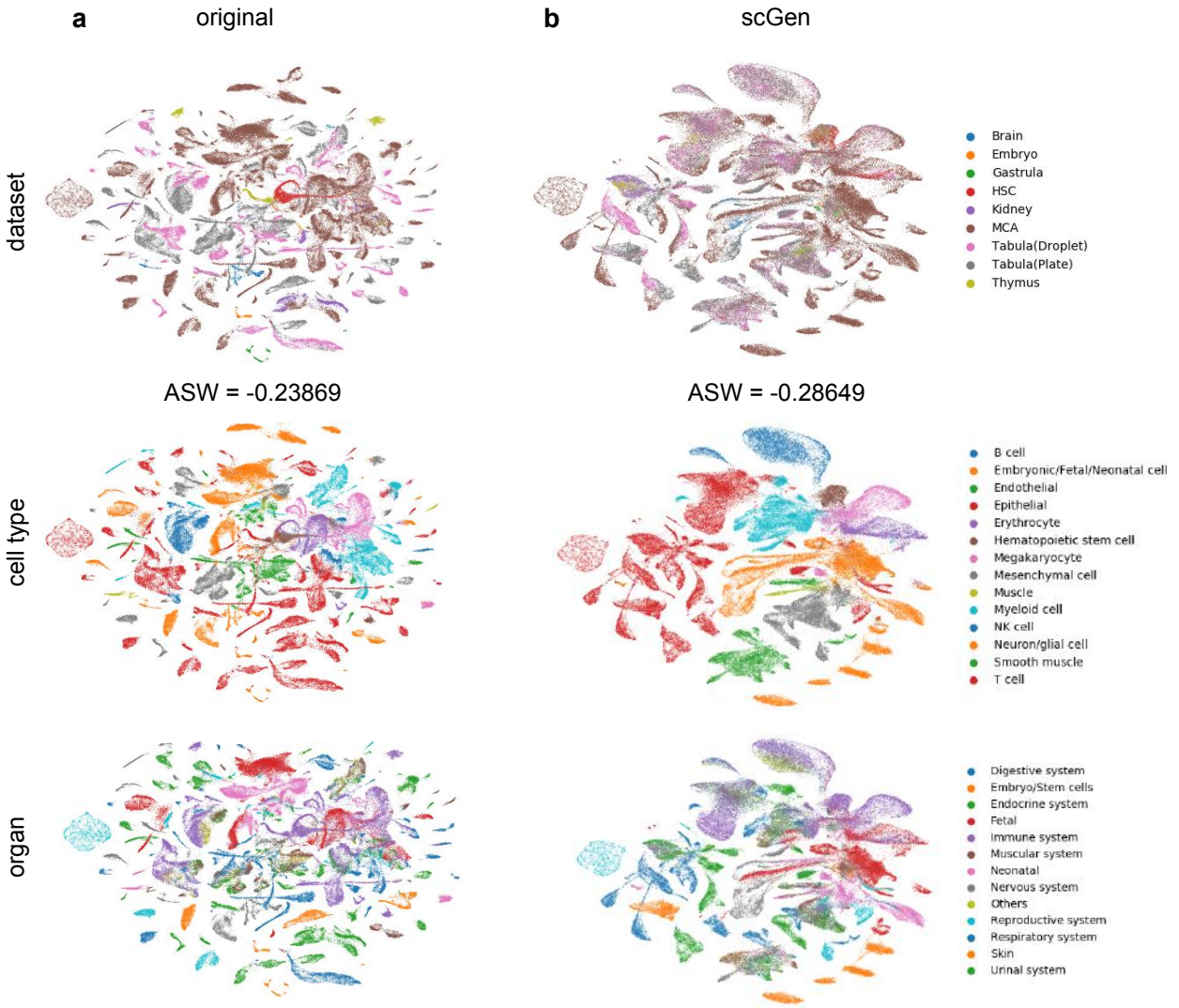


Supplemental Figure 10 | Comparison of existing batch effect removal methods for integrating four different pancreatic datasets. **(a)** Original data. **(b)** scGen. **(c)** Scanorama. **(d)** MultiCCA. **(e)** MNN. **(f)** Bbknn.

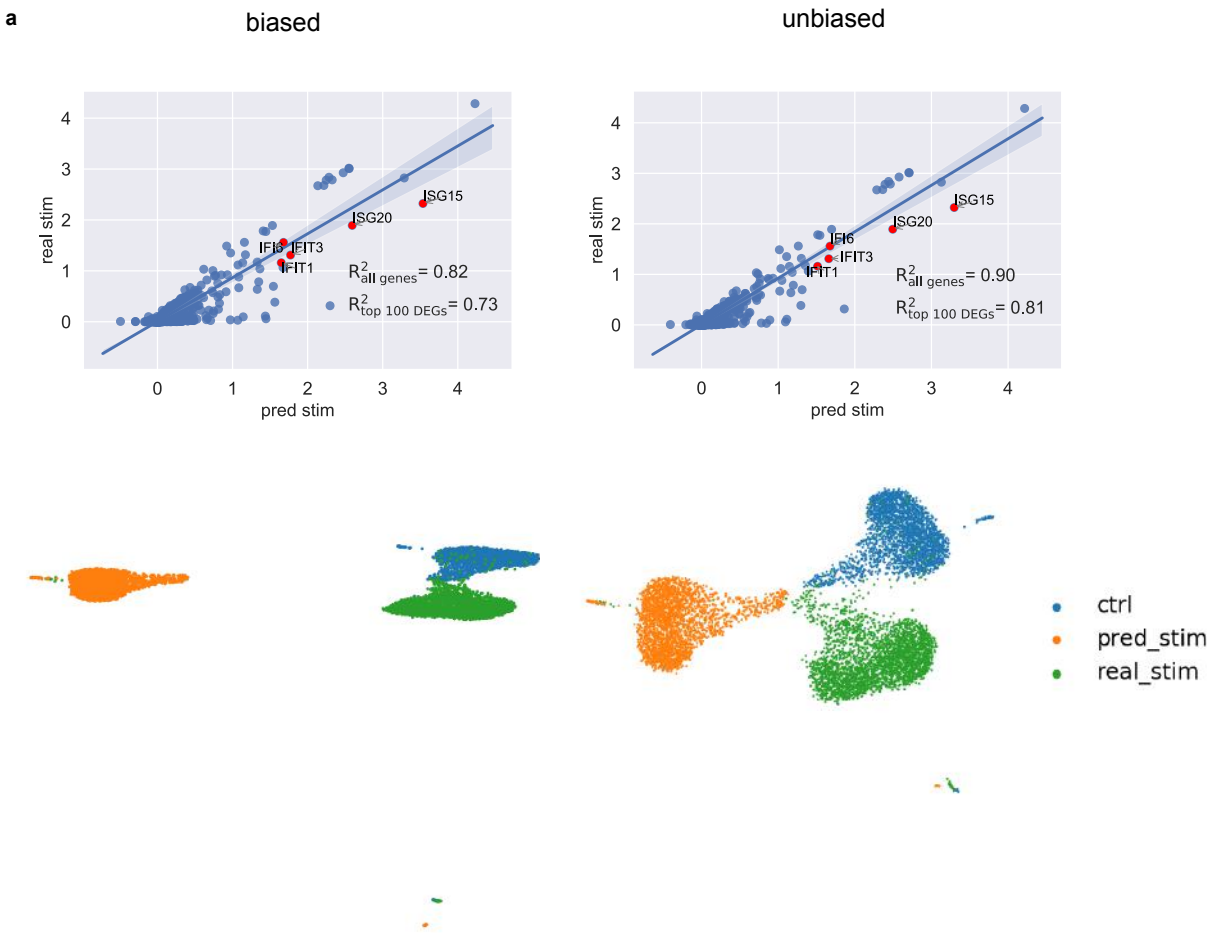
Name	Operation	NoF	Dropout	BN	Activation	Input
input	-	6,998	×	×	-	-
condition	-	1	×	×	-	-
FC-1	FC	700	0.2	√	Leaky ReLU	[input, condition]
FC-2	FC	400	0.2	√	Leaky ReLU	FC-1
mean	FC	20	×	×	Linear	FC-2
var	FC	20	×	×	Linear	FC-2
z-sample	FC	20	×	×	Linear	[mean, var]
FC-3	FC	400	0.2	√	Leaky ReLU	[z-sample, condition]
FC-4	FC	700	0.2	√	Leaky ReLU	FC-3
output	FC	6,998	×	×	ReLU	FC-4

Optimizer	Adam
Learning Rate	0.001
Leaky ReLU slope	0.2
Batch Size	32
# of Epochs	100
α	0.1

Supplemental Table 3 | CVAE detailed architecture



Supplemental Figure 11 | scGen integrates eight mouse single–cell atlasess with 114,600 cells. **(a)** UMAP visualization of eight different datasets with their corresponding study, cell type and organ labels. ASW was calculated based on the 57,300 randomly subsampled cells with their study labels. **(b)** scGen merges the data by connecting the similar cell types according to their cell labels while having lower ASW (-0.28649).



Supplemental Figure 12 | Biased sampling effect. (a) Comparison between biased and unbiased predictions for CD4-T cells using PCA and vector arithmetics.

References

[1] Stubbington, M. J., Rozenblatt-Rosen, O., Regev, A. & Teichmann, S. A. Single-cell transcriptomics to explore the immune system in health and disease. *Science* **358**, 58–63 (2017).

[2] Angerer, P. *et al.* Single cells make big data: new challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology* **4**, 85–91 (2017).

[3] Kang, H. M. *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36**, 89–94 (2017).

[4] Haber, A. L. *et al.* A single-cell survey of the small intestinal epithelium. *Nature* **551**, 333–339 (2017).

[5] Hagai, T. *et al.* Gene expression variability across cells and species shapes innate immunity. *Nature* **563**, 197 (2018).

[6] Dixit, A. *et al.* Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell* **167**, 1853–1866.e17 (2016).

[7] Adamson, B. *et al.* A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell* **167**, 1867–1882 (2016).

[8] Datlinger, P. *et al.* Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods* **14**, 297–301 (2017).

[9] Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature methods* **11**, 740 (2014).

[10] Vallejos, C. A., Marioni, J. C. & Richardson, S. Basics: Bayesian analysis of single-cell sequencing data. *PLoS computational biology* **11**, e1004333 (2015).

[11] Froehlich, F. *et al.* Efficient parameterization of large-scale mechanistic models enables drug response prediction for cancer cell lines. *bioRxiv* 174094 (2017).

[12] Choi, K., Hellerstein, J., Wiley, S. & Sauro, H. M. Inferring reaction networks using perturbation data. *bioRxiv* 351767 (2018).

[13] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature Methods* **15**, 1053–1058 (2018).

[14] Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single cell RNA-seq denoising using a deep count autoencoder. *bioRxiv* 300681 (2018).

[15] Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications* **9**, 2002 (2018).

[16] Ghahramani, A., Watt, F. M. & Luscombe, N. M. Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. *bioRxiv* 262501 (2018).

[17] Marouf, M. *et al.* Realistic in silico generation and augmentation of single cell RNA-seq data using Generative Adversarial Neural Networks. *bioRxiv* 390153 (2018).

[18] Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods: towards more accurate and robust tools. *bioRxiv* 276907 (2018).

[19] Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *The International Conference on Learning Representations (ICLR)* (2014).

[20] Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 3483–3491 (2015).

[21] Abadi, M. *et al.* Tensorflow: a system for large-scale machine learning.

[22] Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: Large-scale single-cell gene expression data analysis. *Genome biology* **19**, 15 (2018).

[23] McInnes, L. & Healy, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv 1802.03426* (2018).

[24] Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36**, 411 (2018).

[25] Bendall, S. C. *et al.* Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. *Cell* **157**, 714–725 (2014).

[26] Wolf, F. A. *et al.* Graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *bioRxiv* 208819 (2017).

[27] Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *The International Conference on Learning Representations (ICLR)* (2016).

[28] Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. *ICLR Workshop* (2013).

[29] Liu, M.-Y. & Tuzel, O. Coupled generative adversarial networks. In *Advances in neural information processing systems*, 469–477 (2016).

[30] Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)* (2017).

[31] Amodio, M. & Krishnaswamy, S. Magan: Aligning biological manifolds. *arXiv 1803.00385* (2018).

[32] Clift, M. J. *et al.* A novel technique to determine the cell type specific response within an in vitro co-culture model via multi-colour flow cytometry. *Scientific reports* **7**, 434 (2017).

[33] Schubert, M. *et al.* Perturbation-response genes reveal signaling footprints in cancer gene expression. *Nature communications* **9**, 20 (2018).

[34] Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8**, 14049 (2017).

[35] Regev, A. *et al.* Science Forum: The Human Cell Atlas. *eLife* **6**, e27041 (2017).

[36] Baron, M. *et al.* A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems* **3**, 346–360 (2016).

[37] Segerstolpe, Å. *et al.* Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell metabolism* **24**, 593–607 (2016).

[38] Wang, Y. J. *et al.* Single-cell transcriptomics of the human endocrine pancreas. *Diabetes* **65**, 3028–3038 (2016).

[39] Muraro, M. J. *et al.* A single-cell transcriptome atlas of the human pancreas. *Cell systems* **3**, 385–394 (2016).

[40] Hie, B. L., Bryson, B. & Berger, B. Panoramic stitching of heterogeneous single-cell transcriptomic data. *bioRxiv* 371179 (2018).

[41] Park, J.-E., Polanski, K., Meyer, K. & Teichmann, S. A. Fast Batch Alignment of Single Cell Transcriptomes Unifies Multiple Mouse Cell Atlases into an Integrated Landscape. *bioRxiv* 397042 (2018).

[42] Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology* **36**, 421 (2018).

[43] Zeisel, A. *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**, 1138–1142 (2015).

[44] Schaum, N. *et al.* Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature* **562**, 367–372 (2018).

[45] Han, X. *et al.* Mapping the mouse cell atlas by microwell-seq. *Cell* **172**, 1091–1107 (2018).

[46] Deng, Q., Ramsköld, D., Reinius, B. & Sandberg, R. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* **343**, 193–196 (2014).

[47] Kernfeld, E. M. *et al.* A Single-Cell Transcriptomic Atlas of Thymus Organogenesis Resolves Cell Types and Developmental Maturation. *Immunity* **48**, 1258–1270.e6 (2018).

[48] Park, J. *et al.* Single-cell transcriptomics of the mouse kidney reveals potential cellular targets of kidney disease. *Science* eaar2131 (2018).

[49] Mohammed, H. *et al.* Single-cell landscape of transcriptional heterogeneity and cell fate decisions during mouse early gastrulation. *Cell reports* **20**, 1215–1228 (2017).

[50] Dahlin, J. S. *et al.* A single-cell hematopoietic landscape resolves 8 lineage trajectories and defects in kit mutant mice. *Blood* **131**, e1–e11 (2018).

[51] Way, G. P. & Greene, C. S. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *bioRxiv* 174474 (2017).

[52] Smillie, C. S. *et al.* Rewiring of the cellular and inter-cellular landscape of the human colon during ulcerative colitis. *bioRxiv* 455451 (2018).

[53] Amodio, M., Montgomery, R., Pappalardo, J., Hafler, D. & Krishnaswamy, S. Neuron interference: Evidence-based batch effect removal. *arXiv 1805.12198* (2018).

[54] Doersch, C. Tutorial on variational autoencoders. *arXiv 1606.05908* (2016).

[55] White, T. Sampling generative networks: Notes on a few effective techniques. *arXiv 1609.04468* (2016).

[56] Durinck, S., Spellman, P. T., Birney, E. & Huber, W. Mapping identifiers for the integration of genomic datasets with the R/bioconductor package biomart. *Nature Protocols* **4**, 1184–1191 (2009).

[57] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958 (2014).

[58] Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 448–456 (2015).

[59] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *The International Conference on Learning Representations (ICLR)* (2015).

[60] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

Appendix C

Conditional out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics (2020).*

This is a pre-copyedited, author-produced PDF of an article accepted for publication Bioinformatics following peer review.

(iv) Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J. Theis, and F. Alexander Wolf. **"Conditional out-of-distribution generation for unpaired data using transfer VAE."** Bioinformatics 36, no. Supplement_2 (2020): i610-i617.
The article is also available online at:
`https://academic.oup.com/bioinformatics/article/36/Supplement_2/i610/6055927?login=true`

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Gene Expression

Conditional out-of-distribution generation for un-paired data using transfer VAE

Mohammad Lotfollahi [1,2], Mohsen Naghipourfar [1,4], Fabian J. Theis [1,2,3†], and F. Alexander Wolf [1‡]

[1]Institute of Computational Biology, Helmholtz Center Munich, Neuherberg, Germany,
[2]School of Life Sciences Weihenstephan, Technical University of Munich, Munich, Germany,
[3]Department of Mathematics, Technische Universität München, Munich, Germany, and
[4]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

[†]fabian.theis@helmholtz-muenchen.de [‡]alex.wolf@helmholtz-muenchen.de

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: While generative models have shown great success in sampling high-dimensional samples conditional on low-dimensional descriptors (stroke thickness in MNIST, hair color in CelebA, speaker identity in WaveNet), their generation OOD poses fundamental problems due to the difficulty of learning compact joint distribution across conditions. The canonical example of the conditional variational autoencoder (CVAE), for instance, does not explicitly relate conditions during training and, hence, has no explicit incentive of learning such a compact representation.
Results: We overcome the limitation of the CVAE by matching distributions across conditions using maximum mean discrepancy (MMD) in the decoder layer that follows the bottleneck. This introduces a strong regularization both for reconstructing samples within the same condition and for transforming samples across conditions, resulting in much improved generalization. As this amounts to solving a style-transfer problem, we refer to the model as *transfer* VAE (trVAE). Benchmarking trVAE on high-dimensional image and single-cell RNA-seq, we demonstrate higher robustness and higher accuracy than existing approaches.We also show qualitatively improved predictions by tackling previously problematic minority classes and multiple conditions in the context of cellular perturbation response to treatment and disease based on high-dimensional single-cell gene expression data. For generic tasks, we improve Pearson correlations of high-dimensional estimated means and variances with their ground truths from 0.89 to 0.97 and 0.75 to 0.87, respectively. We further demonstrate that trVAE learns cell-type-specific responses after perturbation and improves the prediction of most cell-type-specific genes by 65%.
Availability: The trVAE implementation is available via `github.com/theislab/trvae`. The results of this paper can be reproduced via `github.com/theislab/trvae_reproducibility`.

Introduction

The task of generating high-dimensional samples x conditional on a latent random vector z and a categorical variable s has established solutions (Mirza and Osindero, 2014; Ren *et al.*, 2016). The situation becomes more complicated if the support of z is divided into domains d that come with different meanings: say $d \in \{\text{cat}, \text{dog}\}$ and one is interested in out-of-distribution (OOD) generation of samples x in a domain and condition (d, s) that are not part of the training data. Now, predicting how a given brown dog would look like with black fur becomes an OOD problem if the training data does not have observations of black dogs. To still have a chance of solving it, we assume training data with brown dogs, and brown *and* black cats. In an application with higher relevance, there is strong interest in how untreated humans ($s = 0, d = 0$) respond to drug treatment ($s = 1$) based on training data from human in vitro ($d = 1$) and in vivo mouse ($d = 2$) experiments. Hence, the target domain of interest ($d = 0$) does not offer training data for $s = 1$, but only for $s = 0$.

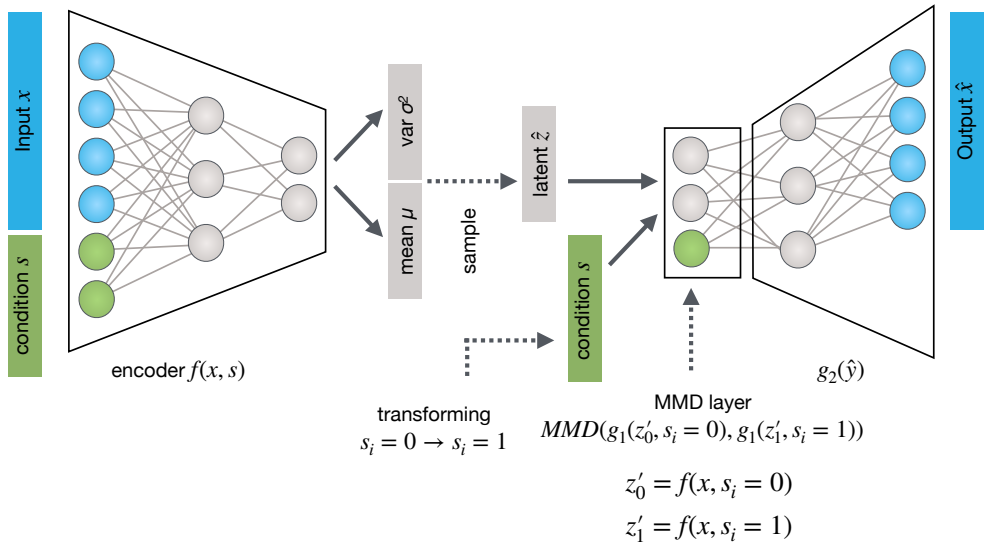


Fig. 1. Transfer VAE (trVAE) is an MMD-regularized conditional VAE. It receives randomized batches of data (x) and condition (s) as input during training, stratified for approximately equal proportions of s. In contrast to a standard CVAE, we regularize the effect of s on the representation obtained after the first-layer $g_1(\hat{z}, s)$ of the decoder g. During prediction time, we transform batches of the source condition $x_{s=0}$ to the target condition $x_{s=1}$ by encoding $\hat{z}_0 = f(x_0, s = 0)$ and decoding $g(\hat{z}_0, s = 1)$.

In the present paper, we suggest to address the challenge of generating samples OOD by regularizing the joint distribution across the categorical variable s using maximum mean discrepancy (MMD) in the framework of a conditional variational autoencoder (CVAE) (Sohn *et al.*, 2015). This produces a more compact representation of a cross-condition distribution that would otherwise display high variance in the standard CVAE. We will show that this leads to more accurate OOD prediction. MMD has proven successful in a variety of tasks. In particular, matching distributions with MMD in variational autoencoders (Kingma and Welling, 2013) has been suggested for unsupervised domain adaptation (Louizos *et al.*, 2015) or for learning statistically independent latent dimensions (Lopez *et al.*, 2018b). In supervised domain adaptation approaches, MMD-based regularization has been shown to be a viable strategy of learning label-predictive features that are stripped off of domain-specific information (Long *et al.*, 2015; Tzeng *et al.*, 2014). In these instances, however, MMD was employed at the bottleneck layer, where it leads to different properties.

Matching distributions across perturbed and control populations has also been studied in the context of causal inference (Johansson *et al.*, 2016), albeit not in the context of generative modeling and OOD generation. (Johansson *et al.*, 2016) showed how to improve counterfactual inference by learning representations that enforce similarity between perturbed and control using a linear discrepancy measure, mentioning MMD as an alternative metric.

In further related work, the OOD generation problem was addressed via hard-coded latent space vector arithmetics (Lotfollahi *et al.*, 2019) and histogram matching (Amodio *et al.*, 2018). The approach of the present paper, however, introduces a data-driven end-to-end approach, which does not involve hard-coded elements and generalizes to more than one condition. We hope the present work further stimulates the recent success of generative models in single-cell biology (Lopez *et al.*, 2018a; Eraslan *et al.*, 2019).

Methods

Variational autoencoder

The motivation of the variational autoencoder (VAE) (Kingma and Welling, 2013) is to provide a neural-network based parametrization for maximizing the likelihood

$$p_\theta(X \mid S) = \int p_\theta(X \mid Z, S) p_\theta(Z \mid S) dZ, \tag{1}$$

where X denotes a high-dimensional random variable, S a random variable representing conditions, θ the model parameters, and $p_\theta(X \mid Z, S)$ the generative distribution that decodes Z into X. Here and in the following we adapt the notation of (Lopez *et al.*, 2018b) while adapting the presentation of (Doersch, 2016).

To assign probability mass to values of Z that are likely to produce actually observed values of X, one introduces an encoding distribution q_ϕ, which can be related to p_θ via

$$\begin{aligned} &\log p_\theta(X \mid S) - (q_\phi(Z|X, S)||p_\theta(Z|X, S)) \\ &= \mathbb{E}_{q_\phi(Z|X,S)}[\log p_\theta(X \mid Z, S)] - (q_\phi(Z|X, S)||p_\theta(Z|S)). \end{aligned}$$

The right hand side of this equation provides the cost function $\mathcal{L}_{\text{VAE}}$ for optimizing neural-network based parametrizations of p_θ and q_ϕ. The left-hand side describes the likelihood subtracted by an error term.

The case in which $S \neq \emptyset$ is referred to as the conditional variational autoencoder (CVAE) (Sohn *et al.*, 2015), and a straight-forward extension of the original framework (Kingma and Welling, 2013), which treated $S \equiv \emptyset$.

Maximum-mean discrepancy

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, $\mathcal{X}$ a separable metric space, $x : \Omega \rightarrow \mathcal{X}$ a random variable and $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ a continuous, bounded, positive semi-definite kernel with a corresponding reproducing kernel Hilbert space

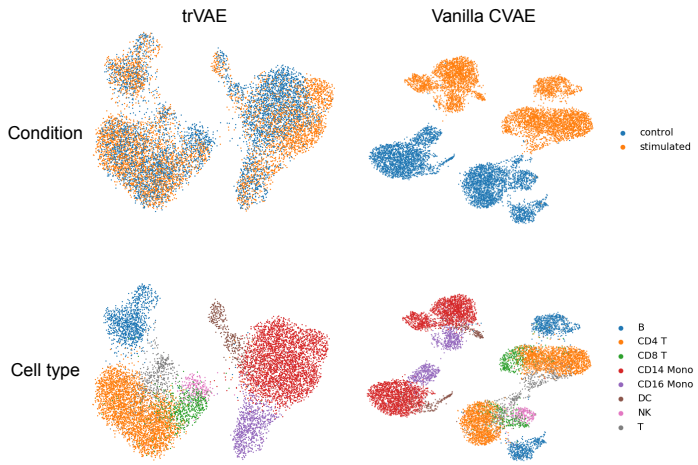


Fig. 2. Comparison of representations for MMD-layer in trVAE and the corresponding layer in the standard CVAE using UMAP (McInnes et al., 2018). The MMD regularization incentivizes the model to learn condition-invariant features resulting in a more compact representation. The figure shows the qualitative effect for the "PBMC data" introduced in experiments section . Both representations show the same number of samples.

(RKHS) $\mathcal{H}$. Consider the kernel-based estimate of a distance between two distributions p and q over the random variables X and X'. Such a distance, defined via the canonical distance between their $\mathcal{H}$-embeddings, is called the maximum mean discrepancy (Gretton *et al.*, 2012) and denoted $l_{\mathrm{MMD}}(p, q)$, with an explicit expression:

$$\ell_{\mathrm{MMD}}(X, X') = \frac{1}{n_0^2}\sum_{n,m} k(x_n, x_m) + \frac{1}{n_1^2}\sum_{n,m} k(x'_n, x'_m) - \frac{2}{n_0 n_1}\sum_{n,m} k(x_n, x'_m), \tag{2}$$

where the sums run over the number of samples n_0 and n_1 for x and x', respectively. Asymptotically, for a universal kernel such as the Gaussian kernel $k(x, x') = e^{-\gamma\|x-x'\|^2}$, $\ell_{\mathrm{MMD}}(X, X')$ is 0 if and only if $p \equiv q$. For the implementation, we use multi-scale RBF kernels defined as:

$$k(x, x') = \sum_{i=1}^{l} k(x, x', \gamma_i) \tag{3}$$

where $k(x, x', \gamma_i) = e^{-\gamma_i\|x-x'\|^2}$ and γ_i is a hyper-parameter.

Addressing the domain adaptation problem, the "Variational Fair Autoencoder" (VFAE) (Louizos *et al.*, 2015) uses MMD to match latent distributions $q_\phi(Z|s=0)$ and $q_\phi(Z|s=1)$ — where s denotes a domain — by adapting the standard VAE cost function $\mathcal{L}_{\mathrm{VAE}}$ according to

$$\mathcal{L}_{\mathrm{VFAE}}(\phi, \theta; X, X', S, S') = \mathcal{L}_{\mathrm{VAE}}(\phi, \theta; X, S) + \mathcal{L}_{\mathrm{VAE}}(\phi, \theta; X', S') - \beta\ell_{\mathrm{MMD}}(Z_{s=0}, Z'_{s'=1}), \tag{4}$$

where X and X' are two high-dimensional observations with their respective conditions S and S'.

In contrast to GANs (Goodfellow *et al.*, 2014) whose training procedure is notoriously hard due to the minmax optimization problem, training models using MMD or Wasserstein distance metrics is comparatively simple (Li *et al.*, 2015; Arjovsky *et al.*, 2017; Dziugaite *et al.*, 2015a) as only a direct minimization of a single loss is involved. It has been shown that MMD-based GANs have some advantages over Wasserstein GANs resulting in a simpler and faster-training algorithm with matching performance (Bińkowski *et al.*, 2018). This motivated us to choose MMD as a metric for implementing distribution matching as a regularization of a CVAE.

Transfer VAE

Let us adapt the following notation for the transformation within a standard CVAE: High-dimensional observations x and a scalar or low-dimensional condition s are transformed using f (encoder, corresponding to distribution q_ϕ) and g (decoder, corresponding to distribution p_θ), which are parametrized by weight-sharing neural networks, and give rise to predictors $\hat{z}$, $\hat{y}$ and $\hat{x}$:

$$\hat{z} = f(x, s) \tag{5a}$$

$$\hat{y} = g_1(\hat{z}, s) \tag{5b}$$

$$\hat{x} = g_2(\hat{y}) \tag{5c}$$

where we distinguished the first (g_1) and the remaining layers (g_2) of the decoder $g = g_2 \circ g_1$ (Fig. 1). While z formally depends on s, it is commonly empirically observed $Z \perp\!\!\!\perp S$, that is, the representation z is disentangled from the condition information s. By contrast, the original representation typically strongly covaries with S: $X \not\perp\!\!\!\perp S$. The observation can be explained by admitting that an efficient z-representation, suitable for minimizing reconstruction and regularization losses, should be as free as possible from information about s. Information about s is directly and explicitly available to the decoder (5b), and hence,

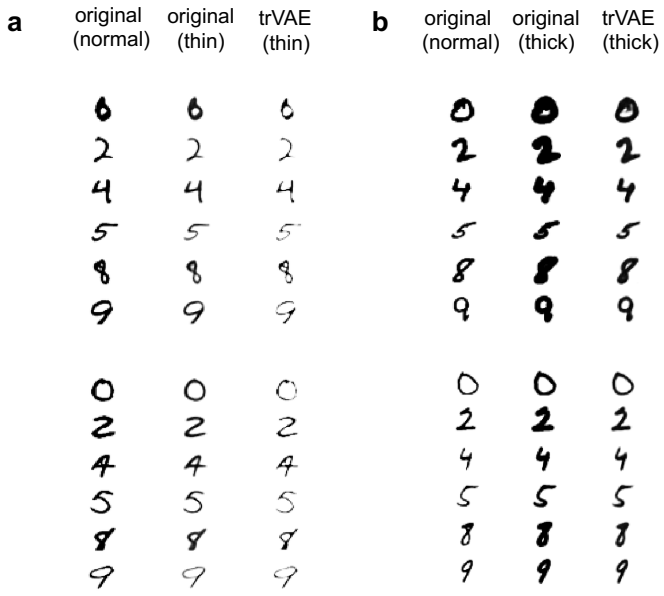


Fig. 3. OOD style transfer for Morpho-MNIST dataset containing normal, thin and thick digits. trVAE successfully transforms normal digits to thin (a) and thick ((b) for digits not seen during training (OOD).

there is an incentive to optimize the parameters of f to *only* explain the variation in x that is *not* explained by s. Experiments below demonstrate that indeed, MMD regularization on the *bottleneck layer* z does not improve performance.

However, even if z is completely free of variation from s, the y representation has a strong s component, $Y \not\perp\!\!\!\perp S$, which leads to a separation of $y_{s=1}$ and $y_{s=0}$ into different regions of their support $\mathcal{Y}$. In the standard CVAE, without any regularization of this y representation, a highly varying, non-compact distribution emerges across different values of s (Fig. 2). To compactify the distribution so that it displays only subtle, controlled differences, we impose MMD (2) in the first layer of the decoder (Fig. 1). We assume that modeling y in the same region of the support of $\mathcal{Y}$ across s forces learning common features across s where possible. The more of these common features are learned, the more accurately the transformation task will performed, and the higher are chances of successful OOD generation. Using one of the benchmark datasets introduced, below, we qualitatively illustrate the effect (Fig. 2).

During training time, all samples are passed to the model with their corresponding condition labels (x_s, s). At prediction time, we pass $(x_{s=0}, s = 0)$ to the encoder f to obtain the latent representation $\hat{z}_{s=0}$. In the decoder g, we pass $(\hat{z}_{s=0}, s = 1)$ and through that, let the model transform data to $\hat{x}_{s=1}$.

The cost function of trVAE derives directly from the standard CVAE cost function, as introduced in the backgrounds section,

$$\mathcal{L}_{\text{CVAE}}(\phi, \theta; X, S, \alpha, \eta) = \eta \mathbb{E}_{q_\theta(Z|X,S)} \log(p_\phi(X|Z, S)) - \alpha(q_\theta(Z|X, S)||p_\phi(Z|X, S)). \quad (6)$$

Consistent with the above, let $\hat{y}_{s=0} = g_1(f(x, s = 0), s = 0)$ and $\hat{y}_{s=1} = g_1(f(x', s = 1), s = 1)$. Through duplicating the cost function for X' and adding an MMD term, the loss of trVAE becomes:

$$\mathcal{L}_{\text{trVAE}}(\phi, \theta; X, X', S, S', \alpha, \eta, \beta) = \mathcal{L}_{\text{CVAE}}(\phi, \theta; X, S, \alpha, \eta) + \mathcal{L}_{\text{CVAE}}(\phi, \theta; X', S', \alpha, \eta) - \beta \ell_{\text{MMD}}(\hat{Y}_{s=0}, \hat{Y}_{s'=1}). \quad (7)$$

Results

We demonstrate the advantages of an MMD-regularized first layer of the decoder by benchmarking versus a variety of existing methods and alternatives:

- Standard CVAE (Sohn *et al.*, 2015)
- CVAE with MMD on bottleneck (MMD-CVAE), similar to VFAE (Louizos *et al.*, 2015)
- MMD-regularized autoencoder (Dziugaite *et al.*, 2015b; Amodio *et al.*, 2019)
- CycleGAN (Zhu *et al.*, 2017)
- scGen, a VAE combined with vector arithmetics (Lotfollahi *et al.*, 2019)
- scVI, a CVAE with a negative binomial output distribution (Lopez *et al.*, 2018a)

First, we demonstrate trVAE's basic OOD style transfer capacity on two established image datasets, on a qualitative level. We then address quantitative comparisons of challenging benchmarks with clear ground truth, predicting the effects of biological perturbation based on high-dimensional structured data. We used convolutional layers for imaging examples in section and fully connected layers for single-cell gene expression datasets in sections and . The optimal hyper-parameters for each application were chosen by using a parameter gird-search for each model.

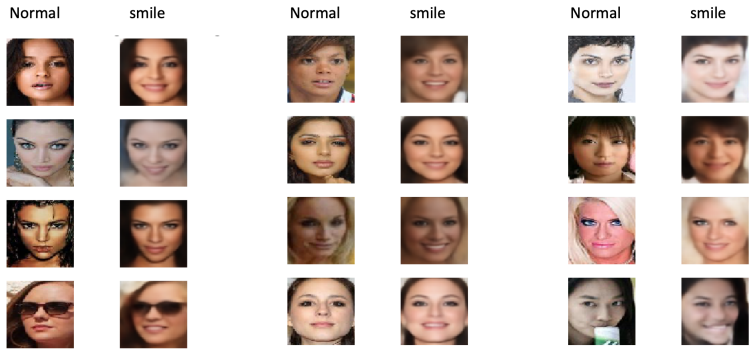


Fig. 4. CelebA dataset with images in two conditions: celebrities without a smile and with a smile on their face. trVAE successfully adds a smile on faces of women without a smile despite these samples completely lacking from the training data (OOD). The training data only comprises non-smiling women and smiling and non-smiling men.

MNIST and CelebA style transformation

Here, we use Morpho-MNIST (Castro *et al.*, 2018), which contains 60,000 images each of "normal" and "transformed" digits, which are drawn with a thinner and thicker stroke. For training, we used all normal-stroke data. Hence, the training data covers all domains ($d \in \{0, 1, 2, \ldots, 9\}$) in the normal stroke condition ($s = 0$). In the transformed conditions (thin and thick strokes, $s \in \{1, 2\}$), we only kept domains $d \in \{1, 3, 6, 7\}$.

We train a convolutional trVAE in which we first encode the stroke width via two fully-connected layers with 128 and 784 features, respectively. Next, we reshape the 784-dimensional into 28*28*1 images and add them as another channel in the image. Such trained trVAE faithfully transforms digits of normal stroke to digits of thin and thicker stroke to the OOD domains (Fig. 3)

Next, we apply trVAE to CelebA (Liu *et al.*, 2015), which contains 202,599 images of celebrity faces with 40 binary attributes for each image. We focus on the task of learning a transformation that turns a non-smiling face into a smiling face. We kept the smiling (s) and gender (d) attributes and trained the model with images from both smiling and non-smiling men but only with non-smiling women.

In this case, we trained a deep convolutional trVAE with a U-Net-like architecture (Ronneberger *et al.*, 2015). We encoded the binary condition labels as in the Morpho-MNIST example and fed them as an additional channel in the input.

Predicting OOD, trVAE successfully transforms non-smiling faces of women to smiling faces while preserving most aspects of the original image (Fig. 4). In addition to showing the model's capacity to handle more complex data, this example demonstrates the flexibility of the the model adapting to well-known architectures like U-Net in the field.

Infection response

Accurately modeling cell response to perturbations is a key question in computational biology. Recently, neural network models have been proposed for OOD predictions of high-dimensional tabular data that quantifies gene expression of single-cells (Lotfollahi *et al.*, 2019; Amodio *et al.*, 2018). However, these models are not trained on the task relying instead on hard-coded transformations and cannot handle more than two conditions.

We evaluate trVAE on a single-cell gene expression dataset that characterizes the gut (Haber *et al.*, 2017) after Salmonella or Heligmosomoides polygyrus (H. poly) infections, respectively. For this, we closely follow the benchmark as introduced in (Lotfollahi *et al.*, 2019). The dataset contains eight different cell types in four conditions: control or healthy cells (n=3,240), H.Poly infection a after three days (H.Poly.Day3, n=2,121), H.poly infection after 10 days (H.Poly.Day10, n=2,711) and salmonella infection (n=1,770) (Fig. 5a). The normalized gene expression data has 1,000 dimensions corresponding to 1,000 genes. Since three of the benchmark models are only able to handle two conditions, we only included the control and H.Poly.Day10 conditions for model comparisons. In this setting, we hold out Tuft infected cells for training and validation, as these consitute the hardest case for OOD generalization (least shared features, few training data).

Figure 5b-c shows trVAE accurately predicts the mean and variance for high-dimensional gene expression in Tuft cells. We compared the distribution of *Defa24*, the gene with the highest change after H.poly infection in Tuft cells, which shows trVAE provides better estimates for mean and variance compared to other models. Moreover, trVAE outperforms other models also when quantifying the correlation of the predicted 1,000 dimensional x with its ground truth (Fig. 5e). In particular, we note that the MMD regularization on the *bottleneck layer* of the CVAE does not improve performance, as argued above.

In contrast to existing approaches, trVAE can handle multiple perturbations at the same time. To illustrate this, we performed another experiment by training eight different models holding out each of the eight cell types form all three conditions. trVAE accurately predicts all cell types across different perturbations (Fig. 5f). The ability to handle multiple perturbations enables analysis and prediction for large drug screening studies.

Stimulation response

Similar to modeling infection response as above, we benchmark on another single-cell gene expression dataset consisting of 7,217 IFN-β stimulated and 6,359 control peripheral blood mononuclear cells (PBMCs) from eight different human Lupus patients (Kang *et al.*, 2018). The stimulation with IFN-β induces dramatic changes in the transcriptional profiles of immune cells, which causes big shifts between control and stimulated cells (Fig. 6a). We studied the OOD prediction of natural killer (NK) cells held out during the training of the model.

trVAE accurately predicts mean (Fig. 6b) and variance (Fig. 6c) for all genes in the held out NK cells. In particular, genes strongly responding to IFN-β (highlighted in red in Fig. 6b-c) are well captured. An effect of applying IFN-β is an increase in ISG15 for NK cells, which the model never sees during training. trVAE predicts this change by increasing the expression of ISG15 as observed in real NK cells (Fig. 6d). A cycle

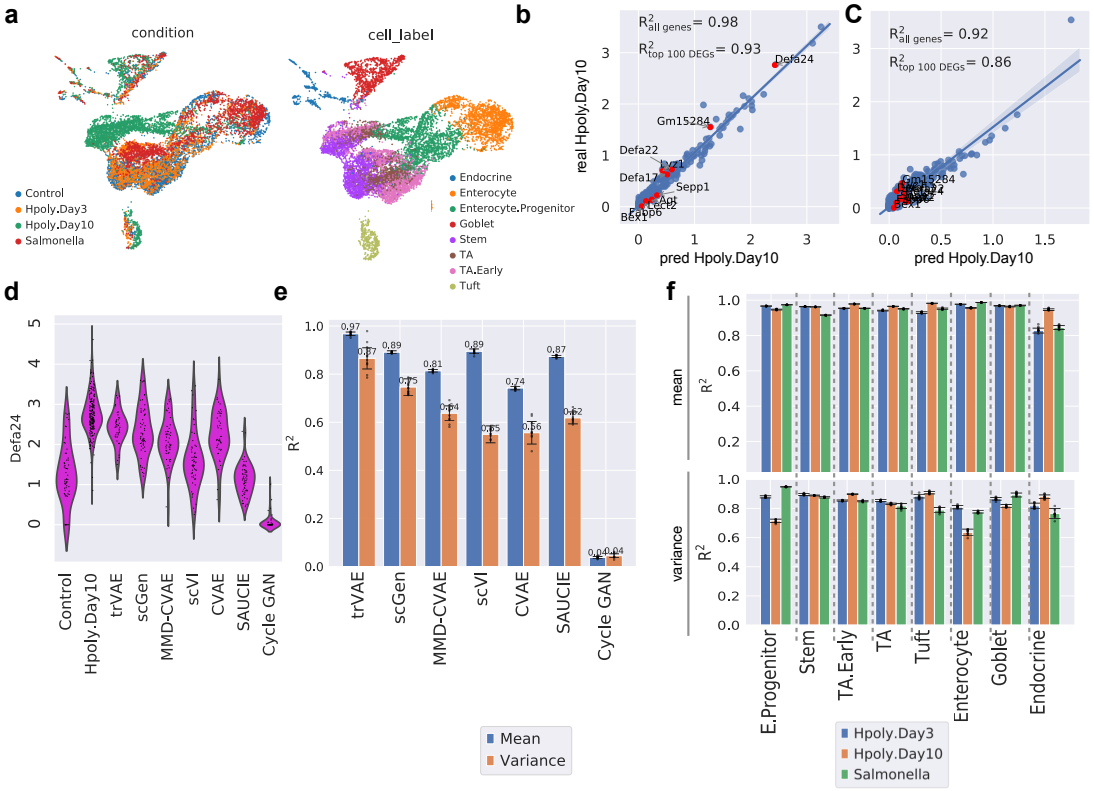


Fig. 5. (a) UMAP visualization of conditions and cell type for gut cells. (b-c) Mean and variance expression of 1,000 genes comparing trVAE-predicted and real infected Tuft cells together with the top 10 differentiall-expressed genes (Methods **??**) highlighted in red (R^2 denotes Pearson correlation between ground truth and predicted values). (d) Distribution of Defa24: the top response gene to H.poly.Day10 infection between control, predicted and real stimulated cells for different models. Vertical axis: expression distribution for Defa24. Horizontal axis: control, real and predicted distribution by different models. (e) Comparison of Pearson's R^2 values for mean and variance gene expression between real and predicted cells for different models. Center values show the mean of R^2 values estimated using $n = 100$ random subsamples for the prediction of each model and error bars depict standard deviation. (f) Comparison of R^2 values for mean and variance gene expression between real and predicted cells by trVAE for the eight different cell types and three conditions. Center values show the mean of R^2 values estimated using $n = 100$ random subsamples for each cell type and error bars depict standard deviation.

GAN and an MMD-regularized auto-encoder (SAUCIE) and other models yield less accurate results than our model. Comparing the correlation of predicted mean and variance of gene expression for all dimensions of the data, we find trVAE performs best (Fig. 6e). To demonstrate the generality of our method we trained seven other models, removing stimulated cells for each of seven different cell types in the study. Our model robustly predicted all other seven cell types (Fig. 6f). .

The specificity of perturbation responses of cells depends on many factors leading to changes in gene expression levels that are either shared across all types or specific to some. Predicting both groups of responses is necessary to address questions such as which cell types are most responsive to a perturbation, and successful drug dose prediction (Hu *et al.*, 2020; Srivatsan *et al.*, 2020).

trVAE can capture specific responses after IFN-β when any of the cell types is absent from training and afterward predicted. To demonstrate this, we scored the specificity of differently expressed genes (DEGs) after IFN-β stimulation using a median-based score (see supplementary methods). trVAE successfully predicts top 10 most cell-type-specific responding genes (Fig. 7a). Specifically, our model predicted the up-regulation of *CCL8*, a *CD14-Mono* specific response gene after IFN-β. As another example, trVAE not only predicted the up-regulation of *ISG15* as a shared response gene but also captured the specific expression pattern of this gene across different cell types. Next, we compared our approach with the state-of-the-art model (scGen) for this task using the top 250 most cell-type-specific DEGs. Our model improves the mean error on the first and the second top 50 specific DEGs by 65% and 44%, respectively (Fig. 7b-c). Further comparison demonstrated that trVAE not only outperforms scGen but also all other benchmarked methods (Supplementary Figs. 1-2).

Discussion

By arguing that the standard CVAE yields representations in the first layer following the bottleneck that vary strongly across categorical conditions, we introduced an MMD regularization that forces these representations to be similar across conditions. The resulting model (trVAE) outperforms existing modeling approaches on benchmark and real-world data sets.

Within the bottleneck layer, CVAEs already display a well-controlled behavior, and regularization does not improve performance. Further regularization at later layers might be beneficial but is numerically costly

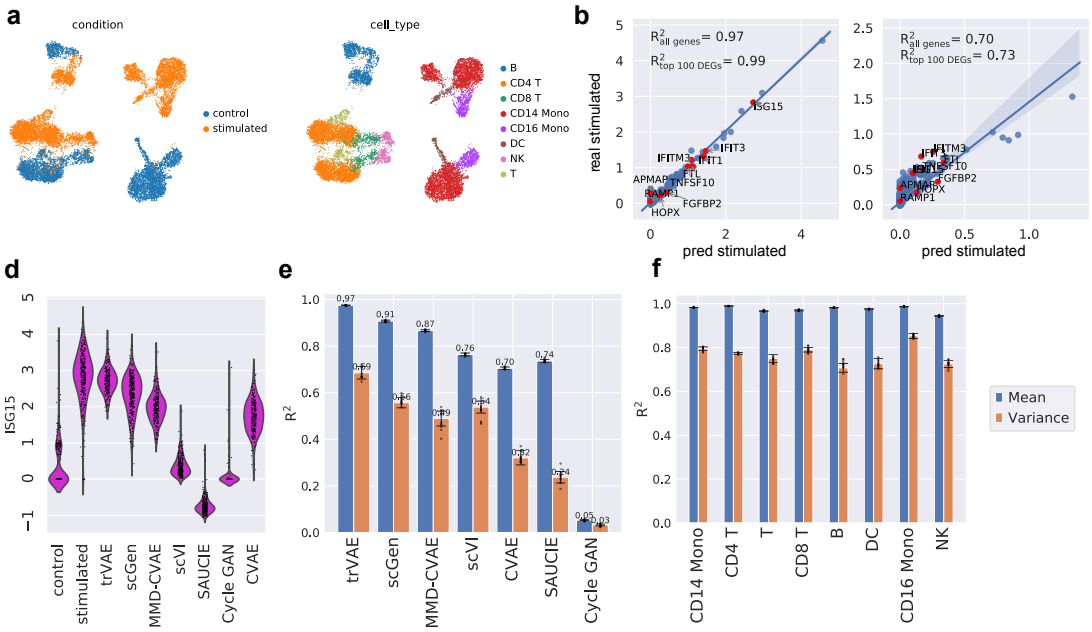


Fig. 6. (a) UMAP visualization of peripheral blood mononuclear cells (PBMCs). (b-c) Mean and variance per 2,000 dimensions between trVAE-predicted and real natural killer cells (NK) together with the top 10 differentially-expressed genes highlighted in red. (d) Distribution of ISG15: the most strongly changing gene after IFN-β perturbation between control, real and predicted stimulated cells for different models. Vertical axis: expression distribution for ISG15. Horizontal axis: control, real and predicted distribution by different models. (e) Comparison of R^2 values for mean and variance gene expression between real and predicted cells for different models. Center values show the mean of R^2 values estimated using $n = 100$ random subsamples for the prediction of each model and error bars depict standard deviation. (f) Comparison of R^2 values for mean and variance gene expression between real and predicted cells by trVAE for eight different cell in the study. Center values show the mean of R^2 values estimated using n = 100 random subsamples for each cell type and error bars depict standard deviation

and unstable as representations become high-dimensional. However, we have not yet systematically investigated this and leave it for future studies.

We have evaluated the predictive power of trVAE by leaving out one cell type and trying to predict it in cases in which the training data contains cell types that are rather similar to the targeted OOD cells Lotfollahi *et al.* (2019). Further evaluation is needed when OOD samples are very different from the training data. Also, further studies are required to understand the uncertainty quantification inherent to the probabilistic nature of the model. Finally, we note that architectures related to Gaussian mixture VAEs or GANs maybe considered as alternatives to the MMD regularisation.

The ability to analyze and predict multiple perturbations allow trVAE to be applied to experiments with many biological conditions. Specifically, recent advances in massive single-cell compounds screening (Srivatsan *et al.*, 2020) provide great potential to exploit our model for further experimental design and the study of interaction effects among different drugs. Future conceptual investigations concern establishing connections to causal-inference-inspired models beyond (Johansson *et al.*, 2016) such as CEVAE (Louizos *et al.*, 2017), establishing further that faithful modeling of an interventional distribution can be re-framed as successful perturbation effect prediction across domains.

Acknowledgements

We are grateful to Anna Klimovskaia for pointing us to the reference of (Johansson *et al.*, 2016) and to Romain Lopez for pointing us to problems with the background section on the variational autoencoder in the arXiv preprint version of this manuscript.

References

Amodio, M., van Dijk, D., Montgomery, R., Wolf, G., and Krishnaswamy, S. (2018). Out-of-sample extrapolation with neuron editing. *arXiv:1805.12198*.

Amodio, M., Van Dijk, D., Srinivasan, K., Chen, W. S., Mohsen, H., Moon, K. R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., *et al.* (2019). Exploring single-cell data with deep multitasking neural networks. *Nature Methods*, **16**, 1139–1145.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia. PMLR.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying mmd gans. *arXiv:1801.01401*.

Castro, D. C., Tan, J., Kainz, B., Konukoglu, E., and Glocker, B. (2018). Morpho-MNIST: Quantitative assessment and diagnostics for representation learning.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv:1606.05908*.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015a). Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 258–267, Arlington, Virginia, United States. AUAI Press.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015b). Training generative neural networks via maximum mean discrepancy optimization. *arXiv:1505.03906*.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. (2019). Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, **10**(1), 390.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, **13**, 723–773.

Haber, A. L., Biton, M., Rogel, N., Herbst, R. H., Shekhar, K., Smillie, C., Burgin, G., Delorey, T. M., Howitt, M. R., Katz, Y., Tirosh, I., Beyaz, S., Dionne, D.,

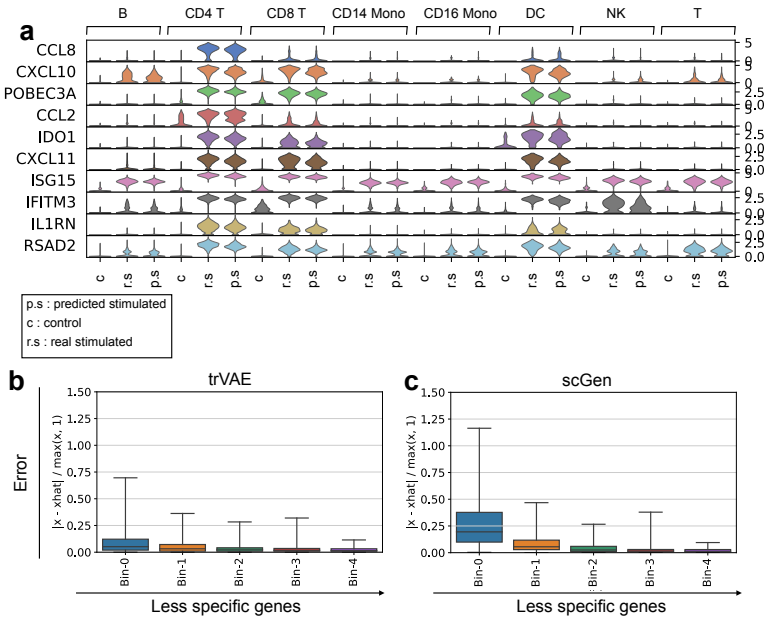


Fig. 7. (a) Violin plot for top 10 specific response genes after out of 250 DEGs according to the gene specificity score across control (c), real stimulated (r.s), and predicted stimulated (p.s) for different cell types. Vertical axis: expression distribution for top specific genes. Horizontal axis: control, real and predicted distribution by trVAE for different cell types. (b-c) Box plots of top 500 DEGs ordered by the gene specificity score. Each bin is composed of 50 genes and each point in the bin shows the error between average expression of that gene within a cell type and average prediction by trVAE and scGen for that cell type. In total, each boxplot has been derived from 50 (number of genes) x 8 (number of cell types) points (n=400). Box plots indicate the median (center lines), interquantile range (hinges), and whiskers represents min and max values.

Zhang, M., Raychowdhury, R., Garrett, W. S., Rozenblatt-Rosen, O., Shi, H. N., Yilmaz, O., Xavier, R. J., and Regev, A. (2017). A single-cell survey of the small intestinal epithelium. *Nature*, **551**, 333.

Hu, Y., Ranganathan, M., Shu, C., Liang, X., Ganesh, S., Osafo-Addo, A., Yan, C., Zhang, X., Aouizerat, B. E., Krystal, J. H., D'Souza, D. C., and Xu, K. (2020). Single-cell transcriptome mapping identifies common and cell-type specific genes affected by acute delta9-tetrahydrocannabinol in humans. *Scientific Reports*, **10**(1), 3450.

Johansson, F., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029.

Kang, H. M., Subramaniam, M., Targ, S., Nguyen, M., Maliskova, L., McCarthy, E., Wan, E., Wong, S., Byrnes, L., Lanata, C. M., *et al.* (2018). Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, **36**(1), 89.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv:1312.6114*.

Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. *arXiv:1502.02791*.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018a). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, **15**(12), 1053–1058.

Lopez, R., Regier, J., Jordan, M. I., and Yosef, N. (2018b). Information constraints on auto-encoding variational bayes. In *Advances in Neural Information Processing Systems*, pages 6114–6125.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. (2019). scGen predicts single-cell perturbation responses. *Nature methods*, **16**(8), 715.

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. (2015). The variational fair autoencoder. *arXiv:1511.00830*.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456.

McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv:1411.1784*.

Ren, Y., Zhu, J., Li, J., and Luo, Y. (2016). Conditional generative moment-matching networks. In *Advances in Neural Information Processing Systems*, pages 2928–2936.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention âĹ“ MICCAI 2015*, page 234âĹ“241.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28*, pages 3483–3491.

Srivatsan, S. R., McFaline-Figueroa, J. L., Ramani, V., Saunders, L., Cao, J., Packer, J., Pliner, H. A., Jackson, D. L., Daza, R. M., Christiansen, L., *et al.* (2020). Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, **367**(6473), 45–51.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv:1412.3474*.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*.

Appendix D

Learning interpretable cellular responses to complex perturbations in high-throughput screens.

BioRxiv (2021).

This is a preprint, CPA paper. The paper is not peer-reviewed, however it is inserted here for the convenience of the reader to understand the the thesis.

(v) Mohammad Lotfollahi*, Anna Klimovskaia*, Carlo De Donno, Yuge Ji, Ignacio L. Ibarra, F. Alexander Wolf, Nafissa Yakubova, Fabian J. Theis, and David Lopez-Paz. **"Learning interpretable cellular responses to complex perturbations in high-throughput screens."** bioRxiv (2021).
The article is also available online at:
`https://www.biorxiv.org/content/10.1101/2021.04.14.439903v2`

Learning interpretable cellular responses to complex perturbations in high-throughput screens

Mohammad Lotfollahi[1,3,⋆], Anna Klimovskaia Susmelj[2,5,⋆], Carlo De Donno [1,7,⋆⋆], Yuge Ji [1,⋆⋆], Ignacio L. Ibarra[1], F. Alexander Wolf [1,○], Nafissa Yakubova [2], Fabian J. Theis[1,3,4,6‡], David Lopez-Paz[2]

1 Helmholtz Center Munich – German Research Center for Environmental Health, Institute of Computational Biology, Neuherberg, Munich, Germany.

2 Facebook AI, 6 Rue Ménars, Paris, 75002, France

3 School of Life Sciences Weihenstephan, Technical University of Munich, Munich, Germany.

4 Department Mathematics, Technical University of Munich, Munich, Munich, Germany.

5 Swiss Data Science Center, Zurich, Switzerland.

6 Wellcome Trust Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridgeshire, UK.

7 Department of Stress Neurobiology and Neurogenetics, Max Planck Institute of Psychiatry, Munich, Bavaria, Germany.

⋆ These authors contributed equally to the work.

⋆⋆ These authors contributed equally to the work.

○ Present address: Cellarity, Inc., Cambridge, MA.

‡ Correspondence to fabian.theis@helmholtz-muenchen.de

Abstract

Recent advances in multiplexed single-cell transcriptomics experiments are facilitating the high-throughput study of drug and genetic perturbations. However, an exhaustive exploration of the combinatorial perturbation space is experimentally unfeasible, so computational methods are needed to predict, interpret, and prioritize perturbations. Here, we present the compositional perturbation autoencoder (CPA), which combines the interpretability of linear models with the flexibility of deep-learning approaches for single-cell response modeling. CPA encodes and learns transcriptional drug responses across different cell type, dose, and drug combinations. The model produces easy-to-interpret embeddings for drugs and cell types, which enables drug similarity analysis and predictions for unseen dosage and drug combinations. We show that CPA accurately models single-cell perturbations across compounds, doses, species, and time. We further demonstrate that CPA predicts combinatorial genetic interactions of several types, implying that it captures features that distinguish different interaction programs. Finally, we demonstrate that CPA can generate *in-silico* 5,329 missing genetic combination perturbations (97.6% of all possibilities) with diverse genetic interactions. We envision our model will facilitate efficient experimental design and hypothesis generation by enabling *in-silico* response prediction at the single-cell level, and thus accelerate therapeutic applications using single-cell technologies.

Introduction

Single-cell RNA-sequencing (scRNA-seq) profiles gene expression in millions of cells across tissues[1, 2] and species[3]. Recently, novel technologies have been developed that extend these measurements to high-throughput screens (HTSs), which measure response to thousands of independent perturbations[4, 5]. These advances show promise for facilitating and thus accelerating drug development[6]. HTSs applied at the single-cell level provide both comprehensive molecular phenotyping and capture heterogeneous responses, which otherwise could not be identified using traditional HTSs[4].

While the development of high-throughput approaches such as "cellular hashing" [4, 7, 8] facilitates scRNA-seq in multi-sample experiments at low cost, these strategies require expensive library preparation[4], and do not easily scale to large numbers of perturbations. These shortcom-

27 ings become more apparent when exploring the effects of combination therapies[9–11] or genetic
28 perturbations[5, 12, 13], where experimental screening of all possible combinations becomes infeasi-
29 ble. While projects such as the Human Cell Atlas[14] aim to comprehensively map cellular states
30 across tissues in a reproducible fashion, the construction of a similar atlas for the effects of pertur-
31 bations on gene expression is impossible, due to the vast number of possibilities. Since brute-force
32 exploration of the combinatorial search space is infeasible, it is necessary to develop computational
33 tools to guide the exploration of the combinatorial perturbation space to nominate promising candi-
34 date combination therapies in HTSs. A successful computational method for the navigation of the
35 combinatorial space must be able to predict the behaviour of cells when subject to novel combinations
36 of perturbations only measured separately in the original experiment. These data are referred to as
37 Out-Of-Distribution (OOD) data. OOD prediction would enable the study of perturbations in the
38 presence of different treatment doses [4, 15], combination therapies[8], multiple genetic knockouts[5],
39 and changes across time[15].

40 Recently, several computational approaches have been developed for predicting cellular responses
41 to perturbations[16–20]. The first approach leverages mechanistic modeling [18, 19] to predict cell
42 viability[19] or the abundance of a few selected proteins[18]. Although they are powerful at interpret-
43 ing interactions, mechanistic models usually require longitudinal data (which is often unavailable in
44 practice) and most do not scale to genome wide measurements to predict high-dimensional scRNA-
45 seq data. Linear models[12, 21] do not suffer from these scalability issues, but have limited predictive
46 power and are unable to capture nonlinear cell-type specific responses. In contrast, deep learning
47 (DL) models do not face these limitations. Recently, DL methods have been used to model gene
48 expression latent spaces from scRNA-seq data [22–25], and describe and predict single-cell responses
49 [16, 17, 20, 26]. However, current DL-based approaches also have limitations: they model only a
50 handful of perturbations; can be difficult to interpret; cannot handle combinatorial treatments; and
51 cannot incorporate continuous covariates such as dose and time, or discrete covariates such as cell
52 types, species, and patients while preserving interpretability. Therefore, while current DL methods
53 have modeled individual perturbations, none have been proposed for HTS.

54 Here, we propose the *compositional perturbation autoencoder (CPA)*, a novel, interpretable method
55 to analyze and predict scRNA-seq perturbation responses across combinations of conditions such
56 as dosage, time, drug, and genetic knock-out. The CPA borrows ideas from interpretable linear
57 models, and applies them in a flexible DL model to learn factorized latent representations of both
58 perturbations and covariates. Given a scRNA-seq dataset, the perturbations applied, and covariates
59 describing the experimental setting, CPA decomposes the data into a collection of embeddings
60 (representations) associated with the cell type, perturbation, and other external covariates. Since
61 these embeddings encode the transcriptomic effect of a drug or genetic perturbation, they can be used
62 by CPA users to study drug effects and similarities useful for drug repurposing applications. By virtue
63 of an adversarial loss, these embeddings are independent from each other, so they can be recombined
64 at prediction time to predict the effect of novel perturbation and covariate combinations. Therefore,
65 by exploring novel combinations, CPA can guide experimental design by directing hypotheses towards
66 expression patterns of interest to experimentalists. We demonstrate the usefulness of CPA on five
67 public datasets and multiple tasks, including the prediction and analysis of responses to compounds,
68 doses, time-series information, and genetic perturbations.

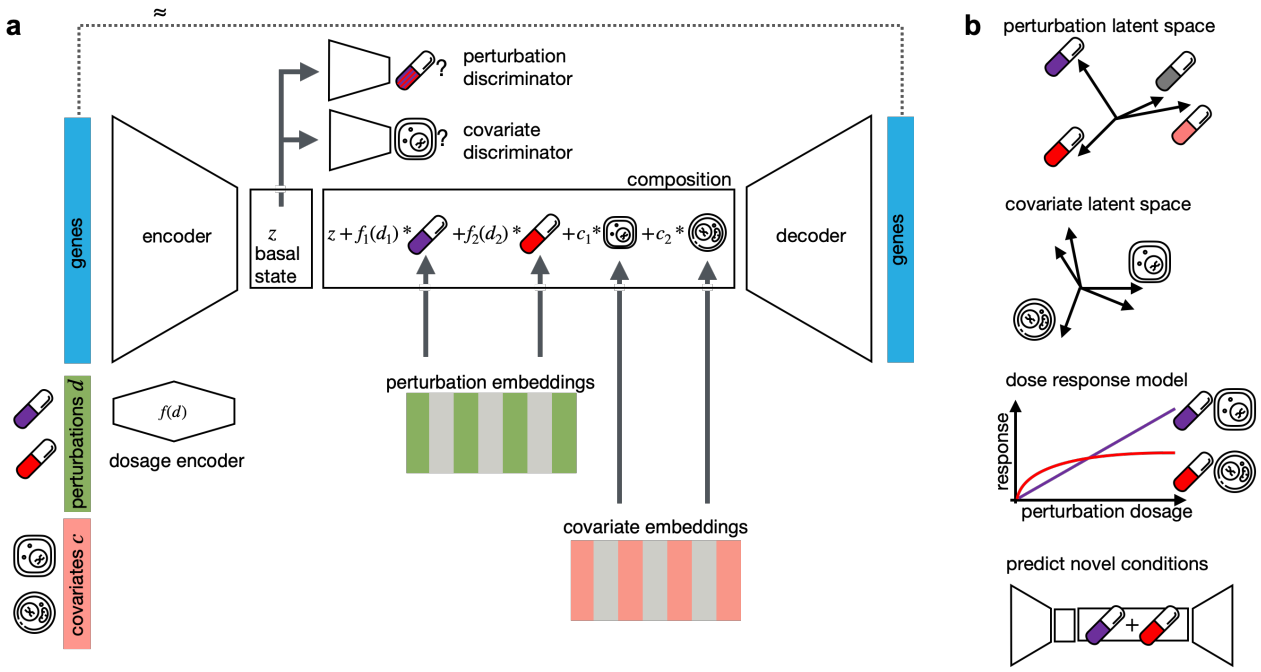


Figure 1: **Interpretable single-cell perturbation modeling using a compositional perturbation autoencoder (CPA). (a)** Given a matrix of gene expression per cell together with annotated potentially quantitative perturbations d and other covariates such as cell line, patient or species, CPA learns the combined perturbation response for a single-cell. It encodes gene expression using a neural network into a lower dimensional latent space that is eventually decoded back to an approximate gene expression matrix, as close as possible to the original one. To make the latent space interpretable in terms of perturbation and covariates, the encoded gene expression vector is first mapped to a "basal state" by feeding the signal to discriminators to remove any signal from perturbations and covariates. The basal state is then composed with perturbations and covariates, with potentially reweighted dosages, to reconstruct the gene expression. All encoder, decoder and discriminator weights as well as the perturbation and covariate dictionaries are learned during training. **(b)** Features of CPA are interpreted via plotting of the two learned dictionaries, interpolating covariate-specific dose response curves and predicting novel unseen drug combinations.

Results

Multiple perturbations as compositional processes in gene expression latent space

Prior work has modeled the effects of perturbations on gene expression as separate processes. While differential expression compares each condition separately with a control, modeling a joint latent space with a conditional variational autoencoder[17, 26, 27] is highly uninterpretable and not amenable to the prediction of the effects of combinations of conditions. Our goal here is to factorize the latent space of neural networks to turn them into interpretable, compositional models. If the latent space were linear, we could describe the observed gene expression as a factor model where each component is a single perturbation.

However, gene expression latent spaces, particularly in complex tissues, are nonlinear and best described by a graph or nonlinear embedding approximations[28, 29]. In scRNA-seq datasets, gene expression profiles of cell populations are often observed under multiple perturbations such as drugs, genetic knockouts, or disease states, in labeled covariates such as cell line, patient, or species. Each cell is labeled with its experimental condition and perturbation, where experimental covariates are

captured in categorical labels and perturbations are captured using a continuous value (e.g. a drug applied with different doses). This assumes a sufficient number of cells per condition to permit the estimation of the latent space in control and perturbation states using a large neural network.

Instead of assuming a factor model in gene expression space, we instead model the nonlinear superposition of perturbation effects in the nonlinear latent space, in which we constrain the superposition to be additive (see **Methods**). We decouple the effects of perturbations and covariates, and allow for continuous effects such as drug dose by encoding this information in a nonlinearly transformed scalar weight: a learned drug-response curve. The linear latent space factor model enables interpretation of this space by disentangling latent space variance driven by covariates from those stemming from each perturbation. At evaluation time, we are able to not only interpolate and interpret the observed perturbation combinations, but also to predict other combinations, potentially in different covariate settings.

Compositional perturbation autoencoder (CPA)

We introduce the CPA (see **Methods**), a method combining ideas from natural language processing [30] and computer vision [31, 32] to predict the effects of combinations of perturbations on single-cell gene expression. Given a single-cell dataset of multiple perturbations and covariates, the CPA first uses an encoder neural network to decompose the cells' gene expression into three learnable, additive embeddings, which correspond to its basal state, the observed perturbation, and the observed covariates. Crucially, the embedding that the CPA encoder learns about a cell's basal state is disentangled from (does not contain information about) the embeddings corresponding to the perturbation and the covariates. This disentangling is achieved by training a discriminator classifier [31] in a competition against the encoder network of the CPA. The goal of the encoder network in the CPA is to learn an embedding representing a cell's basal state, from which the discriminator network cannot predict the perturbation or covariate values. To perform well, the embedding of the cell's basal state should contain all of the information about the cell's specifics. To account for continuous time or dose effects, the learned embeddings about each perturbation are scaled nonlinearly via a neural network which receives the continuous covariate values for each cell, such as the time or the dose. After integration of the learned embeddings about the cell's basal state, perturbations, and covariate values into an unified embedding, the CPA uses a neural network decoder to recover the cell's gene expression vector (**Figure 1**). Similar to many neural network models, the CPA is trained using backpropagation [33] on the reconstruction and discriminator errors (see **Methods**), to tune the parameters of the encoder network, the decoder network, the embeddings corresponding to each perturbation and covariate value, and the dose/time nonlinear scalers. The learned embeddings allow the measurement of similarities between different perturbations and covariates, in terms of their effects on gene expression. The main feature of the CPA is its flexibility of use at evaluation time. After obtaining the disentangled embeddings corresponding to some observed gene expression, perturbation, and covariate values, we can intervene and swap the perturbation embedding with any other perturbation embedding of our choice. This manipulation is effectively a way of estimating the answer to the counterfactual question: what would the gene expression of this cell have looked like, had it been treated differently? This approach is of particular interest in the prediction of unseen perturbation combinations and their effects on gene expression. The CPA can also visualize the transcriptional similarity and uncertainty associated with perturbations and covariates, as later demonstrated.

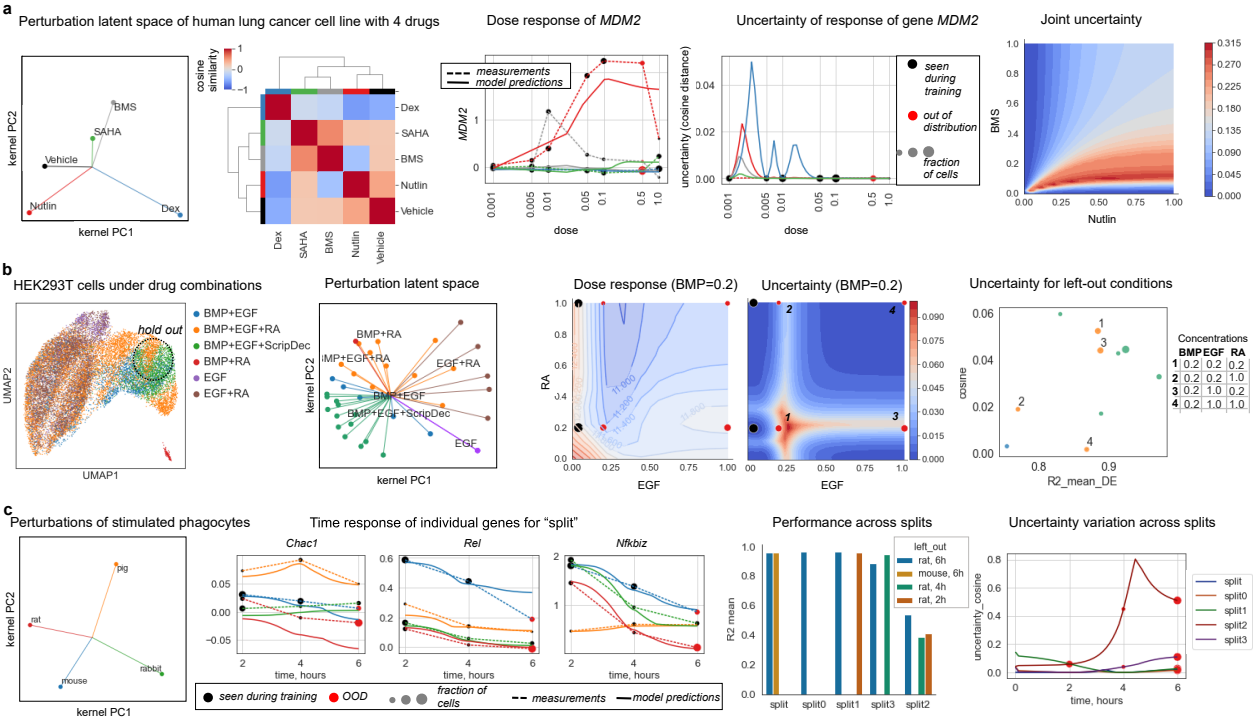


Figure 2: **The CPA learns an interpretable latent space learning across drug dosages, drug combinations and experimental systems**. **(a)** The sci-Plex 2 dataset from Srivatsan et al. [34]. Dose-response curves were generated using the CPA as a transfer from Vehicle cells to a given drug-dose combination. The *MDM2* gene, the top gene differentially expressed after treatment with Nutlin, was selected as an example. Black dots on the dose-response curve denote points seen at training time, red dots denote examples held out for OOD predictions. The sizes of the dots are proportional to the number of cells observed in the experiment. Solid lines correspond to the model predictions, dashed lines correspond to the linear interpolation between measured points. Nutlin and BMS are selected as examples of uncertainty in predictions for drug combinations. **(b)** 96-plex-scRNA-seq experiment from Gehring et al. [8], with UMAP, showing variation of responses in gene expression space. The dashed circle on the UMAP represents the area on the UMAP where the majority of the cells from the left-out (OOD) condition lie. The experiment did not contain samples of individual drugs; therefore we represented the latent space of the drug combinations measured in the experiment. The dose-response surface was obtained via model predictions for a triplet of drugs: BMS at a fixed dose of 0.2, and EGF and RA changing on a grid. **(c)** Cross-species dataset from Hagai et al. [15], with samples of rat and mouse at time point 6 held out from training, and used as OOD. The latent space representation of individual species, and the individual average response of a species across time, demonstrates that the species are fairly different, with a small similarity between rat and mouse. The time response curves of individual genes demonstrate that the model is able to capture nonlinear behavior. The OOD splits benchmark demonstrates the way in which model performance on the distribution case changes when the model is trained on different subsets of the data. Split2 corresponds to the most difficult case, where all three time points for rat were held out from training. Red dots denote examples held out for OOD predictions; the size is of the dots is proportional to the number of cells observed in the experiment.

CPA allows predictive and exploratory analyses of single-cell perturbation experiments.

We first demonstrated the performance and functionality of the CPA on three small single-cell datasets (**Figure** 2): a Sci-Plex2 dataset of human lung cancer cells perturbed by four drugs [35], a 96-plex-scRNA-seq experiment of HEK293T under different drug combinations [8], and a longitoudnal cross-species dataset of lipopolysaccharide (LPS) treated phagocytes [15] (see **Methods**).

All three datasets represent different scenarios of the model application: (i) diverse doses; (ii) drug combinations; and (iii) several Species and variation with respect to time instead of dose. We split each dataset into three groups: train (used for model training), test (used for tuning of the model parameters), and OOD (never seen during training or parameter setting, and intended to measure the generalization properties of the model). **Supplementary Tables** 1–5 shows the R^2 metrics (see Methods) for the performance of the CPA on these datasets and various splits.

Sci-plex from Srivatsan et al. [35] contains measurements of a human lung adenocarcinoma cell line treated with four drug perturbations at increasing doses. In this scenario, the model learns to generalize to the unseen dosages of the drugs. To demonstrate the OOD properties, we withheld cells exposed to the second to largest dose among all drugs. This choice was made because the vast majority of cells are dead for most of the drugs at the highest dosage, and we would not have enough cells to statistically test the generalizability of the CPA model. Since the latent space representation learned by the CPA is still high-dimensional, we can use various dimensionality reduction methods to visualize it, or simply depict it as a similarity matrix (**Figure** 2a). In **Supplementary Table** 2 we compare the performance of the CPA on the OOD example on two simple baselines: taking the maximum dose as a proxy to the previous dose, and a linear interpolation between two measured doses. These results demonstrate that the model consistently achieves high scores (a maximum score of 1 yields perfect reconstruction) on all of the OOD cases, and on two of them significantly outperform the baselines for Nutlin (0.92 vs 0.85) and BMS (0.94 vs 0.89). To demonstrate how well the CPA captured the dose-response dynamics of individual genes, we looked at the top differentially expressed genes upon Nutlin perturbation (**Figure** 2a). The dose-response curve agrees well with the observed data. We additionally propose a simple heuristic to measure the model's uncertainty (see **Methods**) with respect to unseen perturbation conditions. The model shows very low uncertainty on the OOD split. This observation agrees well with the CPA's high R^2 scores on the OOD example. However, when we tested the uncertainty of the model on a combination of two drugs (**Figure** 2a), we saw that it produces much higher uncertainty compared to single drugs. This finding agrees with the fact that the model never saw some drug combinations during training, and that such predictions are more unreliable.

As a the second working example, we took the 96-plex-scRNAse dataset from Gehring et al. [8]. This dataset contains 96 unique growth conditions using combinations of various doses of four drugs applied to HEK293T cells. We hold out several combinations of these conditions as OOD cases, as detailed in (**Supplementary Table** 3). We show that the CPA is able to reliably predict expression patterns of unseen drug combinations (**Supplementary Table** 3) and produce a meaningful latent perturbation latent space (**Figure** 2b). For this dataset, even simple baselines are not applicable anymore, since the expression of cells exposed to the individual drugs were not measured. We also confirmed that our heuristic for the measurement of uncertainty agreed with the model's performance on OOD examples.

As our third example we studied the cross-species dataset from Hagai *et al.*[15]. Here we show that the CPA can also be applied in the setting of multiple covariates, such as different species or cell types, and the dynamics of the covariate can be a non-monotonic function, such as time instead of the dose-response. In this example, bone marrow-derived mononuclear phagocytes from mouse, rat, rabbit, and pig were challenged with LPS (**Figure** 2c). The learned CPA latent space agreed with expected species similarities, with a relatively higher value found between rat and mouse. We compared the generalization abilities of the model by withholding different parts of the data for OOD cases: "splitO" (rat at six hours), "split1" (rat at two and six hours), "split2" (rat at two, four, and six hours), "split3" (rat at four and six hours), and "split" (rat and mouse at six hours. This last split was used for the main analysis) (**Supplementary Table** 4). The model produced high performance values compared to the performance on the test split (see **Supplementary Table** 5) on the majority of the OOD splits, and showed a comparatively lower performance when the model was not exposed to any LPS and rat examples with the exception of control cells. On this dataset, we observed that the model with the lowest performance was the one with the highest number of

held-out examples, yet the model uncertainty also spiked for these OOD cases, suggesting that they might be not reliable (**Figure** 2c). In contrast, for cases with high R^2 scores, models were more certain about these predictions (**Supplementary Table** 4).

CPA finds interpretable latent spaces in large-scale single-cell high-throughput screens

The recently proposed sci-Plex assay [35] profiles thousands of independent perturbations in a single experiment via nuclear hashing. With this high-throughput screen, 188 compounds were tested in 3 cancer cell lines. The panel was chosen to target a diverse range of targets and molecular pathways, covering transcriptional and epigenetic regulators and diverse mechanisms of action. The screened cell lines A549 (lung adenocarcinoma), K562 (chronic myelogenous leukemia), and MCF7 (mammary adenocarcinoma) were exposed to each of these 188 compounds at four doses (10 nM, 100 nM, 1 μM, 10 μM), and scRNA-seq profiles were generated for altogether 290 thousand cells (**Figure** 3a). As above, we split the dataset into 3 subsets: train, test, and OOD. For the OOD case, we held out the highest dose (10 μM) of the 36 drugs with the strongest effect in all three cell lines. Drug, dose, and cell line combinations present in the OOD cases were removed from the train and test sets.

CPA is able to extrapolate to the unseen OOD conditions with unexpected accuracy, as it captures the difference between control and treated conditions also for a compound where it did not see examples with the highest dose. As one example, pracinostat has a strong differential response to treatment compared to control, as can be seen from the distributions of the top 5 differentially expressed genes (**Figure** 3b). Despite not seeing the effect of Pracinostat at the highest dose in any of the three cell lines, CPA correctly infers the mean and distribution of these genes (**Figure** 3b). CPA performs well in modeling unseen perturbations, as the correlation of real and predicted values across OOD conditions is overall better than the correlation between real values (**Figure** 3c). When looking at individual conditions (**Figure** 3d), CPA does well recapitulating genes with low and high mean expression in the OOD condition.

CPA has lower performance when predicting experiments with more unseen covariates. To assess the ability of the model to generalize to unseen conditions, we trained CPA on 28 splits with different held-out conditions, with one of the doses held out in anywhere between 1-3 cell lines (**Figure** 3e). We see here that K562 is the hardest cell line to generalize, when considering training on two cell lines to generalize to another. We also see that extrapolating to the highest dose is a harder task than interpolating intermediate doses, which is consistent with the difficulty of anticipating the experimental effect of a higher dose, versus fitting sigmoidal behavior to model intermediate doses. When examining the shape of the sigmoid per compound learned by the model (**Figure** 3f), we see that epigenetic compounds, which caused the greatest differential expression effects, have higher latent response curves, indicating that CPA learns a general, cell-line agnostic response strength measure for compounds. This learned sigmoid behavior can then be used in conjunction with the latent vectors to reconstruct the gene expression of treated cells over interpolated doses (**Figure** 3g).

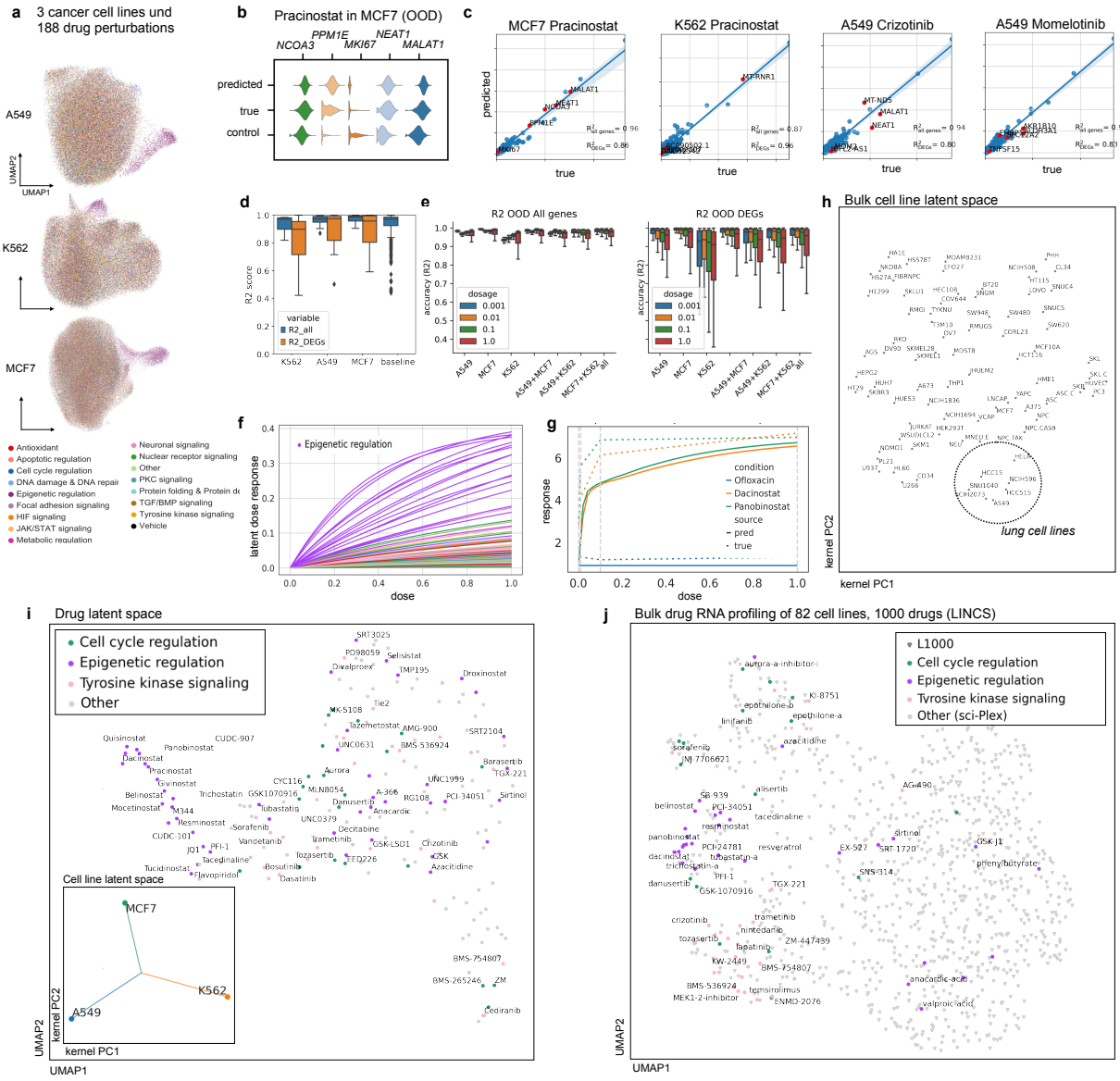


Figure 3: **Learning drug and cell line latent representations from massive single-cell screens of 188 drugs across cancer cell lines.** **(a)** UMAP representation of sci-Plex samples of A549, K562 and MCF7 cell-lines colored by pathway targeted by the compounds to which cells were exposed. **(b)** Distribution of top 5 differentially expressed genes in MCF7 cells after treatment with Pracinostat at the highest dose for real, control and CPA predicted cells. **(c)** Mean gene expression of 5,000 genes and top 50 DEGs between CPA predicted and real cells together with the top five DEGs highlighted in red for four compounds for which the model did not see any examples of the highest dose. **(d)** Box plots of R^2 scores for predicted and real cells for 36 compounds and 108 unique held out perturbations across different cell lines. Baseline indicates comparison of real compounds with each other. **(e)** R^2 scores box plot for all and top 50 DEGs. Each column represents a scenario where cells exposed with specific dose for all compounds on a cell line or combinations of cell lines were held from training and later predicted. **(f)** Latent dose response obtained from dose encoder for all compounds colored by pathways. **(g)** Real and predicted dose response curves based on gene expression data, for a single compound with differential dose response across three cell lines. **(h)** Latent representation of 80 cell lines from L1000 dataset. **(i)** Two dimensional representation of latent drug embeddings as learned by the CPA. Compounds associated with epigenetic regulation, tyrosine kinase signaling, and cell cycle regulation pathways are colored by their respective known pathways. The lower left panel shows latent covariate embedding for three cell lines in the data, indicating no specific similarity preference. **(j)** Latent drug embedding of CPA model trained on the bulk-RNA cell line profiles from the L1000 dataset, with focus on drugs shared with the sci-Plex experiment from (a).

After training, CPA learns a compressed representation of the 188 compounds, where each drug is represented by a single 256 dimensional vector (**Figure** 3i). To test whether the learned drug embeddings are meaningful, we asked if compounds with similar putative mechanisms of action are similar in latent space. This holds for a large set of major mechanisms: we find that epigenetic, tyrosine kinase signaling, and cell-cycle regulation compounds are clustered together by the model, which suggests the effectiveness of drugs with these mechanisms on these three cancer cell-lines which is in line with the findings in the original publication [4].

We additionally demonstrate that the model learns universal relationships between compounds which remain true across datasets and modalities. Using the same set of compounds tested in the sci-Plex dataset together with 853 other compounds (for a total of 1000 compounds), we trained CPA on L1000 bulk perturbation measurement data across 82 cell lines [36]. We observed that CPA works equally well on bulk RNA-seq data, and also that matched epigenetic and tyrosine kinase signaling compounds present in sci-Plex were close to each other in the latent representation, suggesting that the learned model similarities apply across datasets (**Figure** 3j). This holds also for the other learned embeddings: Applying the same similarity metric to the covariate embedding - here the 82 cell lines - we observed that the cell line embedding learned by the model also represents cell line similarity in response to perturbation, as cell lines from lung tissue were clustered together (**Figure** 3h).

CPA allows modeling combinatorial genetic perturbation patterns

Combinatorial drug therapies are hypothesized to address the limited effectiveness of mono-therapies[37] and prevent drug resistance in cancer therapies[37–39]. However, the combined expression of a small number of genes often drives the complexity at the cellular level, leading to the emergence of new properties, behaviors, and diverse cell types [5]. To study such genetic interactions (GIs), recent perturbation scRNA-seq assays allow us to measure the gene expression response of a cell to the perturbation of genes alone or in combination[12, 13]. While experimental approaches are necessary to assess the effect of combination therapies, in practice, it becomes infeasible to experimentally explore all possible combinations without computational predictions.

To pursue this aim, we applied our CPA model to scRNA-seq data collected from Perturb-seq (single-cell RNA-sequencing pooled CRISPR screens) to assess how overexpression of single or combinatorial interactions of 105 genes (i.e., single gene x, single gene y, and pair x+y) affected the growth of K562 cells [5]. In total, this dataset contains 284 conditions measured across $\approx 108,000$ single-cells, where 131 are unique combination pairs (i.e., x+y) and the rest are single gene perturbations or control cells. We observed that the latent genetic interaction manifold placed GIs inducing known and similar gene programs close to each other (**Figure** 4a). For example, consider *CBL* (orange cluster in **Figure** 4a): the surrounding points, comprising its regulators (e.g., *UBASH3A/B*) and multisubstrate tyrosine phosphatases (e.g., *PTPN9/12*), have all been previously reported to induce erythroid markers [5]. Next, we sought to assess our ability to predict specific genetic interactions. We examined a synergistic interaction between *CBL* and *CNN1* in driving erythroid differentiation which has been previously validated [5]. We trained a CPA model with *CBL+CNN1* held out from the training data. Overexpression of either gene leads to the progression of cells from control to single perturbed and doubly perturbed cells (**Supplementary Fig**.2a) toward the erythroid gene program. Overexpression of both *CBL* and *CNN1* up-regulate known gene markers[5] such as hemoglobins (see *HBA1/2* and *HBG1/2* in **Figure 4b**). We observed that our model successfully predicted this synergistic interaction, recapitulating patterns similar to real data and inline with the original findings (**Figure** 4c). We further evaluated CPA to predict a previously reported[5] genetic epistatic interaction between *DUSP9* and *ETS1*, leading to domination of the *DUSP9* phenotype in doubly perturbed cells (**Figure**4 c).

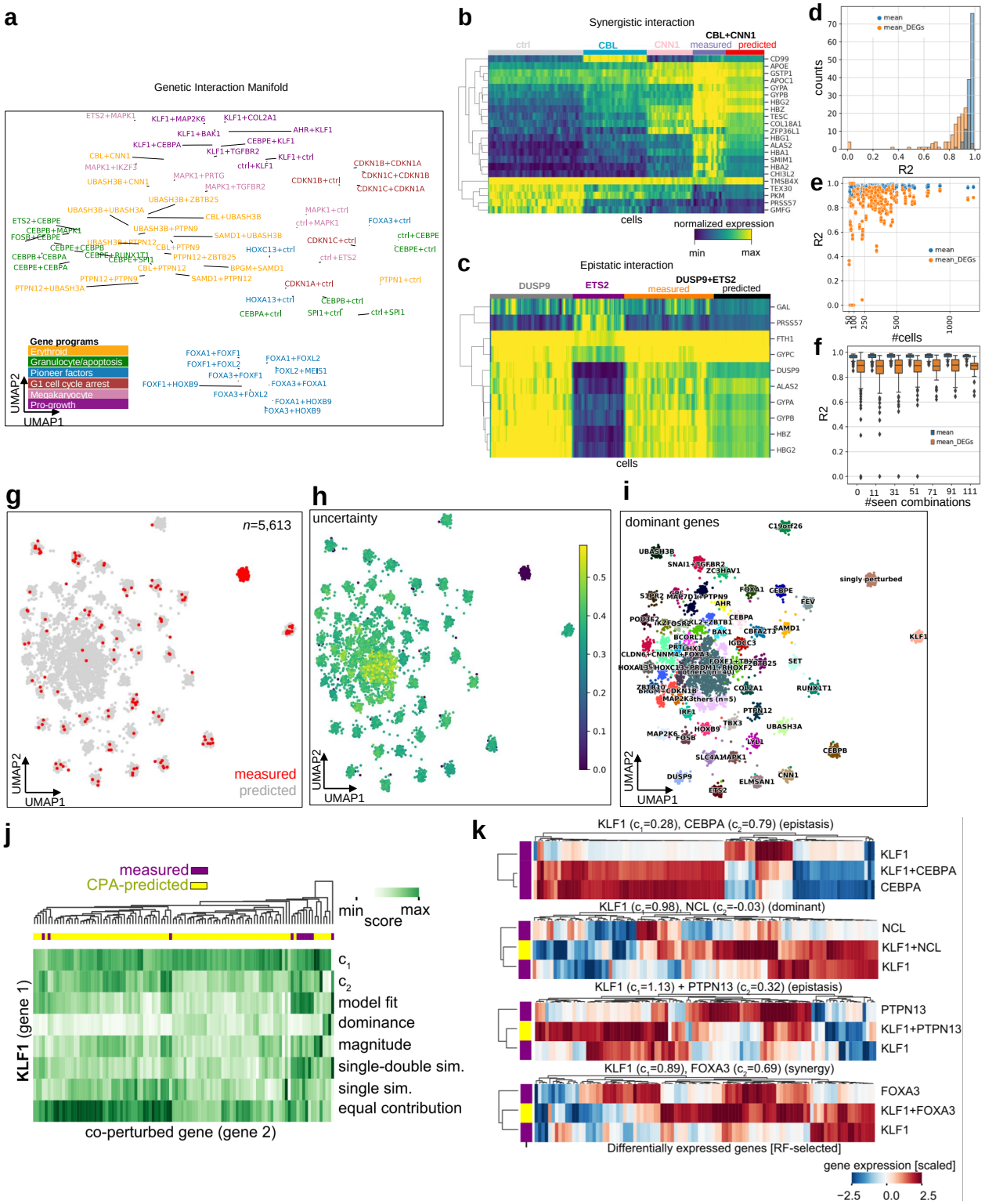


Figure 4: (Continued on the following page)

Figure 4: **Learning and predicting combinatorial genetic perturbations. (a)** UMAP inferred latent space using CPA for 281 single- and double-gene perturbations obtained from Perturb-seq[5]. Each dot represents a genetic perturbation. Coloring indicates gene programs associated to perturbed genes. **(b)** Measured and CPA-predicted gene expression for cells linked to a synergistic gene pair (*CBL+CNN1*). Gene names taken from the original publication. **(c)** As (b) for an epistatic (*DUSP9+ETS*) gene pair. Top 10 DEGs of *DUSP9+ETS* co-perturbed cells versus control cells are shown. **(d)** R2 values of mean gene-expression of measured and predicted cells for all genes (blue) or top 100 DEGs for the prediction of all 131 combinations (13 trained models, with ≈ 10 tested combinations each time) (orange). **(e)** R2 values of predicted and real mean gene-expression versus number of cells in the real data **(h)** R2 values for predicted and real cells versus number of combinations seen during training. **(g)** UMAP of measured (n=284, red dots) and CPA-predicted (n=5,329, gray dots) perturbation combinations. **(h)** As (g), showing measurement uncertainty (cosine similarity). **(i)** As (g), showing dominant genes in leiden clusters (25 or more observations).**(j)** Hierarchical clustering of linear regression associated metrics between *KLF1* with co-perturbed genes, in measured and predicted cells). **(k)** Scaled gene expression changes (versus control) of RF-selected genes (x-axis) in measured (purple) and predicted (yellow) perturbations (y-axis). Headers indicate gene-wise regression coefficients, and interaction mode suggestions[5].

To systemically evaluate the CPA's generalization behavior, we trained 13 different models while leaving out all cells from ≈ 10 unique combinations covering all 131 doubly perturbed conditions in the dataset, which were predicted following training. The reported R^2 values showed robust prediction for most of the perturbations: lower scores were seen for perturbations where the evaluation was noisy due to sample scarcity ($n < 100$), or when one of the perturbations was only available as singly perturbed cells in the data, leading the model to fail to predict the unseen combination (**Figure** 4d-e, see **Supplementary Fig**. 2). To further understand when CPA performance deteriorated, we first trained it on a subset with no combinations seen during training, and then gradually increased the number of combinations seen during training. We found that overall prediction accuracy improved when the model was trained with more combinations, and that it could fail to predict DEGs when trained with fewer combinations (see $n < 71$ combinations in **Figure** 4f).

Hence, once trained with sufficiently large and diverse training data, CPA could robustly predict unseen perturbations. We next asked whether our model could generalize beyond the measured combinations and generate *in-silico* all 5,329 combinations, which were not measured in the real dataset, but made up $\approx 98\%$ of all possibilities. To study the quality of these predictions, we trained a model where all combinations were seen during training to achieve maximum training data and sample diversity. We then predicted 50 single-cells for all missing combinations. We found that, while the latent embeddings did not fully capture all the nuances in the similarity of perturbations compared to gene space, it provided an abstract and easier to perform high-level overview of potential perturbation combinations. Thus, we leveraged our latent space to co-embed (**Figure** 4g) all measured and generated data while proving an uncertainty metric based on the distance from the measured phenotypes (**Figure** 4h). We hypothesized that the closer the generated embedding was to the measured data. the more likely it was to explore a similar space of the genetic manifold around the measured data. Meanwhile, the distant points can potentially indicate novel behaviors, although this would require additional consideration and validation steps. Equipped with this information, we annotated the embedding clusters based on gene prevalence, finding that single genes (i.e. gene x) paired with other genes (i.e., y) as combinations (i.e., x+y) are a main driver of cluster separation (**Figure** 4i). Genes without measured double perturbations were less likely to be separated as independent clusters using the newly predicted transcriptomic expression (**Supplementary Fig**. 3a), suggesting that their interaction-specific effects were less variable than cases with at least one double perturbation available in the training data.

To investigate the type of interaction between the newly predicted conditions, we compared the differences between double and single perturbations versus control cells and thus annotated their

interaction modes (adapted from [5] for *in silico* predictions). In each gene-specific cluster, we observed variability across these values, suggesting that our predictions contained granularity that went beyond single gene perturbation effects, and could not be fully dissected by two dimensional embeddings. Upon curation of gene perturbations using these metrics and the levels of experimental data available (**Supplementary Fig**. 3b), we decided to predict and annotate interaction modes based on these values when double measurements were available for at least one gene. For example, we observed clustering of *KLF1* and partner gene perturbation pairs solely from these metrics, suggesting the existence of several interaction modes (**Figure** 4j). When we further examine the differentially expressed genes in each co-perturbation, our metrics validated previously reported epistatic interactions (*CEBPA*), and proposed new ones with *KLF1*-dominant behavior (*NCL*), gene synergy (*FOXA3*), and epistasis (*PTPN13*), among others (**Figure** 4k). Repeating this analysis across all measured and predicted double perturbations, we found genes with potential interaction prevalences (**Supplementary Fig**. 3c). Among genes which repeatedly respond to several perturbations, we found common gene expression trends in both direction and magnitude (**Supplementary Fig.** 3d), suggesting that variation is modulated by conserved gene regulatory principles that are potentially captured in our learned model.

Altogether, our analysis indicated that double perturbation measurements can be generated by CPA by leveraging genetic perturbation data, which when combined with an uncertainty metric allows us to generate and interpret gene regulatory rules in the predicted gene-gene perturbations.

Discussion

In-silico prediction of cell behavior in response to a perturbation is critical for optimal experiment design and the identification of effective drugs and treatments. With CPA, we have introduced a versatile and interpretable approach to modeling cell behaviors at single-cell resolution. CPA is implemented as a neural network trained using stochastic gradient descent, scaling up to millions of cells and thousands of genes.

We applied CPA to a variety of datasets and tasks, from predicting single-cell responses to learning embeddings, as well as reconstructing the expression response of compounds, with variable drug-dose combinations. Specifically, we illustrated the modeling of perturbations across dosage levels and time series, and have demonstrated applications in drug perturbation studies, as well as genetic perturbation assays with multiple gene knockouts, revealing potential gene-gene interaction modes inferred by our model predicted values. CPA combines the interpretability of linear decomposition models with the flexibility of nonlinear embedding models.

While CPA performed well in our experiments, it is well known that in machine learning there is no free lunch, and as with any other machine learning model, CPA will fail if the test data are very different from the training data. To alert CPA users to these cases, it is crucial to quantify model uncertainty. To do so, we implemented a distance-based uncertainty score to evaluate our predictions. Additionally, scalable Bayesian uncertainty models are promising alternatives for future work[40]. Although we opted to implement a deterministic autoencoder scheme, extensions towards variational models[17, 23], as well as cost functions other than mean squared error[22] are straightforward.

Aside from CPA, existing methods[17, 26] such as scGen[16] have also been shown capable of predicting single-cell perturbation responses when the dataset contains no combinatorial treatment or dose-dependent perturbations. Therefore, it may be beneficial to benchmark CPA against such methods on less complicated scenarios with few perturbations. However, this approach might not be practical, considering the current trend towards the generation of massive perturbation studies[4, 5, 12].

Currently, the model is based on gene expression alone, so it cannot directly capture other levels of interactions or effects, such as those due to post-transcriptional modification, signaling, or cell communication. However, due to the flexibility of neural network-based approaches, CPA could

be extended to include other modalities, for example via multimodal single-cell CRISPR[41, 42] combined scRNA- and ATAC-seq[43, 44] and CUT&Tag[45, 46]. In particular, we expect spatial transcriptomics[47, 48] to be a valuable source for extensions to CPA due to its high sample number and the dominance of DL models in computer vision.

The CPA model is not limited to single-cell perturbations. While we chose the single-cell setting due to the high sample numbers available, the CPA could readily be applied to large-scale bulk cohorts, in which covariates might be patient ID or transcription factor perturbation. These and any other available attributes could be controlled independently[31] to achieve compositional, interpretable predictions. Any bulk compositional model may be combined with a smaller-scale single-cell model to compose truly multi-scale models of observed variance. The flexibility of the DL setting will also allow addition of constraints on perturbation or covariate latent spaces. These could, for example, be the similarity of chemical compounds[49], or clinical-covariate induced differences of patient IDs. The key feature of the CPA versus a normal autoencoder is its latent space disentanglement and the induced interpretability of the perturbations in the context of cell states and covariates. Eventually, any aim in biology is not only blind prediction, but mechanistic understanding. This objective is exemplified by the direction that DL models are taking in sequence genomics, where the aim is not only the prediction of new interactions, but also the interpretation of the learned gene regulation code. We therefore believe that CPA can not only be used as a hypothesis generation tool for *in-silico* screens but also as an overall data approximation model. Deviations from our assumed data generation process (see **Methods**) would then tell us about missing information in the given data set and/or missing aspects in the factor model. By including multiple layers of regulation, the resulting model can grow in flexibility for prediction and for mechanistic understanding on for example synergistic gene regulation or other interactions.

Finally, we expect CPA to facilitate new opportunities in expression-based perturbation screening, not only to learn optimal drug combinations, but also in how to personalize experiments and treatments by tailoring them based on individual cell response.

Code availability

Code to reproduce all of our results is available at `http://github.com/facebookresearch/CPA`.

Data availability

All datasets analyzed in this manuscript are public and have published in other papers. We have referenced them in the manuscript and made available at `http://github.com/facebookresearch/CPA`.

Author Contributions

M.L., A.K.S, and D.L.P. conceived the project with contributions from F.J.T. D.L.P., M.L. and A.K.S designed the algorithm and implemented the first version. Y.J., C.D. and A.K.S. performed the first refactor. The final code is implemented by D.L.P. and A.K.S. with contributions from C.D., Y.J. and M.L. M.L. and C.D. curated all the datasets. F.A.W. helped interpret the model and results. M.L., A.K.S. and F.J.T. designed analyses and use-cases. M.L., A.K.S., Y.J., I.L.I. and C.D performed the analysis. F.J.T. and N.Y. supervised the research. All authors wrote the manuscript.

Acknowledgments

M.L. and F.J.T. are grateful for valuable feedback from Aviv Regev and Dana Pe'er. We appreciate support from all members of Theis lab, specifically Malte D. Luecken and Fabiola Curion for their feedback and proof-reading. M.L is thankful for early graphical designs by Monir Jazaeri (Jaz) which did not make it to the final version of the paper. F.J.T. acknowledges support by the BMBF (grant L031L0214A, grant 01IS18036A and grant 01IS18053A), by the Helmholtz Association (Incubator grant sparse2big, grant ZT-I-0007) and by the Chan Zuckerberg Initiative DAF (advised fund of Silicon Valley Community Foundation, 2018-182835 and 2019-207271). This work was further supported by Helmholtz Association's Initiative and Networking Fund through Helmholtz AI [grant ZT-I-PF-5-01]. I.L.I. has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 874656.

Competing interests

F.J.T. reports receiving consulting fees from Roche Diagnostics GmbH and Cellarity Inc., and ownership interest in Cellarity, Inc. and Dermagnostix. F.A.W. is a full-time employee of Cellarity Inc., and has ownership interest in Cellarity, Inc.

Methods

Data generating process

We consider a dataset $\mathcal{D} = \{(x_i, d_i, c_i)\}_{i=1}^N$, where each $x_i \in \mathbb{R}^G$ describes the gene expression of G genes from cell i. The perturbation vector $d_i = (d_{i,1}, \ldots, d_{i,M})$ contains elements $d_{i,j} \geq 0$ describing the dose of drug j applied to cell i. If $d_{i,j} = 0$, this means that perturbation j was not applied to cell i. Unless stated otherwise, the sequel assumes column vectors. Similarly, the vector of vectors $c_i = (c_{i,1}, \ldots c_{i,K})$ contains additional discrete covariates such as cell-types or species, where each covariate is itself a vector. More specifically, $c_{i,j}$ is a K_j-dimensional one-hot vector.

We assume that an unknown generative model produced our dataset $\mathcal{D}$. The three initial components of this generative process are a latent (unobserved) basal latent state z_i^{basal} for cell i, together with its perturbation vector d_i and covariate vector c_i. We assume that the basal latent state is independent from the perturbation vector d_i and covariate vector c_i. Next, we form the latent (also unobserved) perturbed latent state z_i as:

$$z_i = z_i^{\text{basal}} + V^{\text{perturbation}} \cdot (f_1(d_{i,1}), \ldots, f_M(d_{i,M})) + \sum_{j=1,\ldots,K} V^{\text{cov}_j} \cdot c_{i,j} \tag{1}$$

In this equation, each column of the matrix $V^{\text{perturbation}} \in \mathbb{R}^{d \times M}$ represents a d-dimensional embedding for one of the M possible perturbations represented in d_i. Similarly, each column of the matrix $V^{\text{cov}_j} \in \mathbb{R}^{d \times K_j}$ represents a d-dimensional embedding for the j-th discrete covariate, represented as a K_j-dimensional one-hot vector $c_{i,j}$. The functions $f_j : \mathbb{R} \to \mathbb{R}$ scale non-linearly each of the $d_{i,j}$ in the perturbation vector d_i, therefore implementing M independent dose-response (or time-response) curves. Finally, we assume that the generative process returns the observed gene expression x_i by means of an unknown decoding distribution $p(x_i|z_i)$. This process builds the observation (x_i, d_i, c_i), which is then included in our dataset $\mathcal{D}$.

Compositional Perturbation Autoencoder (CPA)

Assuming the generative process described above, our goal is to train a machine learning model $x'_i = M((x_i, d_i, c_i), d')$ such that, given a dataset triplet (x_i, d_i, c_i) as well as a target perturbation d', estimates the gene expression x'_i. The term x'_i represents what would the counterfactual distribution of the gene expression x_i with covariates c_i look like, had it been perturbed with d' instead of d_i.

Given a dataset and a learning goal, we are now ready to describe our proposed model, the Compositional Perturbation Autoencoder (CPA). In the following, we describe separately how to train and test CPA models.

Training

The training of a CPA model consists in auto-encoding dataset triplets (x_i, d_i, c_i). That is, during training, a CPA model does not attempt to answer counterfactual questions. Instead, the training process consists in (1) encoding the gene expression x_i into an estimated basal state $\hat{z}_i^{\text{basal}}$ that does not contain any information about (d_i, c_i), (2) combining $\hat{z}_i^{\text{basal}}$ with learnable embeddings about (d_i, c_i) to form an estimated perturbed state $\hat{z}_i$, and (3) decoding $\hat{z}_i$ back into the observed gene expression x_i.

More specifically, the CPA model first encodes the observed gene expression x_i into an estimated basal state:

$$\hat{z}_i^{\text{basal}} = \hat{f}^{\text{enc}}(x_i).$$

In turn, the estimated basal state is used to compute the estimated perturbed state $\hat{z}_i$:

$$\hat{z}_i := \hat{z}_i^{\text{basal}} + \hat{V}^{\text{perturbation}} \cdot (\hat{f}_1(d_{i,1}), \dots \hat{f}_M(d_{i,M})) + \sum_{j=1,\dots,K} \hat{V}^{\text{cov}_j} \cdot c_{i,j} \tag{2}$$

Contrary to (1), this expression introduces three additional learnable components: the perturbation embeddings $\hat{V}^{\text{perturbation}}$, the covariate embeddings $\hat{V}^{\text{cov}}$and the learnable dose-response curves $(\hat{f}_1, \dots, \hat{f}_M)$, here implemented as small neural networks constrained to satisfy $\hat{f}_j(0) = 0$.

As a final step, a decoder $\hat{f}^{\text{dec}}$ accepts the estimated perturbed state $\hat{z}_i$ and returns $\hat{f}_\mu^{\text{dec}}(\hat{z}_i)$ and $\hat{f}_{\sigma^2}^{\text{dec}}(\hat{z}_i)$, that is, the estimated mean and variance of the counterfactual gene expression x'_i.

To train CPA models, we use three loss functions. First, the reconstruction loss function is the Gaussian negative log-likelihood:

$$\ell_i := \frac{\log s(\hat{f}_{\sigma^2}^{\text{dec}}(\hat{z}_i))}{2} + \frac{(\hat{f}_\mu^{\text{dec}}(\hat{z}_i) - x'_i)^2}{2 \cdot s(\hat{f}_{\sigma^2}^{\text{dec}}(\hat{z}_i))}, \tag{3}$$

where $s(\sigma^2) = \log(\exp(\sigma^2 + 10^{-3}) + 1)$ enforces a positivity constraint on the variance and adds numerical stability. This loss function rewards the end-to-end auto-encoding process if producing the observed gene expression x_i.

Second, and according to our assumptions about the data generating process, we are interested in removing the information about (d_i, c_i) from $\hat{z}_i^{\text{basal}}$. To achieve this information removal, we follow an adversarial approach [31]. In particular, we set up the following auxiliary loss functions:

$$\ell_i^d := \text{CrossEntropy}(\hat{f}_d^{\text{adv}}(\hat{z}_i^{\text{basal}}), d_i),$$
$$\ell_{i,j}^c := \text{CrossEntropy}(\hat{f}_{c_{i,j}}^{\text{adv}}(\hat{z}_i^{\text{basal}}), c_{i,j}), \quad \forall j = 1, \dots, K.$$

The functions $\hat{f}_d^{\text{adv}}$, $\hat{f}_{c_{i,j}}^{\text{adv}}$ are a collection of neural network classifiers trying to predict about (d_i, c_i) given the estimated basal state $\hat{z}_i^{\text{basal}}$.

Given this collection of losses, the training process is an optimization problem that repeats the following two steps:

1. sample $(x_i, d_i, c_i) \sim \mathcal{D}$, minimize $\ell_i^d + \sum_j \ell_{i,j}^c$ by updating the parameters of $\hat{f}_d^{\text{adv}}$ and $\hat{f}_{c_{i,j}}^{\text{adv}}$, for all $j = 1, \ldots, K$;
2. sample $(x_i, d_i, c_i) \sim \mathcal{D}$, minimize $\ell_i - \lambda \cdot (\ell_i^d + \sum_j \ell_{i,j}^c)$ by updating the parameters of the encoder $\hat{f}^{\text{enc}}$, the decoder $\hat{f}^{\text{dec}}$, the perturbation embeddings $\hat{V}^{\text{perturbation}}$, the covariate embeddings $\hat{V}^{\text{cov}_j}$ for all $j = 1, \ldots, K$, and the dose-response curve estimators $(\hat{f}_1, \ldots, \hat{f}_M)$.

Testing

Given an observation (x_i, d_i, c_i) and a counterfactual treatment d', we can use a trained CPA model to answer what would the counterfactual distribution of the gene expression x_i with covariates c_i look like, had it been perturbed with d' instead of d_i. To this end, we follow the following process:

1. Compute the estimated basal state $\hat{z}_i^{\text{basal}} = \hat{f}^{\text{enc}}(x_i)$;
2. Compute the counterfactual perturbed state $\hat{z}_i'$

$$\hat{z}_i' := \hat{z}_i^{\text{basal}} + \hat{V}^{\text{perturbation}} \cdot (\hat{f}_1(d_{i,1}'), \ldots \hat{f}_M(d_{i,M}')) + \sum_{j=1,\ldots,K} \hat{V}^{\text{cov}_j} \cdot c_{i,j}.$$

   Note that in the previous expression, we are using the counterfactual treatment d' instead of the observed treatment d_i.

3. Compute and return the counterfactual gene expression mean $x_{i,\mu}'$:

$$x_{i,\mu}' = \hat{f}_\mu^{\text{dec}}(\hat{z}_i'),$$

   and variance x_{i,σ^2}':

$$x_{i,\sigma^2}' = \hat{f}_{\sigma^2}^{\text{dec}}(\hat{z}_i').$$

Hyper-parameters and training.

For each dataset, we perform a random hyper-parameter search of 100 trials. The table below outlines the distribution of values for each of the hyper-parameters involved in CPA training.

Group	Hyperparameter	Default value	Random search distribution
general	embedding dimension	256	RandomChoice([128, 256, 512])
	batch size	128	RandomChoice([64, 128, 256, 512])
	learning rate decay, in epochs	45	RandomChoice([15, 25, 45])
nonlinear scalers	hidden neurons, nonlinear scalers	64	RandomChoice([32, 64, 128])
	hidden layers	2	RandomChoice([1, 2, 3])
	learning rate	1e-3	$10^{\text{Uniform}}(-4, -2)$
	weight decay	1e-7	$10^{\text{Uniform}}(-8, -5)$
encoder and decoder	hidden neurons, encoder and decoder	512	RandomChoice([256, 512, 1024])
	hidden layers	4	RandomChoice([3, 4, 5])
	learning rate	1e-3	$10^{\text{Uniform}}(-4, -2)$
	weight decay	1e-6	$10^{\text{Uniform}}(-8, -4)$
discriminator	hidden neurons, discriminator	128	RandomChoice([64, 128, 256])
	hidden layers	3	RandomChoice([2, 3, 4])
	regularization strength	5	$10^{\text{Uniform}}(-2, 2)$
	gradient penalty	3	$10^{\text{Uniform}}(-2, 1)$
	learning rate	3e-4	$10^{\text{Uniform}}(-5, -3)$
	weight decay	1e-4	$10^{\text{Uniform}}(-6, -3)$
	number of learning steps	3	RandomChoice([1, 2, 3, 4, 5])

Model evaluation.

We use several metrics to evaluate the performance of our model: (1) quality of reconstruction for in and OOD cases and (2) quality of disentanglement of cell information from perturbation information.

We split each dataset into 3 subsets: train, test, and OOD. For OOD cases, we choose combinations of perturbations that exhibit unseen behavior. This usually corresponds to the most extreme drug dosages. We select one perturbation combination as "control". Usually these are Vehicle or DMSO if real control samples are present in the dataset, otherwise we choose a drug perturbation at a lower dosage as "control". For the evaluation, we use the mean squared error of the reconstruction of an individual cell and average it over the cells for the perturbation of interest. As an additional metric we use classification accuracy in order to check how well the information about the drugs was separated from the information about the cells.

Uncertainty estimation.

To estimate the uncertainty of the predictions we use as a proxy the minimum distance between the queried perturbation and the set of conditions (covariate + perturbation combinations) seen during training (**Supplementary Fig.1**). Intuitively, we expect predictions on queried conditions that are more distant from the set of seen conditions to be more uncertain. To estimate this distance we first compute the set of embeddings of the training covariate and perturbation combinations:

$$\hat{z}^{comb} = \hat{V}^{\text{perturbation}} \cdot (\hat{f}_1(d'_1), \dots \hat{f}_M(d'_M)) + \sum_{j=1,\dots,K} \hat{V}^{\text{cov}_j} \cdot c_j. \tag{4}$$

The latent vector for the queried condition is obtained in the same manner. The cosine and euclidean distances from the training embedding set are computed and the minimum distance is used as a proxy for uncertainty.

$$u_{cosine} = \min(1 - \mathbf{S_C}(\hat{z}^{query}, \hat{z}^{comb})) \tag{5}$$

$$u_{eucl} = \min(\mathbf{d}(\hat{z}^{query}, \hat{z}^{comb})) \tag{6}$$

Where $\mathbf{S_C}(\mathbf{x}, \mathbf{y})$ stands for the cosine similarity and $\mathbf{d}(x, y)$ for the euclidean distance between the two vectors.

With this methodology, in the case of a drug screening experiment, if we query a combination of cell type, drug, and dosage that was seen during training, we get an uncertainty of zero, since this combination was present in the training set. It is important to note that with this method we obtain a condition-level uncertainty, in that all cells predicted under the same query will have the same uncertainty, thus not taking cell specific information into account.

R2 score

We used the *r2_score* function from *scikit-learn* which reports R2 (coefficient of determination) regression score.

Datasets

Gehring *et al.*

This dataset[8] comprises of 21,191 neural stem cells (NSCs) cells perturbed with EGF/bFGF, BMP4, decitabine, scriptaid, and retinoic acid. We obtained normalized data from the original authors and after QC filtering 19,637 cells remained. We further selected 5,000 highly variable genes (HVGs) using SCANPY's[50] *highly_variable_genes* function for training and evaluation of the model.

Genetic CRISPR screening experiment

We obtained the raw count matrices from Norman *et al.*[5] from GEO (accession ID GSE133344). According to authors guide, we excluded "NegCtrl1_NegCtrl0__NegCtrl1_NegCtrl0" control cells

and merged all unperturbed cells as one "ctrl" condition. We then normalized and log-transformed the data using SCANPY and selected 5,000 HVGs for training. The processed dataset contained 108,497 cells.

Cross-species experiment

The data was generated by Hagai *et al.*[15] and downloaded from ArrayExpress (accession: E-MTAB-6754). The data consists of 119,819 phagocytes obtained from four different species: mouse, rat, pig and rabbit. Phagocytes were treated with lipopolysaccharide (LPS) and the samples were collected at different time points: 0 (control), 2, 4, and 6 hours after the beginning of treatment. All genes from non-mouse data were mapped to the respective orthologs in the mouse genome using Ensembl ID annotations. We filtered out cells with a percentage of counts belonging to mitochondrial genes higher than 20%, then proceeded to normalize and log-transform the count data. For training and evaluation, we selected 5000 HVG using SCANPY. After filtering, the data consists of 113,400 cells.

sci-Plex 2

The data was generated by Srivatsan *et al.* [35] and downloaded from GEO (GSM4150377). The dataset consists of A549 cells treated with one of the following four compounds: dexamethasone, Nutlin-3a, BMS-345541, or vorinostat (SAHA). The treatment lasted 24 hours across seven different doses. The count matrix obtained from GEO consists of 24,262 cells. During QC we filtered cells with fewer than 500 counts and 720 detected genes. We discarded cells with a percentage of mitochondrial gene counts higher than 10%, thus reducing the dataset to 14,811 cells. Genes present in fewer than 100 cells were discarded. We normalized the data using the size factors provided by the authors and log-transformed it. We selected 5000 HVGs for training and further evaluations.

sci-Plex 3

The data was generated by Srivatsan *et al.*[35] and downloaded from GEO (GSM4150378). The dataset consists of three cancer cell lines (A549, MCF7, K562), which are treated with 188 different compounds with different mechanisms of action. The cells are treated with 4 dosages (10, 100, 1000, and 10000 nM) plus vehicle. The count matrix obtained from GEO consists of 581,777 cells. The data was subset to half its size, reducing it to 290,888 cells. We then proceeded with log-transformation and the the selection of 5000 HVGs using SCANPY.

Interpretation of combinatorial genetic interactions by perturbation pairs and responder genes

In the case of genetic screening, previous work by [5] proposed a set of metrics to annotate and classify gene-gene interactions based on responder genes. Based on this, here we used measured or predicted gene expression differences with respect to control cells (δ), for gene perturbations **a** (δa), **b** (δb) and double perturbations **ab** (δab), to calculate interaction types by similarity between these three expression vectors.

More specifically, to calculate association coefficients, we use the linear regression coefficients c_1 and c_2 obtained from the model

$$\delta ab = \delta a c_1 + \delta b c_2 \tag{7}$$

To describe interaction modes, we used the following metrics.

1. **similarity between predicted and observed values**: $dcor(\delta a c_1 + \delta b c_2, \delta ab)$.
2. **linear regression coefficients**: c_1 and c_2.
3. **magnitude**: $(c_1^2 + c_2^2)^{1/2}$.
4. **dominance**: $|log_{10}(c_1/c_2)|$.

5. **similarity of single transcriptomes**: $dcor(a, b)$
6. **similarity of single to double transcriptomes**: $dcor([a, b], ab)$.
7. **equal contributions**: $\frac{\min(dcor(a,b),dcor(b,ab)}{\max(dcor(a,b),dcor(a,ab)}$.

Following clustering and comparison of these metrics across measured and predicted cells, we decided the following rules of thumb to define and annotate interaction modes:

1. **epistatic**: $min(abs(c_1), abs(c_2)) > 0.2$ and either **(i)** $(abs(c_1) > 2abs(c_2))$ or **(ii)** $(abs(c_2) > 2abs(c_1))$
2. **potentiation**: magnitude $>$ 1 and abs($dcor(a, b)$) - 1 $>$ 0.2.
3. **strong sinergy (similar phenotypes)**: magnitude $>$ 1 and abs($dcor([a, b], ab)$) - 1 $>$ 0.2
4. **strong sinergy (different phenotypes)**: magnitude $>$ 1 and abs($dcor(a, b)$) - 1 $>$ 0.5.
5. **additive**: abs(magnitude) - 1 $<$ 0.1.
6. **redundant**: abs($dcor([a, b], ab)$) - 1 $<$ 0.2 and abs($dcor(a, b)$) - 1 $<$ 0.2

More than one genetic interaction can be suggested from these rules. In those cases, we did not assign any plausible interaction. For visualization purposes, we consider perturbed genes with 50 or more interaction modes reported with other co-perturbed genes (**Supplementary Fig.3c**).

To visualize differentially expressed genes with similar response across perturbations (**Supplementary Fig.3d**), we trained a random forest classifier using as prediction labels *control*, a, b and ab cells, and gene expression as features. We retrieved the top 200 genes from this approach. Then, we annotated the direction (positive or negative) and the magnitude of those changes versus control cells, generating a code for clustering and visualization. To label genes with potential interaction effects, we labeled them if the double perturbation predicted magnitude is 1.5x times or higher than the best value observed in single perturbations.

References

[1] Pisco, A. O. et al. A single cell transcriptomic atlas characterizes aging tissues in the mouse. BioRxiv 661728 (2019).

[2] Domcke, S. et al. A human cell atlas of fetal chromatin accessibility. Science **370** (2020).

[3] Han, X. et al. Construction of a human cell landscape at single-cell level. Nature 1–9 (2020).

[4] Srivatsan, S. R. et al. Massively multiplex chemical transcriptomics at single-cell resolution. Science **367**, 45–51 (2020).

[5] Norman, T. M. et al. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. Science **365**, 786–793 (2019). Publisher: American Association for the Advancement of Science Section: Research Article.

[6] Yofe, I., Dahan, R. & Amit, I. Single-cell genomic approaches for developing the next generation of immunotherapies. Nature medicine **26**, 171–177 (2020).

[7] McGinnis, C. S. et al. Multi-seq: sample multiplexing for single-cell rna sequencing using lipid-tagged indices. Nature methods **16**, 619–626 (2019).

[8] Gehring, J., Park, J. H., Chen, S., Thomson, M. & Pachter, L. Highly multiplexed single-cell rna-seq by dna oligonucleotide tagging of cellular proteins. Nature Biotechnology **38**, 35–38 (2020).

[9] Sachs, S. et al. Targeted pharmacological therapy restores β-cell function for diabetes remission. Nature Metabolism **2**, 192–209 (2020).

[10] Kim, K.-T. et al. Application of single-cell rna sequencing in optimizing a combinatorial therapeutic strategy in metastatic renal cell carcinoma. Genome biology **17**, 1–17 (2016).

[11] Al-Lazikani, B., Banerji, U. & Workman, P. Combinatorial drug therapy for cancer in the post-genomic era. Nature biotechnology **30**, 679–692 (2012).

[12] Dixit, A. et al. Perturb-seq: Dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. Cell **167**, 1853–1866.e17 (2016).

[13] Datlinger, P. et al. Pooled crispr screening with single-cell transcriptome readout. Nature methods **14**, 297–301 (2017).

[14] Rozenblatt-Rosen, O., Stubbington, M. J., Regev, A. & Teichmann, S. A. The human cell atlas: from vision to reality. Nature News **550**, 451 (2017).

[15] Hagai, T. et al. Gene expression variability across cells and species shapes innate immunity. Nature **563**, 197–202 (2018).

[16] Lotfollahi, M., Wolf, F. A. & Theis, F. J. scgen predicts single-cell perturbation responses. Nature methods **16**, 715–721 (2019).

[17] 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).

[18] Yuan, B. et al. Cellbox: Interpretable machine learning for perturbation biology with application to the design of cancer combination therapy. Cell Systems **12**, 128–140 (2021).

[19] Fröhlich, F. et al. Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. Cell systems **7**, 567–579 (2018).

[20] Rampášek, L., Hidru, D., Smirnov, P., Haibe-Kains, B. & Goldenberg, A. Dr.VAE: improving drug response prediction via modeling of drug perturbation effects. Bioinformatics **35**, 3743–3751 (2019).

[21] Kamimoto, K., Hoffmann, C. M. & Morris, S. A. Celloracle: Dissecting cell identity via network inference and in silico gene perturbation. bioRxiv (2020).

[22] Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell rna-seq denoising using a deep count autoencoder. Nature communications **10**, 1–14 (2019).

[23] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. Nature methods **15**, 1053–1058 (2018).

[24] Lotfollahi, M. et al. Query to reference single-cell integration with transfer learning. bioRxiv (2020).

[25] Lopez, R., Gayoso, A. & Yosef, N. Enhancing scientific discoveries in molecular biology with deep generative models. Molecular Systems Biology **16**, e9198 (2020).

[26] Russkikh, N. et al. Style transfer with variational autoencoders is a promising approach to rna-seq data harmonization and analysis. Bioinformatics **36**, 5076–5085 (2020).

[27] Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. Advances in neural information processing systems **28**, 3483–3491 (2015).

[28] McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018).

[29] Van der Maaten, L. & Hinton, G. Visualizing data using t-sne. Journal of machine learning research **9** (2008).

[30] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546 (2013).

[31] Lample, G. et al. Fader networks: Manipulating images by sliding attributes. In Advances in neural information processing systems, 5967–5976 (2017).

[32] Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015).

[33] Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. Deep learning, vol. 1 (MIT press Cambridge, 2016).

[34] Srivatsan, S. R. et al. Massively multiplex chemical transcriptomics at single-cell resolution. Science **367**, 45–51 (2020). Tex.ids: srivatsanMassivelyMultiplexChemical2020a, srivatsanMassivelyMultiplexChemical2020b publisher: American Association for the Advancement of Science section: Research Article.

[35] Srivatsan, S. R. et al. Massively multiplex chemical transcriptomics at single-cell resolution. Science **367**, 45–51 (2020).

[36] Musa, A. et al. Systems pharmacogenomic landscape of drug similarities from lincs data: drug association networks. Scientific reports **9**, 1–16 (2019).

[37] Menden, M. P. et al. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. Nature Communications **10**, 2674 (2019). Number: 1 Publisher: Nature Publishing Group.

[38] Adam, G. et al. Machine learning approaches to drug response prediction: challenges and recent progress. npj Precision Oncology **4**, 19 (2020).

[39] Jia, J. et al. Mechanisms of drug combinations: interaction and network perspectives. Nature Reviews Drug Discovery **8**, 111–128 (2009). Number: 2 Publisher: Nature Publishing Group.

[40] Gal, Y. & Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, 1050–1059 (PMLR, 2016).

[41] Frangieh, C. J. et al. Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. Nature genetics 1–10 (2021).

[42] Papalexi, E. et al. Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. Nature Genetics 1–10 (2021).

[43] Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. Nature biotechnology **37**, 1452–1457 (2019).

[44] Clark, S. J. et al. scnmt-seq enables joint profiling of chromatin accessibility dna methylation and transcription in single cells. Nature communications **9**, 1–9 (2018).

[45] Kaya-Okur, H. S. et al. Cut&tag for efficient epigenomic profiling of small samples and single cells. Nature communications **10**, 1–10 (2019).

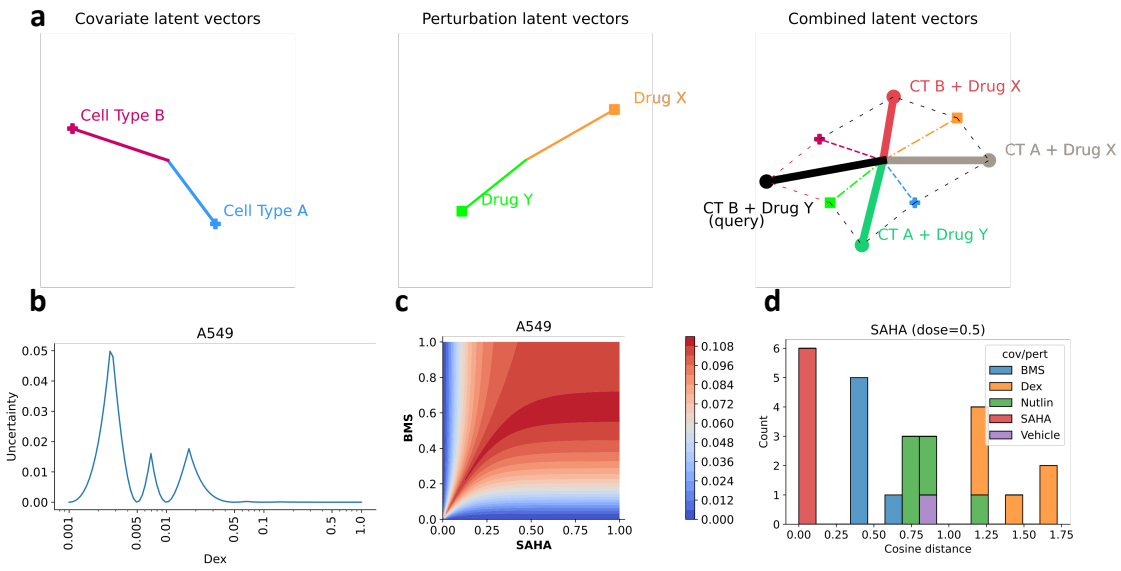
[46] Wu, S. J. et al. Single-cell CUT&Tag analysis of chromatin modifications in differentiation and tumor progression. Nature Biotechnology 1–6 (2021). Publisher: Nature Publishing Group.

[47] van den Brink, S. C. et al. Single-cell and spatial transcriptomics reveal somitogenesis in gastruloids. Nature **582**, 405–409 (2020).

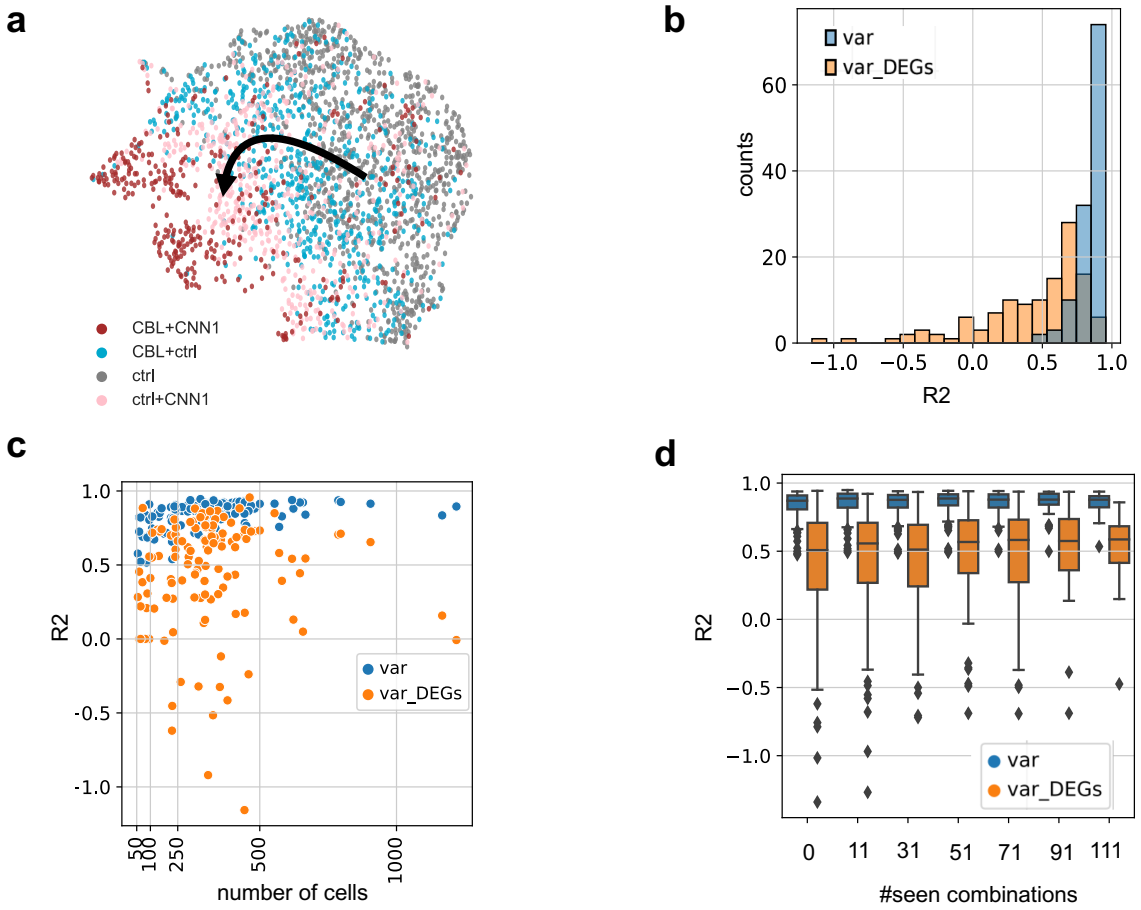
[48] Rodriques, S. G. et al. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. Science **363**, 1463–1467 (2019).

[49] Mater, A. C. & Coote, M. L. Deep learning in chemistry. Journal of chemical information and modeling **59**, 2545–2559 (2019).

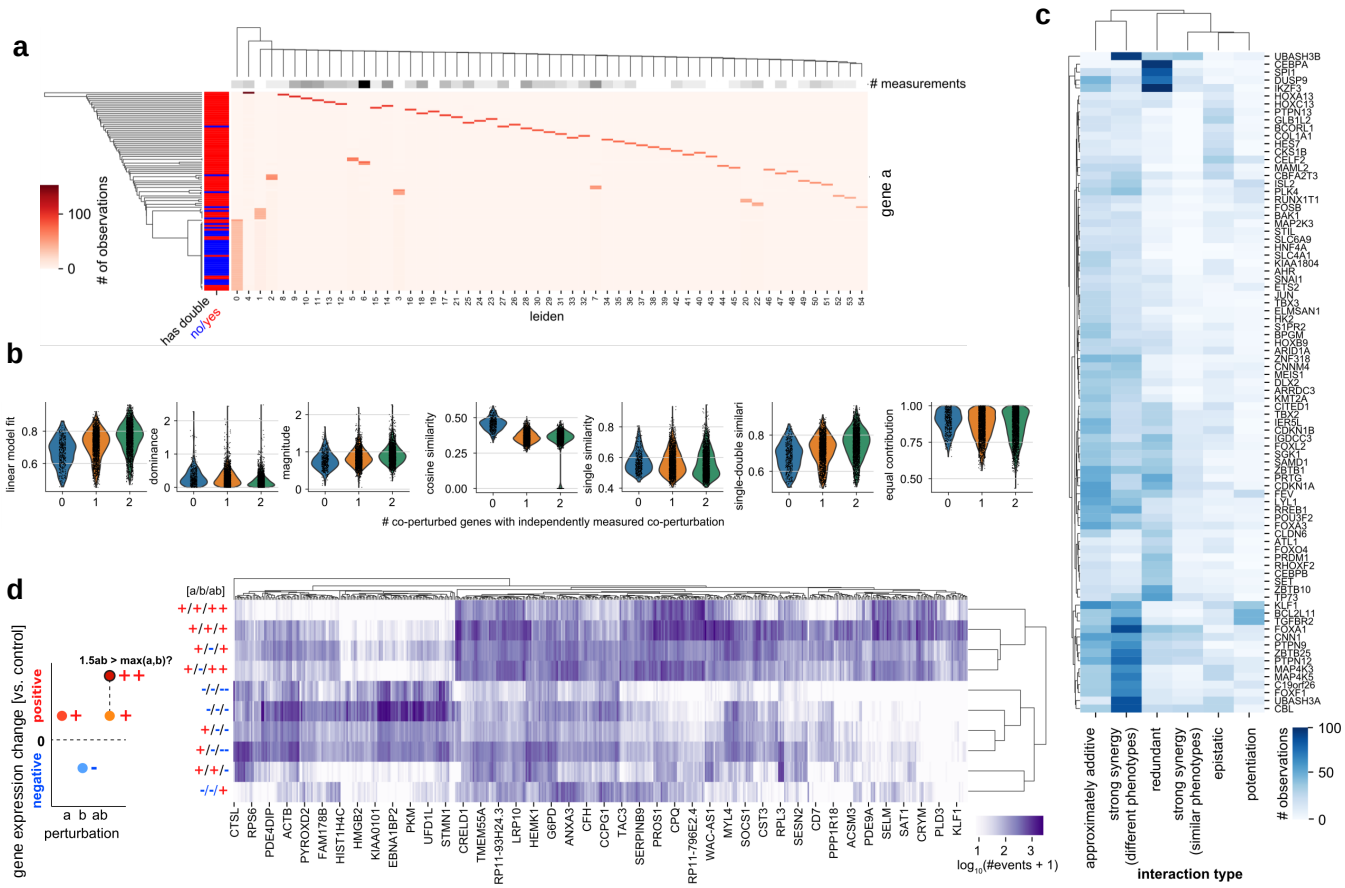
[50] Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. Genome biology **19**, 1–5 (2018).



Supplementary Figure 1: **CPA uncertainty estimation.** **(a)** Schematic representation of the steps involved in uncertainty estimation in the case of a dataset with two cell types and two drugs (single dosage per drug). The covariate and perturbation latent vectors are summed in order to generate the set of combinations in the training set. The distances from the query vector and all the vectors in the set are then computed. The closest distance is used as a proxy for uncertainty in the prediction of the model. **(b)** Example of uncertainty across dosages of Dexamethasone in the sci-Plex 2 dataset. The ticks on the x-axis (log-scaled) indicate dosages seen at training time for which the uncertainty is 0. The dosages were min-max normalized. **(c)** 2D plot of uncertainty across dosages (min-max normalized) of two different drugs and combinations thereof in the sci-Plex 2 dataset. **(d)** Example histogram of cosine distances between the SAHA (dose=0.5) and the vectors in the set of training perturbations. The distribution shows that training vectors belonging to the same perturbation but with different dosages have the lowest uncertainties, with other drugs being increasingly more distant.



Supplementary Figure 2: **Performance evaluation for CPA combinatorial predictions.** **(a)** UMAP representation of control (ctrl), singly perturbed (CBL+ctrl, ctrl+CNN1) and doubly (CBL+CNN1) perturbed cells. **(b)** R^2 scores for all genes (blue) or top 100 DEGs (orange) for the prediction of all 131 combinations in the data by training 13 different models and leaving out ≈ 10 combinations each time. **(c)** Scatter plots of number of samples in the real data for each combination (x-axis) versus R^2 values for the variance of predicted and real for that combination **(d)** Box-plots of R^2 values for variance for predicted and real cells while increasing the number of combinations seen during training.



Supplementary Figure 3: **Gene-gene interaction insights revealed from genetic perturbation predictions using CPA. (a)** Number of single gene observations in leiden clusters for generated measurements (from Figure 4i). Most leiden clusters contain a prevalence for one perturbed gene. The majority of genes without measured double perturbations share a limited number of clusters. **(b)** Quality control and interaction metrics to compare gene expression differences between single and double perturbations. Metrics vary based on number of genes with a measured double perturbation (See **Methods** for definitions). **(c)** Interaction mode counts predicted for all genes based on interaction metrics (based on [5]). **(d)** (*left*) Gene expression changes for double perturbations (ab) versus single perturbations (a, b), are compared by direction and magnitude. Positive (+) and negative (-) labels indicate increase/decrease versus control cells, and double positive/negative (++/- -) indicate values higher than 1.5 times the highest comparable value in single perturbations. (*right*) 500 genes with highest prevalence in differentially expressed genes across datasets, clustered by prevalent response types from single and double perturbations.

condition	dose_val	R2_mean	R2_mean_DE	R2_var	split	num_cells
SAHA	0.01	0.99	0.99	0.95	test	160
SAHA	0.005	0.98	0.98	0.93	test	143
SAHA	0.05	0.98	0.95	0.93	test	118
SAHA	1.0	0.98	0.95	0.91	test	137
SAHA	0.001	0.97	0.97	0.92	test	169
SAHA	0.1	0.96	0.86	0.94	test	129
SAHA	0.5	0.96	0.86	0.89	**ood**	604
Nutlin	0.001	0.98	0.98	0.94	test	135
Nutlin	0.05	0.98	0.98	0.94	test	136
Nutlin	0.005	0.98	0.97	0.94	test	107
Nutlin	0.1	0.98	0.97	0.94	test	200
Nutlin	0.01	0.98	0.97	0.93	test	180
Nutlin	0.5	0.92	0.86	0.84	**ood**	265
Nutlin	1.0	0.26	0.61	0.00	test	1
Dex	0.5	0.99	0.99	0.98	**ood**	864
Dex	1.0	0.99	0.98	0.95	test	222
Dex	0.1	0.99	0.94	0.96	test	218
Dex	0.05	0.98	0.90	0.93	test	210
Dex	0.001	0.95	0.87	0.89	test	123
Dex	0.01	0.95	0.61	0.89	test	238
Dex	0.005	0.94	0.53	0.88	test	108
BMS	0.001	0.98	0.97	0.92	test	212
BMS	0.005	0.97	0.99	0.92	test	151
BMS	0.5	0.95	0.89	0.78	**ood**	34
BMS	0.01	0.95	0.80	0.86	test	82
BMS	0.05	0.93	0.87	0.75	test	59
BMS	0.1	0.92	0.89	0.74	test	50
BMS	1.0	0.55	-0.87	0.21	test	6

Supplementary Table 1 | Performance scores for the sci-Plex 2 dataset. To improve readability the columns are sorted by: condition (first priority) and scores (second priority).

condition	R2_mean	R2_mean_DE	method
SAHA	0.98	0.94	linear
SAHA	0.96	0.86	CPA
Nutlin	0.92	0.86	CPA
Nutlin	0.85	0.80	linear
Dex	1.00	1.00	linear
Dex	0.99	0.99	CPA
BMS	0.95	0.89	CPA
BMS	0.89	0.85	linear

Supplementary Table 2 | A simple benchmark on OOD split for the sci-Plex 2 dataset.

condition	dose_val	R2_mean	R2_mean_DE	R2_var	split	num_cells
EGF+RA	0.2+1.0	0.98	0.95	0.84	test	553
EGF+RA	1.0+1.0	0.97	0.94	0.67	test	199
EGF+RA	0.2+0.2	0.96	0.98	0.89	test	87
EGF+RA	0.04+1.0	0.96	0.90	0.60	test	54
EGF+RA	1.0+0.2	0.95	0.91	0.75	test	30
EGF	0.2	0.98	0.96	0.86	test	90
EGF	1.0	0.94	0.71	0.60	test	73
BMP+RA	0.2+1.0	0.91	0.84	0.60	test	22
BMP+EGF+ScripDec	0.2+0.2+1.0	0.97	0.97	0.82	**ood**	166
BMP+EGF+ScripDec	0.04+0.04+1.0	0.97	0.96	0.61	test	28
BMP+EGF+ScripDec	0.04+0.2+1.0	0.97	0.94	0.50	test	39
BMP+EGF+ScripDec	0.2+0.2+0.2	0.97	0.92	0.65	**ood**	304
BMP+EGF+ScripDec	0.04+0.2+0.2	0.97	0.91	0.74	test	33
BMP+EGF+ScripDec	0.2+0.04+1.0	0.96	0.96	0.26	test	32
BMP+EGF+ScripDec	0.2+0.04+0.2	0.96	0.94	0.34	test	20
BMP+EGF+ScripDec	1.0+0.2+0.2	0.96	0.91	0.74	**ood**	113
BMP+EGF+ScripDec	0.2+1.0+1.0	0.95	0.89	0.51	**ood**	112
BMP+EGF+ScripDec	0.04+0.04+0.2	0.95	0.87	0.52	test	19
BMP+EGF+ScripDec	0.2+1.0+0.2	0.95	0.83	0.58	**ood**	105
BMP+EGF+ScripDec	0.04+1.0+1.0	0.94	0.96	0.57	test	17
BMP+EGF+ScripDec	1.0+0.04+0.2	0.91	0.86	0.51	test	15
BMP+EGF+RA	0.04+1.0+0.2	0.98	0.99	0.85	test	50
BMP+EGF+RA	0.2+0.04+0.2	0.98	0.98	0.74	test	63
BMP+EGF+RA	0.04+1.0+1.0	0.98	0.97	0.86	test	198
BMP+EGF+RA	0.04+0.2+1.0	0.97	0.96	0.88	test	552
BMP+EGF+RA	0.2+0.04+1.0	0.97	0.96	0.70	test	48
BMP+EGF+RA	0.04+0.2+0.2	0.97	0.92	0.80	test	73
BMP+EGF+RA	0.2+0.2+0.2	0.97	0.88	0.52	**ood**	206
BMP+EGF+RA	0.04+0.04+0.2	0.96	0.96	0.73	test	24
BMP+EGF+RA	0.2+1.0+0.2	0.96	0.89	0.66	**ood**	216
BMP+EGF+RA	0.2+1.0+1.0	0.96	0.87	0.66	**ood**	147
BMP+EGF+RA	0.2+0.2+1.0	0.95	0.77	0.59	**ood**	132
BMP+EGF	0.04+0.2	0.97	0.98	0.78	test	96
BMP+EGF	0.04+1.0	0.97	0.92	0.81	test	209
BMP+EGF	0.04+0.04	0.97	0.92	0.75	test	39
BMP+EGF	1.0+0.04	0.95	0.86	0.33	test	19
BMP+EGF	1.0+1.0	0.94	0.75	0.06	**ood**	113

Supplementary Table 3 | Performance scores for the 96-plex-scRNAseq dataset. For the readability the columns are sorted by: condition (first priority) and scores (second priority).

split	species	time	R2_mean	R2_mean_DE	R2_var	num_cells
split4	rat	6.0	0.97	0.91	0.85	7827
split4	rat	6.0	0.96	0.86	0.89	7827
split4	mouse	6.0	0.97	0.90	0.81	5280
split4	mouse	6.0	0.96	0.85	0.93	5280
split3	rat	6.0	0.89	0.70	0.72	7827
split3	rat	6.0	0.86	0.55	0.40	7827
split3	rat	4.0	0.95	0.80	0.80	5755
split3	rat	4.0	0.94	0.77	0.63	5755
split2	rat	6.0	0.55	-0.65	-2.18	7827
split2	rat	6.0	0.54	0.02	0.02	7827
split2	rat	4.0	0.74	-0.47	0.26	5755
split2	rat	4.0	0.39	-0.85	-0.47	5755
split2	rat	2.0	0.81	-0.41	0.49	7025
split2	rat	2.0	0.41	-0.96	-0.64	7025
split1	rat	6.0	0.97	0.91	0.92	7827
split1	rat	6.0	0.97	0.91	0.92	7827
split1	rat	2.0	0.96	0.90	0.90	7025
split1	rat	2.0	0.96	0.90	0.90	7025
split0	rat	6.0	0.97	0.89	0.90	7827
split0	rat	6.0	0.96	0.90	0.81	7827

Supplementary Table 4 | Performance scores for the cross-species dataset across different splits.

species	time	R2_mean	R2_mean_DE	R2_var	split	num_cells
rat	2.0	1.00	1.00	0.99	test	2138
rat	4.0	1.00	1.00	0.99	test	1715
rat	6.0	0.96	0.86	0.89	**ood**	7827
rabbit	6.0	0.99	0.99	0.99	test	2088
rabbit	2.0	0.99	0.99	0.98	test	2662
rabbit	4.0	0.99	0.96	0.98	test	1732
pig	6.0	0.99	0.99	0.99	test	1535
pig	4.0	0.99	0.98	0.99	test	1954
pig	2.0	0.99	0.98	0.98	test	1662
mouse	2.0	1.00	0.99	0.99	test	2904
mouse	4.0	1.00	0.99	0.99	test	2793
mouse	6.0	0.96	0.85	0.93	**ood**	5280

Supplementary Table 5 | Performance scores for the cross-species dataset.