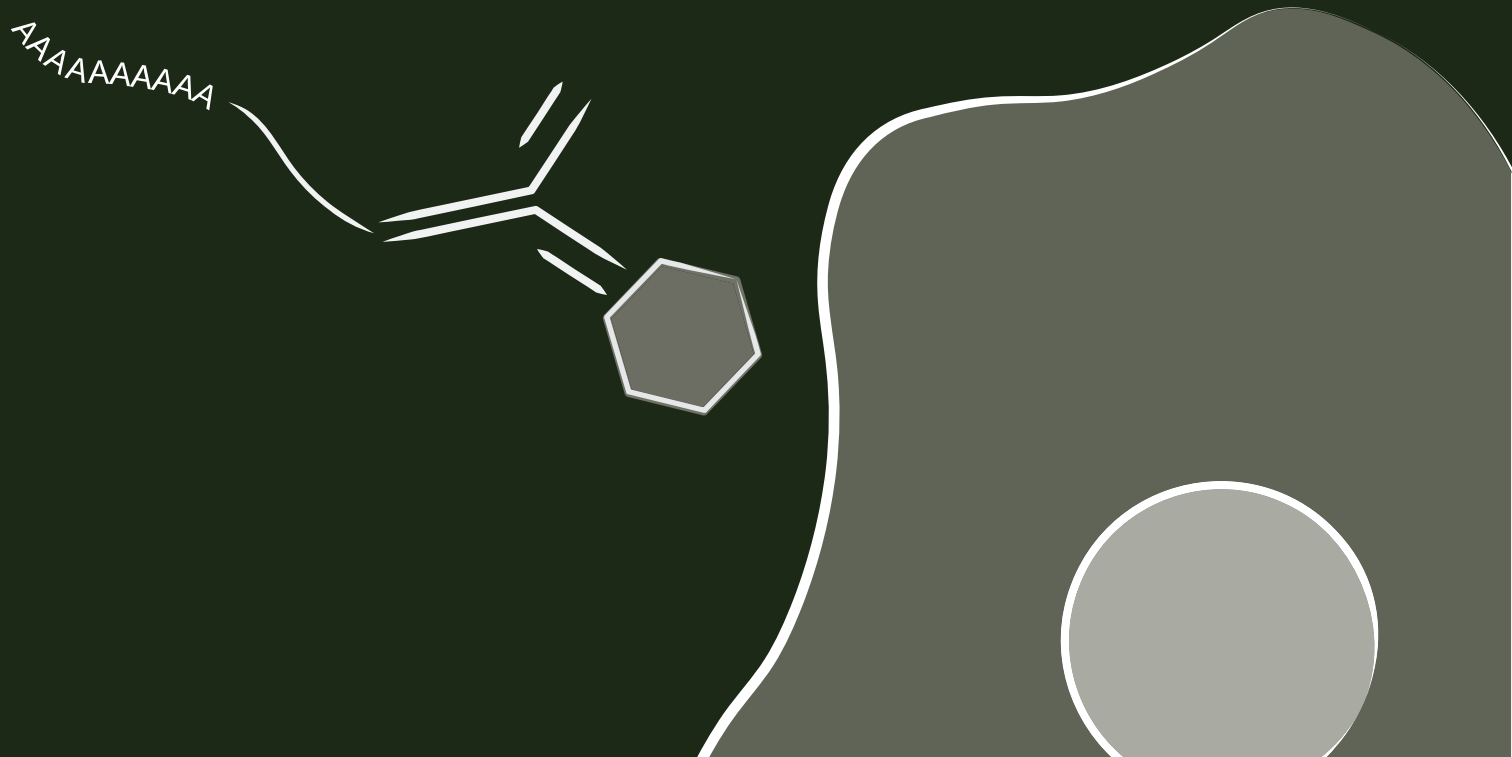
Predicting immune responses on multi-modal single-cell data with variational inference

Francesca Kathrin Drummer

TU Delft | Master Thesis 2022

Faculty of Electrical Engineering,
Mathematics and Computer Science



Predicting immune responses on multi-modal single-cell data with variational inference

by

F.K. Drummer

to obtain the degree of Master of Science at Delft University of Technology,
to be defended publicly on Wednesday July 6, 2022 at 13.00.

Student number:	5413990
Project duration:	November 29, 2021 – July 6, 2022
Master programme:	Computer science, Artificial Intelligence Technology track, Bioinformatics track
Faculty:	Electrical Engineering, Mathematics and Computer Science
Thesis committee:	Prof. Dr. Ir. Marcel Reinders (TU Delft) Dr. Ahmed Mahfouz (TU Delft & LUMC) Dr. Thomas Höllt (TU Delft) Mikhael Manurung (LUMC)

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

With pleasure I present to you my master thesis '*Predicting immune responses on multi-modal single-cell data with variational inference*'. This thesis is written to fulfil the final requirement to obtain the degree of Master of Science at the Delft University of Technology and as such marks the end of an exciting 6-year journey in the Netherlands.

F.K. Drummer
Delft, July 2022

Contents

Abstract

Single-cell sequencing allows measuring individual cells' molecular features and their responses to perturbations. Understanding which cells respond to a particular perturbation and how these responses vary across populations can be used to, for example, improve vaccine immunogenicity. However, an exhaustive exploration of single-cell perturbation responses in every population is usually experimentally unfeasible. Several machine learning models have been developed to predict perturbation responses, but they are limited to single-modality data. Single-modality data alone, such as only transcriptomics, is not suited to capture all cell responses accurately. For example, the identification of immune responses requires transcriptomic and proteomic measurements. Here, we introduce cellPMVI, a method built to predict perturbation responses from multi-modality data. MMVI combines the single-cell data modeling from scVI [21] with a mixture-of-experts posterior integration [31], to allow for multi-modality input data. In this work, we validate cellPMVI for immune response prediction of adjuvants across populations. The model is trained on two-modality CITE-seq data containing gene and protein measurements from three different populations. We show that cellPMVI can model both modalities of the CITE-seq data without information loss in either modality and predict immune responses with a high correlation to the observed responses across different populations. Hence, cellPMVI is the first model to capture and predict immune response for multi-modality data with the potential to be applied for other perturbations, such as drugs.

1 Introduction

A central means of studying cellular networks is to observe cell state changes after perturbations. Perturbations describe a functional alteration to a biological system through an external event such as gene knockdowns or other stimuli, e.g., drugs [12]. Understanding the single-cell responses to perturbations can be helpful in various contexts. For example, gene knockouts can identify cellular pathways that provide the basis for many biological processes, such as tissue repair. Furthermore, recording cell responses to stimuli, e.g., drugs, can help develop combination therapy treatments [18] or give insight into the gene expression variability. Recording gene expression changes under perturbations can help identify cell subsets that get activated from e.g., vaccinations.

An interesting observation in single-cell perturbation experiments is that different population groups do not necessarily show similar cellular responses to vaccinations [13]. Here, the term population group defines a group of people living in an environment with the same urban setting, i.e., (i) urban-dutch, (ii) urban-Senegalese, (iii) rural Senegalese. Research suggests that adjuvants might be responsible for the variation in vaccine efficiency across population groups [13]. Adjuvants are vaccination components that enhance the immune response to the antigen by activating specific cell subsets. For example, we observed that cell type responses to the vaccine adjuvant Monophosphoryl Lipid A (MPL) are distinct across populations (Figure 1.1). The distinct immune activation pattern suggests that vaccine components, including adjuvant, can be further modified to ensure comparable efficacy across populations [29].

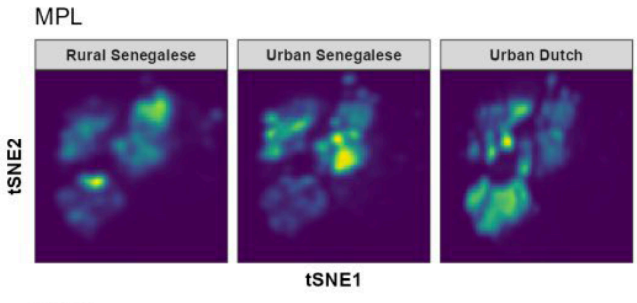


Figure 1.1: **Cell subset activation response to MPL A across populations.** Response to Monophosphoryl Lipid A (MPL) stimulation across population groups (rural Senegalese, urban Senegalese and urban dutch). Cytokine-producing cells were selected and then embedded using tSNE dimensionality reduction. The color indicates the density of the data points, with lighter green indicating a higher density of cells.

A common challenge for perturbation experiments is their need for large-scale evidence. For instance, finding the optimal adjuvant for a population group requires recording the responses of many individuals in a specific population group to draw a valid conclusion. Due to the mainly observational nature and cost intensity of single-cell studies, large-scale experiments are usually infeasible. Therefore, instead of basing the adjuvant selection on purely experimental measurements, one could employ computational methods, such as perturbation modeling to generalize single-cell responses across

population groups. Computational models would overcome the need for large-scale perturbation experiments, reducing the number of experimental screenings needed.

A successful perturbation model must be able to address different prediction scenarios, such as out-of-distribution (OOD) prediction. OOD prediction describes situations where the predicted perturbation is unknown and different from the training data. In this work we distinguish between two OOD prediction scenarios: extrapolation to other adjuvants (Figure 1.2a) and extrapolation to novel population groups (Figure 1.2b). Figure 1.2 illustrates the two scenarios for three population groups and two adjuvants (*unperturbed* (A) and *perturbed* (B)). For the first scenario (Figure 1.2a), the perturbed data from the orange population group is left out during training. The aim is to predict cell responses of the orange population group under perturbation, so for the red adjuvant. This requires extrapolation from unperturbed to perturbed cell responses given the responses from the observed, green and blue, population groups. Figure 1.2b shows the second scenario in which the orange population group is left out entirely during training. The goal is to predict novel population group responses by extrapolation from known population group responses. Both scenarios describe variations of OOD prediction that can be used to preselect several promising adjuvants, which could then be validated experimentally. OOD prediction can potentially reduce the required number of experimental screens drastically [22].

(a) Extrapolation to other adjuvants

(b) Extrapolation to novel population groups

Figure 1.2: **Two scenarios for out-of-distribution prediction that are addressed in this work.** In this thesis OOD prediction will be performed over three population groups and two adjuvants representing the unperturbed and perturbed condition. The task is to predict the response of the unseen population group depicted with a question mark. Subfigure (a) shows the task of predicting the perturbed cell activity given the unperturbed cell response and knowing the perturbed activation of the green and blue population groups and (b) illustrates the prediction of cell activity for a unknown population group.

Currently, only a handful of the more than 1 000 available tools for single-cell data analysis (scrna-tools.org) can be used for perturbation prediction [37]. Three examples of perturbation prediction models are scGen [22], trVAE [24] and CPA [23]. scGen [22] is one of the earliest methods to predict single-cell perturbation responses using a variational autoencoder (VAE) architecture. A major limitation of scGen is that it is restricted to *one*-to-*one* prediction. That means scGen is only suited to predict one kind of perturbations. Predicting new perturbations requires retraining the model and calclating a new difference vector without profiting from the information of the previous perturbation. trVAE (transformer VAE) [24] overcomes the problem of *one*-to-*one* predictions by using a conditional VAE to integrate information of multiple perturbations into one latent space, allowing for *n*-to-*n* predictions. A problem with trVAE is that the model is entirely black-boxed, limiting its interpretability. That means interpretation of the latent space for further research (e.g., differential expression or visualization) or manipulation is impossible. A new VAE-based method, CPA (Compositional Perturbation Autoencoder) [23], achieves interpretability by decomposing the latent space through adversarial training, allowing it flexible recombination in the latent area. However, CPA does not support novel predictions (e.g., predicting new drugs) because of the necessity to learn embeddings for the latent space recombination. Finally, a common limitation of all three models (scGen, trVAE and CPA) is that they only support single-modality data input, ignoring important information accessible through multi-modality data.

Multi-modality, such as CITE-seq, data can often characterize a cell's identity better, especially in immune responses. For example, protein data is necessary to distinguish functionally distinct categories of immune cells that are similar on a transcriptomic level. Furthermore, immune cells are studied using cytometry which analyzes the expression of proteins on the surface of the cells. Hence, protein data is beneficial, or even essential, to leverage knowledge about immune responses such as infections and vaccinations [5]. On the other hand, protein data alone is insufficient because proteome-wide measurements require a preselection of proteins and a monoclonal antibody to target the epitope of the proteins. The preselection would bias the analysis toward a specific collection of proteins, miss

heterogeneity, and bias towards preexisting knowledge. For example, specific cell populations such as CD8+and CD4+T cells can only be detected through a combination of protein and transcriptome data measurements [9]. Thus, it is necessary to account for both transcriptome and proteome data to understand immune cell responses best.

Currenlty, totalVI [7] and scMM [28] are the currently most promising single-cell multi-modality models based on a VAE architecture. TotalVI [7] is designed for integration of multi-modalities but suffers from a 'black-box' nature making the prediction of new perturbations difficult. On the other hand, scMM [28] models modalities separately before joining them to learn common features which improves its interpretability and enables cross-modality prediction. Nevertheless, neither of these two models were designed for perturbation modelling and prediction.

Thus, at the moment there are no single-cell models that combine perturbation modelling with multi-modality data integration. In this work we introduce cellPMVI, a single-cell Perturbation prediction model on Multi-modal data with variational Inference. cellPMVI is based on scVI [21] and MMVAE [31] for integration and prediction of multi-modality data. We will use cellPMVI to answer the research questions introduced in the following section.

1.1. Research questions

The main research question is defined as:

How well can we model immune responses across populations with CITE-seq data?

This is further divided into the following sub-question:

RQ 1: *How does multi-modality information impact the modelling and prediction performance?*

RQ 2: *To what extend can we predict responses for unseen perturbations across populations?*

First, we compare the performance of three models that are either based on single- or multi-modality information to understand whether the additional proteins data increases the performance of a model. After that, the second question aims to explore if a model can generalize as far as to generate new data or make prediction of unseen responses. This is related to the two scenarios shown in Figure 1.2 and answer the following two counterfactual questions:

- *What would have happened if population 1 had received adjuvant B, instead of adjuvant A?* (Figure 1.2a)
- *Given the response from population 1 and 2 to adjuvant A, how would population 3 respond to adjuvant B?* (Figure 1.2b)

2

Preliminaries

This thesis focuses on employing techniques from *generative modelling* and *deep learning* to the problem domain of modelling and predicting immune responses using single-cell data. In this chapter, we will first motivate why generative modelling and specifically, deep generative modelling with variational autoencoder is suited for single-cell perturbation modelling. Then, we provide an overview about the most relevant techniques of *deep generative modelling* with focus on VAEs. The theory in this chapter is necessary to understand the modelling choices in the next chapter.

2.1. Single-cell generative modelling

Single-cell generative modelling is inspired by the success of Deep Generative Models (DGMs) in classical computer vision applications. As the name suggest, DGMs combine generative modelling with deep learning. The generative modelling part is responsible for capturing the underlying distribution of the data points which is useful for OOD prediction. However, without extending generative models to the DL domain they would not be suited to capture distributions of large data quantities, like it is the case for single-cell data. Therefore, do DGMs extend generative models with neural networks (NNs). The two DGM architectures that have proven most successful in the computer vision (CV) domain are: Generative Adversarial Networks (GAN) and Variational Autoencoders (VAE). Despite the success of GANs in CV [19] their adaption to the single-cell biology domain is more challenging than expected. The first challenge is that GANs are usually less interpretable than VAEs, because the latent vector in GANs is is not learned but usually represents noise from a Gaussian [2]. Therefore, there is no immediate way of tracing back the representation of each observation in the latent (input noise) space. Compared to that, VAEs have a better interpretability because the latent space is forced to resemble a prior distribution (often isotropic Gaussian distribution) that enables disentanglement [27]. The second challenge is the incooperation of prior knowledge into the latent space. Again, this is easier for VAEs because of their latent space learning instead of the noise structure in GANs. Finally, it is more difficult to provide a metrics of how “realistic” a generated data point is. Therefore, it might be harder to justify the use of GANs in many other biological applications. Hence, VAEs are the most suitable DGM for single-cell modelling.

2.2. Deep Generative models

Generative modelling is a unsupervised learning approach that explicitly models the underlying joint distribution $p(x, y)$ between observed x and unobserved data points y. Learning the true distribution of the data points x requires maximizing the marginal likelihood

$$\max \log p(x) = \max \log \int p(x, z) d\theta = \max \log \int p(x \mid z) p(z) dz \tag{2.1}$$

Equation 2.1 shows that computing the maximum marginal of data distribution x requires integrating over the likelihood $p(x \mid z)$ and prior $p(z)$. This is problematic because calculating the concrete interval is often intractable. Another technique, Variational Inference (VI) [1] avoids the computation of the

intractable posterior by introducing a variational posterior q that resembles the true posterior p as closely as possible. Thus, instead of relying on an analytical evaluation VI turns the computation of integrals into an optimisation problem.

Variational Autoencoder

A remaining problem is that VI in generative models is not efficient enough to high-dimensional data distributions. Therefore, Kingma and Welling proposed the Variational Autoencoder (VAE) as a way of performing inference in a efficient way. The VAE overcomes the problem of estimating the log-likelihood and posterior distribution by using neural networks (NNs) and stochastic gradient descent (SGD) [14]. More specifically, a VAE consists of two NNs: an Encoder and a Decoder. The encoder is denoted by $q_\Phi(z \mid x)$. $q_\Phi(z \mid x)$ is also referred to as variational posterior. The variational posterior is used as an approximation of the true posterior when calculating true posterior is intractable. As shown in Figure 2.1, the encoder receives data x and outputs parameters μ_z and σ_z that estimate the latent space z. The latent space z is learned to resemble a prior distribution $p(z)$. The prior distribution is chosen depending on the modelling purpose but a common choice is a isotropic Gaussian $\mathcal{N}(0, \mathcal{I})$ because it can best learn a representation such that features are independent of each other. In case of a gaussian variational approximationg the encoder learns the mean μ_z to be close to zero and variance σ_z to a diagonal one-variance matrix. The second NN component is the decoder $p_\Theta(x^{(i)} \mid z)$. Its input is a sampled latent representation z and its output is a reconstruction of the input data x'. The goal of the decoder is to effectively reconstruct the input x using the log-likelihood $p_\Phi(x \mid z)$ of the data given a sample from the variational posterior.

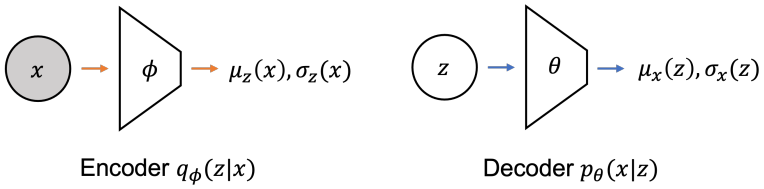


Figure 2.1: **Computational Schematics of Variational Autoencoder (VAE).** Grey colored nodes represent observed inputs. The encoder receives a input x to compute the mean $\mu_z(x)$ and variance $\sigma_z(x)$ describing the variational posterior distribution $q_\Phi(z \mid x)$. Then, the decoder computes the mean $\mu_x(z)$ and variance $\sigma_x(z)$ for the likelihood distribution $p_\Theta(x \mid z)$ given some sampled latent space z. Inspired from Figure 3 in [35].

Loss function

The maximization of the log-likehood (Equation 2.1) can be rewritten into a combination of the *variational lower bound* on the marginal likelihood of the data x and the Kullback Leibler (KL) Divergence [16] between the approximate and true posterior.

$$\log p_\theta(x^{(i)}) = \mathcal{L}(\Theta, \Phi; x^{(i)}) + D_{KL}(q_\Phi(z \mid x^{(i)}) \mid\mid p_\Theta(x^{(i)} \mid z)) \tag{2.2}$$

The first term in the equation is the *variational lower bound* and the second term the KL divergence. Because the KL divergence is non-negative the goal is to maximize the lower bound w.r.t. to the log-likelihood. Moreover, the *variational lower bound* from Equation 2.3 can be rewritten w.r.t to the variational Φ and generative parameters Θ:

$$\begin{aligned} \mathcal{L}(\Theta, \Phi; x^{(i)}) &= \mathbb{E}_{q_\phi(z|x)}\left[\log q_\phi(z \mid x) + \log p_\theta(x, z)\right] \\ &= -D_{KL}(q_\Phi(z \mid x^{(i)}) \mid\mid p_\Theta(z)) + \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)} \mid z)\right] \end{aligned} \tag{2.3}$$

Note that when phrasing Equation 2.3 as a loss function for VAE training the aim is to minimize the negative lower bound:

$$-\mathcal{L}(\Theta, \Phi; x^{(i)}) = D_{KL}(q_\Phi(z \mid x^{(i)}) \mid\mid p_\Theta(z)) - \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)} \mid z)\right] \tag{2.4}$$

The first term of the loss functions, the KL Divergence, encourages robustness to small perturbations along the latent manifold by matching a learned approximation from the encoder $q_\Phi(z \mid x^{(i)}$ to

some chosen prior $p_\Theta(z)$. Choosing a conjugate prior over z makes the integration of the loss function tractable. A common choice for the prior is the Isotropic Gaussian distribution $\mathcal{N}(0, \mathcal{I})$ with zero mean and a diagonal one-variance matrix. The advantage of this is that it forces independence across features in the latent space [27].

The second term of Equation 2.4 is the reconstruction loss (RL) and describes how accurately the output replicates the input. A common metric for this in the single-cell domain is the distribution (Zero Inflated) Negative Binomial ((ZI)NB) or Poisson distribution [33].

Conditional Variational Autoencoder

The main contribution of the VAE is to learn a meaningful representation of the low-dimensional space. Although this latent representation can be used to generate feasible samples, the classic VAE framework does not provide control on the output to be generated [3]. The Conditional Variational Autoencoder (CVAE) [32] modifies the VAE to address this limitation. Figure 2.2 shows that the CVAE model receives information about the inputs data condition c in addition to the input data x. Adding extra conditional information to the encoder and decoder forces the modelled distributions to be conditioned. This provides the model the capability to learn one-to-many mappings.

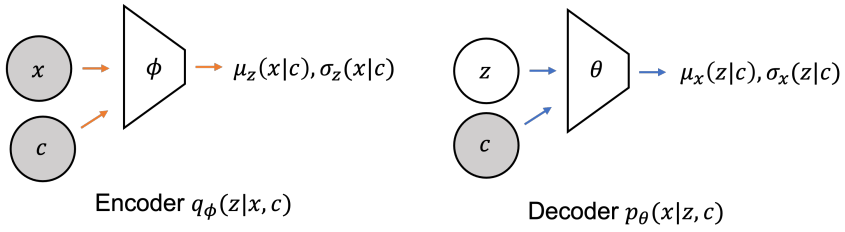


Figure 2.2: **Computational Schematics of Conditional Variational Autoencoder (CVAE).** Grey colored nodes represent observed inputs. The encoder receives a input x and covariate c to compute the mean $\mu_z(x \mid c)$ and variance $\sigma_z(x \mid c)$ describing the variational posterior distribution $q_\Phi(z \mid x, c)$. Then, the decoder computes the mean $\mu_x(z \mid c)$ and variance $\sigma_x(z \mid c)$ for the likelihood distribution $p_\Theta(x \mid z, c)$ given some sampled latent space z and covariate c. Inspired from Figure 3 in [35].

The objective function of the CVAE is the same as VAEs objective function (Equation 2.4) with additional conditional information c:

$$D_{KL}(q_\Phi(z \mid x^{(i)}, c^{(i)}) \mid\mid p_\Theta(z \mid c^{(i)})) - \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)} \mid z, c^{(i)})\right] \tag{2.5}$$

Multi-modal VAEs

The classical VAE frameworks are limited to reconstruction of one data modality x from the latent space z, as shown in Figure 2.3a. Extending the VAE to infer a joint latent space representation of all input modalities would allow the model to reconstruct i data modalities $x_1, ...x_i$ (Figure 2.3b) [35].

(a) Variational Autoencoder

(b) Multi-modality VAE

Figure 2.3: **Latent variable model representation of single- and multi-modality VAE.** Latent variable model of a (a) single-modality VAE showing the relation between the observed variable, x, and the unobserved, latent variable, z and b) multi-modality VAE where the latent space z contains information about multiple i data outputs $x_1, ..., x_i$.

Figure 2.4 shows two different approaches that can be used for multi-modality modelling in VAEs. The architecture in Figure 2.4a is called joint multi-modal VAE. In this model the encoder gets the multi-modality information as a concatenation. This approach is for example employed in totalVI [7]. A common limitation of representing the different modalities with a single posterior is that it might lead

to unwilling masking of information from one modality due to the dominance of another modality [28]. To overcome this problem Shi et al. propose a Mixture-of-Experts Variational Autoencoder (MMVAE) in which each modality is modelled by a seperate VAE pair. Each encoder estimates the modality specific parameters for the MoE posterior distribution independently and then mixes them using mixture-of-experts (MoE). Figure 2.4b shows the basic architecture of the MMVAE model which is also used by scMM [28].

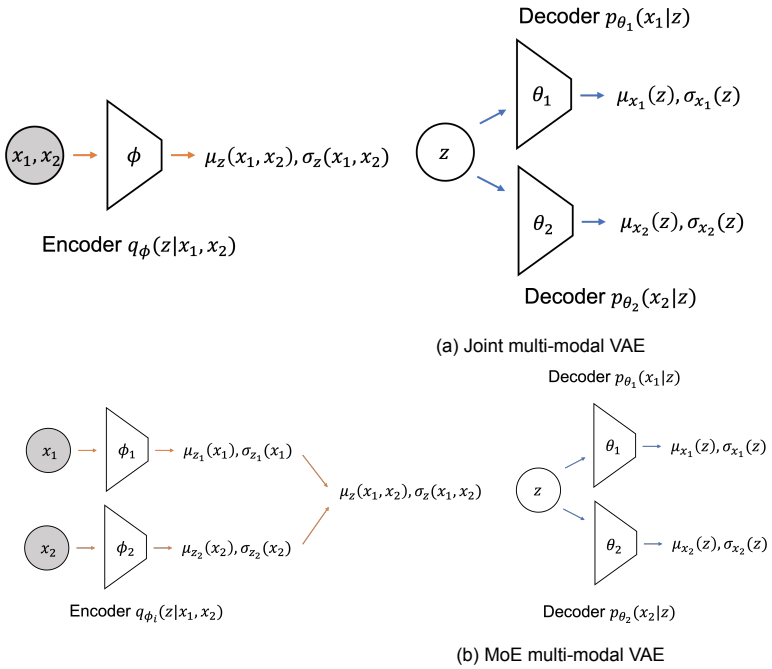


Figure 2.4: **Computational Schematics of two multi-modality VAE (MMVAE) frameworks.** Grey colored nodes represent observed inputs. Both MMVAE frameworks are illustrated for two-modality data but can be extended to more than two modalities. Both inspired from Figure 3 in [35]. a) Joint multi-modal VAE, this framework is used by totalVI [7]; The encoder receives the concatinated two-modality input x_1, x_2 to compute the mean $\mu_z(x_1, x_2)$ and variance $\sigma_z(x_1, x_2)$ describing the variational posterior distribution $q_\Phi(z \mid x_1, x_2)$. Then, each modalities decoder computes the mean $\mu_{x_i}(z)$ and variance $\sigma_{x_i}(z)$ for the modality i likelihood distribution $p_\Theta(x_i \mid z)$ given some sampled latent space z. b) MoE multi-modal VAE, framework introduced by [31] and used by scMM [28]. The difference to a) is that the input x_1 and x_2 is not concatinated because each modality has a seperate VAE pair. Each modality encoder learns the modality specific mean $\mu_{z_i}(x_i)$ and variance $\sigma_{z_i}(x_i)$ before joining the posterior with MoE. The generative process is equivalent to (a).

Mixture-of-experts multimodal VAE

The MoE multimodal VAE (MMVAE) by Shi et al. aims to learn a multi-modal generative model

$$p_\Phi(z, x_{1:M}) = p(z) \prod_m^M p_{\theta_m}(x_m \mid z) \tag{2.6}$$

with $m = 1, ..., M$ modalities, $p(z)$ prior and $p_{\theta_m}(x_m \mid z)$ likelihood of each mth modality. Figure 2.4b shows an example of a MMVAE for $M = 2$ with a encoder-decoder pairs for each modality. The encoder parameterizes the variational posterior $q_{\phi_m}(z \mid x_m)$ and the decoder the likelihood $p_{\theta_m}(x_m \mid z)$ of the m-th modality. To jointly learn the variational posterior across modalities the variational posterior is factorized with a MoE

$$p_\Phi(z \mid x_{1:M}) = \sum_m^M \alpha_m q_\phi(z|x_m) \tag{2.7}$$

with $\alpha_m = \frac{1}{M}$.

Equivalently as to the VAE, the training objective of the MMVAE is to maximize the marginal likelihood $p(x_{1:M})$ through optimization of the ELBO

$$\mathcal{L}_{ELBO}(\mathrm{x}_{1:M}) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{z_m \sim q_{\phi_m}(\mathrm{z}|\mathrm{x}_m)}\left[\log \frac{p_\Theta(\mathrm{z}, \mathrm{x}_{1:M})}{q_\Phi(\mathrm{z} \mid \mathrm{x}_{1:M})}\right] \tag{2.8}$$

$$= \frac{1}{M}\sum_{m=1}^{M} \left\{\mathbb{E}_{z_m \sim q_{\phi_m}(\mathrm{z}|\mathrm{x}_m)}\left[\log p_{\Phi(x_{1:M}|\mathrm{z}_m)}\right] - KL\left[\phi_m(\mathrm{z} \mid \mathrm{x}_m \mid\mid p(\mathrm{z})\right]\right\} \tag{2.9}$$

Like in the VAE loss function (Equation 2.4) the first term refers to the reconstruction loss and the second term enforces the regularization of the variation posterior through the prior. Differently to VAE, the prior $p(\mathrm{z})$ is defined as a Laplacian distribution because it has a heavier tail and thus, can fit outliers better than the isotropic Gaussian prior [27]. The mean of the Laplacian prior is set to zero and the scaling constrained that is learn from the data through SGD. Note that the authors Shi et al. of the MMVAE are proposing more advanced loss functions, such as the doubly reparameterized gradient estimator [34], but in this report we are focusing on Equation 2.8 as has been used in the scMM model.

The main advantage of MMVAE is that the MoE variational posterior learns a multi-modal generative model that satisfies the following four criteria:

1. Latent Factorization: Latent space captures shared and private modality information.
2. Coherent Joint Generation: Generation of different modalities is possible such that they are coherent in shared information.
3. Coherent Cross Generation: Generation of data for modalities different to the input.
4. Synergy: Quality of the generative model improves when trained over multiple modalities rather than just a single modality.

These four criteria, especially latent factorization and synergy, are helpful for the perturbation prediction tasks for this work.

3

Models

The previous chapter summarized the relevant background information to understand the models employed in this work. More specifically, we use scVI [21], as a single-modality model and totalVI and cellPMVI for multi-modality modeling. Both scVI and totalVI, are initially designed for integration rather than the prediction of perturbation effects. Therefore, we adapted them to accommodate perturbation prediction. After that, we present our model cellPMVI. cellPMVI is a multi-modality model using variational inference designed explicitly to predict perturbation effects for single-cell data.

3.1. Common modelling features

All models follow the CVAE framework (see Section 2.2). First, every model infers the variational posterior $p(z \mid D, c)$ and then learns the likelihood distribution $p(D \mid z, c)$ where z represents the sampled latent space. The input data $D = \{x_{ng}, y_{nt}\}$ contains the count data for RNA x and protein y modality for each cell n across all genes g or proteins t. Furthermore, the model receives the categorical covariate information $c_n = \{c_{n1}, ..., c_{ni}\}$ for i different covariates. The categorical covariates provide information about the cell being used as input, i.e., cell type, population group, or perturbation. All models estimate the likelihood distribution of the RNA data $p(x_n \mid z_n, c_n, l_n)$ with a negative binomial distribution $NB(\mu, \theta)$ where θ defines the probability of success or failure and μ decides whether θ is a success or failure. Previous research showed that the negative binomial distribution could model over-dispersion and handle the limited sensitivity of gene expression data the best [33]. Note that the RNA likelihood data in each model is influenced by an RNA size modeling factor ℓ_n, also called library size. The library size represents the sum of amplified mRNA molecules per cell. Initially, the RNA library size in scVI and totalVI was modeled as a latent factor that is sampled from a LogNormal distribution $\mathcal{N}(\mu, \sigma^2)$. However, in the most recent implementation, the library size is treated as observed and set to the total Unique Molecular Identifier (UMI) count of RNA. The following sections introduce the technical details of the single- and multi-modality model.

3.2. Single-Modality Model

The single-modality model only receives information from one modality, the gene expression x.

Single-cell Variational Inference (scVI) model

scVI [21] defines a fully probabilistic approach developed for the normalization and analysis of scRNA-seq data. The model takes as input raw count data x_{ng} with n cells and g genes and categorical covariates c_n.

Figure 3.1 shows the neural network architecture of scVI and Equation 3.1 a simplified version of the inference process. scVI uses a VAE framework by first learning the variational posterior $p(z_n, \log l \mid x_n, c_n)$ and then generative model of the scRNA-seq data $p(x_n \mid z_n, c_n, l_n)$. During the inference process, the variational posterior is approximated by two modeling components: 1) RNA size factor l and 2) latent space z. Both modeling components consist of an encoder network that receives the gene expression counts x_{ng} and categorical covariates c_n. First, the RNA size factor l, also referred to as

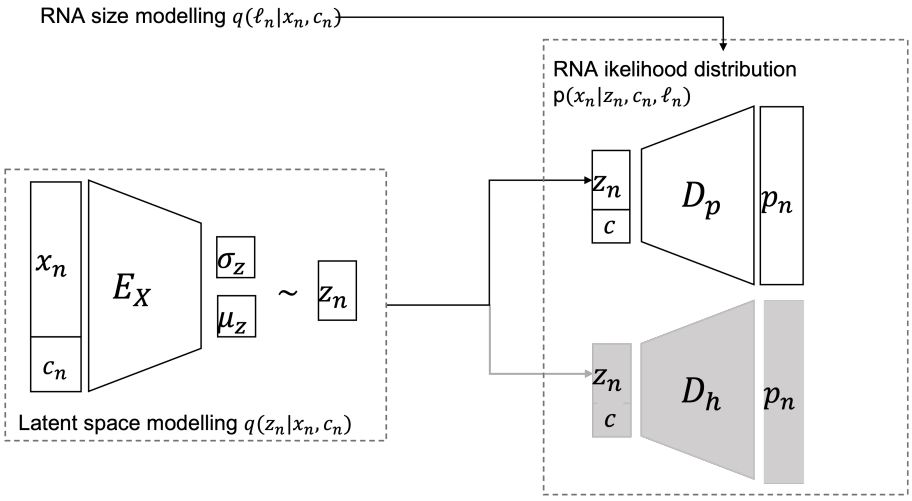


Figure 3.1: **Schematic of neural network architecture of scVI [21].** scVI learns the parameters for the RNA likelihood modelling. The encoder learns a cell specific latent space representation z_n for the gene expression data x_n and categorical covariates c_n. After that the sampled latent space z_n, together with the RNA size modelling factor and categorical covariates, are fit to the decoder D_p to learn the parameters p_n to model the RNA likelihood distribution.

the nuisance factor, is modeled as an observed factor depending on the RNA data x_n and categorical covariates c_n. Next, the encoder network E_z learns a lower-dimensional latent space z_n for each cell n. The inference process can be summarized by:

$$\begin{aligned} \mu_z, \sigma_z &= f^{E_z}(x_{ng}, c_n) \\ z_n &\sim N(\mu_z, \sigma_z) \\ \ell &\sim q(\ell_n \mid x_n, c_n) \end{aligned} \tag{3.1}$$

Both the prior and posterior distributions follow a logistic normal distribution $\mathcal{N}(0, 1)$ with zero mean and standard deviation 1. Next, the generative process uses the RNA size factor and latent space to approximate the parameters of the likelihood distribution $p(x_n \mid z_n, c_n, l_n)$. As mentioned previously, we use a negative binomial distribution for the likelihood modeling instead of the zero-inflated negative binomial (ZINB) used in the original work [21] because current research shows that the ZINB distribution over-represents zero counts [33]. The generative process is summarized in Equation 3.2. First two decoder networks, the first network (D_{px}) approximates the expected frequency p_n and the second network (D_h) models the dropout π_n in each cell separately. The second network D_h is not necessary for the NB likelihood modeling. Therefore, D_h is colored in grey in Figure 3.1 and not included in equation 3.2. The generative process can be summarized as follows:

$$\begin{aligned} p_n &= f^{D_p}(z_n, c_n) \\ \hat{x}_n &\sim NB(\ell_n p_n, \theta_g) \end{aligned} \tag{3.2}$$

with θ denoting a gene-specific inverse dispersion factor.

3.3. Multi-Modality Models

The multi-modality models we use for predicting immune responses on CITE-seq data are cellPMVI and totalVI. TotalVI [7] is an extension of the scVI model to analyze and integrate CITE-seq data. This work uses an adaption of totalVI to benchmark cellPMVI. cellPMVI extends scVI to the multi-modality domain by mixture-of-experts posterior integration. The main difference between the totalVI and cellPMVI is their technique of representing the modalities in the posterior. TotalVI uses a single-encoder for both modalities and jointly models them in the posterior space. On the other hand, cellPMVI encodes each modality with a separate encoder-decoder pair. In that way, cellPMVI first represents the modalities separately in the latent space and then performs MoE posterior integration for a joint

posterior representation [31]. In summary, both models consider that RNA and protein measurements are generated from the same latent space of cells. However, totalVI assumes that RNA and protein measurements have the same latent space characteristics while cellPMVI models shared and individual latent space features for RNA and protein measurements.

TotalVI

TotalVI uses a probabilistic approach to learn a joint representation of paired transcriptome and protein measurements in single cells. Figure 3.2 shows the neural network architecture of totalVI.

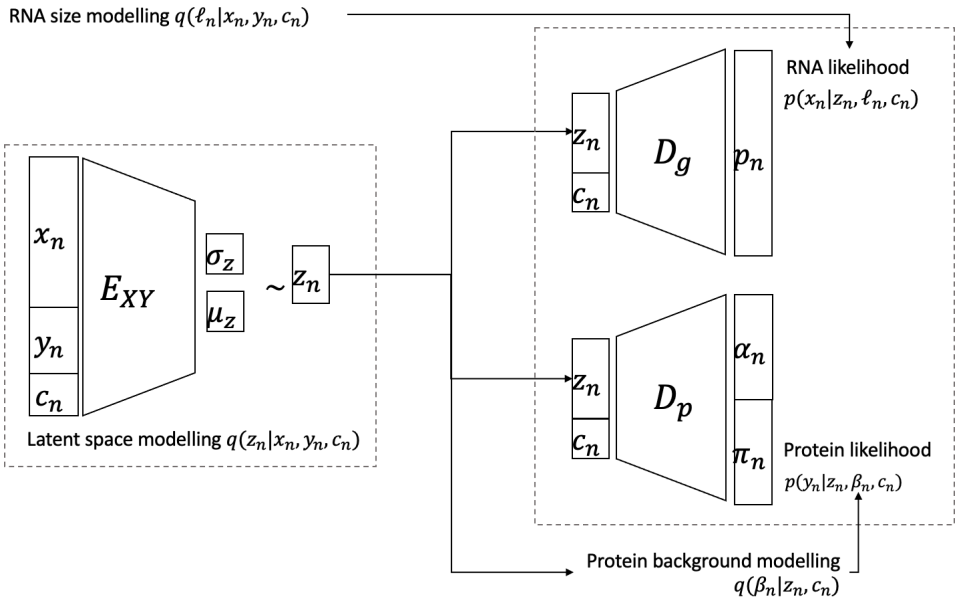


Figure 3.2: **Schematic of neural network architecture of totalVI [7]** totalVI learns the parameters for the RNA and protein likelihood modelling. The encoder learns a cell specific joint latent space representation z_n over the gene x_n and protein expression data y_n and categorical covariates c_n. After that the gene decoder learns the RNA likelihood distribution and the protein decoder the protein likelihood distribution. Each decoder receives the latent space z_n and categorical covariates c_N as input. However, the gene decoder models the RNA likehood together with the RNA size modelling factor ℓ_n and the protein decoder in combination with the protein background β.

The input to totalVI is a concatenation of RNA x_n, protein unique molecular identifies (UMI) counts y_n matrices and categorical covariates c_n. First TotalVI infers a low-dimensional cell representation $q(z_n \mid x_n, y_n, c_n)$ and the RNA size factor $q(l_n \mid x_n, y_n, c_n)$ (Equation 3.3).

$$\begin{aligned} \mu_z, \sigma_z &= f^{E_z}(x_n, y_n, c_n) \\ z_n &\sim N(\mu_z, \sigma_z) \\ \ell &\sim q(\ell_n \mid x_n, c_n) \end{aligned} \tag{3.3}$$

The generative process of TotalVI consists of learning the protein background distribution $q(\beta_n \mid y_n, c_n)$ and likelihood distribution (Equation 3.4). TotalVI learns separate likelihood distributions for each gene g ($p(x_n \mid z_n, l_n, c_n)$ and each protein t ($p(y_n \mid z_n, \beta_n, c_n)$), modelled by a negative binomial and negative binomial mixture respectively.

$$\begin{aligned} \beta_n &\sim q(\beta_n \mid y_n, c_n) \\ \alpha_n, \pi_n &= f^{D_t}(z_n, c_n) \\ v_n &\sim Bernoulli(\pi_n) \\ \hat{x}_n &\sim NB(\ell_n p_n, \theta_g) \\ \hat{y}_n &\sim NB(v_n, \beta_n + (1 - v_n)\beta_n \alpha_n, \phi_t) \end{aligned} \tag{3.4}$$

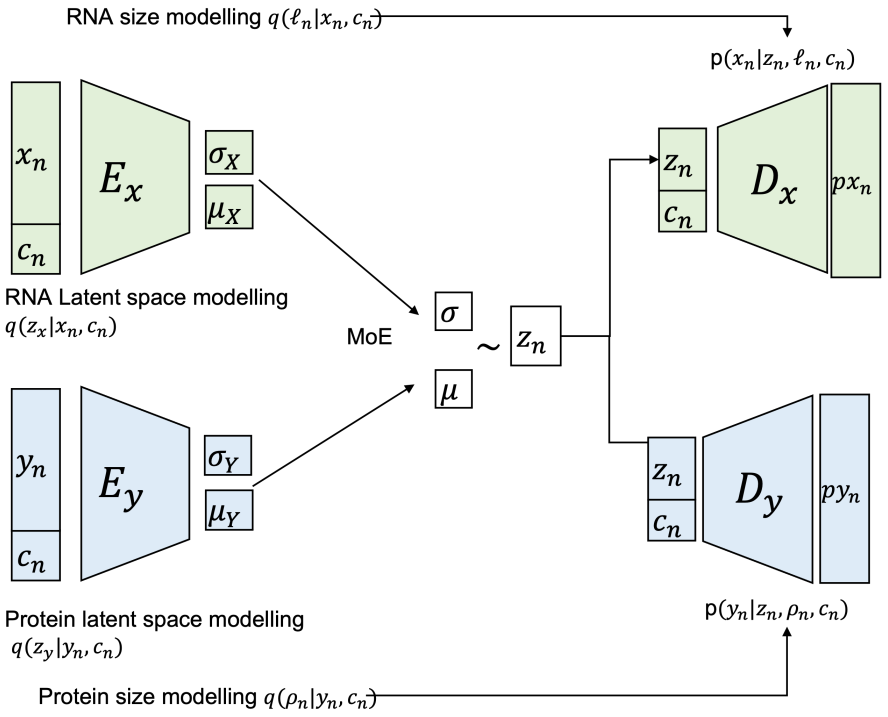


Figure 3.3: **Schematic of neural network architecture of cellPMVI**. cellPMVI learns the parameters for the RNA and protein likelihood modelling with separate encoder-decoder pairs and a mixture-of-experts latent space joining. The RNA encoder-decoder pair (E_X and D_X) is colored in green and models the RNA likelihood together with the RNA library size factor ℓ. The protein encoder-decoder pair (E_Y and D_Y) is colored in blue learning the protein likelihood distribution together with the protein size factor p.

cellPMVI

cellPMVI is developed for predicting perturbations using multi-modality data, specifically CITE-seq data. The underlying idea for cellPMVI is to allow accurate representation of shared and private features of the paired transcriptome and protein measurements. That means cellPMVI should overcome the assumption of totalVI that both modalities can be reduced to a common latent feature space.

A generative modeling approach to learning shared and private information is the MMVAE (see Section 2.2). The principle of MMVAE in the single-cell domain has been proven successful for the cross-modality prediction of CITE-seq data in scMM [28]. In this work, we use the MoE integration from MMVAE with the probabilistic modeling of CITE-seq data.

Figure 3.3 shows the architecture of cellPMVI. cellPMVI consists of two VAE pairs, one for each modality. Each VAE pair is equivalent to a scVI component (Figure 3.1) without the dropout deocer D_h. As explained in section 3.2 the scVI component consists of an encoder-decoder pair modeling the latent space z and a scaling factor ℓ, also called library size. cellPMVI uses a scaling factor for both RNA and protein modality because Lopez et al. showed that the library size had a significant contribution to the superior performance of scVI. For the RNA expression, the scaling factor again represents the scaling of the RNA expression ℓ while the scaling factor for the protein represents the protein scaling p. During the model development, we also considered modeling the protein background (with the totalVI decoder), but the results showed that protein scaling performed better. To summarize, the VAE pair for the RNA expression (E_g and D_g) is connected with an RNA size modeling component, equivalent to the component in scVI. The protein expression VAE pair (E_p and D_p) is connected to a protein size modeling component inspired by the scVI RNA size modeling component instead of the protein background modeling component in totalVI.

The encoder of each modality component learns a latent space approximation of the given modality. That means, the RNA encoder E_x approximates the posterior for the RNA $q(z_n \mid x_n, c_n)$ and the protein encoder E_y for the protein expression $q(z_n \mid y_n, c_n)$. Then the MoE integration is used to approximate a common latent space integrating the shared and single characteristics, by

$$q_\phi(z_n \mid x_n, y_n, c_n) = \frac{1}{2}(q(z_x \mid x_n, c_n) + q(z_y \mid y_n, c_n)) \tag{3.5}$$

In cellPMVI both the Laplace and isotropic normal gaussian have been tried for the prior and posterior distribution. Laplace has a heavier tail and is hence expected to fit outliers better. However, our results showed that a isotropic normal gaussian prior slightly outperformed the Laplace prior. Therefore, cellPMVI is used with a isotropic Gaussian for the experiment in this work. Besides that, the RNA $q(\ell \mid x_n, c_n)$ and protein size modelling factors $q(p \mid x_n, c_n)$ are learned during the inference process.

$$\begin{aligned}
\mu_x, \sigma_x &= f^{E_x}(x_n, c_n) \\
\mu_y, \sigma_y &= f^{E_y}(y_n, c_n) \\
\mu, \sigma &= MoE(q(z_x \mid x_n, c_n), q(z_y \mid y_n, c_n)) \\
z_n &\sim N(\mu, \sigma) \\
\ell &\sim q(\ell_n \mid x_n, c_n) \\
p &\sim q(p_n \mid y_n, c_n)
\end{aligned} \tag{3.6}$$

Lastly, during the inference process the MoE approximated latent space is used to learn both the RNA and protein decoder for reconstruction. cellPMVI uses a negative binomial distribution for both the RNA and protein likelihood distribution.

$$\begin{aligned}
px_n &= f^{D_x}(z_n, c_n) \\
py_n &= f^{D_y}(z_n, c_n) \\
\hat{x}_n &\sim NB(\ell_n p_n, \theta_g) \\
\hat{y}_n &\sim NB(\ell_n p_n, \theta_t)
\end{aligned} \tag{3.7}$$

with θ denoting a gene- g or protein t specific inverse dispersion factor. Note that, the MoE integration makes it possible to optimize a common ELBO value (Equation 2.8) for cellPMVI instead of an ELBO value per modality.

4

Experimental setup

The previous chapter explained the architecture of the single- and multi-modalities models we use for perturbation prediction. In this chapter, we first introduce the general pipeline for our approach, after which we go into more detail about each of the four steps that make up the pipeline.

4.1. General Structure

Figure 4.1 illustrates the general structure for immune response prediction. We execute the following steps:

1. **Data**: First, we preprocess the data.
2. **Training**: Then the preprocessed data is subsetted according to one of the three training scenarios. After data subsetting, one of the models (see Section 3) is trained for 400 epochs.
3. **Reconstruction**: The trained model is used to sample reconstructions of the gene expression X' and for multi-modality models of the protein expression count Y'. There are three different sampling strategies that we are using: a) posterior, b) prior and c) transfer predictive sampling. Each sample n_{sample} represents the reconstructed expression of one cell.
4. **Evaluation**: Lastly, we evaluate the sampled gene or protein expressions of the models. The evaluation mainly focuses on i) the ability of the model to fit CITE-seq data, ii) how well the models can reconstruct and predict gene expressions , i.e. evaluating if there is an added benefit of multi-modality information instead of single-modality information, and iii) comparing multi-modality models on their performance for protein reconstruction.

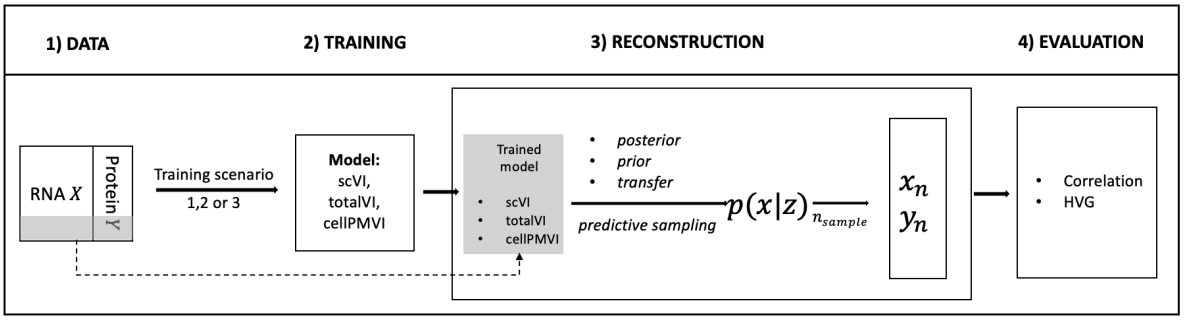


Figure 4.1: **Schematic of the immune response modelling process.** First, the data is pre-processed and then split into training and test set according one of the training scenarios. Then one of the three models, scVI, totalVI or cellPMVI, is trained. Next, a predictive sampling scenario is used to learn the RNA or protein likelihood to retrieve n samples for RNA x_n or protein expressions y_n. Lastly, the expressions are evaluated using correlation and highly variable gene count measurements.

4.2. Dataset and preprocessing

This work requires a multi-modality dataset measuring immune responses for different adjuvant conditions across populations. Therefore, this work uses CITE-seq data containing measurements for transcriptomics and protein across each cell. The CITE-seq dataset is unpublished at the moment of writing. In total the *czi* dataset consists of 26 616 cells, 143 proteins and 11700 genes and label annotations about the e.i. cell type or population origin of the cell. The supplementary methods A.1 provide a more detailed description of the single-cell sequencing analysis and annotation process of the *czi* data. Every cell is annotated for:

- **Population groups**:
  1. DK: Dakar (urban Senegal)
  2. RT: Richard-Toll (rural Senegal)
  3. LD: Leiden (urban European)
- **Perturbation**:
  A) unperturbed: medium
  B) perturbed: PMA/Ionomycin (abbreviation: PI)
- **Individuals**: each population group has two indiduals (DK = DK06,DK68; LD = LD254, LD276; RT = RT55, RT162)
- **Cell type**: B, CD4T, CD8T, DC, Monocyte, NK, non-conventional T cells (abbreviation: OtherT) , Platelet

See Supplementary Table B.2 for the exact number of cells available for each category of annotation. Each annotation category can be used as a covariate c during training. However, in this work, we focus on exploring the conditioning of the model using the population groups, perturbation, and cell types as covariates. We do not condition on individuals because of the limited amount of data.

Preprocessing

Before training the model, the data is preprocessed (Figure 4.1). The preprocessing ensures the removal of lowly expressed genes and other outliers. This preprocessing step helps the model identify important features without basing its prediction ability on outliers. the functions used for preprocessing are from the scanpy package.

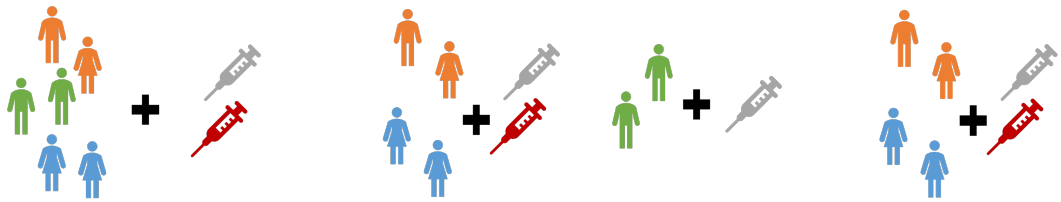
The preprocessing consists of three steps:

1. First, all low count genes are filtered out (all genes with a count lower than 3) with the `filter_genes` function.
2. All top 5000 highly variable genes are selected with the `highly_variable_genes` function. Selecting highly variable genes allows finding the genes that contribute the most to cell-to-cell variation. Highly-variable gene selection is essential for the model to more easily capture variation and define the features that will be the most variable between the perturbation conditions.
3. Lastly, in case any gene markers that belong to a protein marker were removed are added back into the data.

4.3. Training

The model training proceeds according to one of the three different training scenarios in Figure 4.2. A training scenario defines the subset of the data used for training and relates to a prediction task. The first training scenario excludes no data subset (Figure 4.2a). That means, the model has seen samples of each covariate condition information during training. Training scenario 1 can evaluate the model's capacity to learn the data's underlying structure. In this work, we also use training scenario 1 to explore under which training conditions a models performance might improve. For example, how does the combination of covariates influence the model performance? For the second training scenario, the

cells of one perturbed population group are withheld (Figure 4.2b). Training scenario 2 provides the basis for OOD scenario 1 (Figure 1.2a), extrapolation to perturbed conditions. More specifically, training scenario 2 investigates the model's ability to generalize perturbation responses across populations. Lastly, the third training scenario is used for OOD scenario 2 (Figure 1.2b), exploring extrapolation to new population groups (Figure 4.2c). For this scenario 3, the data points of one entire population group are left out during training. Hence, during reconstruction, the model must generalize to an unknown population group given the responses from other population groups.



(a) Training Scenario 1 (b) Training Scenario 2 (c) Training scenario 3

Figure 4.2: **Three different training scenarios used for the experiments.** The dataset consists of three population groups with each two individuals and two perturbation adjuvants (grey and red). Each training scenario includes a selection of the data set. (a) Training scenario 1 includes responses from all individuals to every adjuvant. (b) In training scenario 2 the individual responses of one population group (in this case population group 3 to the perturbed adjuvant B) are excluded. (c) In training scenario 3 all individuals of one population group are excluded.

As mentioned previously, because of the limited amount of data, we use population groups rather than individuals for the covariate conditioning of the models. By that, we assume that variation across population groups is higher than variation across individuals.

Single-Cell Variational Inference Toolbox

The single-cell variational inference toolbox (scvi-tools) is a package for probabilistic modelling of single-cell omics data Gayoso et al. scvi-tools provide a collection of models (e.g. scVI [21] and totalVI [7]) with the same interface and base component structure for probabilistic model development. The standard interface across models allows for reuse of model components, and further development of downstream tasks is possible. In this work, we modified the scVI and totalVI model implementation and built the cellPMVI model using the scvi-tools framework.

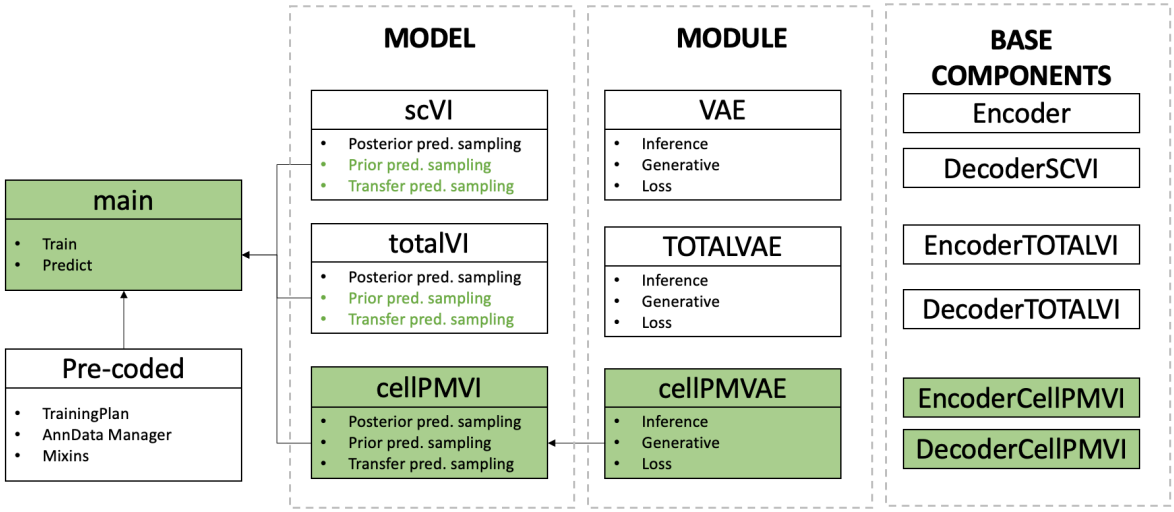


Figure 4.3: **Abstract structure of scvi-tools used in this work.** The implementation follows the scvi-tools structure with models, modules and base components. The boxes represent classes and the items in the boxes' functionalities. The green components or functions are extensions implemented in the context of this work, and the others are reused from scvi-tools. The illustration is inspired by Figure 4 in [6].

Figure 4.3 shows the scvi-tools components used in this work, where the green-colored parts refer to new implementation components. The three elements from scvi-tools used for our implementations are

the i) `model`, ii) `module` and `base components`. The `base components` are the lowest level structures, defining the neural network elements, including their forward passes e.i. encoder and decoder networks. Next, the `module` component implements the variational inference, defining the inference (encoder) and generative process (decoder). The highest-level elements are the `model` classes, defining actions on the lower-level components such as training, subsequent analysis, and evaluation steps on the trained model. In this work, we define three different models:

1. **scVI**: The scVI model uses the `VAE` module (`scvi.module.vae`) with the `Encoder` and `DecoderSCVI` from the base components class. In the model class we implement the prior and transfer predictive sampling procedure and reuse the already existing posterior predictive sampling procedure.
2. **TotalVI**: Similarly, to scVI, totalVI implements the `TOTALVAE` module which integrates the `EncoderTotalVI` and `DecoderTotalVI`. Again, the posterior predictive sampling procedure for TotalVI is already defined, but the prior and transfer predictive sampling are newly implemented.
3. **cellPMVI**: The cellPMVI model is a new component that uses the uses the `cellPMVIVAE` module. In the `cellPMVIVAE` module the joint posterior integration with the `loss` function.

Lastly, all models can be trained using the multiple pre-coded lower level components such as TrainingPlan, AnnDataManager and Mixins. These predefined components allow for flexible training and adjustment of the model. Please refer to the paper from Gayoso et al. for more details about the scvi-tools package.

Training and model parameters

We used the following training parameters:

- Nr. epochs: 400
- Training set size: 80%
- Validation set size: 20%
- Nr. epochs between validation check: 20

Additionally we used the following model parameters:

- Nr. latent dimensions: 20
- Nr. hidden layers: 2
- Nr. nodes hidden layer: 128
- Dropout: 0.1
- Batch size: 128

4.4. Sampling reconstructions

The model aims to predict gene expression and protein counts of population groups to a given perturbation, more specifically adjuvant. One way to predict responses is by sampling a reconstruction x' from the likelihood $p_\Theta(x \mid z)$. This section, we will introduce three different techniques to sample reconstructions from the model: *posterior*, *prior* and *transfer predictive sampling*. After that, section 4.5 provides an overview of matrices used to evaluate the quality of the predicted responses.

Posterior predictive sampling

Posterior predictive sampling aims to reconstruct the input data x as closely as possible. Figure 4.4 shows the posterior predictive sampling procedure:

1. First, the input data x and corresponding conditions c are passed to the encoder.

2. The encoder returns the μ and σ of the input data specific posterior distribution $q_\Phi(z \mid x)$.
3. A sample from the latent space specific distribution is obtained: $z \sim N(\mu, \sigma)$.
4. The encoder receives the latent space sample z and conditions c to model the likelihood $p_\Theta(x \mid z)$.
5. n samples are obtained from the likelihood representing the reconstructed data X'.

Posterior predictive sampling is used to perform posterior predictive checks (PPCs). PPCs are used to validate the fit of a Bayesian model by comparing the sampled reconstruction to observed data [8] Note that PPCs do not provide insight into how much the model has learned to generalize population group responses because they have information from all the available data.

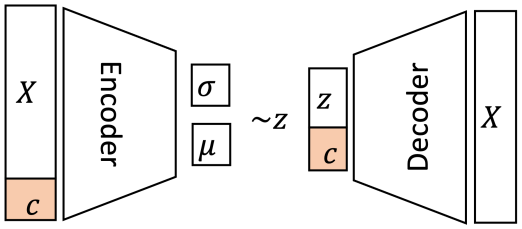


Figure 4.4: **Posterior predictive sampling.** Based on the data X and categorical covariate c specific latent space z, the decoder learns the likelihood distribution to reconstruct data X'.

Transfer predictive sampling

Ideally, the model should answer counterfactual questions like those introduced in Section 1.1. For example, we want to predict the perturbed response from a population given the unperturbed response. In that case, posterior predictive sampling is not applicable because it requires data input of the to be predicted response, which is not available. Instead a reconstruction can be sampled using *transfer predictive sampling* or *prior predictive sampling* (Section 4.4).

Figure 4.5 illustrates the flow for *transfer predictive sampling*. Note that the difference to posterior predictive sampling is that the condition labels c are adjusted when passed to the decoder. In Figure 4.5 this is illustrated through the color difference. Note that not all covariates in the condition need to be changed. In this work, the conditions are usually adjusted for the perturbation covariate: from unperturbed to perturbed. For example, given the unperturbed gene expression of a population 1, the model will predict the perturbed gene expression of the population 1.

Summarizing the *transfer predictive sampling* proceeds as follows:

1. First, the input data x and corresponding conditions c are passed to the encoder.
2. The encoder returns the μ and σ of the input data specific posterior distribution $q_\Phi(z \mid x)$.
3. A sample from the latent space specific distribution is obtained: $z \sim N(\mu, \sigma)$.
4. *The covariates of interest, e.g. perturbations, are adjusted in the conditions c resulting in c'.*
5. The encoder receives the latent space sample z and adjusted categorical covariates c' to model the likelihood $p_\Theta(x \mid z)$.
6. n samples are obtained from the likelihood representing the reconstructed data X'.

Note that the *transfer predictive sampling* only differs from the *posterior predictive sampling* in the additional step 4, where the conditions are adjusted.

Prior predictive sampling

As mentioned in the previous section, the *prior predictive sampling* offers another possibility to reconstruct gene expressions of previously unseen conditions. Besides that, sampling from the prior evaluates how suitable the prior is to represent the data set. As shown in Figure 4.6 this sampling procedure only requires the decoder network. That means, z is sampled from the prior distribution $p(z)$

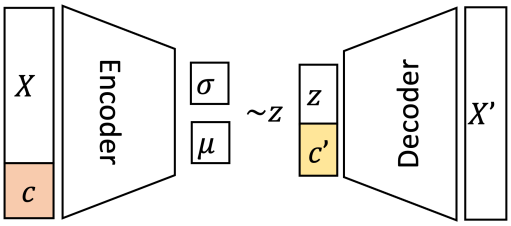


Figure 4.5: **Transfer predictive sampling.** The latent space is sampled given the data X and categorical covariate c. Then the categorical covariate gets adjusted c' to reproduce data x' which is different to the input data x.

instead of the variational posterior $q_\Phi(z \mid x)$. In the case of scVI and totalVI the prior distribution would be the isotropic normal Gaussian distribution and for MMVAE the Laplace distribution.

The steps for the prior predictive distribution are:

1. Sample from the prior distribution: $z \sim p(z)$ and set the RNA size factor l_n to a constant
2. The encoder receives the latent space sample z and categorical covariates c of interest to reconstruct to model the likelihood $p_\Theta(x \mid z)$.
3. n samples are obtained from the likelihood representing the reconstructed data X'.

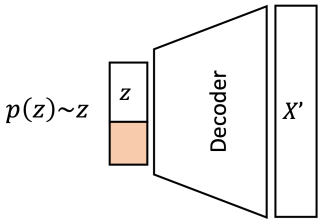


Figure 4.6: **Prior predictive sampling.** The decoder predicts data x from the latent space, sampled from the prior distribution $p(z)$, and categorical covariate information c.

4.5. Evaluation metrics

To understand how reliable the models predictions are, the models are evaluated on their loss values and quality of sampled reconstruction through correlation and highly variable gene count prediction.

Loss and KL divergence

The training and validation losses of the i) ELBO, ii) Reconstruction loss (RL) and iii) KL divergence (KLD). The validation loss is calculated on 20% of the data not included during training. We expect the ELBO and RL to decrease during model training showing that the model learns to represent the data better. For the KL divergence, is it common to first rise as the model can substantially improve its loss by reducing the ELBO and RL. Then, only after some epochs the model brings the posterior distribution closer to the prior distribution to reduce the KL divergence. Note that these values do not necessarily provide insight into the quality of the model's performance but are used to validate that the model is training without overfitting.

Correlation

To evaluate how well the predicted data fits the features of the original data we calculate the correlation. More specifically, the Spearman's rho correlation is calculated. The Spearman's rho is suitable because it accounts for non-linear relationships in the expression values. If the correlation is non-zero then that means that the genes are co-regulated. In case, the correlation is 1.0 the predicted data fits the original data most accurately.

Given the true x_{ng}^{true} and predicted x_{ng}^{pred} gene or protein expression profile with n number of cells and g number of genes or p number of proteins. We want to compare the gene and protein expression

for each gene or protein instead of cells because cell measurements are unpaired while genes are paired. Therefore, the mean and variance are calculated across all cells, resulting in two g long mean and variance arrays. After that we calculate the pairwise correlation between the true and predicted mean or variance arrays:

$$corr_{\mu}(x^{true}, x^{pred}) = corr\left(mean_n(x_{ng}^{true}), mean_n(x_{ng}^{pred}\right) \tag{4.1}$$

$$corr_{\sigma}(x^{true}, x^{pred}) = corr\left(var_n(x_{ng}^{true}), var_n(x_{ng}^{pred}\right) \tag{4.2}$$

For the protein correlation Equations 4.1 and 4.2 would be calculated for the proteins t instead of genes g.

Common highly variable gene count

Comparing the selection of highly variable genes (hvg) from the true and predicted gene expression counts provides insight about whether the model learns which cells have the most influence. The higher the number of common hvg the better the model captures which genes contribute most to the gene expression profile. For the comparison the 1000 most highly variable genes are selected from the true and predicted count matrix and the number of common hvg calculated, ignoring the hvg rank.

4.6. Implementation

The model was implemented using Python 3.9 with PyTorch and the scvi-tools package [6]. For data preparation and preprocessing the annotated dataset format [36] and scanpy package was used.

5

Results

The evaluation focuses on comparing the performance between single- and multi-modality models (**RQ 1**) as well as the complexity of generalization that is possible (**RQ 2**). To answer the second question (**RQ 2**) we perform experiments according to the three training scenarios (see Section 4.3), each evaluating the models for a different generalization complexity. We consider that training scenario 1 requires a minor generalization, and training scenario 3 requires the highest generalization. Furthermore, model performances between i) single- and multi-modality models and ii) multi-modality models (totalVI and cellPMVI) are compared. All experiments with scVI only include the gene data, while experiments with totalVI and cellPMVI fit the gene and protein counts. The evaluation metrices from Section 4.5 are used for experiments evaluation.

5.1. cellPMVI best fits the CITE-seq data (Training scenario 1)

Training scenario 1 does not evaluate the models' ability to generalize to unseen covariates because all covariate combinations are included in the training data set (Figure 4.2a). As no generalization is required, training scenario 1 checks how well the models fit the data set and how certain training conditions influence the model performance.

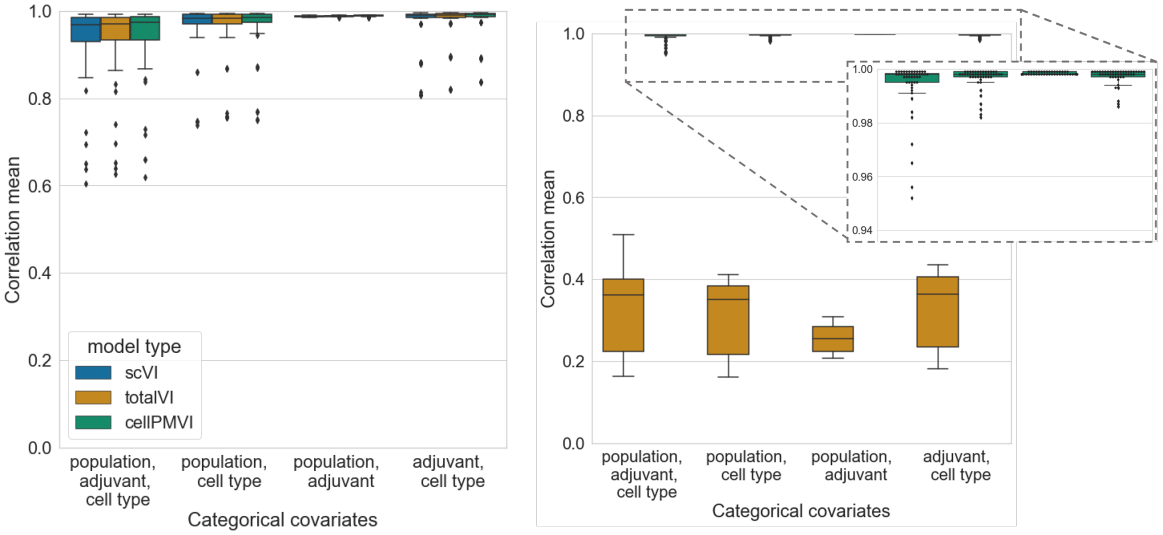
The performance of probabilistic models, such as scVI, totalVI and cellPMVI, depends on how well they fit the underlying data distribution. Validation of the model performance in training scenario 1 uses the loss values and posterior predictive sampling. First of all, the *final* training and validation losses are much lower than the initial values meaning that the model learned a compression of the underlying data distribution (Supplementary Figures C.3, C.4 and C.5). Additionally, the training and validation values did not start to spread apart again, suggesting that no overfitting occurred. Note that, the final ELBO (Supplementary Figure C.2a), RL (Supplementary Figure C.2b) and KL-divergence (Supplementary Figure C.2c) of the three model is lowest for our model (cell PMVI). Thus, the results indicate that all models learned to fit (part of) the data.

After verifying that the models have learned something, the following sections will use the posterior predictive sampling to analyze the quality of fit more in detail.

Posterior predictive sampling performs best when conditioning on two covariates

The correlation between the posterior predictive sampled and actual gene and protein expression indicates how well each model fits the data (Figure 5.1). For Training scenario 1 we calculate the correlation for four different run settings: In the first run setting, the models use three different categorical covariates and, for the other three, a combination of two categorical covariates. Figure 5.1a shows the correlation mean values for the true and sample gene and protein values for each run setting in Table B.3. Together with the correlation variance (Supplementary Figures C.6a and C.6b) this indicates of how well the mean-variance relationship across genes is preserved.

Figure 5.1 shows correlation mean between the predicted and true gene expression under posterior predictive sampling. The correlation mean is calculated for all cells with the same population group, adjuvant and cell-type annotation (see Supplementary Table B.4 for values). Figure 5.1 shows that the average correlation mean is approximately equivalent $(0.93-0.99)$, but not the spread of the correlation mean values. The correlation mean spread is the highest for three categorical covariates and the least when training with population group and adjuvant as covariates. The correlation variance has approximately the same average of $0.9-0.98$ and a similar spread of $0.01-0.11$. We can conclude from the high correlation values that all models can replicate gene expression data with their maintained properties. Still, the covariate combination of population group and adjuvant outperforms the other combinations.



(a) RNA

(b) protein

Figure 5.1: **Boxplot showing the correlation mean for posterior predictive sampling for each model across four run setting (training scenario 1).** The correlation mean is calculated across all cells with the same population group, cell type and adjuvants combination (47 correlation means). The correlation means are clustered per run settings. Each run settings differs from each other in the number of covariates the model is conditioned on (Table B.3). Subfigure (a) shows the correlation mean for the RNA expression sampling and (b) for protein expression sampling.

Figure 5.1b shows the correlation mean between the predicted and true protein measurements under posterior predictive sampling. The correlation mean difference of protein expression reconstruction is much higher than for the RNA (see Supplementary Table B.5 for values). For cellPMVI the average correlation mean for protein reconstruction is slightly higher (~ 0.95 to 1.0) than for RNA reconstruction. On the other hand, the average correlation mean for totalVI is much lower (0.3 to 0.35). Likewise, the average correlation variance of totalVI is worse (-0.4 to -0.35) than for cellPMVI (~ 0.96) (Supplementary Figure C.6). In contrast, the spread of the correlation mean values is equivalent to the RNA expression reconstruction spread (0.01 to 0.11) for both cellPMVI and totalVI. Overall, the results show that cellPMVI clearly outperforms totalVI in the reconstruction of protein measurements.

The quality of fit and the number of cells for training available are positively correlated.

The dataset contains cell measurements from various conditions, i.e. population groups, cell types and adjuvant perturbations. Due to potential variation of expression values across cells, specifically regarding the conditions, it can be that the model fits some parts of the data better than others. Hence, here we are interested in differentiating between less and more favorable conditions for model learning. More specifically, we addressed the questions: 1) *Is the modeling process of some sub-conditions easier than for others?* And 2) *how does the number of categorical covariate conditions impact the modeling process?* This information will optimize the modeling process for training scenarios 2 and 3 to increase the prediction accuracy.

In Section 5.1 we showed that the correlation mean for RNA reconstruction using PPS has overall

good performance. At the same time, Figure 5.1a showed a large spread with outliers. To identify whether these outliers have the same origin we plotted the correlation mean from cellPMVI trained with conditioning on all covariates (run setting one) for all covariates in Figure 5.2. Figures 5.2a and 5.2b show the population group's influence and adjuvant influence on the correlation mean is minimal. However, Figure 5.2c shows that the correlation mean for the seperate cell types is lower for *Platelet* (around 0.7) and the other cell types (all higher than 0.9). A possible reason for this could be that *Platelet* are underrepresented in the dataset making up only approximately 0.5% (295 cells) of the data (Table **??**). The trend of less represented conditions having a lower correlation mean can also be observed for the other conditions but to a lower extent. For example, RT has a slightly lower correlation mean than DK and LD. When comparing that with the amount of data points available, RT is less represented in the data (about 20% - 5 698 nr. of cells) than LD and DK (about 40% (LD: 10 848 and DK: 10 848 nr. of cells) . The same relation between the number of cells and correlation mean holds for scVI and totalVI (Supplementary Table B.6)

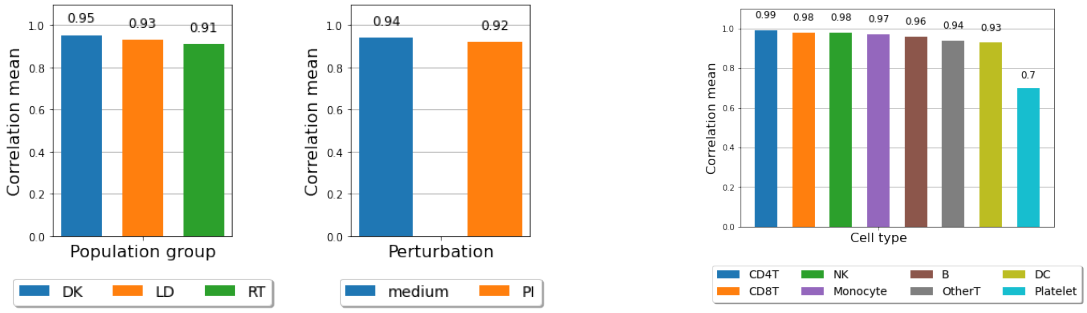


Figure 5.2: **The number of available cells per categorical condition impact the correlation mean.** Posterior predictive sampling results from cellPMVI under run setting one in training scenario one. For each condition the correlation mean is cacluated across all cells (22k) averaged across the conditions of the categorical covariates of interest ((a) Population group, (b) Adjuvant or (c) Cell type).

Additionally, the previous section demonstrates that the PPS also varies for the number of categorical covariates the model is trained on. We summarized that the correlation mean and variance are overall better for training with two covariates instead of three. From the previous and this section, we can see that the correlation mean and variance is highest the more cells are available for the training condition. Hence, the model's quality of fit improves when more cells are available for a training condition. The same relation between the number of cells and correlation mean holds for scVI and totalVI (Supplementary Table B.6)

5.2. cellPMVI is suited for predicting protein measurements and transcriptome data

(Training scenario 2 and 3)

The previous section shows that all models fit the CITE-seq data but that cellPMVI models the gene and protein expression the best. Moreover, the results show that the amount of available data influences the fit. For example, we showed that the model fits cell types with fewer data measurements (i.e. *Platelet* cells) less well and that conditioning on more covariates c decreases the model's fit to the data. Therefore, the models in training scenarios 2 and 3 are only conditioned on the population group and adjuvant because there is not enough data to include more covariates. By limiting the number of categorical covariate conditions, we hope to increase the data availability per condition and overcome inaccurate generalization and uncertainty emerging from little data availability.

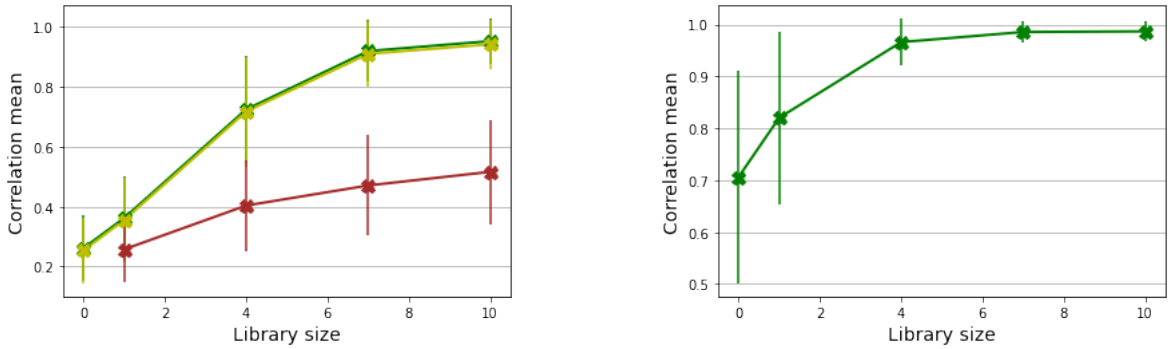
This section aims to answer the research questions posed in the beginning: *To what extent can we predict responses for unseen perturbations across populations?* (**RQ 1**) and *How does multi-modality information impact the modelling and prediction performance?* (**RQ 2**). For question **RQ 1** we focus on the difference in prediction performance between training scenarios 2 and 3. Furthermore, for the second research question, we will analyze the prediction of single-vs multi-modality models and between the multi-modality models. Because we are interested in the prediction performance, we will use prior

and transfer predictive sampling instead of posterior predictive in the previous section.

Library size is an important component for modelling the data

The library size factor is a nuisance factor that influences the likelihood distribution of the RNA $q(x_n \mid z_n, \ell_n, c_n)$ and protein $q(y_n \mid z_n, p_n, c_n)$ data. For posterior and transfer predictive sampling, the library size is estimated from the data passed to the encoder. No such estimation can be made for prior predictive sampling because no data is passed to the encoder. Therefore, the library size is manually set to a constant value (see Section 4.4). The size of the library factor influences how well the likelihood distribution describes the actual underlying distribution of the data.

Figure 5.3 illustrates the effect of the RNA or protein library size on the average and standard deviation of the respective predictions. The correlation mean of the predictions for library sizes of 0, 1, 4, 7 and 10 are shown. Figure 5.3a shows how the RNA size factor ℓ effects the correlation mean. First, note that the correlation mean of scVI and cellPMVI is higher than for totalVI. That means, scVI and cellPMVI are superior at modeling the RNA likelihood distribution. Furthermore, Figure 5.3a illustrates that increasing library size to 10 increases the correlation mean and decreases the correlation variance for all models (Supplementary Table B.7). The average correlation value for scVI and cellPMVI shows a steeper increase than for totalVI, meaning scVI and cellPMVI RNA likelihood distribution are more strongly influenced by the library size factor. Additionally, Figure 5.3b shows the influence of the protein size factor for the protein likelihood modelling of cellPMVI. First, the starting correlation mean value is higher for proteins (~ 0.71) than for RNA (~ 0.26). At the same time, the standard deviation of the correlation mean is almost double as high compared to RNA, about ~ 0.21 for protein and RNA (~ 0.11). The higher standard deviation of the correlation mean might imply that protein features are more distinct from each other even though the underlying distribution of some protein measurements is easier to capture. Potentially due to the less available number of UMI proteins or because protein data is less sparse than RNA. Moreover, the best performance according to the correlation value is at a library size of 7 and 10. Hence, to avoid overfitting, the library size for the RNA and protein reconstruction is set to 7 for prior predictive sampling.



(a) *gene expression* reconstruction

(b) *protein* measurement reconstruction.

Figure 5.3: **Influence of the library size of RNA ℓ and protein p on the average correlation mean and variance of the prior predictive sampling performance.** Models: scVI (yellow), cellPMVI (green) and totalVI (red). The average (cross) and variance (line at cross) of the correlation mean for the library size values of $0, 1, 4, 7, 10$. The results are calculated for all data points for (a) gene expression and (b) protein expression reconstruction.

Transfer predictive sampling enables OOD prediction of perturbed gene expression

We use training scenario 2 to answer the second research question about how well the models can extrapolate to new adjuvants, in this case the perturbed condition (adjuvant PI). Table B.8 shows the training conditions for training scenario 2. The model covariates are the population group and adjuvant. As shown in Table B.8 the perturbed data of each population group was once left out, representing the OOD condition. The models ability to extrapolate to new conditions is measured by the prediction performance of the left-out perturbed gene and protein expression.

Figure 5.4 shows the boxplots of the correlation mean calculated between the true and prior or

transfer predictive sampled gene expression for the excluded data subset, perturbed condition (PI) of each population group. Subfigure 5.4a) shows the correlation mean for the prior predictive samples and b) for the transfer predictive samples. The correlation mean is higher for transfer predictive sampling ($\sim 0.7 - 0.75$) than prior predictive sampling ($\sim 0.05 - 0$). Compared to that, when predicting for the unperturbed condition that was included in the data during training, the prior predictive sampling ($\sim 0.8 - 0.99$) outperforms the predictive transfer sampling ($\sim 0.7 - 0.8$), see Table B.9. Thus, only transfer predictive sampling is suitable for predicting the gene expression responses for new adjuvants.

Figure 5.4 shows no remarkable difference across populations or models when performing OOD prediction. More specifically, the difference between correlation means for each population group is at most 0.04 for prior and transfer predictive sampling. From the results, we can not conclude whether the generalization across African population groups (DK and RT) is better than the European population group (LD).

Lastly, Figure 5.4 shows that while scVI slightly outperforms totalVI and cellPMVI for transfer predictive sampling, it performs equivalent for the prior predictive sampling. That means that single-modality data is sufficient for predicting gene expression values.

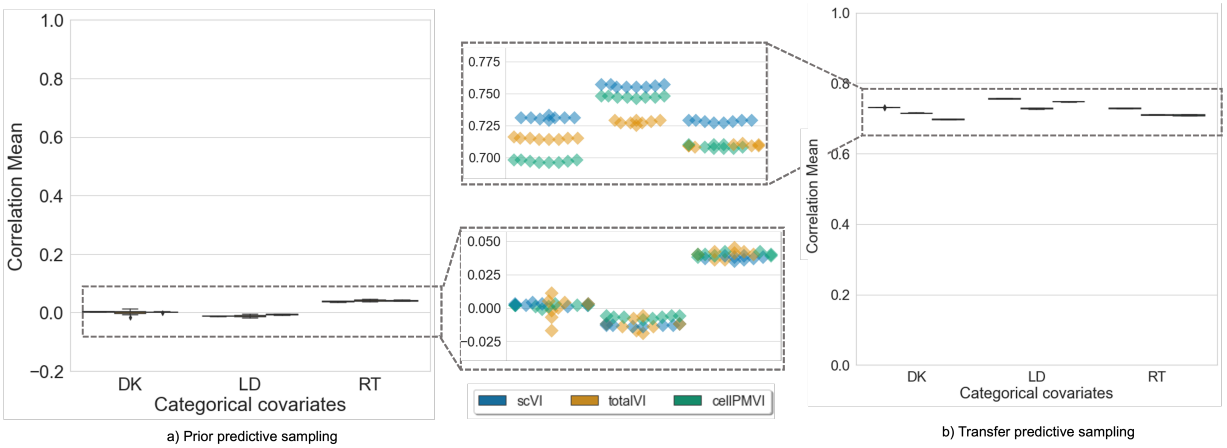


Figure 5.4: **Boxplot showing the correlation mean between the true and prior and transfer predictive sampled gene expression for the OOD perturbed population groups (training scenario 2).** The correlation mean is calculated across all cells with the same population group, either DK, LD or RT, and adjuvant, PI (8 data points per model). For each left out population group the correlation mean of every model (scVI: left, totalVI: middle, cellPMVI: right) is shown. Subfigure (a) shows the correlation mean for the prior predictive sampling and (b) for transfer predictive sampling.

cellPMVI outperforms totalVI for protein measurement prediction

The multi-modality models, totalVI and cellPVMI, are additionally compared on their protein expression prediction performance using training scenario 2. Figure 5.5 shows the average correlation mean values for the reconstruction of protein measurements for the prior and transfer predictive distribution (see Supplementary Table B.10 for exact values).

Figure 5.5 shows that cellPMVI clearly predicts the protein measurements better than totalVI for both, prior and transfer predictive sampling. The correlation mean average for cellPMVI (~ 0.9) is almost four times as much as for TotalVI (~ 0.25). Compared to the gene expression prediction (Section 5.2), there is no significant difference between the prediction quality of prior and transfer predictive sampling. Additionally, the correlation mean difference between the non-OOD (medium) and OOD prediction (PI) scenario is not as large as for the gene expression. Most interestingly, the correlation mean for the OOD is slightly larger than for the non-OOD prediction. Lastly, there is no noticeable difference in the reconstruction quality across populations, equivalent to what has been observed for the gene expression prediction (Figure 5.4). Again a similar trend can be observed for the correlation variance.

Predicting expression ranks is better than count prediction

In this section we are interested in investigating the gene-to-gene correlation between the predicted and original expression instead of comparing the correlation mean of conditions. We are predicting

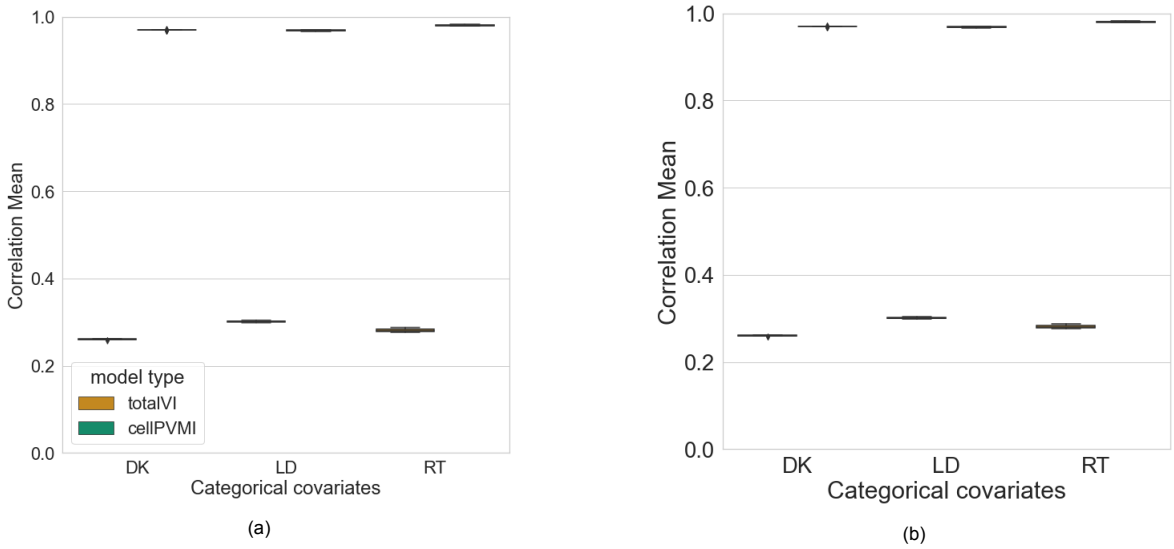


Figure 5.5: **Boxplot showing the correlation mean between the true and prior and transfer predictive sampled protein expression for the OOD perturbed population groups (training scenario 2).** The correlation mean is calculated across all cells with the same population group, either DK, LD or RT, and adjuvant, PI (8 data points). For each left out population group the correlation mean of every model (scVI: left, totalVI: middle, cellPMVI: right) is shown. Subfigure (a) shows the correlation mean for the prior predictive sampling and (b) for transfer predictive sampling.

the perturbed gene expression values with cellPMVI using transfer predictive sampling and the training scenario with population group LD excluded because this has performed best for extrapolation of new adjuvants.

When plotting the correlation values of original and reconstructed against each other we observed that there is one outlier with almost three times as high expression value compared to the other genes. The genes that are sampled too highly are either *MALAT1* or *FTH1*. Because the genes have a higher count than the other genes cellPMVI seems to overrepresent them as well in the modelling process. In Figure 5.6b the genes *MALAT1* or *FTH1* to provide a better overview of the other genes.

Figure 5.6a shows that the higher the original count value is the more the value of the predicted gene expression gets overestimated. While the prediction of the exact value seems to get harder with higher expressed genes, Figure 5.6b shows that the rank gets equally well predicted across all gene expression sizes. Thus, cellPMVI estimates the gene expression ranks better than the actuall gene expression values.

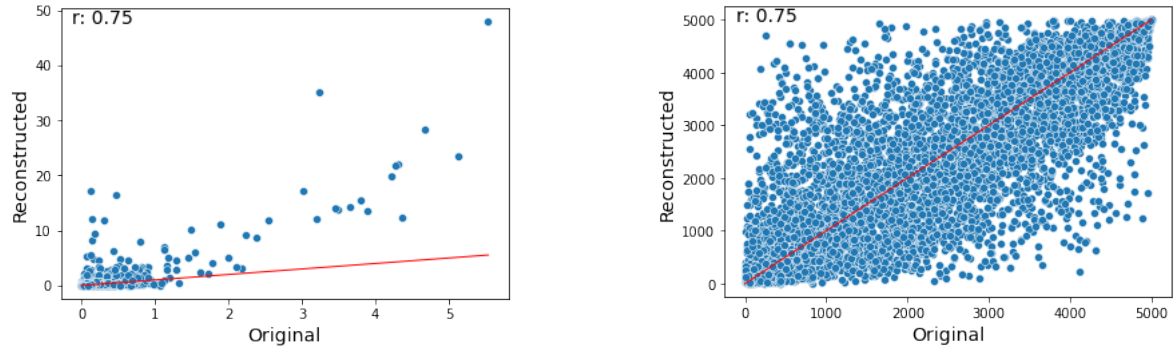


Figure 5.6: **Correlation between predicted and original gene expression values and ranks (Training scenario 2).** The predicted perturbed values are either transfer predictive sampled from the training population group with LD excluded. The left plot shows the correlation of the gene expression values (a) and the right the gene expression ranks (b).

Table 5.1: Common highly variable gene (from 1 000) for transfer predictive sampling with left out LD population group.

Adjuvant	DK	LD	RT
medium	618	**606**	620
PI	614	**576**	616

OOD prediction of RNA expression does not work for novel population groups

The evaluation of the models for extrapolation to adjuvants shows that transfer predictive sampling is best suited for OOD prediction and that cellPMVI outperforms the other models. In addition to the extrapolation to adjuvants the OOD prediction performance is evaluated on the ability to extrapolate to novel population groups. For the OOD prediction evaluation for novel population groups we trained the models with the training set excluding one population group at a time. To be able to perform prediction for new population groups the population groups are not included as categorical covariates. Instead the models are trained with categorical covariates: 1) adjuvant and cell type and 2) only adjuvant. The Figures in this report show the results for the second conditioning, only on adjuvants, to be able to compare it to the previous results that have also not been conditioned on the cell type. After training the correlation mean and variance for the prediction of the perturbed gene and protein expression of the withheld population group are calculated. Here we show the correlation values from prior and transfer predictive sampling for cellPMVI.

Figure 5.7 shows the correlation mean of the perturbed RNA expression for each excluded population group (see Supplementary Table B.11 for the average correlation mean values of all models). Figure 5.7 shows that the prediction using prior predictive sampling is higher than for transfer predictive sampling and that the correlation mean for LD is highest and for DK the lowest for both sampling scenarios. Note that the correlation values for medium and PI condition are approximately equivalent (see Table B.11). That means the model does not seem to learn a difference when only conditioning on adjuvants. As mentioned before we also trained the models using two conditioning factors, cell type and adjuvant. When conditioning on adjuvant and cell type the correlation mean is higher for larger cell types such as B cells (~ 0.08), CD4T(~ 0.065) and CD8T (~ 0.075) cells. However, overall the correlation mean for this OOD prediction is much smaller than for the extrapolation to new adjuvants.

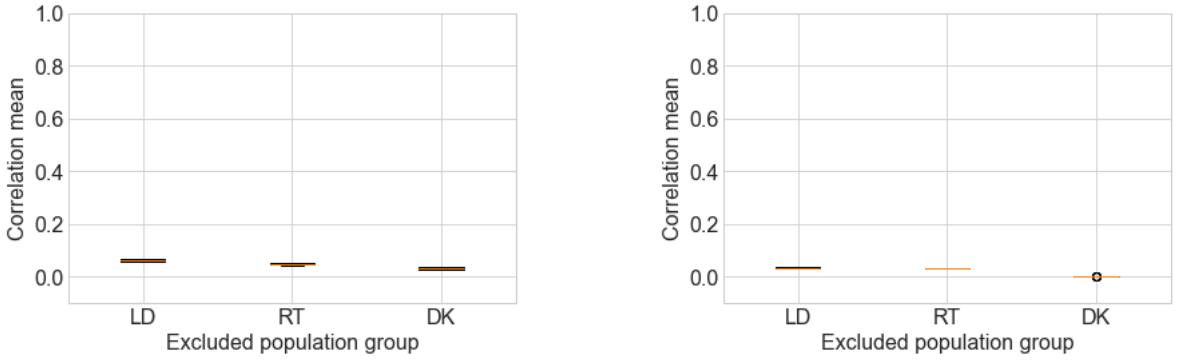


Figure 5.7: **Correlation mean of cellPMVI for gene expression (training scenario 3).** Comparison of the correlation mean for each excluded population group given a) prior predictive sampling and b) transfer predictive sampling.

Next, Figure 5.8 illustrates whether extrapolation to a population group can is better than to another and if prior or transfer predictive sampling is better suited for this OOD prediction. Figure 5.8 shows higher correlation mean values are achieved with prior predictive sampling (~ 0.94). Besides that the correlation means differ at most 0.01 across population groups. Furthermore, cellPMVI achieves a higher correlation mean for the prediction of protein expressions compared to totalVI $0.2 - 0.3$. Note that there is again no performance difference between the prediction of perturbed and unperturbed expressions. Thus, the extrapolation of novel population groups protein expressions is possible with cellPMVI using prior predictive sampling.

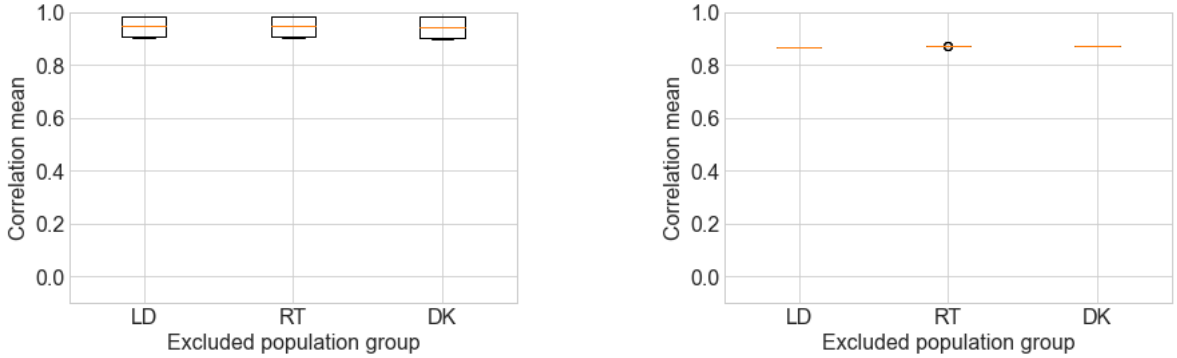


Figure 5.8: **Correlation mean of cellPMVI for protein expression (training scenario 3).** Comparison of the correlation mean for each excluded population group given a) prior predictive sampling and b) transfer predictive sampling.

6

Discussion

Currently, no computational method can predict perturbations for multi-modal single-cell data. Therefore, we designed cellPMVI to model and predict immune responses across populations given multi-modal single-cell data. More specifically, the goal was to evaluate the prediction performance of cellPMVI to answer counterfactual questions and whether multi-modality information improves the modeling compared to single-modality. cellPMVI was evaluated on CITE-seq data to predict immune responses across populations. The results show that cellPMVI can model and predict perturbed gene and protein counts. Nevertheless, analysis indicates that information about a population-specific response improves prediction performance. In this section, we will discuss some of the results in more detail and compare our results to related research.

6.1. Out-of-distribution prediction

The first focus of this work was to investigate whether the models can perform OOD prediction (**Q1)**. More specifically, we consider two OOD prediction scenarios, each representing a different degree of generalization: 1) extrapolation to adjuvants and 2) extrapolation to novel population groups.

The OOD prediction for the first scenario is less complex than for the second scenario because the training data set included at least the unperturbed conditions of the population groups. For the first OOD prediction scenario, cellPMVI could accurately predict both the gene expression and protein expression with transfer predictive sampling. However, cellPMVI failed to predict the measurements with prior predictive sampling. That means, prediction is impossible without any latent space information about the to-be-predicted measurement.

We can make a similar observation when looking at the results for the second situation, extrapolation to novel population groups. In the second OOD scenario, gene expression prediction was impossible for either prior or transfer predictive sampling. This confirms our previous statement that latent space information is necessary for prediction. Generally, it is a known problem that OOD prediction for entirely new data distribution is difficult. Therefore, many models, such as CPA [23], focus on OOD prediction for the first and not the second scenario. For example, CPA predicts drug responses for new dosages but not new drugs. However, there is no comparison between the success of OOD prediction for genes compared to proteins. By looking into the reasons for this difference, one might be able to identify the problem point and potentially improve the gene expression OOD prediction.

Interestingly, while OOD prediction was unsuccessful for gene expression prediction, it was possible to predict proteins. The successful prediction of proteins could be due to various reasons. For example, it might be that protein expression responses require less complex distributions because there are fewer proteins (164) compared to genes (5 000) in the training data. Furthermore, protein data is more expressive (less sparse), which could improve the modeling of the data as the many zero counts in RNA data provide one of the main difficulties in single-cell modeling. Lastly, population groups might have more similar protein expressions than gene expressions and, therefore, easier to model. However, a more detailed investigation of the OOD prediction for genes compared to proteins is required. By looking into the reasons for this difference, one might be able to identify the problem point and potentially improve the gene expression OOD prediction.

6.2. Single- vs multi-modality information

The second research question (**RQ 2**) addresses the impact of multi-modality information on immune response modeling. To evaluate the influence of multi-modality information, we compare the modeling quality of the gene expression data as this can be modeled by both the single-modality (scVI) and multi-modality (totalVI and cellPMVI) models. We hypothesize that immune response modeling is better for multi-modality models than single-modality models (**H1**) because the proteomics data adds information unavailable from transcriptomic measurements alone.

The results indicate that scVI, totalVI and cellPMVI model the gene expression responses equally well, given posterior predictive sampling (Figure 5.1, C.6). This is in line with results reported in [7] where the posterior predictive sampling results in a relative log-likelihood for RNA data for both totalVI and scVI. Because of the already high prediction performance for gene expression measurements, it is unclear whether the prediction benefits from additional modality information, such as protein data. Therefore, we currently do not have enough evidence to reject or accept our hypothesis about the added value of multi-modal data (**H1**). Further experiments could help to understand the influence of multi-modal data on the model prediction. For example, one could introduce noise to the RNA dataset, i.e., by removing a gene from the RNA data and shuffling the data, and then predict the expression with and without the information of protein data. If the protein data supports RNA expression prediction, then the prediction quality should be higher with the protein data.

Although we can not provide clear evidence that multi-modal data benefits the prediction of gene expression (**H1**), the ability of cellPMVI to predict protein counts in conjunction with gene expression is highly valuable. For example, considering that protein measurements are relevant for annotation of cell types, which information is used to train all models, the multi-modality models provide the advantage of reconstructing these protein counts for gene expression across populations compared to single-modality models.

6.3. Protein measurements

Our results illustrate that cellPMVI predicts protein counts more accurately than gene expressions. This observation is not surprising because proteins are pre-selected by their representation of the immune effects. Therefore, it is easier for the model to summarize the features in a low-dimensional space.

Interestingly, our results show that totalVI performs less well at protein modeling than gene modelling. When comparing our results with the original results from [7] we see the same observation, namely that genes have a higher posterior predictive mean than proteins (Extended Data Fig. 2a,b,d in [7]). The inferior performance of totalVI for protein modeling compared to gene modeling suggest that totalVI can not capture the protein features as well in the latent space as the gene expression features. This might imply that totalVI can fails to capture the distinct range values of gene and protein expression.

On the other hand, the cross-modal prediction from protein-protein or transcriptome-protein in [28] shows a similar high correlation (Figure 2g in [28]) as in our results for cellPMVI. Hence, the results suggest that the MoE integration from MMVAE [31] has a better latent space integration for multi-modalities than the joint posterior modeling approach in totalVI.

6.4. Library size

The library size approximates the relative size of a cell because the number of RNA transcripts and protein molecules scales with the size of a cell [26]. Both scVI and totalVI, use the library size for scRNA-seq data modeling as a nuisance factor reflecting a combination of sequencing depth and cell size. However, [7] have decided against taking protein library sizes into account because they believe that the biased sampling procedures for proteins do not approximate the relative size of a cell. In cellPMVI we decided to model the library size for proteins because of the significant consequence it had on the quality of RNA modeling reported in [21] and can be observed in Figure 5.3a. Considering the results from [7] (Supplementary Figure 10 in [7]) show that the impact of the protein size library differs per cell type, meaning that the protein library size value has more impact on some cell types than others. Investigating which cell types are most accurately represented for the protein library size could further improve the prediction of protein expression in cellPMVI.

6.5. Interpretability

Currently, cellPMVI is mainly evaluated on the quality of the decoder output. To increase the confidence about cellPMVIs performance, i.e., in answering counterfactual questions, it is necessary to consider which underlying features the model learns. However, a common problem with VAE-based models is their black-box behavior, limiting the interpretation of the latent space. The limited interpretation of the latent space makes it hard to observe represented features in the latent space. Consequently, the ability to use a method's parameters or apply them in further downstream analysis is restricted. Increasing the interpretability of the latent space is essential to enhance the confidence in a model's performance and use it to directly or indirectly answer questions like: 'Which cell type responds the most to perturbation?' or 'Which cell type responses vary across population groups?' Thus, analyzing the interpretability of the latent space would increase confidence in the performance of cellPMVI.

Disentanglement is one common way to improve the latent space's interpretability. In a disentangled latent space, a single dimension is linked to a single generative feature [25]. cellPMVI disentangles latent features by approximating an isotropic Gaussian or Laplace prior. Both priors are invariant to rotation and therefore encourage the latent variables to take on a meaningful representation [27]. From the low KL divergence of cellPMVI (Supplementary Figure C.2c) we can assume that the prior is approximated sufficiently close in the latent space suggesting disentangled features. Nevertheless, the problem remains that we do not know what features a latent dimension represents. Understanding which sources of variation are disentangled in the latent space would not only improve the trust in cellPMVI but also make it easier to perform prediction tasks. Additionally, it could help us learn more about the underlying biology by comparing shared and private features between the domains.

6.6. Benchmarking

At the moment, there are no models that perform perturbation prediction on multi-modality data. There are, however, models that either integrate multi-modalities (e.g. totalVI [7]), cross-predict modalities (e.g. scMM [28]) or predict perturbations on single-modality data (e.g. CPA [23], chemCPA [10]). Hence, any model requires an adjustment to be adequate as benchmarking model.

We decided to benchmark cellPMVI against the CVAE versions of scVI and totalVI with new implementations for the prior and transfer predictive sampling to predict unseen cell responses. Note that neither of these models are originally made for predicting unseen perturbations, which could be a potential reason for e.i. totalVIs inferior performance for protein prediction. Nevertheless, our results show that cellPMVI also outperformed totalVI for the posterior predictive sampling, a task totalVI was designed for.

We also considered benchmarking cellPMVI against the other models mentioned above. CPA seemed most promising, but it can predict responses to unseen drug dosages rather than new drugs. This restriction is due to the embeddings that CPA uses. If a drug combination was not present during training, the model did not create an embedding for the combination; hence, it cannot predict this combination later on. Consequently, adjusting CPA for benchmarking would require multi-modality integration and changing the embeddings to allow for unseen perturbation predicition. Next, we considered chemCPA, which builds on top of CPA. chemCPA aims to predict unseen compounds. However, chemCPA does not integrate multi-modality data, and its purpose of encoding a drug's molecular structure is not required for the goal of this work. Furthermore, scMM is an extension of MMVAE without being able to model the specific biological factors of RNA and protein. Considering the library size's advantage on the modeling process as described in [21] and shown in Figure 5.3 we decided that benchmarking against models including this factor is more valuable. Nevertheless, it would be interesting to support the hypothesis of the library size by benchmarking cellPMVI against scMM.

6.7. Limitations

The generalizability of the results is limited because all experiments are conducted with the same dataset (see Section 4.2). This dataset has some pitfalls. First, the dataset is relatively small in that it only includes two individuals for each population group. Our results suggest that the modeling and prediction performance would improve with more data points per population group. Secondly, the data set includes only one perturbed condition. Training with a dataset that includes more perturbation measurements is required to analyze the OOD prediction performance better. Lastly, all data was

measured in one lab. That means the model learns from homogenous data, likely not containing batch effects. Testing how well the model can generalize perturbations from a heterogenous dataset would be interesting for further validation. If this is successful, then this provides an opportunity for the model to train with more data, e.i. more individuals per population group. Although training cellPMVI on a larger dataset or a combination of different datasets would increase the confidence in the model's prediction performance, the current results do not suggest that the model can not do that.

We predict all OOD prediction scenarios for population group, ignoring individuals in the data set. That means, we implicitly assume that individual differences within one population group are more diminutive than across population groups. Ideally, one should account for individual differences as well. However, in this work, we decided against modeling individual differences due to the lack of individual data points per population group. While this assumption might impact the modeling, which would impact the generalization, research suggests that immune responses vary more across populations than within.

Lastly, all models use the same hyperparameters for all training scenarios (see Section 4.3). These parameter values are based on the suggestions in [21] and [7]. Although we show that cellPMVI fits the data with these parameter values, we believe that the model performance can be further improved with a specific hyperparameter selection. Especially, the optimization of latent dimensions size can be critial when training VAE based frameworks ([21], [7], [4]).

7

Future work

The experimental results and discussion in this work suggest various opportunities for future research in different directions. In this section, we will pick up some of these directions and propose specific adjustments that could lead to improvements or additions to cellPMVI.

7.1. Loss function of cellPMVI

As described in the methods (Section **??**), cellPMVIwas implemented with the basic ELBO loss function (Equation 2.8). However, [31] show that a more accurate joint posterior can be learned using the doubly reparameterized gradient (DReG) estimator. The DReG estimator [34] offers a tighter lower bound (with a lower variance gradient estimator) compared to the ELBO loss used in this work. An improved lower bound estimate might make it possible to represent the different underlying distributions of the population groups better.

7.2. Cross generation for OOD prediction

Currently, cellPMVIuses an expert for each modality to perform multi-modality prediction. That means the joint latent posterior space captures shared and individual features from RNA and protein measurements. Instead of modeling the features of the RNA and protein measurements, it would be interesting to consider modeling the population group-specific responses. To model population, group-specific responses cellPMVI would use an encoder-decoder pair per population group instead of data modality. An expert per populaiton group would allow considering individual differences as we could fit the individuals as covariates. The single-cell input data could either be concatenated (following totalVI approach) or restricted to one data modality, such as RNA measurements. When predicting gene expression responses, the latter would be sufficient as we have shown that reconstruction of gene expressions does seem to perform equally well without protein information. In both situations, the goal would be to model the shared and individual features of the population groups rather than the RNA and protein modalities. Following this approach, cross-generation could be used for OOD prediction of population group responses.

7.3. Disentanglement

Disentanglement is essential to increase the trust and prediction performance of cellPMVI. Here we suggest two approaches to increase disentanglement in the latent space.

beta-VAE

A common way to give the learning of disentangled representations more weight can be achieved by changing the VAE components of the cellPMVI model to a β-VAE [11]. β-VAE [11] uses the β hyperparameter in its loss function to restrict the encoding capacity of the latent space. In that way, β encourages the factorization of the bottleneck, hence, controlling the latent overlap between each data point.

Adversarial learning

Another way to improve disentanglement is to follow an adversarial approach [17], as has been done in CPA [23]. Adversarial learning methods involve two networks: the generative network tries to learn a feature map from the input data in such a way that the discriminative network cannot predict the domain type given the output of the generative network. In that way, the generative network learns to align domain gaps in the feature level. We suggest adding a discriminative network for each categorical covariate to enhance the disentanglement of features across the categorical covariates in the latent space. Besides the increase in disentanglement, the adversarial loss of each discriminative network would provide insight into the latent space disentanglement and hence, increase the trust in cellPMVI.

7.4. Biological relevance

cellPMVIwas designed to extrapolate immune responses from population groups to novel adjuvants or population groups to reduce the need for large-scale single-cell perturbation experiments. The results show that the prediction of perturbed gene and protein measurements can be inferred from other population groups, given the unperturbed measurements. Here we want to discuss some real-life scenarios in which the model could be used.

Annotation of data

cellPMVIis based on the MMVAE model, which allows for a cross-modal generation. Given the data from one modality, the model can generate data from a different modality. A cross-modal generation has not been in the scope of this research but could be used to annotate data by predicting protein expression from gene measurements. For example, to predict protein expressions of cell populations that require the combination of protein and transcriptome data (e.i. CD4+ and CD4+T) for annotation.

Speed up adjuvant development

Testing adjuvants in clinical trials for approval can take up to years. Current suggestions to speed up the development pipeline is to proceed in small trials such that unsuccessful adjuvants can be eliminated early on [30]. Models, such as cellPMVI, that can predict the response of adjuvants can be used to set up a hypothesis about the adjuvant responses in a new population group and hence, offer a preselection of potentially more effective adjuvants. Preselection increase the development speed of vaccine adjuvants

Other application fields

It should also be investigated whether cellPMVI can predict other perturbation responses. For example, can cellPMVIpredict responses to drugs or even drug combinations? Besides that, cellPMVI should be tested on multi-modality that are not CITE-seq measurements. For example, another two modality data combination would be multiome data containing the single-cell ATAC-seq (scATAC-seq) and gene expression measurements. Moreover, prediction for more than two-modality data would be interesting. One could combine RNA, protein, and ATAC measurements or integrate spatial information. The latter could improve the understanding of perturbations as tissue functions that are spatially close to each other might be impacted [20].

8

Conclusion

This research aimed to investigate how well immune responses across populations can be modelled with CITE-seq data. At the moment there are no single-cell models that can capture perturbations from multi-modality data. Therefore, we proposed cellPMVI which combines scVI with mixture-of-experts integration of the posterior to model immune responses across populations. We showed that cellPMVI fits the CITE-seq the best without any information loss for either modality, gene or proteins, and that it can be used to extrapolate to adjuvant responses across population groups with transfer predictive sampling. To better understand the potentials of cellPMVI it is important to validate cellPMVI with more data and evaluate the latent space distribution.

A

Supplementary methods

A.1. Single-cell RNA analysis

Preprocessing

The gene expression data was mapped with cellranger 5.0.0 to GRCh38 human genome reference (10X-distributed 2020-A version). Feature reference was also included in the cellranger to process Feature Barcode data. Droplet containing cells were called using *emptyDrops* from *DropletUtils* `R` package. Low-quality cells were filtered out using adaptive thresholds of 3 median absolute deviation (MAD) for the number of UMI, number of genes, and proportion of mitochondrial genes. Cell hashes were then demultiplexed using *hashedDrops*.

Data annotation

Data normalization and preprocessing were performed using the Seurat V4 workflow. Gene expression and ADT data were normalised using *sctransform* v1 and *dsb*, respectively. Top 3,000 highly variable genes and all ADT features were used in dimensionality reduction using PCA. 30 principal components from each modality was used as input for multi-modal nearest neighbor analysis using *FindMultiModalNeighbors*. The resulting multi-modal graph was then used for Louvain clustering and UMAP dimensionality reduction. To annotate cell populations, we performed marker gene analysis using the Wilcoxon test and also visualised the expression of marker gene and ADT from Azimuth's Human PBMC reference data. Cells were then annotated in different level of details following both the L1 and L2 cell type annotation as shown in Azimuth

B

Supplementary tables

B.1. Dataset

Table B.1: Overview of differentially expressed genes present in czi data before and after preprocessing step.

Cell type	Differentially expressed genes	
	Before preprocessing	**After preprocessing**
B	4545	1852
CD4T	7777	3113
CD8T	6867	2794
DC	1782	974
Monocyte	4186	1904
NK	4751	2012
OtherT	3629	1466
Platelet	195	77

Table B.2: Number of cells per condition.

Cell type	DK		LD		RT		Total	
	medium	PI	medium	PI	medium	PI	medium	PI
B	404	297	228	340	367	167	999	804
CD4T	1844	1350	1942	2344	1338	804	5124	4498
CD8T	1333	1068	1103	984	482	291	2918	2462
DC	133	69	86	81	68	29	287	179
Monocyte	972	358	958	380	444	73	2374	811
NK	515	354	615	549	700	659	1830	1562
OtherT	1327	730	251	158	159	67	1737	955
Platelet	68	26	25	26	39	11	132	63
	6596	4252	5183	4862	3597	2101	15401	15654
	10848		10045		5698		31055	

B.2. Training scenario 1

Table B.3: Overview of the categorical covariate selection for each run in training scenario 1. Each row indicates which categorical covariates the run was conditioned on, by X.

Run	Categorical covariates		
	Population group	Adjuvant	Cell type
1	X	X	X
2		X	X
3	X		X
4	X	X	

Table B.4: RNA expression posterior predictive sampling

		scVI		TotalVI		MMVI	
		Corr mean	**Corr var**	**Corr mean**	**Corr var**	**Corr mean**	**Corr var**
3: PG, ADJ, CT	Average	0.93	0.91	0.93	0.91	0.93	0.92
	Std	0.1	0.11	0.1	0.11	0.1	0.1
2: PG, CT	Average	0.96	0.93	0.96	0.94	0.96	0.94
	Std	0.07	0.09	0.07	0.08	0.07	0.08
2: PG, ADJ	Average	0.99	0.95	0.99	0.95	0.99	0.95
	Std	0	0.01	0	0.01	0	0.01
2: CT, ADJ	Average	0.97	0.95	0.07	0.95	0.98	0.96
	Std	0.05	0.07	0.05	0.07	0.04	0.06

Table B.5: Protein expression

		TotalVI		MMVI	
		Corr mean	**Corr var**	**Corr mean**	**Corr var**
3: PG, ADJ, CT	Average	0.33	-0.37	0.99	0.96
	Std	0.09	0.11	0.01	0.03
2: PG, CT	Average	0.31	-0.41	1	0.96
	Std	0.09	0.09	0	0.01
2: PG, ADJ	Average	0.31	-0.41	1	0.96
	Std	0.09	0.09	0	0
2: CT, ADJ	Average	0.33	-0.41	1	0.96
	Std	0.09	0.09	0	0.01

Table B.6: Correlation mean for each condition from the categorical covariates (Population group, Adjuvant, Cell type). Posterior predictive sampling has been used for the analysis. Values supporting Figure **??**.

Condition: CT	**scVI**	**TotalVI**	**MMVI**
B	.96	.96	.96
CD4T	.99	.99	.99
CD8T	.98	.98	.98
DC	.93	.93	.93
Monocyte	.97	.97	.97
NK	.98	.98	.98
OtherT	.94	.94	.94
Platelet	.69	.69	.70

Condition: Adjuvant	**scVI**	**TotalVI**	**MMVI**
medium	.94	.94	.95
PI	.92	.92	.92

Condition: Population group	**scVI**	**TotalVI**	**MMVI**
european	.95	.95	.95
african-urban	.93	.93	.93
african-rural	.91	.91	.91

Table B.7: Library size values.

Library size	scVI		TotalVI		MMVI		MMVI	
	Mean	Var	Mean	Var	Mean	Var	Mean	Var
0	0.25	0.1	NaN	NaN	0.26	0.1	0.171	0.21
1	0.36	0.14	0.26	0.11	0.36	0.14	0.182	0.17
4	0.72	0.18	0.4	0.15	0.72	0.18	0.197	0.05
7	0.91	0.11	0.447	0.17	0.92	0.11	0.198	0.02
10	0.94	0.08	0.52	0.17	0.95	0.08	0.199	0.02

Library size 0 1 4 7 10

B.3. Training scenario 2

Table B.8: All runs for training scenario 2 with run setting 1 from Table B.3. Every time the perturbed dataset of a population group is excluded.

Run	Excluded data subset		
	Population group	Adjuvant	Cell type
1	LD	PI	
	DK	PI	
	RT	PI	

Table B.9: Averaged correlation mean for the left out population group

		scVI		TotalVI		MMVI	
		Medium	**PI**	**Medium**	**PI**	**Medium**	**PI**
RT	*Posterior*	*0.98*	*NaN*	*0.98*	*NaN*	*0.98*	*NaN*
	Prior	0.98	0.04	0.79	0.04	0.981	0.04
	Transfer	0.72	0.73	0.71	0.7	0.71	0.71
DK	*Posterior*	*0.98*	*NaN*	*0.98*	*NaN*	*0.98*	*NaN*
	Prior	0.98	0.003	0.84	0.0	0.98	0.002
	Transfer	0.83	0.73	0.72	0.71	0.7	0.7
LD	*Posterior*	*0.98*	*NaN*	*0.98*	*NaN*	*0.98*	*NaN*
	Prior	0.97	0	0.81	0	0.97	0
	Transfer	0.76	0.76	0.97	0.73	0.75	0.75

Table B.10: Average correlation mean of protein reconstruction for training scenario 2. The model was trained with categorical covariates population group and adjuvant with data excluding the perturbed individuals of one population group at a time.

		TotalVI		MMVI	
		Medium	**PI**	**Medium**	**PI**
RT, PI	Prior	0.2	0.28	0.9	0.98
	Transfer	0.21	0.21	0.87	0.87
DK, PI	Prior	0.21	0.26	0.92	0.97
	Transfer	0.21	0.21	0.87	0.87
LD, PI	Prior	0.25	0.3	0.91	0.97
	Transfer	0.25	0.25	0.9	0.9

B.4. Training scenario 3

Table B.11: Correlation mean for each excluded population group for RNA expression

		scVI		TotalVI		MMVI	
		Medium	**PI**	**Medium**	**PI**	**Medium**	**PI**
RT	Prior	0.05	0.05	0.05	0.05	0.05	0.05
	Transfer	0.03	0.03	0.03	0.03	0.04	0.03
DK	Prior	0.07	0.06	0.04	0.03	0.07	0.06
	Transfer	0.04	0.04	0.01	0.01	0.04	0.04
LD	Prior	0.03	0.02	0.06	0.05	0.04	0.03
	Transfer	0.02	0.02	0.04	0.04	0.003	0.003

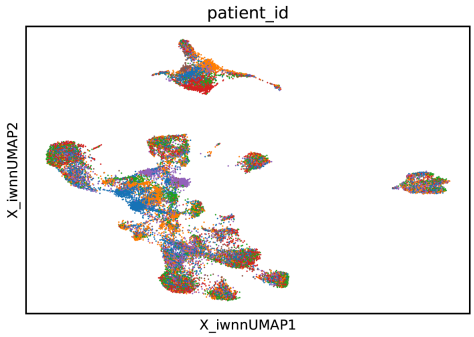
Table B.12: Correlation mean for each excluded population group for protein expression

		TotalVI		MMVI	
		Medium	**PI**	**Medium**	**PI**
RT	Prior	0.22	0.27	0.91	0.98
	Transfer	0.22	0.22	0.87	0.87
DK	Prior	0.19	0.26	0.91	0.98
	Transfer	0.19	0.19	0.87	0.87
LD	Prior	0.21	0.28	0.91	0.98
	Transfer	0.21	0.21	0.87	0.87

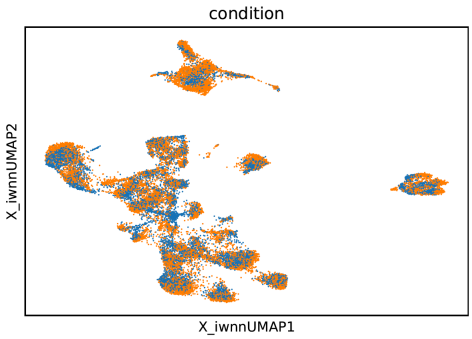
C

Supplementary Figures

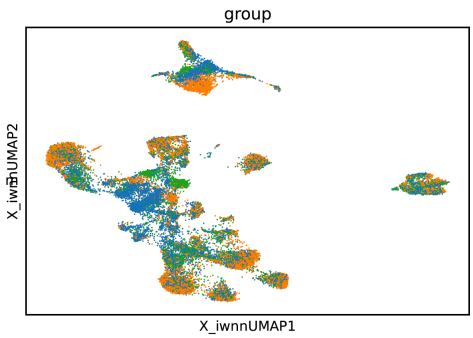
C.1. Dataset



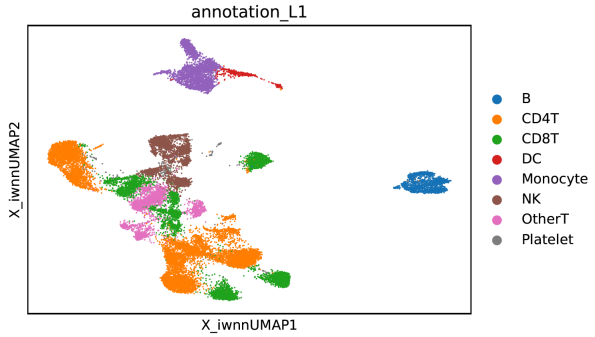
(a) Patient ID coloring: Blue = DK06, Orange = DK68, Green = LD254, Red = LD276, Purple = RT55, Brown = RT162.



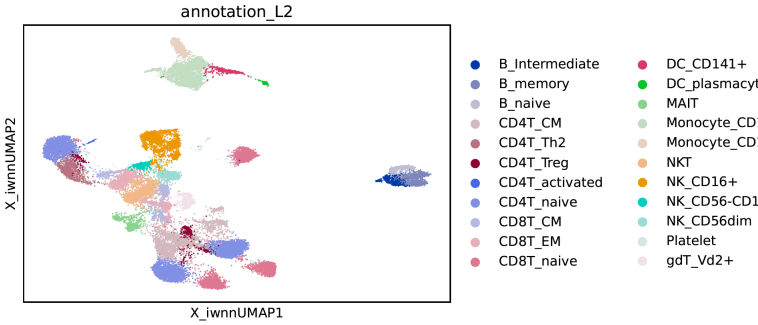
(b) Condition coloring: Blue = PI, Orange = medium.



(c) Group coloring: Blue = DK, Orange = LD, Green = RT.



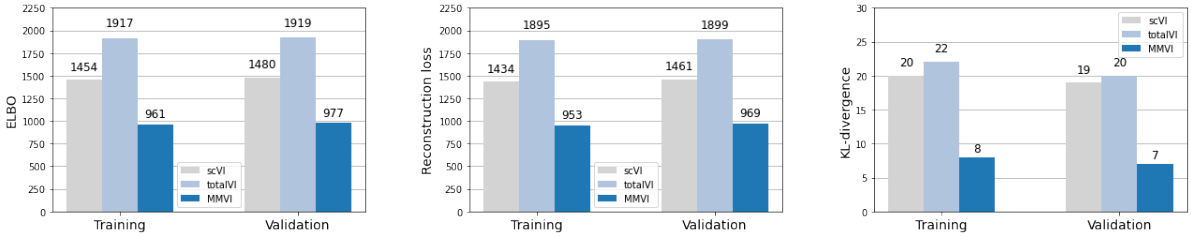
(d) Annotation L1 coloring.



(e) Annotation L2 coloring.

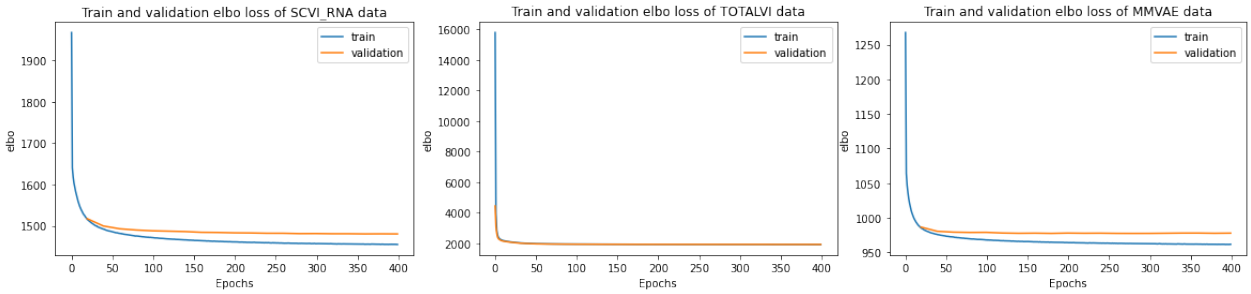
Figure C.1: UMAP of czi data for five different coloring conditions.

C.2. Training scenario 1



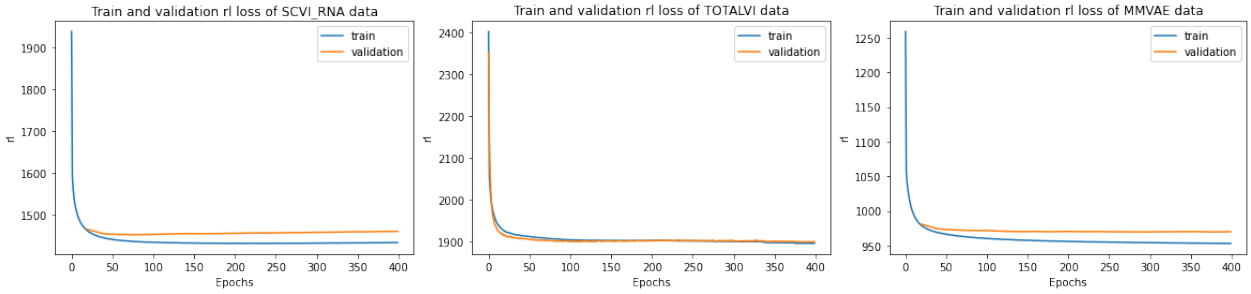
(a) (b) (c)

Figure C.2: **Final training and validation losses for scVI, totalVI and MMVI.** Losses from training scenario one, run setting 1 (Table B.3). The losses visualized are: a) ELBO loss, b) Reconstruction loss and c) KL divergence.



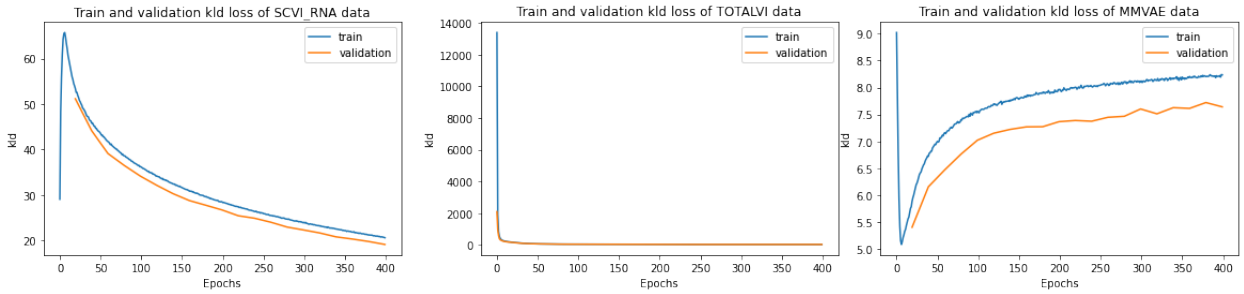
(a) scVI (b) TotalVI (c) MMVI

Figure C.3: Comparison of train and validation ELBO for all three models. All models have been trained under Training scenario 1 with three categorical covariates: population group, cell type and adjuvant.



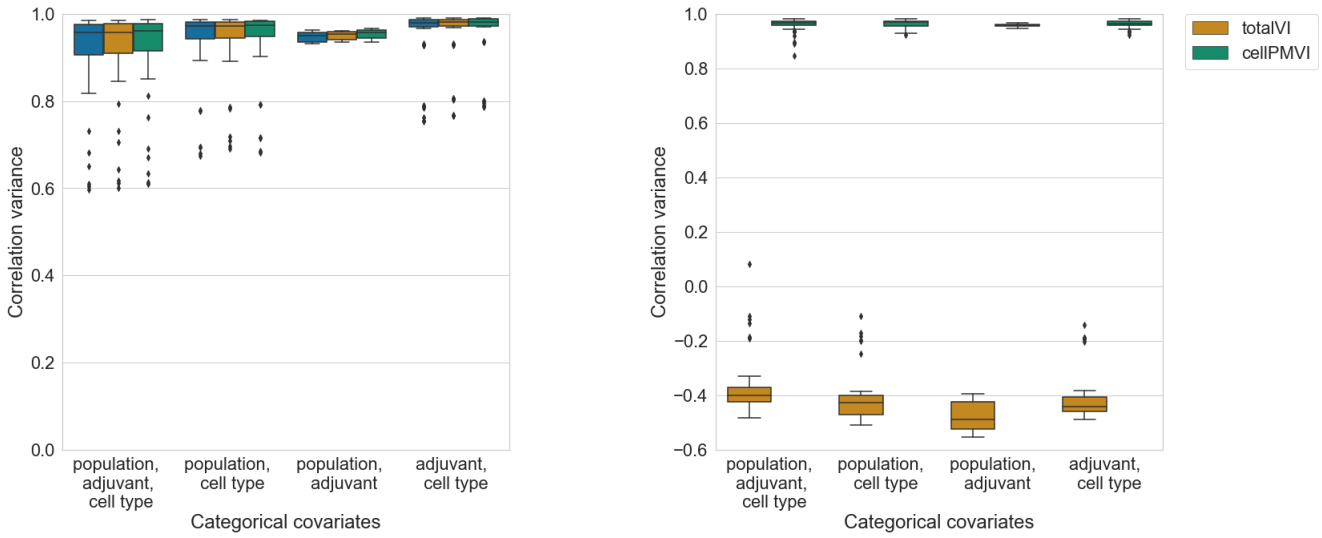
(a) scVI (b) TotalVI (c) MMVI

Figure C.4: Comparison of train and validation RL for all three models. All models have been trained under Training scenario 1 with three categorical covariates: population group, cell type and adjuvant.



(a) scVI (b) TotalVI (c) MMVI

Figure C.5: Comparison of train and validation KL divergence for all three models. All models have been trained under Training scenario 1 with three categorical covariates: population group, cell type and adjuvant.



(a) (b)

Figure C.6: **Boxplot showing the correlation variance for posterior predictive sampling for each model across four run setting (training scenario 1).** The correlation variance is calculated across all cells with the same population group, cell type and adjuvants (47 data points). The correlation means are clustered per run settings. Each run settings differs from each other in the number of covariates the model is conditioned on (Table B.3). Subfigure (a) shows the correlation mean for the RNA expression sampling and (b) for protein expression sampling.

Bibliography

[1] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518 (Apr. 2017), pp. 859–877. doi: 10.1080/01621459.2017.1285773. url: https://doi.org/10.10802F01621459.2017.1285773.

[2] Xi Chen et al. “InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets”. In: (June 2016).

[3] Carl Doersch. “Tutorial on Variational Autoencoders”. In: (2016). doi: 10.48550/ARXIV.1606.05908. url: https://arxiv.org/abs/1606.05908.

[4] Gökcen Eraslan et al. “Single-cell RNA-seq denoising using a deep count autoencoder”. In: *Nature Communications* 10 (Jan. 2019), p. 390. doi: 10.1038/s41467-018-07931-2.

[5] Allison Galassie and Andrew Link. “Proteomic contributions to our understanding of vaccine and immune responses”. In: *Proteomics. Clinical applications* 9 (Dec. 2015), pp. 972–989. doi: 10.1002/prca.201500054.

[6] Adam Gayoso et al. “A Python library for probabilistic analysis of single-cell omics data”. In: *Nature Biotechnology* (Feb. 2022). issn: 1546-1696. doi: 10.1038/s41587-021-01206-w. url: https://doi.org/10.1038/s41587-021-01206-w.

[7] Adam Gayoso et al. “Joint probabilistic modeling of single-cell multi-omic data with totalVI”. In: *Nature Methods* 18 (Mar. 2021), pp. 1–11. doi: 10.1038/s41592-020-01050-x.

[8] Andrew Gelman, Xiao-li Meng, and Hal Stern. “Posterior Predictive Assessment of Model Fitness Via Realized Discrepancies”. In: *Stat. Sin* 6 (Apr. 1997).

[9] Yuhan Hao et al. “Integrated analysis of multimodal single-cell data”. In: *Cell* 184.13 (2021), 3573–3587.e29. issn: 0092-8674. doi: https://doi.org/10.1016/j.cell.2021.04.048. url: https://www.sciencedirect.com/science/article/pii/S0092867421005833.

[10] Leon Hetzel et al. “Predicting single-cell perturbation responses for unseen drugs”. In: (Apr. 2022).

[11] Irina Higgins et al. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *ICLR*. 2017.

[12] Yuge Ji et al. “Machine learning for perturbational single-cell omics”. In: *Cell Systems* 12 (June 2021), pp. 522–537. doi: 10.1016/j.cels.2021.05.016.

[13] Victoria Jiang et al. “Performance of rotavirus vaccines in developed and developing countries”. In: *Human vaccines* 6 (July 2010), pp. 532–42. doi: 10.4161/hv.6.7.11278.

[14] Diederik Kingma and Max Welling. “An Introduction to Variational Autoencoders”. In: (June 2019).

[15] Diederik P Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: (2014). arXiv: 1312.6114 [stat.ML].

[16] Solomon Kullback and Richard Leibler. “On Information and Sufficiency”. In: *Annals of Mathematical Statistics* 22 (Mar. 1951), pp. 79–86. doi: 10.1214/aoms/1177729694.

[17] Guillaume Lample et al. “Fader Networks: Manipulating Images by Sliding Attributes”. In: (June 2017).

[18] Bissan Al-Lazikani, Udai Banerji, and Paul Workman. “Combinatorial drug therapy for cancer in the post-genomic era”. In: *Nature biotechnology* 30 (July 2012), pp. 679–92. doi: 10.1038/nbt.2284.

[19] Christian Ledig et al. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”. In: (July 2017), pp. 105–114. doi: 10.1109/CVPR.2017.19.

[20] Ivano Legnini et al. “Optogenetic perturbations of RNA expression in tissue space”. In: (Sept. 2021). doi: 10.1101/2021.09.26.461850.

[21] Romain Lopez et al. “Deep generative modeling for single-cell transcriptomics”. In: *Nature Methods* 15 (Dec. 2018). doi: 10.1038/s41592-018-0229-2.

[22] Mohammad Lotfollahi, F. Wolf, and Fabian Theis. “scGen predicts single-cell perturbation responses”. In: *Nature Methods* 16 (Aug. 2019), pp. 715–721. doi: 10.1038/s4159f2-019-0494-8.

[23] Mohammad Lotfollahi et al. “Compositional perturbation autoencoder for single-cell response modeling”. In: (2021). doi: 10.1101/2021.04.14.439903. url: https://doi.org/10.1101/2021.04.14.439903.

[24] Mohammad Lotfollahi et al. “Conditional out-of-sample generation for unpaired data using trVAE”. In: (Oct. 2019).

[25] Mohammad Lotfollahi et al. “Out-of-distribution prediction with disentangled representations for single-cell RNA sequencing data”. In: (Sept. 2021). doi: 10.1101/2021.09.01.458535.

[26] Samuel Marguerat and Jürg Bähler. “Coordinating genome expression with cell size”. In: *Trends in genetics : TIG* 28 (Aug. 2012), pp. 560–5. doi: 10.1016/j.tig.2012.07.003.

[27] Emile Mathieu et al. “Disentangling Disentanglement in Variational Autoencoders”. In: (2018). doi: 10.48550/ARXIV.1812.02833. url: https://arxiv.org/abs/1812.02833.

[28] Kodai Minoura et al. “A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data”. In: *Cell Reports Methods* 1 (Sept. 2021), p. 100071. doi: 10.1016/j.crmeth.2021.100071.

[29] Alberta Pasquale et al. “Vaccine Adjuvants: from 1920 to 2015 and Beyond”. In: *Vaccines* 3 (June 2015), pp. 320–343. doi: 10.3390/vaccines3020320.

[30] Bali Pulendran, Prabhu S A, and Derek O’Hagan. “Emerging concepts in the science of vaccine adjuvants”. In: *Nature Reviews Drug Discovery* 20 (Apr. 2021), pp. 1–22. doi: 10.1038/s41573-021-00163-y.

[31] Yuge Shi et al. “Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models”. In: (Nov. 2019).

[32] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. In: 28 (2015). Ed. by C. Cortes et al. url: https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[33] Valentine Svensson. “Droplet scRNA-seq is not zero-inflated”. In: *Nature Biotechnology* 38 (Jan. 2020), pp. 1–4. doi: 10.1038/s41587-019-0379-5.

[34] George Tucker et al. “Doubly Reparameterized Gradient Estimators for Monte Carlo Objectives”. In: (Oct. 2018).

[35] Miguel Vasco et al. “Leveraging hierarchy in multimodal generative models for effective cross-modality inference”. In: *Neural Networks* 146 (Nov. 2021). doi: 10.1016/j.neunet.2021.11.019.

[36] Isaac Virshup et al. “anndata: Annotated data”. In: *bioRxiv* (2021). doi: 10.1101/2021.12.16.473007. eprint: https://www.biorxiv.org/content/early/2021/12/19/2021.12.16.473007.full.pdf. url: https://www.biorxiv.org/content/early/2021/12/19/2021.12.16.473007.

[37] Luke Zappia and Fabian J. Theis. “Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape”. In: *bioRxiv* (2021). doi: 10.1101/2021.08.13.456196. eprint: https://www.biorxiv.org/content/early/2021/09/03/2021.08.13.456196.full.pdf. url: https://www.biorxiv.org/content/early/2021/09/03/2021.08.13.456196.