Joint probabilistic modeling of pseudobulk and single-cell transcriptomics enables accurate estimation of cell type composition

Simon Grouard [*1] **Khalil Ouardini** [*1] **Yann Rodriguez** [1] **Jean-Philippe Vert** [1] **Almudena Espin-Perez** [1]

Abstract

Bulk RNA sequencing provides an averaged gene expression profile of the numerous cells in a tissue sample, obscuring critical information about cellular heterogeneity. Computational deconvolution methods can estimate cell type proportions in bulk samples, but current approaches can lack precision in key scenarios due to simplistic statistical assumptions, limited modeling of cell-type heterogeneity and poor handling of rare populations. We present MixupVI, a deep generative model that learns representations of single-cell transcriptomic data and introduces a mixup-based regularization to enable reference-free deconvolution of bulk samples. Our method creates a latent representation with an additive property, where the representation of a pseudobulk sample corresponds to the weighted sum of its constituent cell types. We demonstrate how MixupVI enables accurate estimation of cell type proportions through benchmarking on pseudobulks simulated from a large immune single-cell atlas. To support reproducibility and foster progress in the field, we also release PyDeconv, a Python library that implements multiple state-of-the-art deconvolution algorithms and provides a comprehensive benchmark on simulated pseudobulk datasets.

1. Introduction

The cellular composition of tissues plays a critical role in their function, and alterations in this composition are associated with various diseases including cancers (Li, 2017; Rooney, 2015; Gentles & Newman, 2015; Mahmoud, 2011; Li, 2016), cardiovascular diseases (Parker, 2020), and neurodegenerative disorders (Lee, 2012; Poduri, 2012; Erickson, 2010; Rivière, 2012). Bulk RNA sequencing (RNA-seq) has been widely used to profile gene expression in tissue samples (Cancer Genome Atlas Research Network, 2013; Greenwood, 2020; GTEx Consortium, 2020), but this technology measures the average expression of all cells in a sample, masking the heterogeneity of different cell types. In contrast, single-cell RNA sequencing (scRNA-seq) provides gene expression profiles of individual cells, allowing direct characterization of cellular heterogeneity. However, scRNA-seq is more expensive, technically challenging, and often limited in the number of samples that can be processed compared to bulk RNA-seq.

Computational deconvolution methods aim to bridge this gap by estimating cell type proportions in bulk RNA-seq samples using reference profiles derived from single-cell data (Cobos, 2020). Traditional approaches rely on linear regression models that express bulk samples as linear combinations of these reference profiles, often referred to as a signature matrix (Erdmann-Pham, 2021; Dong, 2021; Tsoucas, 2019; Newman, 2015). While these approaches have shown promise, they face several limitations. In particular, they often fail to account for the complex differences between single-cell and bulk RNA-seq data, biological variations within cell types or due to simplistic statistical assumptions (e.g modeling transcriptomic data with additive Gaussian noise).

Deep generative models have recently emerged as powerful tools for analyzing single-cell data (Donno, 2023; Cui & Wang, 2024), enabling the learning of low-dimensional representations that capture biological variation while accounting for technical factors (Xu, 2021). Models such as single-cell Variational Inference (scVI) have demonstrated consistent performance in tasks such as clustering, batch correction, and imputation of single-cell data (Lopez, 2018). However,

[*]Equal contribution [1]Owkin, Paris, France. Correspondence to: Simon Grouard <firstname.secondname@owkin.com>, Almudena Espin-Perez <firstname.secondname@owkin.com>.

reconciling the expressive nature of these non-linear deep generative models with the linearity constraints imposed by the deconvolution problem remains a challenge.

To address this gap, we introduce MixupVI, a deep generative model for bulk RNA-seq deconvolution that leverages single-cell representations. Our approach constrains the latent space of a variational autoencoder to enforce an additive property, where the representation of a bulk sample can be approximately expressed as a weighted sum of cell-type specific latent representations. This enables accurate estimation of cell type proportions in pseudobulk samples while maintaining the benefits of deep generative modeling.

1.1. Conventional assumptions in deconvolution approaches

Bulk transcriptomics deconvolution aims to estimate the proportions of distinct cell types within a heterogeneous tissue sample based on its aggregated gene expression profiles. Traditional deconvolution methods build on Least Squares approaches. These techniques rely on the critical assumption of linearity, which states that the expression level of each gene in a bulk sample is a linear combination of the expression levels from the constituent cell types, weighted by their proportions. Formally, let G denote the number of genes and C the number of cell types in the deconvolution setting. The gene expression profile of a bulk sample b can be modeled as:

$$y_b = Sp + \epsilon\,, \tag{1}$$

where:

- $y_b \in \mathbb{N}^G$ is the observed bulk gene expression vector,
- $S \in \mathbb{R}^{G\times C}$ is the cell-type specific reference (signature) matrix,
- $p \in \mathbb{R}^C$ is the vector of cell type proportions (with $\sum_c p_c = 1$),
- $\epsilon \sim \mathcal{N}(0, \sigma^2 I_G)$ is Gaussian noise.

The assumption of linearity holds reasonably well in practice, especially when enough cells are present in a sample and the deconvolution granularity and observed proportions are appropriate. However, this assumption can break down in key scenarios, leading to systematic biases in cell type proportion estimates:

- Log-transformation is commonly used in transcriptomics to stabilize variance and approximate Gaussianity to rely on Least Squares approaches. However, it distorts the linear relationships between mixed cell populations in bulk data. This results in curved trajectories in PCA space (Supplementary Figure S1), leading to systematic errors in proportion estimates - especially for intermediate mixtures. As a result, some deconvolution methods advice to keep the data in linear space (Zhong & Liu, 2011; Newman, 2015). The data thus no longer follows a Gaussian distribution and instead typically exhibits over-dispersed negative binomial behavior with prominent zero-inflation.
- Low abundance. When a cell type is barely present within a sample, its gene expression signal may be too weak or noisy to resemble the average profile captured in the reference signature matrix.
- Low granularity. Broad cell type definitions - such as grouping all lymphoid cells together - fail to capture intra-lineage heterogeneity. While $CD4^+$ T cells, $CD8^+$ T cells and Natural Killer cells originate from the common lymphoid progenitor, they exhibit distinct gene expression profiles and unique markers (Hidalgo, 2008). Overlooking these differences can increase variance and reduce deconvolution accuracy (Supplementary Figure S2).

As a result, traditional linear deconvolution methods often struggle to model the biological variability required for the linearity assumption to hold across diverse samples and contexts.

Even when the linearity assumption is valid, several practical challenges remain regarding the creation of the signature matrix. Collinearity in the signature matrix can severely hinder the stability of linear regression models, making the system ill-conditioned. To mitigate this, one must carefully select genes that are both informative and non-redundant, often removing noisy or highly correlated genes. These constraints may lead to the exclusion of biologically relevant genes (Wang, 2019). Furthermore, many linear methods implicitly favor genes with higher mean expression, potentially overlooking genes that are highly discriminatory between cell types but expressed at lower levels (Nguyen, 2024; Tsoucas, 2019). This can limit the resolution and accuracy of the deconvolution. As a result, while constructing a robust and high-quality signature matrix is common practice, it may not always be sufficient due to these inherent limitations.

1.2. A latent space approach to deconvolution

Our method directly addresses the core limitations of traditional deconvolution by embedding gene expression data into a latent space designed to approximate ideal deconvolution properties:

- Gaussian-like latent structure.
- Linearity between cell-type specific profiles and mixtures of cells.
- Non-linear mappings that capture subtle, non-linear gene - cell-type dependencies across diverse contexts and retain informative signals from rare or noisy cell types.

To achieve this, we leverage variational autoencoders (VAEs) (Kingma, 2013) to learn a latent representation of gene expression that satisfies both statistical assumptions and biological constraints. Building on single-cell deep learning frameworks such as scVI, we introduce a critical modification to enforce linearity in the latent space - a property not naturally satisfied by standard VAEs.

The key novelty in our approach is the incorporation of a mixup penalty (Beckham, 2019; Berthelot & Raffel, 2018; Khan, 2022; Verma & Lamb, 2018) in the evidence lower bound (ELBO) objective that explicitly optimizes the VAE to jointly model single-cell profiles and their mixtures (pseudobulks) in the latent space. This results in:

- A non-linear encoder that maps both individual cells and mixtures (e.g., bulk RNA-seq, spatial transcriptomics spots) into a shared Gaussian latent space.
- Linearly additive structure in the latent space, where mixture representations reflect their underlying cell type proportions.
- A shared decoder that accurately reconstructs both individual and mixed expression profiles, while modeling their distinct statistical characteristics - such as the zero-inflation commonly observed in single-cell data but not in mixtures.

By training on both single-cell data and systematically generated pseudobulk mixtures, our model learns to "fill the gaps" in the latent space, ensuring that mixtures fall precisely where linear interpolation would predict. This contrasts with standard VAE approaches that may learn efficient representations of individual cells but provide no guarantees about the behavior of mixtures in the latent space.

Our framework thus creates a unified representation where the same latent variables can represent both individual cells and mixtures of cells while preserving the critical property of linear additivity. This enables robust and accurate deconvolution across varying levels of cell type heterogeneity and proportion.

2. The MixupVI probabilistic model

For a given single-cell i, scRNA-seq experiments output a set of read counts representing the gene expression levels across G genes. Let's denote the gene expression profile of a single-cell as $x_i \in \mathbb{N}^G$. Similarly, for a bulk sample b, bulk RNA-seq provides a gene expression profile $y_b \in \mathbb{N}^G$. We assume each cell i is annotated with a cell type label c_i from a set of C distinct cell types.

The key insight of our approach is that bulk samples can be represented as a mixture of cells from different cell types, and this mixture relationship should be preserved in the latent space. To achieve this, we design a latent space with an additive property: the representation of the sum of a set of single-cells is approximately the sum of their individual representations.

2.1. Generative model for single-cell transcriptomics

Building upon the scVI framework (Gayoso et al., 2022), we first define a generative model for single-cell data:

$$
\begin{aligned}
&z_i \sim \mathcal{N}(0, I_d)\,, \\
&x_i | z_i \sim \text{ZINB}\left(\mu_i = f_\theta(z_i),\ \pi_i = g_\theta(z_i),\ \psi_g\right)\,,
\end{aligned}
$$

where $z_i \in \mathbb{R}^d$ is a d-dimensional latent variable representing the cell state, I_d is the d-dimensional identity matrix, ZINB denotes the zero-inflated negative binomial distribution, and f_θ, g_θ are neural networks parameterized by θ, which map the latent variable to the expected gene counts μ_i and the expected dropout rate π_i, respectively. ψ_g is a gene-specific

dispersion parameter, learned directly for each gene and shared across cells. This formulation captures the over-dispersed and zero-inflated nature of scRNA-seq data.

As with standard VAEs, the marginal likelihood $p(x_i)$ is intractable, meaning we cannot directly compute the posterior distribution $p_\theta(z_i|x_i)$. Instead, we learn an approximate posterior $q_\varphi(z_i|x_i)$ using variational inference (Kingma & Welling, 2019), where φ denotes the parameters of an inference network. We then optimize the evidence lower bound (ELBO) of $\log p_\theta(x_{1:N})$ with respect to the variational parameters φ and the generative model parameters θ.

2.2. Mixup regularization: enforcing linearity in the latent space

The key novelty in MixupVI is the introduction of a linearity constraint in the latent space, called the mixup loss (Carratino, 2020).

Given a batch of N cells and C cell types, let's denote:

- x_i as the i-th cell in the batch.
- z_i as the latent representation of cell x_i .
- n_{cpp} defines the number of single-cells to aggregate when generating a single pseudobulk sample. By adjusting this hyperparameter, one can flexibly simulate different experimental resolutions: larger values mimic bulk RNA-seq conditions, while smaller values emulate spot-based spatial transcriptomics where each spot contains fewer cells. This allows for benchmarking or modeling across a spectrum of transcriptomic data types.
- $\beta = (\beta_1, ..., \beta_C)$ as the proportions sampled from a Dirichlet distribution over the $C-1$ dimensional simplex (i.e., non-negative values that sum to one). This sampling step enables the over-representation of rare cell types, which are often under-represented in classical deconvolution methods due to their low abundance, resulting in insufficient signal. The parametrization of this Dirichlet distribution is detailed in Supplementary Section B.
- $\mathcal{I}$ as the set of sampled cell indices for creating a pseudobulk, determined by the proportions β and the number of cells per pseudobulk n_{cpp} .

As depicted in Figure 1, the process involves:

- Sampling proportions β from a Dirichlet distribution.
- Sampling cell indices $\mathcal{I}$ from the batch to match the determined proportions given by β and n_{cpp} with $|\mathcal{I}| = n_{cpp}$.
- Creating a pseudobulk $x_\mathcal{I}$ as a linear combination of the cells, by averaging the sampled cells: $x_\mathcal{I} = \sum_{i\in\mathcal{I}} \frac{1}{|\mathcal{I}|} x_i$.
- Sampling the latent representations $z_i \sim q_\varphi(\cdot \mid x_i)$ for each cell and $z_\mathcal{I} \sim q_\varphi(\cdot \mid x_\mathcal{I})$ for the pseudobulk, using the encoder of the model.
- We encourage the latent representation of this linear combination to be close to the linear combination of the individual latent representations with a L2 norm. This L2 regularization is particularly meaningful in the latent space, given that the data is modeled as following a Gaussian distribution. Minimizing this penalty is thus equivalent to maximizing the likelihood estimate of the latent variables.

Definition 2.1. For one given collection of cell indices $\mathcal{I}$, the mixup loss can be formulated as follows:

$$\text{Mix}(\theta, \varphi | x_{i\in\mathcal{I}}) = \mathbb{E}_{\substack{z_\mathcal{I}\sim q_\varphi(\cdot|x_\mathcal{I}),\\ z_i\sim q_\varphi(\cdot|x_i)}} \left\| z_\mathcal{I} - \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} z_i \right\|^2 .$$

Definition 2.2. To minimize the expected risk over the pseudobulk distribution, we define the mixup loss for the full batch of cells with:

$$\text{Mix}(\theta, \varphi | x_1, ..., x_N) = \mathbb{E}_{\mathcal{I}\sim p_{pseudobulk}} \text{Mix}(\theta, \varphi | x_{i\in\mathcal{I}}) .$$

with $p_{pseudobulk}$ the distribution created from the multi-step stochastic process described above. Collections of indices are drawn according to this distribution. We approximate the resulting expectation using Monte Carlo sampling:

$$\text{Mix}(\theta, \varphi | x_1, ..., x_N) \approx \frac{1}{M}\sum_{m=1}^{M} \text{Mix}(\theta, \varphi | x_{i\in\mathcal{I}^{(m)}}) ,$$

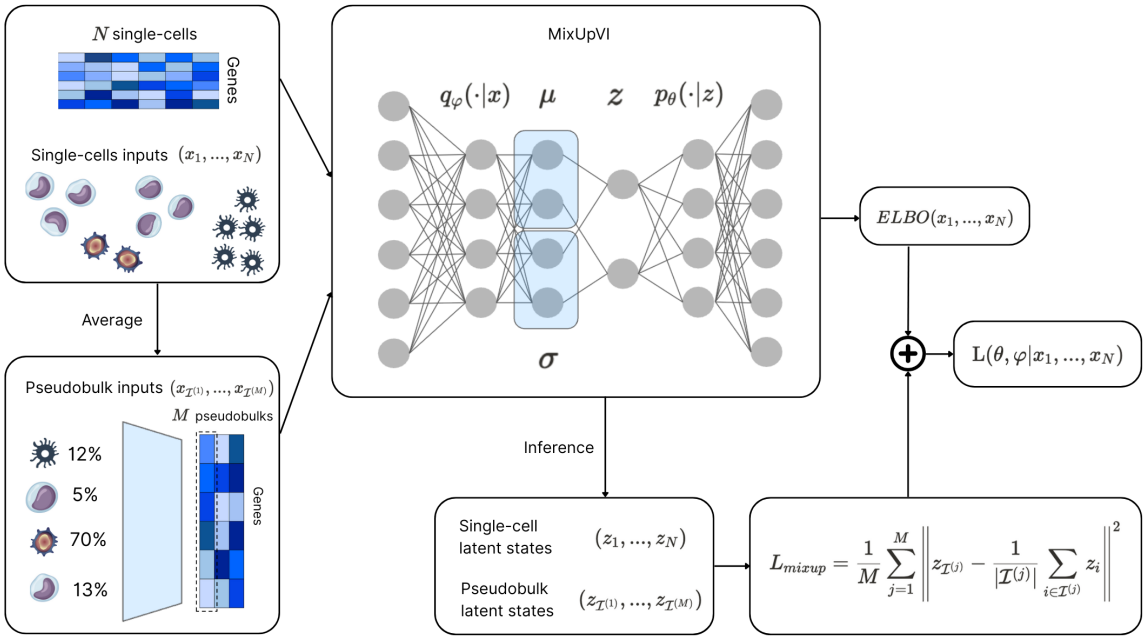


Figure 1. Schematic overview for the MixupVI forward process. Pseudobulks are first created from the batch of single-cells by sampling proportions independently from a Dirichlet distribution and sampling cell indices from the batch to match these proportions. Both single-cells and pseudobulks are then passed to the VAE. Their latent representations are sampled from the encoder to compute the mixup loss. Finally, the mixup loss is added to the ELBO.

where $\mathcal{I}^{(1)}, \ldots, \mathcal{I}^{(M)}$ are sampled i.i.d. following $p_{pseudobulk}$. This mixup loss term is added as a regularization term to the ELBO:

$$\mathrm{L}(\theta, \varphi|x_1, ..., x_N) = ELBO + \mathrm{Mix}(\theta, \varphi|x_1, ..., x_N)\,. \tag{2}$$

2.3. Cell type proportions inference

Given a pseudobulk sample y_b, we model it as a mixture of cell types with proportions $p_b = (p_b^1, ..., p_b^C)$:

$$y_b|z_b \sim \text{NegativeBinomial}\left(f_\theta(z_b),\ \psi_g\right)\ ,$$

where z_b the inferred latent state for sample b.

As depicted in Figure 2, we encode the bulk sample into the latent space $z_b \sim q_\varphi(\cdot|y_b)$. We then estimate the cell type proportions $p_b \in \mathbb{R}^c$ by solving the following constrained linear system (and then dividing the estimated coefficients by their sum to get proportions summing up to 1):

$$\begin{cases} z_b = \sum_{c=1}^{C} p_b^c z^c \\ p_b^c \geq 0 \end{cases}\ , \tag{3}$$

where z^c is the average latent representation for cell type c obtained from the single-cell data. This average latent representation serves as the equivalent of a signature matrix in latent space — which we refer to as the latent signature matrix. Unlike classical approaches, it bypasses the need to manually curate a signature matrix, a process that is prone to issues like collinearity and bias toward highly expressed genes.

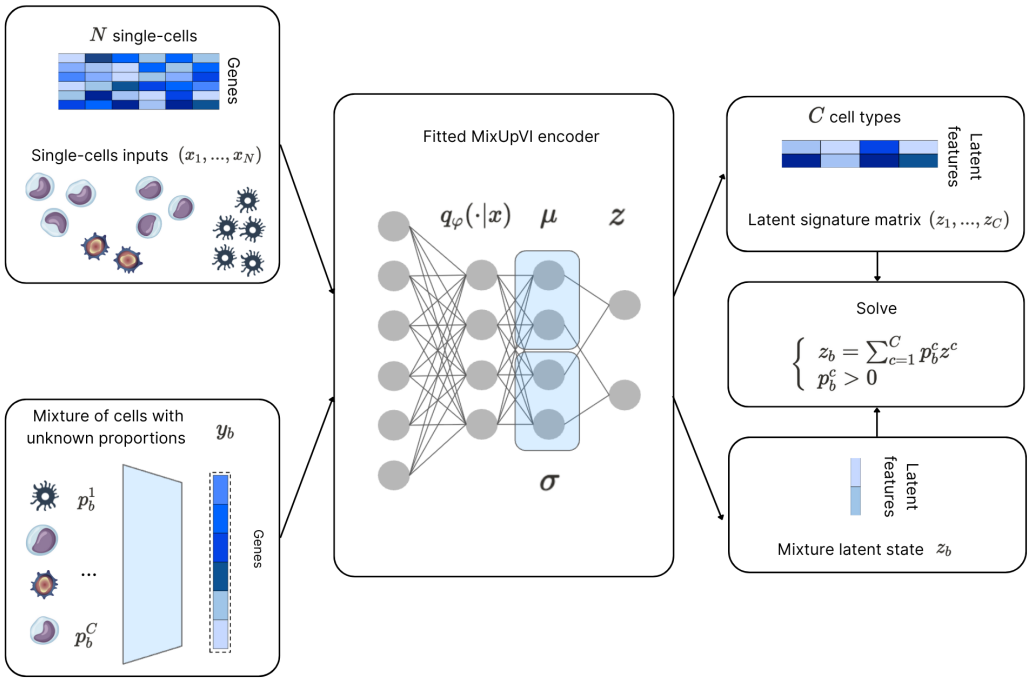


Figure 2. Schematic overview for the cell type proportions inference with MixupVI. The single-cell training dataset is passed to MixupVI. Aggregating the latent representation of the individual cells per cell type allows the creation of the latent signature matrix. The evaluation pseudobulk is passed to MixupVI to infer its latent representation. Finally, the unknown proportions of the pseudobulk sample are inferred with the non-negative least squares approach.

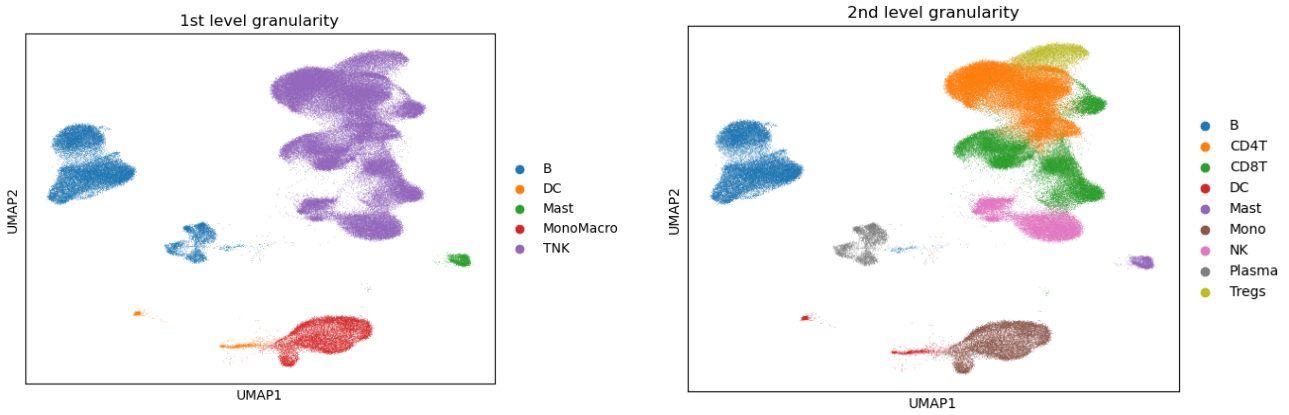
3. Results

3.1. Datasets and pseudobulk simulation

Our analysis was conducted using the Cross-tissue immune (CTI) single-cell dataset (Conde et al., 2022). This analysis was performed on the immune compartment of 16 tissues from 12 adult donors by single-cell RNA sequencing, creating a dataset of approximately 360,000 cells. For each granularity, we divided the dataset between train and test (50/50 split), using the train set to:

- Build signature matrices (Hao, 2021).
  We performed differential expression analysis (DEA) between cell types in the single-cell RNA-seq dataset using a non-parametric Wilcoxon rank-sum test on a gene-by-gene basis. Genes were tested only if expressed in at least 1% of cells in either group (cell type of interest vs rest). Differentially expressed genes were defined as those with an adjusted p-value below 0.05 and an absolute log fold change greater than 0.25. Then, minimization of the condition number was conducted using the kappa function. Values reported in the signature matrix correspond to mean gene expression of the significantly differentially expressed genes.
- Fit deep learning algorithms.
  MixupVI as well as other deep baseline methods (Chen & Wang, 2022; Menden, 2020) included in the benchmark are trained on this set of single-cells.

To ensure robust evaluation, we use the test set to create synthetic pseudobulk datasets with two levels of difficulty defined by cell type granularity (5 and 9 cell types). Pseudobulk samples with known cell type proportions were generated by aggregating single-cells from the test set of the CTI atlas, using proportions given by sampling from a Dirichlet distribution



(a) 1st granularity level

(b) 2nd granularity level

Figure 3. UMAP plots of embedded single-cell gene expression profiles with scVI. (a) First granularity level: B cells, Dendritic cells, Mast cells, Monocytes & Macrophages, T & Natural Killer cells. (b) Second granularity level: B cells, Plasma cells, Dendritic cells, Mast cells, Monocytes, CD8 T cells, CD4 T cells, T regulatory cells, Natural Killer cells.

as explained in Supplementary Section B. The breakdown of the cell types for each granularity is shown in Figure 3. We design two types of benchmarks. The first keeps the number of cells per pseudobulk fixed at 100, intentionally inducing under-representation of certain cell types. This choice is motivated by our observation that using larger pseudobulks (e.g., 200–300 cells or more) results in more stable cell type proportions, making the deconvolution task less challenging. The second benchmark varies the number of cells per pseudobulk from 10 to 1000, enabling a comparison of methods across a range of resolution levels.

3.2. PyDeconv: a library for deconvolution methods benchmarking

Our comparative analysis included linear methods: Ordinary Least Squares (OLS), Non-Negative Least Squares (NNLS), Robust Linear Regression (RLR), Nu-Support Vector Regression (NuSVR) which is the core algorithm of CIBERSORT (Newman, 2015), Weighted Non-Negative Least Squares (WNNLS) and Dampened Weighted Least Squares (DWLS) (Tsoucas, 2019). We also included two deep learning methods: TAPE (Chen & Wang, 2022) and Scaden (Menden, 2020). More information about the methods can be found in Supplementary Section C.

Current deconvolution methods are implemented in various programming languages, leading to fragmented workflows. While unified benchmarking is possible using solutions like multi-container Docker pipelines (Nguyen, 2024), such setups can add overhead and require additional maintenance. To streamline this process, we introduce PyDeconv, a Python package that reimplements a set of representative methods along with a benchmarking pipeline using various pseudobulk simulations. The aim of the package is to provide unified support for all major deconvolution techniques found in the literature, thereby simplifying benchmarking efforts and enhancing method robustness through an open-source and standardized framework.

3.3. Linearity constraint validation

We first evaluated the effectiveness of our linearity constraint in the latent space. Using the CTI dataset, we trained our model and compared it to the standard scVI model (without the mixup linearity constraint). For validation, we generated synthetic bulk samples (pseudobulks) by combining single-cells from a held out set, in proportions sampled from a Dirichlet distribution and assessed how accurately the models preserved these proportions in the latent space.

We monitored the Evidence Lower Bound (ELBO) on a held out set to ensure that our linearity constraint did not compromise the model's ability to reconstruct gene expression profiles. To assess the impact of the mixup loss, we measured the Pearson correlation between pseudobulks of the encoded single-cells and the encodings of the pseudobulks created in the input space, that we call mixup correlation ρ (Supplementary Section D). Additionally, we computed the Pearson correlation between the estimated cell-type proportions within the latent space and the ground truth proportions for pseudobulk simulated from the held-out set, that we call deconvolution correlation d (Supplementary Section D).

Table 1. Monitoring of validation metrics: ELBO loss, mixup correlation ρ, deconvolution correlation d at the end of the training.

METRIC	SCVI	MIXUPVI
ELBO LOSS	641.16	641.37
MIXUP CORRELATION ρ	0.32	0.94
DECONVOLUTION CORRELATION d	0.45	0.88

Both models exhibited comparable ELBO convergence patterns during training (Supplementary Figure S3). The final values of key metrics on the held out set are reported in Table 1. During training, scVI showed a decrease in mixup correlation ρ resulting in an average correlation of 0.32 on the held out set. In contrast, MixupVI maintained a high correlation during training resulting in an average of 0.94 on the held out set.

Furthermore, these correlations in the latent space translate to deconvolution accuracy of synthetic pseudobulks generated from the held out set. scVI demonstrated poor deconvolution performance throughout training, while MixupVI improved its deconvolution capabilities: by the end of training, MixupVI achieved a deconvolution correlation d close to 0.9, twice as much as scVI.

These results indicate that MixupVI maintains reconstruction quality while gaining the critical linearity property necessary for accurate deconvolution.

3.4. Comprehensive benchmarking of deconvolution methods

The metrics presented in this part are the deconvolution correlation d and Mean-Squared Error (deconvolution MSE) between the estimated cell-type proportions of each deconvolution method and the ground truth proportions for pseudobulk simulated from the held-out set (Supplementary Section D).

First-level granularity (5 cell types)

At the coarse granularity level distinguishing major immune cell compartments, MixupVI demonstrated competitive performance with other best performing methods (RLR, WNNLS, Scaden) with a median Pearson correlation of 0.973 (Figure 4.a.). Naive linear methods such as NNLS and OLS, which do not incorporate weighting, tend to underperform. This could be due to collinearity issues in the signature matrix as well as the limitations discussed in the introduction: broad cell type definitions often exhibit high intra-cell type variability, causing the estimated cell type profiles to deviate from those in the signature matrix. Notably, these methods show high Pearson correlation variability and occasionally produced negative correlation values, highlighting their instability for this task.

RLR and MixupVI also achieved the lowest MSE at 0.0011 and 0.0016 respectively, outperforming all other competing methods.

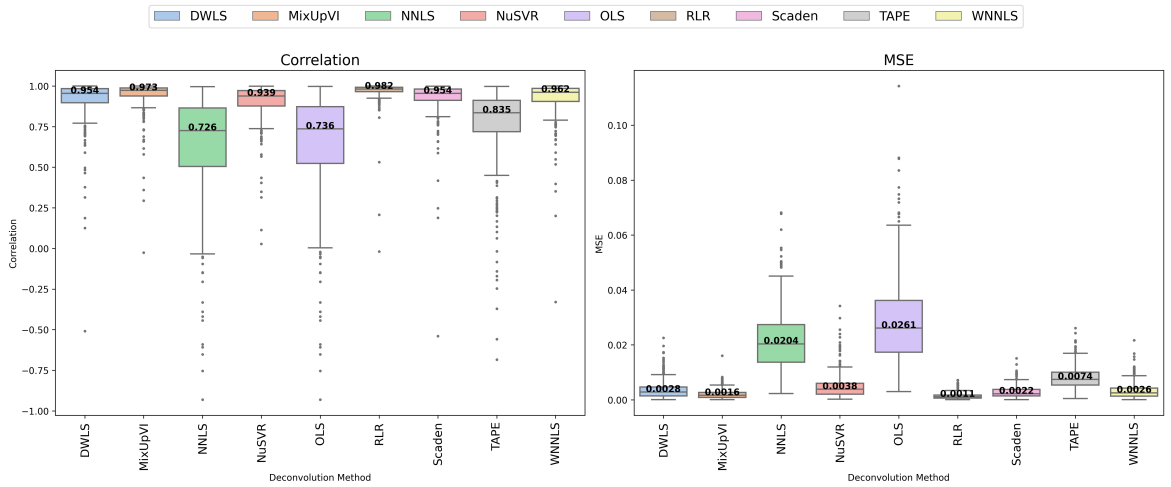
Second-level granularity (9 cell types)

When the deconvolution task difficulty was increased to distinguish finer cell subtypes (including separation of $CD4^+$ T cells, $CD8^+$ T cells, T regulatory cells, and Natural Killer cells), MixupVI maintained robust performance and low variability, with a median correlation of 0.912 (Figure 4.b.). This represented a marked improvement of at least 2 points over all competing methods.

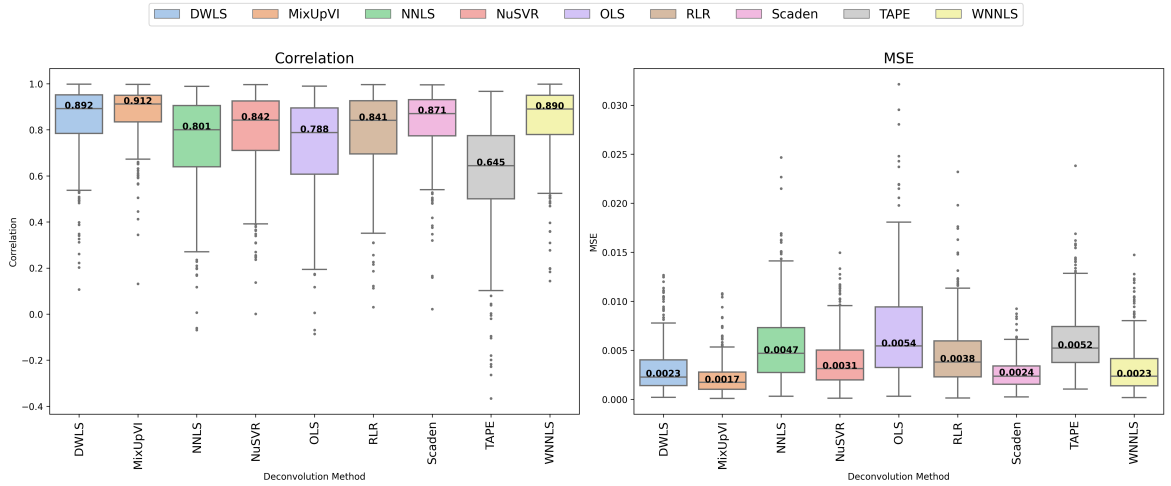
The MSE results further confirmed MixupVI's advantage, with a median error of 0.0017 compared to 0.0024 for Scaden and 0.0023 for DWLS. The performance gap between MixupVI and other methods widened at this higher granularity level, demonstrating our method's particular advantage in more challenging deconvolution scenarios.

Varying number of cells per pseudobulk

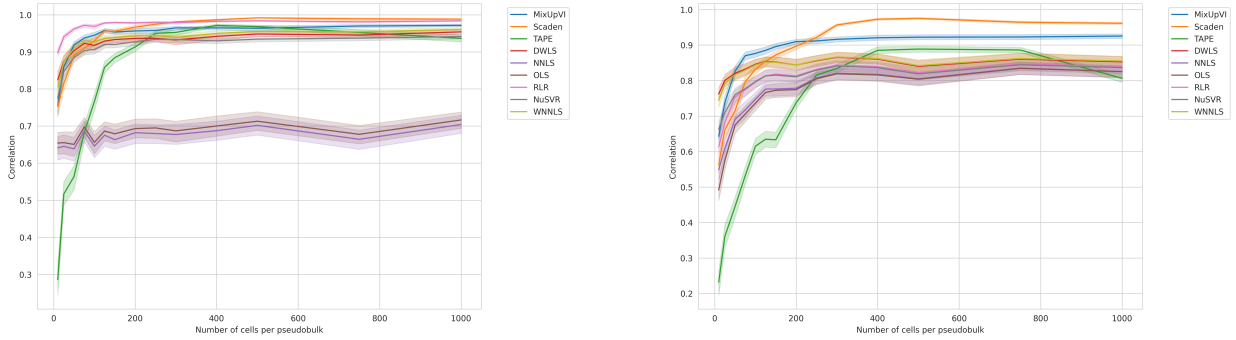
In addition to our primary benchmark results, we investigated how deconvolution performance varies with the number of cells in each pseudobulk sample. This analysis provides important insights into method robustness across varying input data depths.



a.



b.



c. *d.*

Figure 4. Deconvolution results. *a.* 1st granularity results. Pearson and MSE of predicted vs groundtruth proportions (400 pseudobulk samples / 100 cells per pseudobulk). *b.* Same metrics for 2nd granularity. *c.* Trend of the deconvolution results with regards to the number of cells samples for each pseudobulk simulation for the 1st granularity. The y-axis shows the average Pearson correlation and on the x-axis number of cells per pseudobulk. *d.* Same metrics for 2nd granularity.

At the first-level granularity (Figure 4.c.), besides OLS, NNLS and TAPE, most deconvolution methods are competitive, showing similar trends with increasing performance across sample sizes, until it plateaus around 100 cells. In higher sample size regime (> 250) Scaden outperforms other competing methods, which is likely due to Scaden's higher default sample size of 500 used in the simulations during training. In contrast, NNLS and OLS performs poorly across sample sizes, highlighting the limitations of classical methods when cell type definitions are overly broad, as averaging across heterogeneous populations can obscure important expression differences and degrade deconvolution accuracy.

At the second level of granularity (Figure 4.d.), MixupVI is outperformed by weighted linear methods when the number of cells is very low (< 25). However, for sample sizes between 25 and 200 cells, MixupVI consistently outperforms all other methods, highlighting its robustness to input variability, data sparsity, and the under-representation of rare cell types. Similarly to the 1st granularity experiments, we also observe that the the performance of Scaden steadily increases until outperforming other methods in the high cell-count regime (> 250). These findings underscore MixupVI's robustness to input data diversity and sparsity — an important advantage in real-world settings where high-depth bulk RNA-seq data may be difficult to obtain or where only limited samples are available. This robustness also suggests a potential benefit for spatial transcriptomics deconvolution, where each spot captures a limited number of cells. The consistent performance advantage across varying granularities and sample sizes demonstrates the broad applicability of our approach to diverse deconvolution scenarios.

4. Discussion

4.1. Impact of the latent size

MixupVI is designed to be easily adaptable to new datasets, data modalities and levels of granularity. In Supplementary Section E, detailed explanations of every benchmark, training procedure, and model hyperparameters are provided. However, among the various model hyperparameters, the latent space size is particularly crucial, requiring careful tuning.

As shown in (1), the signature matrix used with Non-Negative Least Squares (NNLS) is of size $(n_{genes}, n_{cell_types})$. Similarly, in the case of MixupVI, the latent signature matrix used with NNLS is of size $(n_{latent_size}, n_{cell_types})$. Thus, to avoid high-dimensionality issues, the latent space size should be at least higher than the number of cell types. Plus, it should be chosen to remove noise or collinearity of gene expressions that could impede fitting. Classical methods achieve this by carefully constructing the signature matrix. MixupVI, on the other hand, embeds gene expressions in the latent space to reduce noise and collinearity. Finally, the latent space should be sufficiently large to allow flexible shaping of representations through complex non-linear mappings, as the mixup loss introduces structural constraints that limit representational freedom.

The impact of the latent size is apparent in Figure 5, showing deconvolution correlation d and deconvolution MSE during training, on synthetic pseudobulks generated from the validation set. On the 9 cell types granularity, the performance improves progressively with a larger dimensionality for both metrics: a dimensionality of 30 yields improved performance compared to 20 and 10. We also notice that performance plateaus with this parameter - for instance when the latent space is increased to 100, the MSE plateaus, and even worsens for the Pearson correlation.

4.2. Conclusion

We have presented MixupVI, a deep generative model that learns probabilistic representations of single-cell RNA-seq data with a linearity constraint enabling accurate deconvolution of mixtures of cell types. By enforcing an additive property in the latent space, our model effectively bridges the gap between the expressivity of deep generative models and the linearity requirements of deconvolution.

MixupVI offers several key advantages over existing deconvolution methods. First, it constructs a latent Gaussian space in which samples are naturally expressed as mixtures of constituent cell types, aligning with the assumptions underlying deconvolution. Second, it models biological variability within cell types via its latent representation, enhancing accuracy - especially when distinguishing closely related types. Third, its probabilistic framework yields uncertainty estimates for both cell type proportions and reconstructed gene expression profiles, enabling more informed interpretation. Finally, it eliminates the need for manually curated reference profiles by learning a latent representation directly from the data, avoiding common pitfalls of signature matrix construction. Given its reliance on a latent space, however, the dimensionality of this space becomes an important design choice that should be tailored to each deconvolution task.

Future work will focus on extending MixupVI to handle more complex scenarios, such as higher granularities, deconvolution

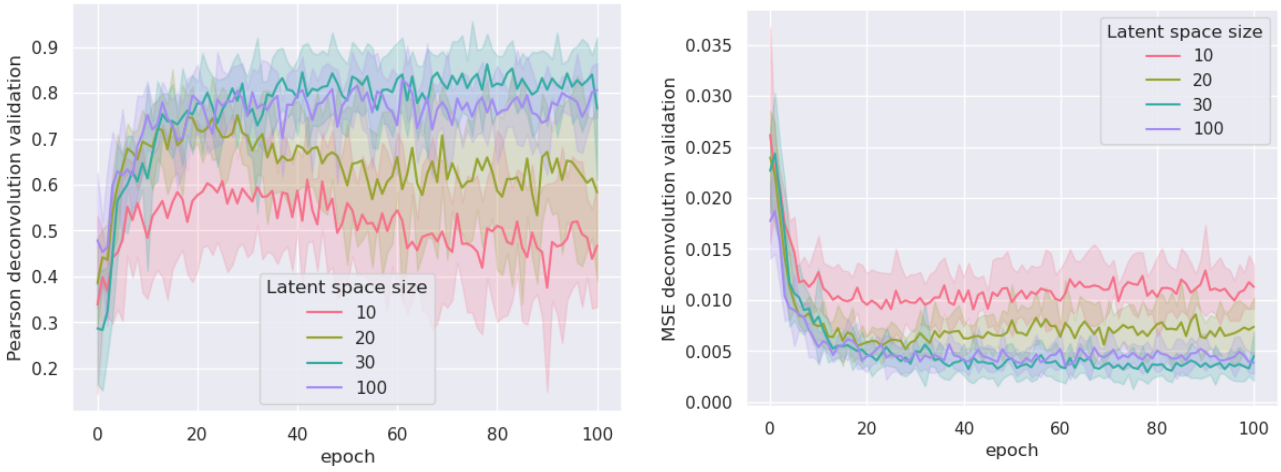


Figure 5. Example of latent space size tuning for the 2nd level granularity (9 cell types). The deconvolution correlation d and deconvolution MSE are computed at every batch of each epoch on a set of pseudobulks simulated from the validation set. Error bands represent the standard deviation across 5 random seeds. Intermediate latent sizes (between 30 and 100) are omitted for clarity.

across different tissues or disease states, and adapting MixupVI for real-world omics data settings. In particular, modeling pseudobulks with a small number of cells enables MixupVI to be applied to spot-based spatial transcriptomics, while using larger pseudobulks makes it well-suited for bulk RNA-seq paired with flow cytometry data. Building on its generative foundation, we also plan to extend the model to the more ambitious task of reconstructing cell-type specific gene expression profiles.

Software and Data

All the referenced methods, the pseudobulk simulations and the benchmarking code were implemented in the python library PyDeconv github.com/owkin/pydeconv .

The Cross-Tissue Immune atlas was dowloaded from www.tissueimmunecellatlas.org.

Acknowledgements

We would like to express our deepest gratitude to Sabrina Carpentier, whose optimistic guidance and insightful feedback supported the project throughout its development, especially regarding data generation processes. We also thank Ulysse Marteau for his initial mathematical formalization of the deconvolution setting and mixup, and Anaïs Chossegros for her help in designing the main model figures. Finally, we warmly thank Alessandro Pranzo, who has now enthusiastically joined us to further improve the model and pipelines. The best is yet to come.

Impact Statement

This paper presents work whose goal is to advance the fields of Machine Learning and Biology. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Beckham, C. On adversarial mixup resynthesis. March 2019.

Berthelot, D. and Raffel, C. Understanding and improving interpolation in autoencoders via an adversarial regularizer. July 2018.

Cancer Genome Atlas Research Network. The cancer genome atlas Pan-Cancer analysis project. *Nat Genet*, 45(10):

1113–1120, October 2013.

Carratino, L. On mixup regularization. June 2020.

Chen, Y. and Wang, Y. Deep autoencoder for interpretable tissue-adaptive deconvolution and cell-type-specific gene analysis. *Nat Commun*, 13(1):6735, November 2022.

Cobos, F. A. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nat Commun*, 11(1):5650, November 2020.

Conde, C. D., Xu, C., and et al., L. B. J. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594):eabl5197, May 2022.

Cui, H. and Wang, C. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nat Methods*, 21(8):1470–1480, February 2024.

Dong, M. SCDC: bulk gene expression deconvolution by multiple single-cell RNA sequencing references. *Brief Bioinform*, 22(1):416–427, January 2021.

Donno, C. D. Population-level integration of single-cell datasets enables multi-scale analysis across samples. *Nat Methods*, 20(11):1683–1692, October 2023.

Erdmann-Pham, D. D. Likelihood-based deconvolution of bulk gene expression data using single-cell references. *Genome Res*, 31(10):1794–1806, July 2021.

Erickson, R. P. Somatic gene mutation and human disease other than cancer: an update. *Mutat Res*, 705(2):96–106, April 2010.

Gayoso, A., Lopez, R., and Xing, G. A python library for probabilistic analysis of single-cell omics data. *Nat Biotechnol*, 40(2):163–166, February 2022.

Gentles, A. J. and Newman, A. M. The prognostic landscape of genes and infiltrating immune cells across human cancers. *Nat Med*, 21(8):938–945, July 2015.

Greenwood, A. K. The AD knowledge portal: A repository for Multi-Omic data on alzheimer's disease and aging. *Curr Protoc Hum Genet*, 108(1):e105, December 2020.

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

Hao, Y. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, May 2021.

Hidalgo, L. G. The transcriptome of human cytotoxic T cells: similarities and disparities among allostimulated CD4(+) CTL, CD8(+) CTL and NK cells. *Am J Transplant*, 8(3):627–636, March 2008.

Khan, A. Adversarial robustness of VAEs through the lens of local geometry. August 2022.

Kingma, D. P. Auto-Encoding variational bayes. December 2013.

Kingma, D. P. and Welling, M. An introduction to variational autoencoders. June 2019.

Lee, J. H. De novo somatic mutations in components of the PI3K-AKT3-mTOR pathway cause hemimegalencephaly. *Nat Genet*, 44(8):941–945, June 2012.

Li, B. Comprehensive analyses of tumor immunity: implications for cancer immunotherapy. *Genome Biol*, 17(1):174, August 2016.

Li, T. TIMER: A web server for comprehensive analysis of Tumor-Infiltrating immune cells. *Cancer Res*, 77(21):e108–e110, November 2017.

Lopez, R. Deep generative modeling for single-cell transcriptomics. *Nat Methods*, 15(12):1053–1058, November 2018.

Mahmoud, S. M. A. Tumour-infiltrating macrophages and clinical outcome in breast cancer. *J Clin Pathol*, 65(2):159–163, November 2011.

Menden, K. Deep learning-based cell composition analysis from tissue expression profiles. *Sci Adv*, 6(30):eaba2619, July 2020.

Newman, A. M. Robust enumeration of cell subsets from tissue expression profiles. *Nat Methods*, 12(5):453–457, March 2015.

Nguyen, H. Fourteen years of cellular deconvolution: methodology, applications, technical evaluation and outstanding challenges. *Nucleic Acids Res*, 52(9):4761–4783, May 2024.

Parker, S. J. Identification of putative early atherosclerosis biomarkers by unsupervised deconvolution of heterogeneous vascular proteomes. *J Proteome Res*, 19(7):2794–2806, April 2020.

Poduri, A. Somatic activation of AKT3 causes hemispheric developmental brain malformations. *Neuron*, 74(1):41–48, April 2012.

Rivière, J. B. De novo germline and postzygotic mutations in AKT3, PIK3R2 and PIK3CA cause a spectrum of related megalencephaly syndromes. *Nat Genet*, 44(8):934–940, June 2012.

Rooney, M. S. Molecular and genetic properties of tumors associated with local immune cytolytic activity. *Cell*, 160(1-2): 48–61, January 2015.

Rosenberg, J. CD8(+) T cells and NK cells: Parallel and complementary soldiers of immunotherapy. *Curr Opin Chem Eng*, 19:9–20, December 2017.

Tsoucas, D. Accurate estimation of cell-type composition from gene expression data. *Nat Commun*, 10(1):2975, July 2019.

Verma, V. and Lamb, A. Manifold mixup: Better representations by interpolating hidden states. June 2018.

Wang, M. Comparative analysis of transcriptional profiling of CD3+, CD4+ and CD8+ T cells identifies novel immune response players in t-cell activation. *BMC Genomics*, 9:225, May 2008.

Wang, X. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat Commun*, 10(1):380, January 2019.

Xu, C. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol Syst Biol*, 17(1):e9620, January 2021.

Zhong, Y. and Liu, Z. Gene expression deconvolution in linear space. *Nat Methods*, 9(1):8–9; author reply 9, December 2011.

A. Supplementary figures

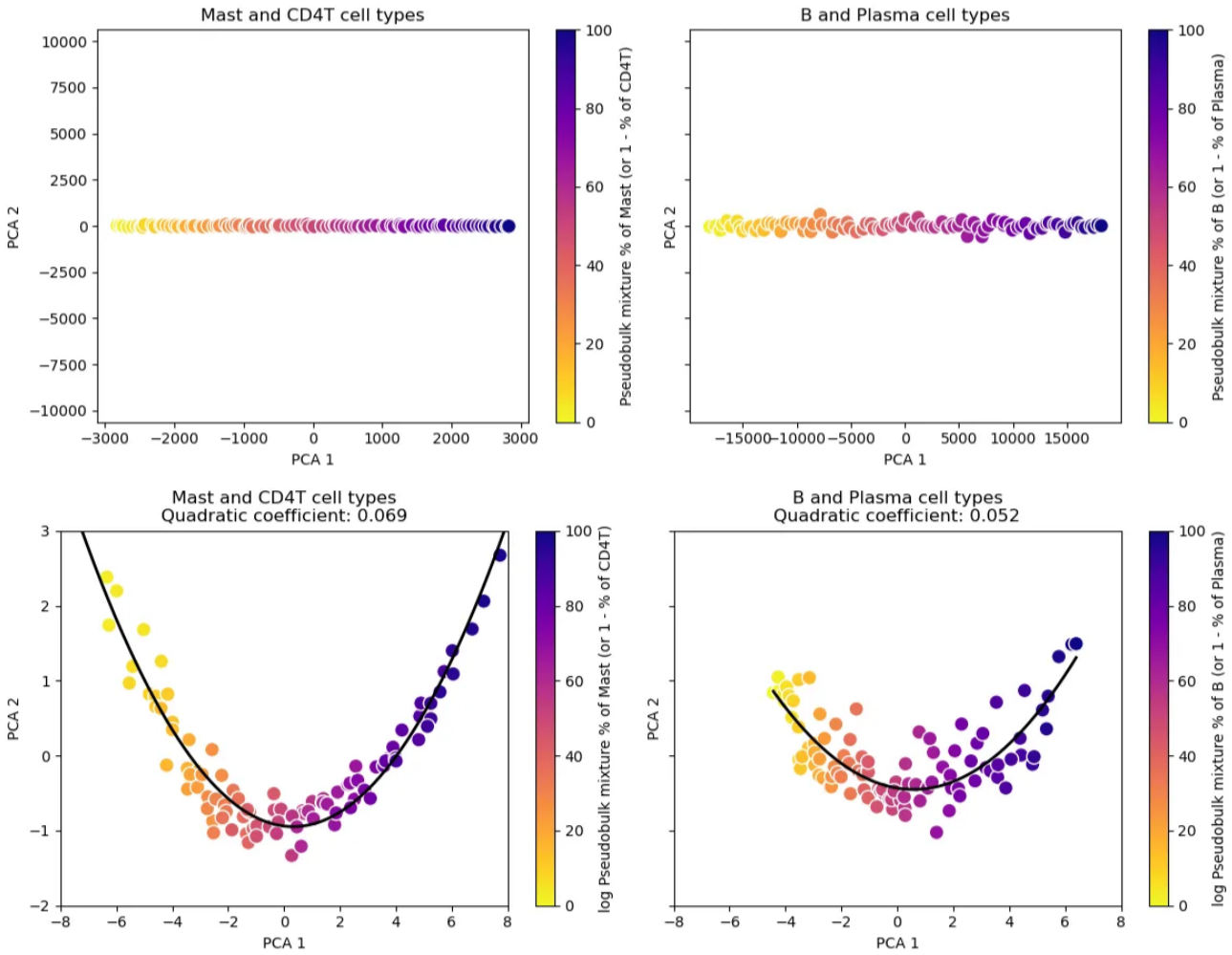


Figure S1. PCA of mixed pseudobulks. With the Cross-tissue immune single-cell atlas (section 3.1), we construct pseudobulks composed of varying proportions of cell types. The interpolation between mixtures is linear when the gene expression is not log-transformed (above figure), while the interpolation between mixtures shows a curved trajectory when the gene expression is log-transformed (below figure). The effect is more pronounced when combining dissimilar cell types, such as Mast and CD4T cells, compared to more similar ones like B and Plasma cells.

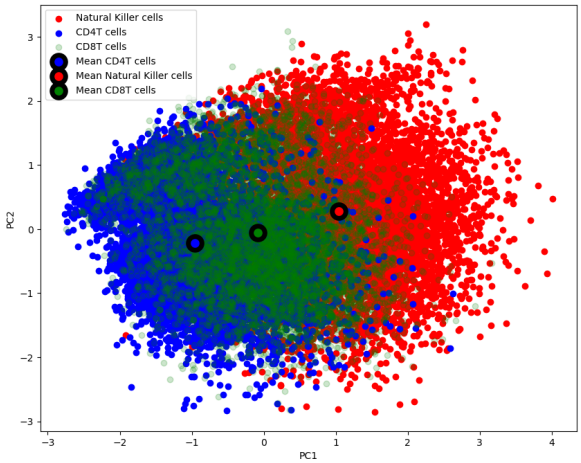


Figure S2. Using the Cross-tissue immune single-cell atlas (Section 3.1), we analyze the variability of lymphoid cells gene expression. Due to shared functions and markers, $CD8^+$ T cells were grouped with both $CD4^+$ T (Wang, 2008) and NK cells (Rosenberg, 2017) in the broader granularity. While $CD4^+$ T and NK cells are mainly separated along PC1, they are both mixed with $CD8^+$ T cells. This highlights underlying similarities between $CD4^+$ T and NK cells, but also sufficient differences that may introduce unwanted variability in coarse cell type definitions.

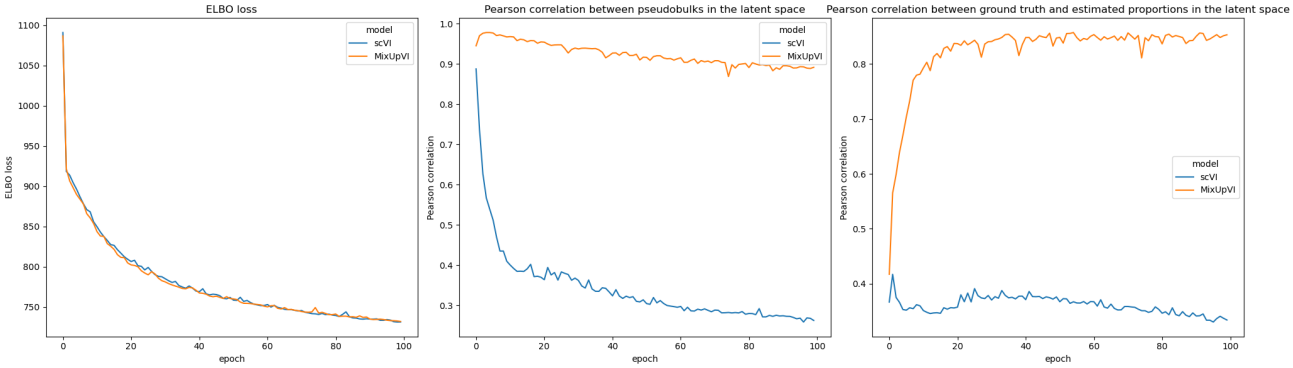


Figure S3. Monitoring of the ELBO loss (left plot), Pearson correlation between encoded pseudobulks in the latent space (middle plot), Pearson correlation between ground truth and estimated proportions in the latent space (right plot) during training.

B. Dirichlet sampling

As explained in Section 2.2, we sample the pseudobulk cell type proportions from a Dirichlet distribution. This is a useful sampling in the case of pseudobulk creation because it sums proportions to 1 and allows for over-representation of rare cell types.

This Dirichlet distribution is the posterior of a Dirichlet-Multinomial conjugate model, assuming a non-informative prior and a multinomial likelihood based on the observed cell counts. More precisely:

- We assume that $(p_1, ..., p_C)$ to be the vector of multinomial parameters, thus the probabilities associated to every cell type.
- We assume that this vector follows a Dirichlet prior to observing the batch of single-cells. More formally: $(p_1, ..., p_C) \sim Dirichlet(\alpha_1, ..., \alpha_C)$.
- In this benchmark, we chose a non-informative prior distribution: $\alpha_1 = ... = \alpha_C = 1$. In that special case, the Dirichlet distribution is actually a Uniform distribution. It is also possible to set this prior to match the belief one has over the cell type proportions of the given tissue from which the single-cell data is drawn.
- Then, a batch of single-cell is drawn. So are the cell type proportions of this given batch: $x_1, ..., x_C$. Given these observed cell type proportions, using the conjugate property of the Dirichlet and Multinomial distributions, we can easily compute the posterior distribution of the vector of multinomial parameters:

$$(p_1, ..., p_C)|(x_1, ..., x_C) \sim Dirichlet(\alpha_1 + x_1, ..., \alpha_C + x_C) .$$

C. Related work

As detailed in Section 3.2, we benchmarked MixupVI against a broad range of established deconvolution methods (Cobos, 2020), all provided as Python implementations in our open-source library PyDeconv. This includes both classical regression-based techniques and deep learning models. Several of these methods were originally implemented in R, but have been re-implemented with Python to ensure consistency and accessibility. Below is an overview of each method:

- Ordinary Least Squares (OLS)
  A standard linear regression technique that estimates cell type proportions by minimizing the squared error between the observed bulk expression and the weighted combination of reference cell-type profiles. While simple and fast, OLS does not constrain predictions to be non-negative or sum to one, which can lead to biologically implausible results.
- Non-Negative Least Squares (NNLS)
  An extension of linear regression that constrains all estimated cell type proportions to be non-negative. This makes NNLS more realistic for compositional data like cell mixtures and has been widely used as a baseline in deconvolution studies.
- Robust Linear Regression (RLR)
  This method modifies standard regression by reducing the influence of outliers or highly variable genes. It improves stability in the presence of noise or batch effects, which are common in real-world transcriptomic datasets.
- Weighted Non-Negative Least Squares (WNNLS)
  W-NNLS improves upon NNLS by incorporating gene-specific weights based on their variance across cell types. This prioritizes genes with higher discriminatory power, enhancing the robustness of proportion estimates.
- Dampened Weighted Least Squares (DWLS)
  DWLS extends WNNLS by further iteratively adjusting the contribution of each gene through dampening weights, reducing the impact of genes with extreme variance. This method is particularly effective for detecting rare cell populations and managing noise in lowly expressed genes.
- Nu-Support Vector Regression (NuSVR)
  This model is based on a linear kernel model using an extra regularization parameters nu which simultaneously controls the number of support vectors and the upper bound on the fraction of training errors. Compared to OLS, the SVR model is robust against noise, can automatically select important genes from the signature matrix, and can account for multicollinearity between cell types.

In addition to classical regression-based methods, we benchmarked two deep learning approaches that model complex non-linear relationships in gene expression data:

- Tissue-Adaptive autoEncoder (TAPE)
  TAPE leverages an autoencoder-based architecture trained on simulated bulk RNA-seq data to learn latent representations that capture cell-type specific expression patterns. The training phase uses deep neural networks trained on pseudobulk datasets derived from single-cell RNA-seq data. During inference, TAPE incorporates an adaptive self-supervised step to perform domain adaptation, taking advantage of the autoencoder reconstruction capabilities to better generalize to real bulk samples.
- Scaden
  Scaden employs an ensemble of deep neural networks trained on simulated pseudobulk datasets derived from single-cell RNA-seq data. As with most neural network approaches, its performance relies on access to a large and diverse dataset in order to generalize well across different tissues.

D. Metrics

To evaluate the impact of the mixup loss and assess latent deconvolution performance, we compute three metrics: two based on Pearson correlation and one based on mean squared error.

First, we measure the alignment between the encoding of a pseudobulk and the average of its constituent single-cell encodings. For a given pseudobulk $x_{\mathcal{I}}$, let $z_{\mathcal{I}} \sim q_\varphi(\cdot \mid x_{\mathcal{I}})$ denote its latent representation, and let $\bar{z} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} z_i$, with $z_i \sim q_\varphi(\cdot \mid x_i)$, be the average of the encoded cells. The mixup correlation for one pseudobulk is then defined as:

$$\rho(z_{\mathcal{I}}, \bar{z}) = \frac{\sum_{j=1}^{d}(z_{\mathcal{I},j} - \bar{z}\mathcal{I})(\bar{z}j - \bar{\bar{z}})}{\sqrt{\sum j = 1^d (z\mathcal{I}, j - \bar{z}\mathcal{I})^2}\sqrt{\sum j = 1^d (\bar{z}_j - \bar{\bar{z}})^2}} \,.$$

Averaging this over M pseudobulks gives the average mixup correlation:

$$\rho = \frac{1}{M}\sum_{m=1}^{M} \rho\left(z_{\mathcal{I}}^{(m)}, \bar{z}^{(m)}\right) \,.$$

Second, to quantify (latent) deconvolution performance, we compare the predicted proportions $\hat{\beta}^{(m)} \in \mathbb{R}^K$ with the true proportions $\beta^{(m)} \in \mathbb{R}^K$ used to generate pseudobulk m. The Pearson correlation for one pseudobulk is:

$$\rho(\beta^{(m)}, \hat{\beta}^{(m)}) = \frac{\sum_{k=1}^{K}(\beta_k^{(m)} - \bar{\beta}^{(m)})(\hat{\beta}k^{(m)} - \bar{\hat{\beta}}^{(m)})}{\sqrt{\sum k = 1^K (\beta_k^{(m)} - \bar{\beta}^{(m)})^2}\sqrt{\sum_{k=1}^{K}(\hat{\beta}_k^{(m)} - \bar{\hat{\beta}}^{(m)})^2}} \,.$$

The overall performance across pseudobulks is denoted the deconvolution correlation d:

$$d = \frac{1}{M}\sum_{m=1}^{M} \rho\left(\beta^{(m)}, \hat{\beta}^{(m)}\right) \,.$$

Third, we compute the mean squared error (MSE) between the predicted and true proportions:

$$\text{deconvolution MSE} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{K}\sum_{k=1}^{K}\left(\hat{\beta}_k^{(m)} - \beta_k^{(m)}\right)^2 \,.$$

E. Hyperparameters

Here we describe the parameters of typical benchmarking configs and hyperparemeters used and tuned to train MixupVI.

Deconvolution benchmark hyperparameters

- Deconvolution method.
  The deconvolution methods to benchmark in this experiment, among MixupVI, NNLS, scVI, DestVI, TAPE, Scaden, OLS, DWLS, RLR, NuSVR, WNNLS.
- Evaluation datasets.
  The single-cell datasets to test for deconvolution, used to create pseudobulks. In this benchmark, the test set of the Cross-tissue immune atlas was used.
- Training dataset.
  The dataset to use to train methods requiring training (e.g. MixupVI). In this benchmark, the train set of the Cross-tissue immune atlas was used.
- Granularities.
  The granularities to benchmark. In this benchmark, a first granularity with 5 cell types and a second granularity with 9 cell types were tested.
- Signature matrices.
  The signature matrices to use depending on the granularity. In this benchmark, signature matrices created with the training dataset were used for each granularity.
- Number of evaluation pseudobulks.
  The number of evaluation pseudobulks simulated for evaluation. In this benchmark, 400 pseudobulks were created.
- Number of cells per evaluation pseudobulk.
  The number of cells constituting pseudobulk simulations. In this benchmark, a list ranging from 10 cells to 1000 cells was used.

MixupVI model hyperparameters

All scvi-tools models hyperparameters

- Latent size.
  As described in Section 4.1, this hyperparameter is crucial and should be carefully tuned depending on the granularity. In the 1st granularity, we fixed it to 10, while on the 2nd granularity, it was fixed to 30.
- Number of input genes.
  The number of input genes to keep in the training datasets for the methods needing training. This parameter is common to all methods requiring training (e.g. MixupVI). It is tunable, as a trade-off needs to be found between high-dimensionality, long training times and preserving cell type markers. In this benchmark, the 2500 most variable genes were kept.
- Maximum number of epochs.
  The maximum number of epochs to train the model on. In this benchmark, it was fixed to 100 epochs.

MixupVI training hyperparameters

- Batch size.
  This will determine, in the mini-batch gradient optimization, the number of cells in a given batch. By extension, this is the maximum number of cells a training pseudobulk can be composed of. In this benchmark, it was fixed to 2048.
- Train size. For the training dataset described in Section 3.1, this is the proportion kept for purely training, and by extension, it sums up to 1 with the proportion kept for validation at every forward pass. In this benchmark, it was fixed to 0.7.

MixupVI model hyperparameters

- Number of pseudobulks.
  The number of pseudobulks created in an ensembling fashion in a given single-cell batch to compute the mixup loss. The more pseudobulks, the longer a run and the more demanding on GPU memory. In this benchmark, it was fixed to 100.
- Number of cells per pseudobulk.
  The number of cells constituting the pseudobulks. By adjusting this hyperparameter, one can flexibly simulate different

experimental resolutions: larger values mimic bulk RNA-seq conditions, while smaller values emulate spot-based spatial transcriptomics where each spot contains fewer cells, with a better modeling of the variance inside the latent space. In this benchmark, it was fixed to 100.

- Size and number of hidden layers.
  In the benchmark, one hidden layer was used, of size 512.
- Continuous and categorical covariates.
  Although theoretically useful to mitigate batch issues, such as center or patient effects (if these effects are present in the training set), they did not improve results in the benchmark, thus they were not used.
- Loss computation.
  Whether to enforce the mixup loss within the latent space or the reconstructed space (or both). In the benchmark, it was fixed to the latent space.
- Pseudobulk computation.
  Whether to create the pseudobulks in the input space, or within the latent space, before passing them through the encoder, or decoder. When the loss is computed in the latent space, only the pre-encoded pseudobulks can be computed.
- Signature type. Whether to compute the latent signature matrix used for deconvolution by using pre-encoded purified cell type vectors and passing them through the encoder, or to sample them directly from the latent space. This hyperparameter does not change the optimization of the model, as no deconvolution is done to optimize the model, only to infer from it.
- Mixup penalty. It can either be a L2 loss between the pseudobulk of the sampled encoded cells and the sampled encoded pseudobulk, or the Kullback-Leibler divergence between the distribution of the sampled encoded pseudobulk and the pseudobulk of the sampled encoded cells. In this benchmark, as explained in Section 2.2, the L2 loss was used.
- Gene likelihood.
  The distribution of the input data, used to compute the reconstruction loss part of the ELBO. In this benchmark, it was fixed to Zero-inflated negative binomial (ZINB) for single-cells and Negative binomial (NB) for pseudobulks.
- Batch normalization.
  Whether to add batch normalization inside the encoder or/and decoder for every batch of a forward pass. In this benchmark, it was chosen to not use it neither in the encoder or the decoder.