Disentangling cellular heterogeneity into interpretable biological factors through structured latent representations

Amir Ali Moinfar[1,2], Fabian J. Theis[1,2,3‡]

1 School of Computing, Information and Technology, Technical University of Munich, Garching, Germany.
2 Institute of Computational Biology, Helmholtz Center Munich, Neuherberg, Germany.
3 TUM School of Life Sciences Weihenstephan, Technical University of Munich, Freising, Germany.

†Correspondence to:
fabian.theis@helmholtz-muenchen.de

Abstract

Single-cell genomics allows for the unbiased exploration of cellular heterogeneity. While deep generative models have become a keyprocessing step for integration and dimensionality reduction of single cell data, they function as black boxes that entangle biological signals and gene programs, restricting their interpretability and utility for downstream analysis. We address this limitation with Disentangled Representation Variational Inference (DRVI), a deep generative framework designed to provide dimension-wise interpretable disentangled representations for single-cell omics. DRVI isolates distinct sources of variation — such as cell-type identity, developmental stage, cytokine signaling, and technical noise — into individual, interpretable, and parallel factors without relying on supervised priors or restrictive linearity assumptions. This is achieved by exploiting additive decoders with a novel nonlinear pooling approach. We validate DRVI's capabilities to isolate cell types, secondary biological processes, perturbation effects, and stages in continuous trajectories on seven diverse datasets. In the Human Lung Cell Atlas, DRVI enables isolation of biological processes beyond cell types, covering interferon signaling, angiogenesis, proliferation, and stress response to metal ions, and identifies rare cell types such as migratory dendritic cells. In other examples, we show how to use interpretable factors to regress out isolated technical stress signals caused by sequencing, and to refine annotations for dendritic cells in an immune dataset. DRVI's interpretability does not come at the cost of integration performance, interpretable factors are consistent and transferable to similar datasets, and it generalizes easily to other modalities, such as single-cell chromatin accessibility. Altogether, DRVI provides a versatile framework for resolving the underlying structure of single-cell omics data.

Introduction

Single-cell RNA-seq technologies are widely used for profiling transcriptomes or other omics levels at single-cell resolution in a high-throughput fashion [1, 2]. These powerful readouts empower researchers to explore developmental processes, assess the impact of drugs and perturbations across diverse cell lines, construct reference atlases, and compare organoids with primary tissues [3–10]. Recent advancements have expanded single-cell measurements beyond gene expression to additional modalities, including chromatin accessibility, spatial information, and surface proteins [11–17]. Consequently, many tools and methods have emerged to facilitate the analysis of these datasets [18]. Among these, generative models have effectively summarized data within a latent space, particularly when dealing with multiple batches [19, 20], outperforming more traditional methods in large-scale benchmarks and competitions for integration and cross-modality inference [21]. Subsequently, several extensions have been developed to address various modalities and use cases [22–25].

Despite their widespread use, generative models commonly produce entangled latent representations, where multiple unrelated biological processes are intertwined within a single dimension [26]. This limits the interpretability of the latent space, restricting its direct application for downstream tasks [27, 28]. As a result, the latent space is often only used as an intermediate, and its utility is limited to neighborhood graphs constructed on top of the cellular representations. Specifically, common techniques like Uniform Manifold Approximation and Projection (UMAP) [29, 30] for visualization and Leiden clustering [31] for cell type identification exclusively rely on these neighborhood graphs. While clustering methods are essential for extracting biological variations from the high-dimensional single-cell omics data [32], they have limitations. They cannot capture continuous processes or identify shared programs among multiple cell types as a single concept and often identify the most dominant signals, such as cell types, which makes it challenging to uncover secondary signals.

Disentangled representation learning addresses these limitations by separating the underlying factors of variation into distinct, semantically meaningful latent variables [33]. However, unsupervised learning of truly disentangled representations presents a fundamental challenge, as demonstrated by theoretical impossibility results, which highlight the necessity of inductive biases on both models and data [34]. This has motivated diverse approaches, from methods tailored for specific data modalities like recommendation systems [35], to more general mathematical frameworks [36, 37].

Several disentangled latent space models have been proposed for single-cell omics data. However, they either rely on linear assumptions [38, 39], auxiliary data, such as variations of interest [40–42], or both [26, 43–45] to enforce disentanglement. In practice, for new discoveries, supervised data is absent or, if present, frequently biased. And, linear models fail to capture the complexity of biological data, particularly in scenarios involving multiple batches. These limitations motivate the development of unsupervised nonlinear disentangled representation learning. While there are a few unsupervised and unconstrained disentanglement models [46, 47], we will experimentally confirm the theoretical impossibility results [34], and show none of these unconstrained models demonstrate satisfactory disentanglement performance, which may explain the fact that they see little practical use. In Supplementary Note 1, we discuss disentangled representation learning in more detail, provide an overview of disentangled latent space models tailored specifically for single-cell omics data (Supplementary Table 1, which includes a less relevant category of block-based disentanglement), and highlight the contributions of this work.

We leverage the recently proposed framework of additive decoders [37], which enables latent variable identification with theoretical guarantees. We present Disentangled Representation Variational Inference (DRVI), an unsupervised generative model capable of learning non-linear interpretable disentangled latent representations from single-cell count data. DRVI addresses the well-known impossibility of unsupervised disentanglement [34] by leveraging additive decoders followed by a suitable non-linear pooling function, which serves as a crucial inductive bias tailored for the unique characteristics of single-cell omics. Empirical evaluations across a diverse range of biological applications demonstrate DRVI's capabilities in learning disentangled latent representations, outperforming existing methods in terms of disentanglement without reducing integration quality. DRVI facilitates the identification of biological states by representing cell types across individual latent factors in cell atlases, enables exploration of cellular heterogeneity by identifying biologically meaningful variations beyond cell types, captures developmental stages in developmental datasets, and identifies coherent groups of effective perturbations in perturbational studies. Additionally, DRVI allows users to selectively highlight or exclude latent factors associated with specific known variations, offering flexible post-hoc analysis capabilities.

Results

DRVI learns interpretable disentangled cell representations

DRVI (Disentangled Representation Variational Inference) aims to integrate datasets and learn unsupervised disentangled latent representations of biological processes from single-cell omics data. A representation is disentangled if there is a one-to-one correspondence between its latent factors and components of a ground truth representation (see 'Definitions and Notations' in Methods)[48, 49]. In this context, disentanglement means identifying distinct biological processes and properties, such as cell types, signaling pathways, and developmental stages, and representing them in individual latent factors (Fig. 1a). DRVI achieves this by exploiting the covariations present in the data. The model is based on the assumption that independent underlying processes combine additively in gene count space (justified in Supplementary Note 2). While DRVI is built within the commonly used conditional variational autoencoders (CVAE) framework [19, 50], its decoder is distinct from existing models. Specifically, it employs additive decoders followed by a pooling function to enforce the disentanglement and represent distinct sources of variations in different latent factors.

The encoder g takes the (commonly) normalized gene expression of a cell x and optionally the covariate variable c — representing batch or other technical information to regress out, where both one hot encoding [19] and embedding [51] is supported — as input and maps them into the latent space z. Then, the decoder f constructs the parameters of the generative distribution such as Gaussian or Negative Binomial (Fig. 1b). Unlike the encoder that employs a multi-layer perceptron (MLP) capable of approximating a wide range of functions, the additive decoder architecture [37] is specialized to enforce disentanglement on the latent space dimensions. Specifically, the decoder takes the latent space vector z, of dimension K, and splits it into K/D splits of dimension D, where we use $D = 1$ for the most powerful form of disentanglement by default. Each split, optionally together with the covariate variable c, is decoded separately using a nonlinear subnetwork f_k (where $k \in \{1, \ldots, K/D\}$) back into the expression space $\hat{x}^k$ to later form the parameters of the generative distribution. Finally, the decoded vectors $\hat{x}^k$ are aggregated using a potentially nonlinear pooling function, combining K/D vectors of the same size into one, to form the final generative distribution. The overall model is optimized based on the variational inference framework, with the encoder representing the approximate posterior and the decoder representing the generative model (see 'The Generative model' in Methods).

By default, DRVI uses log-sum-exp (LSE) pooling for aggregation, so altogether, decoding yields $\hat{x} = \log \sum_{k=1}^{K/D} \exp f_k(z_k, c)$. To emphasize the importance of the LSE aggregation, we explore an alternative scenario when we simply average the parameters, referred to as DRVI with average pooling (DRVI-AP). We practically observe that the introduced architecture with the LSE pooling effectively enforces disentanglement and enables the recovery of biological processes beyond cell types.

A significant advantage of DRVI is the interpretability of its latent space. DRVI individually decodes each latent factor to the expression space. Consequently, we can interpret the decoded effect of each factor as a nonlinear gene program (Fig. 1c). To identify these gene programs, we traverse each latent factor and inspect the affected features (e.g., genes in transcriptomics). Since the effect changes nonlinearly as we traverse the factor, we consider the maximum effect for each factor (see 'Interpretability' in Methods). For each factor, we refer to the genes identified by the interpretability pipeline as 'relevant genes' for that factor. Alternatively, one can perform differential expression (DE) analysis to identify the genes involved in a nonlinear gene program beyond the genes used for model training.

Once we identify the top relevant genes, we can determine some programs through supervised external information, such as existing annotations and gene-set enrichment analysis (GSEA). We note that identifying biological programs doesn't depend on GSEA for derived gene sets. Instead, it

can involve searching scientific literature, leveraging existing atlases of marker genes and biological programs, utilizing automated tools based on language models, or performing expert examination by biologists. It is worth mentioning that since such supervised information is not given to the model, the quality of the derived signatures is neither affected nor biased by it. Finally, unidentified processes are promising candidates for further literature search, additional analysis, and even experimental design.

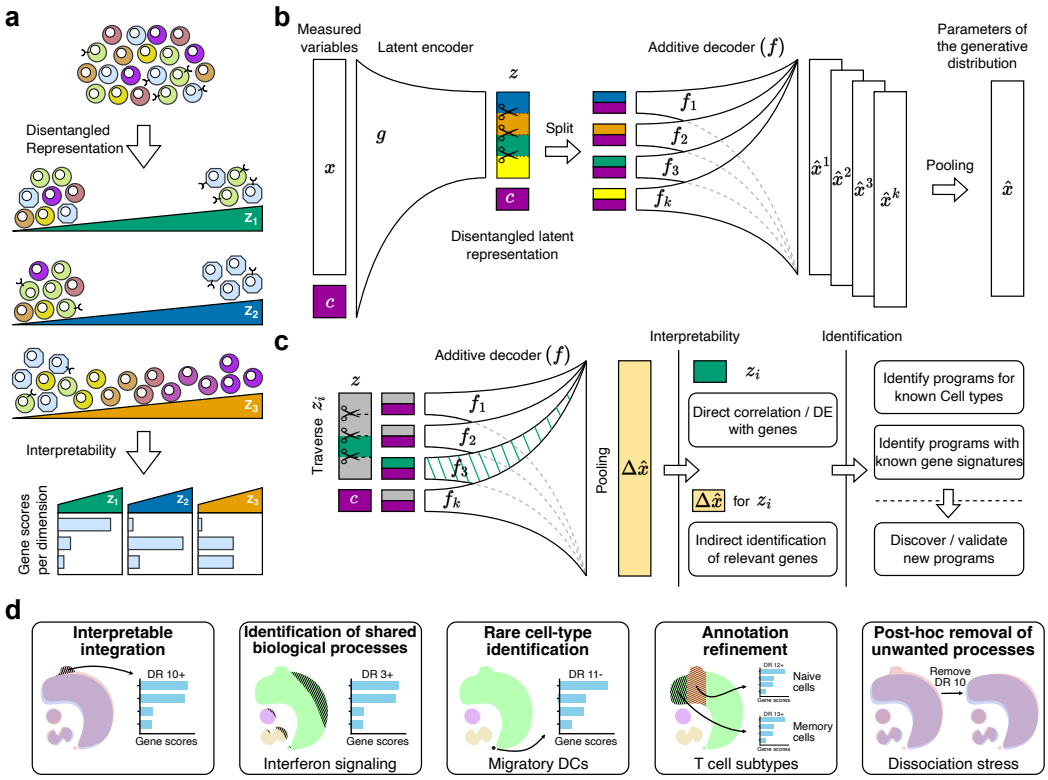


Figure 1: **DRVI enables learning interpretable disentangled latent representations. a,** Disentanglement concept. A disentangled latent model should encode concepts such as cell properties, cell types, and developmental stages in distinct latent factors. Each factor is inherently interpretable. **b,** Schematic of the DRVI model. The model consists of a nonlinear encoder and an additive decoder followed by a pooling function that enforces the disentanglement into the encoder and latent factors. **c,** Interpreting disentangled factors. Relevant genes can be discovered by traversing a latent factor or using DE analysis tools. While some factors can be identified using known gene sets or known cell type annotations, others can be validated as potentially novel programs. **d,** DRVI enables unbiased exploration of single-cell omics. First, it provides atlas-scale, interpretable integration where latent factors can be visualized directly on the cellular manifold (e.g., UMAP). Second, DRVI identifies continuous underlying biological processes that can be shared across the manifold. Third, its additive architecture isolates rare cell types into specific latent dimensions, facilitating their identification better than standard clustering. Fourth, DRVI's unsupervised and unbiased nature allows for uncovering finer cell types along their interpreted signatures when base annotation is coarse or inaccurate. Finally, the disentangled latent space allows for removing unwanted factors, such as technical stress responses, post-integration. DE, differential expression; DR, disentangled representation.

In the following subsections, we demonstrate that DRVI achieves atlas-scale integration and provides inherently interpretable latent representations. Each factor can be visualized on the cellular manifold to reveal biological processes, such as inflammation or the cell cycle, or to highlight specific cell types. DRVI's additive architecture isolates rare cell types into individual latent factors, facilitating their identification, as exemplified by fibroblasts in the immune dataset and migratory dendritic cells in

the HLCA. Furthermore, as an unsupervised method, DRVI identifies factors that reveal cellular subtypes allowing users to refine their annotations based on cross-referencing top-ranking genes of these process with known marker genes. Finally, users can selectively remove unwanted biological or technical processes by excluding their corresponding latent factors (Fig. 1d).

DRVI identifies biological signals beyond cell types

To demonstrate DRVI's disentangled and interpretable latent space, we applied DRVI with 32 latent factors to an immune dataset [52] composed of 9 batches from four human peripheral blood and bone marrow studies, with 16 annotated cell types (Fig. 2a, UMAPs for other benchmarked models in Supplementary Fig. 1a,b).

DRVI ranks the latent factors based on their impact on reconstruction, labeled as DR 1, DR 2, etc (see 'Ordering the learned factors' in Methods). Like many other dimensionality reduction techniques, DRVI is unable to distinguish between positive and negative directions. Thus, interpretation requires considering both direction and dimension (e.g. DR 1+). To illustrate the disentanglement capabilities, we highlighted example factors representing a cell-type identifier, a cell-type sub-process, a shared process, and two consecutive stages in a developmental process. For the example factors, we provided UMAPs colored by their values in different cells, referred to as activity (Fig. 2b-e). Additionally, we visualized the geometry of the developmental process in a scatterplot (Fig. 2e). We identified the top genes related to each factor (Supplementary Fig. 2) and illustrated them for the example factors (Fig. 2f). Here, we explore these example latent dimensions and their biological interpretations. Then, we leverage DRVI's interpretable factors to identify rare fibroblast populations and provide high-resolution annotations for monocyte-derived dendritic cells (Fig. 2g,h).

DR 17 identifies Plasmacytoid dendritic cells (pDCs) in the negative direction (Fig. 2b). CLEC4C and LILRA4 are marker genes for pDCs, and IL3RA overexpressed in this cell type, is an essential gene for the functioning and development of pDCs (Fig. 2f) [53, 54]. We provided a heatmap of the latent embeddings for DRVI and each benchmarked method to demonstrate DRVI's effectiveness in identifying cell types (Supplementary Fig. 3).

DR 19- highlights a subset of CD4+ T-cells (Fig. 2c). The presence of FOXP3, CTLA4 (CD152), and IL2RA (CD25) in the nonlinear gene program (Fig. 2f) suggests that this factor is targeting regulatory T-cells (T-regs) [55, 56]. GSEA analysis also supports the association of this factor with T-cell activation regulation.

DR 26+ indicates a shared process among multiple cell types (Fig. 2d). BTG2, DUSP2, and IER2 are the top genes in the nonlinear gene program, overexpressed at higher levels of this factor (Fig. 2f). BTG2, IER2, DUSP1, FOS, JUN, EGR1, and ZFP36, all within the top 20 relevant genes, have been previously reported as dissociation-induced stress-response genes in single-cell transcriptomic data [57, 58]. We observe that the activity of DR 26+ varies among different cell types, aligning with previous findings [58, 59]. Disentanglement of the dimensions in DRVI allows users to selectively remove irrelevant factors from the latent space. For instance, removing DR 26, a technical variation, enhances integration quality (Supplementary Fig. 4a-c).

Factors DR 21+ and DR 16- represent two consecutive stages in the developmental lineage of erythrocytes (Fig. 2e). The scatter plot illustrates an increase in DR 21 followed by a delayed decrease (increase in magnitude) of DR 16 in erythrocyte progenitors. As erythrocyte progenitors transition to mature erythrocytes, DR 21 decreases to zero while DR 16 reaches its maximum magnitude. Finally, DR 16 increases back to zero as the cells leave this developmental stage. Looking at the relevant genes, CD36 for DR 21+ is a known marker of erythrocyte progenitors [60]. For DR 16-, HEMGN (EDAG) is a key driver of erythropoiesis [60, 61], and RHCE encodes the RhD antigen in the red blood cell (RBC) membrane [62] (Fig. 2f).

A practical application of DRVI in identifying cell types is annotation refinement. We used DR 17, DR 20, and DR 29 to annotate pDC, DC1, and DC2 subtypes of dendritic cells, respectively (Fig. 2g

and Supplementary Fig. 5a-d). We validated this refinement using curated markers of these subtypes (Supplementary Fig. 5j). DRVI also guides users to potential misannotations. For instance, DR 30 revealed an underrepresented population of 12 Fibroblast cells (Fig. 2h and Supplementary Fig. 5e-h), which were previously misannotated as HSPCs, Monocyte progenitors, or Erythroid progenitors. The validity of this revised annotation was confirmed using COL1A1, a known Fibroblast marker not included in the model's HVG input (Supplementary Fig. 5i).

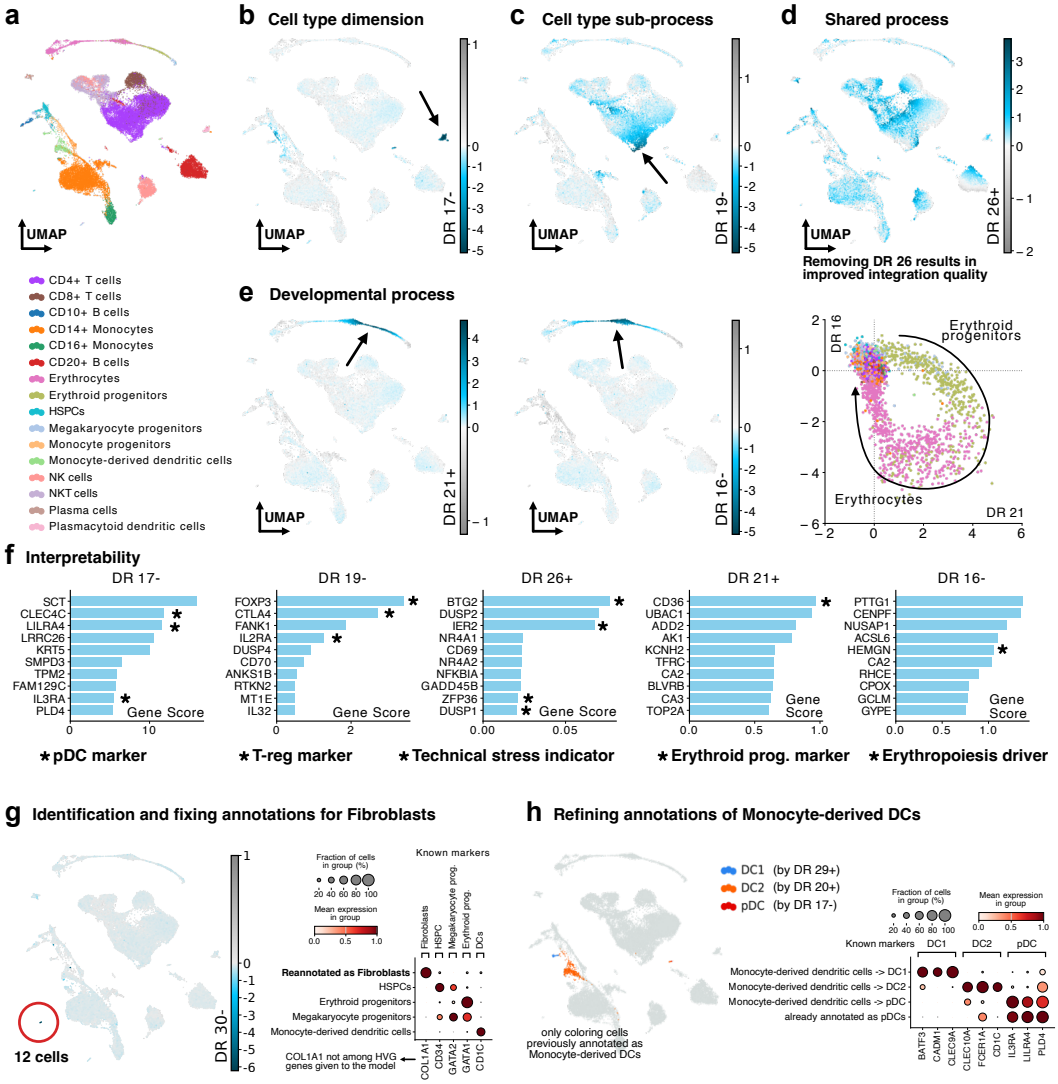


Figure 2: **DRVI provides unbiased disentangled representation in Immune data. a,** Uniform manifold approximation and projection (UMAP) of the immune dataset integrated by DRVI colored by cell types. **b-e,** Examples of factors identified by DRVI indicating a cell type (e), a cell-type-specific process (f), a shared process (g), and a developmental process (h). For the developmental process, a scatterplot visualizing two consecutive stages is provided. The arrow is for visualization and is not derived from the model. **f,** Top identified genes for each example factor. The very well-known marker genes are marked with stars. **g,** Identification of Fibroblasts in the Immune data. We identified and fixed the annotation of 12 Fibroblast cells separated by DR 30-. We verified this subset using COL1A1, a marker of Fibroblast that was not present in the HVGs given to the model. **h,** For cells originally annotated as Monocyte-derived dendritic cells, we used three factors representing subtypes to refine annotations into DC1, DC2, and pDC cells. Cells originally marked as pDC are not updated. The dotplot of known marker genes indicates successful annotation refinement. DC, dendritic cells; pDC, plasmacytoid dendritic cells; T-reg, regulatory T-cells; prog., progenitors.

DRVI outperforms previous disentanglement approaches while maintaining integration quality

Disentangled representation learning aims to identify biological processes in different axes of the latent space. We refer to all (often unknown) biological processes that are represented in the data and have contributed to its formation as ground truth (GT) processes. From the GT processes to the observed single-cell omics data, we can consider the biology describing cells as a stochastic GT generative function. We call a latent space to be disentangled when latent space axes correspond to the GT processes (see 'Definitions and Notations' in Methods). Accordingly, we can measure the disentanglement by finding and assessing the best one-to-one matching between GT processes and latent factors. However, the GT processes and the GT generative function are often unreachable. Here, we propose to address this conundrum by assessing the disentanglement using supervised information such as annotated cell types, applied perturbations, or basic known biological processes such as cell cycle (Fig. 3a). Disentanglement is therefore measured by finding the best one-to-one matching between latent factors and proxy variables, technically called latent matching score (LMS) in the disentanglement literature [37] - by default, we measure the matching quality based on the Scaled Mutual Information (SMI, see 'Metrics' in Methods).

To systematically assess how well DRVI learns disentangled latent representations, we benchmarked DRVI against eight unsupervised methods covering linear (PCA, ICA, MOFA, LIGER, and scETM) and nonlinear (scVI, β-TCVAE, and MICHIGAN) representation assumptions. Among these methods, ICA, MOFA, LIGER, scETM, β-TCVAE, and MICHIGAN are designed to provide meaningful latent representations. We also benchmarked DRVI-AP (DRVI with average pooling) to specifically assess the effect of LSE aggregation. Furthermore, we applied PCA and ICA to the scVI latent space to demonstrate that an inherently entangled latent space cannot be easily disentangled post-hoc. To cover a wide range of scenarios, we evaluated the mentioned methods on two atlas-scale multi-batch data (HLCA [7] and PBMC [63]), a small multi-batch data (immune [52]), a dataset consisting of primary and organoid cells (retina organoid [10]), one small-scale developmental data (developmental pancreas [4]), one large-scale developmental data (Daniocell [64]), and one genetic perturbation dataset (CRISPR screen [5]), summarized in Supplementary Table 2. We used finest-level cell-type annotations as proxy GT processes for all but perturbation data, for which we used perturbation signatures as proxy for underlying GT processes (see 'Datasets' in Methods).

When evaluating DRVI on individual datasets, DRVI consistently outperforms or matches the best benchmarked methods in providing disentangled latent representations (Fig. 3b). Constrained models like ICA, MOFA, LIGER, scETM, and DRVI-AP sometimes achieve acceptable disentanglement, but their performance is inconsistent and generally limited to a few specific datasets. Consistent with the theoretical results [34], unconstrained nonlinear models, including scVI, β-TCVAE, and MICHIGAN, were unable to achieve significant improvements in disentanglement.

To address the inherent variability across different datasets, we normalized each metric by the corresponding PCA score before aggregation. The resulting metric is referred to as the gain over PCA (see 'Metrics' in Methods). The total score was calculated by averaging the gain over PCA across all datasets (Fig. 3b, top left, and Supplementary Fig. 6). We find that also in the overall metric, DRVI outperforms other benchmarked models, achieving a 17% improvement over scETM, the next best-performing model. DRVI's superior performance in disentanglement compared to linear models suggests that the underlying processes in single-cell omics data are inherently nonlinear, and incorporating a nonlinear generative decoder and negative binomial noise model improves the disentanglement over linear baselines. DRVI and DRVI-AP are optimized using the same objective as scVI. The significant gap between DRVI and scVI demonstrates the effectiveness of the DRVI decoder architecture in enforcing disentanglement, and the gap between DRVI and DRVI-AP demonstrates the significant contribution of the LSE pooling function. The results indicate that conventional methods like PCA and ICA are insufficient for disentangling scVI's latent factors, highlighting the necessity for novel architectural approaches, such as DRVI.

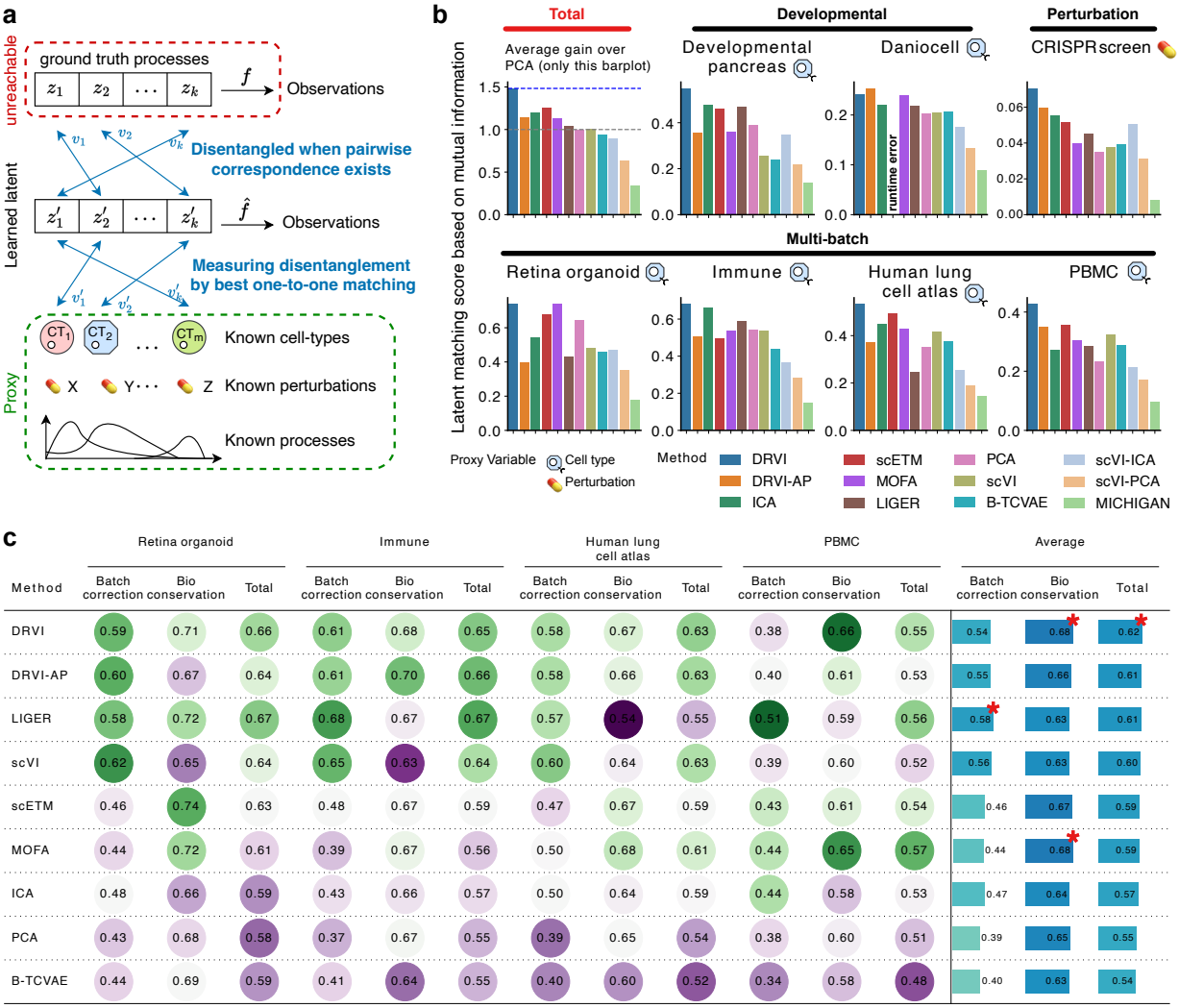


Figure 3: **Benchmarking DRVI in terms of disentanglement and integration quality. a,** Evaluation of disentanglement requires access to ground truth biological processes, which are often unreachable. We use cell type annotations, perturbations, and known processes as proxies when available. **b,** Disentanglement was evaluated using latent matching scores, calculated by finding the best one-to-one alignment of latent factors and proxy variables using mutual information. These scores were obtained from benchmarked models across diverse datasets. For each dataset, the proxy variable that serves as the ground truth is specified. The total score (top left) is determined by averaging the improvements relative to PCA. DRVI achieves the best result. **c,** scIB Integration scores for each individual multi-batch dataset. The last three columns indicate the average scores, with the best method in each category marked with a star. DRVI achieves the best results in unsupervised integration.

To further evaluate the benchmarked methods, we explored the same-process neighbors (SPN) similarity function (see 'Metrics' in Methods). Consistent with the mutual information metrics, DRVI demonstrated superior performance, achieving a 13% improvement in LMS-SPN compared to the closest competitor (Supplementary Fig. 7.). We also present the results from two additional disentanglement metric families (Supplementary Fig. 8): Most Similar Averaging Score (MSAS) and Most Similar Gap Score (MSGS). MSAS measures the maximum similarity between each GT process and latent factors, while MSGS penalizes similarity to other factors (see 'Metrics' in Methods). Particularly, when using the Scaled Mutual Information (SMI) similarity function, Most Similar Gap Score (MSGS) reduces to the Mutual Information Gap (MIG) metric as defined in [65]. Although DRVI consistently outperforms others in MSAS, different individual methods, including DRVI, ICA,

LIGER, and scVI-ICA, achieved the best performance in certain datasets and sub-metrics of the MSGS family. We find that the decline in DRVI's MSGS performance on the Immune dataset stems from a limitation in the ground-truth annotations rather than the model. DRVI identifies finer, more biologically precise cell subtypes (e.g., Naive vs. Th2 cells) that are grouped under single "ground-truth" labels. The MSGS metric incorrectly penalizes this identification of more granular subtypes (Supplementary Fig. 9). Considering this, the LMS metric family is favored due to its alignment with the formal definition of disentanglement, less sensitivity to imperfect GT variables, and its unique ability to penalize overlapping processes within a single factor, a key obstacle to interpretability.

Some unsupervised methods, such as ICA and MOFA, enforce disentanglement through strongly limiting model constraints like linearity and sparsity, restricting their ability to represent rich, multi-batch data. To assess the applicability of disentangled models on complex data, we assess integration quality alongside disentanglement. We consider the datasets containing multiple batches and assessed the biological signal conservation and batch correction quality based on the scIB metrics [52]. Considering the average performance across all datasets, DRVI achieved the highest total integration score among all benchmarked methods (Fig. 3c). This is due to its ability to correct batch effects comparably well (a little worse than scVI) while preserving significantly more biological signals (comparable to the best-performing model in bio-conservation). We can conclude that DRVI can learn disentangled latent representations in complex multi-batch scenarios while integrating the batches comparable to the best unsupervised counterparts.

DRVI allows atlas-level identification of biological processes

To demonstrate the capability of DRVI in disentangling latent factors and identifying biological signals in complex data, we applied DRVI to the Human Lung Cell Atlas (HLCA) [7]. We use the curated core of this atlas consisting of 166 samples from 14 datasets and including a total of 585K cells with curated annotations across 61 distinct cell identities in the finest level annotations. Disentangling 61 cell states requires a minimum of 61 factors. Accordingly, we applied DRVI (Fig. 4a) and other benchmarked methods (Supplementary Fig. 10) to form 64-dimensional latent spaces.

To interpret the learned latent factors, we first excluded 12 vanished factors, which exhibit near-zero values for all cells. The phenomenon of vanishing dimensions is explained later. We then identified 31 latent factors corresponding to cell types in the finest-level annotations. We observed a favorable one-to-one correspondence between latent factors and many cell identities (Fig. 4b top), while such correspondences for other benchmarked methods are much weaker (Supplementary Fig. 11). Disentanglement metrics confirm this observation quantitatively. The remaining 21 factors are expected to disentangle underlying processes that have yet to be identified (Fig. 4b bottom). Some factors exhibit two irrelevant processes in their negative and positive directions. This includes 4 cell-type factors that represent a biological process in their other direction. Therefore, we analyze the two directions separately.

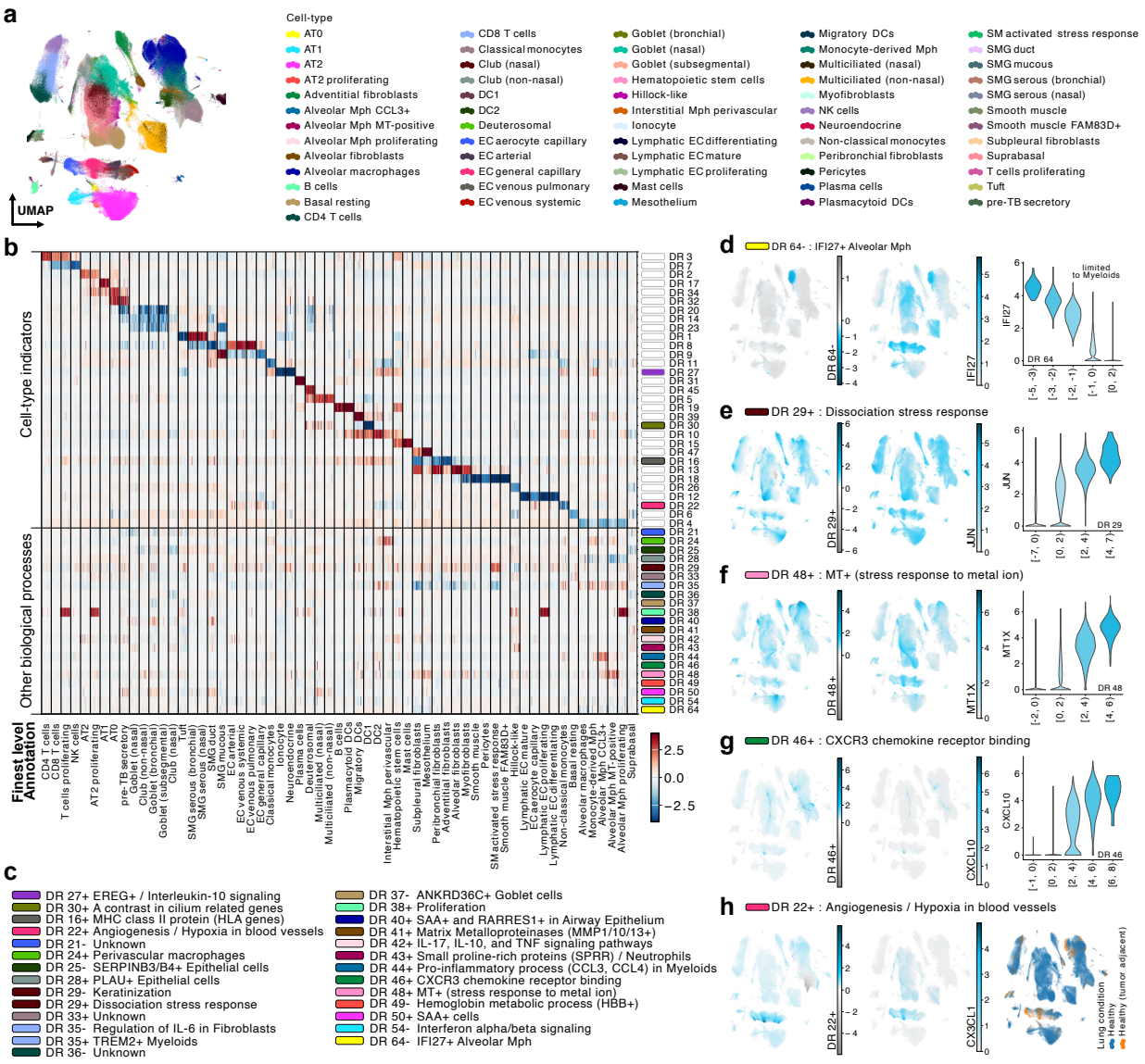


Figure 4: **DRVI uncovers the variations in HLCA beyond cell types. a,** UMAPs of the HLCA integrated by benchmarked methods. Cells are colored by third-level annotations. **b,** DRVI latent factors. The heatmap illustrates each factor's activity in cells vertically grouped by the finest level annotations. Cells are subsampled to have an equal number of cells in each cell type. The factors are horizontally grouped into two categories: factors indicating cell-type (top) and factors indicating other biological processes (bottom). Factors representing non-cell-type processes are color-coded on the right-hand side of the plot. Vanished factors are omitted. For better illustration, some tail values are clipped. **c,** For each factor indicating a non-cell-type biological process, a short description is provided. **d-g,** Demonstration of five example factors. Left: Activity of the example factors on the UMAP. Middle: The expression of a relevant gene on the UMAP. Right: Violin plots indicating the expression levels of relevant genes versus the activity of factors. **h,** The activity of DR 22+ and one of the identified genes in the nonlinear gene program. This factor indicates angiogenesis and hypoxia in endothelial cells and is highly active in some tumor-adjacent cells. Mph, Macrophage.

When examining the cell-type indicator factors, quite a few factors are exclusively activated in specific cell types. For instance, DR 17+ is uniquely active in AT1 cells, DR 31+ in plasma cells, DR 39- in DC1 cells, and DR 39+ in migratory DCs. Some other factors indicate broader categories covering multiple cell types, such as a coarse cell type or a shared behavior. For example, DR 4- indicates Macrophages (Mph), DR 5+ indicates ciliated cells covering deuterosomal and multiciliated

cells, and DR 32+ indicates AT0 and pre-TB secretory, together known as transitional club-AT2 cells [7] (Fig. 3b). Given that the disentanglement metrics already assess the identification quality of the factors identifying cell types, we will not focus on evaluating this aspect. However, it is noteworthy that DRVI effectively identifies most of the challenging cell types highlighted by the HLCA study, including Migratory DCs (identified by DR 39+), AT0 cells (identified by DR 34+), hillock-like epithelial cells (identified by DR 26-), and pre-TB secretory through transitional club-AT2 cell (identified by DR 32+). Other methods often fall short in identifying these cell types, particularly Migratory DCs, which only DRVI can capture along its latent factors (Supplementary Figs. 41 and 42). Interestingly, pre-TB secretory cells are also described in the HLCA study only through marker genes corresponding to transitional club-AT2 cells [7]. This is in full alignment with DRVI's latent geometry where AT0 and pre-TB secretory are together identified by DR 34+. These cell types are only identifiable after multiple iterations of clustering [7] or clustering at very high resolutions with hundreds of clusters (Supplementary Fig. 12). Given that DRVI can capture these cell types in just 52 non-vanished factors, we expect it to significantly facilitate the systematic discovery and seed annotation of novel and rare cell types in future cell atlases or related studies.

To demonstrate the validity of the identified variations beyond cell types, we followed the interpretability pipeline and found the non-linear gene programs related to each non-cell-type factor (Supplementary Fig. 14). We then utilized existing literature and available tools to identify these processes. As a result, we identified relevant biological descriptions for 22 of the 25 factors expected to represent biological processes (Fig. 3c). In the following, we detailed five example factors and visualized them (Fig. 4d-h). The rest of the factors presenting biological processes are identified and briefly explained in Supplementary Note 4.

As the first example, we showcased a subtle variation within Alveolar Macrophages (AMs - a subset of Myeloid cells), identified by DR 64- (Fig. 4d). IFI27 is the only gene significantly associated with this factor, and is known to define a subcluster of AMs [66–68]. Despite its expression in other cell types, such as blood vessels, DRVI has nonlinearly localized its expression to AMs due to the distinct patterns of co-regulated genes. The violin plot of the IFI27 gene with respect to DR 64 in the Myeloid cells confirms the mentioned relationship.

The activity of DR 29+ spans almost all cell types. This factor is highly associated with the genes ATF3, EGR1, JUN, FOSB, and FOS as top indicators of the nonlinear gene program, suggesting that it represents single-cell dissociation stress response [57, 58]. The violin plot of the JUN gene with respect to DR 29 highlights this relationship (Fig. 4e). Interestingly, we observed the same disentangled process (DR 26) in the immune dataset. Likewise, post-hoc removal of DR 29 enhances integration quality (Supplementary Fig. 4d-f).

The upregulation of Metallothionein (MT) family genes characterizes DR 48+, indicating a response to metal ions. The monotonic relationship between DR 48 and MT1X, visualized in the violin plot, further supports this finding (Fig. 4f). While this factor is active in MT-positive AM cells, DR 48+ is not exclusive to this cell type. Moreover, MT+ and IFI27+ AMs (DR 48+ and DR 64-) are not disjoint biological subtypes, indicating DRVI's superiority over traditional clustering methods in representing parallel overlapping processes.

A cell subset characterized by the expression of CXCL9, CXCL10, and CXCL11 is highlighted by DR 46+. GSEA analysis suggests an association with CXCR3 chemokine receptor binding [69], which is particularly relevant in the context of lung diseases [70, 71]. This factor is not specific to any particular cell cluster, indicating a shared response signature.

DR 22+ is primarily active in endothelial cells, with GSEA analysis suggesting an association with angiogenesis. The upregulation of top relevant genes, including CX3CL1, SERPINE1, AKAP12, GRP4, and FSTL3 under hypoxic conditions [72–76], a common phenomenon observed in tumor-adjacent environments [77], indicates a potential role in tumor growth. The increased abundance of tumor-adjacent endothelial cells in the higher end of DR 22+ further supports this interpretation. (Fig. 4h and Supplementary Fig. 15)

Altogether, DRVI enables the disentanglement of biological processes within the HLCA, a large-scale cell atlas. This facilitates the identification of rare cell types and the underlying biological processes. DRVI encodes biological processes such as inflammatory responses, signaling pathways, and processes that introduce variations within cell types. These processes can be shared across related cell types or localized to specific populations, depending on their gene expression signatures.

We note that the cell-type labels are not required to identify variations beyond cell types. One can follow the interpretability pipeline and find relevant gene sets for all factors. These genes may indicate different biological signals, including cell types, known pathways, and novel sets. However, as we see in the HLCA example, a good annotation will always facilitate analysis. Nevertheless, since this information is not given to the model, the quality of the labels does not bias the derived latent space.

DRVI disentangles cell-cycle information

We applied DRVI and other benchmarked methods to single-cell mouse samples of the developmental pancreas at E15.5, consisting of 3,696 cells and 15 states in the finest level annotation (Fig. 5a and Supplementary Fig. 16a). We first considered the cell cycle as one of the well-studied biological processes. While the cell cycle can be a confounding factor in some integration tasks, we expect unsupervised dimension reduction methods to retain this information and disentanglement methods to identify such biological signals. From a practical point of view, disentangling these biological signals enables researchers to analyze the remaining factors without interference from confounding factors.

The cell cycle is a complex, multidimensional process encompassing G1, S, G2, and M phases, frequently modeled as a set of single-dimensional processes corresponding to each phase [78]. Here, we used the cell-cycle G2M and S scores as two ground truth biological processes and measured each method's capability to disentangle these processes. Specifically, for each method, we calculated the scaled mutual information between each factor and the processes of interest. DRVI, MOFA, LIGER, and ICA were the only models that could separate at least one of the S and G2M phases (Supplementary Fig. 16b-e). DRVI achieved 19% higher sum of SMI scores compared to the next best model, and exhibited a smoother representation of cycling cells with clearer state boundaries (Fig. 5b and Supplementary Fig. 16f).

DRVI identifies developmental stages

In addition to the cell cycle in the developmental pancreas example, cells undergo development, namely endocrinogenesis, ultimately reaching four terminal states: alpha, beta, delta, and epsilon.

To quantitatively assess the quality of the embeddings in different settings, we applied CellRank [79] to infer macrostates, cell fates, and terminal states. We first applied scVelo [80] to infer cellular dynamics. scVelo determines possible future cell states by considering (I) spliced and unspliced counts and (II) the latent space where cell similarities are calculated. Therefore, we fed embeddings from DRVI and other benchmarked methods into scVelo as input latent spaces. Our findings indicate that DRVI improves the quality of velocity vector fields (Supplementary Fig. 17a). This improvement is evident in the Terminal State Identification (TSI) score, which quantifies the overlap between identified macrostates and ground-truth terminal states as the number of macrostates increases. Only DRVI and LIGER achieved a full TSI score (Fig. 5c and 17b-d). Furthermore, only DRVI embedding allowed CellRank to automatically identify and differentiate all four terminal states, distinguishing them from the transitioning and initial states (Fig. 5d Supplementary Fig. 17e). Within its interpretable disentangled latent space, DRVI identifies a set of factors leading to these four terminal states, which we selected and described below.

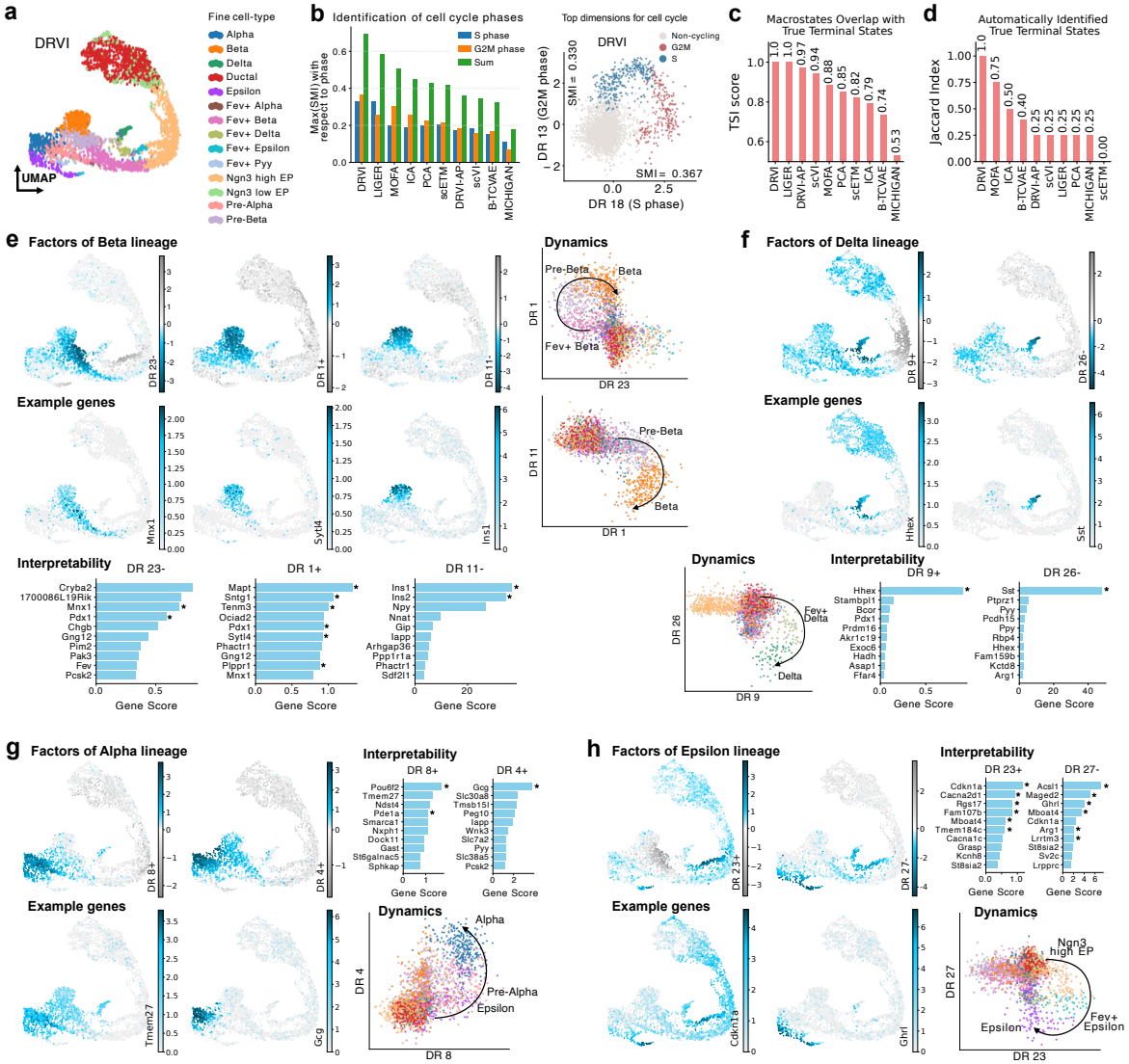


Figure 5: **Identification of cell cycle and developmental stages by DRVI. a,** UMAP of DRVI latent space for the developmental pancreas data. Cells are colored by finest-level annotations. **b,** The barplot shows the SMI score for the factors maximizing the SMI with respect to the S (blue) and G2M (orange) cell cycle phases in each benchmarked model. The sum of the two scores is provided in green. The scatterplot on the right shows DRVI factors representing these two phases, with points colored by the dominant cell cycle phase. Cells are assigned to the non-cycling category if both S and G2M scores are below 0.2. Otherwise, they are assigned to the state with the higher score. **c,** Terminal State Identification (TSI) scores for DRVI and benchmarked methods. This metric quantifies the overlap between identified stationary states and ground-truth terminal states as the number of macrostates increases. **d,** Jaccard index between the true known terminal states and the automatically identified terminal states using CellRank based on the embeddings from different benchmarked methods. **e-h,** Relevant factors for beta (e), delta (f), alpha (g), and epsilon (h) differentiation. For each of these terminal states, a group of relevant factors is provided. For each factor, the top relevant genes in the identified gene program are illustrated as a barplot, and the activity of a relevant gene on the UMAP is plotted. In addition, for consecutive developmental stages, a scatterplot illustrates the differentiation process. The arrows are for visualization and are not derived from the model. SMI, Scaled Mutual Information.

DR 23+, DR 1+, and DR 11- describe beta lineage formation, covering the cells from Fev+ beta cells to the beta cells producing insulin (Fig. 5e). Mnx1 and Pdx1 are among the top genes identified

for DR 23+, known to promote the differentiation of endocrine progenitors into beta cells [81, 82]. Notably, Pdx1 is also among the top genes for DR 1+, aligning with its role as a maintainer of the beta cell state [83]. DR 1+ highlights the earliest mature state of the developmental pancreas. Sytl4, one of the top identified genes for this factor, is a known beta marker gene regulating the maturation of beta cells [84]. Additionally, Klein et al. reported other relevant genes for this factor, such as Sntg1, Mapt, Tenm3, and Plppr1, as beta-specific marker genes [85]. Finally, DR 11- with Ins1 and Ins2 as top identified genes, indicates mature beta cells producing insulin. We illustrated scatterplots highlighting the latent geometry of the endocrine progenitor cells developing towards mature beta cells (Fig. 5e).

DR 9+ and DR 26- model the development towards Delta cells (Fig. 5f). DR 9+ is mainly identified by Hhex, required for delta cell differentiation [86]. Moreover, DR 26- is mainly identified by the Sst gene, secreted by delta cells.

DR 8+ primarily captures alpha and pre-alpha cells, along with a subset of epsilon and pre-beta cells. While Pou6f2 and Pde1a are computationally suggested markers for alpha cells [85], the rest of the genes are not specific to the alpha cells. The inability to fully disentangle the lineage towards alpha cells in DR 8+ may be attributed to the limited number of latent dimensions, insufficient signal in the sequenced data, or modeling limitations. Gcg is the main identifier of DR 4+, suggesting that this factor highlights alpha cells (Fig. 5g).

The lineage from endocrine progenitors towards epsilon cells is specified by DR 23+ and DR 27-. Among the top identified genes of DR 23+, Cdkn1a, Cacna2d1, Rgs17, Mboat4, Fam107b, and Tmem184c are reported as computational drivers of epsilon progenitors [85]. The presence of Ghrl in the nonlinear gene program of factor DR 27- indicates that this factor identifies the epsilon cell type. Additionally, Ascl1, Maged2, Ghrl, Mboat4, Arg1, and Lrrtm3, identified for this factor, were computationally derived as drivers for the epsilon population [85]. The scatterplot of DR 23 and DR 27, illustrates the differentiation of endocrine progenitors towards epsilon cells (Fig. 5h).

In conclusion, we explored DRVI's latent factors that identify processes related to differentiation toward terminal states. We found that DRVI effectively captures most of the developmental stages leading to these terminal states. The interpretability of DRVI allows identifying the relevant genes for each stage. Interestingly, without using additional time points, DRVI identified developmental stages with gene sets that align with those previously identified using optimal transport on additional time points [85]. Besides, a quantitative evaluation based on CellRank and scVelo showed that DRVI improved terminal state identification in this dataset.

Disentanglement allows for the analysis of interventions in latent space

To demonstrate DRVI capabilities on perturbational data, we applied it to a CRISPR screen dataset consisting of 104,339 K562 cells perturbed by 106 single-gene and 131 combinatorial perturbations [5]. Disentanglement enables the visualization, analysis, and differential testing of perturbations on interpretable latent factors, rather than the original high-dimensional noisy gene expression space. We exemplify this by showing three combinatorial perturbations, covering synergistic, epistatic, and mutual inhibitory effects. The CBL+CNN1 perturbation significantly increases DR 24+ (hemoglobin expression, Supplementary Fig. 18a) and DR 5+ (SLC25A37 iron importer, which facilitates heme synthesis, Supplementary Fig. 18b). This suggests hemoglobin overexpression, which was confirmed by flow cytometry [5]. In another example, the effect of ETS2 (DR 15-) is canceled when combined with DUSP9 (Supplementary Fig. 18c). Finally, the mutual inhibition between DUSP9 and MAPK1 can be inferred from the decreased response signatures of both genes (DR 15- and DR 8-) in the combined perturbation (Supplementary Fig. 18d).

From a quantitative point of view, the disentanglement metric already shows that DRVI outperforms other benchmarked models in isolating the effects of perturbations in individual latent dimensions without using guide RNA information as supervision (Fig. 3b). To gain a deeper understanding of

how well different models preserve the underlying geometry and can identify perturbation effects, we focus on IRF1 and SET perturbations. Accordingly, we determined the two factors with the highest mutual information with IRF1 and SET perturbations for DRVI and the other benchmarked models. Only DRVI and DRVI-AP, and to some extent ICA, scVI-ICA, and LIGER were capable of identifying perturbations in distinct latent factors and displaying additive property in their latent spaces (Supplementary Fig. 19), with DRVI outperforming ICA by 12% and 2% in SMI scores for IRF1 and SET, respectively. This is while SET is one of the perturbations with the highest relevance to a factor in the scVI model.

DRVI's learned factors saturate when providing more than enough latent dimensions

So far, we have demonstrated DRVI's use as a tool for discovering underlying biological processes. However, the exact number of these processes and, thus, latent dimensions is mostly unknown. Here, we examine the effect of the number of latent dimensions in DRVI.

To start, we applied DRVI to the immune dataset studied in Fig. 2 using various numbers of latent dimensions. Interestingly, we observe that when increasing the number of latent dimensions K, some factors start to vanish: To quantify this, we examined the maximum absolute value and standard deviation of each latent factor for each model (Supplementary Fig. 20a). We clearly observe a group of vanished latent factors characterized by a very low maximum absolute value and low standard deviation. Accordingly, we simply define a factor (dimension) as vanished if the maximum absolute value in that factor is less than 1. These factors have a negligible impact on the model's latent space variations and reconstruction capabilities. This phenomenon is also observed in other KL-regularized methods [87]. The models trained on the immune dataset use all latent factors until $K = 32$ and saturate at $K = 128$, employing 57 non-vanished latent factors (Supplementary Fig. 20b). We evaluated the disentanglement by the LMS-SMI metric and integration quality by total scIB score. Providing more latent factors does not significantly degrade disentanglement or integration quality (Supplementary Fig. 20c,d).

We repeated this procedure for the remaining datasets. Except for the very complex whole-organism Daniocell example, the number of non-vanished latent dimensions saturates before 512 dimensions (Supplementary Fig. 20d). Daniocell, as an extreme case with 156 cell identities and a more complex network, also saturates, but with a larger number of latent dimensions (Supplementary Fig. 21d,e). The disentanglement metric, LMS-SMI, improves for all datasets while the number of non-vanished factors increases. This is while we observe no significant drop in integration quality using wider networks (Supplementary Fig. 20f,g).

The number of required dimensions is influenced by data complexity factors such as the number of cell types, features, and developmental stages. To confirm that this phenomenon persists when using all genes, we trained DRVI on the immune data with all genes. In addition, we increased the number of hidden neurons in the encoder and decoders from 128 to 512 to force no limitation by these networks. The results indicate that while training on the full gene set requires more latent factors, the saturation property still holds (Supplementary Fig. 21a-c).

Altogether, the number of utilized latent dimensions saturates, with no significant decline in disentanglement and integration performance, when DRVI is provided with a sufficient number of latent dimensions. While this phenomenon is not specific to DRVI, it facilitates choosing the latent dimensionality, when the true number of biological processes is unknown. Therefore, we recommend starting with a reasonable number of latent dimensions and iteratively increasing this number until most dimensions vanish. However, given that none of the baseline methods were designed to handle a large number of latent dimensions, we used 32 or 64 dimensions in our benchmarks to maintain fairness.

The identified programs are consistent and transfer across datasets

To demonstrate the generalizability of DRVI, we individually trained four DRVI models, each with 128 latent factors, on four Human PBMC datasets in CellHint Blood data (Supplementary Fig. 22a). These datasets comprise a total of 336k cells with expert-curated annotations [88]. We observed that the number of non-vanished factors ranged from 31 to 43 (Supplementary Fig. 22b,c), with the variability likely stemming from minor biological and technical differences between the datasets. We then directly applied each model trained on one dataset to the other three and measured the disentanglement capabilities. Interestingly, the disentanglement quality remained intact, which suggests the model is generalizable across datasets with similar underlying biology (Supplementary Figs. 23a, 24). Additionally, we assessed the consistency of the identified programs (the sorted set of relevant genes) across the four models using the Rank-Biased Overlap (RBO) similarity score [89]. The results indicate a favorable one-to-one correspondence for most of the identified programs (Supplementary Figs. 23b). One example is a consistent signature for Tcm/Naive cytotoxic T and Tem/Temra cytotoxic T cell types across these four models (Supplementary Figs. 23c).

DRVI extends to further data modalities

We primarily demonstrate the utility of DRVI in single-cell transcriptomics data. However, DRVI easily extends to other modalities, such as single-cell DNA accessibility (scATAC), to obtain disentangled representations. For this purpose, DRVI supports Negative Binomial, Poisson, and Gaussian (with MSE loss) noise models. As an example, we applied DRVI to the single-cell ATAC modality of the NeurIPS 2021 multimodal data integration challenge [90]. Specifically, we employed the Poisson noise model on ATAC fragments, which has recently been shown to be the proper assumption for this modality [24]. To assess disentanglement and integration, we compared DRVI against DRVI-AP, PoissonVI [24] (a recent best-performing method), and PeakVI [91] (a widely used method in scATAC-seq integration). As in scRNA-seq, we observe that DRVI combines disentaglement with robust integration performance: DRVI improves disentanglement with a 97% improvement in LMS-SMI and a 167% improvement in LMS-SPN over the next best-performing model (Supplementary Fig. 25c). Additionally, DRVI significantly outperforms PoissonVI and PeakVI in total scIB integration score (0.63 compared to 0.60), demonstrating its applicability to the ATAC modality (Supplementary Fig. 25d).

Discussion

We have presented DRVI, an unsupervised deep generative model to learn disentangled representations of single-cell omics data. Disentanglement, as the main focus of DRVI, allows the representation of distinct biological processes in different latent factors. Nonlinearity improves integration and disentanglement quality and makes the model scalable to cellular atlases. With the presented examples and metrics, we demonstrate the utility and value of disentanglement in single-cell omics analysis. We have shown how disentanglement can describe cell types, capture biological processes beyond cell types, identify developmental stages, and isolate unwanted variations like dissociation stress response and cell cycle in different latent factors. Disentangled representations are additionally useful for perturbation modeling, in particular for combinatorial effects. We additionally highlighted the identification of cell-type-specific variations that are challenging to disentangle using linear methods yet captured by DRVI as a nonlinear disentanglement model.

In contrast to clustering in latent space - which provides separate groups based on the dominant signal - disentangled representations allow us to capture dominant and secondary signals in parallel in different latent factors - which enables the identification of variations beyond cell types. Accordingly, considering all latent factors along their non-linear gene programs may enable the building of a dictionary of active biological processes in data. DRVI's ability to generate sparse latent variables

and remove unnecessary factors suggests its potential for scaling to larger, more complex datasets. A future direction may be to expand the model to multiple atlases and create a comprehensive, unified dictionary.

DRVI's additive decoder architecture, combined with LSE pooling, enables interpretability and disentanglement. It imposes an inductive bias independent of the learning framework and training paradigm, applicable to a wide range of generative models in single-cell biology. By adopting DRVI's decoder, future models may achieve enhanced interpretability and disentanglement across diverse biological applications, regardless of noise model, prior distribution, or training paradigm choices. An additional feature of this approach is that it enables opening up the black box of fully-connected autoencoders, thereby enabling a higher explainability of the results and errors. Exploring additional nonlinear pooling functions, their theoretical properties, and practical performance may be another line for future research.

Besides the advantages mentioned, DRVI's unique properties make it a promising choice to extend into further modalities and applications. By extending DRVI to multiple modalities, users can interpret each disentangled factor as a topic and leverage these topics as a bridge to analyze the relationship between modalities, such as the regulation of transcription by chromatin accessibility [92, 93]. We have observed that DRVI's latent factors represent consecutive developmental stages as cycles, suggesting its potential for improved modeling of spliced/unspliced counts in velocity data [80]. Moreover, the model's additive property indicates its potential for modeling combinatorial effects, as seen in perturbation and CIM-seq data [5, 94].

Spatial omics technologies have become increasingly popular for analyzing cellular arrangements and interactions within the spatial context [95, 96]. Given DRVI's effectiveness in identifying variations, we anticipate its ability to identify biologically meaningful regions. In addition, in many spatial technologies, counts are accumulated per spot or segment [97]. We believe DRVI's assumption of additivity in count space makes it a promising framework for modeling such data, potentially leading to improved deconvolution and resilience to imperfect segmentation - the underlying framework of additive decoders has indeed been proposed in the imaging domain, which often has such compositionality questions. The model will require further extensions to leverage spatial information for these applications.

Despite promising results, our work, as a step towards unsupervised disentangled representations in single-cell omics, has limitations. DRVI is more resource-intensive than entangled frameworks like scVI or DCA. While highly effective for current individual atlases, scaling the architecture to tens of millions of cells, all genes, and thousands of latent variables remains a challenge for future optimization. Although we established a systematic framework for evaluating disentanglement, its reliance on supervised proxy variables presents limitations. Cell-type annotations are incomplete, often coarse-grained, and insufficient to evaluate models beyond cell-type variations. Additionally, perturbations are often noisy and redundant and cover multiple processes, making it impossible to achieve high metric scores. The development and standardization of metrics based on well-curated biological processes are crucial for advancing disentangled representation learning.

In conclusion, DRVI enables unsupervised disentangled representation learning in single-cell omics. Researchers can thus explore complex biological processes beyond traditional cell-type annotations and build downstream analyses upon an interpretable latent space.

Availability of data and material

Code availability

DRVI is available as a package at `http://github.com/theislab/drvi`. The code to reproduce the results and figures is available at `http://github.com/theislab/drvi_reproducibility`.

Data availability

All datasets analyzed in this manuscript are public and have been published in other papers. We have referenced them in the manuscript and provided the pre-processing steps at http://github.com/theislab/drvi_reproducibility.

Author Contributions

A.A.M. and F.J.T. conceived the project. A.A.M. designed and implemented the algorithm. A.A.M. performed experiments and analyses. A.A.M. and F.J.T. wrote the manuscript. F.J.T. supervised the work.

Acknowledgments

We thank Malte D. Luecken and Lisa Sikkema for providing constructive feedback on this work and for their expertise in HLCA analysis, Sara Jimenes and Dominik Klein for their valuable insights and discussions on the pancreas data, Leander Dony for a fruitful discussion on the required number of latent dimensions, Lisa Sikkema, Alessandro Palma, Mojtaba Bahrami, and Anastasia Litinetskaya for their constructive criticism and valuable feedback on the manuscript, Mahya Sadat Aramoon for helping in the conceptual visualizations, Mohammad Lotfollahi for discussions on related works, Philipp Weiler for suggestions on the choice of developmental datasets, Daniel Strobl for suggestions on enrichment analysis, and Michaela Müller for a discussion on evaluation framework. Finally, we thank all the members of the Theislab for their insightful discussions and support.

Funding

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) through the Leibniz Prize and project 458958943 (grant number 5010338). A.A.M. is a member of the ELLIS PhD Program of the European Laboratory for Learning and Intelligent Systems (ELLIS) Society. F.J.T. acknowledges support from the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI (grant number ZT-I-PF-5-01) and from the Leibniz Prize from DFG.

Competing interests

F.J.T. consults for Immunai, CytoReason, Genbio, and Valinor Industries, and has ownership interest in RN.AI Therapeutics, Dermagnostix, and Cellarity.

Methods

DRVI: Disentangled Representation Variational Inference

Disentanglement refers to the ability of a trained decoder to identify and recover the underlying latent factors of a ground-truth generative process into individual trained latent factors [33]. In single-cell omics this includes (but is not limited to) recovery of cell types, inflammation processes, and cell cycle stages as latent factors. While some disentangled representation learning frameworks assume statistical independence of underlying factors, aligned with recent advancements [33, 37] and the behavior of biological processes, we do not make such an assumption (Supplementary Note 3 and Supplementary Fig. 26).

DRVI is an unsupervised deep generative model that employs log-sum-exp (LSE) pooling on top of additive decoders to disentangle latent factors. The additive decoder architecture splits the latent space into individual slots, decodes them into the output space, and sums them up [37]. This architecture has been shown to provide provable disentanglement under specific assumptions. However, we practically observe that without proper modifications combining current variational autoencoders in single-cell omics with additive decoders does not result in disentangled representations (DRVI-AP baseline). Thus, We replace the sum aggregation in additive decoders with LSE pooling, which results in higher order interactions within slots and first order interactions across slots. The interaction asymmetry framework explains why this modification works: disentanglement can be achieved when each slot has higher order interactions within itself and lower order interactions across slots [98]. From a mechanistic point of view, LSE pooling assumes that individual biological processes add up in count space (Supplementary Note 2 and Supplementary Fig. 27). To increase the usability and highlight the utility of our architecture, we built our model based on the scvi-tools framework and minimized the deviation from the scVI model [19, 22, 99].

In the following sections, we first define the problem of unsupervised disentanglement. Next, we present the generative model of DRVI and an in-depth explanation of its decoder and pooling function. Finally, we discuss additional details, including interpretability and the evaluation framework.

Definitions and Notations

Here, we use capital letters to represent random variables (e.g., X), lowercase letters to represent realizations (observations) of random variables (e.g. x is a specific value that X takes), and uppercase bold letters to represent a matrix of observations (e.g. $\boldsymbol{X}$, where rows $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are independent and identically distributed (IID) observations). We may also represent constants (such as the number of observations) with capital letters (e.g. N).

To define the disentanglement, we consider a ground truth (GT) generative process that generates the observed samples. Disentanglement implies a universal one-to-one correspondence between learned and GT latent factors exists. The definitions are adapted from additive decoders [37].

Generative process: A generative process is defined based on its latent space $\mathcal{Z} \subset \mathbb{R}^K$, a probability distribution $p_z(z)$ over $\mathcal{Z}$, and a decoder $f : \mathcal{Z} \to \mathbb{R}^M$ that maps latent variable z to the observational space $\mathbb{R}^M$.

Partition: Given a set $\mathcal{A}$, we call a set $\mathcal{P}$ to partition $\mathcal{A}$ when elements of $\mathcal{P}$ are mutually exclusive ($\forall B \neq B' \in \mathcal{P}, B \cap B' = \emptyset$) and their union covers all elements of $\mathcal{A}$ ($\bigcup_{B \in \mathcal{P}} B = \mathcal{A}$).

Projection: Projection of a point $z \in \mathbb{R}^K$ into $B \subseteq \{1, \ldots, K\}$ is defined as $z_B := [z_{d_1} | \ldots | z_{d_l}] \in \mathbb{R}^{|B|}$ where d_i are ordered elements of B ($B = \{d_1 < \cdots < d_l\}$).

Disentanglement: (Adapted from additive decoders [37], Definition 3) Assume that we have a latent space $\mathcal{Z} \subset \mathbb{R}^K$, a partition of latent dimensions $\mathcal{P}$, a GT generative process defined by a decoder f and a learned decoder $\hat{f}$ that is optimized on the samples generated by f. We call $\hat{f}$ to be disentangled with respect to GT decoder f and partition $\mathcal{P}$ if there exist a permutation of latent dimensions π and a function $v : \mathbb{R}^K \to \mathbb{R}^K$ such that for all $z \in \mathcal{Z}$ in the latest space:

- v does not mix different blocks of partition $\mathcal{P}$. Formally, for all distinct blocks $B \neq B' \in \mathcal{P}$ and for all $i \in B, j \in B'$, the (i, j)-th element of the Jacobian of v at point z denoted as $(J_v(z))_{i,j}$ is zero.
- $f(z) = \hat{f}(\pi(v(z)))$, where $\pi(z_1, \ldots, z_K) = (z_{\pi(1)}, \ldots, z_{\pi(K)})$.

In other words, $\hat{f}$ should be identical to f up to blockwise modifications and a permutation of blocks.

Disentanglement is called dimension-wise when it is achieved with respect to the complete partition of the latent space dimensions ($\mathcal{P} = \{\{1\}, \{2\}, \ldots, \{K\}\}$). This is the most powerful and interpretable

form of disentanglement, as it implies that the trained decoder should identify each individual underlying latent factor.

Statistical independence: Factors (blocks) of a latent space $\mathcal{Z} \subset \mathbb{R}^K$ with probability distribution $p_z(z)$ over $\mathcal{Z}$ are independent if for any two blocks $B \neq B'$ the following criteria holds:

$$p_z(Z_B = z_B, Z_{B'} = z_{B'}) = p_z(Z_B = z_B) \times p_z(Z_{B'} = z_{B'})$$

We note that unlike disentanglement, statistical independence is highly influenced by the distribution of latent samples $p_z(z)$, which is usually defined based on the observational samples.

Considering the above definitions, there can be learned latent spaces which are disentangled but not independent, and those that are independent but not disentangled (Supplementary Note 3). Considering the complex underlying causal relationships between biological processes, we define disentanglement based on a one-to-one mapping between the ground-truth underlying processes and learned factors (Properties 1 and 2 of [33]). This assumption does not require independence of underlying factors and supports causal settings where one factor affects another, although we do not explicitly learn any causal model in the latent space (Supplementary Note 3).

The Generative Model of DRVI

A pooling function defined by $\psi, \sigma : \mathbb{R} \to \mathbb{R}$ aggregates a set of real valued numbers $y_1, \ldots, y_T$ by $\sigma\left(\sum_{1 \leq t \leq T} \psi(y_t)\right)$. Pooling functions can also be applied element-wise to a set of vectors or tensors.

Assume that we have an observed count matrix $\boldsymbol{X} \in \mathbb{N}^{N \times M}$, where N is the number of cells and M is the number of genes. Optionally, we can consider a covariate vector $\boldsymbol{C} \in \mathcal{R}^{N \times R}$, where R is the dimensionality of the covariates for each cell. For example, in batch correction frameworks, $\boldsymbol{C}_i$ could be a one-hot vector representing the batch identity if modeled as in scVI [19], or an embedding vector if modeled as in scPoli [51] (DRVI accepts both conventions). Then, the goal is to discover latent representation $\boldsymbol{Z} \in \mathcal{R}^{N \times K}$ where K is the latent space dimension for the observed data. The generative process is defined by the pooling of multiple decoders that map latent space blocks to parameters of the noise distribution. Specifically:

$$\begin{aligned} &Z \sim Normal(0, I) \\ &\log \mu = f(Z, C) = \sigma\left(\sum_{B \in \mathcal{P}} \psi \circ f^{(B)}(Z_B, C)\right) \\ &X_g \sim Dist(\log \mu_g + \log L, \log \theta_g) \end{aligned} \tag{1}$$

Where f is the decoder, $\mathcal{P}$ is a partition of the latent space dimensions, $f^{(B)}$ denotes the additive decoder subunits defined per block (referred to as decoder subnetworks), σ and ψ form pooling function, $Dist$ models the noise distribution, θ_g is a parameter indicating gene-wise dispersion or variance, L corresponds to the random variable indicating library size of the cell, and Z_k denotes kth dimension of Z. DRVI users have an option to alternatively define θ based on batches or optimize it like μ using variational inference. To focus on the architectural contribution, we fixed θ based on genes, a commonly used and simpler approach. Likewise, we deviate from the generative modeling and set L to the observed values of the library size rather than incorporating it into the generative model, which is aligned with standard practices in the field.

DRVI accepts Normal, Poisson, and Negative Binomial distributions as noise models. When using the Normal noise model, the observed data (x) should be log-transformed. The transformations can be as simple as $log(1 + x)$, although users are encouraged to apply other normalization techniques that maintain the validity of the normal noise assumption, such as additional library size normalization. For the Poisson noise model, the dispersion parameter is discarded. In Poisson and Negative

Binomial noise models, all parameters are transformed using the exponential function. Overall, $Dist(\log\mu, \log\theta)$ is defined as:

$$\begin{cases} Normal(\log\mu, \log\theta) & \text{if } Dist = \text{Normal} \\ Poisson(\exp(\log\mu)) & \text{if } Dist = \text{Poisson} \\ NB(\exp(\log\mu), \exp(\log\theta)) & \text{if } Dist = \text{Negative Bonimial} \end{cases} \tag{2}$$

We primarily parametrize the distributions in log space to ensure a correct and clear formulation within our framework. This approach also emphasizes that all other methods calculate their parameters in log space and transform them back to count space in the last step.

Fitting the generative process given observed data involves maximizing the log-likelihood function $\log \mathbb{P}_\theta(\boldsymbol{X}|\boldsymbol{C}) = \sum_n \log \mathbb{P}_\theta(\boldsymbol{X}_n|\boldsymbol{C}_n)$ where $\boldsymbol{X}$ is the matrix of observations, $\boldsymbol{C}$ indicates the matrix of covariates (such as technical batch), and θ represents all parameters in the model. Although the log-likelihood decomposes into sample log-likelihoods, maximizing each term requires integration over all Z values, which is intractable. We employ Variational Bayes, which constructs an approximate posterior probability distribution function $q_\phi(Z|\boldsymbol{X}, \boldsymbol{C})$ parameterized by ϕ [100]. This distribution is typically formulated as a simple multivariate Gaussian distribution. The optimization objective, known as the evidence lower bound (ELBO) , is formulated as:

$$\begin{aligned} &\log \mathbb{P}_\theta(\boldsymbol{X}|\boldsymbol{C}) \\ &\quad = \sum_n \log \mathbb{P}_\theta(\boldsymbol{X}_n|\boldsymbol{C}_n) \\ &\quad \geq \sum_n \mathbb{E}_{Z\sim q_\phi(Z|\boldsymbol{X}_n,\boldsymbol{C}_n)}\Big[\log \mathbb{P}_\theta(\boldsymbol{X}_n|Z, \boldsymbol{C}_n)\Big] - D_{KL}\Big(q_\phi(Z|\boldsymbol{X}_n, \boldsymbol{C}_n) \;\Big|\Big|\; \mathbb{P}(Z)\Big) \\ &\quad = \sum_n -\mathcal{L}(\theta; \boldsymbol{X}_n, \boldsymbol{C}_n) = -ELBO. \end{aligned} \tag{3}$$

Similar to other frameworks, the ELBO is optimized by constructing an autoencoder where $\mathbb{P}_\theta(X|Z, C)$, $q_\phi(Z|X, C)$, and $\mathbb{P}(Z)$ correspond to the decoder, the encoder, and the prior distribution, respectively [19, 51, 101]. Stochastic gradient descent is used to optimize the negative ELBO, and the reparametrization trick is employed to sample from the approximate posterior q [100, 102].

Additive Decoders, Pooling, and Disentanglement

Theoretically, the additive decoder and interaction asymmetry frameworks provide conditions under which disentanglement is guaranteed [37, 98]. The most critical of these are first-order interaction asymmetry and first-order sufficient independence within the interaction asymmetry framework [98]. These conditions are typically proven on very small toy examples [37] or heuristically optimized [98]. Here, we adopt a similar approach and demonstrate that utilizing LSE pooling heuristically encourages higher-order interactions within splits, and the additive decoder architecture keeps interactions across splits to first order (Supplementary Note 5).

Average Pooling: Consider the case where $\sigma(x) = x/K$ and ψ is identity function:

$$f(z) = \frac{1}{|\mathcal{P}|} \sum_{B\in\mathcal{P}} f^{(B)}(z_B). \tag{4}$$

Then $\psi'(x) = 1$ and $\psi''(x) = 0$.

Considering that shallow MLP layers are often used in the representation learning for single-cell omics, decoder subnetworks in this case do not produce enough nonlinearity and higher-order interactions within splits. In the extreme case where the decoder subnetworks are linear, the decoder reduces

to an affine transformation, making disentanglement impossible due to the well-known unidentifiability of linear ICA under independent Gaussian factors. We demonstrate that, although additive decoders with average pooling exhibit slightly better disentanglement than their non-additive counterparts, they still fall short compared to LSE-pooled additive decoders (see Results).

LSE Pooling: Consider the case where $\sigma = \log$ and $\psi = \exp$:

$$f(z) = \log \sum_{B \in \mathcal{P}} \exp\left(f^{(B)}(z_B)\right). \tag{5}$$

The $\sigma = \log$ term in this case can be disregarded through a suitable transformation of the outputs (Remark 1 of additive decoders [37]). Considering $\psi \circ f^{(B)} = \exp\left(f^{(B)}\right)$ as decoder functions and ignoring σ, our generative process can be considered as an additive decoder where a shared activation function ($\psi = \exp$) is used at the end of each decoder subnetwork. The exp activation at the end of each decoder creates higher-order interactions within each split, and the additive decoder architecture keeps interactions across splits to first-order (Supplementary Note 5). Clearly, $\psi = exp$ is effective and cannot be ignored.

Mechanistic interpretation

An important aspect of LSE pooling is its mechanistic interpretation. As described before, the decoder of DRVI (as in most other methods) models the mean parameters of the output distributions in the log-transformed space. Using average pooling means that predicted mean tensors from different factors are added up in log space, equivalent to the product of the tensors in count space. Formally let $\log S_g^{(k)}$ be the output of kth decoder for gene g then for average pooling we have:

$$\begin{aligned} \log \mu_g &= \sum_k \log S_g^{(k)} \\ \Rightarrow \mu_g &= \exp\left(\sum_k \log S_g^{(k)}\right) = \prod_k S_g^{(k)}. \end{aligned} \tag{6}$$

The same phenomenon can be concluded in current matrix factorization and, to some extent, autoencoder models (Supplementary note 6). While this behavior causes no problem when active genes in different factors do not overlap, it makes no sense when a gene is regulated by multiple factors.

Interestingly, using the LSE pooling, as described earlier, corresponds to the aggregation of predicted mean tensors by the LSE function. Since these parameters are defined in log space, their aggregation by LSE corresponds to an addition in the original count space. Formally let $\log S_g^{(k)}$ be the output of kth decoder for gene g, then for LSE pooling we have:

$$\begin{aligned} \log \mu_g &= \log \sum_k \exp\left(\log S_g^{(k)}\right) \\ \Rightarrow \mu_g &= \sum_k \exp\left(\log S_g^{(k)}\right) = \sum_k S_g^{(k)}. \end{aligned} \tag{7}$$

This means using the LSE pooling in the framework of additive decoders corresponds to the assumption that "activities induced by different factors (mechanistically independent biological processes) sum up in the count space." This assumption aligns with the inherent additivity of transcribed genes and is consistent with additive technical effects such as doublets and background expression commonly observed in single-cell omics. (Supplementary Fig. 27 and Supplementary Note 2).

Modeling Details

We employ an MLP network to model the approximate posterior as an encoder. This encoder accepts log-transformed count data and outputs mean and variance vectors of a multivariate Gaussian

distribution with a diagonal covariance matrix. Each layer of the MLP consists of an affine transformation (multiplication by a weight matrix followed by the addition of a bias vector), an activation function, and a normalization layer. Dropout is applied to input gene values and hidden layers. We employ ELU as the default activation function and layer normalization (without elementwise affine transformation) as the default normalization layer. The final layer of the encoder is a simple affine transformation without any additional units. We use the same settings to model each decoder subnetwork $f^{(k)}$.

In the DRVI framework, parameters between different decoder subnetworks can be shared in several ways: sharing nothing, the last affine transformation, or all affine transformations. We note that layer normalizations without elementwise affine transformation do not have any state to share. In practice, even sharing all parameters works perfectly and is the default option.

The outputs of the decoder subnetworks are finally aggregated to form the parameters of the output distribution. Users can choose between average or LSE pooling. As discussed earlier, we utilize LSE pooling, implemented with the *logsumexp* function, as the default and correct aggregation method.

DRVI can handle categorical covariates with one-hot encoding and vector embedding strategies [19, 51]. For the sake of fair comparison with benchmarked unsupervised methods, we use one-hot encoding in all the analyses. However, we encourage users to take the second approach for studies with a high number of batches as it is shown to perform better [51]. It is worth mentioning that DRVI accepts multiple categorical covariates by concatenating their representations.

Choosing the appropriate noise model depends on the specific characteristics of the data. The Negative Binomial distribution is widely accepted for single-cell RNA sequencing (scRNA-seq) data and is extensively used in applications ranging from differential expression analysis to integration models [19, 50, 103–105]. Recent studies have demonstrated that the Poisson distribution may be a more suitable choice for single-cell ATAC sequencing (scATAC-seq) data [24]. Still, many methods such as scGEN and sysVI use normalized data with Normal distribution while performing well in their tasks [20, 87]. DRVI supports Negative Binomial, Poisson, and Normal noise distributions. For the latter users should provide normalized data rather than raw counts. We recommend using Negative Binomial for scRNA-seq and Poisson for modeling fragments in scATAC-seq data.

In current generative models, parameters are transformed from log space to count space, after which the Negative Binomial distribution's likelihood is computed based on these count-space parameters. To enhance numerical precision, we introduce a novel parametrization of the Negative Binomial likelihood that is defined based on $\log \mu$ instead of μ (see Supplementary Note 7 for mathematical details). Log-space parametrization of the Negative Binomial distribution is the default option to calculate the likelihood in DRVI, but users can still use scVI's parameterizations.

DRVI adjusts the final predicted mean parameter (after aggregation) to account for the library size, similar to scVI's default behavior of ensuring that the sum of mean parameters equals the library size. This default behavior can be disabled.

Finally, DRVI users can introduce multidimensional splits by adjusting a model parameter. We set the number of splits by default to the number of latent dimensions. It is important to note that while the DRVI framework supports latent partitions of arbitrary size, our focus here has been on the full partition case where each latent factor is disentangled. The reason is that full partition disentanglement is the most powerful scenario in terms of disentanglement, and interestingly, it also aligns well with the characteristics of single-cell count data. The additive assumption in the full partition case may seem restrictive. However, the Kolmogorov–Arnold representation theorem states that the sum of one-dimensional functions followed by a trainable pooling function can approximate any continuous multivariate function [106, 107]. This theorem doesn't directly apply to our model due to the employment of a fixed shared pooling function for all output variables in the decoder. Nevertheless, we mentioned this theorem to provide additional context for interested readers.

Interpretation and ordering of learned factors

Learned latent representations may present known or unknown abstract concepts. Interpreting latent factors in terms of the genes involved helps understand unknown abstract concepts and find explanations for known ones. In factorization models, the loadings (weights of the low-rank matrices) are interpreted as the effect of each gene on each factor. However, in nonlinear models (such as scVI), interpretability is challenging since the dependencies are nonlinear, and the effect of each latent factor also depends on the values of other latent factors. While the nonlinearity challenge still exists in DRVI, the additive architecture prevents interactions across latent factors, and allows interpretability as each latent factor is decoded separately until the final pooling function. Altogether, unlike autoencoders with MLP decoders, where the effect of one factor is highly coupled and dependent on all other latent factors, DRVI latent factors are decoded separately, making them inherently individually interpretable. Specifically, for each latent factor with value z_i, the ith subnetwork decodes values right into the count space, and $f^{(i)}(z_i)$ is the effect of this factor in count space. Considering their mechanistic independence, we traverse each latent separately to infer top genes relevant to each latent factor (Supplementary Note 8).

Unlike PCA, where principal components are ordered according to the explained variance, the latent factors learned by autoencoders and many other latent space models are inherently unordered. Nevertheless, we take advantage of the additive architecture and apply heuristics to sort the latent factors based on their contribution to the reconstruction (Supplementary Note 9).

Hyperparameters

We explored various choices of normalization functions, activation functions, noise models, parameter sharing between decoder subnetworks, and aggregation functions on the immune dataset. Based on our findings, we selected layer normalization (stateless), ELU activation function, parametrized negative binomial noise distribution (on RNA), full parameter sharing between decoder subnetworks, and exponential aggregation function. We then applied these settings uniformly across all other datasets, without extensive hyperparameter tuning. For the Daniocell dataset as an extreme case consisting of 156 cell identities, only the last layer of the decoder subnetworks is shared. The parameters of the deep learning models (DRVI, DRVI-AP, and scVI) were kept consistent where possible. We used the wandb platform to keep track of the experiments [108].

For smaller datasets, we used 32 latent dimensions with one hidden layer in both the encoder and decoder. For larger datasets, we employed 64 latent dimensions and deeper networks for the encoders and decoders. Due to its complexity, the CRISPR screen dataset required an even deeper architecture. Detailed information on the number of latent dimensions and the hidden layer architecture for autoencoder methods used in each study can be found in Supplementary Table 3.

Metrics

Since DRVI aims to bridge the gap between disentangled linear models and flexible non-disentangled methods, we evaluate its performance from two perspectives: disentanglement and integration. We assess the integration quality of multi-batch datasets based on the scIB metrics framework.

Details on these metrics can be found at [52]. As in the original study, the total score is derived by averaging biological conservation and batch integration scores with weights of 0.6 and 0.4, respectively.

Disentanglement refers to the existence of a one-to-one correspondence between the GT processes and the learned latent factors. Given a one-to-one correspondence, one can use an appropriate similarity function to measure how much each ground truth process is similar to the corresponding latent factor. Accordingly, we measure disentanglement by finding the one-to-one correspondence

that maximizes the average pairwise similarities. This category of metrics is known as the Latent Matching Score (LMS). Since the number of GT processes is unknown to the models, the number of GT processes may differ from the number of latent dimensions. Therefore, we define the LMS metrics by finding the best non-overlapping one-to-one matching between the GT processes and the latent factors [37]:

$$\text{LMS-}S(P, Z) = \frac{1}{|P|} \max_{\pi \in \Pi(P,Z)} \left[\sum_{p_l, z_k \in \pi} S(p_l, z_k) \right], \tag{8}$$

where S can be any similarity function comparing learned factors with respect to GT processes, P indicates the set of vectors corresponding to all ground-truth processes, Z is the set of latent factors, and Π is the set of all injective correspondences from P to Z.

In the following, we define three similarity functions based on Mutual Information and Nearest Neighbor Consistency.

Scaled Mutual Information (SMI): We use Mutual Information as a nonlinear metric to compare GT processes and latent factors. Since mutual information is bounded by the entropy of the GT process, we scale it so that the maximum value of $+1$ becomes achievable.

$$SMI(p_l, z_k) = \frac{\text{MI}(p_l, z_k)}{\text{H}(p_l)}, \tag{9}$$

where MI is the mutual information function and H is the entropy function. SMI is also known by other names such as proficiency or uncertainty coefficient [109].

Same-Process Neighbors (SPN): Consider a process with only two "on" and "off" states. Then, we would like to see the cells in the "on" state to form a group in a latent space. Accordingly, we measure the non-random probability for which a cell and its right neighbor are in the "on" state at the same time. So we have:

$$SPN(p, z_k) = \max\left(0, \frac{\mathbb{P}\big(p_j = 1 \big| p_i = 1 \wedge i, j \in \text{NG}(z_k)\big) - \mathbb{P}_{p_i \in p}\big(p_i = 1\big)}{1 - \mathbb{P}_{p_i \in p}\big(p_i = 1\big)}\right), \tag{10}$$

where $NG(z_k)$ indicates the set of all neighboring cells (no cell in between, not necessarily nearest neighbors) in factor k of the latent space. The function $SPN(p_l, z_k)$ measures the frequency of two neighboring cells being active simultaneously minus the probability of such an outcome when cells are shuffled randomly. This metric is also adjusted to reach $+1$ as its maximum. $LSM - SPN$ can be calculated efficiently in $O(|P|Kn\log(n))$ by sorting the cells with respect to each derived latent factor. Conceptually, the frequency of two neighboring cells being active simultaneously can be interpreted as an approximation to the precision of a 1NN classifier predicting the activation of a GT process from a given latent factor.

We use $LMS - SMI$ as the default metric to evaluate the disentanglement of a trained model with respect to known biological processes. While SMI accepts both discrete and continuous processes, SPN is designed specifically for on/off processes.

In addition to the LMS metrics, we use two other secondary metric families to better understand the model's behavior. To determine if a GT process is represented in at least one factor, we find the most similar factor in the derived latent space for each GT process and then average these similarities across all GT processes. This is called the Most Similar Averaging Score (MSAS). Unlike LMS, MSAS does not penalize the model for presenting multiple irrelevant biological signals within a single factor or representing a signal in multiple factors. Consequently, MSAS shows coverage and serves as an upper bound to LMS. We use the Most Similar Gap Score (MSGS), which builds on MSAS by adding a term that penalizes the presentation of a GT process in multiple factors. Specifically, MSGS calculates the average difference between the most similar factor and the second most similar factor for all GT processes, yet this score still lacks penalizing the presence of multiple

irrelevant biological signals within a single factor. When using the SMI similarity function, MSGS reduces to the Mutual Information Gap (MIG) metric as defined in Ref [65]. Formally, we have:

$$\text{MSAS-}S(P, Z) = \frac{1}{|P|} \sum_{p_l \in P} \max_{z_k \in Z} S(p_l, z_k) \tag{11}$$

$$\text{MSGS-}S(P, Z) = \frac{1}{|P|} \sum_{p_l \in P} \max_{z_k \in Z} \left[S(p_l, z_k) - \max_{z_{k'} \neq z_k \in Z} S(p_l, z_k) \right], \tag{12}$$

where S can be any similarity function, and P and Z are defined as before.

We primarily use the LMS metric family for evaluation and examine the MSAS and MSGS metric families to further evaluate model behavior. This choice is because MSAS and MSGS metrics do not penalize the presence of multiple irrelevant biological signals within a latent factor (Comparison of metric families in Supplementary Table 4). In addition, when using coarse-grained proxy variables for evaluation, the MSGS metric family can heavily penalize models that successfully disentangle finer, biologically meaningful subtypes (Supplementary Fig. 9). Finally, the MSGS metric family assumes statistical independence between latent variables. However, in biological systems, underlying GT processes often exhibit causal relationships or interdependencies (e.g., cell cycle is not independent of cell type, as not all cell types proliferate).

We may know a subset of GT processes in a few developmental datasets. However, GT processes are usually unknown. Therefore, we use perturbations in perturbational data and cell types in data with known cell types as proxies for GT biological processes. Each cell type undergoes specific biological processes, so we expect a disentangled model to represent each cell type in at least one distinct factor. Similarly, since perturbations activate specific biological processes, a disentangled model should be able to identify such processes and, to some extent, such perturbations in separate factors. We emphasize that while there is no one-to-one relationship between biological processes and these proxy variables (cell types and perturbations), we use them in the absence of a comprehensive list of biological processes to measure the effectiveness of the models in terms of disentanglement. This imperfection in the choice of the proxy variables results in limitations such as variability in the range of scores across datasets.

There can be a significant variation in the range of disentanglement scores across different datasets. This variation is influenced by both the complexity of the datasets and the choice of the proxy variable. The disentanglement task becomes more challenging as the number of underlying processes increases. For example, identifying 100 cell types is much harder than identifying 10 cell types. A more significant factor affecting both the range of observed metrics and the maximum achievable values is the use of imperfect proxy variables. For example, perturbations are often redundant, combinatorial, and partially effective. However, metrics expect the models to push all cells affected by each perturbation to an individual axis, which is impossible. The same happens when there are combinatorial effects involved. Another category of imperfect proxy variables is coarse cell-type annotations. When the cell-type annotation is coarser than the true biological variations, a well-disentangled representation identifies subtypes in different axes. However, disentanglement metrics expect to see all cells in the course grouping in one axis, leading to undesirably low metric values for better models. These are the reasons that the metric range varies between datasets. Although not perfect, using available supervised data, such as annotations and perturbations, is currently the best way to measure disentanglement.

In addition to disentanglement metrics, we used Rank Biased Overlap (RBO) similarity score to demonstrate the consistency of interpreted signatures across datasets. For two sorted sets S and T, RBO is defined as:

$$\text{RBO}(S, T) = (1 - p) \sum_{d=1}^{\infty} \frac{\text{overlap}(T_{1\ldots d}, S_{1\ldots d})}{d} p^{d-1} \tag{13}$$

Where $p \in (0, 1)$ is parameter controlling the importance of top elements and $\text{overlap}(T_{1...d}, S_{1...d})$ is the number of shared items between top d elements of S and T. We use $p = 0.9$ by default.

Baseline models

All baseline models are trained with the same number of latent dimensions as used for DRVI. For baselines not accepting count data, we used the log-transformed normalized gene expression provided by the dataset when available. Otherwise, we normalized the total counts per cell and log-transformed the data using scanpy [110, 111].

- **PCA** from scikit-learn [112], was applied to the log-transformed and normalized data. Prior to the transformation, the data was centered without scaling.
- For **ICA**, we applied the FastICA [113] from scikit-learn [112]. Since the number of samples exceeded the number of features across all datasets, we chose the whiten_solver="eigh" option. The data was centered before applying ICA, without any scaling.
- **scVI** is trained with the same number of epochs and hidden layers as DRVI (Supplementary Table 3). To allow for learning a more structured latent space and fair comparison, early stopping is not used. For **scVI-PCA** and **scVI-ICA**, we applied PCA and ICA to the latent spaces learned from the scVI model. PCA from the scikit-learn package with default parameters and FastICA from the scikit-learn package [112] with whiten_solver="eigh" are used.
- For **β-TCVAE** and **MICHIGAN**, we used the implementation provided by the MICHIGAN package [47]. Since default parameters yielded suboptimal performance, we optimized the key scalar hyperparameters (TotalCorrelation_lamb, GradientPenaly_lambda, MutualInformation_lamb) by evaluating values in the range $\{0.1, 1, 10, 100\}$. Consequently, we adopted the default settings with GradientPenaly_lambda=100, which we subsequently confirmed to perform better than the default parameters in almost all benchmarked scenarios.
- We performed **MOFA** on the log-transformed normalized data using the Python implementation in the mofapy2 library [38]. To induce sparsity in the factor loadings and enhance the interpretability of the latent factors, we enabled the spikeslab_weights option. Additionally, the convergence_mode was set to "fast".
- **scETM** [114] was trained on raw count data for the same number of epochs as scVI and DRVI (Supplementary Table 3), with batch information provided and default hyperparameters. The resulting topic mixture (θ) was used to define the latent space.
- For **LIGER**, we utilized the Python implementation in the pyLIGER package [115]. The model was trained on the count matrices, and the resulting normalized cell latent factors (H_norm) were used to define the latent space.

Simulated data

In addition to real-world datasets, we benchmarked models on two simulated datasets (see 'Datasets' and Supplementary Fig. 28). One dataset featured unique cellular processes per cell, and the other included up to four overlapping processes per cell group. While DRVI, ICA, MOFA, PCA, LIGER, and scVI-ICA performed well in the simpler non-overlapping scenario, only DRVI, ICA, and LIGER proved capable of accurately identifying overlapping processes in the second scenario (Supplementary Fig. 29).

Ablation Study

To further investigate the impact of modeling choices, we trained several DRVI variants through ablation studies. Details of these models are provided below:

- **DRVI-AP:** We trained this variant with average pooling instead of LSE pooling to highlight the practical importance of LSE pooling.
- **DRVI-APnoEXP:** Similar to DRVI-AP, this variant averages the decoder outputs to generate predictions. However, unlike scVI, DRVI, and DRVI-AP, which apply the softmax function (exp followed by $l1$ normalization) at the end – a step we hypothesize transfers parameters from log space to count space – we used the smooth softplus function to ensure the positivity of the negative binomial's mean parameter. This variant was trained to demonstrate that the limited effectiveness of average pooling is not due to the assumption that decoder outputs reside in log space.
- **DRVI-noShare:** The default configuration shares weights across all but the first layer of the decoder subnetworks. In this variant, we trained the model without any weight sharing between these subnetworks.
- **DRVI-2D:** This variant employs two-dimensional splits of the latent space. While less powerful than the default DRVI, it highlights DRVI's applicability in multidimensional split scenarios.
- **CVAE:** This variant shares the architectural settings of scVI but uses smooth ELU activations and the log-space parameterized negative binomial loss from DRVI (defined in Supplementary Note 7). Consequently, we can view this as a standard conditional variational autoencoder (CVAE) implemented within the DRVI framework, employing MLP decoders instead of DRVI's additive decoders. We trained this variant to enable a fair comparison of reconstruction performance.

The ablation results indicate that the nonlinear shared function prior to summation, implemented by $\psi = \exp$ in the LSE pooling, is the main critical point for achieving disentanglement in our settings. Furthermore, sharing weights within decoder subnetworks slightly enhances disentanglement capabilities (Supplementary Figs. 30 and 31).

Reconstruction Performance

We compare the negative log likelihood of the DRVI, DRVI-AP, and CVAE models as a measurement of reconstruction. The constrained decoder in DRVI leads to a slight decrease in reconstruction performance (Supplementary Fig. 32). However, the high biological signal conservation observed in scIB metrics and the strong disentanglement results indicate that DRVI forms rich latent representations in terms of known cell-type variations. To further investigate, we varied the target-KL weight from 0.01 to 10 (default 1.0) for DRVI and CVAE on HLCA and Immune datasets. Compared to the decoder architecture, target-KL was the primary driver of reconstruction performance decline (Supplementary Fig. 33a,e). We show that while disentanglement in CVAE was significantly affected by reducing this parameter, DRVI showed considerable robustness (Supplementary Fig. 33c,d,g,h). Thus, users seeking good reconstruction can reduce the target-KL weight without majorly compromising disentanglement.

Runtime Analysis

We benchmarked DRVI against scVI (for scRNA-seq) and peakVI (for scATAC-seq) as widely-used efficient models for dataset integration. We recorded the total runtime and maximum GPU memory allocation. We note that DRVI's runtime scales linearly with the number of cells. Accordingly,

we plotted the runtime divided by the number of cells and the number of epochs (runtime per cell per epoch). Our results indicate that despite DRVI's higher computational cost in terms of floating-point operations, its parallel and efficient implementation maintains runtime on the same order as scVI or peakVI up to a small scaling factor (Supplementary Fig. 34). Furthermore, for each dataset, the runtime for both the interpretability pipeline (gene scores per latent factor) and the latent factor sorting algorithm was measured, with a worst-case execution time of under 80 seconds (Supplementary Fig. 35). The variability between runs arises from the fact that sorting latent factors requires a full iteration of model evaluation over all cells, whereas the interpretability runtime depends solely on the number of latent dimensions, the number of features, and the complexity of the decoder. It is worth mentioning that DRVI is based on the scvi-tools framework and requires the same amount of CPU memory for training. The runtime experiments were performed on a GPU server equipped with two Intel(R) Xeon(R) Platinum 8480+ processors (totaling 112 cores and 224 threads at 2.0 GHz), 2 TB of memory, and eight NVIDIA H100 (80GB) GPUs. Each program execution was constrained to **a single GPU** and a maximum of **20 CPU threads**.

Datasets

In the following, we provide information on the simulated data and datasets used in this work (summarized in Supplementary Table 2. The pre-processing steps for each dataset are described. Additionally, the code to reproduce simulated data as well as the pre-processing steps are available at https://github.com/theislab/drvi_reproducibility.

Simulated

We simulated two single-cell gene expression datasets to evaluate representation learning methods under controlled conditions. Each dataset consists of $G = 1000$ genes and $N = 6000$ cells, each assigned to either a specific condition or a control. For each condition $c \in \{1, \ldots, C\}$, we sample a binary vector $\mathbf{y}_c \in \{0, 1\}^K$, where $K = 20$ is the number of latent biological processes and $y_{c,k} = 1$ indicates that process k is active in condition c. Each cell under condition c inherits the same binary process activity vector. Latent representations $\mathbf{z}_i \in \mathbb{R}^K$ are drawn independently for each cell i, where $z_{i,k} \sim \text{Beta}(10, 2)$ if $y_{c,k} = 1$, and $z_{i,k} = 0$ otherwise. Control cells are assigned zero latent activity ($\mathbf{z}_i = \mathbf{0}$). We ensure that all processes are represented across the dataset and control the degree of overlap across conditions by varying $D = \max \|\mathbf{y}_c\|_0$, the maximum number of active processes per condition. In one dataset, each condition is linked to a single active biological process ($D = 1$). In the other, four distinct processes ($D = 4$) may simultaneously be activated per condition, creating controlled overlap and compositional complexity.

Gene expression values are generated additively by mapping each active process to a sparse (and potentially overlapping) subset of genes using nonlinear monotonic cubic spline functions [116]. Specifically, the expression of each relevant target gene is computed individually using a smooth monotonic transformation of the latent factor, simulating process-specific gene regulation. To account for biological variability consistent with the negative binomial characteristics of single-cell data, we model each process-specific contribution with a Gamma distribution, $\text{Gamma}(\alpha = s, \theta = \mu/s)$, where μ is the derived process-wise mean and s is a shape parameter per gene sampled from $(1, 5)$. In addition to ground-truth-related variation, we introduce background noise by injecting gene-wise Gamma-distributed noise with shape and scale parameters randomly sampled from $(0.05, 0.1)$ and $(0.1, 10)$, respectively. The final expression profile for each cell is obtained by summing the noise vector and process contributions, followed by Poisson sampling. The code to generate simulated data is available in the reproducibility repository.

Immune Dataset

The immune dataset consists of 32,484 cells collected from 4 human PBMC studies, including 9 batches and 16 distinct cell types after pre-processing. The data is obtained from Luecken et al. (2022) [52, 117]. The "Villani" study was excluded due to its non-integer values. The dataset was subset to 2,000 highly-variable genes (HVGs) in a batch-aware manner [52].

The HLCA

The HLCA consists of a core dataset and an extended version [7]. We obtained the curated core part from cellxgene (https://cellxgene.cziscience.com/collections/6f6d381a-7701-4781-935c-db10d30de293), which includes 584,944 cells across 166 samples from 14 datasets. This dataset contains annotations at different levels. We used samples as the batch covariate and the finest-level annotations comprising 61 cell types to benchmark the methods. The dataset was subset to 1,996 HVGs originally used to construct the HLCA reference model available on Zenodo (https://doi.org/10.5281/zenodo.7599104).

CRISPR Screen Dataset

The CRISPR screen dataset, also known as the Norman perturb-seq dataset, comprises 104,339 single-cells of K562 cell line (including control cells) perturbed by 106 single-gene perturbations and 131 combinatorial perturbations [5]. The 238 unique perturbation signatures were used to benchmark methods. We downloaded the pre-processed data using the pertpy package [118]. The data was limited to the 5,000 HVGs originally provided in the pre-processed data.

PBMC Dataset

The PBMC dataset comprises 647,366 cells across 143 samples from 130 healthy and COVID-19 patients [63]. The data was collected at three sites, and we used the site as the batch covariate to account for the highest technical variation. We used the finest-level annotation containing 51 unique cell types for benchmarking the methods. The data was obtained from https://www.covid19cellatlas.org/ and was subset to 2,000 HVGs in a batch-aware manner.

Retina Organoid Dataset

The retina organoid data consists of 19,768 cells from fovea, 34,723 cells from periphery, and 43,857 retina organoid cells (total 98,348 cells) [10]. The cells come from 41 samples and are annotated into 21 unique cell types. Considering that here, we have two very different biological systems, we take the intersection between 3000 HVGs of the two systems, which results in 2001 HVGs shared between primary and organoid data [87]. The data is obtained from https://cellxgene.cziscience.com/collections/2f4c738f-e2f3-4553-9db2-0582a38ea4dc.

Developmental Pancreas Dataset

The developmental pancreas dataset consists of 3,696 pancreatic mouse cells during endocrinogenesis at embryonic day 15.5 [4]. The data was obtained from the scVelo package [80]. We used scVelo to pre-process and subset the data to the top 2,000 HVGs. The CellRank package provides a subset of this data with finer annotations [79, 119]. We transferred these finer annotations to the complete dataset and used the finest annotation comprising 15 unique cell states for benchmarking.

Daniocell Dataset

The Daniocell dataset is a comprehensive transcriptional atlas of early zebrafish development, comprising 489,686 cells from 62 stages during zebrafish embryogenesis [64]. The data covers 20 tissues and 156 identity clusters, which we used to benchmark the data. We obtained the h5ad object from https://zenodo.org/records/8133569 [120]. The pre-processing of this data involved marking data with the annotations provided in the original publication, removing author-identified doublets, and subsetting data to 2,000 HVGs.

NeurIPS 2021 Dataset

The NeurIPS 2021 Dataset contains nearly 120K single-cell paired RNA-ATAC and RNA-ADT (antibody-derived tags) measurements from the human bone marrow of 10 donors [90]. We used the ATAC modality of this dataset consisting of 69,247 cells from 13 batches covering 22 cell types. The dataset is retrieved from GSE194122 and subset to 14,865 highly variable peaks.

Blood Dataset

Blood data consists of 336k cells collected from 4 human PBMC studies, including 65 donors and 27 cell types in its expert-curated annotation [88]. The data is obtained from https://cellxgene.cziscience.com/e/d86edd6a-4b5d-437a-ad80-1fd976a5e23a.cxg/. The dataset was subset to 2,000 highly-variable genes (HVGs) in a batch-aware manner [52].

References

[1] Svensson, V., Vento-Tormo, R. & Teichmann, S. A. Exponential scaling of single-cell RNA-seq in the past decade. Nat. Protoc. **13**, 599–604 (2018).

[2] Angerer, P. et al. Single cells make big data: New challenges and opportunities in transcriptomics. Curr. Opin. Syst. Biol. **4**, 85–91 (2017).

[3] Farrell, J. A. et al. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. Science **360** (2018).

[4] Bastidas-Ponce, A. et al. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Development **146** (2019).

[5] Norman, T. M. et al. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. Science **365**, 786–793 (2019).

[6] Srivatsan, S. R. et al. Massively multiplex chemical transcriptomics at single-cell resolution. Science **367**, 45–51 (2020).

[7] Sikkema, L. et al. An integrated cell atlas of the lung in health and disease. Nat. Med. **29**, 1563–1577 (2023).

[8] Rood, J. E., Maartens, A., Hupalowska, A., Teichmann, S. A. & Regev, A. Impact of the human cell atlas on medicine. Nat. Med. **28**, 2486–2496 (2022).

[9] Kanton, S. et al. Organoid single-cell genomic atlas uncovers human-specific features of brain development. Nature **574**, 418–422 (2019).

[10] Cowan, C. S. et al. Cell types of the human retina and its organoids at Single-Cell resolution. Cell **182**, 1623–1640.e34 (2020).

[11] Buenrostro, J. D. et al. Single-cell chromatin accessibility reveals principles of regulatory variation. Nature **523**, 486–490 (2015).

[12] Zhang, K. et al. A single-cell atlas of chromatin accessibility in the human genome. Cell **184**, 5985–6001.e19 (2021).

[13] Thornton, C. A. et al. Spatially mapped single-cell chromatin accessibility. Nat. Commun. **12**, 1274 (2021).

[14] Moffitt, J. R., Lundberg, E. & Heyn, H. The emerging landscape of spatial profiling technologies. Nat. Rev. Genet. **23**, 741–759 (2022).

[15] Fawkner-Corbett, D. et al. Spatiotemporal analysis of human intestinal development at single-cell resolution. Cell **184**, 810–826.e23 (2021).

[16] Peterson, V. M. et al. Multiplexed quantification of proteins and transcripts in single cells. Nat. Biotechnol. **35**, 936–939 (2017).

[17] Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. Nat. Methods **14**, 865–868 (2017).

[18] Zappia, L. & Theis, F. J. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. Genome Biol. **22**, 301 (2021).

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

[20] Lotfollahi, M., Wolf, F. A. & Theis, F. J. scgen predicts single-cell perturbation responses. Nat. Methods **16**, 715–721 (2019).

[21] Lance, C. et al. Multimodal single cell data integration challenge: Results and lessons learned. In NeurIPS 2021 Competitions and Demonstrations Track, 162–176 (PMLR, 2022).

[22] Gayoso, A. et al. A python library for probabilistic analysis of single-cell omics data. Nat. Biotechnol. **40**, 163–166 (2022).

[23] Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. Nat. Methods **18**, 272–282 (2021).

[24] Martens, L. D., Fischer, D. S., Yépez, V. A., Theis, F. J. & Gagneur, J. Modeling fragment counts improves single-cell ATAC-seq analysis. Nat. Methods **21**, 28–31 (2024).

[25] Cao, Z.-J. & Gao, G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. Nat. Biotechnol. **40**, 1458–1466 (2022).

[26] Kunes, R. Z., Walle, T., Land, M., Nawy, T. & Pe'er, D. Supervised discovery of interpretable gene programs from single-cell data. Nat. Biotechnol. (2023).

[27] Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. Bioinformatics **36**, 3418–3421 (2020).

[28] DeTomaso, D. & Yosef, N. Hotspot identifies informative gene modules across modalities of single-cell genomics. Cell Syst. **12**, 446–456.e9 (2021).

[29] McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv [stat.ML] (2018).

[30] Becht, E. et al. Dimensionality reduction for visualizing single-cell data using UMAP. Nat. Biotechnol. **37**, 38–44 (2018).

[31] Traag, V. A., Waltman, L. & van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. Sci. Rep. **9**, 5233 (2019).

[32] Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat. Rev. Genet. **20**, 273–282 (2019).

[33] Wang, X., Chen, H., Tang, S., Wu, Z. & Zhu, W. Disentangled representation learning. IEEE Trans. Pattern Anal. Mach. Intell. **46**, 9677–9696 (2024).

[34] Locatello, F. et al. Challenging common assumptions in the unsupervised learning of disentangled representations. ICML **97**, 4114–4124 (2018).

[35] Ma, J., Zhou, C., Cui, P., Yang, H. & Zhu, W. Learning disentangled representations for recommendation. Advances in Neural Information Processing Systems **32** (2019).

[36] Lachapelle, S. et al. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. arXiv [stat.ML] (2021).

[37] Lachapelle, S., Mahajan, D., Mitliagkas, I. & Lacoste-Julien, S. Additive decoders for latent variables identification and cartesian-product extrapolation. In Oh, A. et al. (eds.) Advances in Neural Information Processing Systems, vol. 36, 25112–25150 (Curran Associates, Inc., 2023).

[38] Argelaguet, R. et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. **21**, 111 (2020).

[39] Hackenberg, M., Brunn, N., Vogel, T. & Binder, H. Infusing structural assumptions into dimension reduction for single-cell RNA sequencing data to identify small gene sets (2024).

[40] Lopez, R. et al. Learning causal representations of single cells via sparse mechanism shift modeling. In Conference on Causal Learning and Reasoning, 662–691 (PMLR, 2023).

[41] Bereket, M. & Karaletsos, T. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. In Oh, A. et al. (eds.) Advances in Neural Information Processing Systems, vol. 36, 1–12 (Curran Associates, Inc., 2023).

[42] Hediyeh-zadeh, S., Fischer, T. & Theis, F. J. Disentanglement via mechanism sparsity by replaying realizations of the past (2024).

[43] Lotfollahi, M. et al. Biologically informed deep learning to query gene programs in single-cell atlases. Nat. Cell Biol. **25**, 337–350 (2023).

[44] Gut, G., Stark, S. G., Rätsch, G. & Davidson, N. R. pmVAE: Learning interpretable Single-Cell representations with pathway modules (2021).

[45] Nazaret, A. et al. Deep generative model deciphers derailed trajectories in acute myeloid leukemia (2023).

[46] Lotfollahi, M., Dony, L., Agarwala, H. & Theis, F. Out-of-distribution prediction with disentangled representations for single-cell RNA sequencing data. bioRxiv 2021.09.01.458535 (2021).

[47] Yu, H. & Welch, J. D. MichiGAN: sampling from disentangled representations of single-cell data using generative adversarial networks. Genome Biol. **22**, 158 (2021).

[48] Khemakhem, I., Kingma, D., Monti, R. & Hyvarinen, A. Variational autoencoders and nonlinear ICA: A unifying framework. In International Conference on Artificial Intelligence and Statistics, 2207–2217 (PMLR, 2020).

[49] Lachapelle, S. et al. Synergies between disentanglement and sparsity: Generalization and identifiability in multi-task learning. ICML **202**, 18171–18206 (2022).

[50] Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. Nat. Commun. **10**, 390 (2019).

[51] De Donno, C. et al. Population-level integration of single-cell datasets enables multi-scale analysis across samples. Nat. Methods **20**, 1683–1692 (2023).

[52] Luecken, M. D. et al. Benchmarking atlas-level data integration in single-cell genomics. Nat. Methods **19**, 41–50 (2022).

[53] Worah, K. et al. Proteomics of human dendritic cell subsets reveals Subset-Specific surface markers and differential inflammasome function. Cell Rep. **16**, 2953–2966 (2016).

[54] Roussel, X., Garnache Ottou, F. & Renosi, F. Plasmacytoid dendritic cells, a novel target in myeloid neoplasms. Cancers **14** (2022).

[55] Zheng, Y. & Rudensky, A. Y. Foxp3 in control of the regulatory T cell lineage. Nat. Immunol. **8**, 457–462 (2007).

[56] Kolar, P. et al. CTLA-4 (CD152) controls homeostasis and suppressive capacity of regulatory T cells in mice. Arthritis Rheum. **60**, 123–132 (2009).

[57] van den Brink, S. C. et al. Single-cell sequencing reveals dissociation-induced gene expression in tissue subpopulations. Nat. Methods **14**, 935–936 (2017).

[58] Denisenko, E. et al. Systematic assessment of tissue dissociation and storage biases in single-cell and single-nucleus RNA-seq workflows. Genome Biol. **21**, 130 (2020).

[59] Adam, M., Potter, A. S. & Potter, S. S. Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: a molecular atlas of kidney development. Development **144**, 3625–3632 (2017).

[60] Gautier, E.-F. et al. Comprehensive proteomic analysis of human erythropoiesis. Cell Rep. **16**, 1470–1484 (2016).

[61] Guo, X., Zhao, Y., Kim, J. & Dean, A. Hemogen/BRG1 cooperativity modulates promoter and enhancer activation during erythropoiesis. Blood **139**, 3532–3545 (2022).

[62] Dean, L. Blood Groups and Red Cell Antigens (National Center for Biotechnology Information (US), 2005).

[63] Stephenson, E. et al. Single-cell multi-omics analysis of the immune response in COVID-19. Nat. Med. **27**, 904–916 (2021).

[64] Sur, A. et al. Single-cell analysis of shared signatures and transcriptional diversity during zebrafish development. Dev. Cell **58**, 3028–3047.e12 (2023).

[65] Chen, T., Li, X., Grosse, R. & Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. Adv. Neural Inf. Process. Syst. **abs/1802.04942** (2018).

[66] Li, X. et al. ScRNA-seq expression of IFI27 and APOC2 identifies four alveolar macrophage superclusters in healthy BALF. Life Sci Alliance **5** (2022).

[67] Aegerter, H., Lambrecht, B. N. & Jakubzick, C. V. Biology of lung macrophages in health and disease. Immunity **55**, 1564–1580 (2022).

[68] Bailey, J. I. et al. Expansion of profibrotic monocyte-derived alveolar macrophages in patients with persistent respiratory symptoms and radiographic abnormalities after COVID-19 (2023).

[69] Tokunaga, R. et al. CXCL9, CXCL10, CXCL11/CXCR3 axis for immune activation - a target for novel cancer therapy. Cancer Treat. Rev. **63**, 40–47 (2018).

[70] Kameda, M. et al. CXCL9, CXCL10, and CXCL11; biomarkers of pulmonary inflammation associated with autoimmunity in patients with collagen vascular diseases-associated interstitial lung disease and interstitial pneumonia with autoimmune features. PLoS One **15**, e0241719 (2020).

[71] Callahan, V. et al. The Pro-Inflammatory chemokines CXCL9, CXCL10 and CXCL11 are up-regulated following SARS-CoV-2 infection in an AKT-Dependent manner. Viruses **13** (2021).

[72] Korbecki, J. et al. Fractalkine/CX3CL1 in neoplastic processes. Int. J. Mol. Sci. **21**, 3723 (2020).

[73] Azimi, I., Petersen, R. M., Thompson, E. W., Roberts-Thomson, S. J. & Monteith, G. R. Hypoxia-induced reactive oxygen species mediate N-cadherin and SERPINE1 expression, EGFR signalling and motility in MDA-MB-468 breast cancer cells. Sci. Rep. **7**, 15140 (2017).

[74] Finger, E. C. et al. Hypoxic induction of AKAP12 variant 2 shifts PKA-mediated protein phosphorylation to enhance migration and metastasis of melanoma cells. Proc. Natl. Acad. Sci. U. S. A. **112**, 4441–4446 (2015).

[75] Qi, H., Yao, C., Xing, J. & Qin, Y. Hypoxia-induced GPR4 suppresses trophoblast cell migration and proliferation through the MAPK signaling pathway. Reprod. Toxicol. **99**, 1–8 (2021).

[76] Biron-Shental, T. et al. Hypoxia enhances the expression of follistatin-like 3 in term human trophoblasts. Placenta **29**, 51–57 (2008).

[77] Chen, Z., Han, F., Du, Y., Shi, H. & Zhou, W. Hypoxic microenvironment in cancer: molecular mechanisms and therapeutic interventions. Signal Transduct. Target. Ther. **8**, 70 (2023).

[78] Guo, X. & Chen, L. From G1 to M: a comparative study of methods for identifying cell cycle phases. Brief. Bioinform. **25**, bbad517 (2024).

[79] Weiler, P., Lange, M., Klein, M., Pe'er, D. & Theis, F. CellRank 2: unified fate mapping in multiview single-cell data. Nat. Methods (2024).

[80] Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. Nat. Biotechnol. **38**, 1408–1414 (2020).

[81] Pan, F. C., Brissova, M., Powers, A. C., Pfaff, S. & Wright, C. V. E. Inactivating the permanent neonatal diabetes gene Mnx1 switches insulin-producing beta-cells to a delta-like fate and reveals a facultative proliferative capacity in aged beta-cells. Development **142**, 3637–3648 (2015).

[82] Kulkarni, R. N. et al. PDX-1 haploinsufficiency limits the compensatory islet hyperplasia that occurs in response to insulin resistance. J. Clin. Invest. **114**, 828–836 (2004).

[83] Gao, T. et al. Pdx1 maintains beta cell identity and function by repressing an alpha cell program. Cell Metab. **19**, 259–271 (2014).

[84] Huang, C. et al. Synaptotagmin 4 regulates pancreatic beta cell maturation by modulating the Ca2+ sensitivity of insulin secretion vesicles. Dev. Cell **45**, 347–361.e5 (2018).

[85] Klein, D. et al. Mapping cells through time and space with moscot. bioRxiv 2023.05.11.540374 (2023).

[86] Zhang, J., McKenna, L. B., Bogue, C. W. & Kaestner, K. H. The diabetes gene hhex maintains delta-cell differentiation and islet function. Genes Dev. **28**, 829–834 (2014).

[87] Hrovatin, K. et al. Integrating single-cell RNA-seq datasets with substantial batch effects. bioRxiv (2024).

[88] Xu, C. et al. Automatic cell-type harmonization and integration across human cell atlas datasets. Cell **186**, 5876–5891.e20 (2023).

[89] Webber, W., Moffat, A. & Zobel, J. A similarity measure for indefinite rankings. ACM Trans. Inf. Syst. **28**, 1–38 (2010).

[90] Luecken, M. D. et al. A sandbox for prediction and integration of DNA, RNA, and proteins in single cells (2021).

[91] Ashuach, T., Reidenbach, D. A., Gayoso, A. & Yosef, N. PeakVI: A deep generative model for single-cell chromatin accessibility analysis. Cell Rep Methods **2**, 100182 (2022).

[92] Bravo González-Blas, C. et al. cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. Nat. Methods **16**, 397–400 (2019).

[93] Bravo González-Blas, C. et al. SCENIC+: single-cell multiomic inference of enhancers and gene regulatory networks. Nat. Methods **20**, 1355–1367 (2023).

[94] Andrews, N. et al. An unsupervised method for physical cell interaction profiling of complex tissues. Nat. Methods **18**, 912–920 (2021).

[95] Asp, M., Bergenstråhle, J. & Lundeberg, J. Spatially resolved transcriptomes-next generation tools for tissue exploration. Bioessays **42**, e1900221 (2020).

[96] Rao, A., Barkley, D., França, G. S. & Yanai, I. Exploring tissue architecture using spatial transcriptomics. Nature **596**, 211–220 (2021).

[97] Heumos, L. et al. Best practices for single-cell analysis across modalities. Nat. Rev. Genet. **24**, 550–572 (2023).

[98] Brady, J. et al. Interaction asymmetry: A general principle for learning composable abstractions. arXiv [cs.LG] (2024).

[99] Virshup, I. et al. The scverse project provides a computational ecosystem for single-cell omics data analysis. Nat. Biotechnol. **41**, 604–606 (2023).

[100] Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. In 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2 (2014).

[101] Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional out-of-distribution generation for unpaired data using transfer VAE. Bioinformatics **36**, i610–i617 (2020).

[102] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. arXiv [cs.LG] (2014).

[103] Svensson, V. Droplet scRNA-seq is not zero-inflated. Nat. Biotechnol. **38**, 147–150 (2020).

[104] Cao, Y., Kitanovski, S., Küppers, R. & Hoffmann, D. UMI or not UMI, that is the question for scRNA-seq zero-inflation. Nat. Biotechnol. **39**, 158–159 (2021).

[105] Grün, D., Kester, L. & van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. Nat. Methods **11**, 637–640 (2014).

[106] Braun, J. & Griebel, M. On a constructive proof of kolmogorov's superposition theorem. Constr. Approx. **30**, 653–675 (2009).

[107] Diaconis, P. & Shahshahani, M. On nonlinear functions of linear combinations. SIAM J. Sci. Stat. Comput. **5**, 175–191 (1984).

[108] Biewald, L. Experiment tracking with weights and biases (2020). Software available from wandb.com.

[109] White, J. V., Steingold, S. & Fournelle, C. Performance metrics for group-detection algorithms. Proceedings of Interface (2004).

[110] Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. **19**, 15 (2018).

[111] Ahlmann-Eltze, C. & Huber, W. Comparison of transformations for single-cell RNA-seq data. Nat. Methods **20**, 665–672 (2023).

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

[113] Hyvärinen, A. & Oja, E. Independent component analysis: algorithms and applications. Neural Netw. **13**, 411–430 (2000).

[114] Zhao, Y., Cai, H., Zhang, Z., Tang, J. & Li, Y. Learning interpretable cellular and gene signature embeddings from single-cell transcriptomic data. Nat. Commun. **12**, 5261 (2021).

[115] Liu, J. et al. Jointly defining cell types from multiple single-cell datasets using LIGER. Nat. Protoc. **15**, 3632–3662 (2020).

[116] Fritsch, F. N. & Butland, J. A method for constructing local monotone piecewise cubic interpolants. SIAM J. Sci. Stat. Comput. **5**, 300–304 (1984).

[117] Luecken, M. et al. Benchmarking atlas-level data integration in single-cell genomics - integration task datasets (2022).

[118] Heumos, L. et al. Pertpy: an end-to-end framework for perturbation analysis. bioRxiv 2024.08.04.606516 (2024).

[119] Lange, M. et al. CellRank for directed single-cell fate mapping. Nat. Methods **19**, 159–170 (2022).

[120] Creators Kristian K Ullrich1 Show affiliations 1. Max Planck Institute for Evolutionary Biology. zebrafish GSE223922 scRNA data set objects.

[121] Locatello, F. et al. Disentangling factors of variation using few labels. arXiv [cs.LG] (2019).

[122] He, Z. et al. Mosaic integration and knowledge transfer of single-cell multimodal data with MIDAS. Nat. Biotechnol. (2024).

[123] Aliee, H. et al. inVAE: Conditionally invariant representation learning for generating multivariate single-cell reference maps. bioRxiv 2024.12.06.627196 (2024).

[124] Weinberger, E., Lin, C. & Lee, S.-I. Isolating salient variations of interest in single-cell data with contrastiveVI. Nat. Methods **20**, 1336–1345 (2023).

[125] Tu, X. et al. A supervised contrastive framework for learning disentangled representations of cell perturbation data (2024).

[126] Lopez, R., Huetter, J.-C., Hajiramezanali, E., Pritchard, J. K. & Regev, A. Toward the identifiability of comparative deep generative models. In Causal Learning and Reasoning, 868–912 (PMLR, 2024).

[127] Zhang, Z., Zhao, X., Qiu, P. & Zhang, X. scDisInFact: disentangled learning for integration and prediction of multi-batch multi-condition single-cell RNA-sequencing data (2023).

[128] Piran, Z., Cohen, N., Hoshen, Y. & Nitzan, M. Disentanglement of single-cell data with biolord. Nat. Biotechnol. (2024).

[129] Paige, B. et al. Learning disentangled representations with semi-supervised deep generative models. Advances in neural information processing systems **30** (2017).

[130] Eastwood, C. et al. Self-supervised disentanglement by leveraging structure in data augmentations. arXiv [cs.LG] (2023).

[131] Xu, D. et al. A sparsity principle for partially observable causal representation learning. arXiv [cs.LG] (2024).

[132] Theis, F. J. Towards a general independent subspace analysis. In Advances in Neural Information Processing Systems 19, 1361–1368 (The MIT Press, 2007).

[133] Choi, Y., Li, R. & Quon, G. siVAE: interpretable deep generative models for single-cell transcriptomes. Genome Biol. **24**, 29 (2023).

[134] Nalio Ramos, R. et al. Tissue-resident FOLR2+ macrophages associate with CD8+ T cell infiltration in human breast cancer. Cell **185**, 1189–1207.e25 (2022).

[135] Sun, Y., Sheshadri, N. & Zong, W.-X. SERPINB3 and B4: From biochemistry to biology. Semin. Cell Dev. Biol. **62**, 170–177 (2017).

[136] Lee, P. H. et al. Alteration in claudin-4 contributes to airway inflammation and responsiveness in asthma. Allergy Asthma Immunol. Res. **10**, 25–33 (2018).

[137] Martínez, I., Lombardía, L., García-Barreno, B., Domínguez, O. & Melero, J. A. Distinct gene subsets are induced at different time points after human respiratory syncytial virus infection of A549 cells. J. Gen. Virol. **88**, 570–581 (2007).

[138] Ohta, Y. et al. Claudin-4 as a marker for distinguishing malignant mesothelioma from lung carcinoma and serous adenocarcinoma. Int. J. Surg. Pathol. **21**, 493–501 (2013).

[139] Ruiz García, S. et al. Novel dynamics of human mucociliary differentiation revealed by single-cell RNA sequencing of nasal epithelial cultures. Development **146**, dev.177428 (2019).

[140] Yoshida, M. et al. Local and systemic responses to SARS-CoV-2 infection in children and adults. Nature **602**, 321–327 (2022).

[141] Mimata, Y. et al. Interleukin-6 upregulates expression of ADAMTS-4 in fibroblast-like synoviocytes from patients with rheumatoid arthritis: IL-6 upregulates ADAMTS-4 in RA-FLS. Int. J. Rheum. Dis. **15**, 36–44 (2012).

[142] Zhang, H. et al. CRISPLD2 (LGL1) inhibits proinflammatory mediators in human fetal, adult, and COPD lung fibroblasts and epithelial cells. Physiol Rep **4** (2016).

[143] Himes, B. E. et al. RNA-Seq transcriptome profiling identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth muscle cells. PLoS One **9**, e99625 (2014).

[144] Xiong, D., Wang, Y. & You, M. A gene expression signature of TREM2hi macrophages and $\gamma\delta$ T cells predicts immunotherapy response. Nat. Commun. **11**, 5084 (2020).

[145] Wu, K. et al. TREM-2 promotes macrophage survival and lung disease after respiratory viral infection. J. Exp. Med. **212**, 681–697 (2015).

[146] Xiong, D., Wang, Y. & You, M. A gene expression signature of TREM2hi macrophages and $\gamma\delta$ T cells predicts immunotherapy response. Nat. Commun. **11**, 5084 (2020).

[147] Houghton, A. M. Matrix metalloproteinases in destructive lung disease. Matrix Biol. **44-46**, 167–174 (2015).

[148] Bui, T. M., Wiesolek, H. L. & Sumagin, R. ICAM-1: A master regulator of cellular responses in inflammation, injury resolution, and tumorigenesis. J. Leukoc. Biol. **108**, 787–799 (2020).

[149] Liu, W. & Rodgers, G. P. Olfactomedin 4 expression and functions in innate immunity, inflammation, and cancer. Cancer Metastasis Rev. **35**, 201–212 (2016).

[150] Jarczak, J. et al. Defensins: natural component of human innate immunity. Hum. Immunol. **74**, 1069–1079 (2013).

[151] Xu, H. et al. New genetic and epigenetic insights into the chemokine system: the latest discoveries aiding progression toward precision medicine. Cell. Mol. Immunol. **20**, 739–776 (2023).

[152] Sokol, C. L. & Luster, A. D. The chemokine system in innate immunity. Cold Spring Harb. Perspect. Biol. **7** (2015).

[153] Zhou, Z. et al. Single-cell profiling identifies IL1Bhi macrophages associated with inflammation in PD-1 inhibitor-induced inflammatory arthritis. Nat. Commun. **15**, 2107 (2024).

[154] Yang, C. et al. Heterogeneity of human bone marrow and blood natural killer cells defined by single-cell transcriptome. Nat. Commun. **10**, 3931 (2019).

[155] Marozkina, N. et al. Somatic cell hemoglobin modulates nitrogen oxide metabolism in the human airway epithelium. Sci. Rep. **11**, 15498 (2021).

[156] Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289 (2015).

[157] Kolberg, L. et al. g:profiler-interoperable web service for functional enrichment analysis and gene identifier mapping (2023 update). Nucleic Acids Res. **51**, W207–W212 (2023).

Supplemental Material

Contents

1 Related works and paper contributions **III**

2 On the additivity of underlying biological processes in count space **VI**

3 Differences between disentanglement and independence **VII**

4 Identification of the remaining non-cell-type factors for the HLCA **VIII**

5 Considerations for encouraging disentanglement **X**

6 On multiplicative behavior of current autoencoders and factorization models **XII**

7 Log-space Parametrization of the Negative Binomial log-likelihood **XIII**

8 Interpretation of Latent Factors via Additive Decoder Architecture **XIV**

9 Ordering of Latent Factors **XVII**

10 DRVI's code functionality **XVII**

Supplemental Tables **XIX**

Related Methods. XIX
Summary of Datasets. XIX
Summary of Model Parameters. XX
Comparison of Different Metric Families. XX

Supplemental Figures **XXII**

UMAPs and integration results for benchmarked methods on the immune dataset. XXII
Interpretability of DRVI latent factors in immune dataset. XXII
Latent space visualization for benchmarked methods in the immune dataset. XXIII
The effect of removing technical stress response factors. XXIV
Annotation refinement and correction using DRVI in the immune dataset. XXVI
Summary of all metrics. XXVI
Results of the primary disentanglement metric family. XXVII
Results of the secondary disentanglement metrics. XXVIII
DRVI identifies granular immune cell subtypes that are penalized by MSGS metrics. . . . XXIX
UMAPs and integration results for benchmarked methods on the HLCA dataset. XXIX
Latent space visualization for benchmarked methods on the HLCA. XXX
Rare cell-type identification performance of Leiden clustering in the HLCA. XXXI
Activity of DRVI factors representing non-cell-type processes in the HLCA on UMAP . . XXXIII
Interpretability of DRVI latent factors corresponding to non-cell-type processes in the HLCA XXXIII
Indication of tumor-adjacent blood vessels in the HLCA. XXXIV
DRVI disentangles cell cycle stages. XXXV
Developmental fate prediction based on model embeddings. XXXVI
DRVI's latent space effectively models interacting double perturbations. XXXVII
DRVI identifies and preserves the geometry of combinatorial perturbations. XXXVIII
The saturation of the number of employed latent dimensions by DRVI. XXXIX
Further analysis on the saturation of employed latent factor. XL
Training DRVI on four datasets of the Blood data. XLI
Assessing the generalization and disentanglement of latent programs in unseen Blood data XLII

Full disentanglement metrics on generalization to unseen datasets in the Blood data. . . . XLIV
DRVI is applicable to scATAC-seq data. . . . XLIV
Disentanglement versus Statistical Independence. . . . XLV
Plausibility of DRVI's additive assumption in scRNA-seq data. . . . XLVI
Heatmap of simulated datasets with known ground truth. . . . XLVII
Evaluation results for simulated datasets with known ground truth. . . . XLVIII
Summary of all metrics for the ablation study. . . . XLVIII
Details of disentanglement metrics for the ablation study. . . . XLIX
The optimization of training objectives. . . . L
The effect of KL loss weight on reconstruction and disentanglement performance. . . . LI
Runtime analysis of DRVI. . . . LII
Runtime for interpretability and sorting latent factors. . . . LIV
The effective and ignorable parts of the pooling function . . . LV
Numerical validation of sufficient independence in Immune data. . . . LVI
How latent space sampling and KL loss promote disentanglement. . . . LVI
Unaggregated Scaled Mutual Information (SMI) scores for the Immune dataset. . . . LVII
Unaggregated Same-Process Neighbors (SPN) scores for the Immune dataset. . . . LVIII
Unaggregated Scaled Mutual Information (SMI) scores for the HLCA. . . . LIX
Unaggregated Same-Process Neighbors (SPN) scores for the HLCA. . . . LX

Supplemental Note 1: Related works and paper contributions

To position DRVI within the field, we begin by discussing its relation to the broader disentangled representation learning literature. Then, we review existing disentanglement methods applied to single-cell omics and finally highlight the specific contributions of this paper.

Disentangled representation learning

Disentangled representation learning methods can be categorized according to architecture, representation structure, and supervision signal [33].

Common disentangled representation learning methods utilize generative models such as variational autoencoders (VAEs), generative adversarial networks (GANs), and recently, diffusion models [33]. These models are primarily applied in computer vision with various regularization losses to encourage disentanglement. Nevertheless, according to the well-established limitation that unsupervised disentanglement is theoretically impossible without inductive bias on methods and datasets [34], the possibility of disentanglement in these architectures can be attributed to their inherent inductive biases, such as convolutional neural networks (CNNs). However, given that generative models in single-cell representation learning are typically unconstrained and lack such inherent inductive biases, the direct applicability of these paradigms to single-cell omics is questionable. Consequently, DRVI addresses this limitation by incorporating specific architectural designs, such as additive decoders [37] with log-sum-exp pooling, which serve as inductive biases to enforce disentanglement. This means while DRVI is built within the CVAE framework, its inductive bias can be adapted and incorporated within the diverse generative model settings outlined in [33].

Disentangled latent space models can be categorized based on their approach to building conceptual disentangled blocks. Some models partition the latent space into independent blocks (block-based disentanglement), each representing a high-level process. However, dimensions within these blocks may remain entangled. Other disentangled latent space models take a more granular approach, disentangling every dimension in the latent space (dimension-wise disentanglement) to represent underlying processes. DRVI falls into this second category, while also offering users the flexibility to perform block-based disentanglement by adjusting default parameters.

The impossibility of general disentanglement in unsupervised settings motivated the incorporation of supervised and weakly supervised information [33, 121]. In accordance with this, many existing single-cell disentanglement methods rely on auxiliary data or supervised information to enforce disentanglement. In contrast, employing a suitable inductive bias allows DRVI to learn disentangled representations without requiring labeled information about the underlying biological processes.

Disentanglement in single-cell omics

Here, we describe relevant works on disentangled representation learning of single-cell omics, focused on the assumptions and the structure of their latent space representations.

In block-based disentanglement, methods such as MIDAS [122] and inVAE [123] disentangle biological and technical effects into two distinct blocks within the latent space. Similarly, ContrastiveVI [124] and SC-VAE [125, 126] construct two latent space vectors based on the given perturbational signatures to isolate salient variations. Finally, scDisInFact [127] and BIOLORD [128] construct multiple conceptual blocks given variations of interest.

The impossibility of achieving complete dimension-wise disentanglement in the general case of latent space models is a well-established theorem [34]. Researchers have come up with various strategies to address this challenge, including incorporating auxiliary data (supervision) or constraining the model search space [34, 37, 129–132]. In single-cell omics literature, sVAE [40], SAMS-VAE [41], and

sVAE-ligr [42] use perturbations as supervision and sparsely represent groups of similar perturbations in each dimension. Models like expiMap [43] and pmVAE [44] incorporate pathway knowledge into their architectures to achieve disentanglement. Decipher [45] aligns each of the two primary latent dimensions considering given variations of interest. BAE [39] and MOFA+ [38] try to disentangle each latent dimension by enforcing sparsity on their linear models. Finally, Spectra [26] disentangles the factors by using cell-type information and gene-gene knowledge graph as supervision, enforcing sparsity on a non-negative matrix factorization model. In addition to the mentioned models, there are a few other approaches seeking to achieve nonlinear unconstrained disentanglement, such as β-TCVAE and MICHIGAN [46, 47]. Our benchmarking confirms the theoretical results that using fully connected layers and lacking an inductive bias, none of these two models significantly improve disentanglement [34]. We summarized disentangled latent space models tailored specifically for single-cell omics data in Supplementary Table 1.

Interpretability is one of the main advantages of dimension-wise disentanglement methods. Topic modeling and factorization models are two relevant lines of research in single-cell representation learning, providing interpretable latent representations [114, 115, 133]. Topic modeling methods often assume that cells are sampled from disjoint topics with probabilities that sum to one (usually modeled by the Dirichlet distribution) [114]. Consequently, while topic modeling approaches can identify dominant competing signals (interpretable as soft clustering), they may be unable to identifying secondary or shared signals. In contrast, disentangled representation learning allows multiple parallel processes to coexist within a cell. Factorization models represent the observational matrix as the product of two low-dimensional matrices [38, 114, 115, 133]. While the linearity constraint alone isn't enough to mitigate the well-known unidentifiability issue [121], additional assumptions such as sparsity [38] or non-negativity [115] of loadings can potentially enforce disentanglement and facilitate interpretation.

Paper contributions

- **Highlighting the Importance of Disentanglement in Single-Cell Omics**: We demonstrated the significance of unsupervised dimension-wise disentanglement as an important yet understudied problem in single-cell omics. We explained its advantage compared to clustering, its ability to identify rare cell types and variations beyond cell types, and its capacity to represent continuous developmental processes.
- **Integration of Additive Decoders and scVI for Disentangled Representation learning**: For the first time, we successfully integrated scVI [19] with additive decoders [37] to achieve disentangled representations. We demonstrated that a naive combination of these approaches failed to produce disentangled latent spaces for single-cell omics data, showing the need for the introduction of novel architectural modifications, which we will outline next.
- **Novel Shared Activation**: We introduced a neural network architecture where each decoder subnetwork was followed by a shared activation function ψ, and we formulated a more practical implementation as a pooling function. This modification (that still remains within the additive decoders framework) is novel, effective, and not discussed in [37] (Supplementary Fig. 36). In addition, we highlighted the mechanistic interpretation of this decision in single-cell omics: "Independent biological processes add up in count space," and provided a rationale based on biological and technical perspectives.
- **Efficient and Effective Implementation**: Beyond the core architecture, we introduced several practical modifications necessary for achieving optimal results. These include the parametrization of the Negative Binomial distribution in log-space (Supplementary Note 7), the use of ELU as a smooth alternative to ReLU, and, most importantly, the implementation of efficient deep learning modules to enable parallel processing of MLPs and layer normalization.

- **Efficient Interpretability Pipeline**: We developed an efficient interpretability pipeline that allows for the interpretation of latent factors in terms of gene scores. The runtime of this algorithm is independent of the number of cells.
- **Comprehensive Evaluation Framework**: We proposed a comprehensive framework for evaluating disentanglement in single-cell omics. This framework comprises both existing and novel metrics, and we provided a justification for our choices within different metric families.

Supplemental Note 2: On the additivity of underlying biological processes in count space

In DRVI, we proposed the additivity in count space as an alternative to the current trend in white-box models, where data is modeled in log space, and effects are added up. Log-space additivity leads to the assumption that when two underlying processes affect the same gene, their effects multiply in count space (Supplementary Note 6).

We offer a mechanistic justification for this assumption. Considering the decay of expressed RNAs within each process (as a modeling choice, RNA degradation is not modeled explicitly due to the requirement for more complex models and data limitations), if two biological processes affect the same gene in different time frames, the newly transcribed genes will result in an additive effect on the total RNA count and therefore add up in count space when sequenced (Supplementary Fig. 27 top). This aligns with biological scenarios such as developmental bifurcations, induced perturbations, and stress responses, where newly transcribed RNAs contribute additively. Furthermore, this assumption also aligns with many technical processes and artifacts during the generation of scRNA-seq data. For instance, the combined expression of doublet cells, background contamination, hemoglobin (HBB) gene contamination, and even residual counts from neighboring cells due to imperfect segmentation in spatial omics all contribute additively to the observed count space (Supplementary Fig. 27 bottom). Thus, in contrast to prior models, our additive assumption provides a more plausible framework for modeling single-cell count data.

We would like to clarify that here the task of interest in disentanglement is to identify the underlying biological processes, without inferring causal relationships between them. For instance, consider two processes, P1 and P2, where P1 activates P2, and P2 suppresses P1. Each process has its own set of genes that are transcribed when it is active: a subset of genes expressed by P1 may activate P2, while the genes expressed by P2 may inhibit P1. Although the activation of P2 may influence the expression of genes associated with P1, we do not consider these indirectly affected genes as part of the direct transcriptional output of P2. Accordingly, while we are interested in identifying P1 and P2, we are not interested in identifying their causal relationships, although disentanglement may help in this downstream task. In our framework, we define the output of a process solely based on its direct transcriptional effects, disregarding downstream or propagated causal effects. This assumption allows us to model the activity of individual processes additively. And, inferring relationships between identified processes, such as potential causal links between P1 and P2, is beyond the scope of this study.

Finally, when the gene expression from two distinct processes (P1 and P2) creates a novel transcriptional activity, such as the synergistic regulation of transcription factors, their combined effect can be effectively modeled as an entirely new process within our framework.

Supplemental Note 3: Differences between disentanglement and independence

In the disentangled representation learning literature, two perspectives exist: a classical one assumes statistical independence across the underlying factors of variation, while the other permits causal relationships among them [33]. In this study, we defined disentanglement based on a one-to-one alignment between the ground-truth (GT) underlying processes and the learned latent factors, which allows for correlations and dependencies between GT processes, inherent in biological systems.

Here, we discuss the differences between disentanglement and statistical independence by providing examples, including cases where disentanglement is satisfied but latent space factors are not statistically independent and vice versa. In all these examples, we assume the training data consists of images with one red ball and one blue ball. We can then model this data using a generative process with two factors: one representing the height of the red ball ($A \in [0, 1]$) and the other representing the height of the blue ball ($B \in [0, 1]$).

Example 1: Alignment of Independence and Disentanglement.

In some settings, the underlying factors of variation are statistically independent and the data manifold is fully observed with no bias. For instance, assume that red and blue balls move independently and training manifold contains unformly distributed samples ($A, B \in Uniform(0, 1)$). In this case, a model that successfully disentangles the factors (recovering A and B) will naturally yield statistically independent latent dimensions (Supplementary Fig. 26a). Here, metrics penalizing dependence (like MIG) and metrics measuring disentanglement (like LMS) will both be maximized.

Example 2: Disentanglement without Independence. Consider a scenario where the underlying mechanisms A and B are mechanically independent, but the training data is incomplete or subject to a selection bias. A model that correctly learns the disentangled concepts A and B will show correlations inside latent factors, even though the true mechanisms are independent (Supplementary Fig. 26b). This is due to the fact that unlike disentanglement, statistical independence is dependent on the observational distribution. Such selection biases may occur due to data biases or arise from causal settings (as illustrated in the next example). A clear example of bias in data is when two independent perturbations are present, but their combination is not. A causal example in single-cell omics is when the coexistence of two perturbations results in cell apoptosis.

Example 3: Causal Dependency. Assume that our datasets consist of images where the red ball is always positioned lower than the blue ball. Then, the two underlying generative processes exhibit a direct causal relationship $A \rightarrow B$ where $A \in Uniform(0, 1)$ and $P(B|A) = Uniform(0, A)$. A perfect model will recover A and B in distinct latent dimensions, but does not exhibit independence (Supplementary Fig. 26c). Biological processes usually exhibit causal dependencies. For example, a program may induce another program, or two programs may compete and form two individual lineages. Even cell types are not independent as we do not expect to see cells with markers of the both populations. In such scenarios, metrics penalizing dependence (like MIG) discourage true identification, while metrics measuring disentanglement (like LMS) will be maximized on true identification.

Example 4: Independence without Disentanglement. Finally, we consider a case where an entangled model achieves statistical independence. In the red and blue balls example, assume that we include an underlying binary process C. We put a square when $C = 0$ and a triangle otherwise ($C = 1$). A disentangled model should identify A, B, and C in individual latent dimensions. But, we consider an over-compressed model that identifies A in one dimensions and $B + C$ in the other. A model that retrieves such a mixture is not solving the disentanglement task, despite satisfying statistical independence (Supplementary Fig. 26d). This example shows that over-compressed models can encourage statistical independence at the cost of disentanglement. While metrics penalizing dependence (like MIG) are maximized in this setting, metrics measuring disentanglement (like LMS) will penalize existence of overlapping processes within a single dimension.

Supplemental Note 4: Identification of the remaining non-cell-type factors for the HLCA

Here, we try to identify and explain the rest of the non-cell-type factors not described in the results section. This is done based on the activity of the factors (Supplementary Fig. 13) and corresponding identified gene programs (Supplementary Fig. 14). Factors are described with the same order as Fig. 4b.

DR 27+: This factor indicates an inflammatory process in myeloid cells. The primary gene associated with this factor is EREG, with genes such as IL1B, IL1A, CXCL8, and CXCL3 upregulated in this subpopulation. GSEA analysis suggests an association with the regulation of apoptotic signaling and interleukin-10 signaling. As mentioned earlier, there are other factors representing inflammation, each with its own specific gene signature.

DR 30+: This factor highlights a gradient in ciliated cells. DNAH11, DNAAF1, DNAH12, and DNAH6 are among the top identifiers of this factor, expressed across ciliated cells, with significantly higher expression observed in the upper gradient of DR 30+. GSEA also suggests cilium for this factor.

DR 16+: All the genes identified for DR16+ are human leukocyte antigen gene complex (HLA) genes, including HLA-DRA, HLA-DRB1, and HLA-DPA. GSEA suggests that this factor corresponds to the MHC class II protein complex. Notably, while HLA genes are generally more highly expressed in certain cell types like myeloid cells, this factor provides a per-cell-type normalized representation, covering similar ranges across relevant cell types, such as myeloid cells and blood vessels.

DR 21-: This factor highlights a subset of epithelial cells mostly active in nasal goblet cells, identified by KLK7, TMPRSS11B, SPINK7, ECM1, SCEL, and KLK6 as their top relevant genes. We could not identify a biologically meaningful explanation for this factor.

DR 24+: FOLR2, F13A1, STAB1, CCL13, and SELENOP are the main identifiers of the gene program associated with DR 24+. These genes indicate a perivascular population of macrophages [68, 134]. High activity of this factor in Interstitial Mph perivascular cells confirms this relevance.

DR 25-: SERPINB3 and SERPINB4 are the top identified gene for DR 25-, active in epithelial cells. Although the coexpression of these two genes is biologically relevant [135], we could not find a specific biological explanation for the subpopulation of cells active in this factor.

DR 28+: The activity of DR 28+ indicates the expression of PLAU, its receptor PLAUR, and an elevated expression of CLDN4 (Claudin-4) in airway epithelium. While PLAU and CLDN4 are separately suggested as markers of lung injury or infection, we could not find a comprehensive description for this factor [136–138].

DR 29-: This factor highlights a subset of epithelial cells, including Hillock-like cells. Our interpretability pipeline identified KRT6B, KRT13, KRT16, and KRT6A as key markers, suggesting a keratization process within this factor. Previous studies ([139, 140]) have employed a diverse set of KRT genes to characterize epithelial subpopulations. In particular, KRT16 and KRT6A are among the markers of certain Hillock-like cell populations.

DR 33+: EDN1, ERRFI1, DST, and TNFRSF12A are the main indicators of DR 33+. This factor is more active in basal cells, with a significant upregulation of cells from the Nawijn_2021 dataset. However, a clear technical or biological explanation for this factor remains open.

DR 35-: This factor indicates a subset of fibroblast lineage and smooth muscle cells identified by a gene program composed of genes such as CRISPLD2, IL6, and ADAMTS4. IL-6 is a pro-inflammatory cytokine, reported to induce expression of ADAMTS4 in rheumatoid arthritis samples [141]. Conversely, CRISPLD2 is a known inhibitory regulator of IL-6 [142, 143]. The coexpression of these genes suggests a dynamic of IL-6 regulation. The exact role of this process needs further

experiments.

DR 35+: C1QA, C1QB, C1QC, and TREM2 are the top indicators of the DR 35+, Suggesting that this factor represents TREM2 and the C1Q complex in myeloid cells, including TREM2+ macrophages [144–146].

DR 36-: LYPD3, CALML3, AQP3, TASCTD2, and LY6D are the genes identifiers of the nonlinear gene program for DR 36-. The role of this factor is not identified.

DR 37-: ANKRD36C gene is the main driver of the DR 37- expressed only in a subset of goblet cells. Comparison with cell types suggests that this factor is closely related to the Bronchial subsets of the Goblet cells. However, we could not find any evidence of the validity of such a subcluster in the literature, and this factor seems to be dataset-specific.

DR 38+: DR 38+, characterized by MKI67, TYMS, and PCLAF expression, is associated with cell-cycle processes as identified by GSEA. This aligns with the cell-type annotations (Fig. 4b).

DR 40+: This factor highlights a subset of airway epithelial cells, with RARRES1, CXCL6, SERPINB4, and IL17C as the key drivers of its gene expression program. While GSEA suggested an IL-17 signaling pathway, we found the relationship between these genes and the pathway to be weak. As a result, we were unable to identify a meaningful explanation for this factor.

DR 41+: This factor identifies a subset of epithelial cells, with MMP1, STC1, MMP13, and MMP10 as their top relevant genes. Accordingly, this factor highlights the activity of certain proteins in the Matrix Metalloproteinase family, suggesting an ongoing destructive process [147].

DR 42+: With genes such as CCL20, CXCL2, CXCL3, CSF2, CSF3, and ICAM1 among the top associated genes, GSEA identifies IL-17, IL-10, and TNF signaling pathways for DR 42+ (Supplementary Fig. 14). This suggests an ongoing inflammation process [148].

DR 43+: This factor highlights two subsets of cells (Supplementary Fig. 13). The tiny subset in Myeloid cells is associated with genes such as OLFM4, S100A7, and DEFA3, suggesting a small population of neutrophils [149, 150]. The other subset in airway epithelium is targeted by S100A7, SPRR2D, SPRR1B, SPRR2E, SPRR2F, and SPRR3 in the top relevant genes. GSEA suggests keratinization and formation of the cornified envelope for this factor. The coexistence of two unrelated cell subsets within this factor suggests a potential failure in disentanglement, possibly due to insufficient number of latent dimensions.

DR 44+: This factor indicates a subset of myeloid cells characterized by but not limited to CCL3, CCL4, CCL3L1, CCL4L2, CXCL3, IL1A, and IL1B genes. CCL3 (MIP-1α) and CCL4 (MIP-1β) and their paralogs CCL3L1 and CCL4L2, are proinflammatory chemokines that attract immune cells [151]. IL-1β is an inflammatory cytokine that activates cells to produce more CCL3 and CCL4 chemokines [152]. GSEA describes this factor by pro-inflammatory process and cytokine activity. This subpopulation of myeloid cells is of biological and clinical interest [153]. It is worth mentioning that DR 27+ complements this by highlighting another population of myeloid cells undergoing interleukin-1 signaling pathway.

DR 49-: With HBA2, HBA1, and HBB as its top identifiers, GSEA analysis suggests that DR 49- represents the hemoglobin metabolic process. While this process is expected to be most active in erythroid cells that are not present in the HLCA, we observed non-negligible expression of hemoglobin genes in other cell types. The expression of hemoglobin genes in the HLCA could be attributed to technical or biological factors [154, 155].

Dim DR 50+: SAA1 and SAA2 are the main identifiers of the dim DR 50+. So, we can conclude that this factor presents the activity of SAA as an inflammatory marker.

DR 54-: The nonlinear gene program associated with DR 54- includes multiple genes such as ISG15, IFIT1, IFI44L, and IFI6. GSEA describes this factor by interferon alpha/beta signaling and response to virus pathways.

Supplemental Note 5: Considerations for encouraging disentanglement

Here, we discuss our considerations for encouraging disentanglement in DRVI. Theorem A.22 of the interaction asymmetry [98] provides a few criteria based on which disentanglement is guaranteed:

(I) Number of processes: In disentanglement theorems, it is implicitly assumed that the number of latent dimensions is the same as the number of GT processes. However, this is rarely known. Accordingly, we show that the model will saturate and extra dimensions will vanish when more latent dimensions are provided (see Results). This will allow users to provide more than enough latent dimensions for training. Accordingly, we assume that all vanished dimensions are removed before checking any other disentanglement criteria.

(II) 1st Order Interaction Asymmetry: The first-order interaction asymmetry requires simple interactions across splits ($D^2_{i,i'}f(z) = 0$ where i, i' are dimensions of two different splits) and higher-order interactions within splits ($D^2_{i,i'}f(z) > 0$ where i, i' are two dimensions within an split). The pooled additive decoders obviously have at most 1st order interactions between splits (Definition 3.3 of [98]). Accordingly, we require at least second-order interactions within each split of our model. Using Average pooling does not necessarily enforce such nonlinearity within each split. However, since $\psi = \exp$ has infinitely many derivatives, using this shared activation function in the LSE pooling heuristically introduces at least second-order interactions within each split.

(III) 1st Order Sufficient Independence: This is the second main criterion for disentanglement. This criterion depends on the choice of the ψ function. For the Average pooling case, we cannot make any extra claim. For DRVI's generative process with LSE pooling, the criteria (Definition A.9 of [98]) will be:

$$\operatorname{rank}\left(\left[FOI(B)\middle|SOI(B)\right]_{B\in\mathcal{P}}\right) = \sum_{B\in\mathcal{P}} \operatorname{rank}\left(FOI(B)\right) + \operatorname{rank}\left(SOI(B)\right) \tag{14}$$

$$FOI(B) = \left[D_i f^{(B)}(z_B) \times \exp(f^{(B)}(z_B))\right]_{i\in B} \tag{15}$$

$$SOI(B) = \left[\left(D^2_{i,i'} f^{(B)}(z_B) + D_i f^{(B)}(z_B) \times D_{i'} f^{(B)}(z_B)\right) \times \exp\left(f^{(B)}(z_B)\right)\right]_{i,i'\in B} \tag{16}$$

Where $FOI(B)$ and $SOI(B)$ are the matrices of first order and second order interactions, $D_i f^{(B)}(z_B)$ and $D^2_{i,i'} f^{(B)}(z_B)$ denote first-order and second-order derivatives of $f^{(B)}$ at point z_B.

Here, we assume each underlying ground-truth biological process (each block B) has its own set of marker genes, without loss of generality we can limit features to the set of markers. The term $\exp(f^{(B)}(Z_B))$ appears in all elements of both the first-order interaction ($FOI(B)$) and second-order interaction ($SOI(B)$) matrices. Since the aggregation of these values ($\sum_{B\in\mathcal{P}} \exp(f^{(B)}(Z_B))$) will reconstruct the expression matrix and the outputs of each split of the generative process ($\exp(f^{(B)}(Z_B))$) are non-negative, they are nonzero if and only if the gene is a marker of the split. This means that for each split B, only the rows corresponding to the marker genes of that split are nonzero. Therefore, $[FOI(B)|SOI(B)]$ matrices have no common nonzero rows between splits. Accordingly:

$$\operatorname{rank}\left(\left[FOI(B)\middle|SOI(B)\right]_{B\in\mathcal{P}}\right) = \sum_{B\in\mathcal{P}} \operatorname{rank}\left([FOI(B)|SOI(B)]\right) \tag{17}$$

This significantly narrows down the gap between the two sides of the criteria.

In addition to this theoretical justification, we considered DRVI trained on Immune data using 32 and 64 latent dimensions and constructed the $[FOI|SOI]$ matrix based on the trained decoder. We empirically observed that the non-vanished dimensions of DRVI form full-rank sufficient independence matrices (Supplementary Fig. 37).

(IV) Diffeomorphism: The decoder is assumed to be a C^2-diffeomorphism (f and its inverse being C^2). We employ exponential linear units (ELU) [156] activation function in the neural networks

defining f, offering a C^1 smoothness alternative to ReLU. However, f still does not satisfy the C^2-diffeomorphism requirement. In practice, we are not aware of any practical model satisfying this criterion.

(V) Reconstruction Loss: Since we estimate the parameters of the noise distribution, solving the reconstruction problem cannot be guaranteed. Nevertheless, achieving an acceptable reconstruction loss is sufficient in practice. This assumption is usually relaxed.

(VI) Domain Assumptions: We can assume that our domain is regularly closed. While path connectivity is typically required to derive disentanglement from local disentanglement, this assumption often falls short in non-developmental data. However, sampling in the latent space promotes greater connectivity and thus encourages global disentanglement. Additionally, the KL term in the loss function helps prevent the coexistence of multiple irrelevant processes within a single dimension, assuming sufficient latent dimensions are available (Supplementary Fig. 38).

Supplemental Note 6: On multiplicative behavior of current autoencoders and factorization models

Let $\boldsymbol{Z} \in \mathbb{N}^{N \times K}$, $\boldsymbol{W} \in \mathbb{N}^{K \times G}$ be the left-hand-side and right-hand-side matrices in a factorization model that models the data in log-space. Generally, log-transformed counts are modeled as the multiplication of these two matrices. So, formally, we have

$$log(\boldsymbol{X} + 1) = \boldsymbol{Z}\boldsymbol{W}. \tag{18}$$

Generally, $\boldsymbol{Z}_{n,k}$ and $\boldsymbol{W}_{k,g}$ are interpreted as the activity of factor k in cell n and the activity of gene g in factor k, respectively. Likewise, $\boldsymbol{Z}_{n,k}\boldsymbol{W}_{k,g}$ shows if gene g should be active in cell n because of the activation of the process k. Let's map the factors in log-space to the original count space:

$$\begin{aligned} log(\boldsymbol{X}_{n,g} + 1) &= \sum_k \boldsymbol{Z}_{n,k}\boldsymbol{W}_{k,g} \\ \Rightarrow \boldsymbol{X}_{n,g} + 1 &= \exp\left(\sum_k \boldsymbol{Z}_{n,k}\boldsymbol{W}_{k,g}\right) \\ \Rightarrow \boldsymbol{X}_{n,g} + 1 &= \prod_k \exp(\boldsymbol{Z}_{n,k}\boldsymbol{W}_{k,g}) \end{aligned} \tag{19}$$

As we see, the values $\boldsymbol{Z}_{n,k}\boldsymbol{W}_{k,g}$ multiply in count space. This leads to the fact that modeling relying on additivity in log space implies that if gene g in cell n is active because of the activation of multiple independent processes, their effect is multiplied in count space.

We can conclude the same in autoencoder models. Let $\boldsymbol{Z}_{n,k}$ be the latent factors in an autoencoder model with a linear decoder. As in the factorization case, let $\boldsymbol{W}$ be the weight matrix in the last layer of the decoder. Interpreting these $\boldsymbol{Z}_{n,k}$ as activity of some factors for cell n and $\boldsymbol{W}_{k,g}$ as the signature of each factor, we can conclude the same.

In the case of deeper decoders, one can define $\boldsymbol{Z}_{n,k}$ as the value for cell n in node k in the last hidden layer of the decoder (instead of latent variables). Considering these values as "hypothetical" factors indicating activity of hypothetical factor k in cell n, we can conclude that autoencoders with MLP decoders also rely on the assumption that some activities (not latent activities but activities in the last layer) aggregate through multiplication in count space.

Supplemental Note 7: Log-space Parametrization of the Negative Binomial log-likelihood

In current generative models that use Negative Binomial distribution, the decoder calculates the parameters in log-space, and then these log-space parameters are transferred to count space by applying the exponential function. The count-space likelihood of the Negative Binomial distribution is then calculated based on the following expression:

$$\begin{aligned}
&\log\mathcal{L}(\mu, \theta \ ; \ x) = \\
&\quad \theta\left(\log(\theta+\epsilon) - \log(\theta+\mu+\epsilon)\right) + \\
&\quad x\left(\log(\mu+\epsilon) - \log(\theta+\mu+\epsilon)\right) + \\
&\quad \log\Gamma(x+\theta) - \log\Gamma(\theta) - \log\Gamma(x+1)
\end{aligned} \tag{20}$$

Where μ and θ represent mean and dispersion in Negative Binomial distribution, x is the observed expression in count space, Γ is the gamma function used as a smoothed version of the factorial function ($\Gamma(n) = (n-1)!$), and ϵ is a small value to increase numerical stability.

As we see above, this approach involves canceling exponential and logarithm functions that may reduce the numerical precision (i.e., $\log(\theta)$ since θ itself is an output of the exponential function applied to the decoder output). This is our motivation to define the likelihood of the Negative Binomial distribution directly based on the log-space parameters as much as possible (especially the mean parameter). Here we need an intermediate trick to simplify $\log\left(\frac{a}{a+b}\right)$:

$$\begin{aligned}
\log\left(\frac{a}{a+b}\right) &= -\log\left(1+\frac{b}{a}\right) \\
&= -\log\left(1+\exp\left(\log b - \log a\right)\right) \\
&= -\text{SoftPlus}\left(\log b - \log a\right)
\end{aligned} \tag{21}$$

Now we have:

$$\begin{aligned}
&\log\mathcal{L}(\log\mu, \log\theta \ ; \ x) \\
&= \log\left(\binom{x+\theta-1}{x}\left(1-\frac{r}{\mu+r}\right)^{x}\left(\frac{r}{\mu+r}\right)^{\theta}\right) \\
&= \log\binom{x+\theta-1}{x} + \log\left(\left(\frac{\mu}{\mu+r}\right)^{x}\right) + \log\left(\left(\frac{r}{\mu+r}\right)^{\theta}\right) \\
&= \log\binom{x+\theta-1}{x} + x\log\left(\frac{\mu}{\mu+r}\right) + \theta\log\left(\frac{r}{\mu+r}\right) \\
&= \log\Gamma(x+\theta) - \log\Gamma(x+1) - \log\Gamma(\theta) \\
&\quad - x\ \text{SoftPlus}(\log\theta - \log\mu) - \theta\ \text{SoftPlus}(\log\mu - \log\theta)
\end{aligned} \tag{22}$$

Finally, we add some small values to make the expression numerically stable:

$$\begin{aligned}
&\log\mathcal{L}(\log\mu, \log\theta \ ; \ x) \\
&= \log\Gamma(x+\theta+\epsilon) - \log\Gamma(x+1+\epsilon) - \log\Gamma(\theta+\epsilon) \\
&\quad - x\ \text{SoftPlus}(\log\theta - \log\mu + \epsilon) - \theta\ \text{SoftPlus}(\log\mu - \log\theta + \epsilon)
\end{aligned} \tag{23}$$

Where $\log\mu$ and $\log\theta$ represent mean and dispersion in Negative Binomial distribution, x is the observed expression in count space, Γ is the gamma function used as a smoothed version of the factorial function ($\Gamma(n) = (n-1)!$), and ϵ is a small value to increase numerical stability. Please note that although θ and $\log\theta$ both appear in the expression, we were able to parametrize the mean parameter (μ) solely based on the log value. This increases the precision in our model since $\log\mu$ is the parameter composed of multiple values aggregated together, and the gradients for this parameter are of great interest.

Supplemental Note 8: Interpretation of Latent Factors via Additive Decoder Architecture

To interpret the latent factors in terms of the genes involved, we make use of the additive decoder architecture and explain each decoder subnetwork separately. We start with the average pooling as the simpler case. For DRVI-AP, the generative process is defined as:

$$\log \mu = \frac{1}{K} \sum_{1 \leq k \leq K} f^{(k)}(z_k, c) \tag{24}$$

where μ, z_k, $f^{(k)}$, and c indicate the mean parameter of the generative distribution, kth latent factor, kth decoder subnetwork in the additive decoder, and the covariate vector, respectively. In this case, for gene g and factor i of the latent space we have:

$$\log \mu_g = \sum_{k \in K \wedge k \neq i} \left[\frac{1}{K} f^{(k)}(z_k, c) \right]_g + \left[\frac{1}{K} f^{(i)}(z_i, c) \right]_g \tag{25}$$

When perturbing z_i, the LFC of μ_g solely depends on $f^{(i)}(z_i, c)$. We take the mean over covariates to consider the average effect over different batches. Since $f^{(i)}$ is a nonlinear function, we are interested in the maximum effect of this latent factor. Accordingly, we maximize the effect over different choices of z_i. In practice, we observe factors encoding two different concepts in their negative and positive directions. So, we split these two directions for better interpretability. Consequently, the effect of dimension Z_i on gene g is defined as:

$$\text{effect}^{\text{AP}}_{\text{lfc}}(Z_i^-, g) = \mathop{\mathbb{E}}_{c \in C} \left[\max_{z \in [\min(Z_i), 0]} \left[\frac{1}{K} f^{(i)}(z, c) \right]_g - \min_{z \in [\min(Z_i), \max(Z_i)]} \left[\frac{1}{K} f^{(i)}(z, c) \right]_g \right] \tag{26}$$

$$\text{effect}^{\text{AP}}_{\text{lfc}}(Z_i^+, g) = \mathop{\mathbb{E}}_{c \in C} \left[\max_{z \in [0, \max(Z_i)]} \left[\frac{1}{K} f^{(i)}(z, c) \right]_g - \min_{z \in [\min(Z_i), \max(Z_i)]} \left[\frac{1}{K} f^{(i)}(z, c) \right]_g \right] \tag{27}$$

These effects can be efficiently approximated by sampling from covariates and traversing (one-dimensional) discrete values over each latent factor.

For LSE pooling function, the generative process is:

$$\log \mu = \log \sum_{1 \leq k \leq K} \exp\left(f^{(k)}(z_k, c) \right) \tag{28}$$

where μ, z_k, $f^{(k)}$, and c are defined as in the average pooling case.

Accordingly, for each gene g and factor i of the latent space we have:

$$\mu_g = \sum_{k \in K \wedge k \neq i} \left[\exp\left(f^{(k)}(z_k, c) \right) \right]_g + \left[\exp\left(f^{(i)}(z_i, c) \right) \right]_g \tag{29}$$

Here, the effect of perturbing z_i in count space solely depends on $f^{(i)}(z_i, c)$. Formally:

$$\begin{aligned} \text{effect}^{\text{LSE}}_{\text{counts}}(Z_i, g) = \mathop{\mathbb{E}}_{c \in C} \Bigg[& \max_{z \in [\min(Z_i), \max(Z_i)]} \left[\exp\left(f^{(i)}(z, c) \right) \right]_g - \\ & \min_{z \in [\min(Z_i), \max(Z_i)]} \left[\exp\left(f^{(i)}(z, c) \right) \right]_g \Bigg] \end{aligned} \tag{30}$$

However, since we are usually interested in the LFC rather than the count space effect, we also have to consider the effect of other latent factors. The LFC will be more significant if other factors do not affect the target gene and will be smaller otherwise. Here, we make use of the additiveness property to write the minimum/maximum effect of a group of factors as the sum of the minimum/maximum effect of each individual factor. Accordingly, to calculate the maximum possible effect, we have:

$$\text{effect}^{\text{LSE}}_{\text{max-lfc}}(Z_i^-, g) = \mathop{\mathbb{E}}_{c\in C}\left[\log\left(\frac{\Delta_{min} + \max_{z\in[\min(Z_i),0]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g}{\Delta_{min} + \min_{z\in[\min(Z_i),\max(Z_i)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g + \epsilon}\right)\right] \tag{31}$$

$$\text{effect}^{\text{LSE}}_{\text{max-lfc}}(Z_i^+, g) = \mathop{\mathbb{E}}_{c\in C}\left[\log\left(\frac{\Delta_{min} + \max_{z\in[0,\max(Z_i)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g}{\Delta_{min} + \min_{z\in[\min(Z_i),\max(Z_i)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g + \epsilon}\right)\right] \tag{32}$$

Where Δ_{min} is defined as $\sum_{k\neq i}\min_{z\in[\min(Z_k),\max(Z_k)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g$ and ϵ is a small number defined for numerical stability.

Likewise, for the minimum effect, we define:

$$\text{effect}^{\text{LSE}}_{\text{min-lfc}}(Z_i^-, g) = \mathop{\mathbb{E}}_{c\in C}\left[\log\left(\frac{\Delta_{max} + \max_{z\in[\min(Z_i),0]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g}{\Delta_{max} + \min_{z\in[\min(Z_i),\max(Z_i)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g + \epsilon}\right)\right] \tag{33}$$

$$\text{effect}^{\text{LSE}}_{\text{min-lfc}}(Z_i^+, g) = \mathop{\mathbb{E}}_{c\in C}\left[\log\left(\frac{\Delta_{max} + \max_{z\in[0,\max(Z_i)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g}{\Delta_{max} + \min_{z\in[\min(Z_i),\max(Z_i)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g + \epsilon}\right)\right] \tag{34}$$

Where Δ_{max} is defined as $\sum_{k\neq i}\max_{z\in[\min(Z_k),\max(Z_k)]}\left[\exp\left(f^{(i)}(z,c)\right)\right]_g$.

All the minimum and maximum values over decoder subnetworks can be efficiently approximated by traversing discrete ranges over the latent factors. Averaging over categorical covariates can be done by sampling. The same approach applies to DRVI with any other pooling function.

The minimum and maximum effects measure two different aspects of the latent factors.

The $\text{effect}_{\text{max-lfc}}$ measures the maximum effect of a latent factor regardless of the baseline expression. For example, for two genes with similar average expression, changing a log-mean parameter of a gene from 0.1 to 0.5 has the same score as changing the other from 1 to 5. The $\text{effect}_{\text{min-lfc}}$ score is robust to such cases by taking the maximum expression of other factors into account as a baseline. Still, it drops significantly if a gene participates in multiple latent factors. Consequently, we combine the two scores by multiplying them. Formally: $\text{effect}^{\text{LSE}}_{\text{combined}}(Z_i^*, g) = \text{effect}^{\text{LSE}}_{\text{max-lfc}}(Z_i^*, g) * \text{effect}^{\text{LSE}}_{\text{min-lfc}}(Z_i^*, g)$ where Z_i^* can be either Z_i^- or Z_i^+. Additionally, we remove all the insignificant effects meeting any of the following criteria:

$$\text{remove effect}^{\text{LSE}}_{\text{combined}}(Z_i^*, g) \text{ if any } \begin{cases} \text{effect}^{\text{LSE}}_{\text{max-lfc}}(Z_i^*, g) < 1 \\ \text{effect}^{\text{LSE}}_{\text{max-lfc}}(Z_i^*, g) < \max_{g'\in G}\left[\text{effect}^{\text{LSE}}_{\text{max-lfc}}(Z_i^*, g')\right]/2 \\ \text{effect}^{\text{LSE}}_{\text{min-lfc}}(Z_i^*, g) < \max_{g'\in G}\left[\text{effect}^{\text{LSE}}_{\text{min-lfc}}(Z_i^*, g')\right]/10 \end{cases} \tag{35}$$

We found the mentioned interpretability pipeline accurate enough for our analysis. However, further tuning of the pipeline and the combining logic can be a follow-up, which is out of the scope of the current study. A pseudocode for the DRVI pipeline, including training and interpretability, is provided in Supplementary Notes 10.

Since the introduced approach directly reflects the model's understanding of the generative process, it is limited to the set of genes included in the training data. Alternatively, one can perform differential expression (DE) analysis to identify the genes involved in a nonlinear gene program beyond the genes used for model training. To this end, cells can be categorized into high-valued and low-valued groups based on a latent factor, or latent factors can be incorporated into the design matrix of a regression model for DE testing. While DE analysis is a well-established approach, it may highlight genes that

are not specific to a process but are simply expressed at higher-than-average levels in the target population. In addition, lowly expressed genes may be missed.

Besides looking up the literature and marker databases, we used the Python interface of the gProfiler service [157] for GSEA analysis. For each factor, we selected the top 100 genes associated with the nonlinear gene program as the input set. To account for the biological bias introduced by the selection of HVGs, they were given as the reference set to the gProfiler.

Supplemental Note 9: Ordering of Latent Factors

Unlike PCA, where principal components are ordered according to the explained variance, the latent factors learned by autoencoders and many other latent space models are inherently unordered. Nevertheless, we take advantage of the additive architecture and apply heuristics to sort the latent factors based on their contribution to the reconstruction.

Heuristically, we remove each factor and check how much it will affect the reconstruction. Assume we have an observed count matrix $\boldsymbol{X} \in \mathbb{N}^{N \times M}$, a trained DRVI model with latent space $\boldsymbol{Z}$, and the corresponding fitted generative process based on the decoder f. Then for kth latent factors we construct matrix $\boldsymbol{S}^{(k)} \in \mathbb{R}^{N \times M}$ where its ith row (corresponding to the sample i) is defined as:

$$\boldsymbol{S}_i^{(k)} = f^{(k)}(\boldsymbol{Z}_{i,k}, \boldsymbol{C}_i). \tag{36}$$

The reconstructed log-mean parameter in DRVI-AP will be $\sum \boldsymbol{S}^{(k)}$. Therefore, removing the ith factor will change the log-space parameters by $|\boldsymbol{S}^{(k)}|$. So, for the average pooling case, we order latent factors based on the L1 norm of the $\boldsymbol{S}^{(k)}$ matrix.

For DRVI with LSE pooling, the effect of removing the ith factor will be:

$$\log \sum_k \exp(\boldsymbol{S}^{(k)}) - \log \sum_{k \neq i} \exp(\boldsymbol{S}^{(k)}). \tag{37}$$

For stability of the effect, we add $\epsilon = 1$ to the logarithm function. Therefore, the overall impact of factor i is defined as:

$$\log \left(\epsilon + \sum_k \exp(\boldsymbol{S}^{(k)}) \right) - \log \left(\epsilon + \sum_{k \neq i} \exp(\boldsymbol{S}^{(k)}) \right) = -\log \left(1 - \frac{\exp(\boldsymbol{S}^{(i)})}{\epsilon + \sum_k \exp(\boldsymbol{S}^{(k)})} \right), \tag{38}$$

where the latter can be efficiently computed by the softmax function.

Supplemental Note 10: DRVI's code functionality

Below is a pseudocode demonstrating how to apply DRVI for training, latent space visualization, and interpretability. The parts unrelated to DRVI are provided with abstract functions starting with `abs_`.

```
# Load Dataset
adata = abs_read_data()

# Highly-Variable Gene Selection
hvg_genes = abs_hvg_calculation(adata)
adata = adata[:, hvg_genes]

# Train DRVI
DRVI.setup_anndata(adata, layer="counts", categorical_covariate_keys=["batch"])
model = DRVI(adata, n_latent=64, encoder_dims=[256, 256], decoder_dims=[256, 256])
model.train(max_epochs=400)                    # Train Model for 400 Epochs

# Latent Space Analysis
```

```
embed = model.get_latent_representation()      # Obtain Latent Representation
embed = abs_preprocess_latent_space(embed)     # Neighbors and UMAP Embedding

# Latent Factor Statistics (e.g., Find Vanished Factors)
drvi.utils.tl.set_latent_dimension_stats(model, embed)
drvi.utils.pl.plot_latent_dimension_stats(embed)

# Visualize Latent Space and Activity of Individual Factors
abs_visualize_umap(embed, color=["batch", "annotation"])
drvi.utils.pl.plot_latent_dims_in_umap(embed)  # Plot Latent Activities on UMAP
# Make a Heatmap of Latent Factors Grouped by Cell Types (e.g., Fig. 3b)
drvi.utils.pl.plot_latent_dims_in_heatmap(embed, "annotation", title_col="title")

# Traverse Latent Space and Find Top Relevant Genes
traverse_adata = drvi.utils.tl.traverse_latent(model, embed, n_samples=20)
drvi.utils.tl.calculate_differential_vars(traverse_adata)
drvi.utils.pl.show_top_differential_vars(traverse_adata, key="combined_score")

# Explore Individual Factors (e.g., DR 8+) to Get Insights
# Details of Differential Variables
drvi.utils.pl.show_differential_vars_scatter_plot(
    traverse_adata,
    key_x="max_possible",
    key_y="min_possible",
    key_combined="combined_score",
    dim_subset=["DR 8+"],
)
# Plot Top Relevant Genes on UMAP
drvi.utils.pl.plot_relevant_genes_on_umap(embed, traverse_adata, dims=["DR 8+"])
```

Detailed code and usage are available in the project github repository.

Supplemental Tables

Method	Disentangled Entities	Supervised input	Representational limitations
inVAE [123]	Two groups of latent factors	Batch / [biological and technical factors]	-
MIDAS [122]		Batch	
ContrastiveVI [124], SC-VAE [125]		Perturbations	
BIOLORD [128]	Multiple groups of latent factors	Variations of interest for each group	
scDisInFact [127]		Variations of interest for each group / Perturbations	
sVAE / sVAE+ [40]	Each latent factor	Perturbations	-
SAMS-VAE [41]		Perturbations	
sVAE-ligr [42]		Perturbations / Variations of interest	
Decipher [45]	Two level-2 latent factors	Variables of interest	-
pmVAE [44]	Each latent factor	Pathways	Nonlinear with pathway masks
expiMap [43]		Pathways	Linear decoder with pathway masks and sparsity
Spectra [26]		Gene-gene knowledge graph and Cell-types	Linear decomposition and sparsity
MOFA / MOFA+ [38]	Each latent factor	No	Linear decomposition and sparsity
BAE [39]			Sparse linear encoder
β-TCVAE, MICHIGAN [46, 47]			Unconstrained
DRVI (Ours)			Additive decoders

Supplemental Table 1 | Comparison of various methods for disentangling latent factors in single-cell omics.

Dataset Name	# Cells	# Batches	# Cell Types	# HV features
Developmental Pancreas [4]	3,696	-	15	2,000
Immune PBMC [52]	32,484	9	16	2,000
Retina Organoid [10]	98,348	41	21	2,001
CRISPR screen [5]	104,339	-	-	5,000
Daniocell [64]	489,686	-	156	2,000
HLCA-core [7]	584,944	166	61	1,996
PBMC [63]	647,366	3	51	2,000
NeurIPS 2021 - ATAC [90]	69,247	13	22	14,865

Supplemental Table 2 | Summary of Datasets.

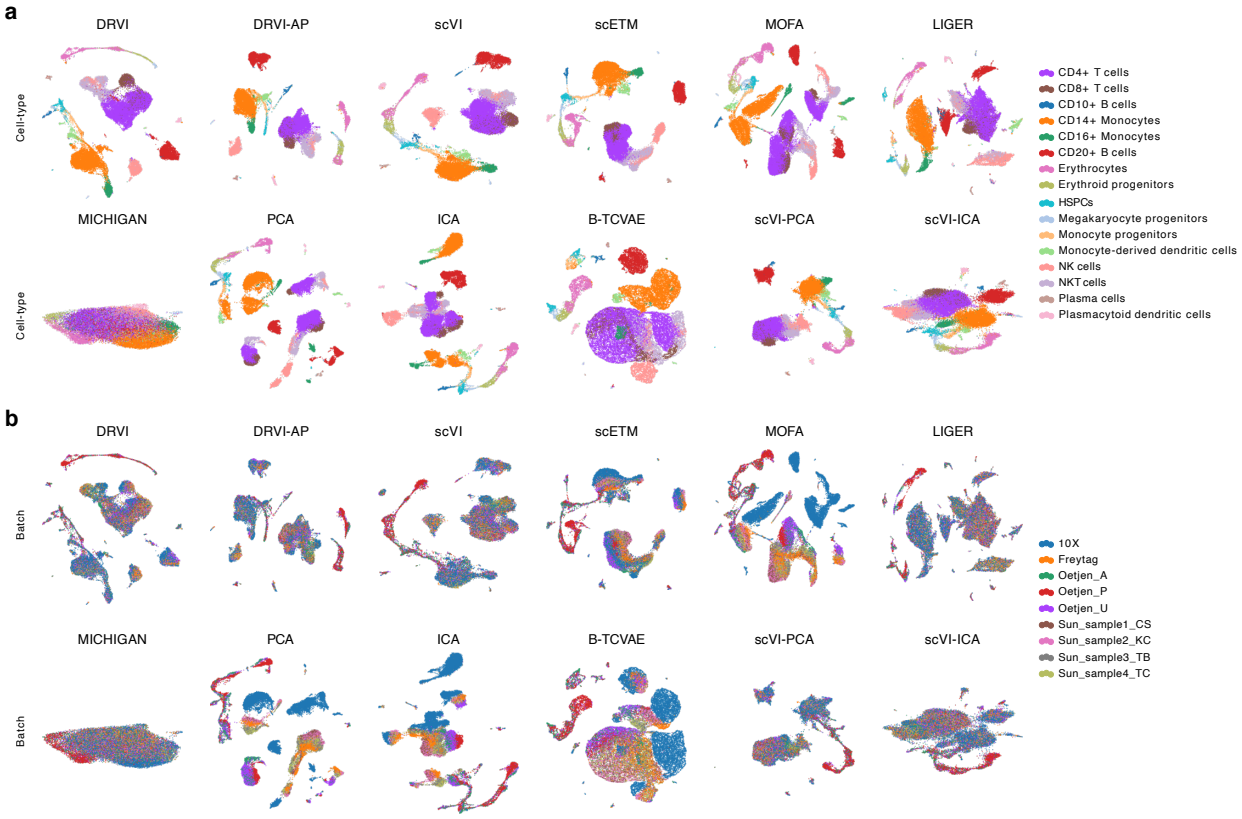
Dataset Name	**# Latent for all methods**	**Hidden Layers for DRVI / scVI**	**Sharing decoder subnetworks**	**Training epochs**
Developmental Pancreas	32	128	all layers	1,000
Immune PBMC	32	128	all layers	400
Retina Organoid	32	128	all layers	400
CRISPR screen	64	512, 512, 512	all layers	400
Daniocell	64	256, 256	last layer	400
HLCA-core	64	256, 256	all layers	400
PBMC	64	256, 256	all layers	400
NeurIPS 2021 - ATAC	64	256, 256	all layers	400

Supplemental Table 3 | Summary of Model Parameters.

Metric Family	Latent Matching Score (LMS)	Most Similar Averaging Score (MSAS)	Most Similar Gap Score (MSGS)
What is measured?	Best one-to-one matching of targets and latent dimensions (no latent can be selected twice in a matching)	Average of the most similar latent for each target	Average of the most similar latent for each target minus the average of the second most similar latent for each target
Intuitive meaning	Best one-to-one matching	Coverage	Coverage and independence
Penalizes one latent representing more than one target variable (Most important aspect, as it prevents interpretability)	**Implicitly Yes (desirable)** **Since no latent can be selected more than once.**	**No (undesirable)**	**No (undesirable)**
Penalizes one target variable being represented by multiple latents (redundancy, does not affect interpretability, but increases user efforts)	**No (undesirable)**	**No (undesirable)**	**Yes (desirable)** Heavily penalizes redundancy
Penalizes one coarse target variable being represented with more granular latents (Supplementary Fig. 9)	**No (desirable)** It does not give credit to better models either	**No (desirable)** It does not give credit to better models either	**Yes (undesirable)** Heavily penalizes better models
Penalizes models when target variables are not independent (e.g., inflammation and cytokine activity, or cell cycle and cell types)	**No (desirable)**	**No (desirable)**	**Yes (undesirable)** Penalized models favorably keep the correlations of ground-truth

Supplemental Table 4 | Comparison of Different Metric Families.

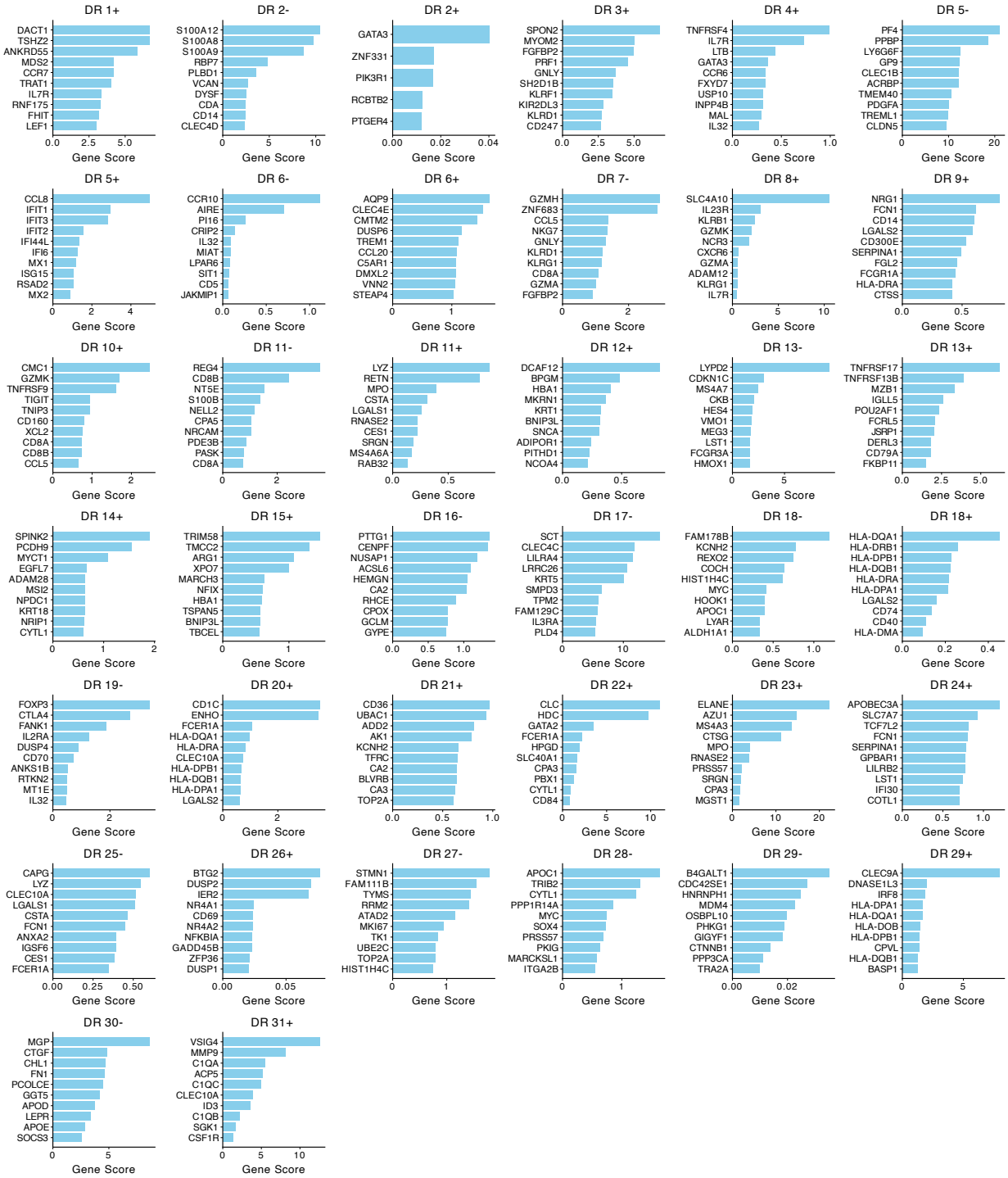
Supplemental Figures



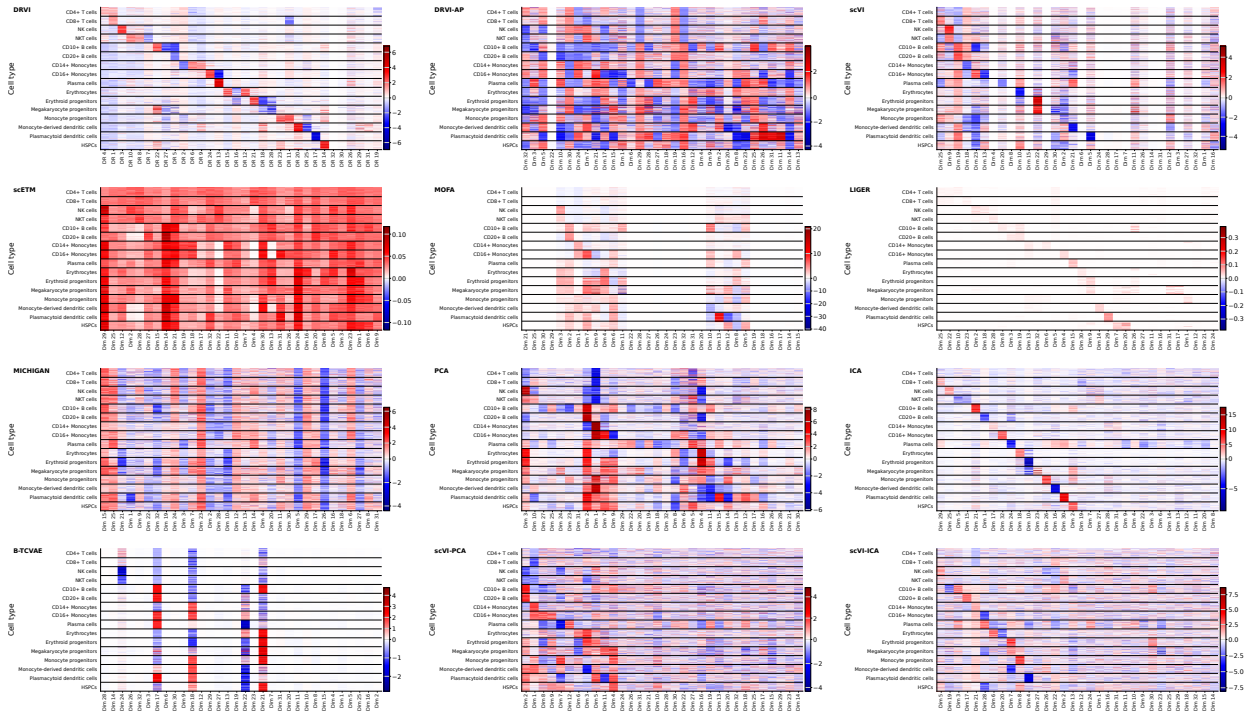
c

Method	Bio conservation					Batch correction					Aggregate score		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	Silhouette batch	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	Total
LIGER	0.55	0.67	0.52	0.59	1.00	0.86	0.33	0.59	0.78	0.83	0.68	0.67	0.67
DRVI-AP	0.63	0.74	0.53	0.58	1.00	0.91	0.26	0.23	0.87	0.78	0.61	0.70	0.66
DRVI	0.59	0.71	0.50	0.61	1.00	0.88	0.22	0.32	0.84	0.81	0.61	0.68	0.65
scVI-PCA	0.54	0.67	0.42	0.52	0.99	0.95	0.35	0.30	0.87	0.87	0.67	0.63	0.64
scVI	0.59	0.66	0.38	0.54	1.00	0.92	0.30	0.31	0.91	0.81	0.65	0.63	0.64
scVI-ICA	0.53	0.60	0.38	0.51	0.98	0.95	0.39	0.34	0.84	0.92	0.69	0.60	0.64
scETM	0.64	0.69	0.42	0.60	1.00	0.89	0.10	0.19	0.88	0.31	0.48	0.67	0.59
ICA	0.66	0.70	0.42	0.54	1.00	0.89	0.07	0.11	0.90	0.20	0.43	0.66	0.57
MOFA	0.64	0.67	0.45	0.59	1.00	0.82	0.08	0.18	0.87	0.00	0.39	0.67	0.56
PCA	0.66	0.67	0.42	0.60	1.00	0.86	0.04	0.10	0.85	0.00	0.37	0.67	0.55
B-TCVAE	0.68	0.63	0.32	0.58	1.00	0.84	0.11	0.20	0.92	0.00	0.41	0.64	0.55
MICHIGAN	0.52	0.28	0.17	0.46	0.91	0.95	0.30	0.68	0.49	0.85	0.65	0.47	0.54

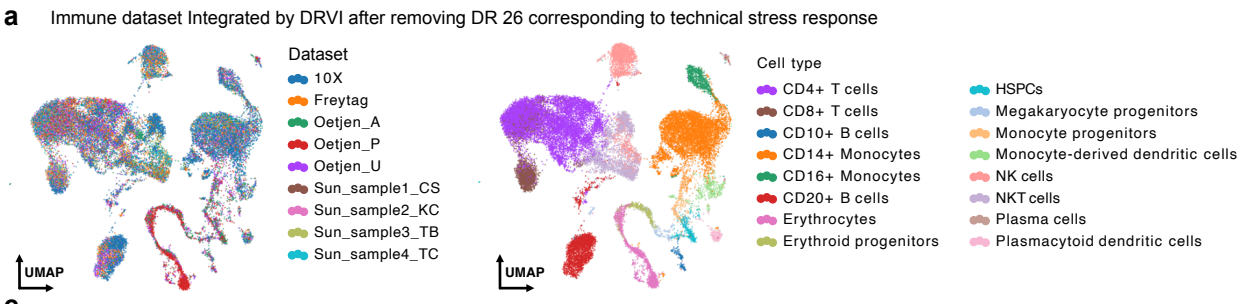
Supplemental Figure 1: **UMAPs and integration results for benchmarked methods on the immune dataset. a,b,** UMAPs of the immune dataset integrated by DRVI and benchmarked methods colored by cell types and batches. **c,** Integratin results based on the scIB metrics for DRVI and benchmarked methods.



Supplemental Figure 2: **Interpretability of DRVI latent factors in immune dataset.** The top relevant genes and their scores are plotted for each latent factor of DRVI trained on the immune dataset.

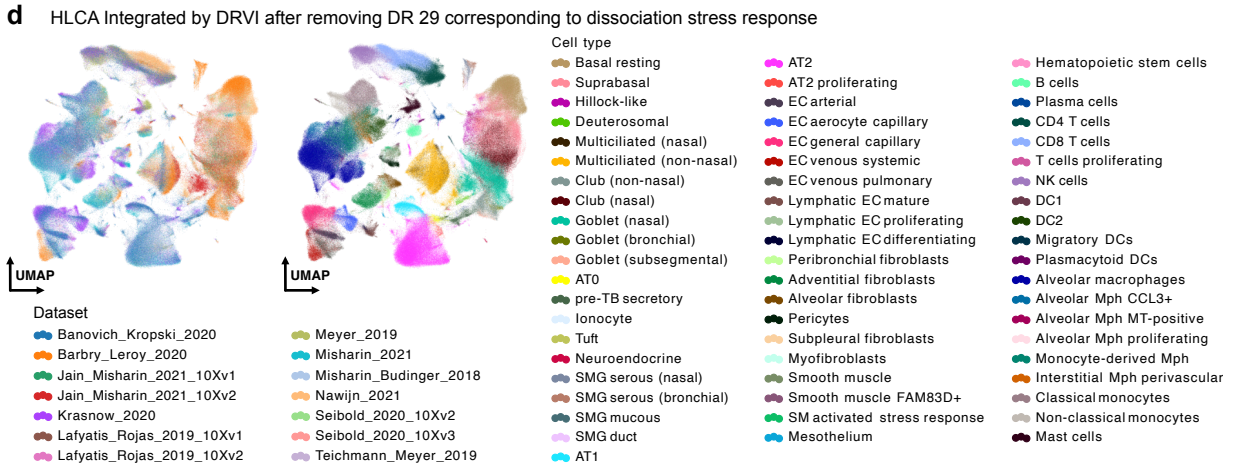


Supplemental Figure 3: **Latent space visualization for benchmarked methods in the immune dataset.** For each method, a heatmap visualizing latent representation activity across different cell types is provided. Cells were subsampled to ensure a balanced number of cells from each cell type. Dimensions are reordered using a shared algorithm to maximize the diagonal appearance.



c

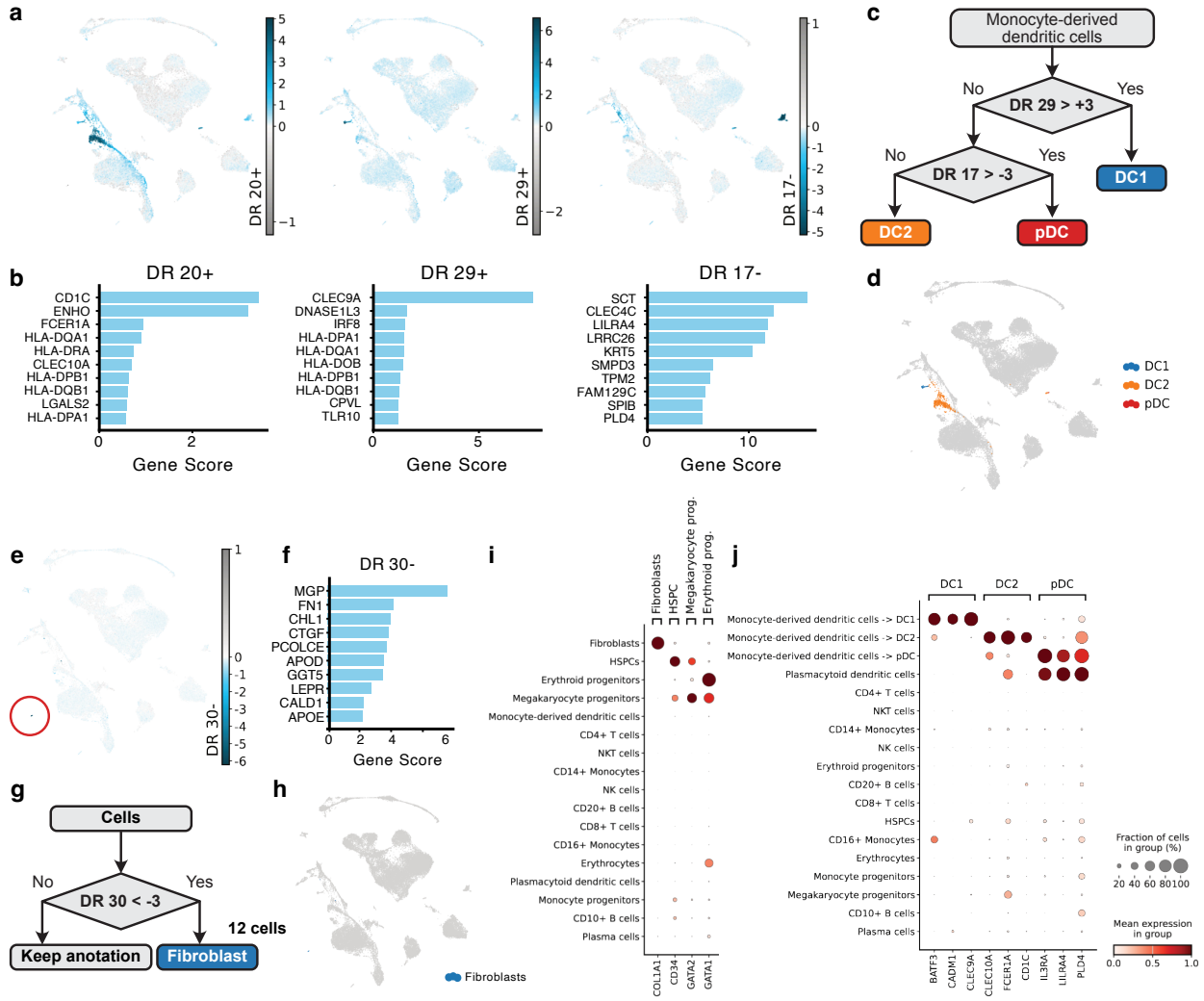
Method	Bio conservation					Batch correction					Aggregate score		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	Silhouette batch	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	Total
DRVI-AP	0.63	0.74	0.53	0.58	1.00	0.91	0.26	0.23	0.87	0.78	0.61	0.70	0.66
DRVI without DR 26	0.60	0.70	0.49	0.61	1.00	0.88	0.23	0.35	0.85	0.82	0.63	0.68	0.66
DRVI	0.59	0.71	0.50	0.61	1.00	0.88	0.22	0.32	0.84	0.81	0.61 (+2)	0.68	0.65 (+1)
scVI	0.59	0.66	0.38	0.54	1.00	0.92	0.30	0.31	0.91	0.81	0.65	0.63	0.64



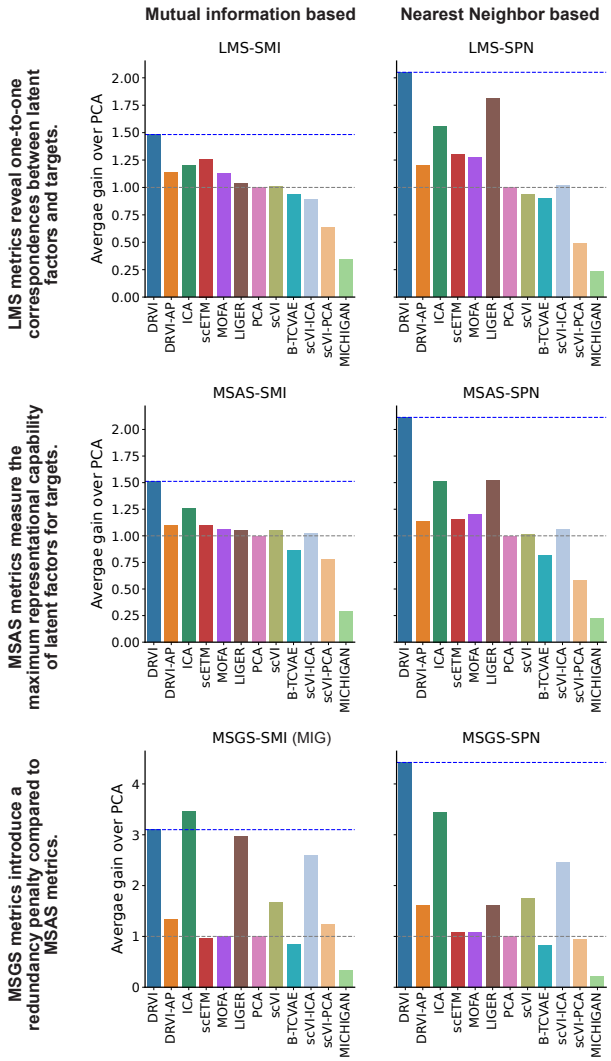
f

Method	Bio conservation					Batch correction					Aggregate score		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	Silhouette batch	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	Total
DRVI without DR 29	0.69	0.71	0.41	0.56	1.00	0.90	0.06	0.32	0.85	0.80	0.59	0.68	0.64
DRVI	0.69	0.71	0.39	0.56	1.00	0.90	0.05	0.29	0.86	0.80	0.58 (+1)	0.67 (+1)	0.63 (+1)
scANVI (HLCA original)	0.68	0.74	0.42	0.55	1.00	0.90	0.02	0.23	0.95	0.66	0.55	0.68	0.63
DRVI-AP	0.62	0.73	0.42	0.54	1.00	0.92	0.04	0.24	0.90	0.79	0.58	0.66	0.63
scVI	0.64	0.68	0.36	0.53	1.00	0.91	0.06	0.28	0.91	0.83	0.60	0.64	0.63

Supplemental Figure 4: **The effect of removing technical stress response factors. a,b,** UMAP representations of the integrated immune dataset, colored by dataset and cell type, following DRVI integration and removal of stress response factor (DR 26). **c,** Integration quality assessment using scIB metrics for DRVI before and after DR 26 removal. Results are compared to DRVI-AP and scVI, two other top-performing models. **d,e,** UMAP representations of the integrated HLCA, colored by dataset and finest-level annotations, following DRVI integration after removal of stress response factor (DR 29). **f,** Integration quality assessment using scIB metrics for DRVI before and after DR 29 removal. Results are compared to DRVI-AP and scVI, two other top-performing models, and scANVI provided by the HLCA. scANVI is the best-performing integration method provided by the HLCA paper, included for readers' information. However, it is not fairly comparable due to its use of supervised information and a different number of latent dimensions.

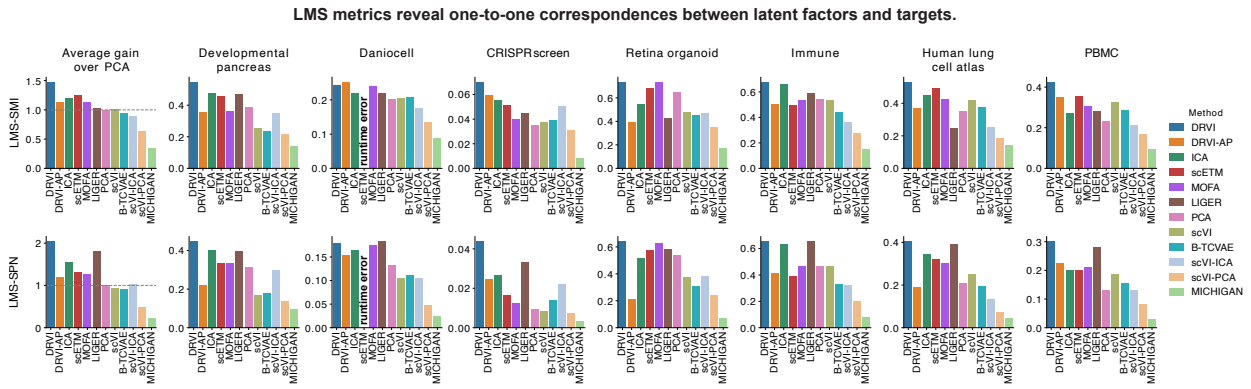


Supplemental Figure 5: **Annotation refinement and correction using DRVI in the immune dataset. a,** UMAP visualization of latent factors identifying DC subtypes. **b,** Top genes associated with DC subtype factor. **c,** Flowchart for refining DC subtype annotations. **d,** UMAP visualization of refined DC subtype annotations. **e,** UMAP visualization highlighting the latent factor identifying Fibroblast cells. The population is encircled in red. **f,** Top genes associated with the Fibroblast factor. **g,** Flowchart for annotating Fibroblast cells. **h,** UMAP visualization of annotated Fibroblast cells. **i,** Dotplot validating new annotations for Fibroblasts, HSPCs, Megakaryocyte progenitors, and Erythroid progenitors using a representative curated marker gene for each. **j,** Dotplot of marker genes for DC subtypes for validation of DC subtype improvement. For validation, pDCs are intentionally labeled differently from the original 'plasmacytoid dendritic cells' annotations. DC, dendritic cells; HSPC, Hematopoietic Stem and Progenitor Cells; pDCs, plasmacytoid dendritic cells.

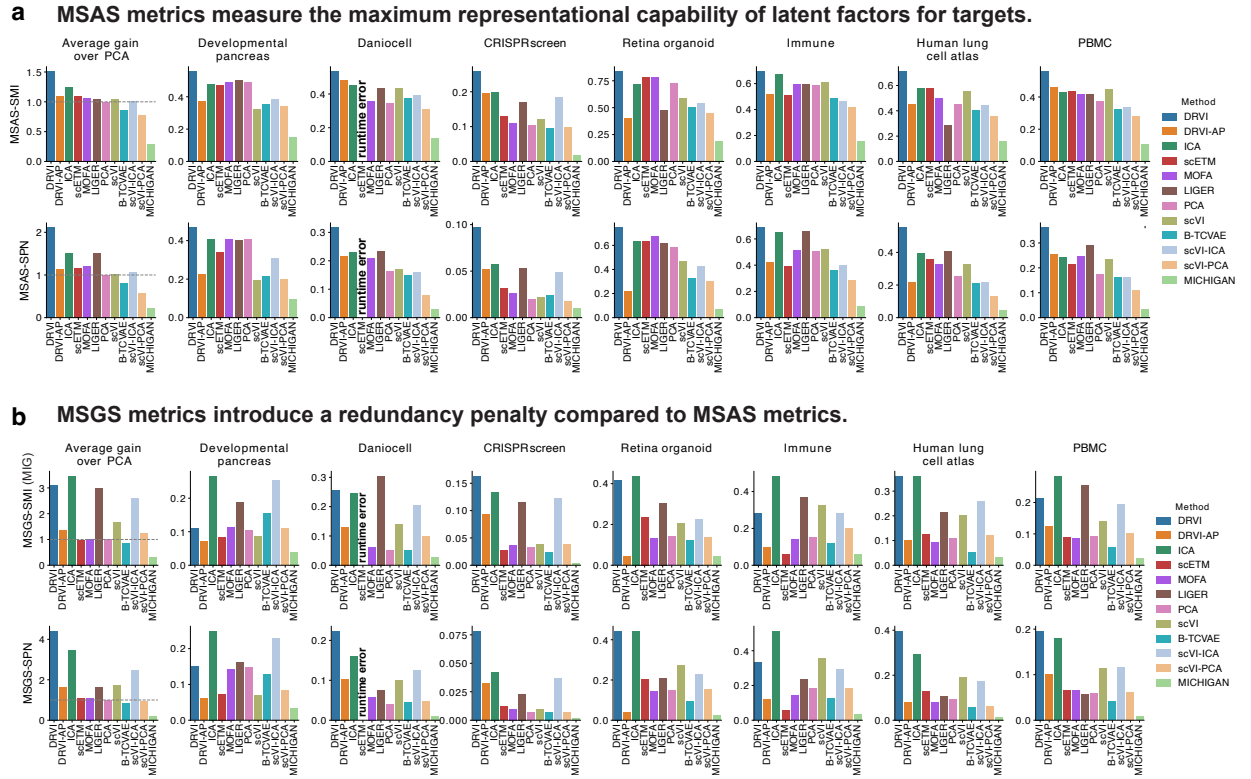


Supplemental Figure 6: **Summary of all metrics.** This plot displays the average gain achieved by each of the benchmarked methods over PCA. The gains are shown for every combination of the SMI and SPN similarity functions with the LMS, MSAS, and MSGS aggregation functions. LMS, latent matching score; MSAS, Most Similar Averaging Score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; SPN, same-process neighbors.

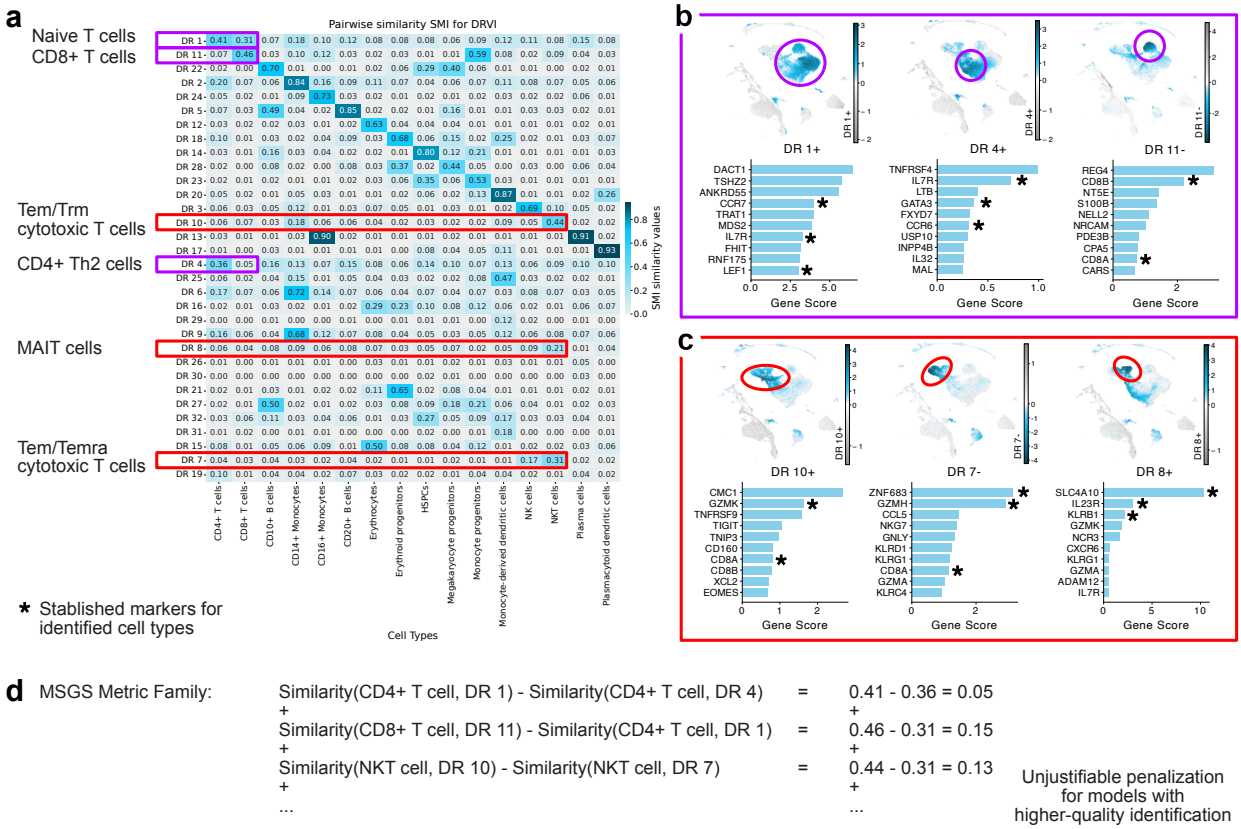
2056



Supplemental Figure 7: **Results of the primary disentanglement metric family.** The results of the LMS metrics, as primary metric family, over different datasets are provided. LMS measures the ability of the model to capture biological processes in individual latent factors by matching latent factors and ground-truth biological processes. LMS, latent matching score; SMI, scaled mutual information; SPN, same-process neighbors.



Supplemental Figure 8: **Results of the secondary disentanglement metrics. a,b,** The results of the MSAS (a) and MSGS (b) as secondary metrics over different datasets are provided. MSAS shows the capability of the model to capture biological processes in individual latent factors regardless of the redundancy. MSGS measures the amount of captured biological information that is specific to a single (probably overlapping) factor by penalizing redundancy. Unlike LMS, as the main metric, neither MSAS nor MSGS penalizes overlapping processes in a single factor. The MSGS-SMI metric is known as the Mutual Information Gap (MIG) in the field of disentangled representation learning [65]. LMS, latent matching score; MSAS, Most Similar Averaging Score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; SPN, same-process neighbors; MIG, mutual information gap.

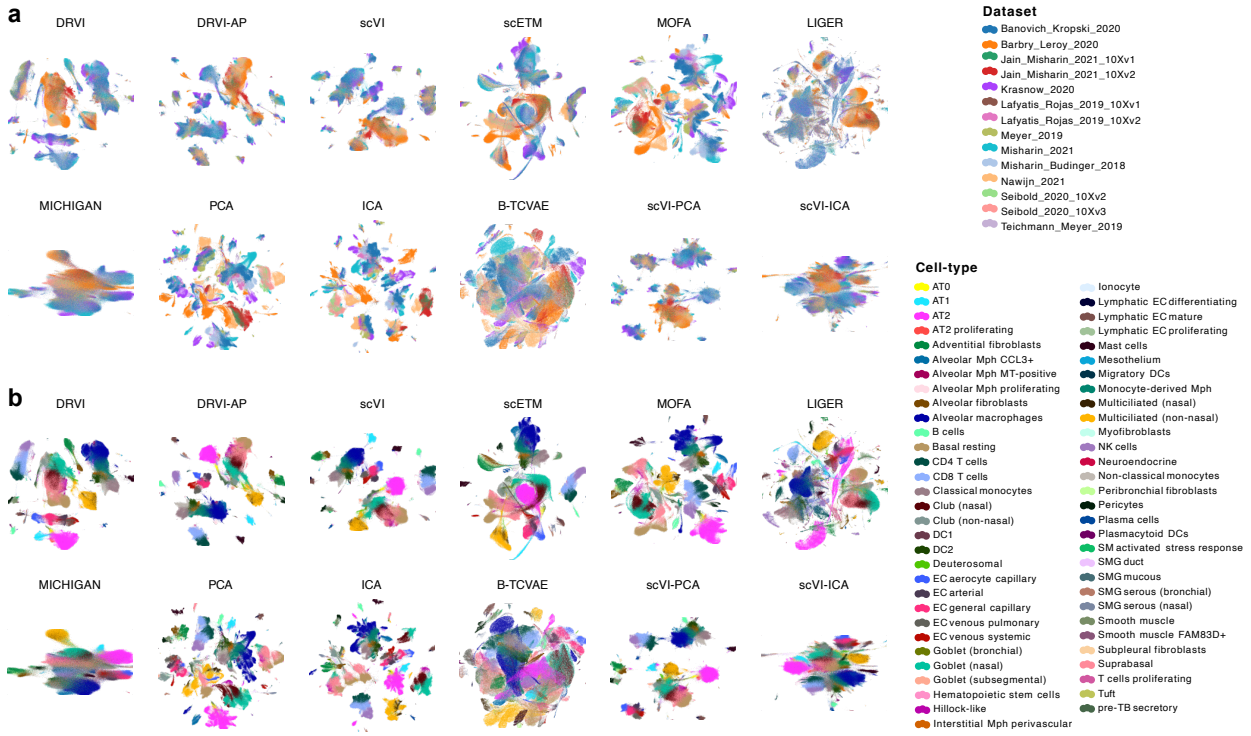


d MSGS Metric Family:

Similarity(CD4+ T cell, DR 1) - Similarity(CD4+ T cell, DR 4) = 0.41 - 0.36 = 0.05
+
Similarity(CD8+ T cell, DR 11) - Similarity(CD4+ T cell, DR 1) = 0.46 - 0.31 = 0.15
+
Similarity(NKT cell, DR 10) - Similarity(NKT cell, DR 7) = 0.44 - 0.31 = 0.13
+
...

Unjustifiable penalization for models with higher-quality identification

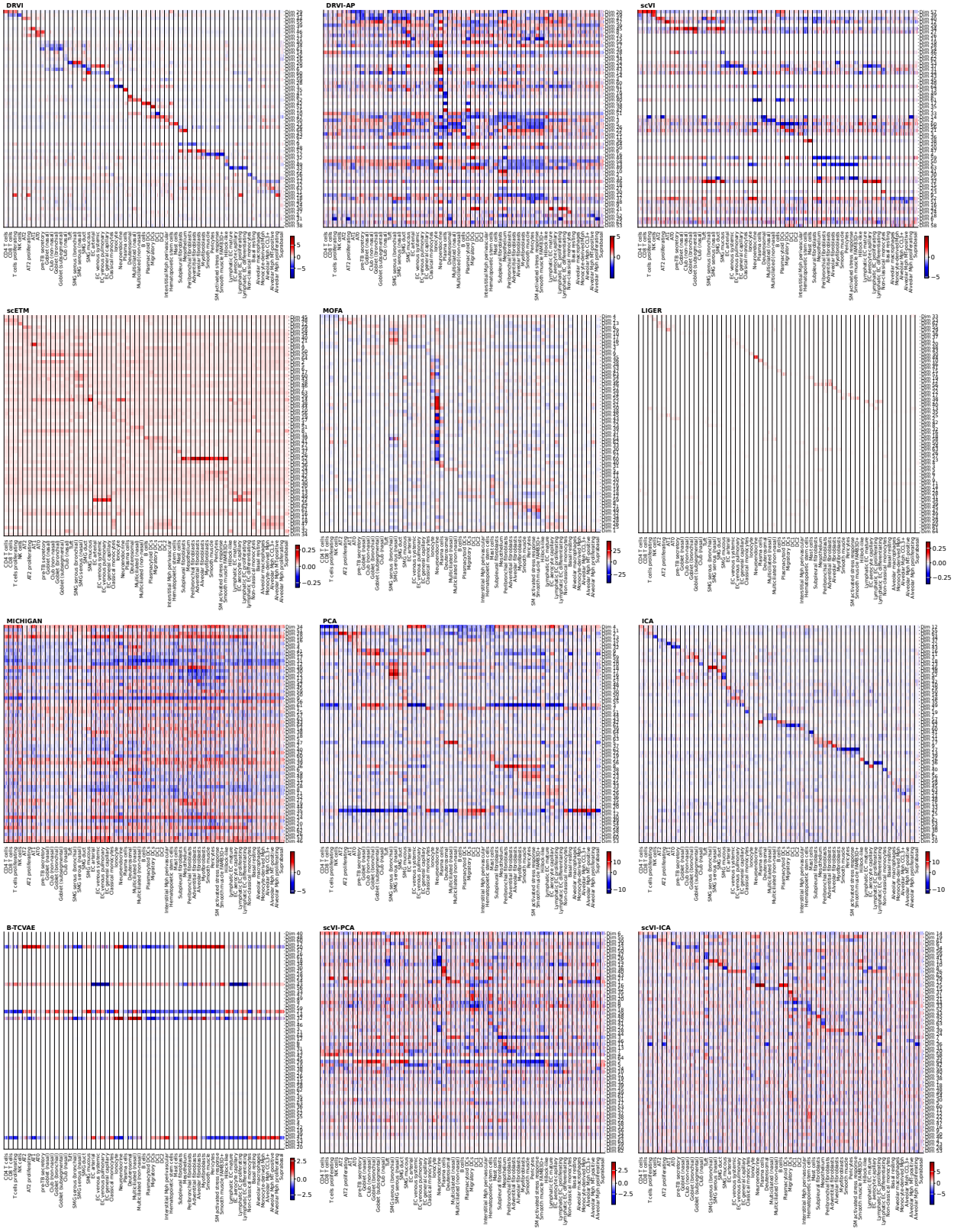
Supplemental Figure 9: **DRVI identifies granular immune cell subtypes that are penalized by MSGS metrics. a,** Pairwise Similarity Heatmap. This heatmap compares the similarity of DRVI latent factors against broad, "ground truth" cell type annotations. Highlighted rows show factors of DRVI capturing more granular subtypes. **b, c,** Granular Subtype Identification. These panels show the specific gene markers for DRVI's clusters, confirming they represent distinct, biologically relevant cell states that were grouped together in the original annotations. Panel (b) shows that DRVI correctly separates Naive T cells (DR 1), marked by CCR7 (a classic marker for naive T cells) and LEF1 (a transcription factor that is highly expressed in naive T cells) from CD4+ Th2 cells (DR 4), marked by GATA3 (transcription factor for Th2 cells) and IL7R (a cytokine critical for the survival of memory T cells). The ground truth annotation simply labels both as CD4+ T cells.. Panel (c) shows that DRVI also distinguishes between highly specific cytotoxic and innate-like T cells. It identifies Tem/Trm cytotoxic T cells (GZMK+ inflammatory CD8+ T cells, in DR 10), Tem/Temra cytotoxic T cells marked by ZNF683 (Tissue-Resident Memory CD8+ T cells, in DR 8), and MAIT cells marked by SLC4A10 and KLRB1 (DR 7). The ground truth annotations grouped these distinct types under a broad label of NKT cells. **d** problem of MSGS Metrics when ground-truth labels are imperfect. This metric family unjustifiably penalizes the DRVI model for its superior performance. By subtracting the similarity scores of valid subtypes (e.g., Similarity(CD4+ T cell, DR 1) - Similarity(CD4+ T cell, DR 4)), the metric punishes the model precisely because it was able to distinguish between Naive T cells and Th2 cells — a higher-quality identification that was more accurate than the provided ground truth.



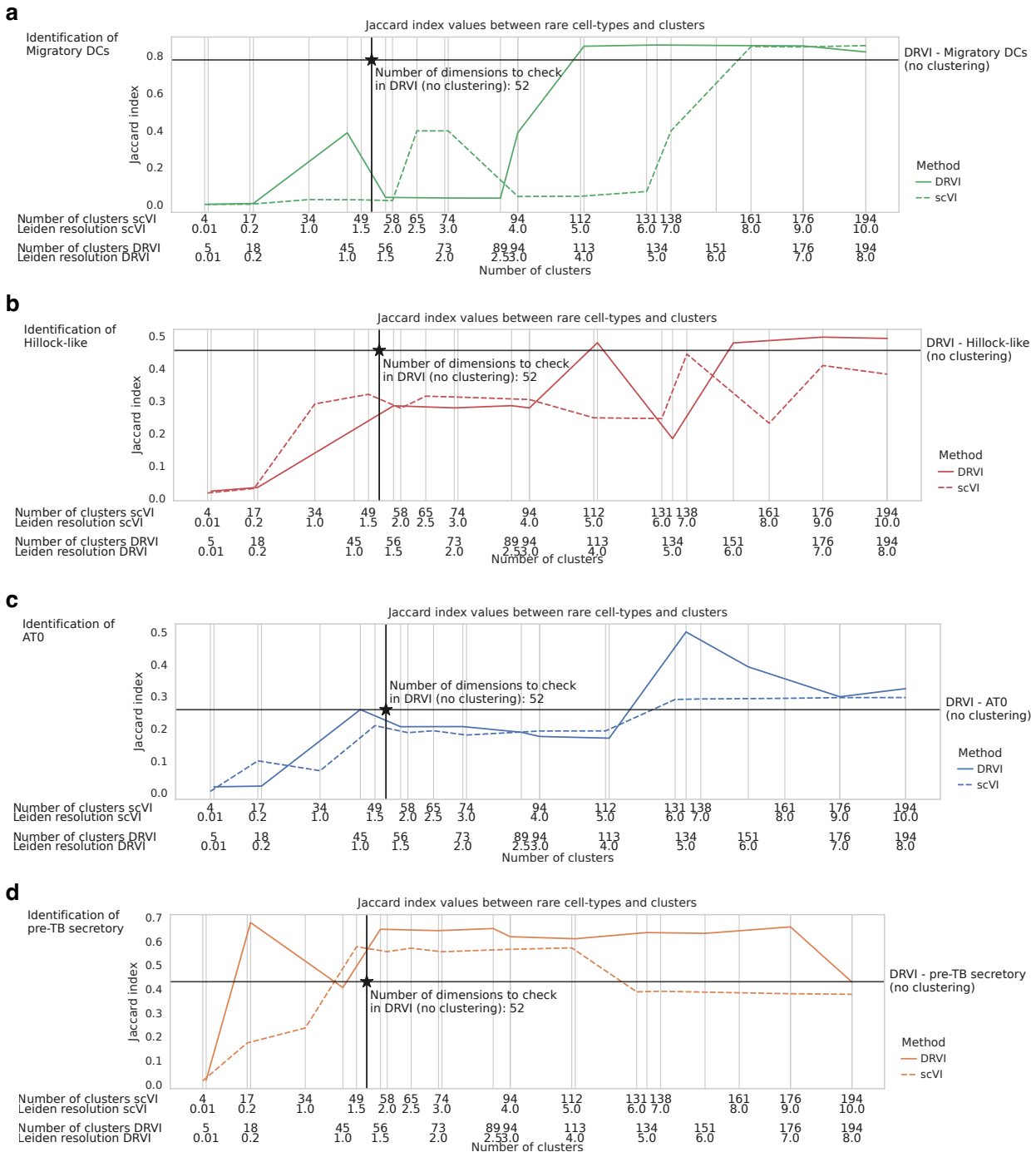
c

Method	Bio conservation					Batch correction					Aggregate score		
	Isolated labels	KMeans NMI	KMeans ARI	Silhouette label	cLISI	Silhouette batch	iLISI	KBET	Graph connectivity	PCR comparison	Batch correction	Bio conservation	Total
DRVI	0.69	0.71	0.39	0.56	1.00	0.90	0.05	0.29	0.86	0.80	0.58	0.67	0.63
scANVI (HLCA original)	0.68	0.74	0.42	0.55	1.00	0.90	0.02	0.23	0.95	0.66	0.55	0.68	0.63
DRVI-AP	0.62	0.73	0.42	0.54	1.00	0.92	0.04	0.24	0.90	0.79	0.58	0.66	0.63
scVI	0.64	0.68	0.36	0.53	1.00	0.91	0.06	0.28	0.91	0.83	0.60	0.64	0.63
MOFA	0.63	0.74	0.47	0.56	1.00	0.88	0.00	0.21	0.91	0.52	0.50	0.68	0.61
scETM	0.61	0.73	0.45	0.55	1.00	0.86	0.01	0.27	0.86	0.33	0.47	0.67	0.59
ICA	0.61	0.72	0.37	0.51	1.00	0.91	0.00	0.15	0.84	0.62	0.50	0.64	0.59
LIGER	0.42	0.54	0.30	0.45	0.99	0.82	0.14	0.51	0.56	0.84	0.57	0.54	0.55
PCA	0.60	0.73	0.39	0.53	1.00	0.88	0.00	0.15	0.87	0.03	0.39	0.65	0.54
MICHIGAN	0.46	0.52	0.28	0.46	0.99	0.92	0.12	0.43	0.42	0.80	0.54	0.54	0.54
B-TCVAE	0.55	0.67	0.28	0.49	1.00	0.80	0.01	0.21	0.88	0.11	0.40	0.60	0.52

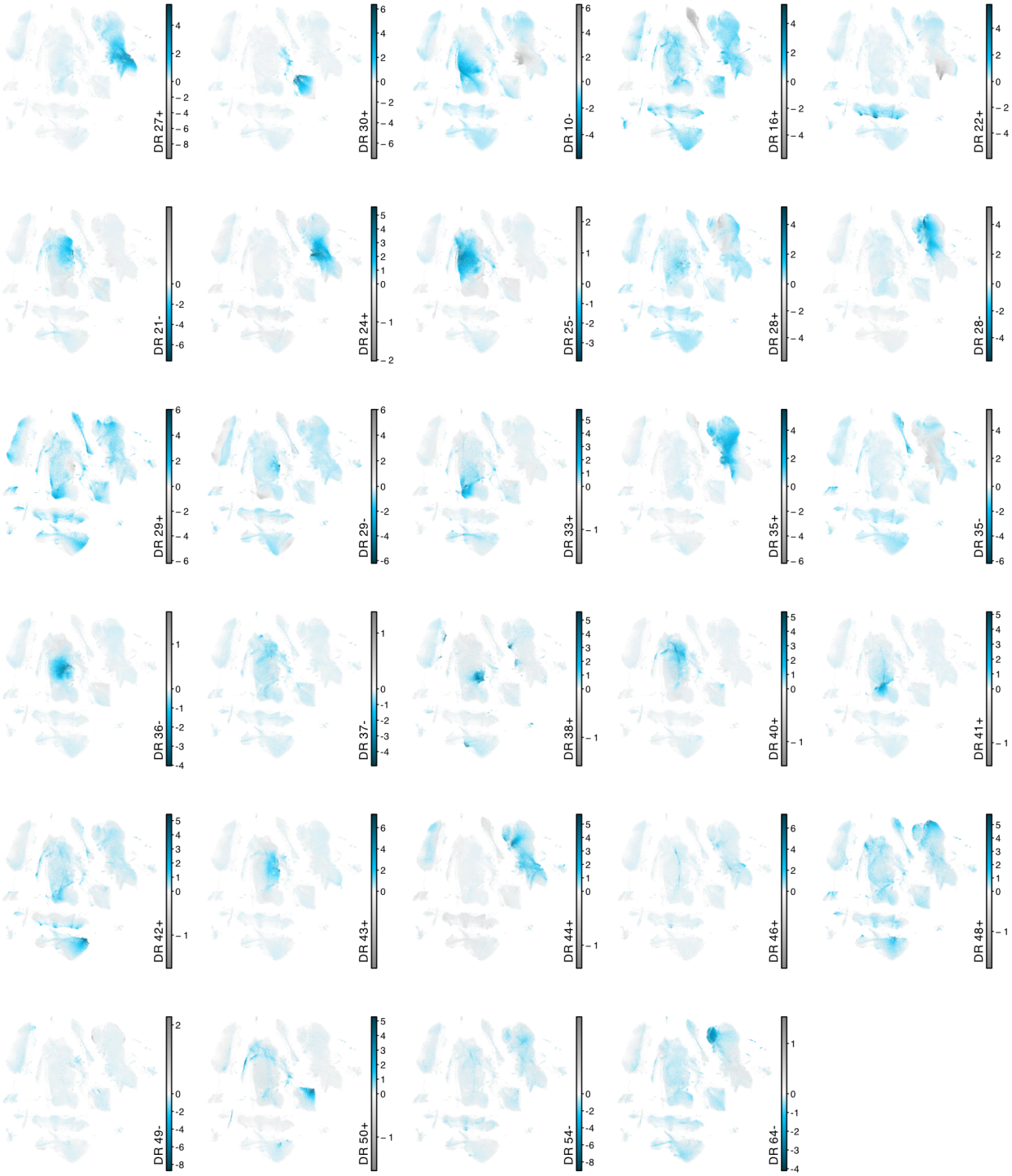
Supplemental Figure 10: **UMAPs and integration results for benchmarked methods on the HLCA dataset. a,b,** UMAPs of the HLCA integrated by DRVI and benchmarked methods colored by datasets and finest-level annotations. **c,** Integratin results based on the scIB metrics for DRVI and benchmarked methods. scANVI was originally chosen as the best-performing integration method provided by the HLCA paper, included for readers' information. However, it is not fairly comparable due to its use of supervised information and a different number of latent dimensions.



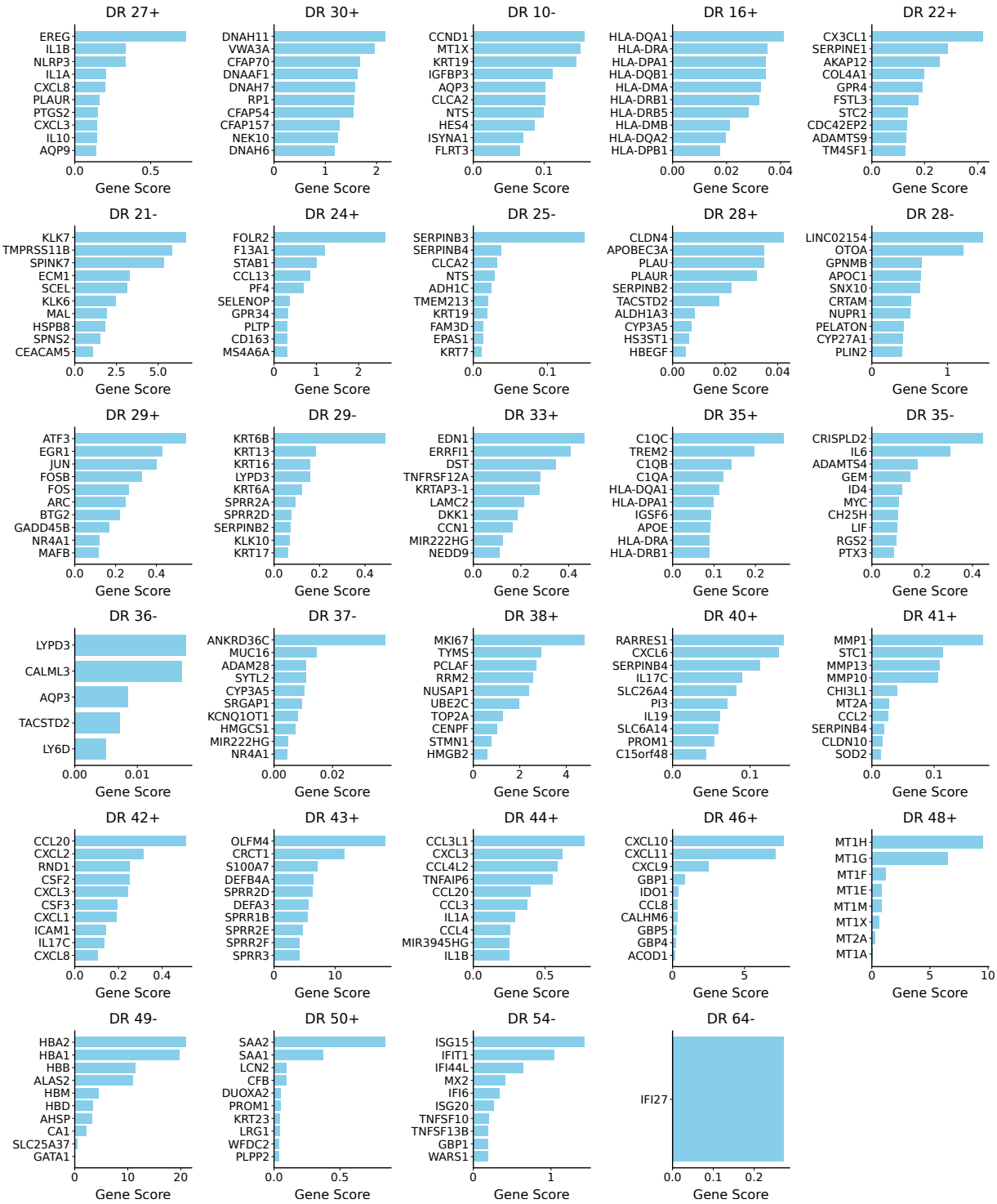
Supplemental Figure 11: **Latent space visualization for benchmarked methods on the HLCA.** For each method, a heatmap visualizing latent representation activity across different cell types is provided. Cells were subsampled to ensure a balanced number of cells from each cell type. Factors are reordered using a shared algorithm to maximize the diagonal appearance.



Supplemental Figure 12: **Rare cell-type identification performance of Leiden clustering in the HLCA.** Leiden clustering was applied to DRVI and scVI embeddings at varying resolutions. For each cell type, we calculated the Jaccard index between the cell type and its most relevant cluster at different resolutions. Additionally, we evaluated the performance of DRVI by calculating the Jaccard index between the cell type and the separation achieved using a simple threshold on the most relevant DRVI factor (marked by a star). Migratory DCs were accurately captured by the Leiden algorithm only at the resolution of 8.0 with 161 clusters. AT0 cells were identified with comparable quality to DRVI starting from the resolution of 6.0 with 131 clusters. Hillock-like cells were captured as well as DRVI at the resolution of 7.0 with 138 clusters. Pre-TB secretory cells were captured well at a resolution of 1.5 with 49 clusters. These results demonstrate that identifying at least three rare cell types requires the creation of approximately 130 to 160 clusters using Leiden clustering. Given that DRVI can capture these cell types within its 52 non-vanished factors, we suggest DRVI as an efficient valuable tool for seed annotation and discovery of rare cell populations.

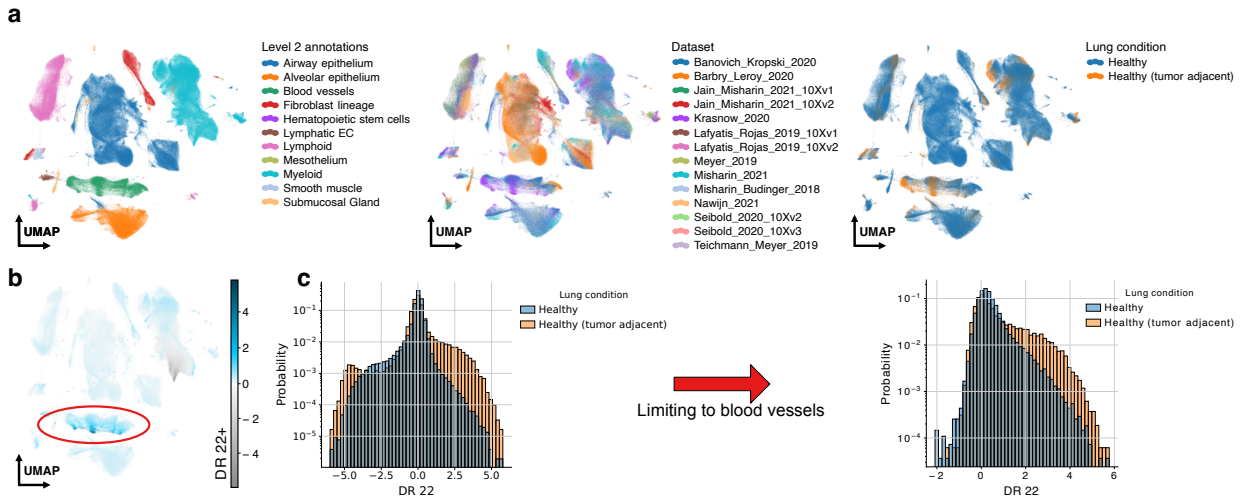


Supplemental Figure 13: **Activity of DRVI factors representing non-cell-type processes in the HLCA on UMAP** For factors representing non-cell-type processes, the activity on the UMAP is provided.

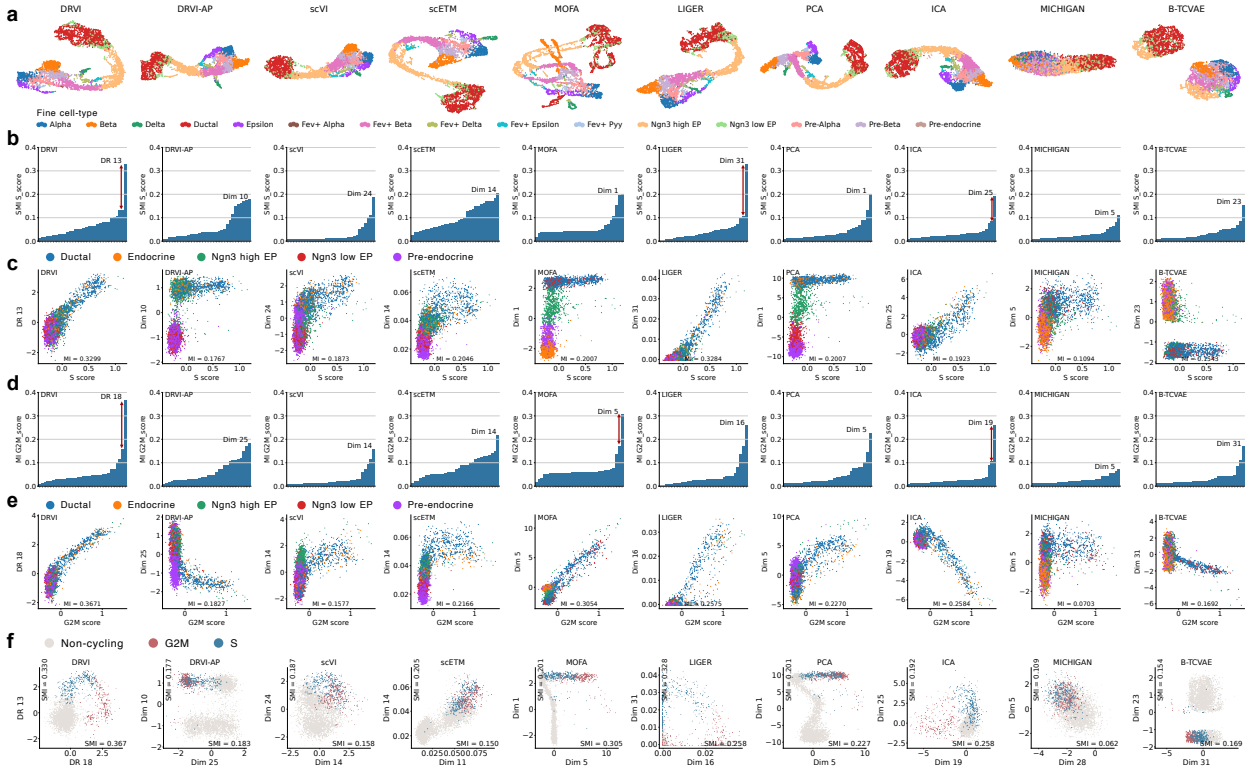


Supplemental Figure 14: **Interpretability of DRVI latent factors corresponding to non-cell-type processes in the HLCA.** The top relevant genes and their scores are plotted for each non-cell-type latent factor of DRVI trained on the HLCA.

2064

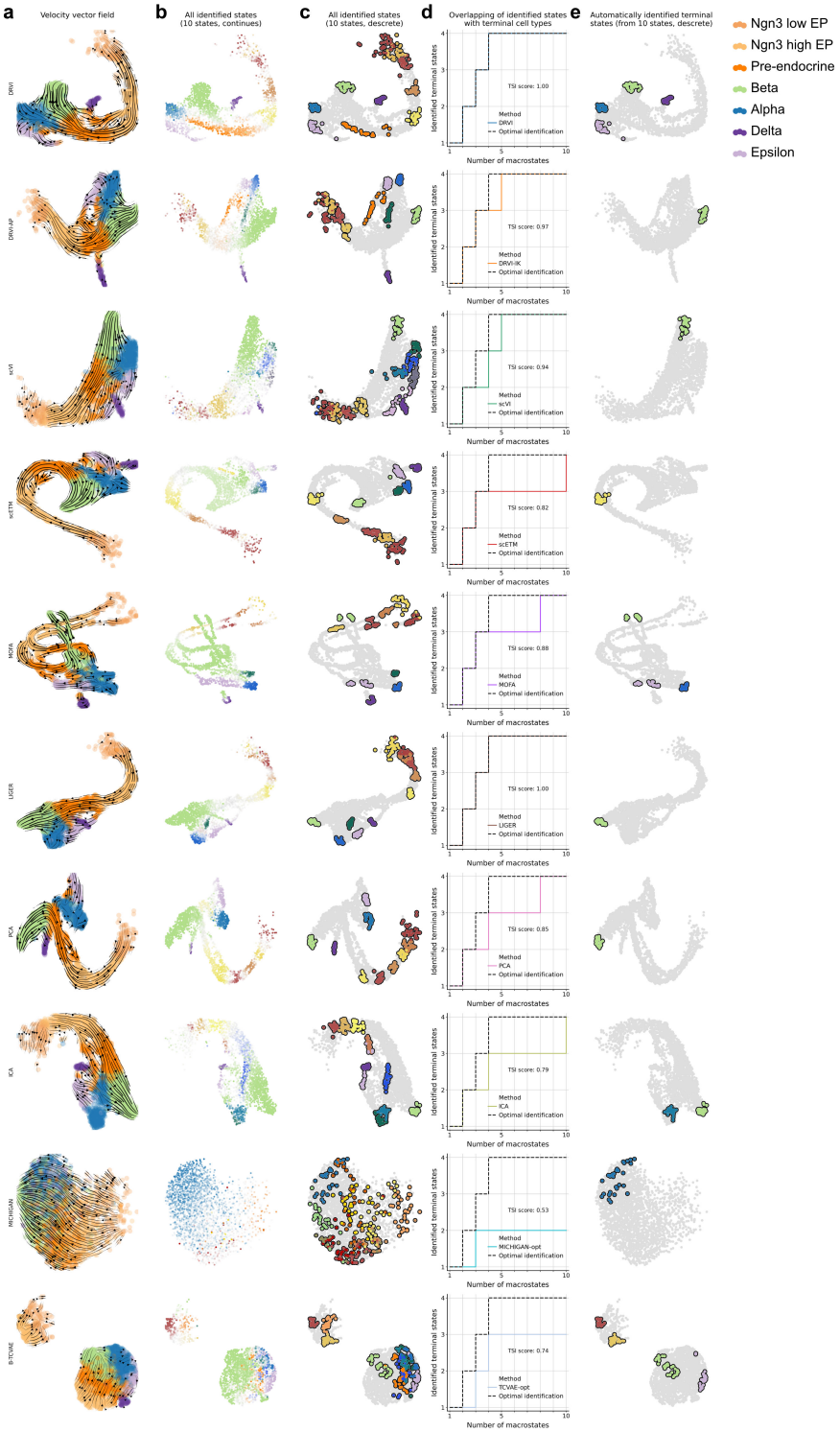


Supplemental Figure 15: **Indication of tumor-adjacent blood vessels in the HLCA. a**, UMAPs of the HLCA integrated by DRVI, colored by second-level annotations, datasets, and tumor adjacency. Tumor-adjacent samples originate from the Krasnow_2020 dataset. By observing the blood vessels in the UMAPs, we find two groups of tumor-adjacent cells. Some integrate seamlessly with normal blood vessels, while others exhibit a degree of separation, suggesting the presence of underlying biological variation. **b**, Activity of DR 22+ on the UMAP. This factor highlights a subset of tumor-adjacent blood vessels that are distinctly separated in the UMAP, potentially representing a unique biological process or state within this cell population. **c**, Histogram of cells distributed across DR 22+, colored by tumor adjacency status before and after filtering for blood vessels. A significant proportion of cells active in the higher end of DR 22+ correspond to blood vessels located in tumor-adjacent regions, indicating a biological process or state specific to this subset of cells.



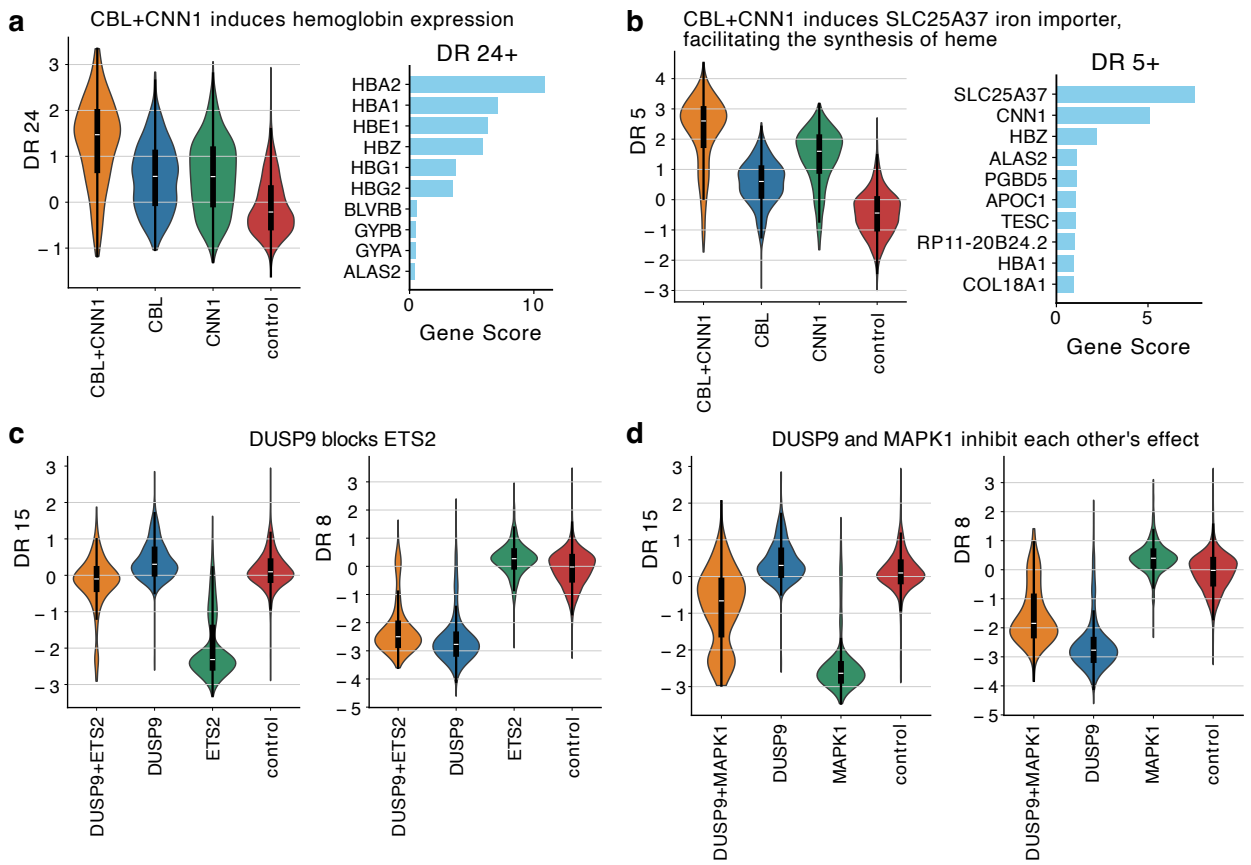
Supplemental Figure 16: **DRVI disentangles cell cycle stages. a,** UMAPs of the developmental pancreas data applied to the latent spaces generated by benchmarked methods. Cells are colored by finest-level annotations. **b-e,** Barplots indicating Mutual information of each latent factor with cell-cycle S-score (b) and G2M-score (d). The factor maximizing the mutual information with the S-score(c) and G2M-score(d) are plotted versus the S-score (c) and G2M-score (e) for each method. Cells are colored by coarse annotations.**f,** The learned cell-cycle manifolds. The scatterplots of factors maximizing the mutual information with respect to the S-score and G2M-score are plotted. Cells are colored by the dominant cell cycle phase. MI, mutual information.

2066



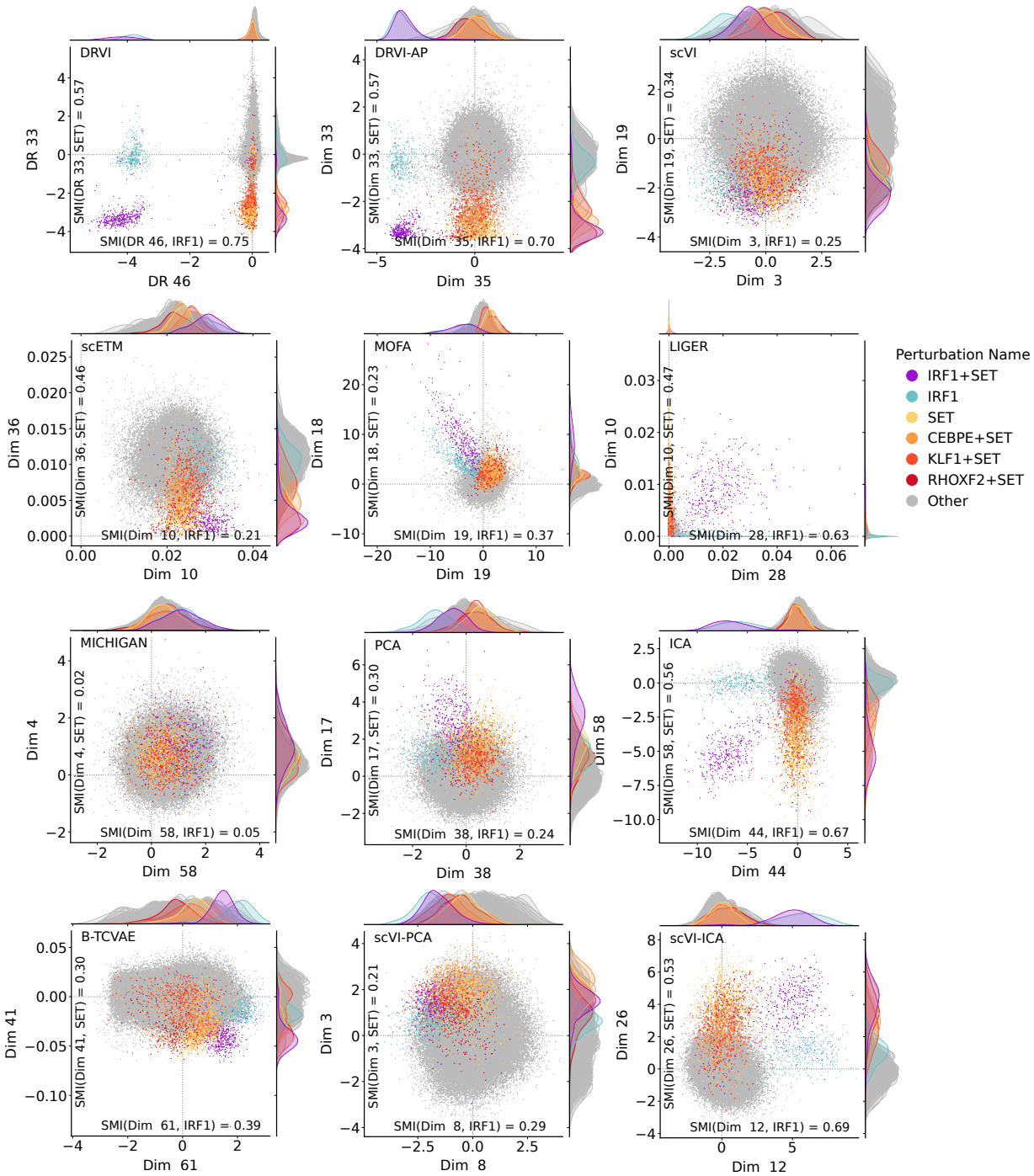
Supplemental Figure 17: **Developmental fate prediction based on model embeddings. a,** UMAP visualizations of scVelo RNA velocity vector fields, generated based on embeddings derived from various benchmarked models. **b,c,** UMAP visualizations of stationary states identified by CellRank, based on the velocity vector fields shown in (a). **d,** Terminal State Identification (TSI) scores for different benchmarked methods. This metric quantifies the overlap between identified stationary states and ground-truth terminal states as the number of macrostates increases. **e,** UMAP visualization of automatically identified terminal states from ten macrostates using CellRank. TSI: Terminal State Identification.

2067



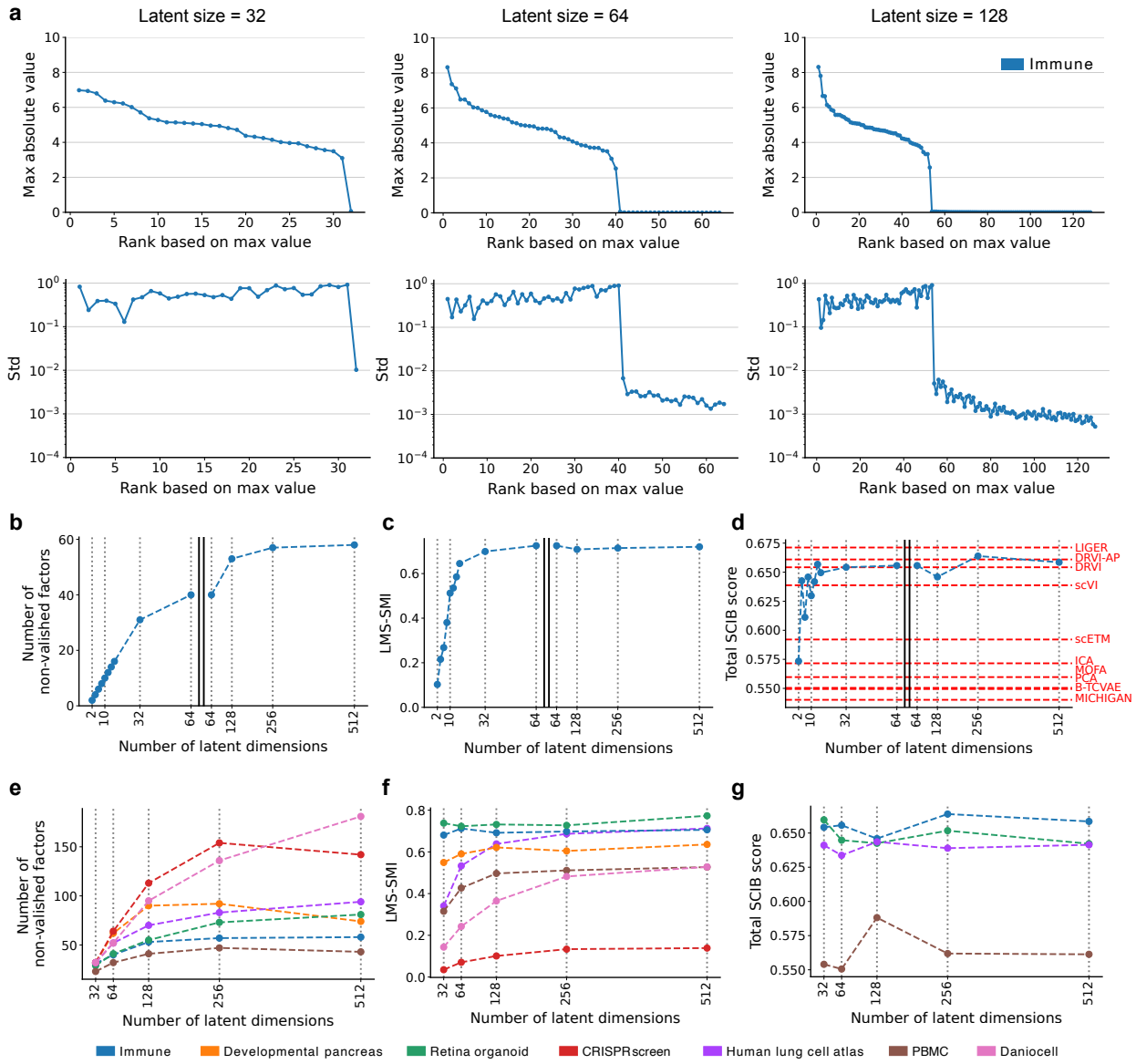
Supplemental Figure 18: **DRVI's latent space effectively models interacting double perturbations. a,** When the **CBL+CNN1** gene combination is perturbed, there's a significant increase in the expression of **hemoglobin** (DR 24+). **b,** Perturbing **CBL+CNN1** leads to the overexpression of the iron importer **SLC25A37** (DR 5+), which supports the role of CBL+CNN1 in regulating hemoglobin expression. **c, DUSP9** acts as an inhibitor of **ETS2**, as its effect (DR 15-) is neutralized when DUSP9 is also perturbed. **d, DUSP9** and **MAPK1** exhibit a mutual inhibitory relationship, which is evident from the decreased response signatures of both genes (DR 15- and DR 8-) when they are simultaneously perturbed.

2068



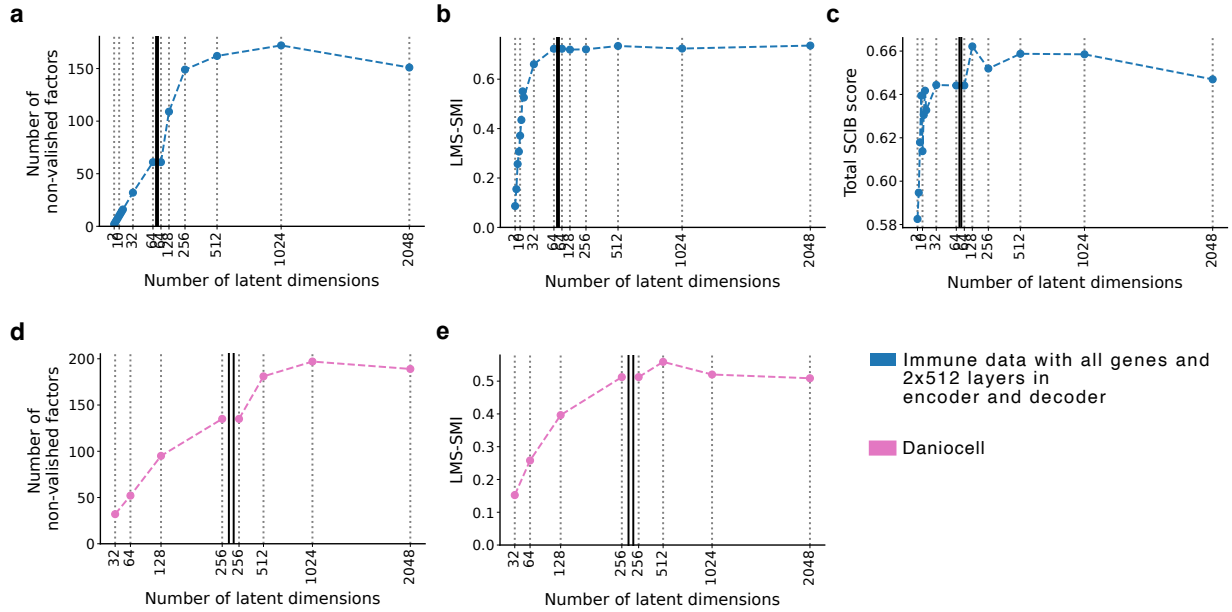
Supplemental Figure 19: **DRVI identifies and preserves the geometry of combinatorial perturbations.** Each scatterplot illustrates the activity of the top two factors having the highest mutual information with respect to IRF1 and SET perturbations for a benchmarked method. The cells perturbed by IRF1 and SET and all combinatorial perturbations containing them are highlighted with distinct colors. Any other perturbation signature is colored grey. SMI, scaled mutual information.

2069

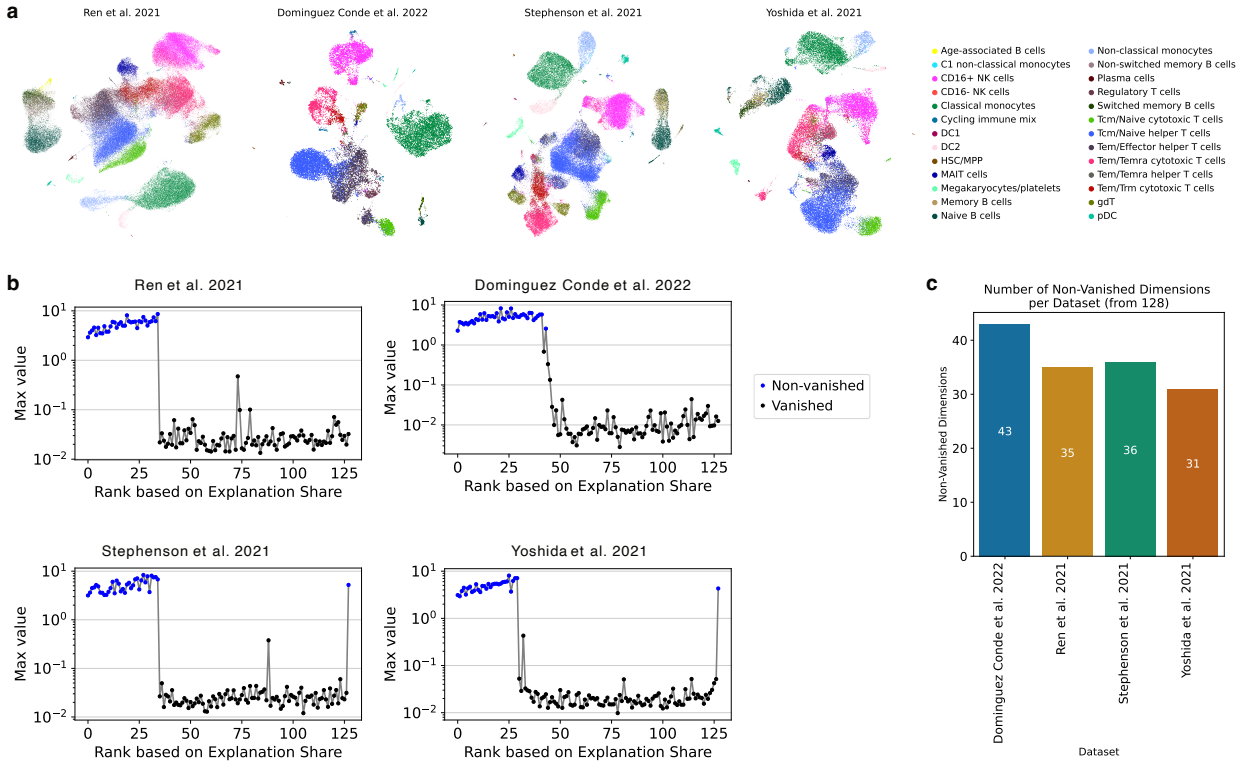


Supplemental Figure 20: **The saturation of the number of employed latent factors by DRVI. a,** The vanishing dimensions phenomenon. DRVI is trained with 32, 64, and 128 latent factors on the immune dataset. Each point is a latent factor. The top row indicates max absolute values, and the bottom row indicates log-scale standard deviations of latent factors. latent factors are ranked based on max absolute value. Each column indicates a run with a different number of latent dimensions. **b-d,** Saturation of the number of non-vanished factors (b), disentanglement score (c), and integration score (d) when more latent factors are provided to DRVI trained on the immune dataset. Red lines annotate integration scores for benchmarked methods with default parameters. The x-axes on the right-hand side of the plots are shrunk by a factor of 5. **e-g,** Saturation of the number of non-vanished factors (e), disentanglement score (f), and integration score (f) when more latent factors are provided to DRVI trained on the benchmarking datasets. LMS, latent matching score; SMI, scaled mutual information.

2070

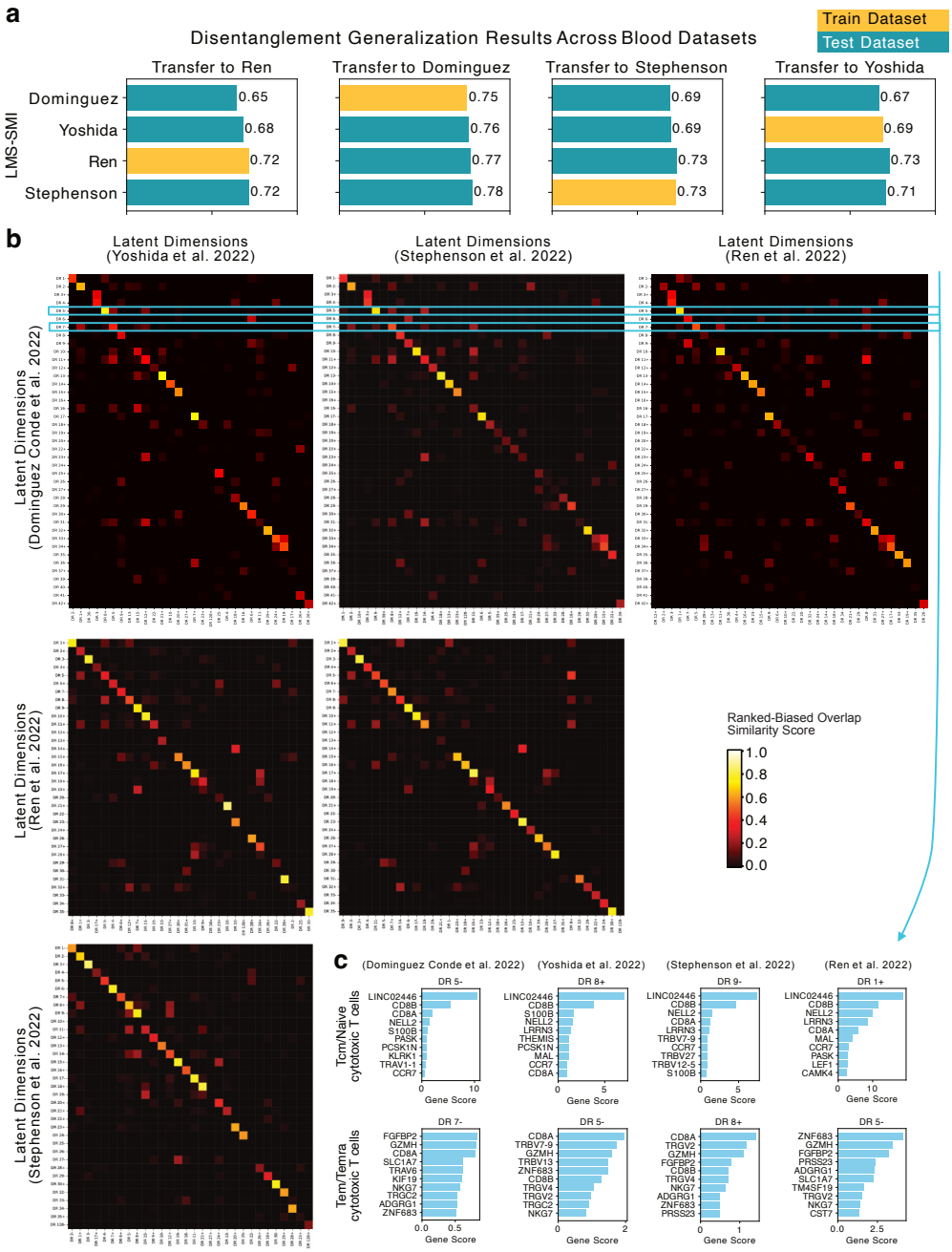


Supplemental Figure 21: **Further analysis on the saturation of employed latent factor. a-c,** DVRI's behavior trained on all genes of the immune dataset with different number of latent dimensions. The number of non-vanished dimensions (a), Disentanglement quality (b), and integration performance (c) are plotted when we change the number of latent dimensions. DRVI is trained on the full gene set of the immune dataset. Encoder and decoders contain 2 layers of 512 hidden neurons. As we observe, DRVI saturates at some specific point. At this point, disentanglement performance also saturates. Integration performance saturates and drops a bit. **d,e,** DVRI's behavior trained on Daniocell data beyond 512 latent dimensions. The number of non-vanished dimensions (d) and disentanglement quality (e) are plotted when we change the number of latent dimensions. As we observe, even in Daniocell data where there are 156 cell types, DRVI saturates at some point. At this point, disentanglement performance also saturates. LMS, latent matching score; SMI, scaled mutual information.

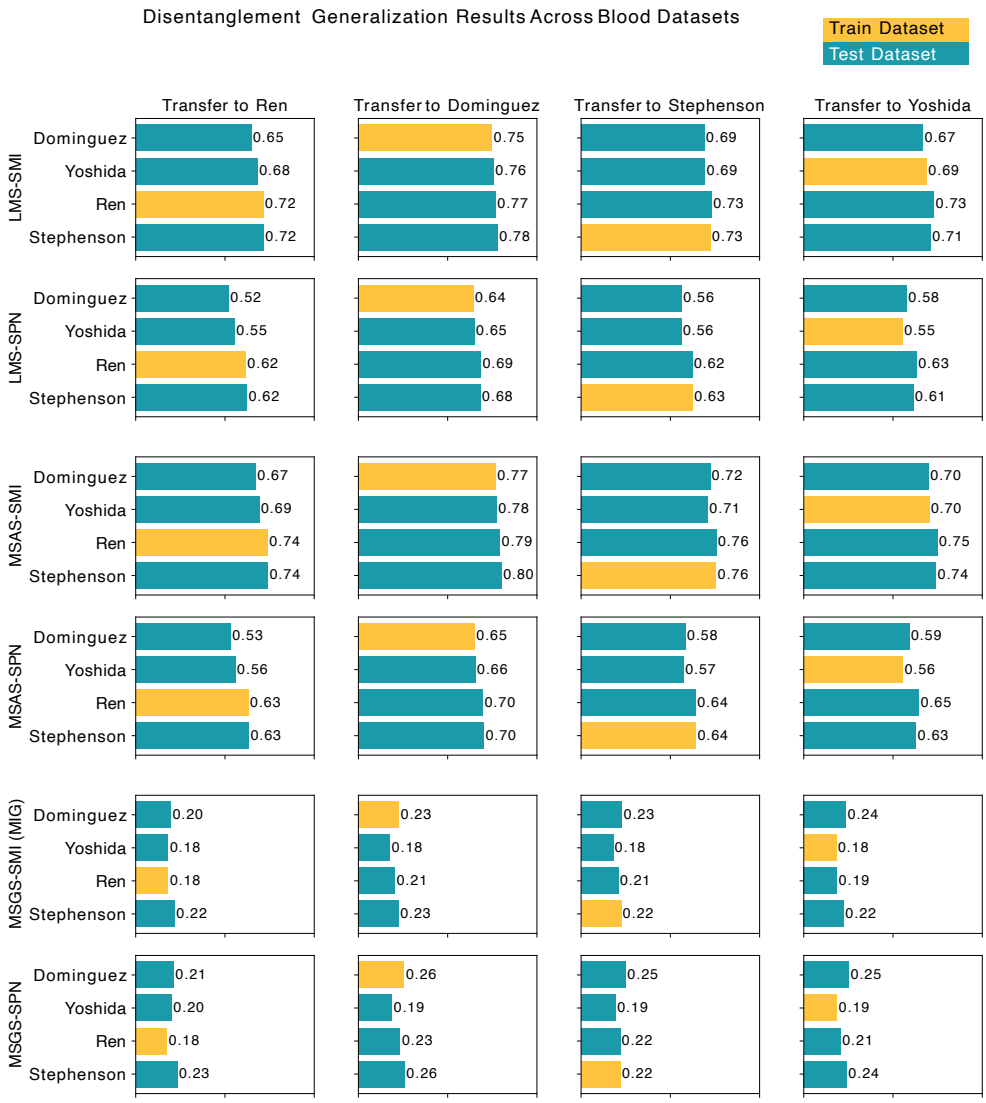


Supplemental Figure 22: **Training DRVI on four datasets of the Blood data. a,** UMAP visualizations of individual Blood datasets, integrated by separate DRVI models. Cells are colored according to their curated annotations. **b,** Visualization of the vanished factors for each of the four DRVI models trained on the Blood datasets. **c,** The total number of non-vanished factors for each model trained on an individual Blood dataset.

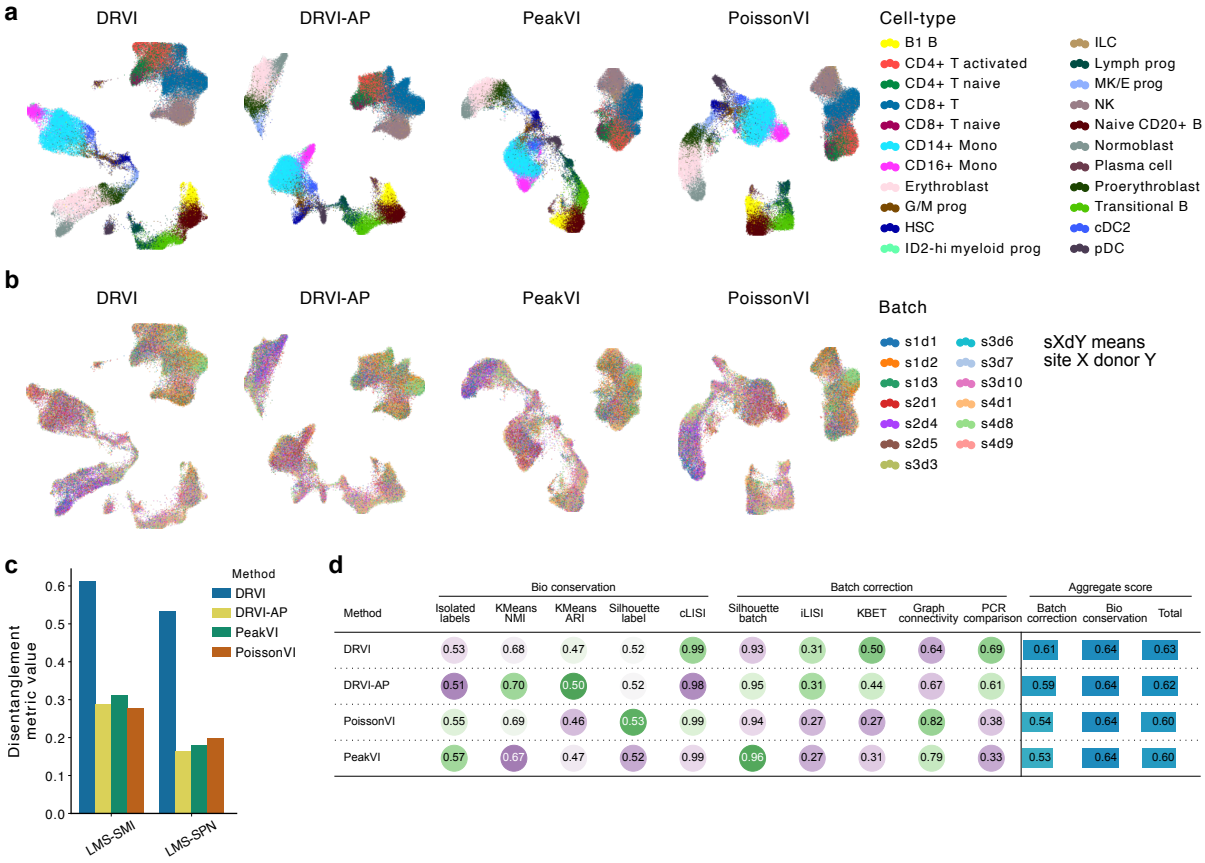
2072



Supplemental Figure 23: **DRVI learned factors are consistent and generalizable to unseen datasets.** Here we assess the generalization and disentanglement of latent programs in unseen Blood data. **a,** A separate DRVI model is trained on each of the Blood datasets. This trained model is then used to generate latent representations for the remaining, unseen datasets. The quality of disentanglement is measured for each of these transfer scenarios. The results demonstrate that the disentangled concepts are robustly transferred to the unseen test datasets. **b,** For each trained DRVI model, we identify the relevant genes associated with each latent factor. We then compare these gene sets across datasets using the Rank-Biased Overlap (RBO) similarity score [89]. The heatmap indicates a favorable correspondence between identified concepts across datasets. **c,** This panel displays the top relevant genes for factors representing Tcm/Naive cytotoxic T cells and Tem/Temra cytotoxic T cells across the four models. The positive RBO scores observed between the factors representing these two cell types (as shown in panel **b**) are a result of the intrinsic gene overlap between these closely related subtypes. LMS, latent matching score; SMI, scaled mutual information.

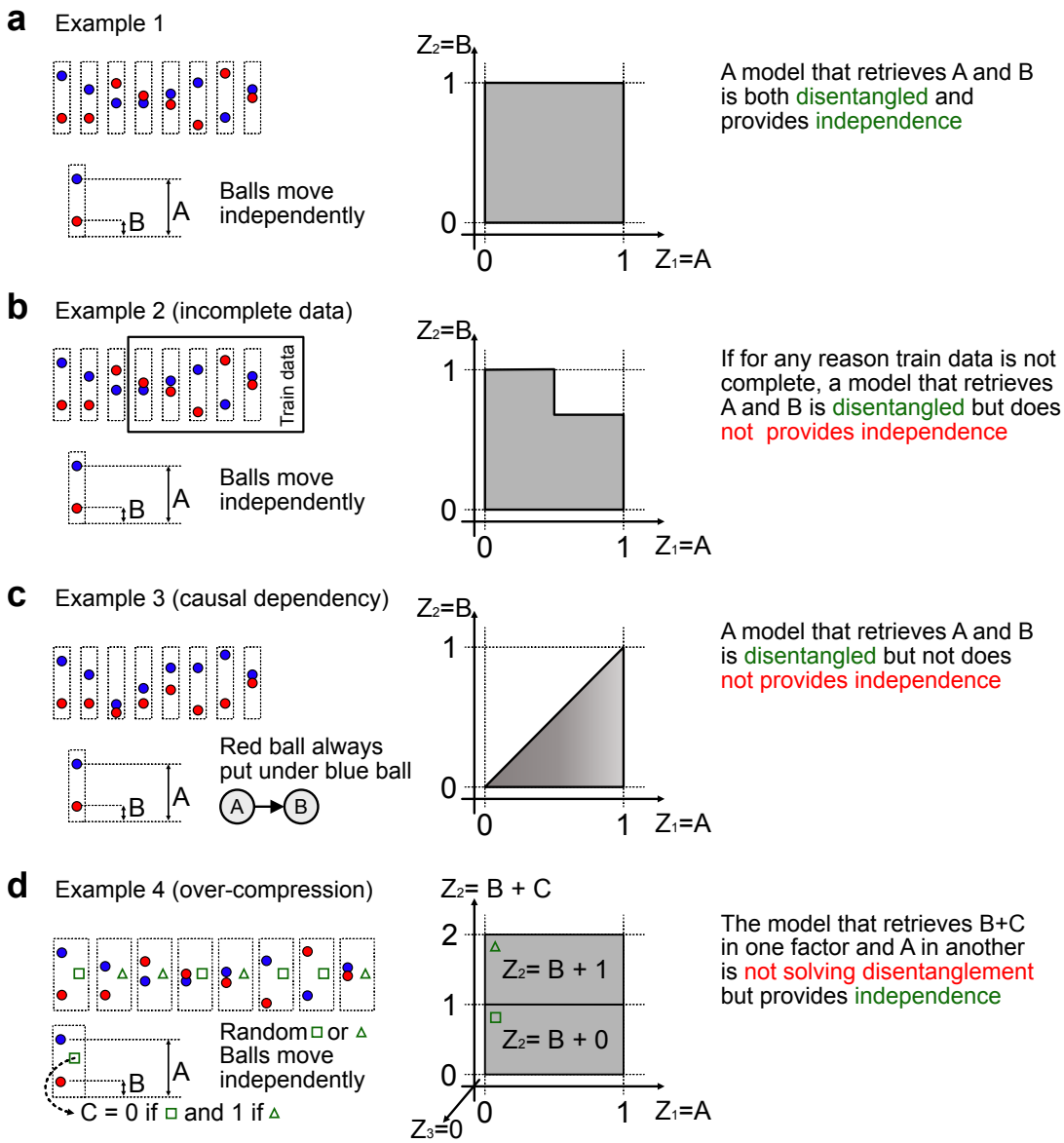


Supplemental Figure 24: **Full disentanglement metrics on generalization to unseen datasets in the Blood data.** A separate DRVI model is trained on each of the Blood datasets. This trained model is then used to generate latent representations for the remaining, unseen datasets. The quality of disentanglement is measured for each of these transfer scenarios. The results demonstrate that the disentangled concepts are robustly transferred to the unseen test datasets. LMS, latent matching score; MSAS, Most Similar Averaging Score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; SPN, same-process neighbors.



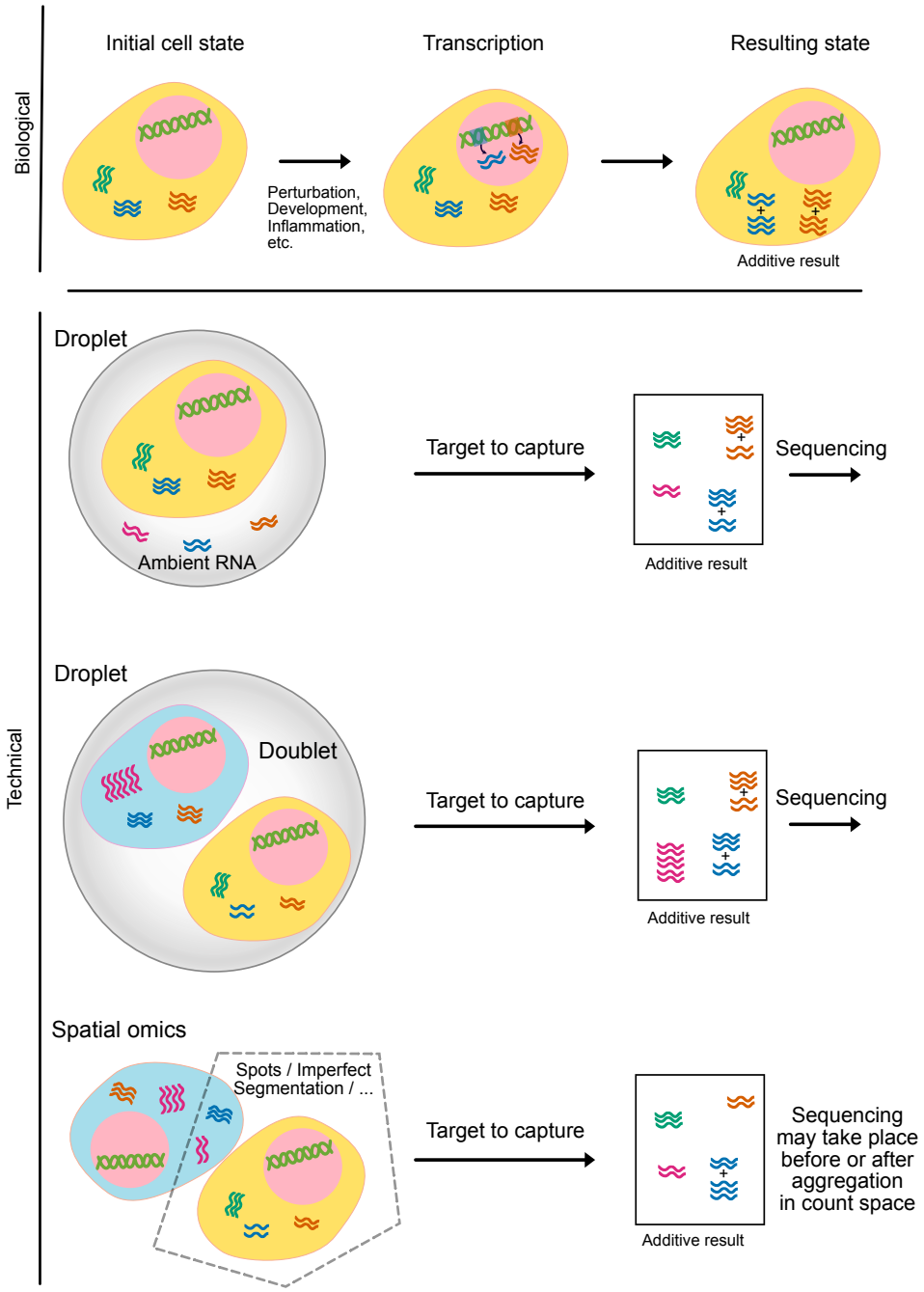
Supplemental Figure 25: **DRVI is applicable to scATAC-seq data. a,b,** UMAPs of the ATAC modality of the NeurIPS 2021 data integrated by DRVI, DRVI-AP, PeakVI, and PoissonVI colored by cell-type annotations (a) and batch identity (b). **c,** Disentanglement performance of the benchmarked methods for scATAC-seq data. DRVI performs significantly better in this task. **d,** Integration performance of the benchmarked methods based on the scIB metrics. DRVI performs better compared to the three other methods. LMS, latent matching score; SMI, scaled mutual information; SPN, same-process neighbors.

2075



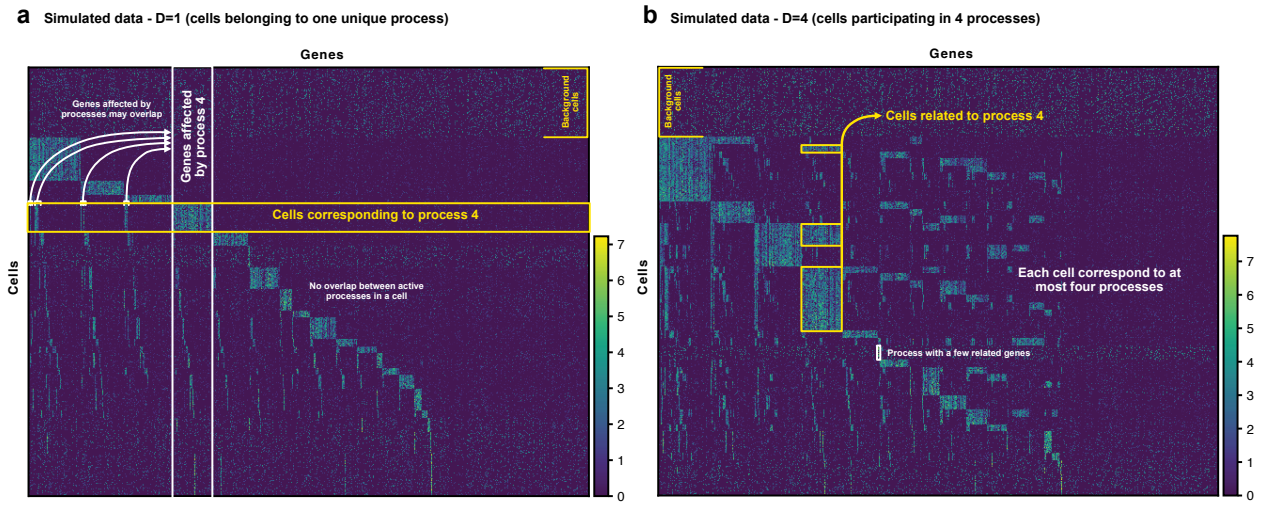
Supplemental Figure 26: **Disentanglement versus Statistical Independence. a,** Alignment of Independence and Disentanglement. In the ideal scenario with independent underlying factors (e.g., A and B) and complete data coverage (square support), a disentangled model recovers factors that are also statistically independent. **b,** Disentanglement without Independence (Incomplete Data): When the training data is incomplete or biased (L-shaped support), correctly disentangled factors A and B will exhibit statistical dependence reflecting the observed distribution, despite being mechanically independent. **c,** Causal Dependency: A causal constraint, such as the red ball (A) always being positioned lower than the blue ball (B), restricts the data manifold (triangular support). A perfect disentanglement model recovers the distinct concepts A and B but yields correlated latent dimensions. **d,** Independence without Disentanglement: An over-compressed model that mixes factors (e.g., learning A and $B+C$) can achieve statistical independence in the latent space (rectangular support) but fails to disentangle the true underlying generative processes.

2076

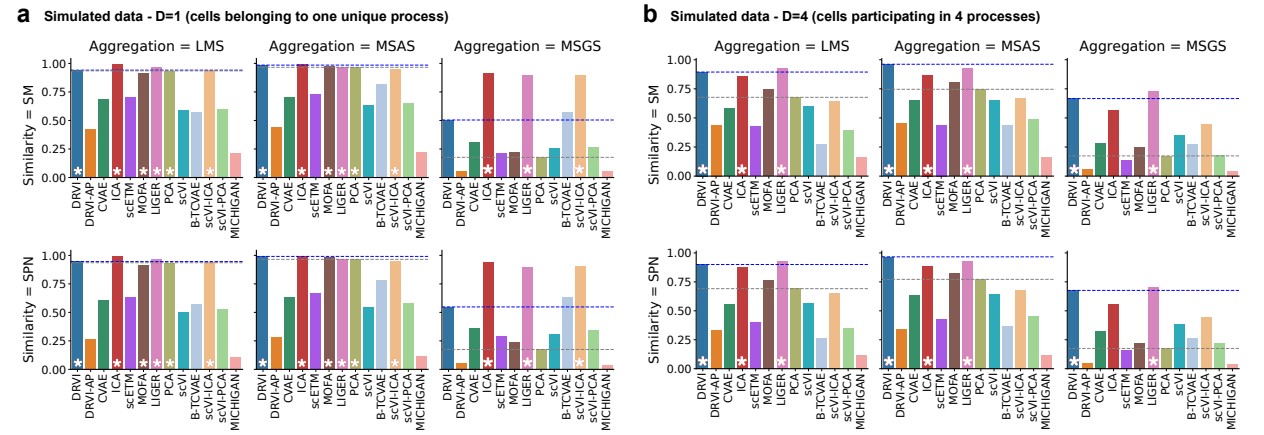


Supplemental Figure 27: **Plausibility of DRVI's additive assumption in scRNA-seq data.** The additivity of independent biological processes makes sense from both biological and technical perspectives. Newly transcribed genes (for example, during developmental bifurcations, induced perturbations, stress, etc.) add up within the cell and therefore add up in count space when sequenced. In addition, technical effects such as doublet cells, ambient background counts, and residual counts from neighboring cells due to imperfect segmentation in spatial omics show additive behaviors.

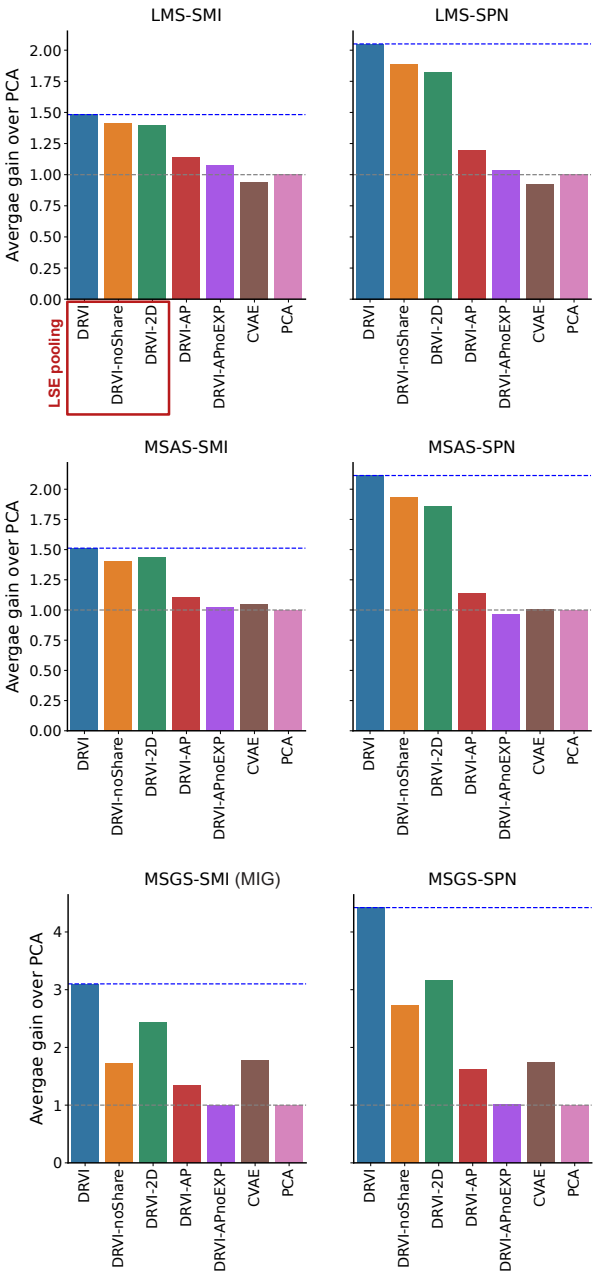
2077



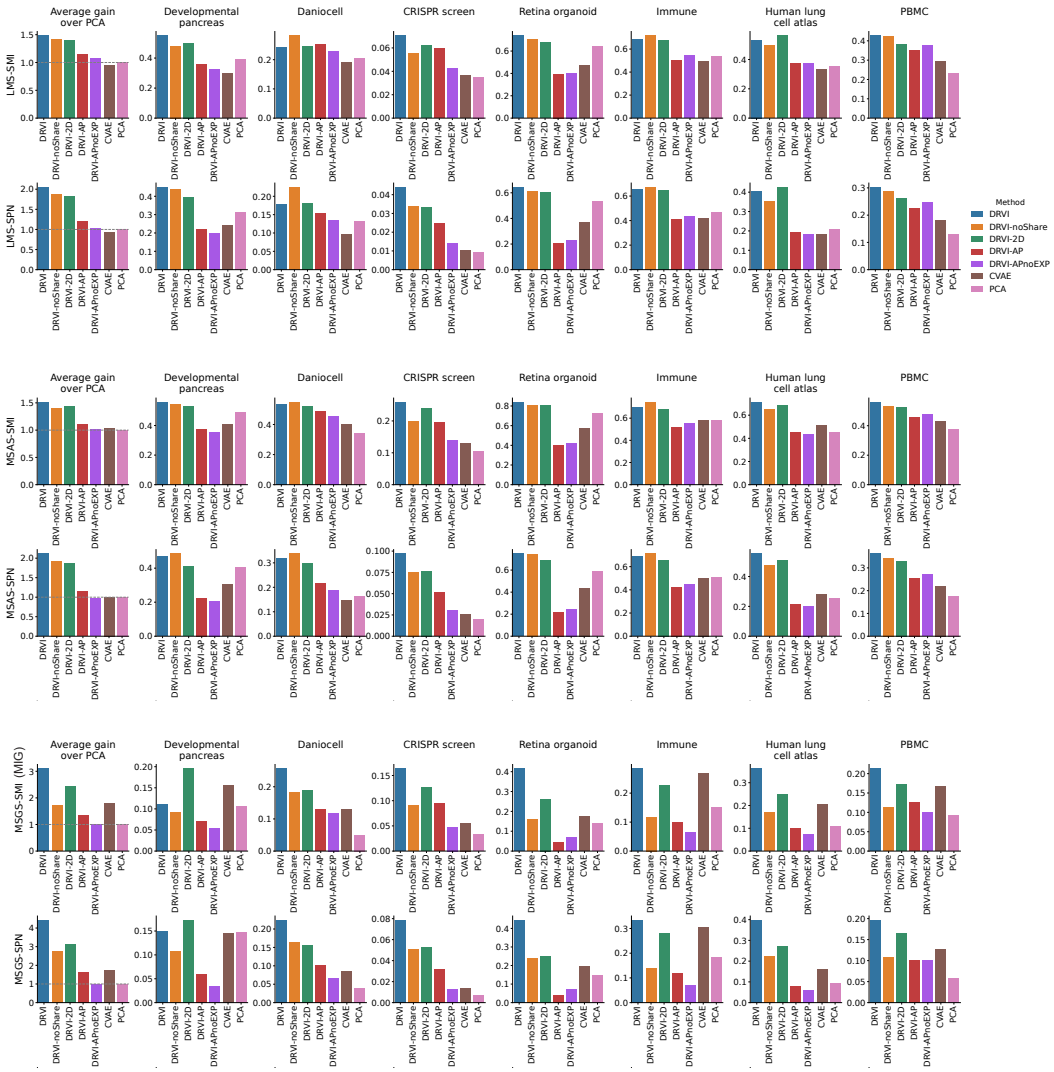
Supplemental Figure 28: **Heatmap of simulated datasets with known ground truth.** The first simulated dataset (**a**) assigns non-background cells to a single process, while genes can be influenced by multiple processes. The second simulated dataset (**b**) allows cells to participate in up to four processes, introducing controlled overlap and compositional complexity. Cells with similar ground truth are grouped, as are genes affected by the same programs. Both cell and gene groups are then arranged to maximize their diagonal presentation.



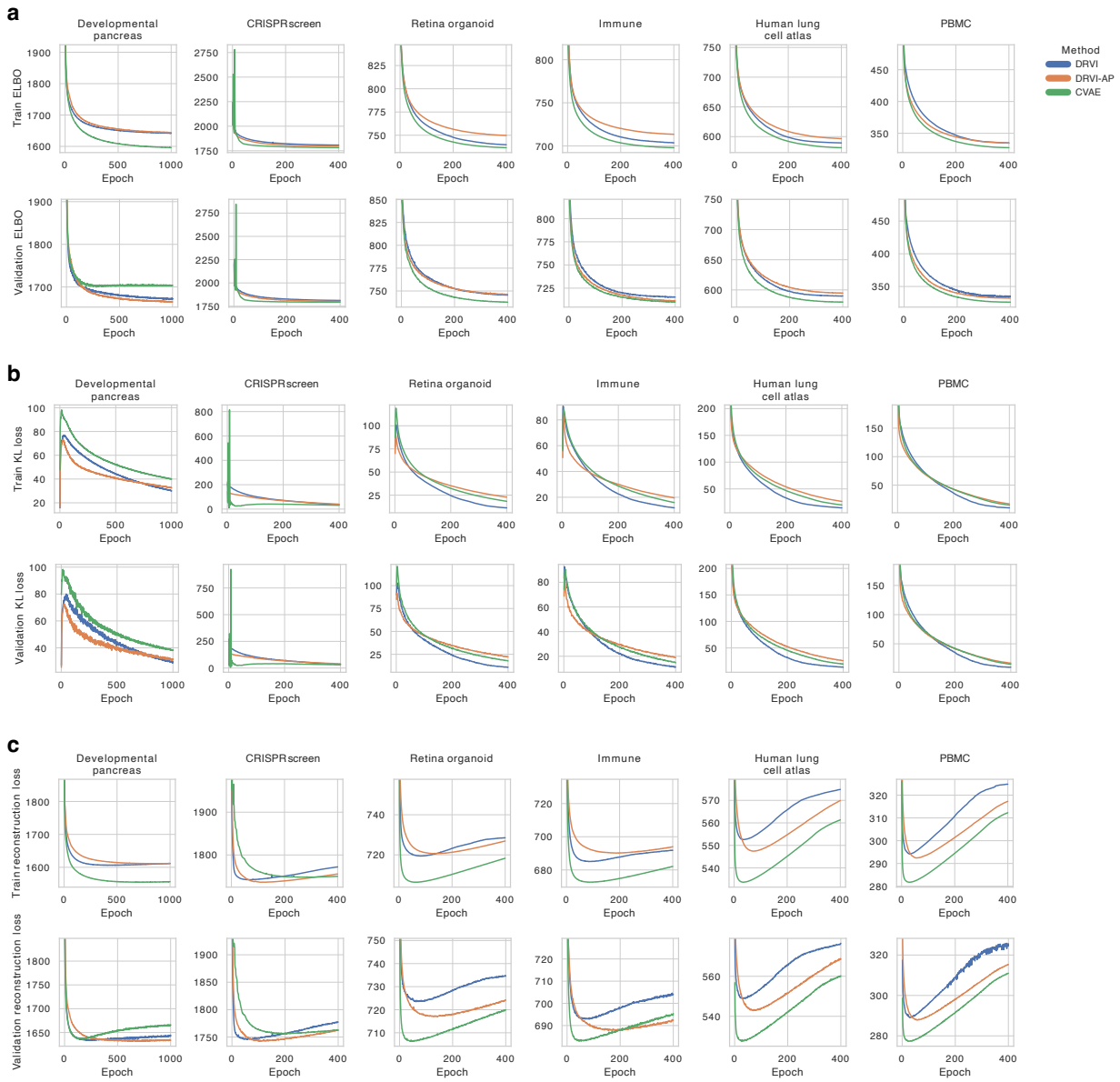
Supplemental Figure 29: **Evaluation results for simulated datasets with known ground truth. a,b,** We calculated metric values for each benchmarked model across the simulated datasets. These metrics are subsequently presented for all possible combinations formed by the SMI and SPN similarity functions and the LMS, MSAS, and MSGS aggregation functions. In cases where cellular activities are mutually exclusive (a), DRVI, ICA, MOFA, PCA, LIGER, and scVI-ICA all perform effectively. However, when processes overlap (b), only DRVI and LIGER are capable of accurately identifying ground-truth processes. LMS, latent matching score; MSAS, Most Similar Averaging Score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; SPN, same-process neighbors.



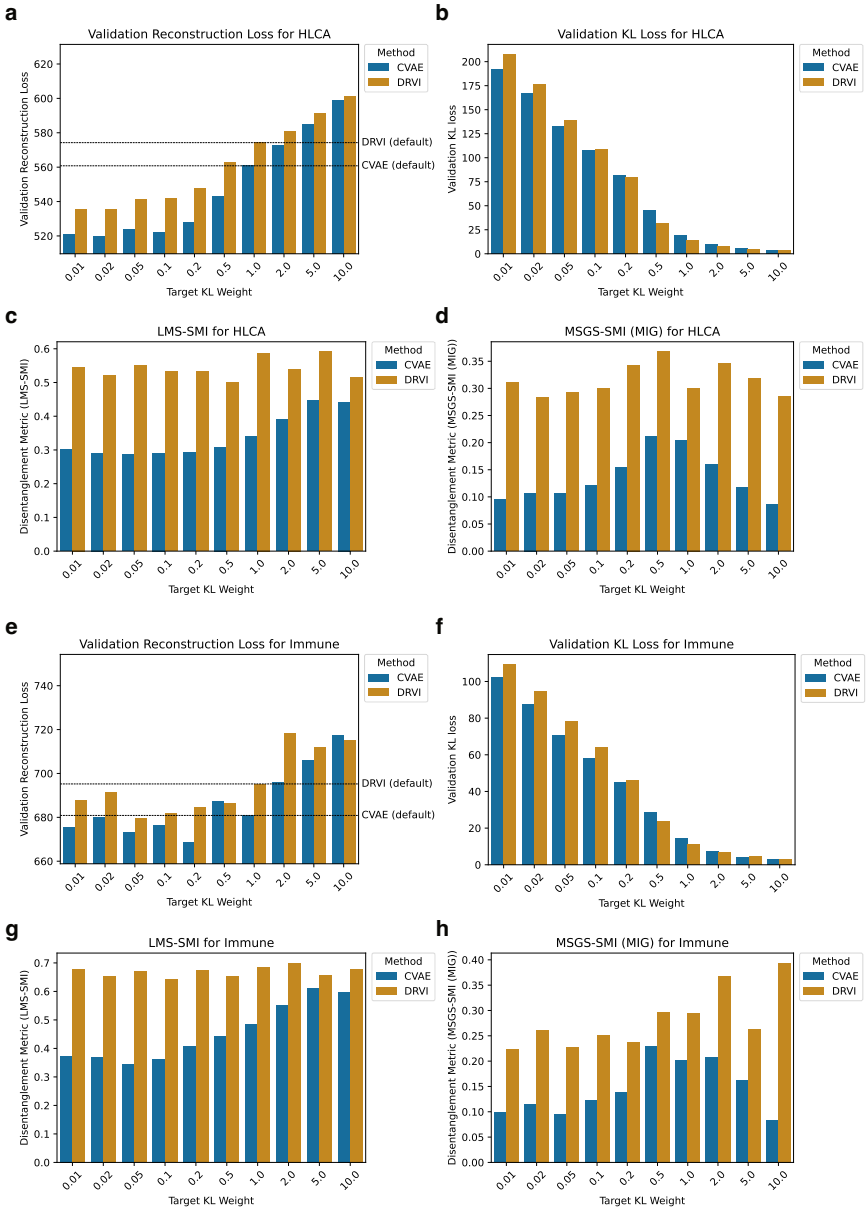
Supplemental Figure 30: **Summary of all metrics for the ablation study.** This plot displays the average gain achieved by each of the DRVI variants over PCA. The gains are shown for every combination of the SMI and SPN similarity functions with the LMS, MSAS, and MSGS aggregation functions. Results indicate the importance of employing log-sum-exp (LSE) pooling to achieve disentanglement. LMS, latent matching score; MSAS, Most Similar Averaging Score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; SPN, same-process neighbors; MIG, mutual information gap.



Supplemental Figure 31: **Details of disentanglement metrics for the ablation study.** For all benchmarked datasets and each of the DRVI variants, detailed results of the LMS, MSAS, and MSGS aggregation functions over the SMI and SPN, similarity functions are provided. LMS, latent matching score; MSAS, Most Similar Averaging Score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; SPN, same-process neighbors; MIG, mutual information gap.

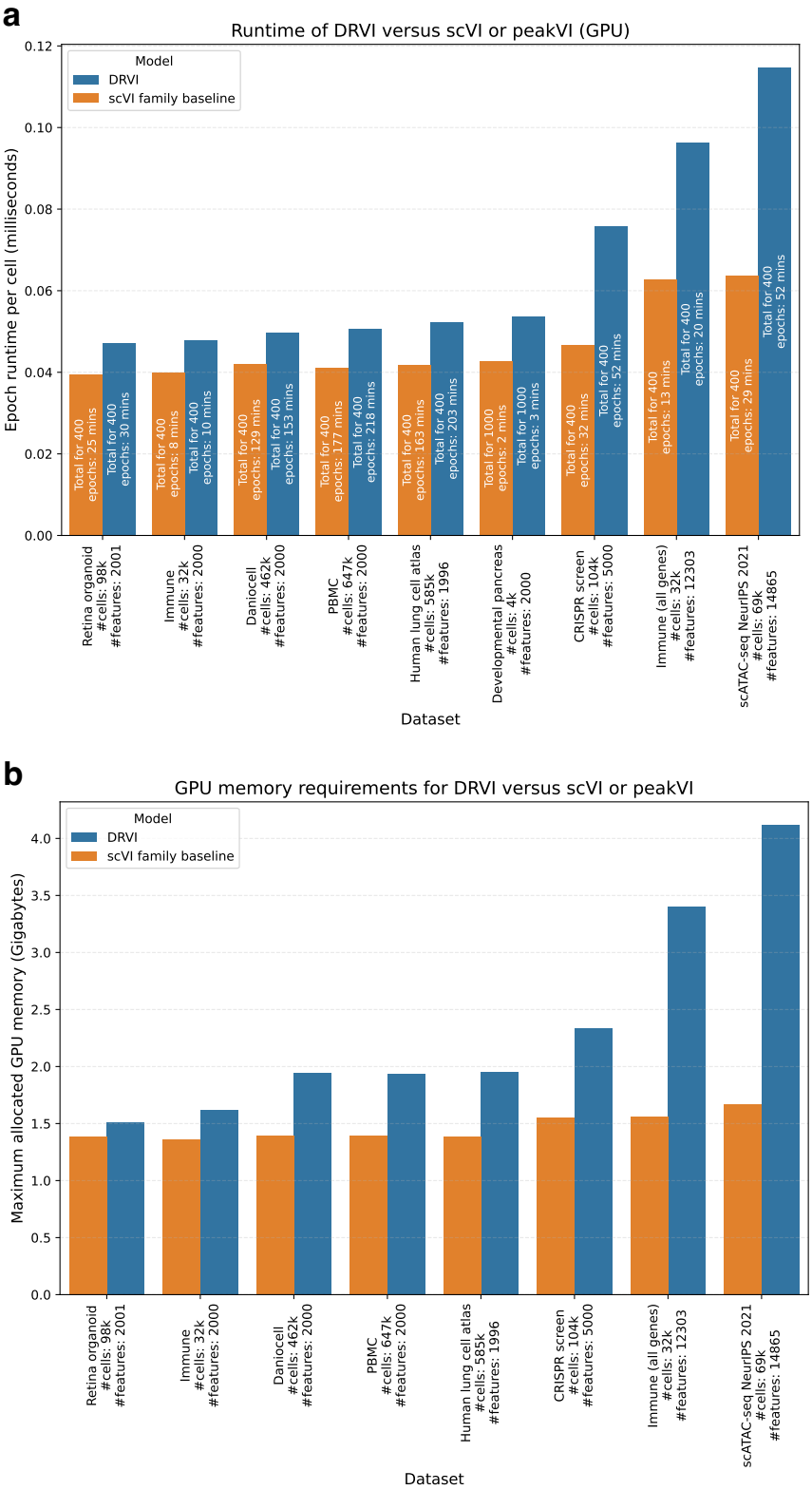


Supplemental Figure 32: **The optimization of training objectives. a-c** During training on various datasets, we monitored the Evidence Lower Bound (ELBO) (a), latent KL loss (b), and negative binomial reconstruction loss (c). We compared DRVI, DRVI-AP, and a CVAE trained specifically for comparison (without latent space splitting), all trained with the same objectives; ELBO as the sum of the KL loss and log-space parametrization of negative binomial for reconstruction. In general, the constrained models (DRVI, DRVI-AP) exhibited worse reconstruction performance but achieved higher KL loss compared to the CVAE on both training and in-distribution validation sets.

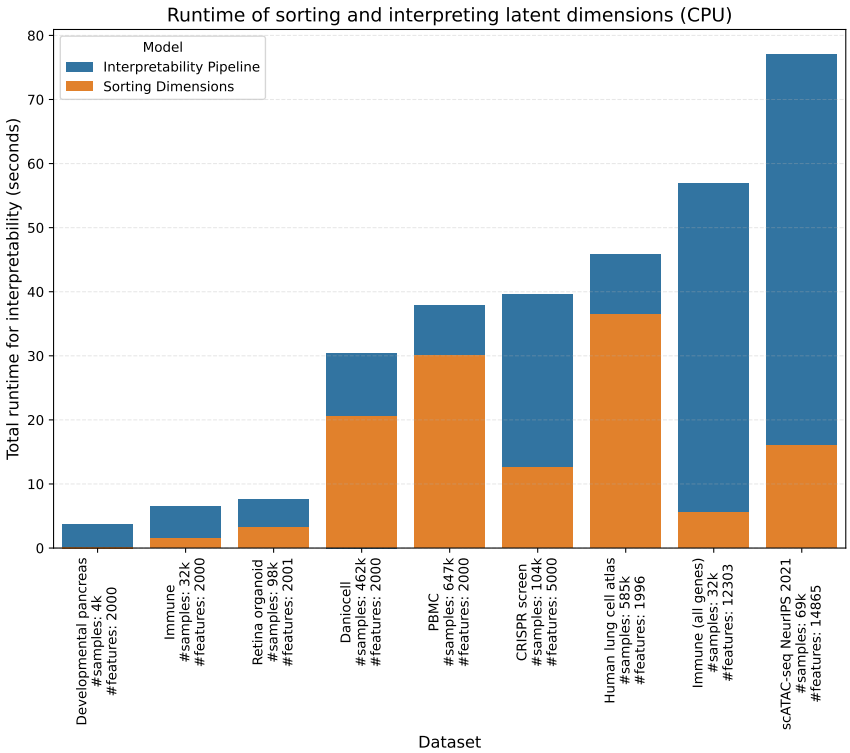


Supplemental Figure 33: **The effect of KL loss weight on reconstruction and disentanglement performance.** To investigate the impact of KL loss on reconstruction and disentanglement, we varied the target-KL weight from 0.01 to 10 (defaulting to 1.0). DRVI models and CVAE baselines (with identical likelihood objectives) were trained on both the HLCA (a-d) and Immune (e-h) datasets. As expected, increasing the target-KL weight significantly increased validation reconstruction loss (a,d) while decreasing the validation KL objective (b,f). The effect of target-KL weight on disentanglement, however, varied by model. For CVAE, this parameter significantly impacted performance: decreasing the target-KL weight results in a decline in disentanglement capability (c,g), and deviating from the default (both higher and lower) decreased independence (d,h). In contrast, DRVI showed robustness; decreasing the target-KL weight led to a very slight decline in identification capability (c,g) and a slight overall decline in independence (d,h). This shows that users requiring good reconstruction performance can safely reduce the target-KL weight without much affecting disentanglement capabilities. LMS, latent matching score; MSGS, Most Similar Gap Score; SMI, scaled mutual information; MIG, mutual information gap.

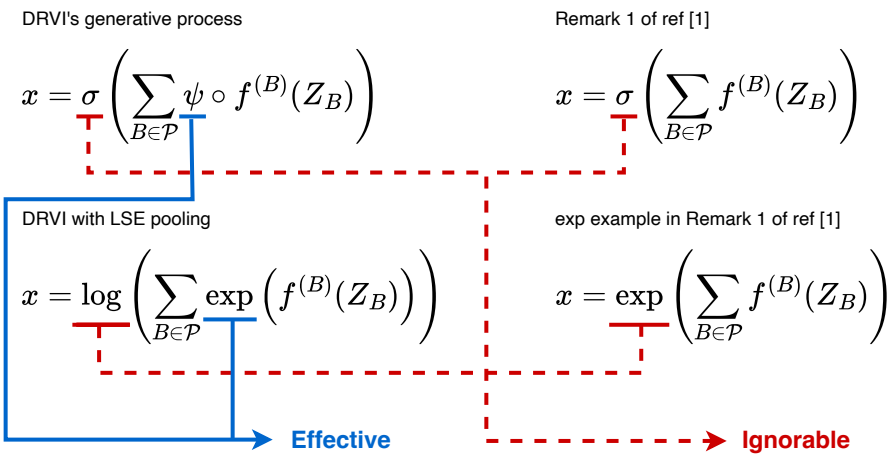
2083



Supplemental Figure 34: **Runtime analysis of DRVI. a,b,** The runtime (a) and maximum allocated GPU memory (b) for DRVI, scVI, and peakVI are shown for each benchmarked dataset. To normalize for varying dataset sizes and training lengths, the y-axis in (a) represents the runtime divided by the number of cells and the number of epochs (runtime per sample per epoch). The total runtime for each experiment is also displayed on the respective bars. DRVI's training runtime is on the same order as scVI or peakVI up to a small scaling factor, with a maximum twofold increase observed in our experimental setups. In addition, training DRVI remains manageable when applied to the full gene set.

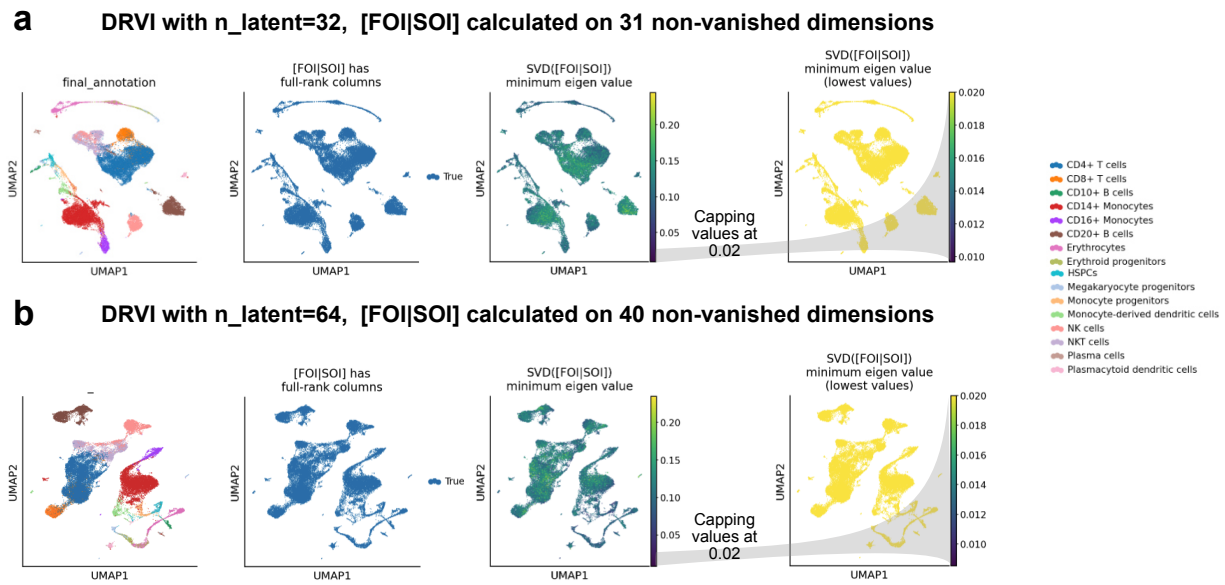


Supplemental Figure 35: **Runtime for interpretability and sorting latent factors.** The time required to sort the latent factors and interpret gene programs for all latent factors is calculated. This demonstrates that the interpretability runtime is negligible.

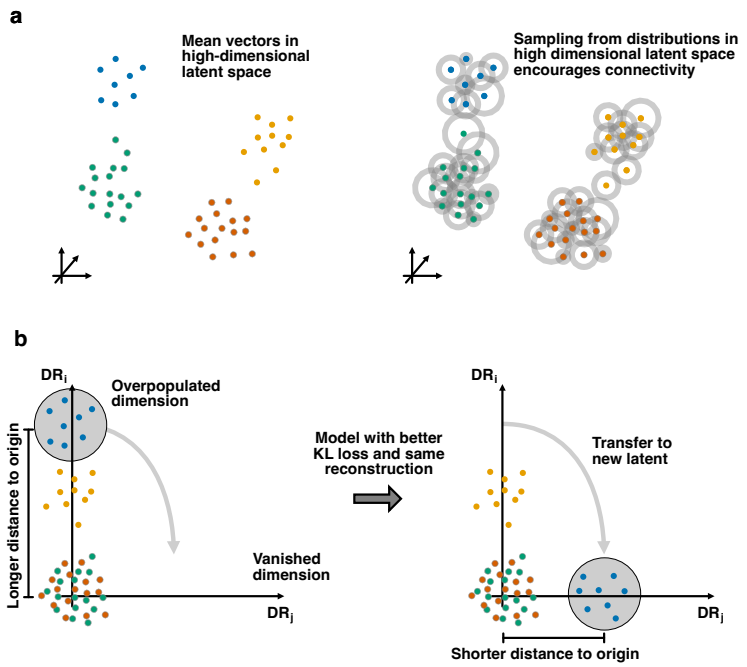


Supplemental Figure 36: **The effective and ignorable parts of the pooling function.** Remark 1 from [37] suggests that activation functions applied after summing additive decoders can be disregarded. In contrast, here, we make use of a shared activation function (ψ) before the sum operator, which is not ignorable. We practically observe that this novel constraint plays a crucial role in enforcing disentanglement.

2086

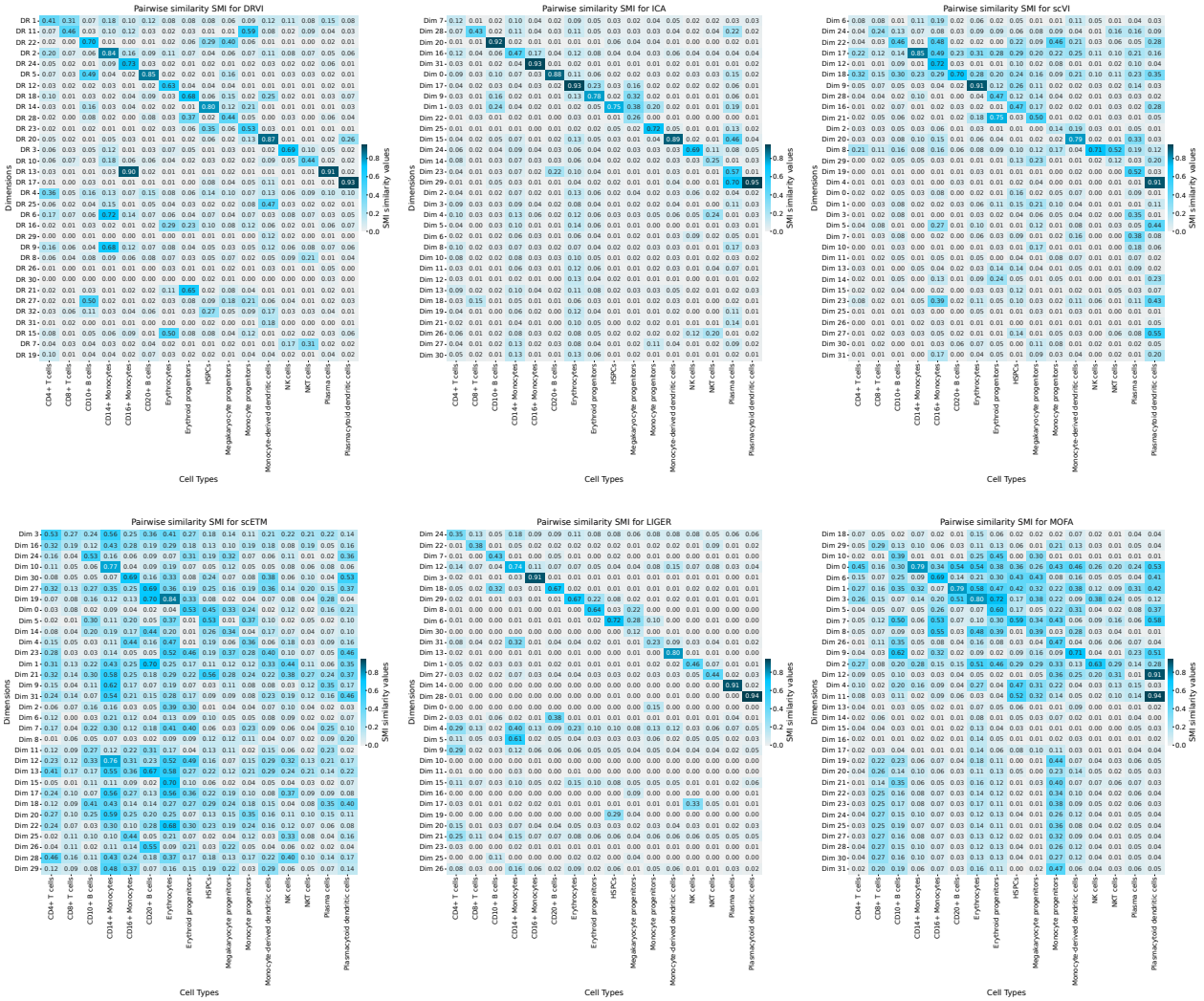


Supplemental Figure 37: **Numerical validation of sufficient independence in Immune data. a,b,** The calculation of the rank of the sufficient independence matrix ($[FOI|SOI]$) on training data for DRVI models with (a) 32 and (b) 64 latent dimensions. The plot shows UMAP colored by cell types. The plot illustrates the status of the sufficient independence matrix being of full rank, the minimum eigenvalue of the SVD of this matrix after column-wise normalization, and the lower end of the minimum eigenvalues, demonstrating that the minimum values in both cases are more than zero (approximately 0.01) which indicates the matrices are numerically full-rank.

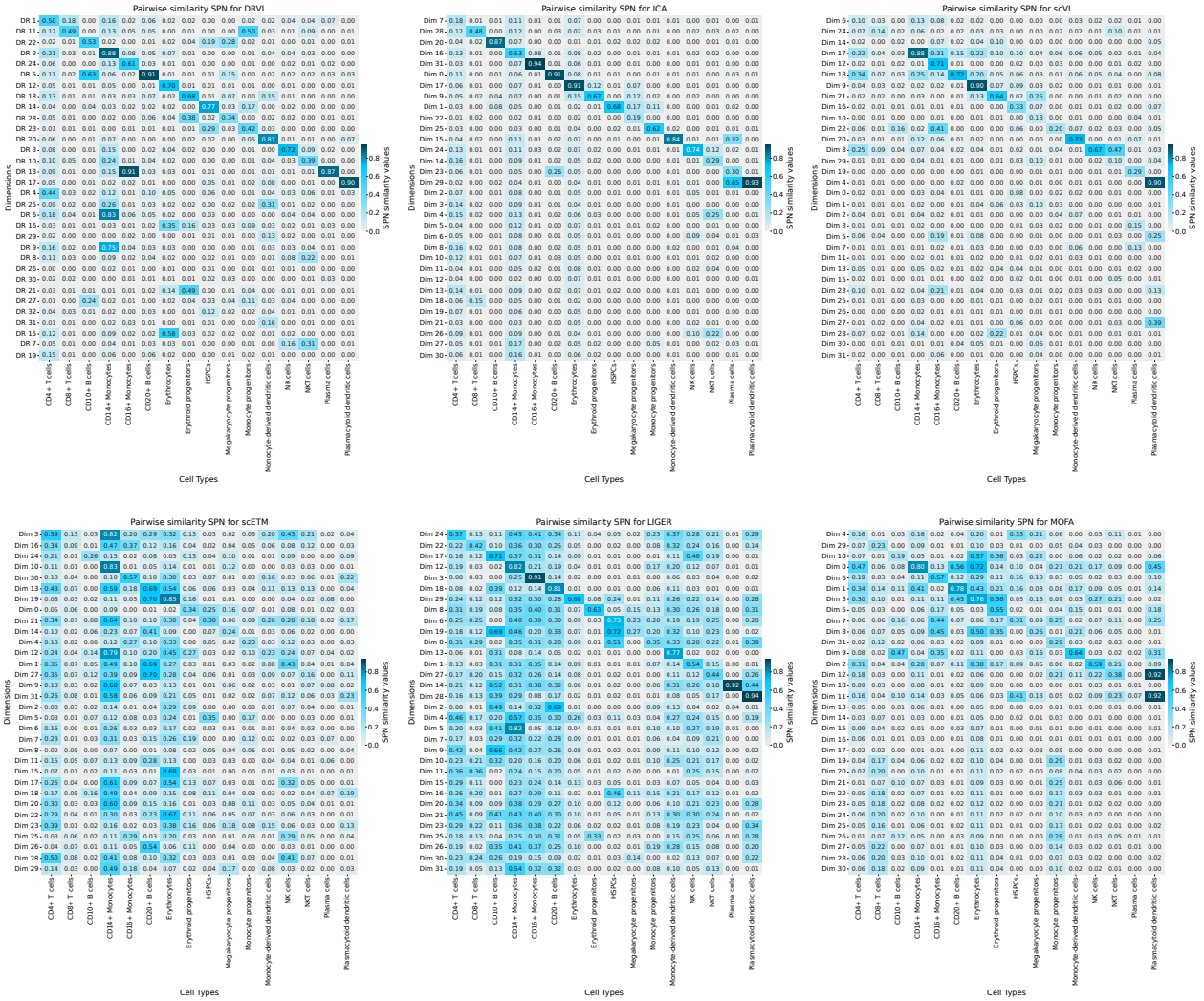


Supplemental Figure 38: **How latent space sampling and KL loss promote global disentanglement. a,** In DRVI, like other VAE-based methods, the latent representation of each cell isn't a vector, but rather a distribution, typically represented by a multivariate Gaussian with mean μ and a diagonal covariance matrix derived from a σ vector. The KL loss term encourages μ values to be near zero and σ values to be near one. As the model converges, these non-negligible σ values cause the distributions of neighboring cells to overlap. This prevents the model from indexing individual cells and promotes a continuous manifold of nearby cells. Thus, the KL loss term encourages path connectivity. **b,** Assuming the model has achieved local disentanglement and a sufficient number of latent dimensions are available, the KL loss term indirectly encourages irrelevant processes to be represented in individual latent factors. If two irrelevant processes are present within a single latent factor and an additional dimension is available, further optimization of the model is possible. By relocating the process further from the center to an available dimension, one can achieve the same reconstruction quality while simultaneously improving the KL loss.

2088

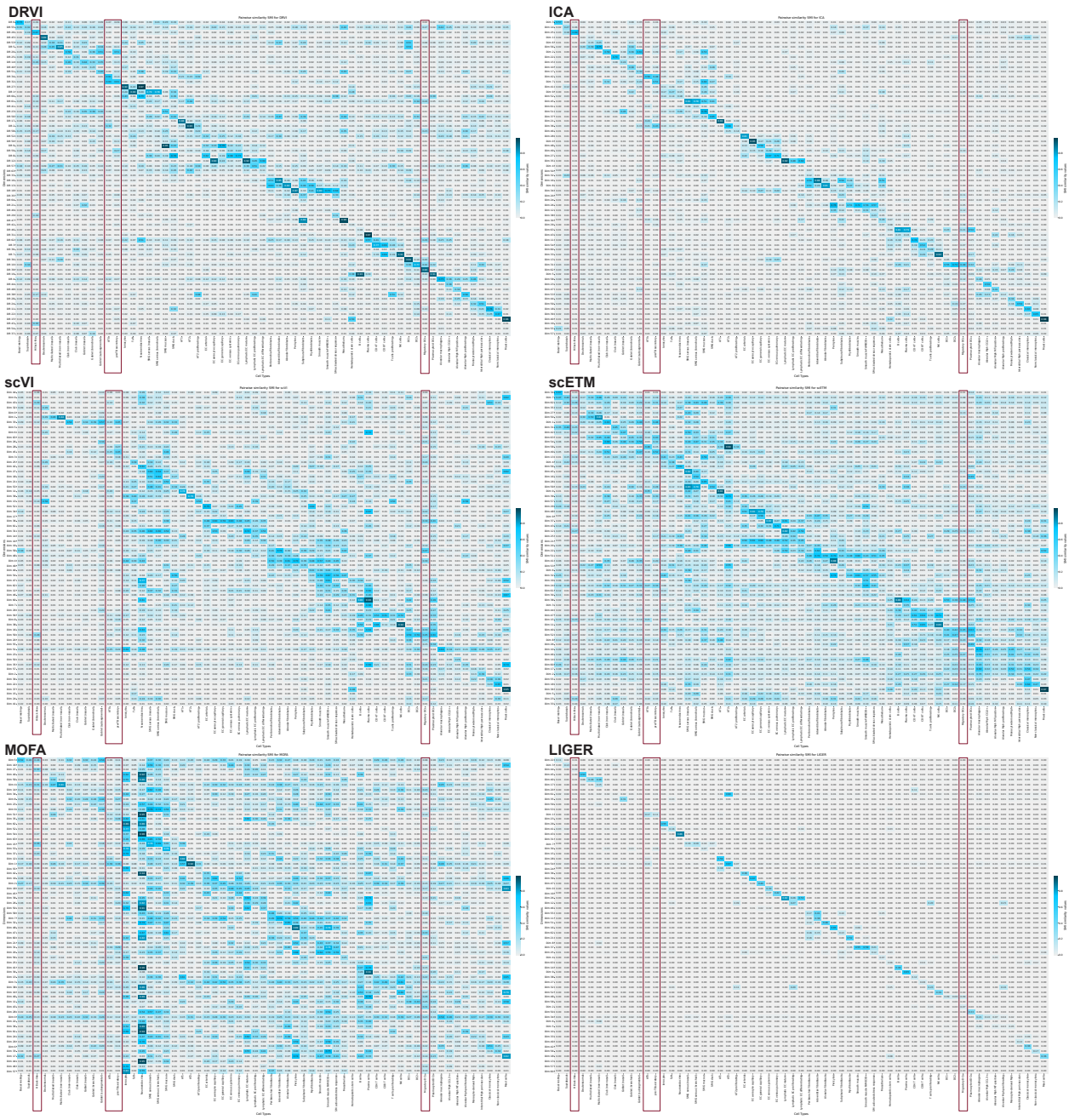


Supplemental Figure 39: **Unaggregated Scaled Mutual Information (SMI) scores for the Immune dataset.** For the top-picked methods, a heatmap visualizing SMI scores between cell types and latent factors is provided. Latent factors are reordered using a shared algorithm to maximize a diagonal appearance.

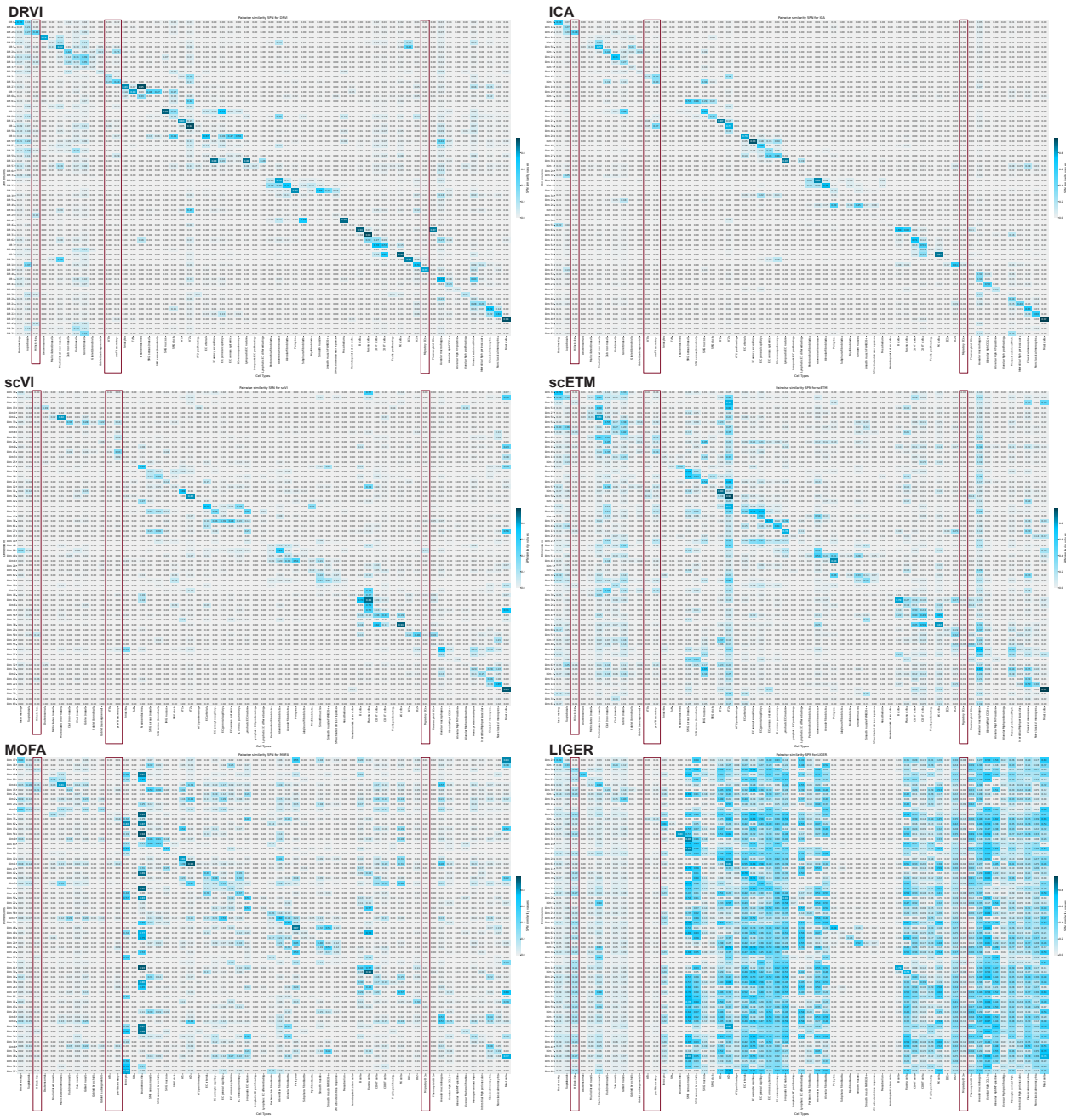


Supplemental Figure 40: **Unaggregated Same-Process Neighbors (SPN) scores for the Immune dataset.** For the top-picked methods, a heatmap visualizing SPN scores between cell types and latent factors is provided. Latent factors are reordered using a shared algorithm to maximize a diagonal appearance.

2090



Supplemental Figure 41: **Unaggregated Scaled Mutual Information (SMI) scores for the HLCA.** For the top-picked methods, a heatmap visualizing SMI scores between cell types and latent factors is provided. Latent factors are reordered using a shared algorithm to maximize a diagonal appearance. Rare cell types mentioned in the paper are highlighted with red rectangles.



Supplemental Figure 42: **Unaggregated Same-Process Neighbors (SPN) scores for the HLCA.** For the top-picked methods, a heatmap visualizing SPN scores between cell types and latent factors is provided. Latent factors are reordered using a shared algorithm to maximize a diagonal appearance. Rare cell types mentioned in the paper are highlighted with red rectangles.