nature computational science

Article

https://doi.org/10.1038/s43588-025-00948-w

Partially shared multi-modal embedding learns holistic representation of cell state

Received: 2 March 2025

Accepted: 16 December 2025

Published online: 25 February 2026

Check for updates

Xinyi Zhang[1,2], G. V. Shivashankar[3,4] ✉ & Caroline Uhler[1,2] ✉

Current technologies enable the simultaneous measurement of diverse data types at the single-cell level. However, data are often processed separately, or integrated via representation learning methods that obscure the contributions of each data modality. Here we present a computational framework that automatically learns partial information sharing between multiple modalities by using an Autoencoder with a Partially Overlapping Latent space learned through Latent Optimization (APOLLO). We tested APOLLO on simulated data, and on four applications involving paired single-cell data: SHARE-seq (scRNA-seq and scATAC-seq), CITE-seq (scRNA-seq and protein abundance), and two multiplexed imaging datasets. APOLLO enables the prediction of missing modalities, such as unmeasured protein stains, and allows disentangling which modality or cellular compartment is linked with a specific phenotype, such as the variability in protein localization observed across single cells. Overall, APOLLO efficiently integrates diverse data modalities and, by retaining and distinguishing between shared and modality-specific information, provides a more interpretable and holistic view of cell state.

Recent advances in experimental techniques have enabled the simultaneous measurement of multiple modalities, including multiplexed imaging and different sequencing modalities at the single-cell level[1–6], and various spatial transcriptomic methods in the tissue context[7–9]. While each data modality offers a different view, the true underlying cell state remains not directly observable. Given that the physical, transcriptional and functional state of a cell are correlated, we expect some information to be shared among the different modalities, while some aspects of cell state are only captured in one of the modalities (Fig. 1a).

Current computational methods for analyzing multi-modal data fall into two main categories: those that process each modality separately and compare results post hoc, or those that learn a combined, integrated representation. The first approach is inefficient and often fails to fully exploit the information in the multi-modal dataset, requiring manual interpretation that typically restricts analysis to a few cell clusters or features[5,10]. For example, in the context of paired single-cell assay for transposase-accessible chromatin using sequencing (scATAC-seq) and single-cell-RNA sequencing (scRNA-seq) data, chromatin accessibility is often summarized at the gene level to allow direct comparison with gene expression[5]. Such coarse-graining of the data into a shared set of features may lose important information in each modality and can only be done when the modalities are similar. To overcome this limitation, various integration methods have been developed. Linear transformations and factor analysis[11–13] are effective for sequencing-based modalities with common features (such as genes) but cannot easily be extended to data such as imaging, and do not explicitly identify shared information. Nonlinear methods such as optimal transport and generative adversarial networks[14–16], and neighborhood-based approaches[17,18], also present limitations as they either do not learn an integrated latent space that account for all modalities, or are restricted to data with natural distance metrics, such as sequencing-based data. More recently, various representation learning methods[19], including autoencoders, have been introduced to single-cell biology[20–22] for the automatic integration of multi-modal data for joint downstream analysis including clustering and biomarker identification[23–30]. In previous work, we developed autoencoder-based

[1]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. [2]Eric and Wendy Schmidt Center, Broad Institute of MIT and Harvard, Cambridge, MA, USA. [3]Department of Health Sciences and Technology, ETH Zurich, Zurich, Switzerland. [4]Paul Scherrer Institute, Villigen, Switzerland. ✉e-mail: gshivasha@ethz.ch; cuhler@mit.edu

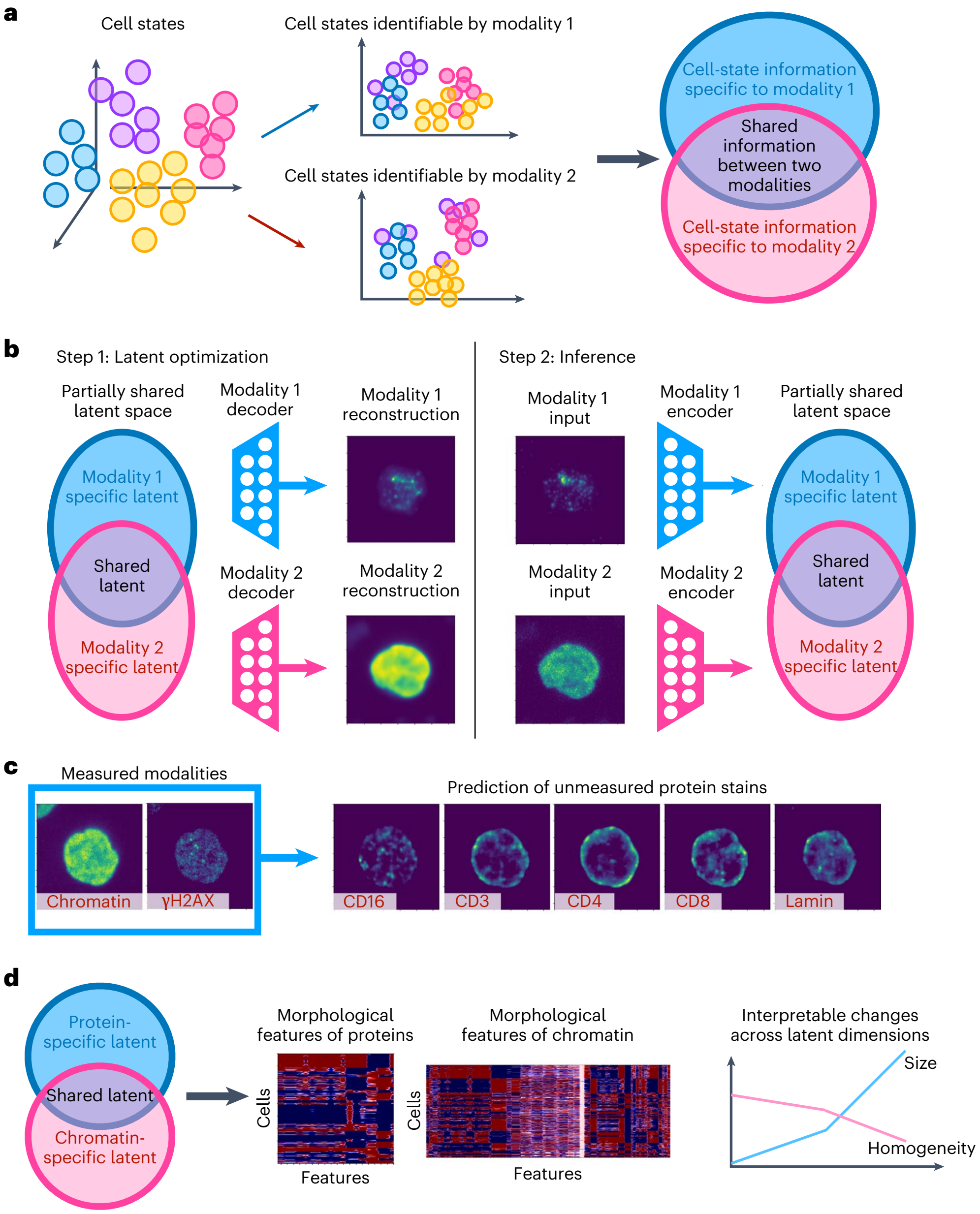

a
Cell states
Cell states identifiable by modality 1
Cell states identifiable by modality 2
Cell-state information specific to modality 1
Shared information between two modalities
Cell-state information specific to modality 2
b
Step 1: Latent optimization
Partially shared latent space
Modality 1 specific latent
Shared latent
Modality 2 specific latent
Modality 1 decoder
Modality 1 reconstruction
Modality 2 decoder
Modality 2 reconstruction
Step 2: Inference
Modality 1 input
Modality 1 encoder
Partially shared latent space
Modality 1 specific latent
Shared latent
Modality 2 specific latent
Modality 2 input
Modality 2 encoder
c
Measured modalities
Chromatin
γH2AX
Prediction of unmeasured protein stains
CD16
CD3
CD4
CD8
Lamin
d
Protein-specific latent
Shared latent
Chromatin-specific latent
Morphological features of proteins
Cells
Features
Morphological features of chromatin
Cells
Features
Interpretable changes across latent dimensions
Size
Homogeneity

Fig. 1 | Overview of APOLLO for partially shared multi-modal embeddings and cross-modality prediction. a, As different experimental technologies capture different aspects of the cell state, we expect some cell-state information to be shared among different modalities and some information to be modality specific. **b**, APOLLO is an autoencoder-based approach that learns three latent spaces to disentangle information captured by each modality. APOLLO uses a two-step training procedure. In step 1, the decoders of each modality are trained so that the decoders can reconstruct the input data from the latent spaces. In step 2, modality-specific encoders are trained to enable inferring the latent space embedding for cells not used in training the model. **c**, Our model enables the prediction of missing modalities. The explicit learning of partial information sharing allows APOLLO to perform accurate cross-modality prediction. The example shows the predicted protein (CD16, CD3, CD4, CD8, lamin) fluorescence images of a cell in a held-out patient, given the cell's chromatin and γH2AX protein stain. **d**, The information captured by the shared and modality-specific latent spaces learned by APOLLO is interpretable. The partially shared latent space can be interpreted by morphological features or genetic pathways.

approaches to integrate diverse modalities, such as chromatin imaging, RNA-seq and ATAC-seq in the single-cell domain, as well as spatial transcriptomics[23,24]. These frameworks extracted features from each modality and embedded the features of all input modalities into a shared latent space for downstream analysis.

Existing multi-modal integration methods learn the union of information from the input modalities, incorporating both the shared and modality-specific information, without disentanglement[17,24,27]. As a result, it is challenging to infer which modality contributes to certain features or phenotypes. For example, while chromatin staining is standardly used as a reference in large multiplexed single-cell-imaging datasets, as it has been shown to contain rich cell-state information and is predictive of protein subcellular localization[31,32], little is known about the information shared between chromatin organization and protein localization. Given the explosion of large-scale paired measurements in the sequencing and imaging domain at the single-cell and tissue levels and also in the medical domain in large-scale biobanks that collect paired information on many individuals[33,34], computational methods are needed that automatically and comprehensively learn the shared and modality-specific information to fully exploit multi-modal measurements.

We present a method that automatically learns partial information sharing between multiple modalities by using an Autoencoder with a Partially Overlapping Latent space learned through Latent Optimization (APOLLO). APOLLO uses one autoencoder per each modality and a two-step training procedure: in step 1, the decoders of each modality are trained while the latent spaces are updated simultaneously, so that the decoders can reconstruct the input data from the latent spaces; in step 2, modality-specific encoders are trained to enable inferring the latent space embedding for cells not used in training the model (Fig. 1b). We test APOLLO's disentanglement performance on five simulated datasets with known ground-truth latent structures. We then demonstrate using a paired scRNA-seq and scATAC-seq dataset (SHARE-seq[5]) that APOLLO can comprehensively and automatically identify and distinguish between gene activity captured by both transcriptomics and chromatin accessibility, as well as by only one of the modalities. We further apply APOLLO to a paired scRNA-seq and surface protein dataset (CITE-seq[25]) to demonstrate that APOLLO can correctly disentangle known shared and modality-specific information. Beyond sequencing modalities, using multiplexed single-cell imaging datasets, we show that APOLLO identifies features of chromatin organization and protein localization that correspond to cell-state information shared between the two modalities, as well as morphological features that are captured by only one modality. The explicit learning of partial information sharing by APOLLO also enables accurate cross-modality predictions, such as predicting unmeasured proteins from chromatin imaging (Fig. 1c). Incorporating additional cellular stains, including microtubule and endoplasmic reticulum (ER), we demonstrate on multiplexed imaging data from the Human Protein Atlas (HPA)[35] that APOLLO can be used to uncover associations between variations in protein subcellular localization and the morphology of different cellular compartments. Overall, APOLLO provides a general framework that enables disentangling and interpreting the shared and modality-specific information contained in multi-modal datasets to obtain biological insights by learning partially shared latent spaces (Fig. 1d).

Results

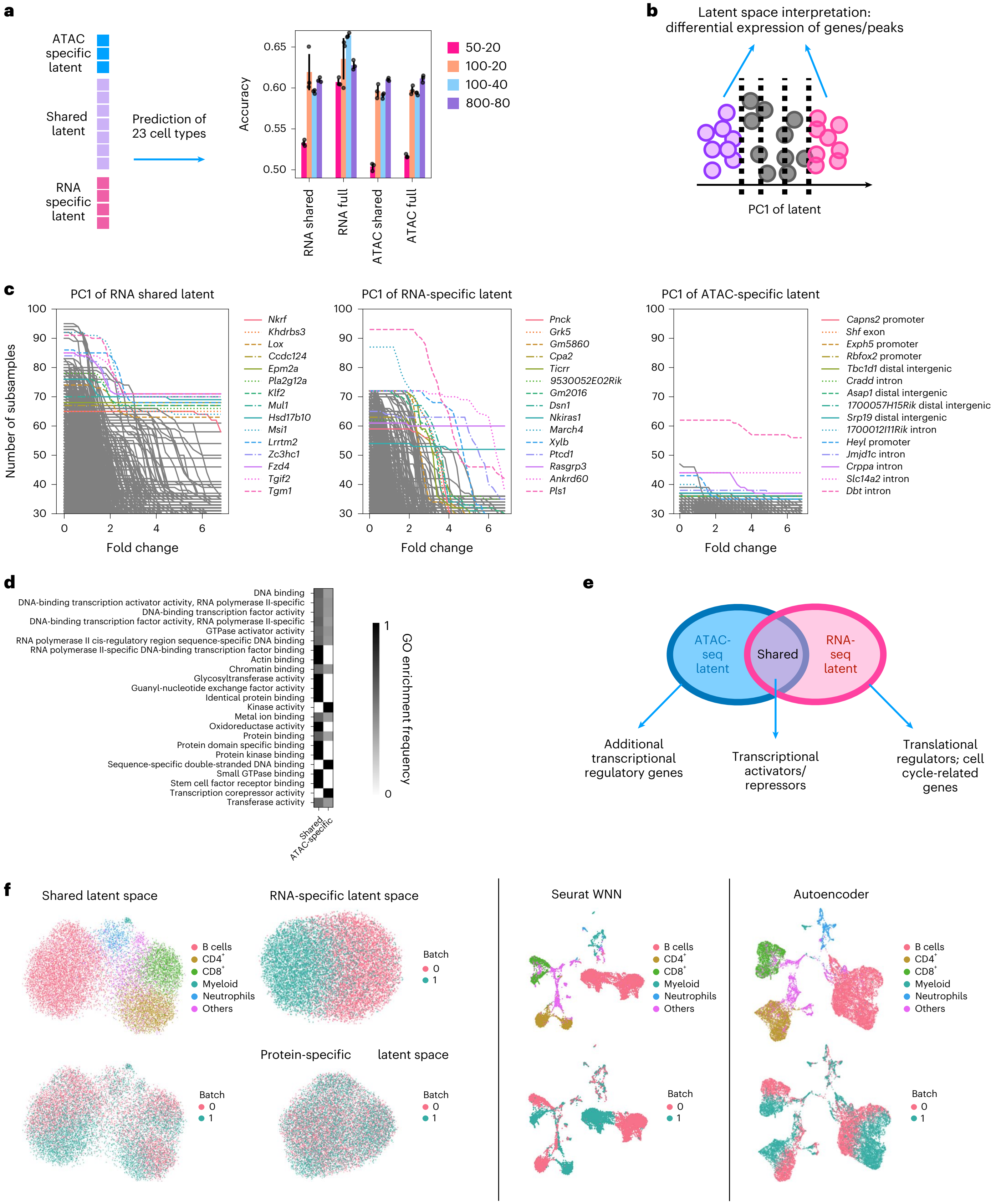
APOLLO enables learning of partial information sharing across data modalities

While a regular autoencoder learns a representation of a single modality, our previous works have demonstrated that a joint representation of multiple modalities can be obtained by using one autoencoder per modality and aligning their latent spaces[23,24]. However, such alignment of the latent spaces results in a single latent space that entangles shared and modality-specific information. To overcome this limitation, APOLLO only enforces alignment between a subset of latent dimensions and allows the remaining latent dimensions to account for domain-specific information. More specifically, let $\mathbf{Z} = (\mathbf{Z}_i)_{i\in H}$ be a latent random vector with distribution P_Z representing the true underlying state of a cell. Each single-cell technology may only measure some aspect of cell state. Let $H_j \subseteq H$ represent the subset of latent variables that can be accessed by technology j and let D_j be a modality-specific map, which takes in the latent vector $\mathbf{Z}_{H_j} := (\mathbf{Z}_i)_{i\in H_j}$ and outputs a data sample $\mathbf{X}^{(j)}$ from modality j, meaning $\mathbf{X}^{(j)} = D_j(\mathbf{Z}_{H_j})$. Note that $\mathbf{X}^{(j)}$ may be high-dimensional (for example, 20,000-dimensional for scRNA-seq technologies or consisting of the number of pixels in a chromatin image). Assuming that we have access to $1 \le j \le J$ paired modalities that can be measured on the same cell, then each cell gives rise to samples $(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(J)})$ from a joint distribution $P(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(J)})$. Given such samples, the problem then becomes to identify the latent features $\mathbf{Z}$ and in particular understand which latent features are shared across modalities and which are modality specific; in the case of $J = 2$, this leads to the task of identifying the shared features $\mathbf{Z}_S = \mathbf{Z}_{H_1\cap H_2}$ as well as the modality-specific features $\mathbf{Z}_{S_1} = \mathbf{Z}_{H_1\setminus H_2}$ and $\mathbf{Z}_{S_2} = \mathbf{Z}_{H_2\setminus H_1}$. To generalize to $J > 2$, $\mathbf{Z}_S$ could represent the shared information across all modalities, meaning $\mathbf{Z}_S = \mathbf{Z}_{\cap_{i=1}^{n} H_i}$, and $\mathbf{Z}_{S_j} = \mathbf{Z}_{H_j\setminus(\cap_{i=1}^{n} H_i, i\neq j)}$.

Without additional assumptions, the shared and modality-specific latent features may not be identifiable. For linear maps D_j, a recent theoretical study proved identifiability under additional assumptions, such as non-Gaussianity of each marginal distribution[36]. Furthermore, when parameterizing the maps D_j with neural networks and using a latent optimization strategy for training, a recent study demonstrated empirically in the context of computer vision that simple image aspects could be disentangled and represented by different latent features[37]. APOLLO builds on these works to automatically learn partial information sharing between multiple modalities by using an autoencoder framework with a partially overlapping latent space learned through latent optimization. While our framework is general, for simplicity we describe it in the bi-modal setting where $J = 2$. In this context, D_1 and D_2 are decoders that map from shared and modality specific latent features to each data modality. Given data $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ from $P(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, the corresponding shared latent features as well as the modality-specific features are learned simultaneously with the decoders D_1 and D_2 by minimizing the reconstruction error $L(\mathbf{x}^{(1)}, D_1(\mathbf{z}_{H_1})) + L(\mathbf{x}^{(2)}, D_2(\mathbf{z}_{H_2}))$, where $L(\cdot)$ is the mean-squared error (MSE) or another appropriate loss function. In practice, the dimension of the shared latent space ($S := H_1 \cap H_2$) is chosen to be much larger than the dimension of the modality-specific latent spaces ($S_1 := H_1\setminus H_2$ and $S_2 := H_2\setminus H_1$), to enforce that information shared by the two modalities is represented in the shared latent space. To encourage this further as well as to enable cross-modality prediction (described below), we introduce two additional decoders D'_1 and D'_2 that map from the shared latent space to each of the modalities, trained by minimizing the reconstruction error $L(\mathbf{x}^{(1)}, D'_1(\mathbf{z}_S)) + L(\mathbf{x}^{(2)}, D'_2(\mathbf{z}_S))$. In the applications of APOLLO to the SHARE-seq and multiplexed imaging datasets described in the following sections, we demonstrate that APOLLO is robust to the choice of latent space dimensions as well as to whether the two additional decoders D'_1 and D'_2 are included. The full set-up and objective function can be found in Methods.

Fig. 2 | APOLLO for identifying shared and modality-specific information in paired scRNA-seq and scATAC-seq data. a, Comparison of four separate models with different latent space dimensions. The dimensions are listed as 'shared latent space dimension'-'modality-specific dimension'; for example '50-20' indicates that the shared latent space has 50 dimensions and the modality-specific latent spaces have 20 dimensions each. For each model, we train four separate classifiers, with three random train–test splits, that predict cell type based on the following inputs respectively: (1) the shared latent space encoded using scRNA-seq (RNA shared) or scATAC-seq (ATAC shared), and (2) the shared latent space concatenated with the RNA-specific latent space (RNA full) or the ATAC-specific latent space (ATAC full) encoded using scRNA-seq or scATAC-seq. The plot shows the mean and standard deviation of each model. The baseline accuracy computed by randomly permuting cell-type assignments is 0.108 with a standard deviation of 0.03 across 5 random permutations. **b**,**c**, The genes or peaks explained by each latent space can be identified by performing differential expression using the cells at the two ends and at the center of each PC (**b**). The curves in **c** show the number of times among 108 random train–test splits that a gene passes a particular fold-change threshold, while the *P*-value cut-off is set to 0.05 after multiple testing correction (see Methods for details). The top genes with the highest area under the curve are colored. **d**, GO enrichment analysis using the differentially expressed genes of the top PCs identified as described in **c**, for the shared and ATAC-specific latent spaces. The heatmap shows the proportion of times a GO term is observed to be enriched in one of the latent spaces. **e**, Summary of genes captured in each of the three latent spaces. **f**, Application of APOLLO to the CITE-seq data from ref. 25. Left: UMAPs of the shared and the two modality-specific latent spaces colored by cell types (top row) or experimental batches (bottom row). Middle: UMAPs obtained from Seurat WNN analysis. Right: UMAPs obtained from the shared latent space of standard multi-modal autoencoder with an encoder and a decoder for each input modality (see Methods for details).

To generalize to unseen samples and to enable cross-modality prediction, we use a second training step to obtain encoders E_j that map from each data modality to their respective latent spaces. More precisely, given a sample $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(J)})$ and the corresponding features $\mathbf{z}$ obtained from the first training step, the encoders E_j, $1 \le j \le J$, are obtained by minimizing the MSE $L(\mathbf{z}_{H_j}, E_j(\mathbf{x}^{(j)}))$. The encoders and the previously described decoders can be any neural network architecture that is suitable for the particular data modality, such as convolutional networks for images and fully connected networks for gene expression. For cross-modality prediction, a sample $\mathbf{x}^{(j)}$ from input modality j is



first encoded to the latent space to obtain $\mathbf{z}_{H_j} = E_j(\mathbf{x}^{(j)})$. Then the decoder D'_k of the target modality k is applied to the dimensions corresponding to the shared latent space to obtain $\mathbf{x}_k = D'_k(\mathbf{z}_{H_j \cap H_k})$. Additional details of model set-up and training are provided in Methods.

Given the limited understanding of information sharing between multiple modalities, to systematically evaluate disentanglement performance, we design simulated datasets with known ground-truth latent structure with increasing complexity and relevance to real biological data (Extended Data Figs. 1–4 and Methods). We assess APOLLO's performance across five simulation settings that differ in two aspects: (1) whether some modality-specific latent features are children of the shared latent features; and (2) whether the observed features depend on more than one latent feature. It is worth noting that depending on the structure of the latent causal graph, the shared and modality-specific features may not necessarily be independent (Extended Data Figs. 3a and 4a). In all five simulation settings, APOLLO correctly identifies the associations between latent features and the three latent spaces as measured by the separation of known latent features in the corresponding latent spaces (Methods). Importantly, APOLLO has comparable performance when there is dependence between shared and modality-specific latent features (Extended Data Figs. 3 and 4) as well as when the observed features depend on more than one latent feature (Extended Data Figs. 2 and 4), which is to be expected in biological data.

In the following, we demonstrate that APOLLO is generally applicable to any multi-modal data by discussing four different multi-modal settings, including paired-sequencing-based modalities as well as multiplexed imaging.

APOLLO provides a general framework for the integration of paired-sequencing-based measurements

We demonstrate that APOLLO learns meaningful shared and modality-specific latent spaces and is generally applicable to paired-sequencing-based measurements using two representative datasets: paired scRNA-seq and scATAC-seq measured using SHARE-seq[5] (Extended Data Fig. 5 and Supplementary Fig. 1) and paired scRNA-seq and cellular surface protein abundance measured using CITE-seq[25] (Extended Data Fig. 6a). Training APOLLO using the SHARE-seq dataset on a 24 GB graphics processing unit (GPU) (A5000) took 10.8 seconds per epoch for step 1 and 7.5 seconds per epoch for step 2.

Using the SHARE-seq data, we first assess whether the modality-specific latent spaces capture additional information compared with the shared latent space using a cell-type classification task (see Methods for details). As a baseline, we used the cell types defined by Ma et al.[5] to train a classifier that predicts the cell types using the shared latent space. In addition, we trained two separate classifiers that use the ATAC-seq latent space and RNA-seq latent space respectively, in addition to the shared latent space. To ensure that our model is robust to the choice of latent space dimensions, we tested different dimensions for the shared and modality-specific latent spaces as well as different ratios between the latent space dimensions (Fig. 2a). The incorporation of the RNA-specific latent space improves the cell-type classification accuracy compared with using the shared latent space alone for all latent space dimensions. This shows that the modality-specific latent spaces can capture biologically meaningful information that is not represented in the shared latent space, and that our model is robust to the choice of latent space dimensions. Uniform Manifold Approximation and Projection (UMAP) visualizations of the encoded shared and full latent spaces further support the separation of cell types captured by APOLLO (Supplementary Fig. 1d).

We next demonstrate how principal component (PC) analysis can be used to interpret the information contained in the shared and modality-specific latent spaces. Along each PC, we bin the cells based on their positions along the PC and then compare the gene expression or peak counts of the cells at the two ends of the PC and at the center of the PC (Fig. 2b and Methods). We identify cell cycle-related genes among the top genes explained by the first PC of the RNA-specific latent space, such as *Ticrr* and *Dsn1*[38,39] (Fig. 2c), while the top PCs of the ATAC-specific latent space capture the activity of promoters of transcriptional regulators, for example, the promoter of the transcription factor *Heyl* (Fig. 2c and Supplementary Fig. 2b). Furthermore, the top genes explained by the first three PCs of the RNA and ATAC shared latent space contain known transcriptional regulators, some of which are known transcriptional activators or repressors (for example *Zeb1*) based on the correlation between transcription factor activity and transcription factor RNA expression levels[5] (Fig. 2c and Supplementary Fig. 3a). Interestingly, the top genes of the ATAC-specific or the shared latent space are generally not related to the cell cycle and the top genes of the RNA-specific latent space are not related to transcriptional regulation; exceptions are the gene *Nkrf* captured in the shared latent space and the gene *CBx2* captured in the RNA-specific latent space, which are known to regulate the cell cycle[40,41] (Fig. 2c and Supplementary Fig. 3b).

Gene ontology (GO) enrichment analysis is performed on the genes explained by each latent space, identifying GO terms related to transcriptional regulation in the shared latent space and the ATAC-specific latent space (Fig. 2d and Supplementary Fig. 3c). This could indicate a time lag between the change in chromatin accessibility and the resulting change in gene expression. Interestingly, GO terms related to post-transcriptional regulation are found in the shared latent space. These analyses demonstrate that the shared and modality-specific latent spaces identified by APOLLO are biologically meaningful and can be efficiently analyzed to interpret the shared and modality-specific information (Fig. 2e).

To further demonstrate that APOLLO can disentangle shared and modality-specific information for different kinds of sequencing-based measurements, we applied APOLLO to a CITE-seq dataset of mouse spleen and lymph nodes with paired scRNA-seq and surface protein measurements obtained from two wild-type mice processed in two separate experiments[25]. We used the same training strategy and model architecture as in the application to the SHARE-seq data. As cross-modality prediction is not the goal in this application, we removed the two decoders D'_1 and D'_2 that map from the shared latent space to each of the modalities (Extended Data Fig. 6a). Using a standard preprocessing and visualization procedure implemented in Scanpy[42], the resulting UMAP plots show that the major cell types can be separated by either scRNA-seq or protein abundance and that the scRNA-seq data show further separation of the cells by mouse, indicative of experimental batch effects (Extended Data Fig. 6b,c). The latent spaces of our APOLLO model sufficiently disentangle the multi-modal information for downstream applications: the shared latent space between scRNA-seq and protein abundance shows separation by cell types but not by experimental batches, while batch separation is captured by the modality-specific latent space of scRNA-seq (Fig. 2f and Extended Data Fig. 7a–h). In contrast, applying existing multi-modal integration methods without additional batch correction, such as the popular weighted-nearest neighbor (WNN) method in Seurat[17] or a standard multi-modal autoencoder[24], results in a latent space that learns the union of information from both modalities, in which cells are separated by both cell types and batches without disentanglement (Fig. 2f, Extended Data Fig. 6d and Methods). Existing integration methods as well as using APOLLO's full scRNA-seq latent space both result in worse performance in cell-type clustering (Extended Data Fig. 7i–l). This analysis further demonstrates that APOLLO correctly disentangles the shared and modality-specific information while integrating different kinds of paired-sequencing modalities, which could not be achieved by existing methods that perform only integration.

APOLLO learns partially shared latent spaces of chromatin and different proteins through conditioning

APOLLO is a general framework that can be applied to any paired multi-modal data by choosing the appropriate modality-specific encoders and decoders. Furthermore, APOLLO can directly be applied to integrate more than two modalities. We introduce a version of APOLLO that incorporates conditioning in the latent space to allow for the integration of chromatin images and images of multiple protein types.

In the following, we apply APOLLO to a multiplexed imaging dataset of human peripheral blood mononuclear cells (PBMCs) from ref. 43. The dataset consists of a total of 32,345 PBMCs from 40 patients with 1 of 4 diagnoses: healthy, meningioma, glioma, head and neck tumor (Supplementary Fig. 4). For each patient, two different multiplexed imaging datasets were obtained. One subset of cells was stained with 4′,6-diamidino-2-phenylindole (DAPI) to label the chromatin, along with antibodies for CD4, CD8 and CD16. The other subset of cells was stained with DAPI and antibodies for lamin, CD3 and γH2AX. To incorporate multiple proteins, a trainable vector of protein ID, shared across all images of the same protein, is concatenated to the inputs to the decoders as well as to the last hidden layer of the encoders (Extended Data Fig. 8, Supplementary Fig. 5 and Methods). This conditional model accurately reconstructs images of cells in held-out individuals (Fig. 3a and Supplementary Fig. 6), which ensures that APOLLO can comprehensively capture biologically relevant information from the input modalities. Training APOLLO on this dataset on a 24 GB GPU (A5000) took 40 seconds per epoch for step 1 and 20 seconds per epoch for step 2.

APOLLO enables accurate cross-modality prediction of protein localization from chromatin imaging

The number of proteins that can be simultaneously imaged in a cell is limited, ranging from a single protein using endogenous tagging[44] to around 30 proteins in fixed samples[1]. In the following, we show that the shared latent space learned by APOLLO enables cross-modality predictions and can be applied to predict the unmeasured proteins in a cell from the chromatin image of that cell (Extended Data Fig. 8a and Methods).

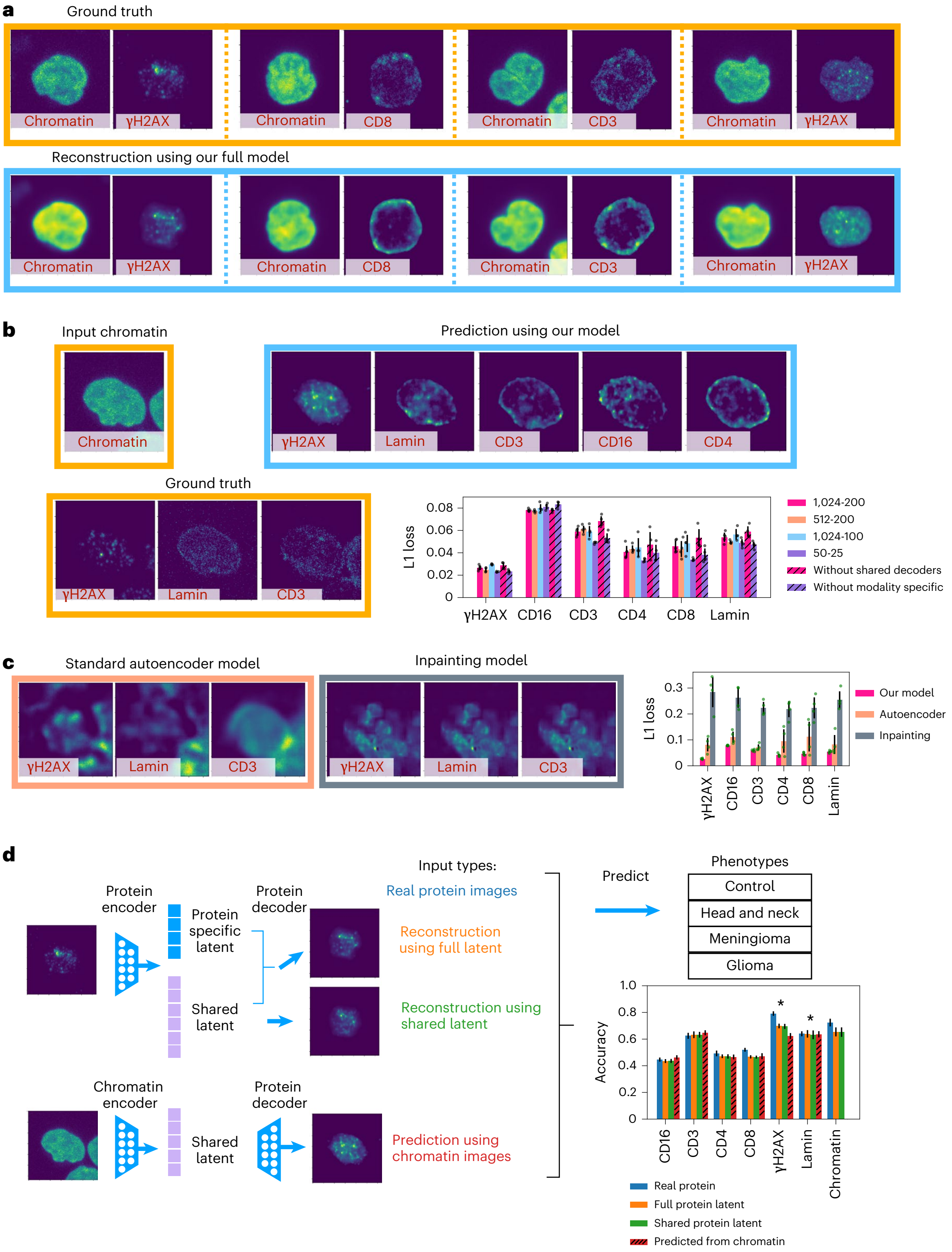
The task of predicting images of unmeasured protein stains has been considered using supervised neural network models[45,46]. In particular, image inpainting methods have been applied to predict the protein image of a target cell from its chromatin and microtubule stains given all three images of other cells[45]. While such inpainting methods can capture the average distribution, they do not produce realistic single-cell images (Fig. 3b,c, Supplementary Fig. 7e and Extended Data Fig. 9). In contrast, APOLLO can accurately predict unmeasured proteins at the single-cell level in held-out individuals based on chromatin images (Extended Data Fig. 9), and the performance of APOLLO is robust to the choice of latent space dimensions (Fig. 3b,c). To interrogate the necessity of each component of APOLLO, we perform an ablation study for the cross-modality prediction task. We first compare our two-step latent optimization training to the standard autoencoder training procedure, where encoders and decoders have the same architecture as our default APOLLO model but are trained simultaneously without directly updating the parameters of the latent spaces (Supplementary Fig. 7a and Extended Data Fig. 10a). We find that standard one-step training is not able to generate realistic protein images in held-out individuals (Fig. 3b,c and Extended Data Fig. 9). Furthermore, we tested the effect of removing the separation between the shared and modality-specific latent spaces while keeping the same two-step training procedure (Supplementary Figs. 7b–d and Extended Data Fig. 10b). This model without disentanglement of modality-specific information shows comparable performance to our APOLLO model that uses only the (smaller-dimensional) shared latent space for prediction (Fig. 3b), which indicates that the shared latent space has captured all shared information. These results suggest that the improvement in prediction performance by APOLLO is mainly a result of the two-step training procedure and that the latent space disentanglement allows for the use of a smaller latent space dimension (two-thirds of the full latent space) for cross-modality prediction at test time. Similar to the applications to paired-sequencing modalities, it is also possible to remove the decoders that map from the shared latent space to each of the modalities while maintaining comparable performance in cross-modality prediction (Fig. 3b).

Next, we demonstrate that the protein images predicted from chromatin images have similar performance on a downstream classification task as the real protein images. For each protein, we train four separate convolutional neural network classifiers with the same architecture to predict the phenotype of the individual from whom the input cell image was obtained from: healthy, meningioma, glioma, or head and neck tumor. For each protein, the four classifiers use the following inputs respectively: (1) real protein images; (2) reconstructed protein images using the full protein latent space (the shared latent and protein-specific latent spaces); (3) reconstructed protein images using only the shared latent space; and (4) protein images predicted from chromatin images through the shared latent space. While the reconstructed protein images have similar phenotype classification accuracy across different proteins as the real protein images (Fig. 3d), reconstruction from the full latent space has slightly better classification accuracy than reconstruction from the shared latent space for γH2AX and lamin (Fig. 3d), thereby indicating that the protein-specific latent spaces are able to learn disease-relevant information that is not shared by chromatin. Moreover, Fig. 3d shows that the predicted protein images and the real protein images have comparable phenotype classification accuracy, indicating that CD3 is a better predictor of phenotype than CD16, CD4 or CD8, which suggests that the protein images predicted from chromatin capture similar disease-relevant information as the real protein images. These results demonstrate that APOLLO is capable of accurately predicting protein localization from chromatin organization and the predicted protein images can be used for downstream tasks with performance mimicking that of real protein images.

Fig. 3 | APOLLO for predicting protein localization from chromatin imaging. **a**, Examples of reconstructed images of cells from patients held-out from the PBMCs imaging dataset in ref. 43, compared with ground-truth images. **b**, Protein predictions from chromatin images for a cell where the ground-truth protein images are available for the three proteins (γH2AX, lamin and CD3). The average prediction loss for each protein is quantified comparing the APOLLO model with different latent space dimensions to two alternative models, namely, our model without the decoders that map from the shared latent space to the output (without shared decoders) and our model without separating the shared and modality-specific latent spaces (without modality specific). The different latent space dimensions tested are listed as 'shared latent space size'-'modality-specific latent space size', for example, 1,024-200. The error bars show the standard deviations across 5 random initializations for the model with sizes 1,024-200 and 4 random initializations for all other models. **c**, Cross-modality predictions obtained using our full model compared with those obtained using a standard multi-modal autoencoder training procedure as well as a prior image inpainting model[45] (see Methods for details). Data are presented as mean ± s.d. of 5 random initializations for our model and 4 random initializations for the other models. Protein predictions from chromatin images are shown for the same cell used in a. **d**, Comparison of the model predictive performance using different input types: real protein/chromatin images (real protein), the reconstructed images from the full latent space (full protein latent), the reconstructed images from only the shared latent space (shared protein latent), and the protein images predicted from chromatin images (predicted from chromatin). Data are presented as mean ± s.d. of 36 different train–test splits. The asterisks indicate that the prediction performance of the protein images reconstructed from the full latent space is significantly better than the protein images reconstructed from only the shared latent space ($P = 0.0011$ for γH2AX and $P = 0.098$ for lamin using one-sided two-sample t-test. $P = 0.00088$ for γH2AX and $P = 0.047$ for lamin using Wilcoxon signed-rank test.)

APOLLO identifies interpretable morphological features between chromatin organization and protein localization

In contrast to paired-sequencing measurements, where the different modalities can be compared and interpreted at the level of genes[5], shared features are not directly available for paired image modalities. APOLLO provides a systematic framework that can be applied also in these settings to interrogate the information that is shared between modalities and specific to each modality. To identify the morphological features captured by the shared and modality-specific latent spaces, we perform PC analysis and bin cells along each PC (see Methods for details). Cells along a particular PC can then be sampled and visualized to interrogate the information captured in the shared or modality-specific latent spaces (Fig. 4a). We use the set of chromatin and protein features defined in ref. 43 to compare the cells at the two ends of each PC and at the center of the PC for each latent space, thereby allowing us to identify the main features captured by each latent space (Fig. 4a, Supplementary Figs. 8–14 and Methods). For example, we find that the first PC of the shared latent space mainly captures bounding box area and homogeneity of chromatin (Fig. 4a).



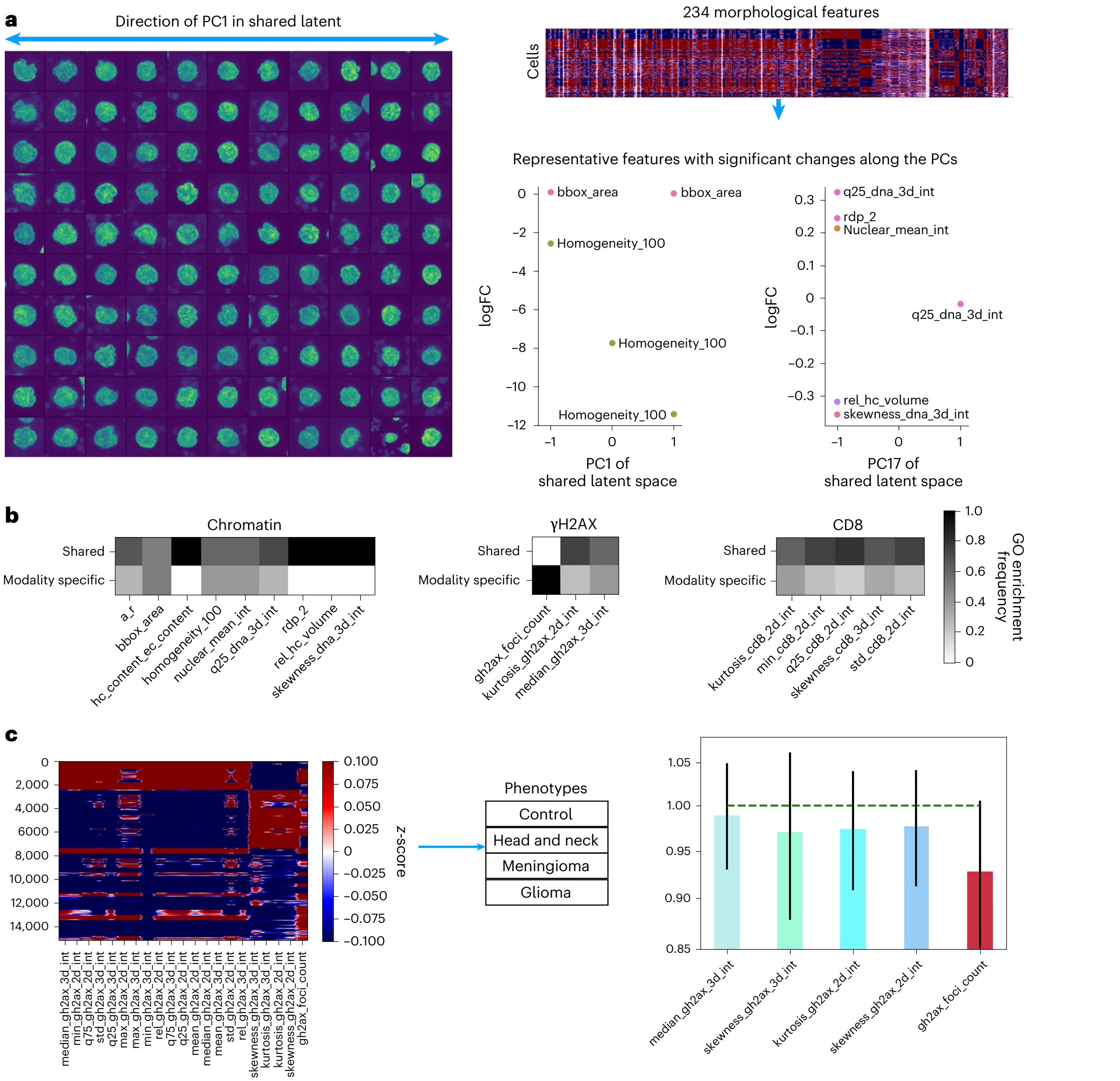


Fig. 4 | Interpretable morphological features in the shared and modality-specific APOLLO latent spaces of chromatin organization and protein localization. a, Interpretation of the latent space learned from the PBMCs multiplexed imaging dataset[43]. Left: cells separated along the first PC of the chromatin shared latent space into 11 bins of equal percentiles; randomly selected cells in each bin are shown. Right: 234 predefined handcrafted morphological features are computed for each cell and the values of these features are computed for the shared and modality-specific latent spaces at the two ends and at the center of each PC. The changes of representative morphological features along PC1 and PC17 of the chromatin shared latent space are plotted. FC, fold change. **b**, Heatmaps showing the proportion of times each representative morphological feature is explained by a top PC in the shared or the modality-specific latent space. **c**, Heatmap showing the morphological feature values in each cell, where each row represents a cell. The bar plot shows the fold change of prediction accuracy per each feature ablation compared with the prediction accuracy using all features. Error bars indicate standard deviations across 36 different random initializations. *P* values of two-sided *t*-tests between the prediction accuracy of each feature ablation and the accuracy using all features are 0.46 (median_gh2ax_3d_int), 0.16 (skewness_gh2ax_3d_int), 0.11 (kurtosis_gh2ax_2d_int), 0.14 (skewness_gh2ax_2d_int) and 0.00022 (gh2ax_foci_count).

To more comprehensively characterize the information captured by each latent space, we compare the image features contained in the top PCs that explain 70% of total variance in each latent space (Fig. 4b and Supplementary Figs. 9c, 10c, 11c and 14c). Interestingly, such analysis shows that heterochromatin volume, a feature associated with aging[47] and Alzheimer's disease[23], is exclusively captured by the chromatin and protein shared latent space (namely, in PC17; Fig. 4a). Whereas homogeneity, mean intensity and aspect ratio of chromatin are captured by both the shared and the chromatin-specific latent spaces. A similar analysis applied to each protein shows that the foci count of γH2AX is a protein-specific feature that is not captured by the chromatin and protein shared latent space (Fig. 4b). When training separate regression models to predict the three γH2AX features shown in Fig. 4b based on chromatin images (see Methods for details), the regression outputs show the lowest correlation with the ground-truth feature values for the foci count of γH2AX compared with the other two features that are mainly captured by the shared latent space (Supplementary Fig. 13c). This indicates that APOLLO correctly identified the γH2AX foci count as a protein-specific feature while the other two features are also captured by chromatin imaging. A feature ablation test indicates that removing γH2AX foci count results in the largest reduction of phenotype classification accuracy (Fig. 4c) and is consistent with the significant reduction of phenotype classification accuracy observed when the protein-specific latent space of γH2AX is not used in reconstructing the protein images (Fig. 3d). This confirms that the modality-specific latent space can capture important disease-relevant features and demonstrates how APOLLO can be used to interrogate the shared and modality-specific information between imaging modalities, where shared features are not directly available.

APOLLO identifies associations between protein subcellular localization and cell morphology

Function and activity of a protein is known to be tightly coupled to its subcellular localization, which can vary across single cells even within the same cell line[35,44,48–50]. Computational tools have been developed to analyze single-cell variability in protein localization, showing that cellular and nuclear morphology can be used to predict protein subcellular localization in single cells[32,45,51]. However, these models take as input images of multiple cellular components and little is known about the association of each cellular component with the change in protein localization across single cells. We apply APOLLO to images of U2OS cells (the most abundant cell line) in the HPA[35] to learn shared and modality-specific information between the morphology of nucleus, microtubule and ER with respect to variations in protein subcellular localization (see Methods for details). In HPA, all cells are stained for nucleus, microtubule and ER (the 3 reference stains), together with 1 additional target protein (total of 11,657 proteins across all U2OS cells; see Methods for details). Concentrating on the 25 proteins with the most variable subcellular localization (see Methods for details), we disentangle the information in the three cellular components to analyze their association with the variability in protein subcellular localization.

Toward this, we separately train three APOLLO models to learn the shared and modality-specific information between each pair of the three reference stains (see Methods for details). We then cluster the cells in each latent space into 2 clusters and test for each of the 25 proteins whether the 2 clusters capture differences in their subcellular localization as measured by the proportion of protein localized within the cell nucleus (Fig. 5a and Methods). Notably, the three models consistently indicate that variation in intranuclear localization of many proteins is differentially captured by the three different compartments. For example, morphological features of both ER and microtubule, but not the nucleus, are associated with the variability in intranuclear localization of DDB1 (Fig. 5a-c), which is known to change localization in response to ultraviolet exposure[52]. Both the microtubule and ER modality-specific latent spaces indicate that cells with smaller cytoplasmic volume and lower intensity of microtubule and ER are associated with high intranuclear localization of DDB1 (Fig. 5b,c). In contrast, the intranuclear localization of CLNS1A and C8orf59 is only associated with morphological features of the nucleus (Fig. 5a,d,e). Interestingly, for CLNS1A, a protein involved in small nuclear ribonucleoprotein biogenesis and control of cell volume[53–55], our model suggests that higher intranuclear localization is found in cells with higher heterochromatin content (Fig. 5d); for C8orf59, a protein involved in ribosome biogenesis[56], our model suggests that increased intranuclear localization is associated with more circular nuclei (Fig. 5e).

For some proteins, the difference in intranuclear localization is captured by more than one modality-specific latent space but not the shared space in the same model; for example, both the nucleus-specific and ER-specific latent spaces but not the shared space capture differences in intranuclear localization of ATE1 across cells (Fig. 5a). This indicates that multiple aspects of nuclear and cellular morphology, captured by the different latent spaces, could simultaneously contribute to the localization of a particular protein in different ways. Indeed, analyzing the clusters in the different modality-specific latent spaces shows that while they are associated with differences in the intranuclear localization of a protein, the cell clusters are different (Supplementary Fig. 15a–c and Methods). For example, while both the ER-specific and nucleus-specific latent spaces show significant separation of cells by their intranuclear localization of ATE1 (Fig. 5a,f), visualizing cells in the clusters of the different latent spaces suggests that the ER-specific clusters capture differences in ER intensity and cytoplasmic volume (Fig. 5g), while the nucleus-specific clusters capture differences in nuclear sizes (Fig. 5h). A similar analysis for CIZ1 of the clusters in the microtubule-specific latent space suggests that low intranuclear localization is associated with larger ratio of cytoplasmic-to-nuclear volume (Supplementary Fig. 15e), which is not a distinctive feature between the clusters of the nucleus-specific latent space (Supplementary Fig. 15d). This analysis demonstrates that APOLLO can generalize to different multiplexed imaging experiments and provide insights on relations between protein subcellular localization and the morphology of different cellular components.

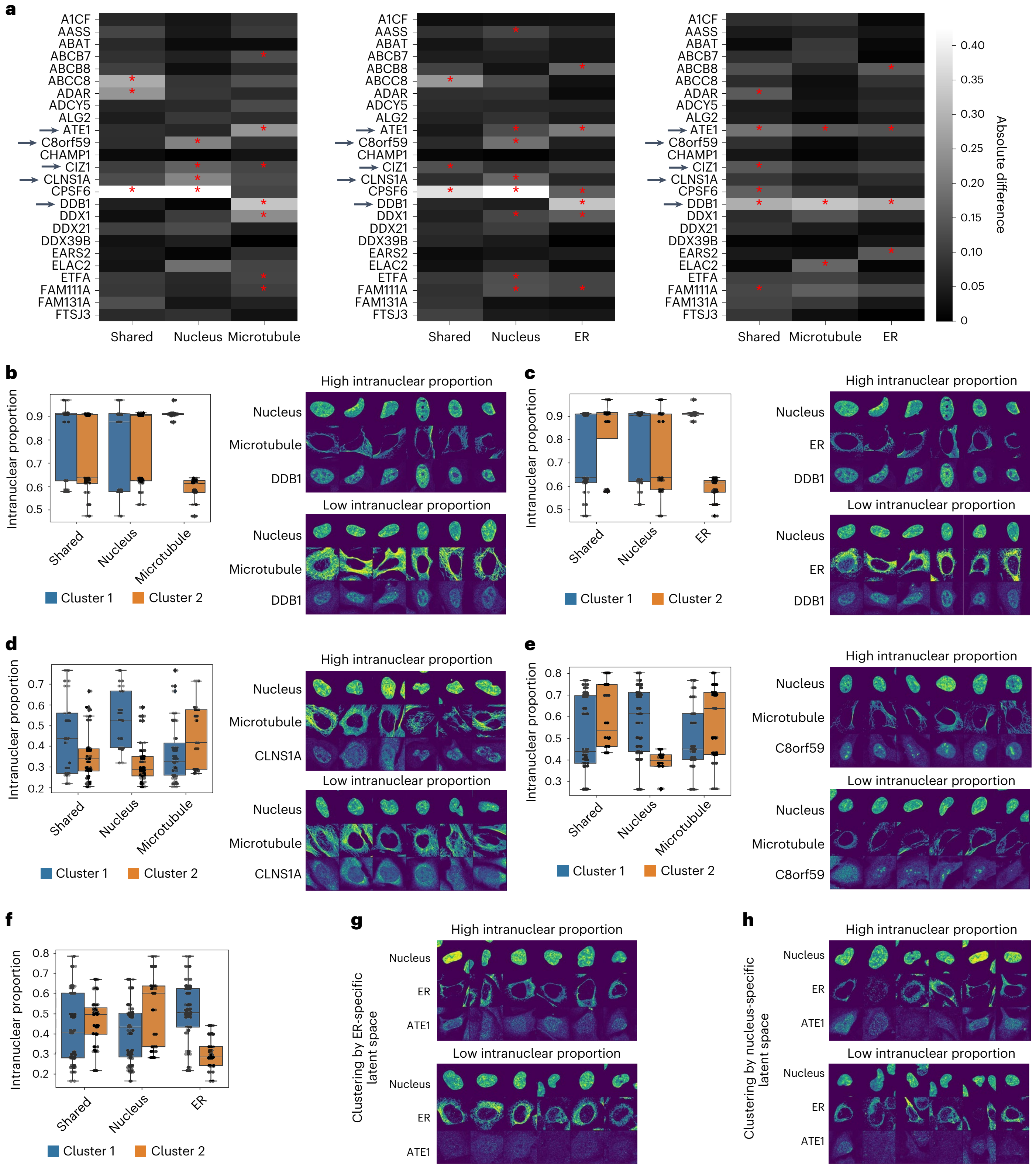
Discussion

We introduced APOLLO, a general computational framework that uses autoencoders with partially overlapping latent spaces to explicitly model the shared and modality-specific information across diverse multi-modal single-cell datasets. This design enables accurate cross-modality prediction and provides a systematic way to link interpretable features to cell state. Although we have demonstrated that APOLLO performs well on a variety of simulated and real data and APOLLO is based on theoretical causal foundations, the latent optimization implementation does not have known theoretical guarantees and thus the exact conditions under which APOLLO correctly disentangles multi-modal information are unknown. Our work motivates several future research directions. First, although we demonstrated in the SHARE-seq and multiplexed imaging applications that APOLLO is robust to the choice of latent space sizes, accurate estimation of the intrinsic dimensionality of the shared and modality-specific latent spaces could provide additional guidance for parameter selection and insight into underlying biological mechanisms. While recent theoretical work shows that these dimensions are identifiable under suitable conditions[36], further methodological development is needed to devise practical algorithms for estimating intrinsic dimensionality across different modalities. Second, the Gaussian noise added to APOLLO latent spaces could be replaced with learnable noise parameters to

Fig. 5 | APOLLO for modeling different cellular components with respect to protein subcellular localization. a, Comparison of three APOLLO models trained using nucleus and microtubule stains (left), nucleus and ER stains (middle), and microtubule and ER stains (right) from the HPA dataset[35]. For each model, each of the shared and the two modality-specific latent spaces is clustered into two clusters. The heatmap shows the absolute value of the difference in the mean intranuclear proportion in each cluster, averaged across the five random initializations (see Methods for details). *$P < 0.00022$ using two-sided t-test and absolute difference between clusters >0.1 in all 5 random initializations (see Methods for details). The arrows indicate proteins with example images shown below and in Supplementary Fig. 15. **b**–**e**, Intranuclear proportions are computed in a total of 37,674 single-cell images, with at least 150 single-cell images per protein. Left: Box plot of the intranuclear proportions of DDB1 (**b**,**c**), CLNS1A (**d**) and C8orf59 (**e**) in the two clusters of each latent space, obtained using the models trained with either the nucleus and microtubule images (**b**,**d**,**e**) or the nucleus and ER images (**c**). Right: examples of cells in each cluster of the microtubule-specific latent space (**b**), the ER-specific latent space (**c**) or the nucleus-specific latent space (**d**,**e**) stained for the particular proteins. **f**, Box plot of the intranuclear proportions of ATE1 in the two clusters of each latent space, obtained using the model trained with nucleus and ER images. **g**, Examples of cells stained for ATE1 in each cluster of the ER-specific latent space. **h**, Examples of cells stained for ATE1 in each cluster of the nucleus-specific latent space. Boxplots: center line, median; box limits, upper and lower quartiles; whiskers, 1.5x interquartile range.

enable uncertainty quantification. Third, as multi-modal measurements become increasingly common, extending APOLLO to unpaired measurements could provide important insights when paired assays are not feasible—for example, when obtaining spatial transcriptomics and multiplexed protein stains on adjacent tissue slices. One potential approach is to model unpaired data using two shared latent spaces learned during step 1 training, with a distribution-matching loss, such as maximum mean discrepancy, that encourages the shared spaces to represent the same distribution of cells across modalities.

Collectively, APOLLO enables explicit learning of shared and modality-specific information, leading to a more holistic understanding of cell states and their underlying regulatory mechanisms. While we demonstrated APOLLO on SHARE-seq, CITE-seq and multiplexed protein staining, the method is broadly applicable to other multi-modal single-cell measurements by adapting the encoder and decoder architectures. More broadly still, APOLLO could be applied beyond single-cell settings to individual-level multi-modal data, leveraging diverse medical modalities in large biobanks[33,34]. APOLLO



thus offers a framework that goes beyond multi-modal integration[57] by enabling mechanistic interpretation of the learned latent space through information disentanglement.

Methods

Latent optimization

In practice, similar to variational sampling in autoencoders, to improve generalization to unseen samples, we add Gaussian noise to each component of the latent space and add a regularization term (ℓ_2-norm of the latent features) in addition to the reconstruction losses. Thus, the full objective function is:

$$\begin{aligned}&\mathbf{E}_{(x^{(1)},x^{(2)})\sim P(X^{(1)},X^{(2)})}[L(x^{(1)},D_1(z_S+\epsilon_S,z_{S_1}+\epsilon_{S_1}))\\&+L(x^{(2)},D_2(z_S+\epsilon_S,z_{S_2}+\epsilon_{S_2}))+L(x^{(1)},D_1'(z_S+\epsilon_S))\\&+L(x^{(2)},D_2'(z_S+\epsilon_S))]+\lambda(\parallel z_S\parallel_2+\parallel z_{S_1}\parallel_2+\parallel z_{S_2}\parallel_2),\end{aligned}\tag{1}$$

where ϵ_S, ϵ_{S_1} and ϵ_{S_2} denote Gaussian noise in each component of the latent space and λ is a hyperparameter. We minimize equation (1) to obtain the decoders D_1, D_2, D_1' and D_2' as well as the shared latent features z_S and the modality-specific latent features z_{S_1} and z_{S_2}.

Model architecture and training of paired scRNA-seq and scATAC-seq

For both training steps, we split the data into training and validation sets, as is standard in neural network training. We use the same randomly selected 85% of cells from the SHARE-seq dataset[5] to train our model and the remaining 15% to validate and test the generalization performance of our model.

Step 1 latent optimization. In this step, we train the latent spaces and the corresponding decoders to reconstruct the scRNA-seq and scATAC-seq data (Extended Data Fig. 5b and Supplementary Fig. 1a). The training was performed on one 24 GB GPU for 35 hours with 10.8 seconds per epoch.

Latent spaces of paired scRNA-seq and scATAC-seq. The shared latent space has 50 dimensions and each of the two modality-specific latent spaces have 20 dimensions. The shared latent space is chosen to be much larger than the modality-specific latent spaces to ensure that the shared space has enough capacity to contain all shared information. Latent spaces are initialized using 'torch.nn.Embedding' with the default parameters. Independent Gaussian noise with zero mean and unit variance is added to the latent spaces at each training epoch. The hyperparameter λ in equation (1) for the ℓ_2 regularization of the latent spaces is set to 0.001 and the learning rate of the ADAM optimizer is set to 0.001, which are the same hyperparameter values as in ref. 37.

Decoders of paired-sequencing-based modalities. Four decoders are trained in step 1: (1) a decoder that reconstructs scRNA-seq from the shared latent space; (2) a decoder that reconstructs scRNA-seq from the full RNA-seq latent space (the shared latent space embedding concatenated with the RNA-specific latent space embedding); (3) a decoder that reconstructs scATAC-seq from the shared latent space; and (4) a decoder that reconstructs scATAC-seq from the full ATAC-seq latent space (the shared latent space embedding concatenated with the ATAC-specific latent space embedding). The architectures of the decoders are shown in Extended Data Fig. 5b. The input feature dimension is 70 for the full latent space decoder and 50 for the shared latent space decoder except when different latent space sizes are tested (Fig. 2a). Each decoder has 3 hidden layers, each of which has dimension 1,024. All hidden layers are linear layers with a dropout rate of 0.01 and are followed by LeakyReLU activation and a batch normalization layer. The output layer is a linear layer with a dropout rate of 0.01 and sigmoid activation. This decoder architecture follows standard autoencoder set-ups[23,58]. For both modalities, we use binary cross-entropy loss to minimize the objective function defined in equation (1), which is calculated using 'torch.nn.BCEWithLogitsLoss' based on the output before the sigmoid activation with pos_weight set inversely proportional to the total counts of genes or peaks. The learning rate of the decoders is 0.0001 and ADAM is used for optimization, which is the same as in ref. 37.

Step 2 inference. In this step, we train the encoders to infer the learned latent spaces from the scRNA-seq and scATAC-seq data without updating the latent spaces or the decoders (Extended Data Fig. 5c and Supplementary Fig. 1b). The training was performed on one 24 GB GPU for 9 hours with 7.5 seconds per epoch to train the model past convergence.

Encoders of paired scRNA-seq and scATAC-seq. We train two separate encoders for scRNA-seq and scATAC-seq with identical structures (Extended Data Fig. 5c). Each encoder starts with two linear layers with a dropout rate of 0.01, each of which has 1,024 dimensions and is followed by LeakyReLU activation and batch normalization, which is a standard set-up for autoencoders[23,58]. After the first two hidden layers, two separate linear layers with LeakyReLU activation are used to obtain separate hidden layers for the shared and modality-specific latent spaces. A linear layer is applied to each hidden layer to obtain the shared or modality-specific latent space. The inferred latent spaces from the autoencoder are compared with the latent spaces learned in step 1 through MSE loss. The MSE loss is minimized using the ADAM optimizer with a learning rate of 0.0001, the same learning rate as the decoder.

Application to paired scRNA-seq and protein abundance. The same set-up is applied to the paired scRNA-seq and protein abundance data measured by CITE-seq[25], with dimension 50 for the shared latent space and dimension 30 for the modality-specific latent space. Given that cross-modality prediction is not the goal in this application and to demonstrate robustness of our approach, the model is trained without the two decoders D_1' and D_2' that map from the shared latent space to each of the modalities (Extended Data Fig. 6a).

Pre-processing of scRNA-seq and scATAC-seq data

The paired scRNA-seq and scATAC-seq datasets from ref. 5 are separately filtered for genes or peaks that are non-zero in at least 300 cells and then filtered for cells with at least 300 non-zero genes or peaks. The common cells between the two modalities are then selected for another round of filtering with the same criteria. These two rounds of filtering result in 28,098 cells with 9,153 genes in the scRNA-seq data, and 58,170 peaks in the scATAC-seq data. The data are then log-transformed and min–max scaled, such that the minimum count in each cell is 0 and the maximum count in each cell is 1. Our filtering and normalization steps follow standard procedures of preprocessing scRNA-seq and scATAC-seq data (see, for example, refs. 5,22,23,42).

Preprocessing of scRNA-seq and cellular surface protein abundance data

Following the approach taken by a previous study that analyzes this dataset[25], we use the top 4,005 highly variable genes in the scRNA-seq data and all 110 proteins. Same as in the preprocessing of scRNA-seq and scATAC-seq described above, the data are log-transformed and min–max scaled, such that the minimum count in each cell is 0 and the maximum count in each cell is 1.

Alternative models for scRNA-seq and cellular surface protein abundance data

We use the default parameter setting for the Seurat WNN method[17]. For testing the standard multi-modal autoencoder method, we use the same encoder and decoder architectures as the full APOLLO model with

80 latent dimensions, which is equal to the full latent space dimension of the APOLLO model. Instead of performing a two-step training, this model trains the encoder and decoder together in a single step, without directly updating the latent spaces as parameters for optimization. MSE loss is added between the latent spaces obtained from the two encoders of the two input modalities. Similar to the APOLLO model that adds Gaussian noise in the latent space, this model performs a variational sampling step as in a standard variational autoencoder and the resulting latent spaces are passed to the decoder for reconstruction. In each epoch, we alternate between training the scRNA-seq autoencoder and training the protein autoencoder.

Cell-type prediction based on the inferred latent spaces of scRNA-seq and scATAC-seq

We train four separate neural network classifiers with the same architecture to predict cell types based on one of the following inputs: (1) shared latent space inferred from scRNA-seq data using the RNA encoder; (2) shared latent space inferred from scATAC-seq using the ATAC encoder; (3) both shared and modality-specific latent spaces inferred from scRNA-seq data using the RNA encoder; and (4) both shared and modality-specific latent spaces inferred from scATAC-seq data using the ATAC encoder. We use a standard feedforward neural network architecture in our classifiers; see, for example, ref. 19. Each classifier consists of 4 layers and outputs the probability of a given cell being assigned to each of the 23 cell types. The first three layers are followed by LeakyReLU activation and batch normalization. All 4 layers have a dropout rate of 0.1 and a hidden dimension of 128. We use the ADAM optimizer with a learning rate of 0.00001 to minimize the cross-entropy loss between our prediction and the cell-type labels assigned by ref. 5. We use the same train–validation split to train the classifiers as in the training of the APOLLO model, which means the same 85% of cells are used to train the APOLLO model and the cell-type classifiers.

Model architecture and training of paired chromatin and protein imaging

For both training steps, we hold-out all images from one patient in each of the four phenotypes for testing the model (Supplementary Fig. 4).

Step 1 latent optimization. In this step, we train the latent spaces, protein IDs, and the corresponding decoders to reconstruct the chromatin and protein images (Extended Data Fig. 8a and Supplementary Fig. 5a). The training was performed on one 24 GB GPU with 40 seconds per epoch.

Latent spaces and protein IDs of paired chromatin and protein imaging. For each protein, we randomly initialize a trainable 64-dimensional vector of protein ID that is shared across all images of that protein by using 'torch.nn.Embedding'. The protein IDs are used in both the encoding and decoding steps by concatenating them to the latent space embeddings, similar to a conditional autoencoder model (Extended Data Fig. 8). The shared latent space has 1,024 dimensions and each of the 2modality-specific latent spaces has 200 dimensions except when different latent space sizes are tested (Fig. 3b). Similar to the application to paired scRNA-seq and scATAC-seq, the shared latent space is chosen to be much larger than the modality-specific latent spaces to ensure that the shared space has enough capacity to contain all the shared information. Latent spaces are initialized using 'torch.nn.Embedding' with the default parameters. Independent Gaussian noise with zero mean and unit variance is added to the latent spaces at each training epoch. The hyperparameter λ in equation (1) for the ℓ_2 regularization of the latent spaces is set to 0.001 and the learning rate of the ADAM optimizer is set to 0.001, which are the same hyperparameter values as in ref. 37.

Decoders of paired chromatin and protein imaging. Four decoders are trained in step 1: (1) a decoder that reconstructs chromatin images from the shared latent space; (2) a decoder that reconstructs chromatin images from the full chromatin latent space (the shared latent space embedding concatenated with the chromatin-specific latent space embedding); (3) a decoder that reconstructs protein images from the shared latent space; and (4) a decoder that reconstructs protein images from the full protein latent space (the shared latent space embedding concatenated with the protein-specific latent space embedding). The architectures of the decoders are shown in Extended Data Fig. 8a. The latent space embedding, after adding noise and being concatenated with the protein ID, is passed through a linear layer with ReLU activation and reshaped to 4 × 4 × 96 dimensions for subsequent convolutions. This is followed by 5 convolutional layers with a kernel size of 4 and stride of 2. The number of channels in each hidden layer is listed in Extended Data Fig. 8a. The first four convolutional layers are followed by batch normalization and LeakyReLU activation. The last convolutional layer is followed by sigmoid activation to scale the output image from 0 to 1. For both imaging modalities, we use binary cross-entropy loss to minimize the objective function defined in equation (1), which is calculated using 'torch.nn.BCEWithLogitsLoss' based on the output before the sigmoid activation. The learning rate of the decoders is 0.0001 and ADAM is used for optimization. The decoder architectures and training procedure follow standard set-ups for autoencoders (see, for example, refs. 23,37,45).

Step 2 inference. In this step, we train the encoders to infer the learned latent spaces from the chromatin and protein images without updating the latent spaces, protein IDs or the decoders (Extended Data Fig. 8b and Supplementary Fig. 5b). The training was performed on one 24 GB GPU with 20 seconds per epoch.

Encoders of paired chromatin and protein imaging. We train two separate encoders for chromatin and protein images with identical structures (Extended Data Fig. 8b). Each encoder starts with five convolutional layers with LeakyReLU activation, a standard set-up for autoencoders[23,37,45]. The dimensions of the hidden layers are listed in Extended Data Fig. 8b. The output of the last convolutional layer is divided into two sets of channels that are used to derive the shared and modality-specific latent spaces respectively. Eighty out of the 96 channels of the last hidden layer are flattened, concatenated with protein ID, and passed through a linear layer to obtain the shared latent space. Similarly, the modality-specific latent space is obtained from the remaining 16 channels. The inferred latent spaces from the autoencoder are compared with the latent spaces learned in step 1 through MSE loss. The MSE loss is minimized using the ADAM optimizer with a learning rate of 0.001.

Cross-modality predictions. To predict unmeasured proteins, the shared latent space is first inferred from the chromatin image of a cell using the chromatin encoder. Then each protein image can be predicted by decoding the inferred shared latent space through the protein decoder using the protein ID of the target protein (Extended Data Fig. 8a).

Application to HPA data. The same set-up is applied to the HPA data[35], with a shared latent space dimension of 1,024 and a modality-specific latent space dimension of 200. The two decoders D_1' and D_2' that map from the shared latent space to each of the modalities are not used, as the omission does not impact cross-modality-prediction performance (Fig. 3b) and results in correct disentanglement of the shared and modality-specific information (Fig. 2f).

Alternative models. The models used for benchmarking our full APOLLO model are described below. In addition to using binary cross entropy (BCE) loss for the decoder outputs, we also tested the use of an alternative loss function, namely, the MSE loss. All model training and testing use the same train–test split. For benchmarking cross-modality prediction, the prediction results are first thresholded using mode intensity and then compared using ℓ_1 loss, which is not used for training any of the models.

Standard autoencoder training using a single step. This model has the same encoder and decoder architectures as the full APOLLO model, and the same dimensions of the shared, chromatin-specific and protein-specific latent spaces are used. Protein IDs are concatenated to the hidden layers during encoding and decoding as described in the full model. However, instead of performing a two-step training, this model trains the encoder and decoder together in a single step, without directly updating the latent spaces as parameters for optimization (Extended Data Fig. 10a). MSE loss is added between the shared latent spaces obtained from the two encoders of the two input modalities. Similar to the APOLLO model that adds Gaussian noise in the latent space, this model performs a variational sampling step as in a standard variational autoencoder and the resulting latent spaces are passed to the decoder for reconstruction. In each epoch, we alternate between training the chromatin autoencoder and training the protein autoencoder.

Our model without modality-specific latent space. This model has the same two-step training procedure as the full APOLLO model, but does not separate the shared and the modality-specific latent spaces (Extended Data Fig. 10b). There are only two decoders that decode the two modalities from the shared latent space and they are trained in step 1. The encoders only output the shared latent space, which has the same dimension as the combined dimension of the shared and the modality-specific latent spaces in the full APOLLO model.

Inpainting model. For the inpainting model developed in ref. 45, we adapted the original code from TensorFlow to PyTorch. We used the same hyperparameter values as in the original publication, including the use of MSE loss, except the slight modification to adapt to the single-channel input image without a microtubule channel.

Data simulation and assessment of APOLLO's disentanglement performance

To systematically evaluate APOLLO's ability to disentangle shared and modality-specific representations, we generate simulated multi-modal datasets with ground-truth latent structure. Given the recent theoretical results on identifiability of the shared and modality-specific variables[36], we only allowed directed edges from the shared latent variables to the modality-specific latent variables but not vice versa. For all simulations, we assign the latent variables Z_1 and Z_2 to the shared latent space, Z_3 and Z_4 to the modality 1 specific latent space, and Z_5 to the modality 2 specific latent space. The latent causal graph and the corresponding probability distribution are specified in panel a of Extended Data Figs. 1–4, from which 2,000 samples of latent features are independently drawn for each simulation and visualized in panel b of Extended Data Figs. 1–4. Given the causal graphs, we expect an accurate disentanglement to identify that the modality 1 specific latent space also captures Z_2 in simulations 3–5 but not in simulations 1 and 2. Each observed feature is a linear combination of one or more latent features with the coefficients independently drawn from a standard normal distribution. In simulations 1 and 3, where each observed feature depends on just 1 latent feature, 10 observed features per modality are generated from each latent feature, resulting in a feature dimension of 40 in modality 1 and 30 in modality 2. In simulations 2, 4 and 5, where we include observed features with multiple parents, 10 observed features in modality 1 are generated from: (1) each one of Z_2, Z_3 and Z_4 (30 observed features total); (2) each pair of Z_2, Z_3, and Z_4 (30 observed features total); (3) Z_2, Z_3, and Z_4; and (4) all four modality 1 latent features. Similarly, a total of 40 observed features in modality 2 are generated from: (1) each one of Z_2 and Z_5 (20 observed features total); (2) Z_2 and Z_5 (10 observed features); and (3) all three modality 2 latent features. This results in a feature dimension of 40 in modality 1 and 30 in modality 2. In simulation 5, we tested the effects of higher weights for the observed features that depend only on one latent feature by using the same set-up as simulation 4 while drawing coefficients of the observed features from $\mathcal{N}(0,16)$. For APOLLO training, we use fully connected decoders with a similar architecture as in the paired-sequencing applications and an MSE loss. Each of the shared and modality-specific latent spaces has two dimensions. Accuracy in disentanglement is assessed by the separation of the binary variables Z_1 and Z_3 in the three latent spaces, which we quantify using silhouette scores[59] and UMAP visualizations (Extended Data Figs. 1–4).

Pre-processing of paired chromatin and protein imaging

The imaging data are obtained from ref. 43, where the patients are annotated with one of the following four phenotypic classes: healthy, meningioma, glioma, or head and neck tumor (Supplementary Fig. 4). Each single-cell image of each protein stain and chromatin stain is min–max scaled such that the minimum pixel value is 0 and the maximum is 1, a standard image normalization step for neural networks[43,45]. For each cell nucleus, an image patch centered at the centroid of the nucleus of dimension 128 × 128 pixels is cropped from the whole image. A total of 29,174 cell images were used in training.

Pre-processing of the HPA images

The same procedure as in a previous study[32] is used to pre-process the images. We use all images of U2OS cells (the cell line with the most data in the HPA) that are also used in ref. 32 for studying protein localization variabiltiy (2,973 proteins used) and exclude proteins that are stained in less than 150 cells, resulting in a total of 141 proteins used for training the model. The subset of proteins has been shown in ref. 32 to have diverse subcellular localizations. Nuclear segmentation obtained using StarDist[60] is used for computing the intranuclear proportion of the protein in each image. The standard deviation of the intranuclear proportion of each protein is computed across all images stained for the protein. The 25 proteins with the largest standard deviations are used for interpreting the disentanglement results (Fig. 5).

Phenotype classification using real or reconstructed protein and chromatin images

We use the ResNet-18 model[61], a popular neural network model for image classification, in PyTorch for the phenotype classifiers to predict phenotypes from real protein/chromatin images, reconstructed protein/chromatin images, or protein images predicted from chromatin (Fig. 3d). The first layer of the ResNet-18 model is adjusted to take single-channel input images. We use the cross-entropy loss and set the weight of each phenotype class to be inversely proportional to the fraction of cells in that particular phenotype class. The classifiers are trained using the ADAM optimizer with a learning rate of 0.001. The training of each classifier is repeated 36 times with 6 different groups of held-out patients, where each held-out group contains one patient from each phenotype class. One-sided two-sample *t*-test and Wilcoxon signed-rank test are used for the null hypothesis that the phenotype prediction accuracy using the reconstruction from the full latent space is smaller than or equal to the accuracy using the reconstruction from the shared latent space.

Interpretation of the partially shared latent spaces

In the following, we describe a procedure to analyze the shared latent space as well as the modality-specific latent space, which can be applied to any data modality. We compute the PCs of the latent spaces and group the cells based on their positions along the PCs. For both of our applications, we group cells into 11 bins with equal percentile ranges along each PC. In the following description of our approach, we denote the percentile bin of PC_i that is centered around 0 as PC_i^0 and the bins ordered from negative to positive PC values are denoted as $PC_i^{-5}, \cdots PC_i^{-1}, PC_i^0, PC_i^1, \cdots PC_i^5$. When comparing cells along PC_i we only use the cells at the center of all other PCs, meaning the 15% of cells with the smallest Euclidean distance to the PC origin calculated using all PC_j for $j \neq i$ and $j < 10$. The number of PCs considered in the analysis can be adjusted given the variance explained by each PC in a particular dataset. Using cells at the center of all other PCs, except the PC being interpreted, ensures that we are only considering the variation of cells along one PC to tease apart the variations explained by the different PCs. In the following, we explain this procedure in more detail in the context of the paired-sequencing and paired-imaging datasets, where we also perform subsampling for a more robust identification of the features explained by each PC.

Interpretation of the latent spaces of paired scRNA-seq and scATAC-seq. We test for significant gene expression and peak count changes along each PC. The cells are divided into 36 random train–test splits with 20% of cells used for testing in each split, to validate the significant genes and peaks identified along each PC. We consider three groups of cells for each PC_i that are all at the center of other PCs: (1) cells in PC_i^{-5} and PC_i^{-4}; (2) cells in PC_i^0; and (3) cells in PC_i^4 and PC_i^5. Cells in each group are compared with all cells in the other two groups through *t*-tests using Scanpy's 'rank_genes_groups' function[42]. The threshold of *P* values after multiple testing correction is set to 0.05 for both modalities. The number of times in all train–test splits a gene or peak is tested to be significant in both the training and testing cells with the same direction of fold change are plotted for different thresholds of log fold change magnitude (Fig. 2c and Supplementary Figs. 2 and 3). The ATAC-seq peaks in the shared and modality-specific latent spaces are summarized in Fig. 2d. For each ATAC-seq peak, we counted the number of PCs in the shared and modality-specific latent spaces respectively, in which the peak is found to be significant. The genes are then grouped by their GO annotations and the total counts of PCs in the two latent spaces are normalized to 1.

Interpretation of the latent spaces of paired chromatin and protein imaging. Imaging data can be visually inspected by sampling cells at each percentile bin along a particular PC (Fig. 4a). In addition, for a more comprehensive assessment, predefined handcrafted image features[43,62] can be used to test for significant feature changes along each PC. For the following analysis, we use a total of 234 chromatin image features and around 30 image features per protein obtained from[43]. The patients are divided into 12 different train–test splits, which are summarized in Supplementary Fig. 4, to validate the significant features identified along each PC. As in the application to paired scRNA-seq and scATAC-seq, we test for significant feature changes among three groups of cells for each PC_i that are all at the center of other PCs: (1) cells in PC_i^{-5} and PC_i^{-4}; (2) cells in PC_i^0; and (3) cells in PC_i^4 and PC_i^5. A feature is considered significant if the *P* value after multiple testing correction is less than 0.05 and the absolute value of log fold change is greater than $\log_2(FC_t)$ in both the training and testing patients, where FC_t is set to be 1.2 for chromatin features and 1.4 for protein features. The fold-change thresholds are chosen so that most representative morphological features are represented by at least one PC. In addition, the direction of fold change is required to be the same in the training and testing patients. Figure 4a and Supplementary 19 show the chromatin features that are significant in at least 8 out of the 12 train–test splits (Supplementary Fig. 4). We summarize the morphological features in all top PCs in the shared and modality-specific latent spaces respectively in Fig. 4b, and Supplementary Figs. 9c, 10c, 11c and 14c. For this, we counted the number of PCs for which each feature is found to be significant in the shared and modality-specific latent spaces and normalized the total counts in the two latent spaces to 1.

To obtain a more concise representation, we group the morphological features of chromatin and each protein by Pearson correlation across all cells and plot one representative feature per group in Fig. 4a,b, and Supplementary Figs. 8–14. The feature groupings for each stain are obtained as follows: we build a network with the features as nodes and an edge is added between each pair of features whose Pearson correlation is greater than 0.7. Features within the same connected component of the graph are considered in the same group. The representative feature of a group is the node with the highest degree. This results in 18 feature groups for chromatin, 5 groups for γH2AX, 5 groups for lamin, 6 groups for CD8, 4 groups for CD4, 5 groups for CD3 and 4 groups for CD16.

Interpretation of the latent spaces of the HPA images. For each subset of U2OS cells stained with the same protein, we separately perform *k*-means ($k = 2$) clustering in the shared and the two modality-specific latent spaces of each model. The clustering in each latent space is repeated five times with different random seeds. For each random seed, a *t*-test is performed to compare the proportion of intranuclear protein localization in the two clusters, and the absolute value of the difference between the cluster means is computed. A clustering shows significant difference of intranuclear proportions if $P < 0.00022$ and the absolute difference between clusters >0.01 in all 5 random initializations. The *P*-value threshold is chosen by applying a Bonferroni correction to the overall probability of type 1 error at 0.05 for a total of 225 tests performed for 25 proteins in each of the three latent spaces of the three separate models.

γH2AX feature prediction using chromatin images

We use the ResNet-18 model[61] in PyTorch to train separate regression models that predict the predefined γH2AX features from chromatin images. Similar to the previous section, for each of the 3 γH2AX features (Fig. 4b), a regression model is trained 12 times with different train–test splits of the patients (Supplementary Fig. 4). The outputs of each regression model on the held-out patients are then compared with the ground truth using Pearson's correlation (Supplementary Fig. 13c).

Statistics and reproducibility

Publicly available datasets were used in this study. Standard filtering steps were performed, and otherwise, no data were excluded from our analyses. Randomization and blinding were not applicable in this study.

Reporting summary

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

Data availability

All datasets used in this study are publicly available. The SHARE-seq data from ref. 5 are available under accession number GEO GSE140203. The CITE-seq data from ref. 25 are available under accession number GEO GSE150599. The PBMCs multiplex imaging dataset from ref. 43 is available from the PSI Public Data Repository at https://doi.org/10.16907/b039dc4e-9366-413c-8f34-92ce9110cc14. The Human Protein Atlas images from ref. 35 are available at https://www.proteinatlas.org. Source data for Figs. 2 and 3b,c is available with this paper. Data in Figs. 3d, 4 and 5 can be reproduced using our deposited code and publicly available data.

Code availability

The code is available via Zenodo at https://doi.org/10.5281/zenodo.17841315 (ref. 63) and in via GitHub at https://github.com/uhlerlab/APOLLO/. For ease of use, a modularized version of our method, for each application, is available on GitHub with clear step-by-step instructions regarding data input and use of the method.

References

1. Tan, W. C. C. et al. Overview of multiplex immunohistochemistry/immunofluorescence techniques in the era of cancer immunotherapy. *Cancer Commun.* **40**, 135–153 (2020).
2. Lin, J.-R., Fallahi-Sichani, M., Chen, J.-Y. & Sorger, P. K. Cyclic immunofluorescence (CycIF), a highly multiplexed method for single-cell imaging. *Curr. Prot. Chem. Biol.* **8**, 251–264 (2016).
3. Kuett, L. et al. Three-dimensional imaging mass cytometry for highly multiplexed molecular and cellular mapping of tissues and the tumor microenvironment. *Nat. Cancer* **3**, 122–133 (2022).
4. Altemose, N. et al. EQ DamID: a microfluidic approach for joint imaging and sequencing of protein–DNA interactions in single cells. *Cell Syst.* **11**, 354–3669 (2020).
5. Ma, S. et al. Chromatin potential identified by shared single-cell profiling of RNA and chromatin. *Cell* **183**, 1103–111620 (2020).
6. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
7. Zeng, H. et al. Integrative in situ mapping of single-cell transcriptional states and tissue histopathology in a mouse model of Alzheimer's disease. *Nat. Neurosci.* **26**, 430–446 (2023).
8. Xia, C., Fan, J., Emanuel, G., Hao, J. & Zhuang, X. Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression. *Proc. Natl Acad. Sci. USA* **116**, 19490–19499 (2019).
9. Eng, C.-H. L. et al. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature* **568**, 235–239 (2019).
10. Laber, S. et al. Discovering cellular programs of intrinsic and extrinsic drivers of metabolic traits using LipocyteProfiler. *Cell Genom.* **3**, 100346 (2023).
11. Gower, J. C. Generalized procrustes analysis. *Psychometrika* **40**, 33–51 (1975).
12. Stuart, T. et al. Comprehensive Integration of single-cell data. *Cell* **177**, 1888–190221 (2019).
13. Argelaguet, R. et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology* **21**, 111 (2020).
14. Amodio, M. & Krishnaswamy, S. MAGAN: aligning biological manifolds. *Proc. 35th International Conference on Machine Learning*, Vol. 80 (eds Dy, J. & Krause, A.) 215–223 (PMLR, 2018).
15. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision (ICCV)* 2242–2251 (2017).
16. Klein, D. et al. Mapping cells through time and space with moscot. *Nature* **638**, 1065–1075 (2025).
17. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–358729 (2021).
18. Jain, M. S. et al. MultiMAP: dimensionality reduction and integration of multimodal data. *Genome Biol.* **22**, 346 (2021).
19. Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning (MIT Press, 2016); https://www.deeplearningbook.org/
20. Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics* **36**, 3418–3421 (2020).
21. Brbifá, M. et al. Annotation of spatially resolved single-cell data with STELLAR. *Nat. Methods* **19**, 1411–1418 (2022).
22. 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).
23. Zhang, X., Wang, X., Shivashankar, G. V. & Uhler, C. Graph-based autoencoder integrates spatial transcriptomics with chromatin images and identifies joint biomarkers for Alzheimer's disease. *Nat. Commun.* **13**, 7480 (2022).
24. Yang, K. D. et al. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *Nat. Commun.* **12**, 31 (2021).
25. Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nat. Methods* **18**, 272–282 (2021).
26. Ashuach, T. et al. MultiVI: deep generative model for the integration of multimodal data. *Nat. Methods* **20**, 1222–1231 (2023).
27. Gong, B., Zhou, Y. & Purdom, E. Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome Biol.* **22**, 351 (2021).
28. Zhang, Z., Yang, C. & Zhang, X. scDART: integrating unmatched scRNA-seq and scATAC-seq data and learning cross-modality relationship simultaneously. *Genome Biol.* **23**, 139 (2022).
29. Boehm, K. M., Khosravi, P., Vanguri, R., Gao, J. & Shah, S. P. Harnessing multimodal data integration to advance precision oncology. *Nat. Rev. Cancer* **22**, 114–126 (2022).
30. Cao, Z.-J. & Gao, G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nat. Biotechnol.* **40**, 1458–1466 (2022).
31. Zhang, X. et al. Unsupervised representation learning of chromatin images identifies changes in cell state and tissue organization in DCIS. *Nat. Commun.* **15**, 6112 (2024).
32. Zhang, X., Tseo, Y., Bai, Y., Chen, F. & Uhler, C. Prediction of protein subcellular localization in single cells. *Nat. Methods* **22**, 1265–1275 (2025).
33. Sudlow, C. et al. UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Med.* **12**, 1001779 (2015).
34. All of Us Research Program Investigators. The 'All of Us' Research Program. *N. Engl. J. Med.* **381**, 668–676 (2019).
35. Thul, P. J. et al. A subcellular map of the human proteome. *Science* **356**, eaal3321 (2017).
36. Sturma, N., Squires, C., Drton, M. & Uhler, C. Unpaired multi-domain causal representation learning. *37th Conference on Neural Information Processing Systems (NeurIPS)* (eds Oh, A. et al.) 34465–34492 (Curran Associates, 2023).
37. Gabbay, A. & Hoshen, Y. Demystifying inter-class disentanglement. *International Conference on Learning Representations (ICLR)* (2020).
38. Giotti, B. et al. Assembly of a parts list of the human mitotic cell cycle machinery. *J. Mol. Cell Biol.* **11**, 703–718 (2019).
39. Sansam, C. L. et al. A vertebrate gene, *ticrr*, is an essential checkpoint and replication regulator. *Genes Dev.* **24**, 183–194 (2010).
40. Chen, F., Castranova, V. & Shi, X. New insights into the role of nuclear factor-kappaB in cell growth regulation. *Am. J. Pathol.* **159**, 387–397 (2001).
41. Clermont, P.-L. et al. Identification of the epigenetic reader CBX2 as a potential drug target in advanced prostate cancer. *Clin. Epigenetics* **8**, 16 (2016).
42. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology* **19**, 15 (2018).
43. Challa, K. et al. Imaging and AI based chromatin biomarkers for diagnosis and therapy evaluation from liquid biopsies. *npj Precis. Oncol.* **7**, 135 (2023).
44. Cho, N. H. et al. OpenCell: endogenous tagging for the cartography of human cellular organization. *Science* **375**, 6983 (2022).
45. Lu, A. X., Kraus, O. Z., Cooper, S. & Moses, A. M. Learning unsupervised feature representations for single cell microscopy images with paired cell inpainting. *PLoS Comput. Biol.* **15**, 1007348 (2019).
46. Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F. & Johnson, G. R. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nat. Methods* **15**, 917–920 (2018).

47. Lee, J.-H., Kim, E. W., Croteau, D. L. & Bohr, V. A. Heterochromatin: an epigenetic point of view in aging. *Exp. Mol. Med.* **52**, 1466–1474 (2020).
48. Viana, M. P. et al. Integrated intracellular organization and its variations in human iPS cells. *Nature* **613**, 345–354 (2023).
49. Handfield, L.-F., Chong, Y. T., Simmons, J., Andrews, B. J. & Moses, A. M. Unsupervised clustering of subcellular protein expression patterns in high-throughput microscopy images reveals protein complexes and functional relationships between proteins. *PLoS Comput. Biol.* **9**, 1003085 (2013).
50. Kobayashi, H., Cheveralls, K. C., Leonetti, M. D. & Royer, L. A. Self-supervised deep learning encodes high-resolution features of protein subcellular localization. *Nat. Methods* **19**, 995–1003 (2022).
51. Donovan-Maiye, R. M. et al. A deep generative model of 3D single-cell organization. *PLoS Comput. Biol.* **18**, 1009155 (2022).
52. Li, J. et al. DNA damage binding protein component ddb1 participates in nucleotide excision repair through DDB2 DNA-binding and cullin 4A ubiquitin ligase activity. *Cancer Res.* **66**, 8590–8597 (2006).
53. Guderian, G. et al. RioK1, a new interactor of protein arginine methyltransferase 5 (PRMT5), competes with pICln for binding and modulates PRMT5 complex composition and substrate specificity. *J. Biol. Chem.* **286**, 1976–1986 (2011).
54. Chari, A. et al. An assembly chaperone collaborates with the SMN complex to generate spliceosomal SnRNPs. *Cell* **135**, 497–509 (2008).
55. Krapivinsky, G., Pu, W., Wickman, K., Krapivinsky, L. & Clapham, D. E. pICln binds to a mammalian homolog of a yeast protein involved in regulation of cell morphology. *J. Biol. Chem.* **273**, 10811–10814 (1998).
56. Badertscher, L. et al. Genome-wide RNAi screening identifies protein modules required for 40S subunit synthesis in human cells. *Cell Rep.* **13**, 2879–2891 (2015).
57. Radhakrishnan, A. et al. Cross-modal autoencoder framework learns holistic representations of cardiovascular state. *Nat. Commun.* **14**, 2436 (2023).
58. 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).
59. Rousseeuw, P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
60. Weigert, M., Schmidt, U., Haase, R., Sugawara, K. & Myers, G. Star-convex polyhedra for 3D object detection and segmentation in microscopy. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* 3655–3662 (IEEE, 2020); https://doi.org/10.1109/WACV45572.2020.9093435
61. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770–778 (2016).
62. Stirling, D. R. et al. CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics* **22**, 433 (2021).
63. Zhang, X. uhlerlab/APOLLO: manuscript partially shared multi-modal embedding learns holistic representation of cell state. *Zenodo* https://doi.org/10.5281/zenodo.17841315 (2025).

Acknowledgements

We thank all members of the Uhler and Shivashankar labs for discussions and E. Forte for editing the manuscript. X.Z. was supported by the Eric and Wendy Schmidt Center at the Broad Institute. G.V.S. was partially supported by the Swiss National Foundation (310030 208046). C.U. was partially supported by NCCIH/NIH (1DP2AT012345), ONR (N00014-22-1-2116), AstraZeneca, the MIT-IBM Watson AI Lab, MIT J-Clinic for Machine Learning and Health, and a Simons Investigator Award.

Author contributions

X.Z. designed the research, developed and implemented the algorithms, performed model and data analysis, and wrote the paper. G.V.S. and C.U. designed and supervised the research, and wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s43588-025-00948-w.

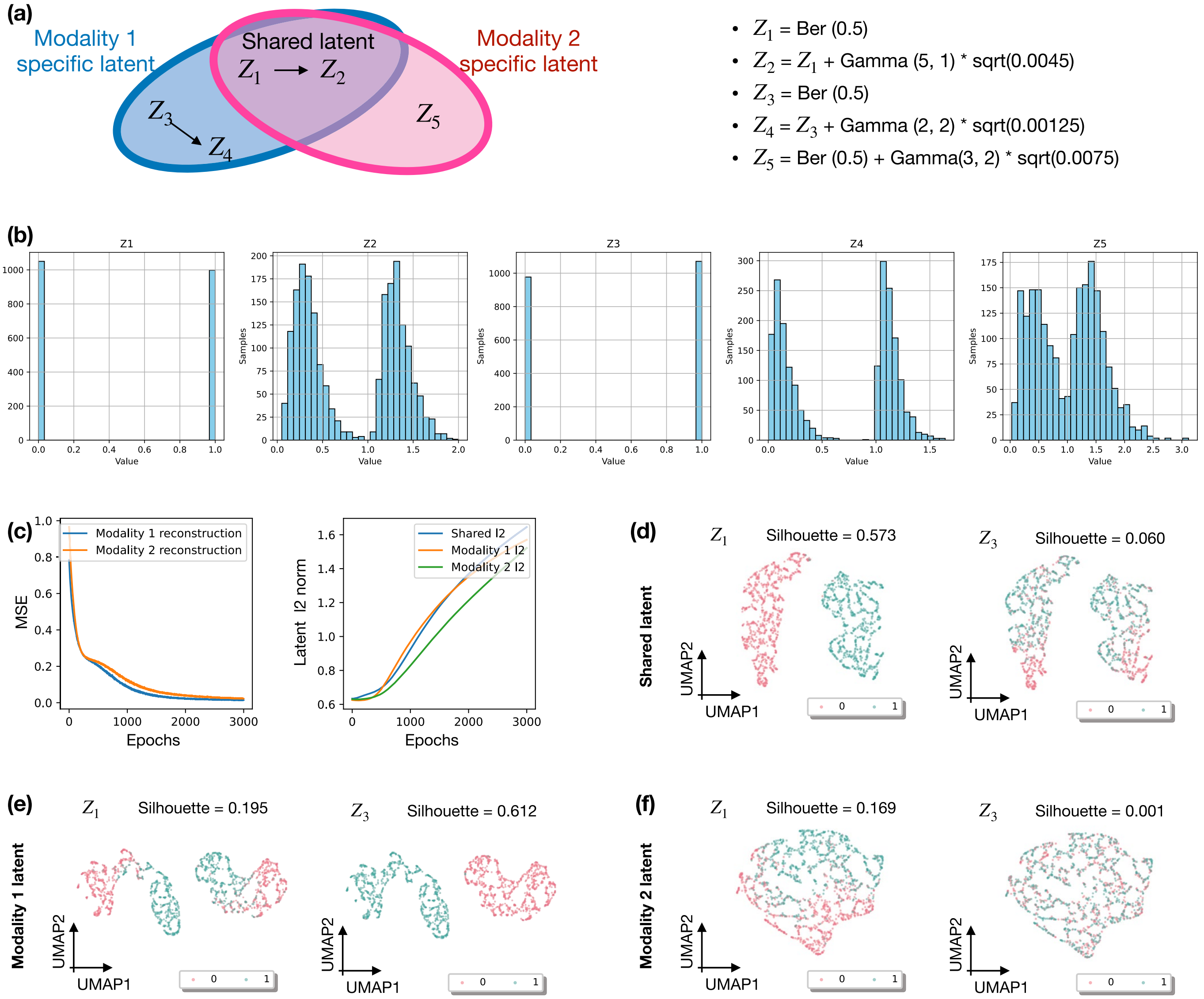
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-025-00948-w.

Correspondence and requests for materials should be addressed to G. V. Shivashankar or Caroline Uhler.

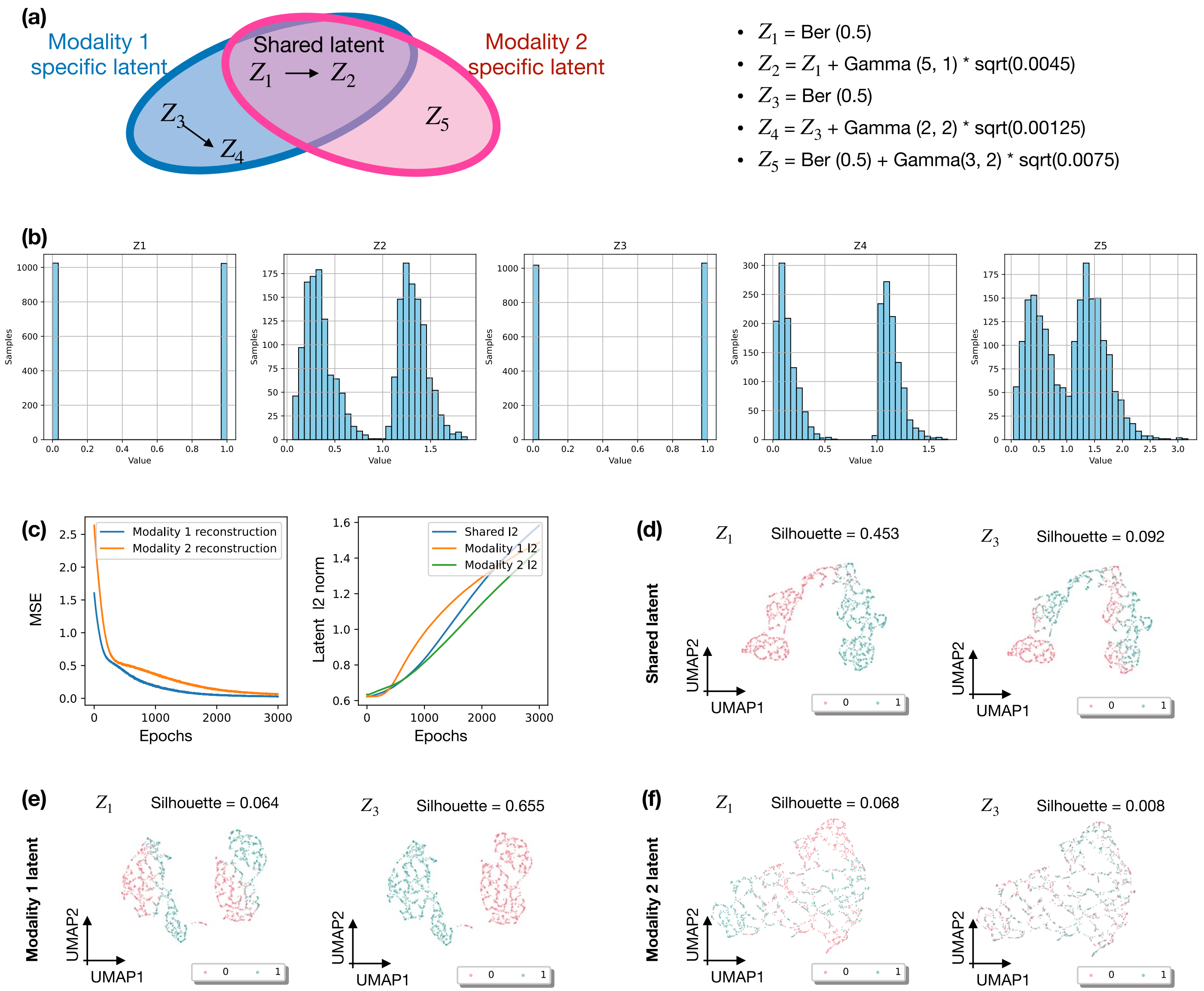
Peer review information *Nature Computational Science* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editor: Michelle Badri, in collaboration with the *Nature Computational Science* team. Peer reviewer reports are available.

Reprints and permissions information is available at www.nature.com/reprints.

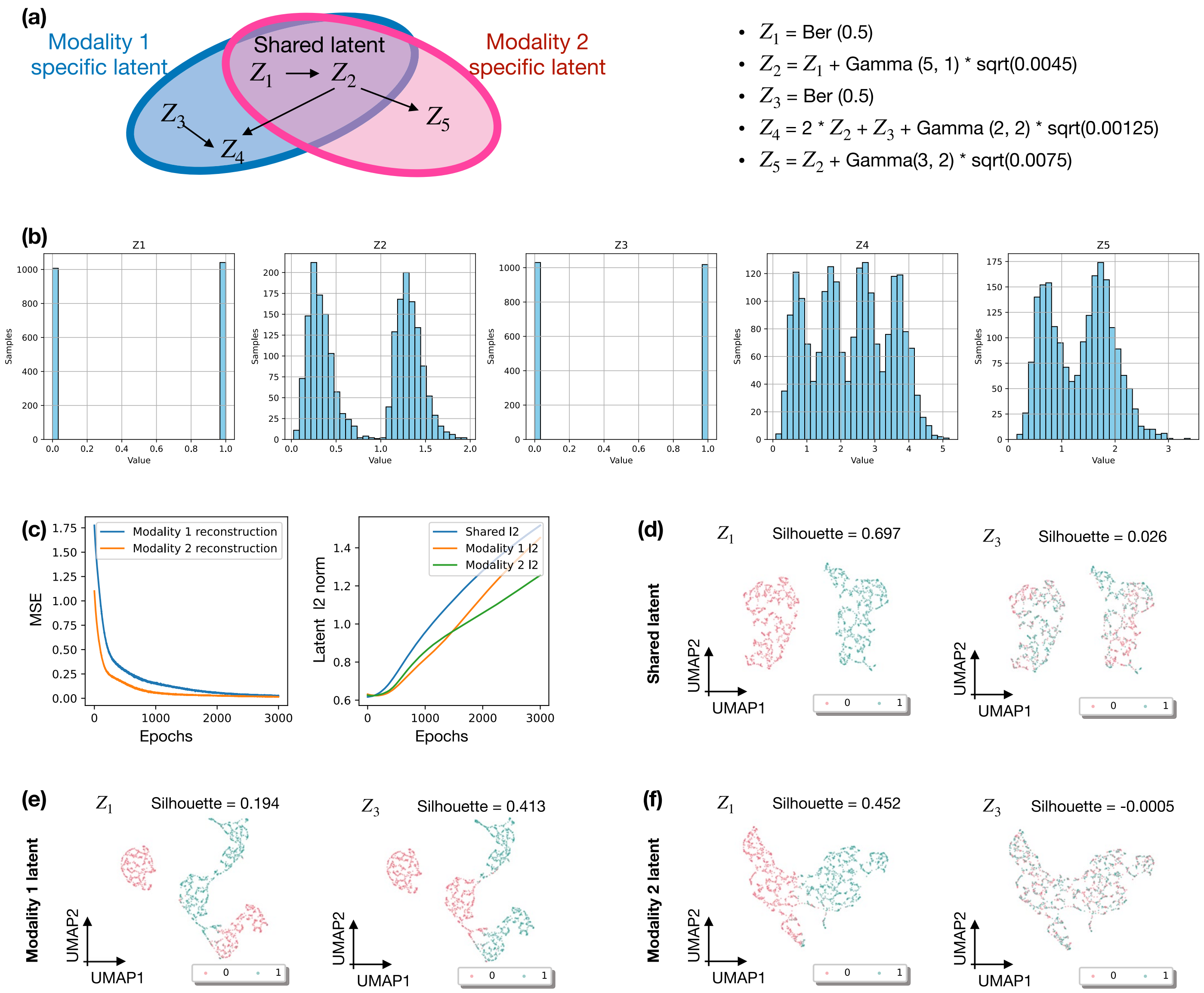
Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



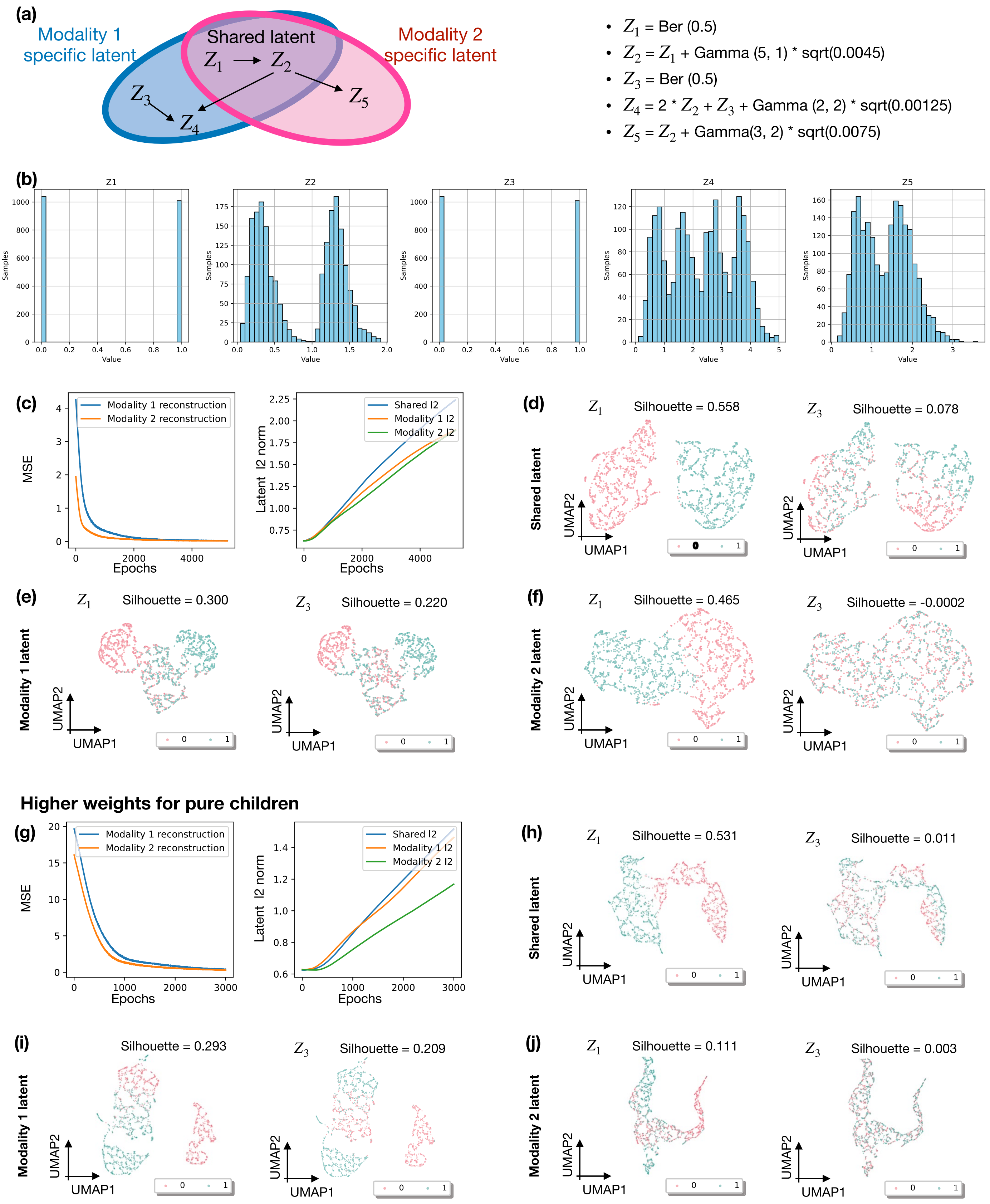
Extended Data Fig. 1 | Simulation 1 - All shared latent features are independent of all modality-specific latent features; each observed feature depends on a single latent feature. (**a**) Left: Latent causal graph of the two modalities. Right: Probability distributions from which the latent features are sampled. (**b**) Histograms of the 2000 samples of each latent feature (Z_1, Z_2, Z_3, Z_4, Z_5). (**c**) Training curves of the APOLLO model. Left: Reconstruction losses of the two modalities measured by mean squared error (MSE). Right: ℓ_2 norms of the three latent spaces. (**d**) UMAPs of the shared latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the shared latent features and Z_1 or Z_3 as labels. (**e**) UMAPs of the Modality 1 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 1 specific latent features and Z_1 or Z_3 as labels. (**f**) UMAPs of the Modality 2 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 2 specific latent features and Z_1 or Z_3 as labels.



Extended Data Fig. 2 | Simulation 2 - All shared latent features are independent of all modality-specific latent features; each observed feature depends on multiple latent features. (**a**) Left: Latent causal graph of the two modalities. Right: Probability distributions from which the latent features are sampled. (**b**) Histograms of the 2000 samples of each latent feature (Z_1, Z_2, Z_3, Z_4, Z_5). (**c**) Training curves of the APOLLO model. Left: Reconstruction losses of the two modalities measured by mean squared error (MSE). Right: ℓ_2 norms of the three latent spaces. (**d**) UMAPs of the shared latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the shared latent features and Z_1 or Z_3 as labels. (**e**) UMAPs of the Modality 1 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 1 specific latent features and Z_1 or Z_3 as labels. (**f**) UMAPs of the Modality 2 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 2 specific latent features and Z_1 or Z_3 as labels.

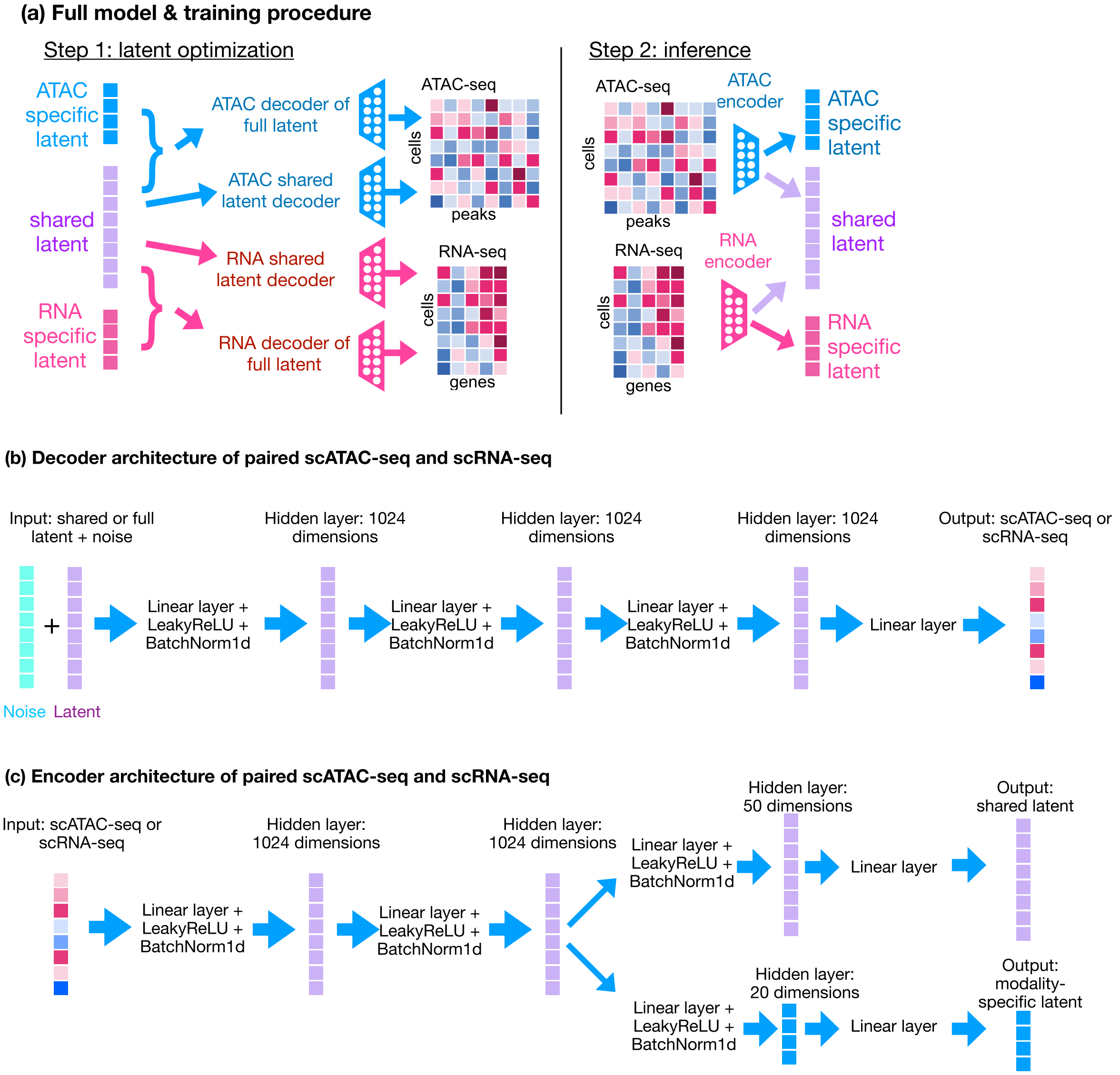


Extended Data Fig. 3 | Simulation 3 - Some modality-specific latent features are children of shared latent features; each observed feature depends on a single latent feature. (**a**) Left: Latent causal graph of the two modalities. Right: Probability distributions from which the latent features are sampled. (**b**) Histograms of the 2000 samples of each latent feature (Z_1, Z_2, Z_3, Z_4, Z_5). (**c**) Training curves of the APOLLO model. Left: Reconstruction losses of the two modalities measured by mean squared error (MSE). Right: ℓ_2 norms of the three latent spaces. (**d**) UMAPs of the shared latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the shared latent features and Z_1 or Z_3 as labels. (**e**) UMAPs of the Modality 1 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 1 specific latent features and Z_1 or Z_3 as labels. (**f**) UMAPs of the Modality 2 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 2 specific latent features and Z_1 or Z_3 as labels.

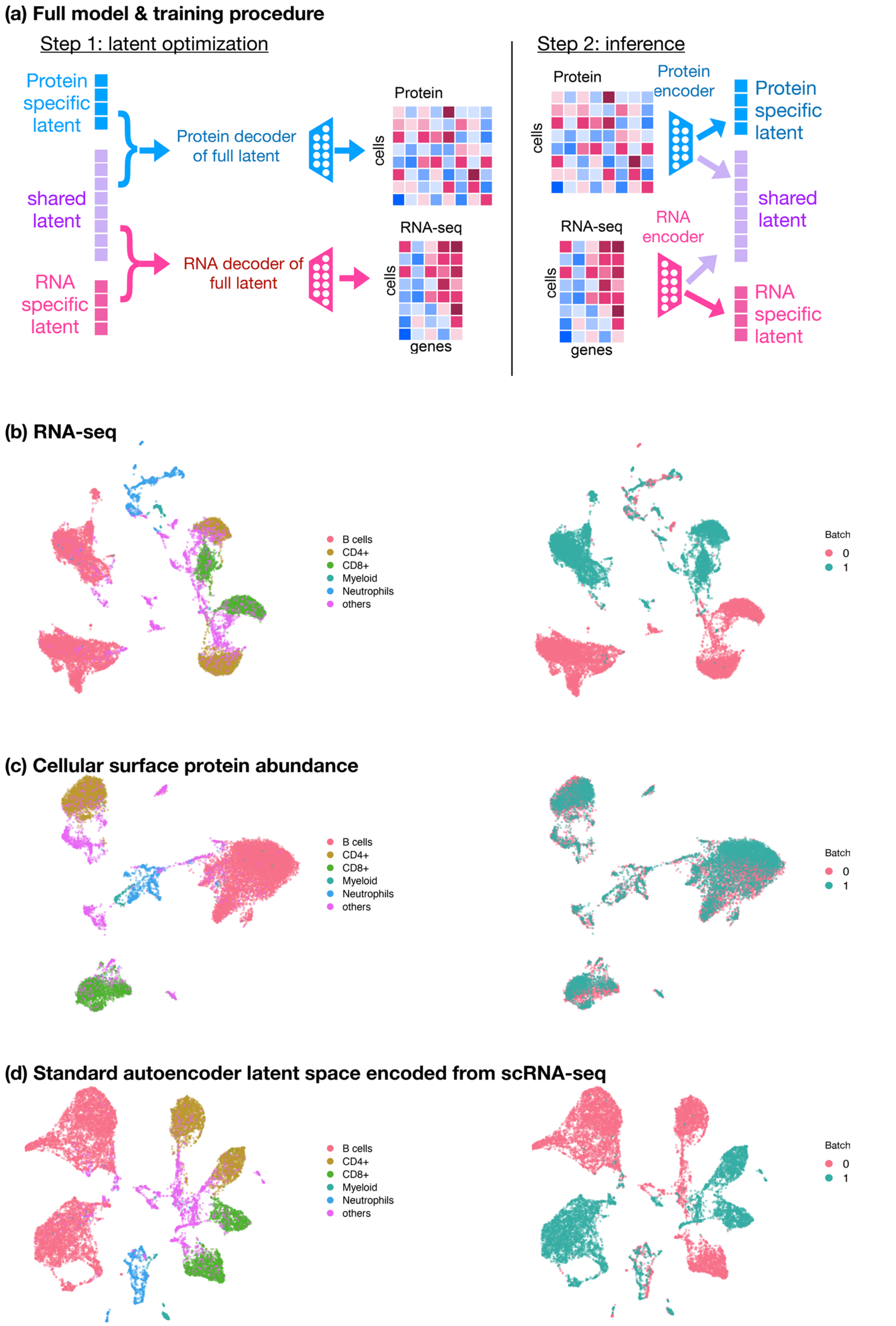


Extended Data Fig. 4 | See next page for caption.

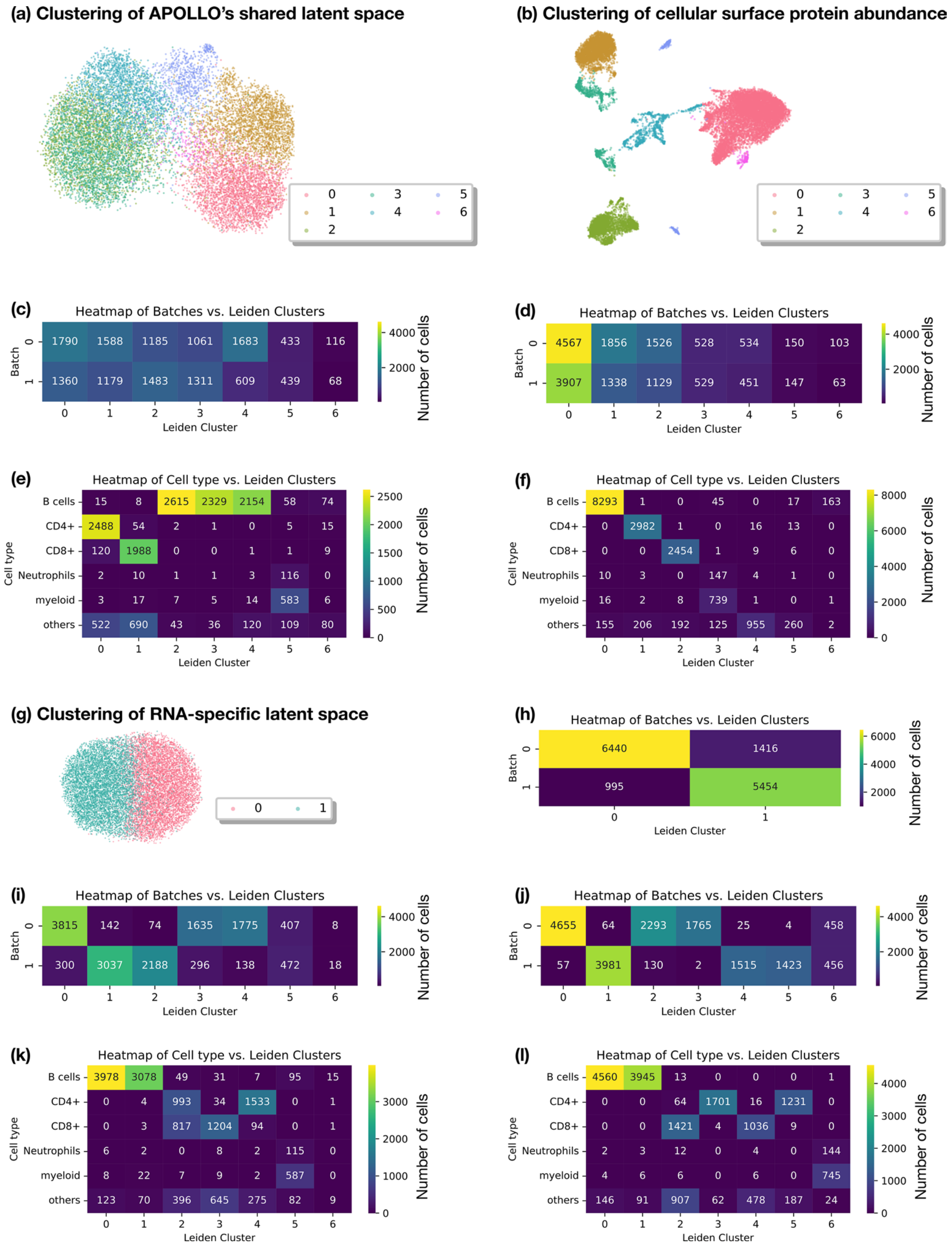
Extended Data Fig. 4 | Simulations 4 and 5 - Some modality-specific latent features are children of shared latent features; each observed feature depends on multiple latent features. (**a**) Left: Latent causal graph of the two modalities. Right: Probability distributions from which the latent features are sampled. (**b**) Histograms of the 2000 samples of each latent feature (Z_1, Z_2, Z_3, Z_4, Z_5). (**c**) Training curves of the APOLLO model trained on simulation 4 data. Left: Reconstruction losses of the two modalities measured by mean squared error (MSE). Right: ℓ_2 norms of the three latent spaces. (**d**) UMAPs of the shared latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the shared latent features and Z_1 or Z_3 as labels. (**e**) UMAPs of the Modality 1 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 1 specific latent features and Z_1 or Z_3 as labels. (**f**) UMAPs of the Modality 2 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 2 specific latent features and Z_1 or Z_3 as labels. (**g**) Training curves of the APOLLO model trained on simulation 5 data, which has higher weights for the observed features that depend on a single latent feature by using the same setup as in simulation 4 (see panel **a**) while drawing coefficients of the observed features from N (0, 16). Left: Reconstruction losses of the two modalities measured by mean squared error (MSE). Right: ℓ_2 norms of the three latent spaces. (**h**) UMAPs of the shared latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the shared latent features and Z_1 or Z_3 as labels. (**i**) UMAPs of the Modality 1 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 1 specific latent features and Z_1 or Z_3 as labels. (**j**) UMAPs of the Modality 2 specific latent space colored by values of Z_1 (left) or Z_3 (right). Silhouette scores are computed using the Modality 2 specific latent features and Z_1 or Z_3 as labels.


(a) Full model & training procedure
Step 1: latent optimization
ATAC specific latent
shared latent
RNA specific latent
ATAC decoder of full latent
ATAC shared latent decoder
RNA shared latent decoder
RNA decoder of full latent
ATAC-seq
cells
peaks
RNA-seq
cells
genes
Step 2: inference
ATAC-seq
cells
peaks
ATAC encoder
RNA-seq
cells
genes
RNA encoder
ATAC specific latent
shared latent
RNA specific latent
(b) Decoder architecture of paired scATAC-seq and scRNA-seq
Input: shared or full latent + noise
Hidden layer: 1024 dimensions
Hidden layer: 1024 dimensions
Hidden layer: 1024 dimensions
Output: scATAC-seq or scRNA-seq
+
Linear layer + LeakyReLU + BatchNorm1d
Linear layer + LeakyReLU + BatchNorm1d
Linear layer + LeakyReLU + BatchNorm1d
Linear layer
Noise
Latent
(c) Encoder architecture of paired scATAC-seq and scRNA-seq
Input: scATAC-seq or scRNA-seq
Hidden layer: 1024 dimensions
Hidden layer: 1024 dimensions
Hidden layer: 50 dimensions
Output: shared latent
Hidden layer: 20 dimensions
Output: modality-specific latent
Linear layer + LeakyReLU + BatchNorm1d
Linear layer + LeakyReLU + BatchNorm1d
Linear layer + LeakyReLU + BatchNorm1d
Linear layer
Linear layer + LeakyReLU + BatchNorm1d
Linear layer

Extended Data Fig. 5 | APOLLO architecture for paired scATAC-seq and scRNA-seq data. (**a**) APOLLO uses a two-step training procedure. In step 1, the shared and modality-specific latent spaces as well as the modality-specific decoders are trained so that the decoders can reconstruct the scATAC-seq and scRNA-seq data from the latent spaces. For each modality, there is one decoder that reconstructs the modality using only the shared latent space and another decoder that reconstructs the modality using the full latent space, that is the shared latent space and the modality-specific latent space. In step 2, modality-specific encoders are trained to enable inferring the latent space embedding for ells not used in training the model and thus enable cross modality prediction. The encoders are trained to map the input features, for example scRNA-seq and scATAC-seq, to the latent space embeddings obtained in step 1. (**b**) The decoders of scRNA-seq and scATAC-seq have the same architectures with four fully connected layers, except that the output layer decodes to different feature dimensions for the two modalities. The input feature dimension is 70 for the full latent space decoder and 50 for the shared latent space decoder. Each decoder has 3 hidden layers, each of which has dimension 1024. All hidden layers are linear layers with a dropout rate of 0.01 and are followed by LeakyReLU activation and a batch normalization layer. The output layer is a linear layer with a dropout rate of 0.01 and sigmoid activation. (**c**) The encoders of scRNA-seq and scATAC-seq have the same architectures, except the difference in the input feature dimensions. Each encoder starts with two linear layers with a dropout rate of 0.01, each of which has 1024 dimensions and is followed by LeakyReLU activation and batch normalization, which is a standard setup for autoencoders [23,58]. After the first two hidden layers, two separate linear layers with LeakyReLU activation are used to obtain separate hidden layers for the shared and modality-specific latent spaces. A linear layer is applied to each hidden layer to obtain the shared or modality-specific latent space.

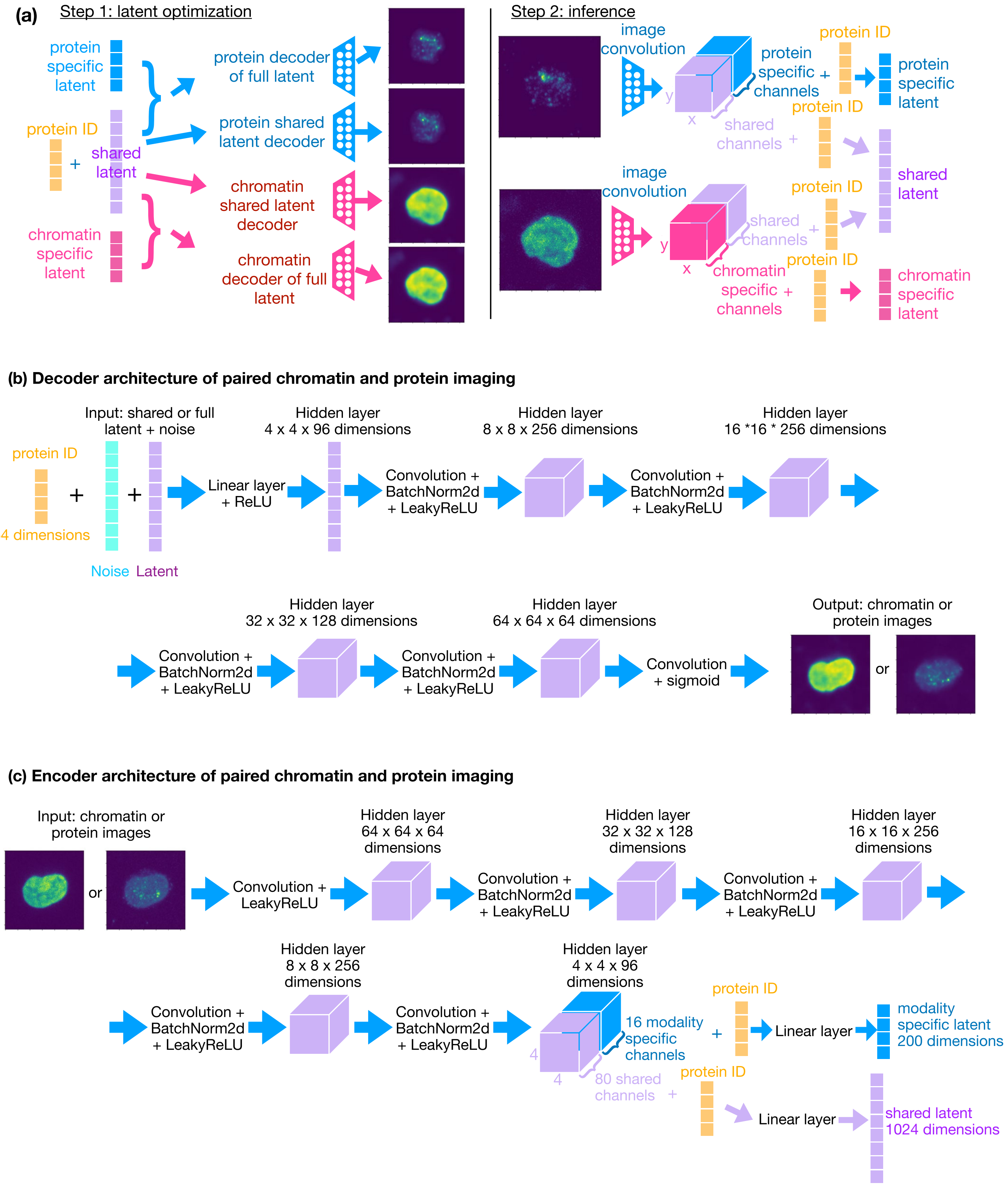


Extended Data Fig. 6 | Application of APOLLO to paired scRNA-seq and protein abundance data. (**a**) We applied the same encoder and decoder architecture as well as the default two-step training procedure to the paired scRNA-seq and protein abundance data (see Methods for details). (**b**) UMAP visualization of scRNA-seq data using the Scanpy package[42]. (**c**) UMAP visualization of the protein abundance data using the Scanpy package[42]. (**d**) UMAPs obtained from the shared latent space of standard autoencoders, encoded using scRNA-seq data, separately colored by cell types and experimental batches.



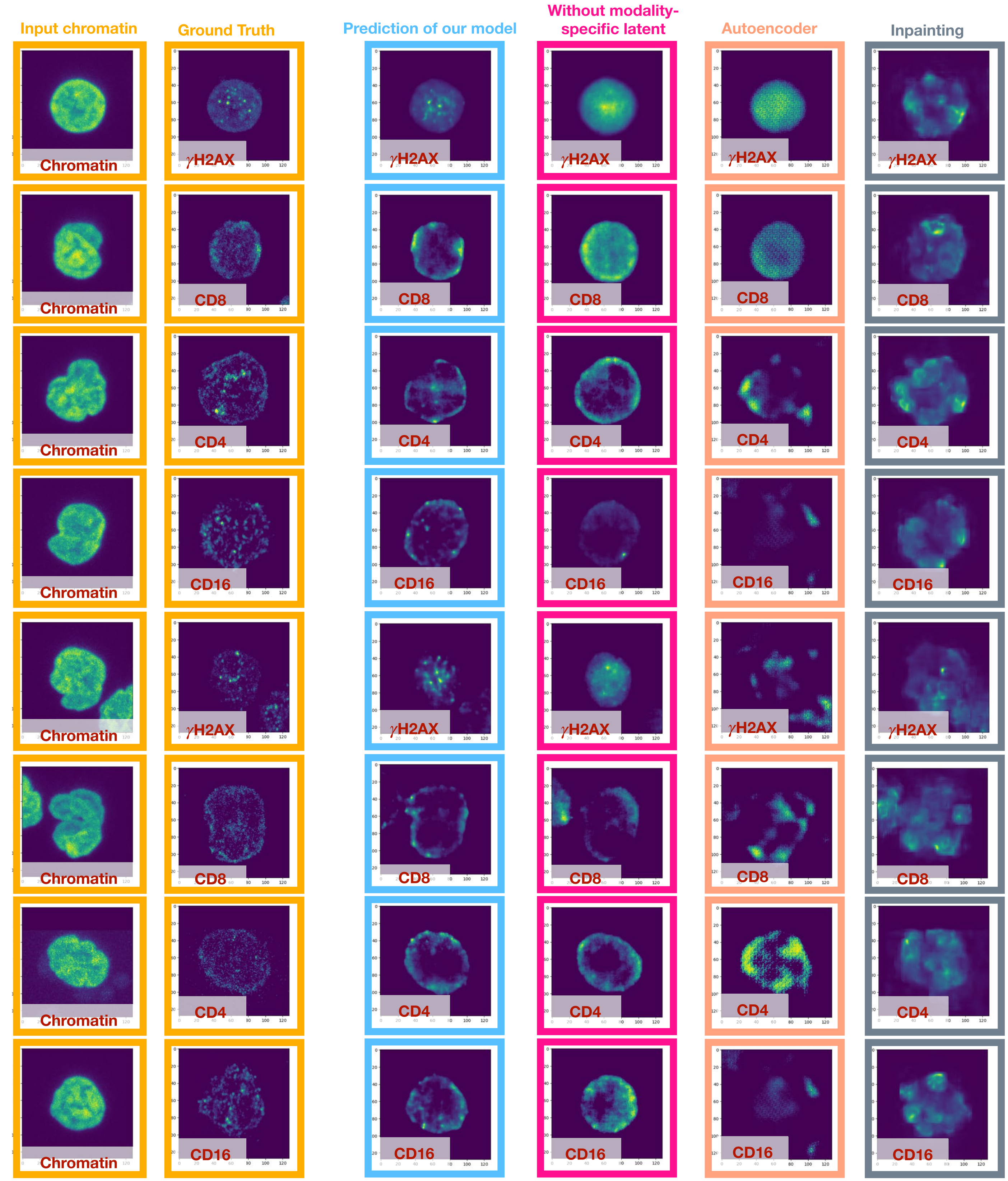
Extended Data Fig. 7 | Comparison of clustering results using APOLLO's latent spaces, the protein abundance data, and a standard variational autoen-coder. (**a**) Leiden clustering of APOLLO's shared latent space. Resolution of leiden clustering was adjusted to obtain the minimum number of clusters such that the major cell types are separated into different clusters. (**b**) Leiden clustering of cellular surface protein abundance data. Resolution of leiden clustering was adjusted to obtain the same number of clusters as (**a**). (**c**) Heatmap showing the number of cells in each of the two batches, for each cluster of the shared latent space. (**d**) Heatmap showing the number of cells in each of the two batches, for each cluster of the cellular protein abundance data. (**e**) Heatmap showing the number of cells in each cell type, for each cluster of the shared latent space. (**f**) Heatmap showing the number of cells in each cell type, for each cluster of the cellular protein abundance data. (**g**) Leiden clustering of APOLLO's RNA-specific latent space. (**h**) Heatmap showing the number of cells in each of the two batches, for each cluster of the RNA-specific latent space. (**i**) Heatmap showing the number of cells in each of the two batches, for each cluster of the full latent space of scRNA-seq learned using APOLLO. (**j**) Heatmap showing the number of cells in each of the two batches, for each cluster of the latent space of scRNA-seq learned using a standard variational autoencoder (Methods). (**k**) Heatmap showing the number of cells in each cell type, for each cluster of the full latent space of scRNA-seq learned using APOLLO. (**l**) Heatmap showing the number of cells in each cell type, for each cluster of the full latent space of scRNA-seq learned using a standard variational autoencoder (Methods).

https://doi.org/10.1038/s43588-025-00948-w

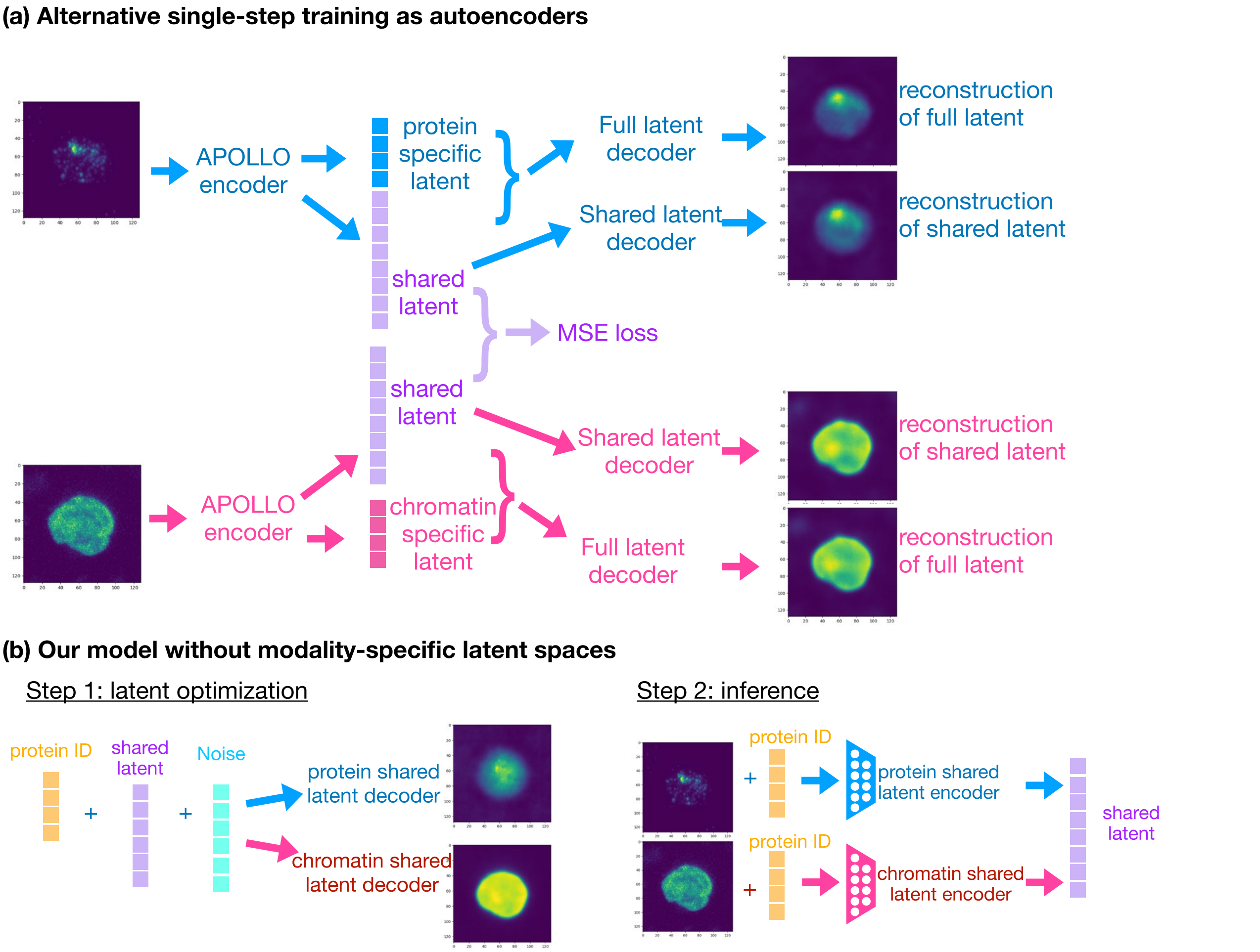

(a)
Step 1: latent optimization
protein specific latent
protein ID
+
shared latent
chromatin specific latent
protein decoder of full latent
protein shared latent decoder
chromatin shared latent decoder
chromatin decoder of full latent
Step 2: inference
image convolution
y
x
protein specific + channels
shared channels +
protein ID
protein specific latent
shared latent
image convolution
y
x
shared channels +
chromatin specific + channels
protein ID
chromatin specific latent
(b) Decoder architecture of paired chromatin and protein imaging
Input: shared or full latent + noise
protein ID
64 dimensions
+
+
Noise
Latent
Linear layer + ReLU
Hidden layer 4 x 4 x 96 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 8 x 8 x 256 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 16 *16 * 256 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 32 x 32 x 128 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 64 x 64 x 64 dimensions
Convolution + sigmoid
Output: chromatin or protein images
or
(c) Encoder architecture of paired chromatin and protein imaging
Input: chromatin or protein images
or
Convolution + LeakyReLU
Hidden layer 64 x 64 x 64 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 32 x 32 x 128 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 16 x 16 x 256 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 8 x 8 x 256 dimensions
Convolution + BatchNorm2d + LeakyReLU
Hidden layer 4 x 4 x 96 dimensions
4
4
16 modality specific channels
80 shared channels +
+
protein ID
Linear layer
modality specific latent 200 dimensions
protein ID
Linear layer
shared latent 1024 dimensions

Extended Data Fig. 8 | See next page for caption.

Extended Data Fig. 8 | APOLLO architecture for paired chromatin and protein images. (**a**) APOLLO can be applied to multiplexed imaging data using conditional decoders and encoders. In step 1, the shared and modality-specific latent spaces as well as the modality-specific decoders are trained to minimize reconstruction error of chromatin and protein images from the latent spaces. In addition to the latent space embeddings, the decoders also take in a trainable vector representing each protein ID. In step 2, modality-specific encoders are trained to enable inferring the latent space embedding for cells not used in training the model and thus enable cross modality prediction. The encoders are trained to map protein and chromatin images together with the trainable protein IDs to the latent space embeddings obtained in step 1. (**b**) The decoders of chromatin and protein images have the same architectures with five convolutional layers following a fully connected layer. The input to the decoder is the full or shared latent space added with noise and concatenated with learnable protein ID. The latent space embedding, after adding noise and being concatenated with the protein ID, is passed through a linear layer with ReLU activation and reshaped to 4 × 4 × 96 dimensions for subsequent convolutions. This is followed by 5 convolutional layers with a kernel size of 4 and stride of 2. The first four convolutional layers are followed by batch normalization and LeakyReLU activation. The last convolutional layer is followed by sigmoid activation to scale the output image from 0 to 1. (**c**) The encoder of each modality starts with 5 convolutional layers with LeakyReLU activation, a standard setup for autoencoders [23,37]. The output of the last convolutional layer is divided into two sets of channels that are used to derive the shared and modality-specific latent spaces respectively. 80 out of the 96 channels of the last hidden layer are flattened, concatenated with protein ID, and passed through a linear layer to obtain the shared latent space. Similarly, the modality-specific latent space is obtained from the remaining 16 channels.



Extended Data Fig. 9 | Examples of protein images predicted from chromatin images of held-out patients obtained from our full APOLLO model are com-pared to the outputs of different alternative models. 'Autoencoder' has the same encoder and decoder structures as our full APOLLO model but is trained in a single step without directly optimizing the latent spaces (see Methods for details). 'Without modality-specific latent' represents our APOLLO model with only the shared latent space. 'Inpainting' corresponds to the model developed by [45].


(a) Alternative single-step training as autoencoders
APOLLO encoder
protein specific latent
shared latent
Full latent decoder
Shared latent decoder
reconstruction of full latent
reconstruction of shared latent
MSE loss
shared latent
chromatin specific latent
APOLLO encoder
Shared latent decoder
Full latent decoder
reconstruction of shared latent
reconstruction of full latent
(b) Our model without modality-specific latent spaces
Step 1: latent optimization
protein ID
shared latent
Noise
+
+
protein shared latent decoder
chromatin shared latent decoder
Step 2: inference
protein ID
+
protein shared latent encoder
protein ID
+
chromatin shared latent encoder
shared latent

Extended Data Fig. 10 | Architecture details of model ablation tests and comparison of reconstruction loss. (**a**) The schematic shows the architecture of our model with partially overlapping latent space trained as a standard autoencoder in a single step. The encoders and decoders have the same architecture as our APOLLO model, but the encoder and decoder are trained together in a single step without directly optimizing the latent spaces. Mean-squared error loss is applied to the shared latent space encoded from each modality. (**b**) The schematic shows the architecture of our model without the modality-specific latent spaces. The model is trained with the two-step training procedure as used for the full APOLLO model. There are only two decoders that decode the two modalities from the shared latent space and they are trained in step 1. The encoders only output the shared latent space, which has the same dimension as the combined dimension of the shared and the modality-specific latent spaces in the full APOLLO model.

nature portfolio

Corresponding author(s): GV Shivashankar and Caroline Uhler

Last updated by author(s): 12/6/2025

Reporting Summary

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

Statistics

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

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

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

Software and code

Policy information about availability of computer code

Data collection

No software was used.

Data analysis

The code is available in the Github repository: https://github.com/uhlerlab/APOLLO/. Our code uses the following Python packages:
_libgcc_mutex=0.1=main
_openmp_mutex=5.1=1_gnu
_tflow_select=2.3.0=mkl
abseil-cpp=20211102.0=hd4dd3e8_0
absl-py=1.4.0=pypi_0
aicsimageio=4.9.4=pypi_0
aicspylibczi=3.0.5=pypi_0
aiohttp=3.9.5=py310h5eee18b_0
aiosignal=1.2.0=pyhd3eb1b0_0
anndata=0.9.1=pyhd8ed1ab_0
anyio=4.2.0=py310h06a4308_0
aom=3.6.0=h6a678d5_0
argon2-cffi=21.3.0=pyhd3eb1b0_0
argon2-cffi-bindings=21.2.0=py310h7f8727e_0
arpack=3.7.0=hdefa2d7_2
asciitree=0.3.3=pypi_0
asttokens=2.0.5=pyhd3eb1b0_0
astunparse=1.6.3=py_0

```
async-lru=2.0.4=py310h06a4308_0
async-timeout=4.0.3=py310h06a4308_0
asyncio=3.4.3=pypi_0
attrs=23.1.0=py310h06a4308_0
babel=2.11.0=py310h06a4308_0
beautifulsoup4=4.12.3=py310h06a4308_0
bioformats-jar=2020.5.27=pypi_0
blas=1.0=mkl
bleach=4.1.0=pyhd3eb1b0_0
blinker=1.6.2=py310h06a4308_0
blosc=1.21.3=h6a678d5_0
bottleneck=1.3.7=py310ha9d4c09_0
brotli=1.0.9=h5eee18b_8
brotli-bin=1.0.9=h5eee18b_8
brotli-python=1.0.9=py310h6a678d5_8
brunsli=0.1=h2531618_0
bzip2=1.0.8=h5eee18b_6
c-ares=1.19.1=h5eee18b_0
ca-certificates=2024.3.11=h06a4308_0
cachetools=5.3.0=pypi_0
certifi=2024.6.2=py310h06a4308_0
cffi=1.16.0=py310h5eee18b_1
cfitsio=3.470=h5893167_7
charls=2.2.0=h2531618_0
charset-normalizer=2.0.4=pyhd3eb1b0_0
chex=0.1.86=pypi_0
click=8.1.3=pypi_0
cloudpickle=2.2.0=pypi_0
comm=0.2.1=py310h06a4308_0
contourpy=1.2.0=py310hdb19cb5_0
cryptography=38.0.1=py310h9ce1e76_0
csbdeep=0.7.3=pypi_0
cuda=11.6.1=0
cuda-cccl=11.6.55=hf6102b2_0
cuda-command-line-tools=11.6.2=0
cuda-compiler=11.6.2=0
cuda-cudart=11.6.55=he381448_0
cuda-cudart-dev=11.6.55=h42ad0f4_0
cuda-cuobjdump=11.6.124=h2eeebcb_0
cuda-cupti=11.6.124=h86345e5_0
cuda-cuxxfilt=11.6.124=hecbf4f6_0
cuda-driver-dev=11.6.55=0
cuda-gdb=12.4.127=h122497a_1
cuda-libraries=11.6.1=0
cuda-libraries-dev=11.6.1=0
cuda-memcheck=11.8.86=0
cuda-nsight=12.4.127=h06a4308_1
cuda-nsight-compute=12.0.0=0
cuda-nvcc=11.6.124=hbba6d2d_0
cuda-nvdisasm=12.4.127=h6a678d5_1
cuda-nvml-dev=11.6.55=haa9ef22_0
cuda-nvprof=12.4.127=h6a678d5_1
cuda-nvprune=11.6.124=he22ec0a_0
cuda-nvrtc=11.6.124=h020bade_0
cuda-nvrtc-dev=11.6.124=h249d397_0
cuda-nvtx=11.6.124=h0630a44_0
cuda-nvvp=12.4.127=h6a678d5_1
cuda-runtime=11.6.1=0
cuda-samples=11.6.101=h8efea70_0
cuda-sanitizer-api=12.4.127=h99ab3db_1
cuda-toolkit=11.6.1=0
cuda-tools=11.6.1=0
cuda-version=12.4=hbda6634_3
cuda-visual-tools=11.6.1=0
cycler=0.11.0=pyhd3eb1b0_0
cytoolz=0.12.2=py310h5eee18b_0
dask=2022.12.1=pypi_0
dask-core=2024.5.0=py310h06a4308_0
dav1d=1.2.1=h5eee18b_0
dbus=1.13.18=hb2f20db_0
debugpy=1.6.7=py310h6a678d5_0
decorator=5.1.1=pyhd3eb1b0_0
defusedxml=0.7.1=pyhd3eb1b0_0
dm-tree=0.1.7=py310h6a678d5_1
dnspython=2.2.1=pypi_0
docrep=0.3.2=pyh44b312d_0
```

```
elementpath=3.0.2=pypi_0
email-validator=1.3.0=pypi_0
et_xmlfile=1.1.0=py310h06a4308_0
exceptiongroup=1.2.0=py310h06a4308_0
executing=0.8.3=pyhd3eb1b0_0
expat=2.6.2=h6a678d5_0
fasteners=0.18=pypi_0
ffmpeg=4.2.2=h20bf706_0
filelock=3.11.0=pypi_0
flatbuffers=23.3.3=pypi_0
flax=0.6.1=pyhd8ed1ab_1
fontconfig=2.14.1=h4c34cd2_2
fonttools=4.51.0=py310h5eee18b_0
freetype=2.12.1=h4a9f257_0
frozenlist=1.4.0=py310h5eee18b_0
fsspec=2024.3.1=py310h06a4308_0
future=0.18.3=py310h06a4308_0
gast=0.4.0=pyhd3eb1b0_0
gdown=4.7.1=pypi_0
gds-tools=1.9.1.3=h99ab3db_1
giflib=5.2.1=h5eee18b_3
glib=2.78.4=h6a678d5_0
glib-tools=2.78.4=h6a678d5_0
glpk=4.65=h276157c_3
gmp=6.2.1=h295c915_3
gnutls=3.6.15=he1e5248_0
google-auth=2.17.3=pypi_0
google-auth-oauthlib=1.0.0=pypi_0
google-pasta=0.2.0=pyhd3eb1b0_0
grpc-cpp=1.46.1=h33aed49_1
grpcio=1.54.0=pypi_0
gst-plugins-base=1.14.1=h6a678d5_1
gstreamer=1.14.1=h5eee18b_1
h5py=3.8.0=pypi_0
hdf5=1.12.1=h70be1eb_2
icu=58.2=he6710b0_3
idna=3.7=py310h06a4308_0
igraph=0.9.8=hf5496dd_0
imagecodecs=2022.12.24=pypi_0
imageio=2.33.1=py310h06a4308_0
importlib-metadata=7.0.1=py310h06a4308_0
importlib_metadata=7.0.1=hd3eb1b0_0
intel-openmp=2021.4.0=h06a4308_3561
ipykernel=6.28.0=py310h06a4308_0
ipython=8.25.0=py310h06a4308_0
ipywidgets=8.1.2=py310h06a4308_0
jax=0.4.30=pypi_0
jaxlib=0.4.30=pypi_0
jedi=0.18.1=py310h06a4308_1
jgo=1.0.5=pypi_0
jinja2=3.1.4=py310h06a4308_0
joblib=1.4.2=py310h06a4308_0
jpeg=9e=h5eee18b_1
jpype1=1.4.1=pypi_0
json5=0.9.6=pyhd3eb1b0_0
jsonschema=4.19.2=py310h06a4308_0
jsonschema-specifications=2023.7.1=py310h06a4308_0
jupyter=1.0.0=py310h06a4308_8
jupyter-lsp=2.2.0=py310h06a4308_0
jupyter_client=8.6.0=py310h06a4308_0
jupyter_console=6.6.3=py310h06a4308_0
jupyter_core=5.7.2=py310h06a4308_0
jupyter_events=0.10.0=py310h06a4308_0
jupyter_server=2.14.1=py310h06a4308_0
jupyter_server_terminals=0.4.4=py310h06a4308_1
jupyterlab=4.0.11=py310h06a4308_0
jupyterlab_pygments=0.1.2=py_0
jupyterlab_server=2.25.1=py310h06a4308_0
jupyterlab_widgets=3.0.10=py310h06a4308_0
jxrlib=1.1=h7b6447c_2
keras=2.12.0=pypi_0
keras-preprocessing=1.1.2=pyhd3eb1b0_0
kiwisolver=1.4.4=py310h6a678d5_0
krb5=1.19.4=h568e23c_0
lame=3.100=h7b6447c_0
lcms2=2.12=h3be6417_0
```

```
ld_impl_linux-64=2.38=h1181459_1
leidenalg=0.8.10=py310hd8f1fbe_0
lerc=3.0=h295c915_0
libaec=1.0.4=he6710b0_1
libavif=0.11.1=h5eee18b_0
libblas=3.9.0=12_linux64_mkl
libbrotlicommon=1.0.9=h5eee18b_8
libbrotlidec=1.0.9=h5eee18b_8
libbrotlienc=1.0.9=h5eee18b_8
libclang=16.0.0=pypi_0
libclang13=14.0.6=default_he11475f_1
libcublas=11.9.2.110=h5e84587_0
libcublas-dev=11.9.2.110=h5c901ab_0
libcufft=10.7.1.112=hf425ae0_0
libcufft-dev=10.7.1.112=ha5ce4c0_0
libcufile=1.9.1.3=h99ab3db_1
libcufile-dev=1.9.1.3=h99ab3db_1
libcurand=10.3.5.147=h99ab3db_1
libcurand-dev=10.3.5.147=h99ab3db_1
libcurl=7.86.0=h91b91d3_0
libcusolver=11.3.4.124=h33c3c4e_0
libcusparse=11.7.2.124=h7538f96_0
libcusparse-dev=11.7.2.124=hbbe9722_0
libdeflate=1.17=h5eee18b_1
libedit=3.1.20230828=h5eee18b_0
libev=4.33=h7f8727e_1
libevent=2.1.12=h8f2d780_0
libffi=3.4.4=h6a678d5_1
libgcc-ng=11.2.0=h1234567_1
libgfortran-ng=11.2.0=h00389a5_1
libgfortran5=11.2.0=h1234567_1
libglib=2.78.4=hdc74915_0
libgomp=11.2.0=h1234567_1
libiconv=1.16=h5eee18b_3
libidn2=2.3.4=h5eee18b_0
liblapack=3.9.0=12_linux64_mkl
libllvm14=14.0.6=hdb19cb5_3
libnghttp2=1.46.0=hce63b2e_0
libnpp=11.6.3.124=hd2722f0_0
libnpp-dev=11.6.3.124=h3c42840_0
libnvjpeg=11.6.2.124=hd473ad6_0
libnvjpeg-dev=11.6.2.124=hb5906b9_0
libopus=1.3.1=h7b6447c_0
libpng=1.6.39=h5eee18b_0
libpq=12.9=h16c4e8d_3
libprotobuf=3.20.3=he621ea3_0
libsodium=1.0.18=h7b6447c_0
libssh2=1.10.0=h8f2d780_0
libstdcxx-ng=11.2.0=h1234567_1
libtasn1=4.19.0=h5eee18b_0
libtiff=4.5.1=h6a678d5_0
libunistring=0.9.10=h27cfd23_0
libuuid=1.41.5=h5eee18b_0
libvpx=1.7.0=h439df22_0
libwebp-base=1.3.2=h5eee18b_0
libxcb=1.15=h7f8727e_0
libxkbcommon=1.0.1=h5eee18b_1
libxml2=2.10.4=hcbfbd50_0
libzopfli=1.0.3=he6710b0_0
lightning-utilities=0.9.0=py310h06a4308_0
llvmlite=0.39.1=pypi_0
locket=1.0.0=py310h06a4308_0
lz4-c=1.9.4=h6a678d5_1
markdown=3.4.3=pypi_0
markdown-it-py=2.2.0=py310h06a4308_1
markupsafe=2.1.3=py310h5eee18b_0
matplotlib=3.6.2=py310h06a4308_0
matplotlib-base=3.6.2=py310h945d387_0
matplotlib-inline=0.1.6=py310h06a4308_0
mdurl=0.1.0=py310h06a4308_0
metis=5.1.0=hf484d3e_4
mistune=2.0.4=py310h06a4308_0
mkl=2021.4.0=h06a4308_640
mkl-service=2.4.0=py310h7f8727e_0
mkl_fft=1.3.1=py310hd6ae3a3_0
mkl_random=1.2.2=py310h00e6091_0
```

```
ml-dtypes=0.4.0=pypi_0
mpfr=4.0.2=hb69a4c5_1
msgpack-python=1.0.3=py310hd09550d_0
multicor-fa=1.0.2=pypi_0
multidict=6.0.4=py310h5eee18b_0
multipledispatch=0.6.0=py310h06a4308_0
natsort=7.1.1=pyhd3eb1b0_0
nbclient=0.8.0=py310h06a4308_0
nbconvert=7.10.0=py310h06a4308_0
nbformat=5.9.2=py310h06a4308_0
ncurses=6.4=h6a678d5_0
nest-asyncio=1.6.0=py310h06a4308_0
nettle=3.7.3=hbbd107a_1
networkx=3.3=py310h06a4308_0
notebook=7.0.8=py310h06a4308_0
notebook-shim=0.2.3=py310h06a4308_0
nsight-compute=2022.4.0.15=0
nspr=4.35=h6a678d5_0
nss=3.89.1=h6a678d5_0
numba=0.59.1=py310h6a678d5_0
numcodecs=0.11.0=pypi_0
numexpr=2.8.4=py310h8879344_0
numpy=1.25.2=pypi_0
numpy-base=1.24.3=py310h8e6c178_0
numpyro=0.13.2=pyhd8ed1ab_0
oauthlib=3.2.2=py310h06a4308_0
ome-types=0.3.2=pypi_0
openh264=2.1.1=h4ff587b_0
openjpeg=2.4.0=h9ca470c_1
openpyxl=3.1.2=py310h5eee18b_0
openssl=1.1.1w=h7f8727e_0
opt_einsum=3.3.0=pyhd3eb1b0_1
optax=0.1.4=py310h06a4308_0
overrides=7.4.0=py310h06a4308_0
packaging=24.1=py310h06a4308_0
pandas=1.5.2=pypi_0
pandocfilters=1.5.0=pyhd3eb1b0_0
parso=0.8.3=pyhd3eb1b0_0
partd=1.3.0=pypi_0
patsy=0.5.6=py310h06a4308_0
pcre2=10.42=hebb0a14_1
pexpect=4.8.0=pyhd3eb1b0_3
pillow=10.3.0=py310h5eee18b_0
pint=0.20.1=pypi_0
pip=24.0=py310h06a4308_0
platformdirs=3.10.0=py310h06a4308_0
ply=3.11=py310h06a4308_0
pooch=1.7.0=py310h06a4308_0
prometheus_client=0.14.1=py310h06a4308_0
prompt-toolkit=3.0.43=py310h06a4308_0
prompt_toolkit=3.0.43=hd3eb1b0_0
protobuf=4.22.3=pypi_0
psutil=5.9.0=py310h5eee18b_0
ptyprocess=0.7.0=pyhd3eb1b0_2
pure_eval=0.2.2=pyhd3eb1b0_0
pyasn1=0.5.0=pypi_0
pyasn1-modules=0.3.0=pypi_0
pycparser=2.21=pyhd3eb1b0_0
pydantic=1.10.4=pypi_0
pydeprecate=0.3.1=pyhd8ed1ab_0
pygments=2.15.1=py310h06a4308_1
pyjwt=2.8.0=py310h06a4308_0
pynndescent=0.5.10=py310h06a4308_0
pyopenssl=22.0.0=pyhd3eb1b0_0
pyparsing=3.0.9=py310h06a4308_0
pyqt=5.15.10=py310h6a678d5_0
pyqt5-sip=12.13.0=py310h5eee18b_0
pyro-api=0.1.2=pyhd8ed1ab_0
pyro-ppl=1.8.6=pyhd8ed1ab_0
pysocks=1.7.1=py310h06a4308_0
python=3.10.8=h7a1cb2a_1
python-dateutil=2.9.0post0=py310h06a4308_2
python-fastjsonschema=2.16.2=py310h06a4308_0
python-flatbuffers=2.0=pyhd3eb1b0_0
python-igraph=0.9.10=py310h04c1b7f_1
python-json-logger=2.0.7=py310h06a4308_0
```

```
python-tzdata=2023.3=pyhd3eb1b0_0
python-version=0.0.2=pypi_0
python_abi=3.10=2_cp310
pytorch=1.13.1=py3.10_cuda11.6_cudnn8.3.2_0
pytorch-cuda=11.6=h867d48c_1
pytorch-lightning=1.5.8=pyhd8ed1ab_0
pytorch-mutex=1.0=cuda
pytz=2024.1=py310h06a4308_0
pywavelets=1.5.0=py310ha9d4c09_0
pyyaml=6.0.1=py310h5eee18b_0
pyzmq=25.1.2=py310h6a678d5_0
qt-main=5.15.2=h8373d8f_8
qtconsole=5.5.1=py310h06a4308_0
qtpy=2.4.1=py310h06a4308_0
re2=2022.04.01=h295c915_0
readline=8.2=h5eee18b_0
referencing=0.30.2=py310h06a4308_0
requests=2.32.2=py310h06a4308_0
requests-oauthlib=1.3.1=pypi_0
resource-backed-dask-array=0.1.0=pypi_0
rfc3339-validator=0.1.4=py310h06a4308_0
rfc3986-validator=0.1.1=py310h06a4308_0
rich=13.3.5=py310h06a4308_0
rpds-py=0.10.6=py310hb02cf49_0
rsa=4.9=pypi_0
scanpy=1.9.3=pyhd8ed1ab_0
scikit-image=0.19.3=py310h6a678d5_1
scikit-learn=1.4.2=py310h1128e8f_1
scipy=1.10.1=py310hd5efca6_0
scvi-tools=0.16.3=pyhd8ed1ab_0
scyjava=1.8.1=pypi_0
seaborn=0.12.2=py310h06a4308_0
send2trash=1.8.2=py310h06a4308_0
session-info=1.0.0=pyhd8ed1ab_0
setuptools=69.5.1=py310h06a4308_0
sip=6.7.12=py310h6a678d5_0
six=1.16.0=pyhd3eb1b0_1
snappy=1.1.10=h6a678d5_1
sniffio=1.3.0=py310h06a4308_0
soupsieve=2.5=py310h06a4308_0
sqlite=3.45.3=h5eee18b_0
stack_data=0.2.0=pyhd3eb1b0_0
stardist=0.8.3=pypi_0
statsmodels=0.14.2=py310h5eee18b_0
stdlib-list=0.10.0=py310h06a4308_0
suitesparse=5.10.1=h446ee2e_2
tbb=2021.8.0=hdb19cb5_0
tensorboard=2.12.2=pypi_0
tensorboard-data-server=0.7.0=pypi_0
tensorboard-plugin-wit=1.8.1=py310h06a4308_0
tensorflow=2.10.0=mkl_py310h24f4fea_0
tensorflow-base=2.10.0=mkl_py310hb9daa73_0
tensorflow-estimator=2.12.0=pypi_0
tensorflow-io-gcs-filesystem=0.32.0=pypi_0
termcolor=2.2.0=pypi_0
terminado=0.17.1=py310h06a4308_0
texttable=1.6.4=pyhd3eb1b0_0
threadpoolctl=3.5.0=py310h2f386ee_0
tifffile=2022.10.10=pypi_0
tinycss2=1.2.1=py310h06a4308_0
tk=8.6.14=h39e8969_0
tomli=2.0.1=py310h06a4308_0
toolz=0.12.0=py310h06a4308_0
torchaudio=0.13.1=py310_cu116
torchmetrics=1.4.0.post0=py310h06a4308_0
torchvision=0.14.1=py310_cu116
tornado=6.4.1=py310h5eee18b_0
tqdm=4.66.4=py310h2f386ee_0
traitlets=5.14.3=py310h06a4308_0
typing-extensions=4.11.0=py310h06a4308_0
typing_extensions=4.11.0=py310h06a4308_0
tzdata=2024a=h04d1e81_0
umap-learn=0.5.3=py310hff52083_0
unicodedata2=15.1.0=py310h5eee18b_0
urllib3=2.2.2=py310h06a4308_0
wcwidth=0.2.5=pyhd3eb1b0_0
```

```
webencodings=0.5.1=py310h06a4308_1
websocket-client=1.8.0=py310h06a4308_0
werkzeug=3.0.3=py310h06a4308_0
wheel=0.43.0=py310h06a4308_0
widgetsnbextension=4.0.10=py310h06a4308_0
wrapt=1.14.1=py310h5eee18b_0
x264=1!157.20191217=h7b6447c_0
xarray=2022.12.0=pypi_0
xmlschema=2.1.1=pypi_0
xz=5.4.6=h5eee18b_1
yaml=0.2.5=h7b6447c_0
yarl=1.9.3=py310h5eee18b_0
zarr=2.13.3=pypi_0
zeromq=4.3.5=h6a678d5_0
zfp=1.0.0=h6a678d5_0
zipp=3.17.0=py310h06a4308_0
zlib=1.2.13=h5eee18b_1
zstd=1.5.5=hc292b87_2
```

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

Data

Policy information about availability of data

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

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All datasets used in this study are publicly available. The SHARE- seq data from [5] are available under accession number GEO: GSE140203. The CITE-seq data from [17] are available under accession number GEO: GSE150599. The PBMCs multiplex imaging dataset from [36] is available from the PSI Public Data Repository at https://doi.org/10.16907/b039dc4e-9366-413c-8f34-92ce9110cc14. The Human Protein Atlas images from [27] are available at https://www.proteinatlas.org. Source data for Figures 2 and 3b-c is available with this manuscript. Data in Figures 3d, 4 and 5 can be reproduced using our deposited code and publicly available data.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	Not applicable.
Reporting on race, ethnicity, or other socially relevant groupings	Not applicable.
Population characteristics	Not applicable.
Recruitment	Not applicable.
Ethics oversight	Not applicable.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

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

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

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

Life sciences study design

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

Sample size	This study did not generate new data. Public datasets with large sample sizes were selected for each application of our model.

Data exclusions	No data were excluded.
Replication	No replication was performed. Random subsampling of data and train-test splits were performed to ensure robustness.
Randomization	This study did not generate new data.
Blinding	No blinding was performed. Random subsampling of data was performed to ensure robustness.

Reporting for specific materials, systems and methods

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

Materials & experimental systems

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

Methods

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

Plants

Seed stocks	*Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.*
Novel plant genotypes	*Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.*
Authentication	*Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.*