nature methods

Article

https://doi.org/10.1038/s41592-024-02257-y

Dependency-aware deep generative models for multitasking analysis of spatial omics data

Received: 2 January 2023

Accepted: 25 March 2024

Published online: 23 May 2024

Check for updates

Tian Tian[1,2,6], Jie Zhang[3,6], Xiang Lin[4], Zhi Wei[4] ✉ & Hakon Hakonarson[2,5]

Spatially resolved transcriptomics (SRT) technologies have significantly advanced biomedical research, but their data analysis remains challenging due to the discrete nature of the data and the high levels of noise, compounded by complex spatial dependencies. Here, we propose spaVAE, a dependency-aware, deep generative spatial variational autoencoder model that probabilistically characterizes count data while capturing spatial correlations. spaVAE introduces a hybrid embedding combining a Gaussian process prior with a Gaussian prior to explicitly capture spatial correlations among spots. It then optimizes the parameters of deep neural networks to approximate the distributions underlying the SRT data. With the approximated distributions, spaVAE can contribute to several analytical tasks that are essential for SRT data analysis, including dimensionality reduction, visualization, clustering, batch integration, denoising, differential expression, spatial interpolation, resolution enhancement and identification of spatially variable genes. Moreover, we have extended spaVAE to spaPeakVAE and spaMultiVAE to characterize spatial ATAC-seq (assay for transposase-accessible chromatin using sequencing) data and spatial multi-omics data, respectively.

Spatially resolved transcriptomics (SRT) have enabled high-throughput profiling of gene expression while retaining spatial information, which has fostered significant progress in a variety of biomedical research fields[1]. Despite technical differences, similar to the single-cell RNA sequencing (scRNA-seq) data, the expressions measured by SRT are often discrete, over-dispersed and noisy[2]. Besides gene expression, relative locations of cells or spots in a tissue are also measured in SRT. The natural spatial dependencies in biological studies are highly informative if they are captured accurately. With sophisticated dependency-aware models, we expect to optimize the whole computational pipeline to deliver more accurate analytical results. Existing computational methods for regular scRNA-seq data[3–7] may not be superior in spatial domain detection because they fail to leverage valuable spatial information and simply assume that cells or spots are independent.

Several deep learning methods have been proposed to explicitly model spatial information for analysis of SRT data (Supplementary Note 1). Many of them, including SpaGCN[8], GraphST[9], STAGATE[10], DSSC[11] and stLean[12], need to explicitly build the dependency relationship between spatial locations before model training. We note that capturing the dependency relationship between spatial locations is a

[1]School of Computer Science, National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, Wuhan, Hubei, China. [2]Center for Applied Genomics, The Children's Hospital of Philadelphia, Philadelphia, PA, USA. [3]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, China. [4]Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. [5]Division of Human Genetics, Department of Pediatrics, The Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA. [6]These authors contributed equally: Tian Tian, Jie Zhang.
✉e-mail: zhiwei@njit.edu

separate first step and is independent of downstream analytical modeling in these graph-based, constraint-based or normalization-based methods. Any artifactual bias introduced into the built graphs, constraints or normalization is likely to be passed down to downstream analytical tasks and may lead to suboptimal results.

Besides deep learning approaches, a few statistical methods have also been proposed to model SRT data[13,14]. For technical convenience, these statistical approaches generally make strong model assumptions. Given the diversity and complexity of SRT data and the challenges in data normalization, it is difficult for most SRT data to satisfy these distribution assumptions. The performance of these statistical methods would then deteriorate when these assumptions do not hold.

In addition to the problems discussed above, many existing deep learning approaches and statistical methods for SRT data also share one common limitation. Namely, they aim only at solving certain specific analytical tasks, such as clustering, and consider only a few key aspects of spatial omics data, for example, spatial information, while ignoring other nuisance factors, including over-dispersed discrete count data, library size, batch effects and so on. We expect further improvement when these critical factors are taken into consideration.

Here, we propose spaVAE, a fully probabilistic and dependency-aware, deep generative spatial variational autoencoder model for analyzing SRT data. spaVAE is a hierarchical Bayesian model with conditional distributions specified by deep neural networks, capturing spatial dependencies through a hybrid embedding of both a Gaussian process prior[15] and a Gaussian prior[16]. spaVAE has the following major merits. First, spatial dependency modeling is an integral part of the whole learning procedure. The strength of spatial dependency will be learned adaptively from data. Second, it proposes a hybrid approach that combines Gaussian process and standard Gaussian embeddings, both of which can capture spatially dependent and independent variations in SRT data. Consequently, spaVAE can efficiently characterize the underlying distribution of SRT data, facilitating a range of analysis tasks, including dimensionality reduction, visualization, clustering, batch integration, denoising, differential expression, spatial interpolation and resolution enhancement (Supplementary Table 1). Third, spaVAE models the discrete count data directly by a negative binomial model-based reconstruction loss. This technique has been proven effective in modeling over-dispersion and library size in several previous scRNA-seq methodology studies[4–7]. Fourth, SRT data from different batches can be integrated effectively in spaVAE. And last, we apply the sparse Gaussian process regression technique to accelerate calculation[17–19], making the model efficient even for large datasets. Based on spaVAE, we propose spaLDVAE to identify spatially variable genes (SVGs). Meanwhile, we note that recent advances in spatial omics technologies enable the profiling of a broader array of data types with location information. For example, the spatial ATAC-seq (assay for transposase-accessible chromatin using sequencing) data profile the genome-wide chromatin accessibility spatially[20,21], and spatial-CITE-seq (spatial co-indexing of transcriptomes and epitopes for multi-omics mapping by highly parallel sequencing) data provide concurrent measurements of gene expression and surface protein intensity across tissue spots[22,23]. To accommodate these evolving data types, we have expanded our dependency-aware deep generative model for the spatial ATAC-seq data (spaPeakVAE) and the spatial multi-omics data (spaMultiVAE).

Results

Spatial dependency-aware deep generative models

A schematic diagram of the proposed spatial variational autoencoder models for SRT data (spaVAE) and spatial ATAC-seq data (spaPeakVAE) is shown in Fig. 1a. Our spaVAE model is based on a variational autoencoder (VAE) to probabilistically model the SRT data. Different from the typical VAE model with a standard Gaussian prior, we model the spatial dependency by introducing a Gaussian process prior. Technical details and the rationale of spaVAE are given in Methods and Supplementary Note 2. The outputs of spaVAE can be used for a series of downstream analyses. First, the latent representation learned by the bottleneck layer can be used for visualization and clustering analysis. Second, the output of the negative binomial model-based decoder provides an estimate of the distribution underlying the observed SRT count data, and can be used for denoising and differential expression analysis. Finally, owing to its generative nature, the Gaussian process prior enables us to interpolate the observations of unseen locations. This favorable feature enables the interpolation of unobserved spots and the enhancement of spatial resolution. Resolution enhancement could provide more biological insights that are not detectable in the original data. Theoretically, our Gaussian process prior could interpolate the latent representations on any suitable locations, enabling the model to flexibly enhance the spatial resolution at user-desired distinguishability.

The spaPeakVAE model serves as an extension of the spaVAE, tailored specifically to the analysis of spatial ATAC-seq data. The spatial ATAC-seq data profile the accessibility of chromatin with location information, and the observation is highly sparse and also binary (that is, whether the peak region is accessible or not). Following previous studies[24,25], we use a Bernoulli model-based decoder in the spaPeakVAE to accommodate the binary nature of the ATAC-seq data. Additionally, spaPeakVAE considers sequencing bias for both spots and peaks. As shown in Fig. 1b, we can also extend the deep generative model to the spatial multi-omics data (spaMultiVAE). In the spatial multi-omics data, the expression of both genes and surface proteins is profiled, and the intensity of surface proteins is measured using barcoded antibodies. However, this technology will introduce distinct technical biases, such as undesirable background due to ambient or nonspecific antibody bindings. To overcome this problem we introduce a negative binomial mixture model-based decoder to separate the background and foreground protein signals, which has been successfully used in the totalVI analysis of single-cell CITE-seq data[26]. Both spaPeakVAE and spaMultiVAE share a similar framework with spaVAE. As a result, the multiple analyses applicable in spaVAE can also be performed using these two variant models.

Figure 1c illustrates a variant of the spaVAE and spaPeakVAE models that involves a spatial linear decoder variational autoencoder (that is, spaLDVAE and spaPeakLDVAE). Like the spaVAE model, the latent embedding in this model also contains two parts: one follows a Gaussian process prior and the other follows a Gaussian prior. Concurrently, a linear decoder is used. Both the latent embedding and the weighting of the linear decoder are constrained to be nonnegative. spaLDVAE and spaPeakLDVAE can be considered as nonnegative factor analysis models. The weighting of the linear decoder can be used to measure the contributions of the two priors to the reconstruction. This enables the prioritization of SVGs or spatially variable peaks based on the contribution score of the Gaussian process embedding part.

spaVAE for multiple analyses of human dorsolateral prefrontal data

We applied spaVAE to the LIBD human dorsolateral prefrontal cortex (DLPFC) dataset[27] to illustrate the performance of various analyses. This dataset sequenced 12 tissue sections spanning six neuronal layers and the white matter from the DLPFC in three human brains.

First, we obtained the low-dimensional embedding of spaVAE on the 12 samples and then conducted k-means clustering to identify spatial domains. The authors provided the manually annotated layer labels that enable us to evaluate the embedding quality and clustering performance. Figure 2a,b shows the clustering accuracy quantified using two metrics: adjusted Rand index (ARI) and normalized mutual information (NMI) (Supplementary Note 3). As we can see, spaVAE and GraphST achieve the highest clustering accuracies out of all of the competing methods, including those specifically designed for the clustering analysis of SRT data, such as STAGATE, BayesSpace, SpaGCN

https://doi.org/10.1038/s41592-024-02257-y

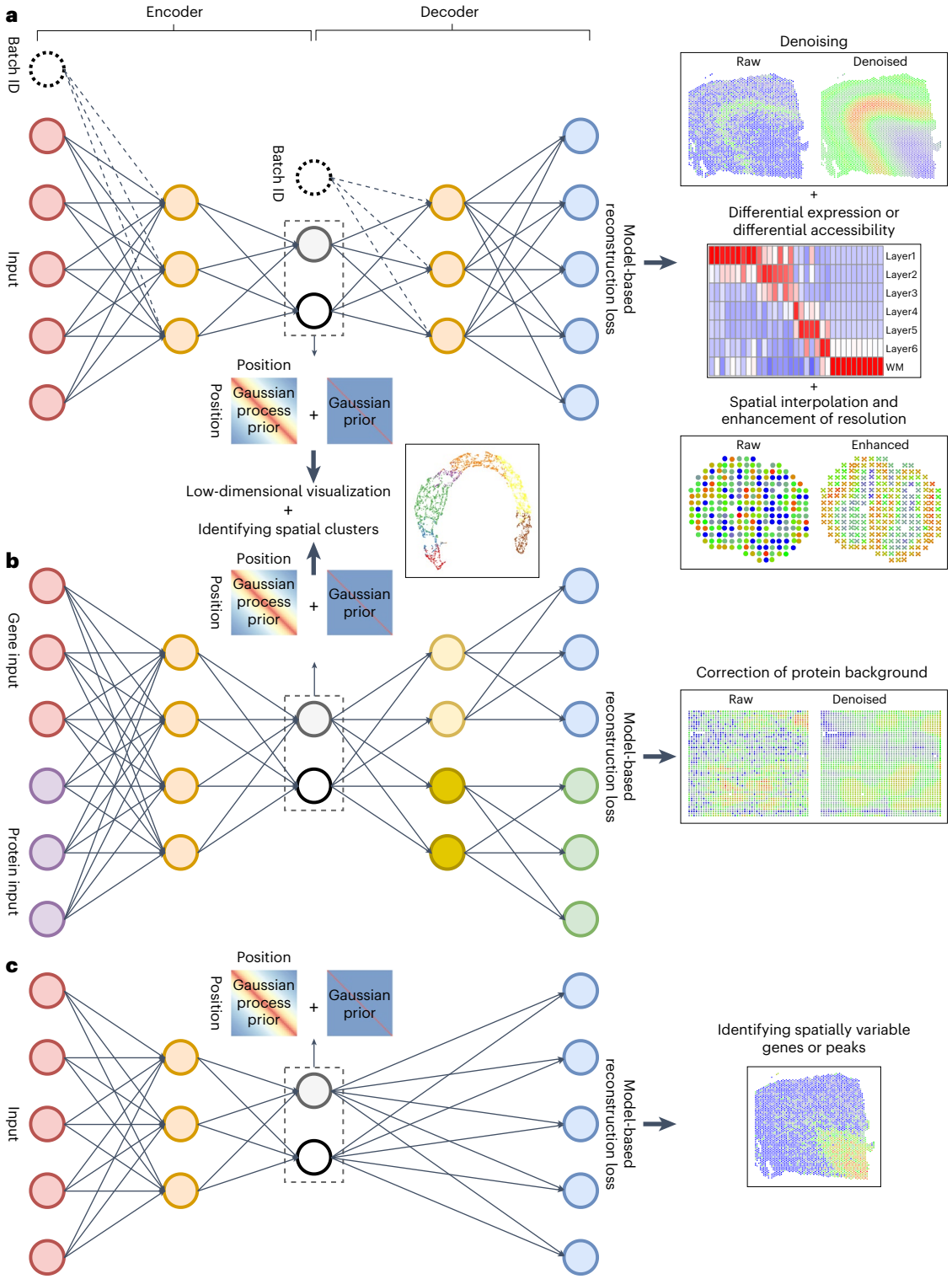


Fig. 1 | The network architecture of dependency-aware deep generative models. a, The network architecture of spaVAE and spaPeakVAE. spaVAE and spaPeakVAE are variational autoencoder models with a hybrid of Gaussian process prior and Gaussian prior. The encoder and decoder are fully connected neural networks. The Gaussian process and Gaussian priors account for the spatial dependent and independent variances, respectively. Batch IDs can be incorporated by an optional conditional autoencoder. Negative binomial reconstruction loss characterizes the discrete mRNA count data in spaVAE, and Bernoulli reconstruction loss characterizes the binary ATAC data in spaPeakVAE. The bottleneck layer can be used for low-dimensional embedding and clustering analysis. The decoder network can be used for denoising, differential expression and differential accessibility analysis. The Gaussian process prior can be used for spatial interpolation and resolution enhancement. WM, white matter. **b**, The network architecture of spaMultiVAE. Spatial multi-omics data simultaneously measure mRNA and surface protein profiles. In the spaMultiVAE model, the negative binomial reconstruction loss characterizes the mRNA data, and the negative binomial mixture reconstruction loss characterizes the background and the foreground intensities of protein data. Thus, the functional analyses implemented in spaVAE can also be fulfilled in spaMultiVAE, including latent representation, denoising, differential expression, enhancing resolutions and so on. **c**, Illustration of the network architecture of the spaLDVAE and spaPeakLDVAE models for identifying SVGs and peaks. The latent embedding also contains two parts: one follows a Gaussian process prior and the other follows a Gaussian prior. A linear decoder is used to quantify the contribution of the two parts for the reconstruction, and the contribution of the Gaussian process part can be used as the score to prioritize the SVGs or spatially variable peaks.

and Giotto[28]. The overall performances of spaVAE and GraphST are comparable. Given that spaVAE is a deep learning-based method, the training procedure may be affected by randomness. To demonstrate the robustness of spaVAE, we repeated the experiments five times with different random seeds and evaluated the performance of SVGs and highly variable genes (HVGs). We observed that spaVAE is quite robust against various random seeds and different gene selection approaches (Supplementary Figs. 1 and 2, and Supplementary Note 4). We show the clustering results of various methods for the DLPFC section 151673 in Fig. 2c,d. The clustering labels inferred by spaVAE have clear layer structures, which are consistent with the ground truth labels. In the scVI result the spatial pattern is very vague. These results illustrate the importance of integrating the location information into the deep learning model for characterizing SRT data. Clustering results and low-dimensional embeddings of spaVAE for other tissue sections are given in Supplementary Figs. 3–6. We present the latent representations of different methods using UMAP (uniform manifold approximation and projection)[29] for DLPFC section 151673 in Fig. 2e. Furthermore, we quantified the separation of different true labels in the latent representations produced by various methods, as shown in Supplementary Fig. 7. The latent representation of spaVAE was found to perform best in differentiating the neuronal layers.

Second, we evaluate the performance of denoising and differential expression analysis for the trained model. Figure 2f presents the denoised counts of marker genes in different layers (the list of marker genes is obtained from the original paper[27]). After denoising by spaVAE we observe that the marker gene expression has clear layer-wise structures, while the raw counts are very noisy and, in most cases, do not have any obvious spatial pattern. Figure 2g shows the result of differential expression analysis. We first aggregated the counts of the same neuronal layer across the 12 sections and built the pseudo-bulk RNA-seq data. Then we used the DESeq2 (ref. 30) package to conduct differential expression analysis on the pseudo-bulk RNA-seq data by comparing each layer with the others. Following Lopez et al.[5], the Wald statistics of DESeq2 for the pseudo-bulk RNA-seq data are treated as the ground truth differential expression result. Next, we calculated the Spearman's correlation between the differential expression statistics inferred using the different methods for the counts of each tissue section and the ground truth differential expression statistics across layers in all 12 sections. Notably, the Bayes factor produced by spaVAE outperforms the other methods significantly (as shown in Fig. 2h for layer 1 versus the other layers in the DLPFC section 151673). This result underscores the importance of incorporating spatial information, and of the capability of spaVAE to leverage the entire dataset for inference. More differential expression results are summarized in Supplementary Figs. 8–10.

Third, we illustrate the functionality of spatial interpolation, which is an appealing feature of the Gaussian process prior. In Fig. 2i we present the results from a masking experiment. We masked a varying proportion (5–20%) of data and used the remaining spots as the training data. After training the spaVAE model, we interpolated the latent representations of the masked spots by using their spatial locations. We used an 11-nearest neighbors (11-NN) predictor based on the training spots to predict the labels of the masked spots and evaluated the accuracy (the results of the 5–15-NN predictors are summarized in Supplementary Fig. 11). As shown in Fig. 2i, the accuracy across the 12 sections is excellent, and robust (>90%) against an increasing percentage of masked locations. This shows that the Gaussian process prior can perform precise interpolation analysis. The latent representation of training and interpolated masked spots for the DLPFC section 151673 using different masking proportions is shown in Supplementary Fig. 12. In Fig. 2j we show the interpolated counts of the layer-marker genes[27] *MOBP*, *PCP4* and *SNAP25* in the DLPFC section 151673 with 20% of spots masked. The plot demonstrates that the interpolation could restore underlying spatial patterns of gene expression. We also show the interpolated counts of layer-marker genes with other masking proportions in Supplementary Fig. 13. Additionally, we conducted the masking experiment using squares of various sizes and observed similar results (Supplementary Figs. 14–16). We assessed the correlation between the interpolated counts and the real counts and found that it is robust even as the proportion of masking and the area of the square are increased (Supplementary Fig. 17).

Finally, we assess the computational efficiency and robustness of spaVAE (Supplementary Figs. 18–24). Taken together, we applied spaVAE to the DLPFC dataset, and show that spaVAE could achieve good performance in the analysis of visualization, clustering, denoising, differential expression, batch integration and spatial interpolation.

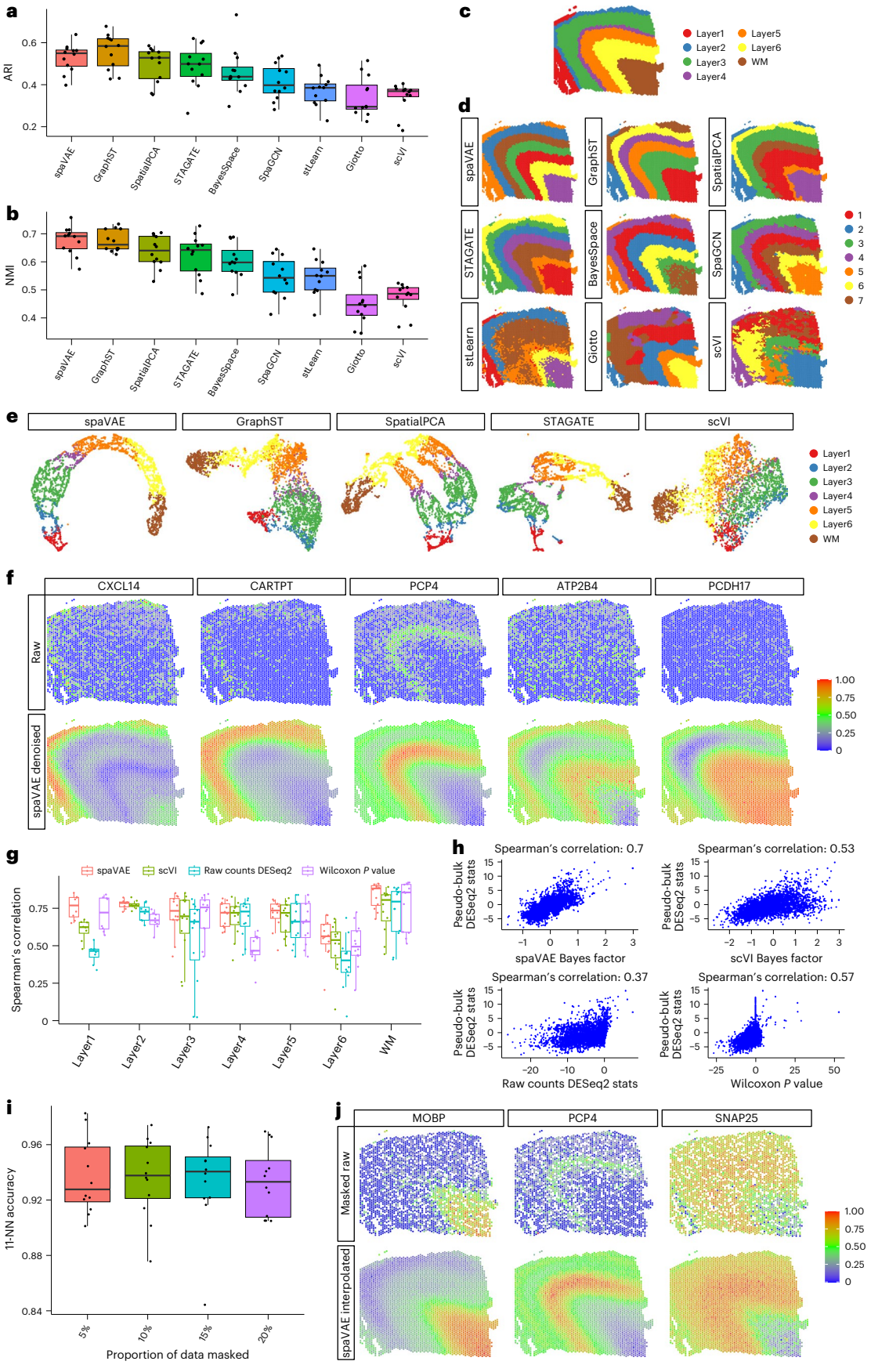
spaVAE for complex spatial transcriptomics data

The mouse hippocampus Slide-seq V2 dataset[31] contains more than 50,000 spots and 20,000 genes. We filtered the genes and selected the top 3,000 SVGs for the analysis. We trained the spaVAE model on the dataset and conducted k-means clustering on the learned embedding (Supplementary Fig. 25). Figure 3a shows the structure of the mouse hippocampus as annotated using the Allen Brain Atlas[32]. In Fig. 3b we compare clustering results obtained from spaVAE and other competing methods capable of processing large datasets. We observed that the clustering labels predicted by spaVAE identified most spatial domains, including cortical layers (clusters 2, 16), corpus callosum (cluster 6), CA1 (cornu ammonis 1; cluster 14), CA3 (cluster 4), dentate gyrus (cluster 9), hippocampus regions (clusters 3, 5, 17), thalamus regions (clusters 1, 8, 10) and the third ventricle (cluster 7). In contrast, GraphST and STAGATE did not identify the third ventricle, and SpaGCN failed to correctly identify CA1 and CA3. Next, we applied RCTD[33] (robust cell type decomposition) to the dataset and deconvoluted the cell type proportions of every spot. As shown in Fig. 3c, spaVAE is one of the best methods, and its clustering results produce a high purity of cell types (JS divergence < 0.25, Supplementary Fig. 26 and Supplementary

Fig. 2 | Application of spaVAE to the LIBD human DLPFC data. a,**b**, Clustering accuracy across 12 samples using various methods, quantified using ARI (**a**) and NMI (**b**), with dots representing individual samples. **c**, Ground truth labels for brain regions in the DLPFC section 151673, manually annotated in the original paper. **d**, Predicted clustering labels by different methods for the DLPFC section 151673, with colors denoting cluster labels. **e**, Latent representations of the DLPFC section 151673 learned by GraphST, spaVAE, SpatialPCA, STAGATE and scVI. Representations are reduced by UMAP, with colors indicating ground truth labels. **f**, Raw counts and spaVAE denoised counts of the regional marker genes in the DLPFC section 151673, with colors representing relative expression levels. **g**, Evaluation of region-specific (one region versus the others) differential expression analysis in the 12 samples by spaVAE, scVI, DESeq2 and two-tailed Wilcoxon test, quantified using Spearman's correlation between the statistics of the different methods and the DESeq2 statistics of the pseudo-bulk data. Pseudo-bulk data are an aggregate of the read counts in each region across the 12 samples. **h**, Differential expression analysis of neural layer 1 versus the others in the DLPFC section 151673, with points representing individual genes (n = 3,000). Statistics (log-transformed Bayes factors for spaVAE and scVI, Wald statistics for DESeq2, signed $\log_{10}$-transformed adjusted P values for two-tailed Wilcoxon test) are compared with the DESeq2 statistics of the pseudo-bulk data. **i**, Accuracy of the 11-NN predictor to predict the labels of masked spots in 12 samples, by proportion of spots masked. The 11-NN predictor exploits the latent representations interpolated by spaVAE and the ground truth labels of the training spots to predict the labels of the masked spots. Dots represent samples. **j**, Interpolation results of the marker genes in the DLPFC section 151673 with 20% of spots masked. The first row shows the raw counts, with masked positions denoted as white spots; the second row shows the counts interpolated by spaVAE. Colors represent relative expression levels. All boxplots are standard boxplots: the ends of the box represent the interquartile range (IQR, first to third quartile); the horizontal line indicates the median; and the whiskers indicate 1.5-fold the IQR.

https://doi.org/10.1038/s41592-024-02257-y

Note 5). For instance, cluster 14 is enriched in CA1 cells, cluster 4 is enriched in CA3 cells, cluster 9 is enriched in dentate cells, and cluster 7 is enriched in choroid cells. Finally, we conducted differential expression analysis and show the top differentially expressed genes for different clusters in Fig. 3d. As expected, cluster 14 was the CA1 region, and we identified the CA1 marker gene *Wfs1*; cluster 9 was the dentate



https://doi.org/10.1038/s41592-024-02257-y

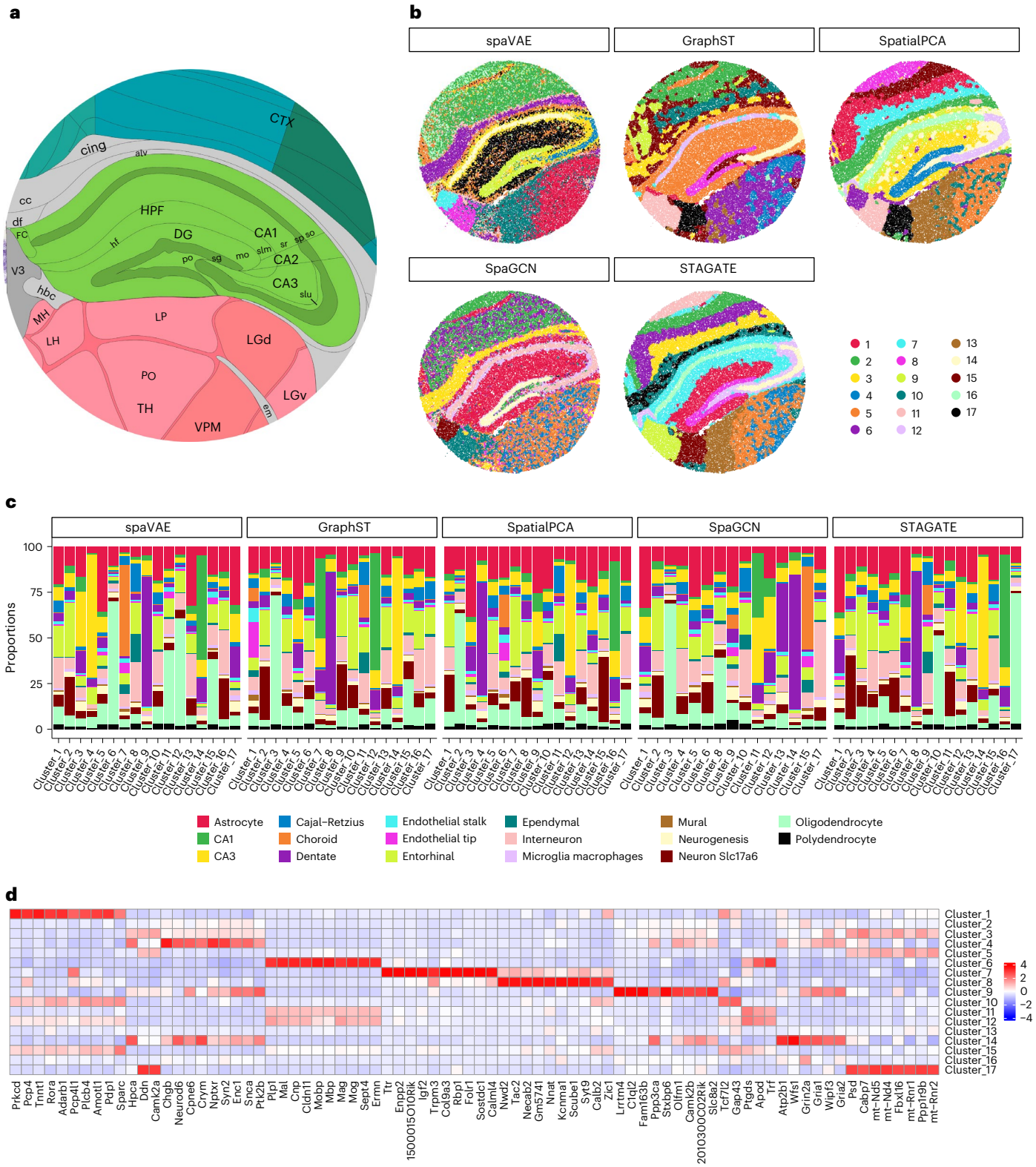


Fig. 3 | Application of spaVAE to the mouse hippocampus Slide-seq V2 data. **a**, An illustration of the structure of the mouse hippocampus, annotated from the Allen Brain Atlas[32]. alv, alveus; CA1–3, cornu ammonis 1–3; cc, corpus callosum; cing, cingulum bundle; CTX, cerebral cortex; df, dorsal fornix; DG, dentate gyrus; em, external medullary lamina of the thalamus; FC, fasciola cinerea; hbc, habenular commissure; hf, hippocampal fissure; HPF, hippocampal formation; LGd, dorsal part of the lateral geniculate complex; LGv, ventral part of the lateral geniculate complex; LP, lateral posterior nucleus of the thalamus; mo, somatomotor areas; po, dentate gyrus, polymorph layer; PO, posterior complex of the thalamus; sg, dentate gyrus, granule cell layer; slm, stratum lacunosum-moleculare; so, stratum oriens; sp, pyramidal layer; sr, stratum radiatum; slu, stratum lucidum; TH, thalamus; V3, third ventricle; VPM, ventral posteromedial nucleus of the thalamus. **b**, Clustering results of different methods, with colors denoting cluster labels. **c**, Cell type proportions inferred by RCTD in clusters as predicted by different methods. **d**, Top genes (log fold change > 1 and Bayes factor > 10) identified by spaVAE in clusters. Heatmap shows the relative denoised averaged expression levels across clusters.

region, and we identified the dentate marker gene *C1ql2*. Furthermore, we performed pre-ranked gene set enrichment analysis (GSEA)[34] based on the Bayes factors inferred by spaVAE, as shown in Supplementary Fig. 27, and found that the top upregulated pathways were related to synapse and neural signaling. This shows that spaVAE can be used to identify complex spatial structures in large datasets, as well as marker genes in the detected spatial regions.

Next, we demonstrated that spaVAE can perform batch integration of the mouse anterior and posterior brains (Extended Data Fig. 1, Supplementary Figs. 28 and 29, and Supplementary Note 6).

spaLDVAE for identifying spatially variable genes

Identifying SVGs is a critical step in the SRT data analysis[35,36]. For this experiment we used spaLDVAE to identify SVGs. The significance cut-off to select SVGs is determined by the random permutation of spatial locations. First, we evaluated spaVAE on simulated datasets with various numbers of SVGs and concluded that spaVAE can outperform competing methods such as SpatialDE[35] and SPARK[36], and control the false discovery rate at a reasonable level (Supplementary Fig. 30). We next applied spaLDVAE to the human DLPFC dataset. We observed clear spatial patterns for SVGs (Extended Data Fig. 2), while the expression levels of the top nonspatial genes were dispersed randomly among spatial spots (Extended Data Fig. 3). This observation was supported by the evaluation of Moran's *I* for both spatially and nonspatially variable genes, with SVGs having a significantly higher Moran's *I* statistic (Supplementary Fig. 31 and Supplementary Note 3). Next, we applied spaLDVAE to the mouse hippocampus Slide-seq V2 dataset and observed clear spatial gene expression patterns (Supplementary Fig. 32).

spaVAE for enhancing resolutions in spatial transcriptomics data

Current next-generation sequencing-based SRT data cannot provide gene expression profiles at single-cell resolution. Due to this technological limitation, the spot size is usually greater than the size of a single cell. For example, in the 10X Visium platform, one spot typically contains tens of cells. Thus, there is a need for the computational model to enhance spatial resolution. For this purpose, we applied spaVAE to two SRT datasets: mouse olfactory bulb data[37] and HER2 breast tumor data[38]. Figure 4a shows the hematoxylin and eosin (H&E)-stained image of one of the 12 samples from the mouse olfactory bulb dataset. The H&E images of all 12 samples are given in Supplementary Fig. 33. As shown, there are clear spatial patterns in the H&E images. We show the raw counts and spaVAE-enhanced counts of some of the marker genes in Fig. 4b. The marker gene list was obtained from a previous study[39]. Additional results are shown in Supplementary Figs. 34 and 35. For each spot, we enhanced the resolution into five subspots by spaVAE. As the Figures show, spatial patterns are vague in raw counts, but we can see clear patterns consistent with the H&E image after spatial enhancement.

In the HER2 breast tumor data, Andersson et al. provided tissue labels annotated by a pathologist[38] (Fig. 4c). We trained spaVAE on the dataset and conducted k-means clustering on the latent representations. The clustering results of spaVAE and other baseline methods are shown in Fig. 4d, and their quantified clustering performance in Fig. 4e. We observed that spaVAE produced superior clustering results compared with SpatialPCA and BayesSpace. We also repeated the training process of spaVAE ten times and found that the clustering results were robust against different random initiations (Supplementary Fig. 36). The spatially enhanced result is plotted in Fig. 4f and, as shown, after enhancement spaVAE produces more sophisticated spatial patterns, which is consistent with the H&E image. Using the Bayes factor and log fold change inferred by spaVAE, we detected the top upregulated genes in the clusters (Fig. 4g). For example, cluster 2 was enriched in *CXCL14*, which is a marker of adipose tissue. *LYZ* and *B2M* are significantly upregulated in cluster 1, which is expected because they are immunity-related genes. *ITGB6* and *ERBB2* are breast tumor marker genes enriched in cluster 3. Figure 4h compares the raw counts and spaVAE-enhanced counts of these marker genes. We conclude that the spatial enhancement function of spaVAE can significantly improve the spatial patterns of these marker genes. The results of other samples from the dataset are summarized in Supplementary Figs. 37–43. Taken together, spaVAE is the first deep generative model that deals with the discrete SRT count directly and can enhance the spatial resolution of both latent representations and gene expressions.

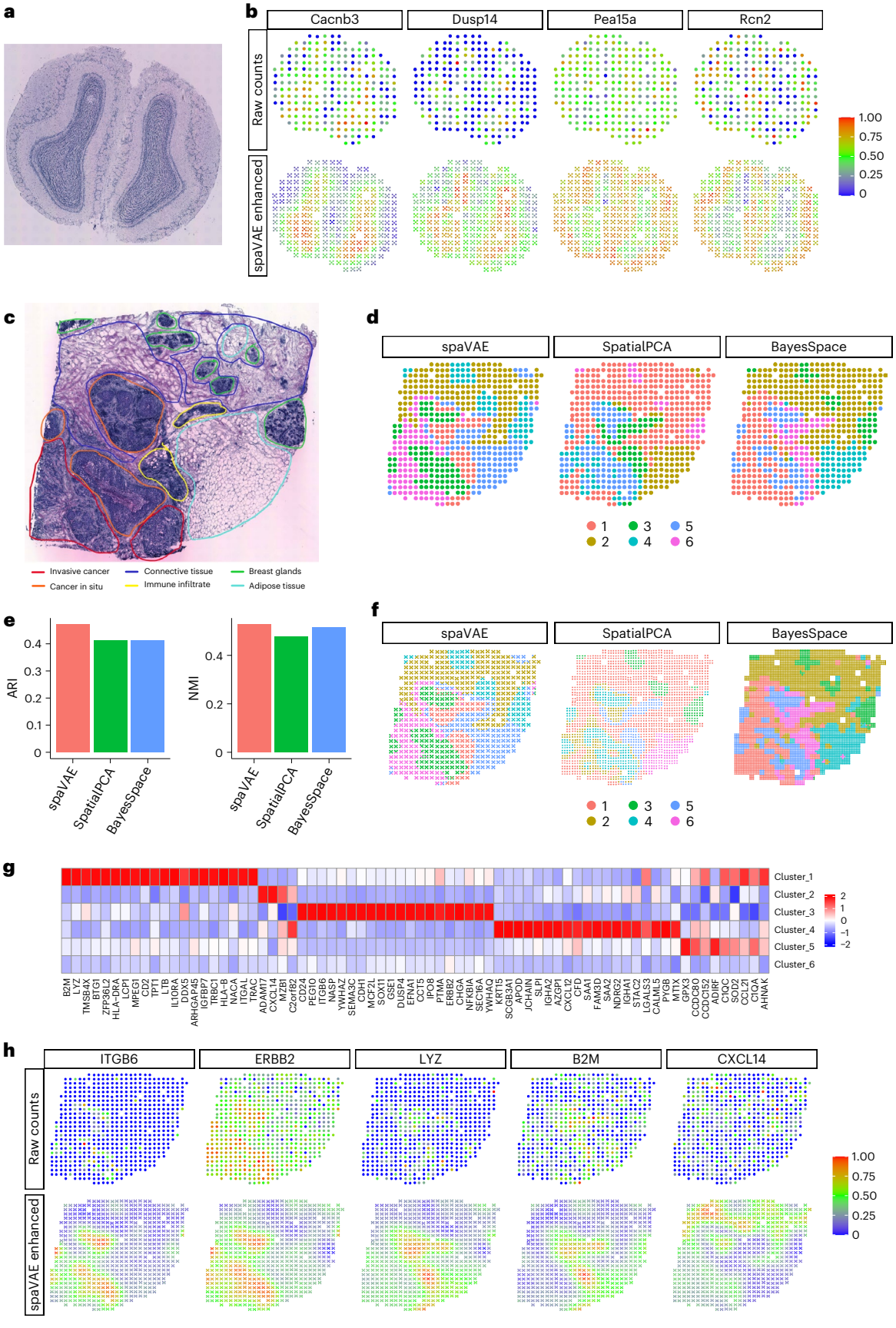
spaPeakVAE for spatial bimodal ATAC-seq data

We applied spaPeakVAE to a spatial bimodal ATAC-seq dataset of mouse embryonic (E15.5) brain. A new sequencing technology, MISAR-seq, was recently introduced to jointly profile chromatin accessibility and gene expression with location information[21]. That study measured ATAC and messenger RNA counts in mouse embryonic brain across various regions (Fig. 5a and Supplementary Fig. 44a,b). First we trained the spaVAE model and obtained a latent representation of the mRNA data. We observed that spaVAE's embedding of mRNA could differentiate various brain regions, such as the forebrain, midbrain and hindbrain (Supplementary Fig. 44c). Next, we evaluated spaPeakVAE's performance by comparing it with several computational models for the ATAC data, such as PeakVI[25], SCALE[24], cisTopic[40], chromVAR[41] and LSA (latent semantic analysis)[42]. All of these competing methods are designed for regular single-cell ATAC-seq data and are not capable of utilizing spatial information. We observe that the latent representation of spaPeakVAE can separate different labels well, but forebrain, midbrain and hindbrain are always entangled in the embeddings of the competing methods (Fig. 5b). We quantified the similarities between different methods' embeddings of ATAC data and spaVAE's embedding of mRNA data, for a range of k-NN similarity scores (Fig. 5c). We observe that the embedding of spaPeakVAE has the best similarity scores, suggesting that spaPeakVAE's embedding is the closest to the mRNA embedding. This supports the substantial contribution of spatial information utilized by spaPeakVAE. Figure 5d shows the top spatially and nonspatially variable peaks identified by spaPeakLDVAE for the ATAC data, and we can see obvious spatial patterns in the top spatial peaks. More top spatial and nonspatial peaks are shown in Supplementary Fig. 45. We also applied the denoising and resolution enhancement functionalities to the top spatial peaks and found that spatial patterns became much clearer in these peaks after being processed by spaPeakVAE (Supplementary Fig. 46). Like its spaVAE counterpart, spaPeakVAE can perform differential accessibility analysis of the spatial ATAC-seq data. We conducted differential accessibility analysis of forebrain versus midbrain and compared the differential accessibility results with the corresponding bulk ATAC-seq data. Again, we observe notable superiority of spaPeakVAE's differential accessibility result

Fig. 4 | spaVAE for enhancing spatial resolution. a, H&E staining image of mouse olfactory bulb tissue. **b**, Raw counts and spaVAE resolution-enhanced counts of the marker genes in the mouse olfactory bulb data. **c**, H&E staining image of the HER2 tumor data, with distinct tissue regions manually annotated by a pathologist in the original paper[38]. **d**, Clustering labels predicted by different methods, with colors denoting cluster labels. **e**, Clustering performance, quantified using ARI and NMI for different methods. **f**, Resolution-enhanced clustering results of different methods. **g**, Top genes (log fold change > 0.5 and Bayes factor > 3.2) identified by spaVAE within clusters. The heatmap shows the relative denoised averaged expression levels across clusters. **h**, Enhanced resolution of expression levels of the top cluster marker genes. **a** and **b** are based on the analysis of mouse olfactory bulb data, and **c**–**h** are based on the analysis of HER2 breast tumor data.

https://doi.org/10.1038/s41592-024-02257-y

compared with the differential accessibility analysis using PeakVI, DESeq2 and Wilcoxon signed-rank test (Fig. 5e). More differential accessibility analysis results (forebrain versus hindbrain, midbrain versus hindbrain) are summarized in Supplementary Fig. 47. To further validate the differential accessibility analysis of spaPeakVAE, we used MEME-ChIP, a web-based tool[43], to analyze the significant peak regions



https://doi.org/10.1038/s41592-024-02257-y

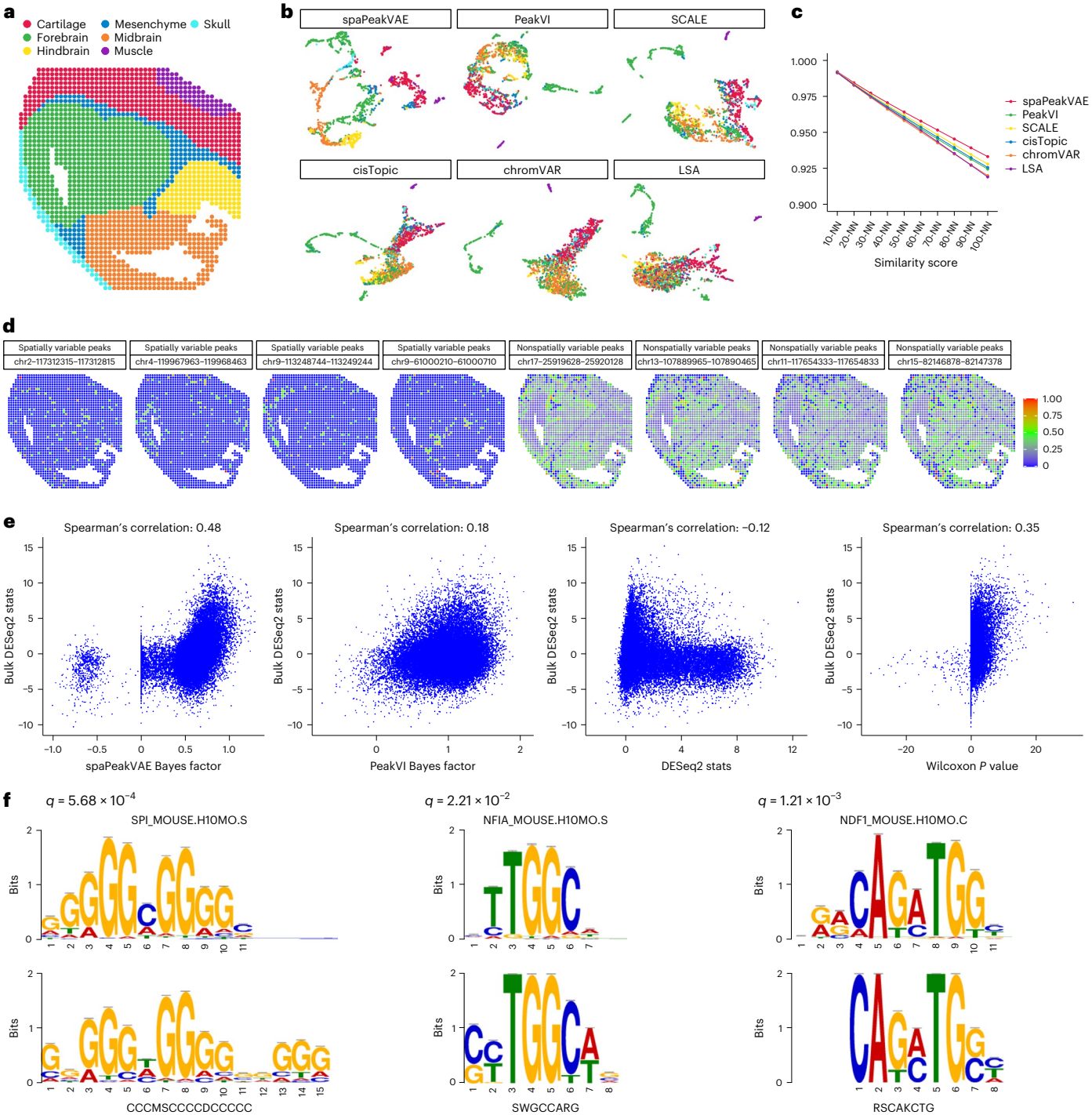


Fig. 5 | Application of spaPeakVAE to the spatial ATAC-seq data. a, Manually annotated labels of mouse embryonic (E15.5) brain tissues in the MISAR-seq dataset. **b**, 2D UMAP visualization of latent representations of ATAC data learned by spaPeakVAE, PeakVI, SCALE, cisTopic, chromVAR and LSA, with colors denoting manually annotated labels. **c**, Quantification of consistency between the latent representations and the corresponding mRNA embedding, measured by a range of k-NN similarity scores. **d**, Top four spatially variable and nonspatially variable peaks identified by spaLDVAE. Colors represent relative ATAC counts. **e**, Differential accessibility analysis of forebrain versus midbrain. Points represent individual peaks (n = 30,136). Statistics (log-transformed Bayes factor for spaPeakVAE and PeakVI, Wald statistics for DESeq2, signed $\log_{10}$-transformed adjusted P values for two-tailed Wilcoxon test) are compared with the DESeq2 statistics of the corresponding bulk ATAC data. **f**, Sequence motifs identified in the significant peak regions (log fold change > 1 and Bayes factor > 5) of spaPeakVAE differential accessibility analysis of forebrain versus midbrain. The upper panel shows the known motifs in mouse, and the bottom panel shows the de novo motifs identified from the significant peak regions. q values for the match between the known and de novo motifs are also given.

(log fold change > 1 and Bayes factor > 5 inferred by spaPeakVAE) to identify enriched motifs. Figure 5f shows motifs identified from these significant peaks and their significant matches to known motifs in mouse. We noted that the motif 'NFIA_MOUSE' was enriched in the top peaks, and the transcription factor *Nfia* acts upstream of or takes part in synapse maturation[44]. We also observe the motif 'NDF1_MOUSE', known as neurogenic differentiation 1. It is upregulated in the forebrain and uses chromatin to enhance regulatory elements in genes that encode

https://doi.org/10.1038/s41592-024-02257-y

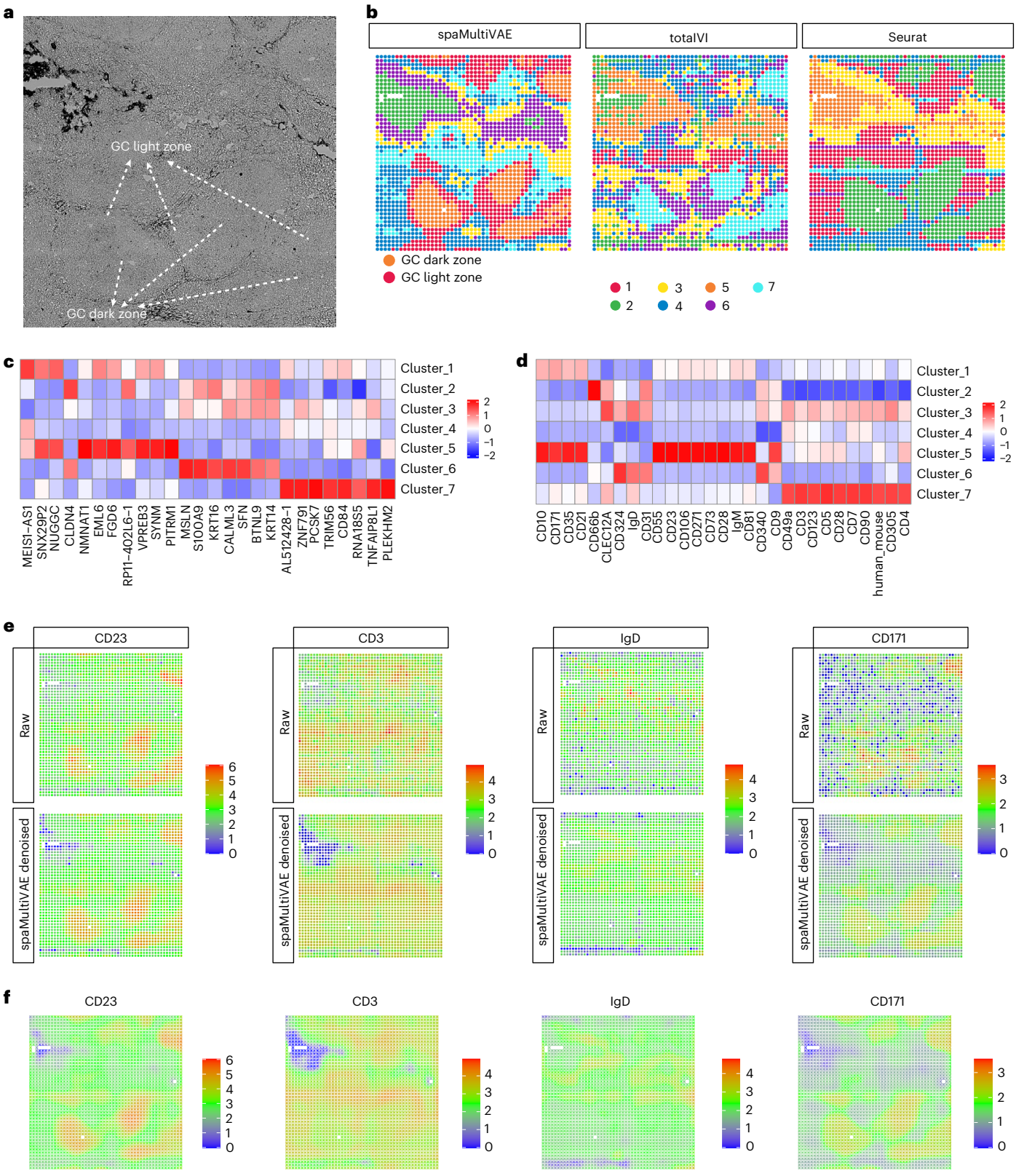


Fig. 6 | Application of spaMultiVAE to the spatial-CITE-seq data. a, Tissue image of a human tonsil sample. The GC light and dark zones are marked. **b**, Clustering results of spaMultiVAE, totalVI and Seurat, with colors denoting cluster labels. **c**,**d**, Top genes (**c**) and top proteins (**d**) identified by spaMultiVAE (log fold change > 0.5 and Bayes factor > 3.2) in different clusters. The heatmaps show the relative denoised averaged expression levels across clusters. **e**, Raw and spaMultiVAE denoised protein expression levels. **f**, Enhanced resolution of expression levels of top cluster marker proteins.

key transcriptional regulators of neurogenesis[45]. These results are expected because these peak regions were prioritized in the differential accessibility analysis of forebrain versus midbrain. Next, we compared motifs enriched in the significant peaks from spatial ATAC-seq data with motifs identified in significantly up-accessible peaks from the corresponding bulk ATAC-seq data (Supplementary Fig. 48). We found

clearly similar motif patterns from the two data sources. Based on these analyses, we conclude that spaPeakVAE can effectively characterize spatial ATAC-seq data.

spaMultiVAE for spatial multi-omics data

Recent advancements in sequencing technology permit the profiling of both gene expression and surface protein intensity in the same cell or spot in the spatial genomics data[22,23]. Unlike mRNA sequencing, protein profiling requires the barcode-tagged antibody to bind to the specific protein. However, the binding is not always specific, and this may result in an unfavorable background intensity that needs to be eliminated. For this purpose, we designed spaMultiVAE, which models the mixture of background and foreground protein intensity. We first applied spaMultiVAE to an annotated CITE-seq dataset of spleen tissue to illustrate the performance of background denoising[26] (Supplementary Fig. 49 and Supplementary Note 7).

Next, we applied spaMultiVAE to analyze the spatial-CITE-seq data of human tonsil tissue[22]. Initially, we used a dataset with SVGs and all proteins. We compared our spaMultiVAE model with the totalVI model, which was a deep generative model for the regular CITE-seq data. We applied k-means clustering to the latent embedding of the two methods to find seven clusters. The weighted nearest neighbor (WNN) function in the Seurat package[3], which integrates gene and protein modalities, was also compared. Shown in Fig. 6a,b and Supplementary Fig. 50, the clustering result of spaMultiVAE has a smoother spatial pattern and better concordance with the tissue image than does totalVI and Seurat. The subpar clustering performance of totalVI underscores the importance of incorporating spatial information. Although superior to totalVI, Seurat still lags behind spaMultiVAE. Notably, Seurat fails to identify the dark zone of the germinal center, which corresponds to cluster 5 as reported by spaMultiVAE clustering. It also exhibits some noise in spatial patterns of clusters. To assess the robustness of the stochastic training process in spaMultiVAE, we conducted five repetitions of the training and show the k-NN similarity scores in Supplementary Fig. 51 (k-NN similarity scores ≈ 99%). We conducted differential expression analysis of both genes and proteins using spaMultiVAE to identify the top genes and proteins in each cluster (Fig. 6c,d). As expected, clusters 1 and 5 are the germinal center light and dark zones, where we observe the upregulated marker proteins of germinal center B cells in cluster 5, including *CD23* and *IgM*. *CD171*, a neuronal cell adhesion molecule highly restricted to the dark zone germinal center, is also upregulated in cluster 5. Cluster 7 indicates specific T-cell zones, with T-cell marker proteins such as *CD3* and *CD5* being upregulated in this cluster. Figure 6e shows both the raw and the spaMultiVAE denoised protein intensities. spaMultiVAE is observed to not only correct the background noise in the protein intensities but also to improve the spatial patterns. Figure 6f presents spatial resolution enhancement of proteins by spaMultiVAE. So far, all analyses have been based on SVGs. We also evaluated spaMultiVAE on the dataset with HVGs and observed the same superiority of spaMultiVAE, as shown in Supplementary Fig. 52. The pre-ranked GSEA analysis based on gene Bayes factors is summarized in Supplementary Fig. 53. Next, we applied spaLDVAE to the protein part of this spatial-CITE-seq data to identify spatially variable proteins. In Supplementary Fig. 54, we observed clear spatial patterns in spatially variable proteins, and these patterns became even clearer after spaMultiVAE denoising. With the cut-off obtained through random permutation, 16 proteins meet the significance threshold, and the Moran *I* of these spatial proteins is significantly larger than that of nonspatial proteins (Supplementary Fig. 55). In conclusion, spaMultiVAE could efficiently elucidate the spatial structures and remove undesired background noise in the spatial multi-omics data.

Finally, we applied spaMultiVAE to another set of spatial multi-omics data: the DBiT-seq data of the mouse embryonic brain region[23], and observed similar improvements (Supplementary Figs. 56–58).

Discussion

spaVAE, spaPeakVAE and spaMultiVAE are deep generative models that account for spatial dependency and enable automatic optimization of the strength of the dependency from the data. Our models characterize the discrete count data directly by model-based loss functions. Using various types of spatial omics data (for example, spatial transcriptomics, spatial ATAC-seq, spatial multi-omics) from various platforms and tissues, we demonstrate that our proposed models achieve strong performance in various analytical tasks, including visualization, clustering, differential expression and accessibility (Figs. 2–6), denoising (Fig. 2), batch integration (Extended Data Fig. 1), spatial interpolation, and resolution enhancement (Fig. 4). These analytical steps are critical for uncovering new biological insights from spatial data, such as new subpopulations and novel marker genes. Although our models may not rank as the best performers in every task, they offer a uniform framework that leverages spatial information, demonstrating good performance and the potential to provide enhanced benefits for various analytical tasks. Additionally, we introduced spaLDVAE and spaPeakLDVAE, variants of the spaVAE and spaPeakVAE models that include a linear decoder, specifically designed for detecting SVGs and spatially variable peaks. Furthermore, we evaluated the spaVAE model on simulated datasets with known ground truth labels. Our findings suggest that it outperforms competing methods, as shown in Supplementary Figs. 59–63 and detailed in Supplementary Note 8.

SpaVAE and spaMultiVAE take the discrete count data as input and eliminate undesired technical biases, including but not limited to sequencing library size, batch effects, and protein background intensity. By contrast, spaPeakVAE is configured to directly process binary input, also considering sequencing biases for both spots and peaks. These features make our models an end-to-end analytical solution for the analysis of spatial omics data. Preprocessing and normalization are usually critical steps in genomics data analysis, and the performance of many analytical tools relies heavily on these steps. Our results demonstrate that spaVAE, spaPeakVAE and spaMultiVAE deal with these challenges effectively by modeling the input directly.

The computation for the covariance matrix in the Gaussian process is very time-consuming. To accelerate this process we apply a sparse Gaussian process regression technique, which relies on the inducing points. The incorporation of inducing points could reduce the computational complexity from cubic to linear as a function of sample size. The number of inducing points controls the rank of the covariance matrix between samples. If an SRT dataset has complex spatial structures, more inducing points are required. Although several experiments have demonstrated that spaVAE is robust to the choice of different numbers of inducing points, our current models still require users to specify the number of inducing points. In addition, it should be noted that the proposed models, like other deep learning-based models, perform well with large or medium-sized datasets, but training can be challenging for small datasets, for example, those with 500 spots or fewer. Finally, a large amount of spatial data may be generated alongside microscopy imaging data, which is expected to provide useful complementary information but which has not yet been incorporated into our models. We will address these limitations in future work, for example, by developing a method to estimate the optimal number of inducing points from the data and by exploring ways to integrate microscopy imaging data or other types of information into our models.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-024-02257-y.

References

1. Asp, M., Bergenstrahle, J. & Lundeberg, J. Spatially resolved transcriptomes: next generation tools for tissue exploration. *Bioessays* **42**, e1900221 (2020).
2. Rao, A., Barkley, D., Franca, G. S. & Yanai, I. Exploring tissue architecture using spatial transcriptomics. *Nature* **596**, 211–220 (2021).
3. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 (2021).
4. 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).
5. 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).
6. Tian, T., Wan, J., Song, Q. & Wei, Z. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nat. Mach. Intell.* **1**, 191–198 (2019).
7. Tian, T., Zhang, J., Lin, X., Wei, Z. & Hakonarson, H. Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. *Nat. Commun.* **12**, 1873 (2021).
8. Hu, J. et al. SpaGCN: integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nat. Methods* **18**, 1342–1351 (2021).
9. Long, Y. et al. Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with GraphST. *Nat. Commun.* **14**, 1155 (2023).
10. Dong, K. & Zhang, S. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nat. Commun.* **13**, 1739 (2022).
11. Lin, X., Gao, L., Whitener, N., Ahmed, A. & Wei, Z. A model-based constrained deep learning clustering approach for spatially resolved single-cell data. *Genome Res.* **32**, 1906–1917 (2022).
12. Pham, D. et al. stLearn: integrating spatial location, tissue morphology and gene expression to find cell types, cell-cell interactions and spatial trajectories within undissociated tissues. *Nat Commun.* **14**, 7739 (2023).
13. Shang, L. & Zhou, X. Spatially aware dimension reduction for spatial transcriptomics. *Nat. Commun.* **13**, 7203 (2022).
14. Zhao, E. et al. Spatial transcriptomics at subspot resolution with BayesSpace. *Nat. Biotechnol.* **39**, 1375–1384 (2021).
15. Casale, F. P., Dalca, A. V., Saglietti, L., Listgarten, J. & Fusi, N. Gaussian process prior variational autoencoders. In *Proc. 32nd International Conference on Neural Information Processing Systems (NIPS 2018)* (eds Bengio, S. et al.) (Curran Associates, Inc., 2018).
16. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations* (2013).
17. Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. *Proceedings of Machine Learning Research* **5**, 567–574 (2009).
18. Hensman, J., Fusi, N. & Lawrence, N. D. Gaussian processes for big data. In *Proc. 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)* (eds Nicholson, A. & Smyth, P.) (AUAI Press, 2013).
19. Jazbec, M. et al. Scalable Gaussian process variational autoencoders. *Proceedings of Machine Learning Research* **130**, 3511–3519 (2021).
20. Deng, Y. et al. Spatial profiling of chromatin accessibility in mouse and human tissues. *Nature* **609**, 375–383 (2022).
21. Jiang, F. et al. Simultaneous profiling of spatial gene expression and chromatin accessibility during mouse brain development. *Nat. Methods* **20**, 1048–1057 (2023).
22. Liu, Y. et al. High-plex protein and whole transcriptome co-mapping at cellular resolution with spatial CITE-seq. *Nat. Biotechnol.* **41**, 1405–1409 (2023).
23. Liu, Y. et al. High-spatial-resolution multi-omics sequencing via deterministic barcoding in tissue. *Cell* **183**, 1665–1681 (2020).
24. Xiong, L. et al. SCALE method for single-cell ATAC-seq analysis via latent feature extraction. *Nat. Commun.* **10**, 4576 (2019).
25. Ashuach, T., Reidenbach, D. A., Gayoso, A. & Yosef, N. PeakVI: a deep generative model for single-cell chromatin accessibility analysis. *Cell Rep. Methods* **2**, 100182 (2022).
26. Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nat. Methods* **18**, 272–282 (2021).
27. Maynard, K. R. et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nat. Neurosci.* **24**, 425–436 (2021).
28. Dries, R. et al. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome Biol.* **22**, 78 (2021).
29. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: uniform manifold approximation and projection. *Journal of Open Source Software* **3**, 861 (2018).
30. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
31. Stickels, R. R. et al. Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2. *Nat. Biotechnol.* **39**, 313–319 (2021).
32. Lein, E. S. et al. Genome-wide atlas of gene expression in the adult mouse brain. *Nature* **445**, 168–176 (2007).
33. Cable, D. M. et al. Robust decomposition of cell type mixtures in spatial transcriptomics. *Nat. Biotechnol.* **40**, 517–526 (2022).
34. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA* **102**, 15545–15550 (2005).
35. Svensson, V., Teichmann, S. A. & Stegle, O. SpatialDE: identification of spatially variable genes. *Nat. Methods* **15**, 343–346 (2018).
36. Sun, S., Zhu, J. & Zhou, X. Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. *Nat. Methods* **17**, 193–200 (2020).
37. Stahl, P. L. et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).
38. Andersson, A. et al. Spatial deconvolution of HER2-positive breast cancer delineates tumor-associated cell type interactions. *Nat. Commun.* **12**, 6012 (2021).
39. Bergenstrahle, L. et al. Super-resolved spatial transcriptomics by deep data fusion. *Nat. Biotechnol.* **40**, 476–479 (2022).
40. Bravo Gonzalez-Blas, C. et al. cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nat. Methods* **16**, 397–400 (2019).
41. Schep, A. N., Wu, B., Buenrostro, J. D. & Greenleaf, W. J. chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nat. Methods* **14**, 975–978 (2017).
42. Dumais, S. T. Latent semantic analysis. *Annu. Rev. Inf. Sci. Technol.* **38**, 188–230 (2005).
43. Machanick, P. & Bailey, T. L. MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics* **27**, 1696–1697 (2011).
44. Wong, Y. W. et al. Gene expression analysis of nuclear factor I-A deficient mice indicates delayed brain maturation. *Genome Biol.* **8**, R72 (2007).
45. Tutukova, S., Tarabykin, V. & Hernandez-Miranda, L. R. The role of neurod genes in brain development, function, and disease. *Front. Mol. Neurosci.* **14**, 662774 (2021).

Methods

Dependency-aware variational autoencoder

Unlike the classical VAE, spaVAE, spaPeakVAE and spaMultiVAE are VAEs with hybrid latent embeddings sampled from both Gaussian process and standard Gaussian priors. The classical VAE has a fully factorized prior[16], which assumes that samples (spots) are independent. To account for the spatial information, spaVAE, spaPeakVAE and spaMultiVAE use a Gaussian process prior to model the dependency between spots. It is also important to acknowledge that not all variances in the data are spatially dependent. As a result, our models also integrate a standard Gaussian prior. If we denote the number of dimensions of latent representations as D, the first L dimensions follow a Gaussian process prior, and the other $D-L$ dimensions follow a standard Gaussian prior. Next, we denote the sample size as N, the spatial locations as $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times 2}$ (typically N by 2), and the gene, peak or protein count matrix as $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T \in \mathbb{N}^{N\times G}$ (N by G, G is the number of features). Our objective is to learn a model that could infer low-dimensional latent representation $\mathbf{z}$ and generate $\mathbf{y}$ conditioned on auxiliary data $\mathbf{x}$. In the framework of VAE, the evidence lower bound (ELBO) can be written as:

$$\mathrm{ELBO} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p(\mathbf{y}|\mathbf{z})] - \beta \mathrm{KL}(q(\mathbf{z}|\mathbf{x},\mathbf{y})||p(\mathbf{z})), \tag{1}$$

where $\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p(\mathbf{y}|\mathbf{z})]$ can be considered as the reconstruction loss, and we use β to control the weight of Kullback–Leibler (KL) divergence loss[46]. To characterize the gene, peak and protein count data, we use different types of model-based reconstruction loss functions.

In the same way as the classical VAE, our spatial-aware VAEs estimate the latent mean and the latent variance by the encoder networks

$$\tilde{\mathbf{w}} = f_w(f(\mathbf{y})),$$

$$\tilde{\Phi}^2 = \exp(f_\Phi(f(\mathbf{y})),$$

where f_w, f_ϕ are two neural networks that parameterize mean and variance, f is the latent encoder, and $\tilde{q}(\tilde{\mathbf{z}}|\mathbf{y}) = N(\tilde{\mathbf{w}}, \tilde{\Phi}^2)$. Here we use the exponential as the activation function for the variance, because it is always positive. The Gaussian process regression is applied to the first L dimensions of latent mean $\tilde{\mathbf{w}}$ and the latent variance $\tilde{\Phi}^2$ to characterize the spatial dependency (denoted as $\tilde{\mathbf{w}}^{1:L}$ and $\tilde{\Phi}^{1:L^2}$), and the other $D-L$ dimensions of $\tilde{\mathbf{w}}$ and $\tilde{\Phi}^2$ follow a standard Gaussian prior (denoted as $\tilde{\mathbf{w}}^{L+1:D}$ and $\tilde{\Phi}^{L+1:D^2}$). As a result, the KL loss contains two components: the Gaussian process and Gaussian, and the ELBO can be written as

$$\begin{aligned}\mathrm{ELBO} = \mathbb{E}_{(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p(\mathbf{y}|\mathbf{z})] - \beta\big[\mathrm{KL}\left(q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y})||p(\mathbf{z}^{1:L})\right) \\ +\mathrm{KL}\left(q\left(\mathbf{z}^{L+1:D}|\,\mathbf{y}\right)||p\left(\mathbf{z}^{L+1:D}\right)\right)\big].\end{aligned} \tag{2}$$

We will derive the equations of KL loss for the two parts in the following sections.

Gaussian process prior

The prior distribution of the Gaussian process (GP) latent embedding $\mathbf{z}^{1:L}$ is

$$p\left(\mathbf{z}^{1:L}|\mathbf{x}\right) = \mathrm{GP}\left(0, \mathbf{K}_{NN}\right),$$

where $\mathbf{K}_{NN}$ is the covariance matrix of the GP prior: $\mathbf{K}_{NN} = k_\theta(\mathbf{x},\mathbf{x})$, and k_θ is the kernel function. If setting $\mathbf{K}_{NN} = \mathbf{I}$, then it will reduce to a traditional variational autoencoder.

Different kernel functions can be used in our model to capture different spatial dependencies. Four kernel functions have been considered (Supplementary Note 9) and their effects are shown in Supplementary Figs. 22–24. As shown, spaVAE does not appear to be very sensitive to the choice of these kernels, while the Cauchy and EQ (Gaussian) kernels achieve the highest clustering accuracy. For numerical stability, spaVAE defaults to the Cauchy kernel. The Cauchy kernel between spots i and j is defined as

$$k_\theta(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + \|\mathbf{x}_i - \mathbf{x}_j\|^2/\theta},$$

where θ is a trainable parameter to adapt dependency strength from data, and $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ represents the Euclidean distance between spots.

The GP embedding part can be considered as a form of GP variational autoencoder (GP-VAE)[19]. Given the outputs of encoder $\tilde{\mathbf{z}}^{1:L}$, GP-VAE combines latent representation learning and GP regression to generate $\mathbf{z}^{1:L}$. Following Pearce[47], we can rewrite the KL loss of GP embedding as

$$\mathrm{KL}(q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y})||p(\mathbf{z}^{1:L})) = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{z}^{1:L}|\mathbf{x}_i,\,\mathbf{y}_i)}\left[\log \tilde{q}(\tilde{\mathbf{z}}^{1:L}|\,\mathbf{y}_i)\right] - \sum_{l=1}^{L} \log Z^l(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}), \tag{3}$$

where $\tilde{q}(\tilde{\mathbf{z}}^{1:L}|\,\mathbf{y})$ and $q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y})$ represent the posterior distributions of the outputs of the encoder and GP regression, respectively. In the equation, i is the spot index, and $\mathbf{x}_{1:N}$ and $\mathbf{y}_{1:N}$ represent the whole dataset of N spots. The term $Z^l(\mathbf{x}_{1:N}, \mathbf{y}_{1:N})$ represents the marginal likelihood of the GP regression of the l-th latent dimension, and we assume that latent dimensions are independent.

To accelerate the calculation and enable mini-batch stochastic optimization, we apply the sparse GP regression technique, which relies on inducing points[19]. Let us denote the vector of p inducing points as $\mathbf{p}$. Given a set of inputs (spatial locations, inferred latent mean, and variance: $\mathbf{x}, \tilde{\mathbf{w}}, \tilde{\Phi}^2$), following Hensman et al.[18], then the sparse GP regression outputs $\mathbf{f}_\mathbf{p} = N(\mu, \mathbf{A})$. We can calculate the ELBO L_H^l of the marginal likelihood of the sparse GP regression on the l-th latent dimension (in the first L dimensions) on the total N spots:

$$\begin{aligned}&\log Z^l(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}) \geq \mathrm{L}_\mathrm{H}^l \\ &= \sum_{i=1}^{N}\left(\log N\left(\tilde{w}_i^l|\mathbf{k}_i^T\mathbf{K}_{pp}^{-1}\mu^l, \tilde{\Phi}_i^{l2}\right) - \frac{1}{2\tilde{\Phi}_i^{l2}}\left(\tilde{k}_{ii} + \mathrm{tr}\left(\mathbf{A}^l\Lambda_i\right)\right)\right) - \mathrm{KL}(q_s^l(\mathbf{f}_\mathbf{p})||p(\mathbf{f}_\mathbf{p})),\end{aligned} \tag{4}$$

where $\tilde{w}_i^l$ and $\tilde{\Phi}_i^{l2}$ represent the l-th latent dimension of the latent mean $\tilde{\mathbf{w}}$ and the latent variance $\tilde{\Phi}^2$ for spot i respectively; $\boldsymbol{\mu}^l$ and $\mathbf{A}^l$ are the trainable parameters $\boldsymbol{\mu}$ and $\mathbf{A}$ for the l-th latent dimension; $q_s^l(\mathbf{f}_\mathbf{p}) = N(\mathbf{f}_\mathbf{p}|\mu^l, \mathbf{A}^l)$ and $p(\mathbf{f}_\mathbf{p}) = N(\mathbf{f}_\mathbf{p}|0, \mathbf{K}_{pp})$. Here, $\mathbf{K}_{pp} = k_\theta(\mathbf{p},\mathbf{p})$, $\mathbf{k}_i$ represents the i-th column of $\mathbf{K}_{pN} = k_\theta(\mathbf{p},\mathbf{x}_N) = \mathbf{K}_{Np}^T$, $\Lambda_i = \mathbf{K}_{pp}^{-1}\mathbf{k}_i\mathbf{k}_i^T\mathbf{K}_{pp}^{-1}$, and $\tilde{k}_{ii}$ is the i-th diagonal element of $\mathbf{K}_{NN} - \mathbf{K}_{Np}\mathbf{K}_{pp}^{-1}\mathbf{K}_{pN}$. Following refs. 17,19, we can write the stochastic estimates of $\boldsymbol{\mu}$ and $\mathbf{A}$ for each latent dimension l of a mini-batch b of data:

$$\begin{aligned}\overset{l}{\underset{b}{\boldsymbol{\Sigma}}} &= \mathbf{K}_{pp} + \frac{N}{b}\mathbf{K}_{pb}\mathrm{diag}\left(\tilde{\Phi}_b^{l-2}\right)\mathbf{K}_{bp}, \\ \mu_b^l &= \frac{N}{b}\mathbf{K}_{pp}\left(\overset{l}{\underset{b}{\boldsymbol{\Sigma}}}\right)^{-1}\mathbf{K}_{pb}\mathrm{diag}\left(\tilde{\Phi}_b^{l-2}\right)\tilde{\mathbf{w}}_b^l, \\ \mathbf{A}_b^l &= \mathbf{K}_{pp}\left(\overset{l}{\underset{b}{\boldsymbol{\Sigma}}}\right)^{-1}\mathbf{K}_{pp}.\end{aligned} \tag{5}$$

The total GP regression ELBO combines L latent dimensions $\mathrm{L}_H = \Sigma_{l=1}^{L}\mathrm{L}_H^l$, and the stochastic GP regression ELBO on a mini-batch b of data can be obtained based on $\boldsymbol{\mu}_b$ and $\mathbf{A}_b$.

After obtaining the GP regression ELBO, we can infer the posterior of the GP regression. Following Hensman et al.[18], for the l-th latent dimension, we have the posterior distribution of the total N spots:

$$\begin{aligned}q(\mathbf{z}^l|\mathbf{x},\mathbf{y}) &= q\left(\mathbf{z}_{1:N}^l\right) \\ &= N\left(\mathbf{z}_{1:N}^l|\mathbf{K}_{Np}\mathbf{K}_{pp}^{-1}\mu^l, \mathbf{K}_{NN} - \mathbf{K}_{Np}\mathbf{K}_{pp}^{-1}\mathbf{K}_{pN} + \mathbf{K}_{Np}\mathbf{K}_{pp}^{-1}\mathbf{A}^l\mathbf{K}_{pp}^{-1}\mathbf{K}_{pN}\right).\end{aligned} \tag{6}$$

Given a mini-batch b of data, we can write the stochastic estimate of the posterior distribution of latent embedding z^l as

$$\mathbf{m}_b^l = \frac{N}{b}\mathbf{K}_{bp}\left(\sum_b^l\right)^{-1}\mathbf{K}_{pb}\mathrm{diag}\left(\tilde{\phi}_b^{l-2}\right)\tilde{\mathbf{w}}_b^l,$$
$$\mathbf{B}_b^l = \mathrm{diag}\left(\mathbf{K}_{bb} - \mathbf{K}_{bp}(\mathbf{K}_{pp})^{-1}\mathbf{K}_{pb} + \mathbf{K}_{bp}\left(\sum_b^l\right)^{-1}\mathbf{K}_{pb}\right). \quad (7)$$

Thus, by combining all L latent dimensions, the posterior of the GP latent embedding becomes $q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y}) = \prod_{l=1}^{L} q(\mathbf{z}^l|\mathbf{x},\mathbf{y}) = N(\mathbf{m},\mathbf{B})$.

Next, we can calculate the term $\mathbb{E}_{q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y})}[\log\tilde{q}(\tilde{\mathbf{z}}^{1:L}|\mathbf{y})]$. Note that $\mathbb{E}_{q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y})}[\log\tilde{q}(\tilde{\mathbf{z}}^{1:L}|\mathbf{y})] = -\mathrm{CE}(N(\mathbf{m},\mathbf{B})||N(\tilde{\mathbf{w}}^{1:L},\tilde{\Phi}^{1:L^2}))$, and the cross-entropy (CE) between the two Gaussian distributions is

$$\mathrm{CE}\left(N(\mathbf{m},\mathbf{B})||N\left(\tilde{\mathbf{w}}^{1:L},\tilde{\Phi}^{1:L^2}\right)\right)$$
$$= \frac{1}{2}\left(L\times\log(2\pi) + \log\left(\left|\mathrm{diag}(\tilde{\Phi}^{1:L^2})\right|\right) + (\mathbf{m}-\tilde{\mathbf{w}})^T\mathrm{diag}\left(\tilde{\Phi}^{1:L^{-2}}\right)\right. \quad (8)$$
$$\left.(\mathbf{m}-\tilde{\mathbf{w}}) + \mathrm{tr}\left(\mathrm{diag}(\mathbf{B})\mathrm{diag}\left(\tilde{\Phi}^{1:L^2}\right)\right)\right).$$

Now we have all of the components to calculate the KL loss of the GP embedding on a mini-batch b of data, which combines equations (4) and (18):

$$\mathrm{KL}(q(\mathbf{z}^{1:L}|\mathbf{x},\mathbf{y})||p(\mathbf{z}^{1:L})) = -\left[\mathrm{CE}\left(N(\mathbf{m},\mathbf{B})||N\left(\tilde{\mathbf{w}}^{1:L},\tilde{\Phi}^{1:L^2}\right)\right) + \frac{b}{N}\mathrm{L}_H\right]. \quad (9)$$

The detailed derivation of equations (3)–(9) is given in Supplementary Note 10.

Standard Gaussian prior

Following Kingma and Welling[16] and our notations, the posterior of standard Gaussian embedding follows $q(\mathbf{z}^{L+1:D}|\mathbf{y}) = q(\tilde{\mathbf{z}}^{L+1:D}|\mathbf{y}) = N(\tilde{\mathbf{w}}^{L+1:D},\tilde{\Phi}^{L+1:D^2})$, and we can easily write the KL loss of the standard Gaussian embedding $\mathrm{KL}(q(\mathbf{z}^{L+1:D}|\mathbf{x},\mathbf{y})||p(\mathbf{z}^{L+1:D}))$ as

$$\mathrm{KL}\left(q(\tilde{\mathbf{z}}^{L+1:D}|\mathbf{y})||N(0,\mathbf{I})\right) = -\frac{1}{2}\sum_{l=L+1}^{D}\left[\log\tilde{\phi}^{l^2} - \tilde{\phi}^{l^2} - \tilde{\mathbf{w}}^{l^2} + 1\right]. \quad (10)$$

Total ELBO equation

By combining equations (1)–(10) we can derive the total ELBO of our dependency-aware VAEs on a mini-batch b of data:

$$\mathrm{ELBO_b} = \sum_{i=1}^{b}\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i,\mathbf{y}_i)}(\log p(\mathbf{y}_i|\mathbf{z}))$$
$$-\beta\left[-\mathrm{CE}\left(N(\mathbf{m},\mathbf{B})||N(\tilde{\mathbf{w}}^{1:L},\tilde{\Phi}^{1:L^2})\right) - \frac{b}{N}\mathrm{L_H} + \mathrm{KL}\left(q\left(\tilde{\mathbf{z}}^{L+1:D}|\mathbf{y}_i\right)||N(0,\mathbf{I})\right)\right], \quad (11)$$

where $-\left[\mathrm{CE}(N(\mathbf{m},\mathbf{B})||N(\tilde{\mathbf{w}}^{1:L},\tilde{\Phi}^{1:L^2})) + \frac{b}{N}\mathrm{L_H}\right]$ represents the KL loss of GP embedding, $\mathrm{KL}(q(\tilde{\mathbf{z}}^{L+1:D}|\mathbf{y})||N(0,\mathbf{I}))$ is the KL loss of Gaussian embedding, and β controls the weight of the two KL losses. An overview of the probabilistic graphical model is shown in Supplementary Fig. 64. As a result, the posterior of the combined GP and Gaussian embeddings $\mathbf{z}$ is

$$q(\mathbf{z}|\mathbf{x},\mathbf{y}) = N\left(\left[\mathbf{m},\tilde{\mathbf{w}}^{L+1:D}\right],\left[\mathbf{B},\tilde{\Phi}^{L+1:D^2}\right]\right). \quad (12)$$

The computational time cost of typical GP regression is $O(N^3)$, where N is the sample size. By using inducing points, we can reduce the computational complexity to $O(bp^2+p^3)$, where b is the mini-batch size and p is the number of inducing points[19]. This improvement is significant in that reduces the computational time cost from cubic to linear as a function of the sample size.

To integrate datasets from different sources, we use the conditional autoencoder technique[48] to learn a batch-free representation. Specifically, if we denote the encoder as f and decoder as l, then $\mathbf{z} = f(\mathbf{y},\mathbf{x}_{\mathrm{batch}})$ and $\mathbf{y}' = l(\mathbf{z},\mathbf{x}_{\mathrm{batch}})$, where $\mathbf{x}_{\mathrm{batch}}$ is the one-hot encoded batch identification number (ID). The resulting $\mathbf{z}$ is expected to be disentangled from the batch IDs. For the GP part, we need a kernel to integrate batches. In this situation the auxiliary information $\mathbf{x}$ can be divided into two parts, one is the one-hot encoded batch ID $\mathbf{x}_{\mathrm{batch}}$ and the other is the spatial location $\mathbf{x}_{\mathrm{spatial}}$. The kernel function to integrate batches can be written as

$$k_\theta(\mathbf{x},\mathbf{x}) = \mathbf{x}_{\mathrm{batch}}\mathbf{x}_{\mathrm{batch}}^T k_\theta(\mathbf{x}_{\mathrm{spatial}},\mathbf{x}_{\mathrm{spatial}}).$$

This kernel function considers the independency between different batches, as well as the spatial dependency within one batch of data.

To make the latent embedding more robust, especially when training on small datasets, we use an optional argument in our models, which introduces the denoising technique to the encoder network, as in the previous study[49]. The noise is incorporated into the GP embedding only. For a mini-batch input $\mathbf{y}_b$, we estimate two sets of latent means and variances: $\{\tilde{\mathbf{w}}^{1:L} = f_w(f(\mathbf{y}))[:,1:L], \tilde{\Phi}^{1:L^2} = \exp(f_\Phi(f(\mathbf{y})))[:,1:L]\}$ and $\{\tilde{\mathbf{w}}'^{1:L} = f_w(f(\mathbf{y}+\gamma\cdot\delta))[:,1:L], \tilde{\Phi}'^{1:L^2} = \exp(f_\Phi(f(\mathbf{y}+\gamma\cdot\delta)))[:,1:L]\}$, where $\boldsymbol{\delta}\sim N(0,I)$ and γ controls the intensity of the random Gaussian noise. By equation (7), we can estimate two posterior means of the GP regression: $\mathbf{m}$ and $\mathbf{m}'$. Then the denoising loss function can be written by minimizing the mean square error between the two posterior means:

$$\mathrm{L_{noise}} = \sum_{i=1}^{b}(\mathbf{m}_i - \mathbf{m}_i')^2.$$

Gene likelihood

SpaVAE models the mRNA count data by a negative binomial (NB) model-based reconstruction loss. The NB model-based autoencoder has been successfully applied to scRNA-seq count data in previous studies[4–7]. The variational inference models the count matrix $\mathbf{y}^{\mathrm{gene}}$ likelihood by the NB distribution:

$$p(\mathbf{y}^{\mathrm{gene}}|\mathbf{z}) = \prod_i \mathrm{NB}(\mathbf{y}_i^{\mathrm{gene}}|\mu_i^{\mathrm{gene}},\theta^{\mathrm{gene}}).$$

Here, the NB likelihood of y_{ig}^{gene} is calculated as

$$\mathrm{NB}\left(y_{ig}^{\mathrm{gene}}|\mu_{ig}^{\mathrm{gene}},\theta_g^{\mathrm{gene}}\right)$$
$$= \frac{\Gamma(y_{ig}^{\mathrm{gene}}+\theta_g^{\mathrm{gene}})}{y_{ig}^{\mathrm{gene}}!\Gamma(\theta_g^{\mathrm{gene}})}\left(\frac{\theta_g^{\mathrm{gene}}}{\theta_g^{\mathrm{gene}}+\mu_{ig}^{\mathrm{gene}}}\right)^{\theta_g^{\mathrm{gene}}}\left(\frac{\mu_{ig}^{\mathrm{gene}}}{\theta_g^{\mathrm{gene}}+\mu_{ig}^{\mathrm{gene}}}\right)^{y_{ig}^{\mathrm{gene}}},$$

where i represents the spot index and g represents the gene index. The NB parameter mean $\boldsymbol{\mu}^{\mathrm{gene}}$ is parameterized by decoder networks with respect to the latent embedding $\mathbf{z}$. Specifically,

$$\mu_i^{\mathrm{gene}} = \mathrm{diag}(s_i)\times\exp\left(l_{\mu^{\mathrm{gene}}}(l(\mathbf{z}_i))\right), \quad (13)$$

where $l_{\mu^{\mathrm{gene}}}$ is a neural network that parametrizes the mean parameter, l is the latent decoder for genes, and s_i is the library size factor of spot i that is calculated in the preprocessing step. An exponential activation function is appended to the network, given that the mean is always positive. In the model, the estimated mean can be used as denoised

counts. The NB parameter dispersion θ_g^{gene} for gene g is a trainable parameter (we also apply an exponential activation function to ensure that it is always positive).

The reconstruction loss of spot i's gene counts is

$$\mathrm{L}_{\text{gene}} = -\log\left(NB\left(\mathbf{y}_i^{\text{gene}} | \mu_i^{\text{gene}}, \theta^{\text{gene}}\right)\right).$$

By combining this with equation (11), we can write the learning objective of spaVAE as minimizing the equation on a mini-batch b of data:

$$\mathrm{L}_{\text{spaVAE}} = \sum_{i=1}^{b} -\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i^{\text{gene}})} \log\left(NB\left(\mathbf{y}_i^{\text{gene}} | \mathbf{z}, \theta^{\text{gene}}\right)\right) + \beta\left[-\mathrm{CE}\left(N(\mathbf{m}, \mathbf{B}) || N\left(\tilde{\mathbf{w}}^{1:L}, \tilde{\phi}^{1:L^2}\right)\right) - \frac{b}{N}\mathrm{L}_{\mathrm{H}} + \mathrm{KL}\left(q\left(\tilde{\mathbf{z}}^{L+1:D} | \mathbf{y}_i\right) || N(0, \mathbf{I})\right)\right]. \quad (14)$$

ATAC peak likelihood

The spatial ATAC-seq data can be treated as a binary matrix of spots by peak regions. SpaPeakVAE models the binary observations $\mathbf{y}^{\text{peak}}$ by a Bernoulli distribution $y_{iq}^{\text{peak}} \sim \text{Bernoulli}(\lambda_{iq}^{\text{peak}} \times s_i \times r_q)$, where i is the spot index and q is the peak index. Following a similar setting in SCALE[24] and PeakVI[25], the probability of spot i in the Bernoulli distribution consists of three components. The first is parametrized by the decoder

$$\lambda_i^{\text{peak}} = \text{sigmoid}\left(l_{\lambda^{\text{peak}}}\left(l(\mathbf{z}_i)\right)\right), \quad (15)$$

where $\mathbf{z}$ is the latent embedding and $l_{\lambda^{\text{peak}}}$ and l are decoder networks. Here, estimated $\boldsymbol{\lambda}^{\text{peak}}$ can be used as the denoised peak value. s_i captures spot-specific biases and r_q captures peak-specific biases (also estimated by neural networks). Here we use the sigmoid function to ensure that the three probabilities are within the interval [0, 1].

The reconstruction loss of spot i's peak observations is

$$\mathrm{L}_{\text{peak}} = \mathrm{BCE}\left(\lambda_i^{\text{peak}} \odot \mathbf{r} \times s_i, \mathbf{y}_i^{\text{peak}}\right),$$

where BCE is the binary cross-entropy, and $\odot$ denotes the element-wise multiplication of two vectors.

Thus, we can write the learning objective of spaPeakVAE as minimizing the equation on a mini-batch b of data:

$$\mathrm{L}_{\text{spaPeakVAE}} = \sum_{i=1}^{b} \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i^{\text{peak}})} \mathrm{BCE}\left(\lambda_i^{\text{peak}} \odot \mathbf{r} \times s_i, \mathbf{y}_i^{\text{peak}}\right) + \beta\left[-\mathrm{CE}\left(N(\mathbf{m}, \mathbf{B}) || N\left(\tilde{\mathbf{w}}^{1:L}, \tilde{\Phi}^{1:L^2}\right)\right) - \frac{b}{N}\mathrm{L}_{\mathrm{H}} + \mathrm{KL}\left(q\left(\tilde{\mathbf{z}}^{L+1:D} | \mathbf{y}_i\right) || N(0, \mathbf{I})\right)\right]. \quad (16)$$

Protein likelihood

SpaMultiVAE captures observed protein counts by a mixture of two negative binomial distributions to account for background and foreground intensities, in the same way as the technique used in totalVI[26]. The protein count matrix $\mathbf{y}^{\text{protein}}$ is characterized by four parameters, $\mathrm{v}_i^{\text{back}}$, $\mathrm{v}_i^{\text{fore}}$, Φ^{protein} and π_i^{protein}:

$$\begin{aligned} p\left(\mathbf{y}^{\text{protein}} | \mathbf{z}\right) &= \prod_i \text{MixtureNB}\left(\mathbf{y}_i^{\text{protein}} | \mathrm{v}_i^{\text{back}}, \mathrm{v}_i^{\text{fore}}, \Phi^{\text{protein}}, \pi_i^{\text{protein}}\right) \\ &= \prod_i \pi_i \times \mathrm{NB}\left(\mathbf{y}_i^{\text{protein}} | \mathrm{v}_i^{\text{back}}, \Phi^{\text{protein}}\right) \\ &\quad + (1 - \pi_i) \times \mathrm{NB}\left(\mathbf{y}_i^{\text{protein}} | \mathrm{v}_i^{\text{fore}}, \Phi^{\text{protein}}\right), \end{aligned}$$

where i represents the spot index; $\mathrm{v}_i^{\text{back}}$ is the background component of protein counts, $\mathrm{v}_i^{\text{fore}}$ is the foreground component of protein counts, $\boldsymbol{\Phi}^{\text{protein}}$ is the dispersion (Φ_p^{protein} is a trainable parameter for protein p), and π_i^{protein} is the background probability.

Here, we assume that $\mathrm{v}_{ip}^{\text{back}}$ of spot i and protein p follows a prior log-normal distribution (the protein intensity is always positive):

$$p\left(\mathrm{v}_{ip}^{\text{back}}\right) = \text{lognormal}\left(m_p^{\text{prior}}, \sigma_p^{\text{prior}}\right),$$

where $\mathbf{m}^{\text{prior}}$ and $\boldsymbol{\sigma}^{\text{prior}}$ are the trainable prior parameters of the background intensity. Before the training process, we first fit a two-component Gaussian mixture model (GMM) to the log-transformed counts of each protein p, and use the smaller component of the GMM as the initial values of m_p^{prior} and σ_p^{prior}.

The posterior distribution of the variable $\mathrm{v}_i^{\text{back}}$ is inferred by decoder networks with the input of latent embedding $\mathbf{z}$. Specifically,

$$\begin{aligned} \mathbf{m}_i^{\text{back}} &= l_{m^{\text{back}}}(l'(\mathbf{z}_i)), \\ \sigma_i^{\text{back}} &= \exp(l_{\sigma^{\text{back}}}(l'(\mathbf{z}_i))), \\ q\left(\mathrm{v}_i^{\text{back}} | \mathbf{z}_i\right) &= \text{lognormal}\left(\mathbf{m}_i^{\text{back}}, \sigma_i^{\text{back}}\right). \end{aligned} \quad (17)$$

The parameter $\boldsymbol{\pi}_i^{\text{protein}}$ controls the probability of the background intensity and is also inferred by the neural network

$$\boldsymbol{\pi}_i^{\text{protein}} = \text{sigmoid}\left(l_{\pi^{\text{protein}}}\left(l'\left(\mathbf{z}_i\right)\right)\right), \quad (18)$$

where we use the sigmoid activation function to make $\boldsymbol{\pi}_i^{\text{protein}}$ fall in the range 0–1.

We multiply by a value that is greater than 1 to ensure that the foreground is always larger than the background:

$$\mathbf{v}_i^{\text{fore}} = \left(1 + \boldsymbol{\alpha}_i^{\text{protein}}\right) \times \mathbf{v}_i^{\text{back}}, \quad (19)$$

and

$$\boldsymbol{\alpha}_i^{\text{protein}} = \text{softplus}\left(l_{\alpha^{\text{protein}}}\left(l'\left(\mathbf{z}_i\right)\right)\right), \quad (20)$$

which is also inferred by the neural network.

The estimated foreground variable can be used for the denoised protein intensity. In equations (17)–(20), $l_{m^{\text{back}}}$, $l_{\sigma^{\text{back}}}$, $l_{\pi^{\text{protein}}}$ and $l_{\alpha^{\text{protein}}}$ are neural networks that parametrize different variables with suitable activation functions, and l' is the latent protein decoder.

Hence, the reconstruction loss of spot i's protein part can be written as

$$\begin{aligned} \mathrm{L}_{\text{protein}} = &-\log\left(\text{MixtureNB}\left(\mathbf{y}_i^{\text{protein}} | \mathbf{v}_i^{\text{back}}, \mathbf{v}_i^{\text{fore}}, \boldsymbol{\Phi}^{\text{protein}}, \boldsymbol{\pi}_i^{\text{protein}}\right)\right) \\ &+\mathrm{KL}\left(\text{lognormal}\left(\mathbf{m}^{\text{back}}, \boldsymbol{\sigma}^{\text{back}}\right) || \text{lognormal}\left(\mathbf{m}^{\text{prior}}, \boldsymbol{\sigma}^{\text{prior}}\right)\right), \end{aligned}$$

where we apply variational inference to approximate the marginal likelihood of protein counts.

As in the spaVAE model, mRNA counts are characterized by an NB model-based decoder in spaMultiVAE. Thus, we can write the learning objective of spaMultiVAE as minimizing the equation on a mini-batch b of data:

$$\begin{aligned} \mathrm{L}_{\text{spaMultiVAE}} = &\sum_{i=1}^{b} -\mathbb{E}_{q\left(\mathbf{z}|\mathbf{x}_i, \mathbf{y}_i^{\text{gene}}, \mathbf{y}_i^{\text{protein}}\right)} \left[\log\left(\mathrm{NB}\left(\mathbf{y}_i^{\text{gene}} | \mathbf{z}, \theta^{\text{gene}}\right)\right)\right. \\ &\left. + \log\left(\text{MixtureNB}\left(\mathbf{y}_i^{\text{protein}} | \mathbf{z}, \Phi^{\text{protein}}\right)\right)\right] \\ &+\mathrm{KL}\left(\text{lognormal}\left(\mathbf{m}^{\text{back}}, \sigma^{\text{back}}\right) || \text{lognormal}\left(\mathbf{m}^{\text{prior}}, \sigma^{\text{prior}}\right)\right) \\ &+\beta\left[-\mathrm{CE}\left(N(\mathbf{m}, \mathbf{B}) || N\left(\tilde{\mathbf{w}}^{1:L}, \tilde{\Phi}^{1:L^2}\right)\right)\right. \\ &\left. - \frac{b}{N}\mathrm{L}_{\mathrm{H}} + \mathrm{KL}\left(q(\tilde{\mathbf{z}}^{L+1:D} | \mathbf{y}_i) || N(0, \mathbf{I})\right)\right]. \end{aligned} \quad (21)$$

spaLDVAE and spaPeakLDVAE models

The spaLDVAE model is based on spaVAE. For interpretability purposes, we use a linear decoder in spaLDVAE. Thus, for the SRT data, the mean parameter in the NB likelihood is estimated by

$$\mu_i^{\text{gene}} = \text{diag}(s_i) \times (\exp(\mathbf{z}_i) \times \exp(\mathbf{W}_D)), \tag{22}$$

where $\mathbf{z}$ is the latent embedding and $\mathbf{W}_D$ is the trainable decoder weight. We use an exponential function to ensure that these two parameters are always positive.

For the spatial ATAC-seq data, spaPeakLDVAE estimates the probability parameter in the Bernoulli likelihood by

$$\lambda_i^{\text{peak}} = \frac{\exp(\mathbf{z}_i) \times \exp(\mathbf{W}_D)}{1 + \exp(\mathbf{z}_i) \times \exp(\mathbf{W}_D)}, \tag{23}$$

where we scale $\boldsymbol{\lambda}^{\text{peak}}$ to the interval $[0, 1]$.

spaLDVAE and spaPeakVAE are interpretable nonnegative factor models that combine the strength of a flexible deep encoder model with a linear reconstruction function[50,51].

To separate the spatial and nonspatial genes and peaks, we divide the latent embedding $\mathbf{z}$ into two parts. If the total dimension number is $2T$, then the first T dimensions follow a GP prior

$$\mathbf{z}[:, 1 : T] \sim \text{GP}(0, \mathbf{K}_{NN}),$$

and the last T dimensions follow a Gaussian prior

$$\mathbf{z}[:, (T+1) : 2T] \sim N(\mathbf{v}, \boldsymbol{\rho}^2),$$

where $\mathbf{v}$ and $\boldsymbol{\rho}^2$ are trainable mean and variance parameters of the Gaussian prior distribution. The two components account for spatially dependent and spatially independent genes or peaks, respectively.

After training the model, we can quantify the contribution of the two components of $\mathbf{z}$ by the weight of the linear decoder. The weight of the decoder is normalized by the average of latent embeddings

$$\tilde{\mathbf{f}} = \sum_{i=1}^{N} \exp(\mathbf{z}[i, :]),$$

$$\mathbf{W}_D' = \text{diag}\left(\tilde{\mathbf{f}}\right)^{-1} \times \exp(\mathbf{W}_D).$$

Then we define the spatial score of gene or peak j as

$$\delta_j = \sum_{k=1}^{T} \mathbf{W}_D'[k, j] / \sum_{k=1}^{2T} \mathbf{W}_D'[k, j], \tag{24}$$

which represents how much variance of feature j can be explained by the GP latent factors. The value of δ_j ranges from 0 to 1, and can be used to prioritize spatial genes or peaks. Larger δ_j means that the gene or peak j is more spatially dependent.

To establish a significance cut-off for spatially dependent features, we propose a random permutation strategy. In this approach we randomly shuffle the spatial locations of spots once. In the resulting dataset, all features can be treated as spatially independent. Subsequently, we retrain the model on this permuted data. The maximum δ score obtained from this shuffled data can serve as an estimate for the significance cut-off. Note that the cut-off found by permutation can be used for analytical purposes. A comprehensive estimation of the null distribution of δ requires multiple permutation experiments that may not be feasible in a timely manner in most cases.

Implementation

Models are implemented in Pytorch[52]. All hidden layers are fully connected layers that use the ELU (exponential linear unit) activation function[53] and the batch normalization technique[54]. The default layer sizes of spaVAE are set to (128, 64) for the encoder and (128) for the decoder. For spatial ATAC-seq data, spaPeakVAE uses (1,024, 128) for the encoder and (128, 1,024) for the decoder. spaLDVAE and spaPeakLDVAE use the same corresponding encoder structures as in spaVAE and spaPeakVAE, respectively, and have an appended linear decoder. SpaMultiVAE uses the same architecture for the encoder part as in spaVAE, and the hidden layers for gene and protein decoders are set to (128) and (128), respectively. The bottleneck layer sizes for spaVAE, spaPeakVAE and spaMultiVAE are set to 10, 13 and 20, respectively (dimension numbers of GP embeddings are 2 for spaVAE and spaMultiVAE, and 5 for spaPeakVAE; others are set to be Gaussian embeddings). The bottleneck layer size for spaLDVAE and spaPeakLDVAE is 30 (15 GP embeddings and 15 Gaussian embeddings). The AdamW optimizer[55,56] is used for training models with the settings $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and weight decay $= 1 \times 10^{-6}$. In spaVAE, spaLDVAE and spaPeakLDVAE we set the learning rate at $lr = 0.001$; in spaPeakVAE, $lr = 0.0001$; and in spaMultiVAE, $lr = 0.005$. We set the default mini-batch size based on the sample size of the data for spaVAE, spaPeakVAE and spaMultiVAE: if there are fewer than 1,024 spots, then it is set at 128; if there are 1,024–2,048 spots, then it is set at 256; if there are more than 2,048 spots, we set it at 512. For spaLDVAE and spaPeakLDVAE, we set mini-batch size to 256. We use the dynamic VAE technique[57] (Supplementary Note 11) to automatically tune the weight of the KL loss β in a reasonable range for deep decoder models (spaVAE, spaPeakVAE and spaMultiVAE); for spaLDVAE and spaPeakLDVAE, we fix β at 10. The default setting of random Gaussian noise is $\gamma = 0$. The maximum number of training epochs is set at 5,000 for deep decoder models, and early stopping with a patience of 200 iterations based on the ELBO value of the validating set (95% training set and 5% validating set) is used to determine the optimal number of training iterations. For spaLDVAE and spaPeakLDVAE we set the number of training iterations to 2,000. To make it more numerically stable during training, the Cauchy kernel is used in spaVAE, spaPeakVAE and spaMultiVAE. Meanwhile, the Matern kernel is used in spaLDVAE and spaPeakLDVAE. One critical parameter in our models is the number of inducing points, which controls the ranking of the covariance matrix of the latent embedding. The choice of the number of inducing points is based on datasets. If the dataset has complex spatial patterns, we need more inducing points; otherwise, we can use a small number. We provide two methods to generate inducing points. The first method uses a two-dimensional grid: if the grid step is set to n, then the total number of inducing points is $(n+1)^2$, and we use this method for datasets including human DLPFC, mouse olfactory bulb, mouse anterior and posterior brains, HER2 breast tumor, mouse E15.5 brain MISAR-seq, spatial-CITE-seq and spatial DBiT-seq (these datasets have a square tissue shape). The other method involves k-means clustering on positions, with centroids as the inducing points, and this is applied to mouse hippocampus Slide-seq V2 data (the tissue shape is circular). The setting of hyperparameters is kept consistent for datasets of the same type (for example, SRT, spatial ATAC-seq, spatial-CITE-seq), and the choice of hyperparameters for each dataset used in this study is summarized in Supplementary Table 2.

Downstream tasks

Clustering. After training the model, we output the mean parameter $[\mathbf{m}, \tilde{\mathbf{w}}^{L+1:D}]$ of equation (12) as the latent representation. The k-means clustering and Louvain clustering[58] are applied to the latent embedding to obtain the cluster labels. For the Louvain clustering, we search the resolution parameter from small to large, until the predefined number of clusters is obtained. Following clustering, we provide an optional refining procedure, which refines the predicted label of each spot based on the major labels of its spatial neighbors (4 or 6 nearest neighbors, which is based on sequencing platforms). Similar procedures have been applied in previous studies[8,13].

Denoising. In the spaVAE model we use the output of $\tilde{\boldsymbol{\mu}} = \exp(l_{\mu^{\text{gene}}}(l(\mathbf{z})))$ on the right-hand side of equation (13) as the denoised counts of genes, which eliminates the effect of library size. Given that spaVAE is a deep generative model, and the latent embedding $\mathbf{z}$ is inferred by equation (12), we can generate multiple sets of $\mathbf{z}$ and $\tilde{\boldsymbol{\mu}}$, then use the average of the sets of $\tilde{\boldsymbol{\mu}}$ as a more robust result.

In the spaPeakVAE model, we use the variable $\boldsymbol{\lambda}$ in equation (15) as the denoised observation of ATAC peaks, which eliminates the bias effect of both spot and peak.

For the protein part of the spatial multi-omics data, we use the variable $\mathbf{v}^{\text{fore}}$ from equation (19) as the denoised protein intensity. We first sample multiple sets of binary values based on background probabilities from equation (18):

$$\mathbf{r} \sim \text{Bernoulli}(\boldsymbol{\pi}),$$

then we take the average of multiple inferred foreground $\mathbf{v}^{\text{fore}} = (1-\mathbf{r})(1+\boldsymbol{\alpha})\mathbf{v}^{\text{back}}$ based on equation (19).

Spatial interpolation and resolution enhancement. In the analysis of spatial interpolation and enhancement of spatial resolution, the model needs to infer spots in unobserved locations. For a new set of spatial locations $\mathbf{x}_{\text{test}}$, we can infer the l-th of the L dimension GP embedding of the testing locations by equation (6)

$$q\left(\mathbf{z}^l_{\text{test}}\right) = N\left(\mathbf{z}^l_{\text{test}} | \mathbf{K}_{tp}\mathbf{K}_{pp}^{-1}\boldsymbol{\mu}^l_{1:N}, \mathbf{K}_{tt} - \mathbf{K}_{tp}\mathbf{K}_{pp}^{-1}\mathbf{K}_{pt} + \mathbf{K}_{tp}\mathbf{K}_{pp}^{-1}\mathbf{A}^l_{1:N}\mathbf{K}_{pp}^{-1}\mathbf{K}_{pt}\right),$$

where $\mathbf{K}_{tp} = k_\theta(\mathbf{x}_{\text{test}}, \mathbf{p}) = \mathbf{K}_{pt}^T$, and $\mu^l_{1:N}$ and $\mathbf{A}^l_{1:N}$ are the estimates of the whole training set using equation (5). By combining the first L dimensions, we can obtain the latent GP embedding $\mathbf{z}^{1:L}_{\text{test}}$ of the test spots. For the remaining $D-L$ dimensions of latent embedding at each test location, we simply select the Gaussian embedding of the nearest training spot to serve as its latent Gaussian embedding, denoted as $\mathbf{z}^{L+1:D}_{\text{test}}$. After obtaining $\mathbf{z}_{\text{test}}$ we can also generate denoised counts of genes, peaks or proteins for the test spots. We can use any reasonable new test location for spatial interpolation and resolution enhancement, which is a favorable feature of our spatial-aware variational autoencoder models.

Differential expression and differential accessibility analysis. Following ref. 5, we use Bayes factor to detect differentially expressed genes and proteins, as well as differentially accessible ATAC peaks in the spatial genomics data. If we have two group labels A and B in the dataset, we generate a predefined number (default = 10,000) of paired spots (a, b) from the two groups, and calculate denoised counts (μ_a, μ_b). When comparing the two groups of denoised counts, the pairwise log fold change (LFC) can be written as $\text{LFC}_{a,b} = \log_2 \frac{\mu_a+\epsilon}{\mu_b+\epsilon}$, where ϵ is a small offset added to make the result more robust. Then we can formulate two mutually exclusive hypotheses:

$$\mathcal{H}_0 : |\text{LFC}_{a,b}| < \delta \text{ and } \mathcal{H}_1 : |\text{LFC}_{a,b}| \geq \delta,$$

where δ is a threshold for the effect size. Intuitively, we are measuring the probability that the absolute LFC is equal to or larger than δ. Following a previous study[59], we assume that most of the pairwise LFCs will concentrate around 0, meaning that they are equally expressed. To determine the value of δ, we fit pairwise LFC values to a three-component GMM. Then δ is defined as the mean of the largest modes of GMM (in absolute value), the corresponding distribution of which should predominantly encompass differentially expressed or differentially accessible features. Having established δ, we compare the probability of the two hypotheses using a Bayes factor (BF),

$$\text{BF} = \frac{p(|\text{LFC}_{a,b}| \geq \delta)}{p(|\text{LFC}_{a,b}| < \delta)}.$$

Based on ref. 60, the significance threshold for the Bayes factor is defined as follows: a score of 3.2 is regarded as substantially significant, while a score of 10 is considered strongly significant.

In a manner similar to standard differential expression analyses, our Bayes factor can serve as an analogous measure to a P value. Concurrently, we also require the group-level fold change to identify differentially expressed or differentially accessible features. The LFC between groups A and B can be computed as

$$\text{LFC}_{A,B} = \log_2 \frac{\mu_A + \epsilon}{\mu_B + \epsilon},$$

where μ_A and μ_B are the average of denoised counts in A and B, respectively. In conclusion, both the Bayes factor and the fold change can be used collectively to identify differential expression or accessibility.

Data preprocessing

Preprocessing of spatial transcriptomics data. Following Shang and Zhou[13], we filter genes to reserve a set of SVGs. Specifically, we apply SPARK[36] for SVG analysis in small datasets due to its higher statistical power and use SPARK-X[61] for SVG analysis in larger datasets to save time and memory. We select up to 3,000 significant SVGs for each single dataset, with false discovery rate ≤ 0.05, as input to our model. For batch integration experiments we use all of the SVGs detected from different datasets, if the number of distinct SVGs is not too large (Extended Data Fig. 1 and Supplementary Figs. 28 and 29); otherwise, we select only the genes that proved to be statistically significant in multiple samples (Supplementary Fig. 19).

After filtering genes using the above criteria, we follow previous studies[6,7] and use the Python package SCANPY[62] to preprocess the raw spatial transcriptomics read count data. First, we filter out genes and spots with zero counts, and then compute the size factor for each spot. Let us denote the library size (that is, the total number of read counts) of spot i as l_i. The size factor of spot i is calculated as $s_i = \frac{l_i}{\text{median}(\{l_j\}_{j=1,\cdots,N})}$. Second, the raw read counts are normalized by total counts over genes, so that every spot has the same total count after normalization. Finally, we take the log transformation and scale the data to have unit variance and zero mean. The transformed and scaled expression matrix is used as the input for our spaVAE model, and we use the original count matrix when calculating the NB loss of genes.

Preprocessing of spatial ATAC-seq data. As in SCALE[24], we first trim the count matrix to binary (set values > 0 to 1). Next, we filter the spatial ATAC-seq count matrix by two criteria: we keep only peaks in >1% of spots and the top 30,000 variable peaks.

Preprocessing of spatial multi-omics data. The protein count matrix is log-transformed and scaled. To integrate the gene and protein expressions for each spot, we concatenate the preprocessed gene and protein counts, and use this combined matrix as input for the spaMultiVAE model. When calculating the gene and protein reconstruction losses, we use the raw counts to compute the NB loss for genes and the NB mixture loss for proteins.

Competing methods

GraphST[9] (https://github.com/JinmiaoChenLab/GraphST), SpatialPCA[13] (https://github.com/shangll123/SpatialPCA), STAGATE[10] (https://github.com/zhanglabtools/STAGATE), BayesSpace[14] (https://github.com/edward130603/BayesSpace), SpaGCN[8] (https://github.com/jianhuupenn/SpaGCN), stLearn[12] (https://stlearn.readthedocs.io), Giotto[28] (https://rubd.github.io/Giotto_site), cisTopic[40] (https://github.com/aertslab/cisTopic), chromVAR[41] (https://github.com/GreenleafLab/chromVAR), LSA[42] (implemented by scikit-learn[63]), SCALE[24] (https://github.com/jsxlei/SCALE), scVI[5] (https://scvi-tools.org), PeakVI[25]

(https://scvi-tools.org) and totalVI[26] (https://scvi-tools.org) are used as the competing methods for different tasks. The raw data are preprocessed based on the tutorials or steps described in previous works for each competing method. Software and packages implemented by original authors are used to conduct the experiments. A detailed description of the competing methods is given in Supplementary Note 12.

Spatial genomics datasets

The human DLPFC datasets[27] were downloaded from http://spatial.libd.org/spatialLIBD/. The 12 sections contain 3,000–4,000 spots that span six neural layers and the white matter. The layers were annotated manually by the authors. After SVG gene filtering, the datasets contain 1,000–3,000 genes.

Mouse hippocampus Slide-seq V2 data[31] were obtained from the Broad Institute Single Cell portal (https://singlecell.broadinstitute.org/single_cell/study/SCP815/sensitive-spatial-genome-wideexpression-profiling-at-cellular-resolution#study-summary). We used the file 'Puck_200115_08'. The dataset contains approximately 23,000 genes in 53,000 spatial locations. We first filtered out some outlier spots. The filtered matrix was used for SVG selection. The resulting matrix is 3,000 genes by 51,367 spots.

Mouse anterior and posterior brains data were downloaded from the 10X genomics website: https://www.10xgenomics.com/resources/datasets/mouse-brain-serial-section-1-sagittal-anterior-1-standard-1-0-0, https://www.10xgenomics.com/resources/datasets/mouse-brain-serial-section-1-sagittal-posterior-1-standard-1-0-0 and https://www.10xgenomics.com/resources/datasets/mouse-brain-serial-section-2-sagittal-anterior-1-standard-1-0-0, https://www.10xgenomics.com/resources/datasets/mouse-brain-serial-section-2-sagittal-posterior-1-standard-1-0-0. After SVG filtering, the dataset contains 3,313 genes and 12,164 spots across the four batches (two anterior and two posterior).

The mouse olfactory bulb dataset[37] was downloaded from https://www.spatialresearch.org/resources-published-datasets/doi-10-1126science-aaf2403/. The dataset contains 12 samples. Each sample contains approximately 200 spots. We selected a union set of top HVGs across the 12 samples for the analysis, which leads to data matrices of 2,949 genes and around 200 spots in each sample.

HER2 breast tumor data[38] were downloaded from https://github.com/almaan/her2st. The dataset contains eight samples with pathologist-annotated labels, and we used the H1 sample to demonstrate the result in the Fig. 4, with the others given in Supplementary Figs. 37–43. The sample sizes are small, containing 100–600 spots. After filtering, the number of genes is around 100–1,000.

MISAR-seq data[21] and the spot coordinates were downloaded from https://doi.org/10.5281/zenodo.7480069, which profiled mouse embryonic E15.5 brain. Manually annotated labels were obtained based on the Allen Brain Atlas[32] and H&E images (Supplementary Fig. 44). We use the mRNA counts, generated directly from Cellranger, and use ArchR (v1.0.2)[64] to build the scATAC-seq counts from a fragment file. We completely follow the pipeline provided by the author of MISAR-seq to preprocess and filter the data. After preprocessing, 1,949 out of 2,500 spots are retained. ArchR calls macs2 (v2.2.7.1)[65] to detect the peak regions and then compute the counts for each peak per cell. The raw peak matrix contains 105,350 peaks over 1,949 spots. We then use the 'FindTopFeatures' function in Signac (v1.4.0)[66] to select the peaks that are detected in at least 200 cells for the analysis. A total of 47,287 out of 105,350 peaks pass the filtering. mRNA counts are filtered by Spark with the default settings. A total of 2,144 out of 26,272 genes are selected as the SVGs for the analysis. We downloaded mouse E15.5 forebrain, midbrain and hindbrain bulk ATAC-seq data from the ENCODE project[67]. Details of the processing steps for the bulk ATAC-seq data are given in Supplementary Note 13. The motif analysis steps are described in Supplementary Note 14.

Spatial-CITE-seq data[22] were downloaded from GEO https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE213264. We used a human tonsil sample for the analysis, which contains 2,492 spots with 28,417 genes and 283 proteins. After SVG filtering, there are 984 genes left.

Spatial DBiT-seq data[23] were obtained from the GEO database https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE137986. We used the '0713aL' and '0713cL' samples, which are the sequencing data for the mouse embryonic brain region. The dataset contains 1,789 spots with approximately 18,000 genes. After SVG filtering, there are 254 genes selected. We used all 22 proteins.

For the experiments on HVGs (for example, human DLPFC datasets, spatial-CITE-seq and spatial DBiT-seq data), we selected the top 2,000 HVGs to generate datasets with HVGs.

Reporting summary

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

Data availability

All data supporting the findings of the study are deposited and available at https://doi.org/10.6084/m9.figshare.21623148.v5 (ref. 68).

Code availability

An open-source software implementation of spaVAE, spaPeakVAE, spaMultiVAE, spaLDVAE and spaPeakLDVAE is available on GitHub: https://github.com/ttgump/spaVAE. It is also available in the Zenodo repository[69].

References

46. Higgins, I. et al. beta-VAE: learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations* (2017).
47. Pearce, M. The Gaussian process prior VAE for interpretable latent dynamics from pixels. *Proceedings of Machine Learning Research* **118**, 1–12 (2020).
48. Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. In *Proc. 28th International Conference on Neural Information Processing Systems (NIPS 2015)* (eds Cortes, C. et al.) 3483–3491 (MIT Press, 2015).
49. Ding, J. & Regev, A. Deep generative model embedding of single-cell RNA-seq profiles on hyperspheres and hyperbolic spaces. *Nat. Commun.* **12**, 2554 (2021).
50. Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics* **36**, 3418–3421 (2020).
51. Townes, F. W. & Engelhardt, B. E. Nonnegative spatial factorization applied to spatial genomics. *Nat. Methods* **20**, 229–238 (2023).
52. Paszke, A. et al. Automatic differentiation in PyTorch. In *Proc. 31st International Conference on Neural Information Processing Systems (NIPS 2017)* (eds Wallach, H. M. et al.) (Curran Associates, Inc., 2017).
53. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). In *International Conference on Learning Representations* (2015).
54. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proc. 32nd International Conference on International Conference on Machine Learning (ICML 2015)* (eds Bach, F. & Blei, D.), Vol. 37, 448–456 (JMLR.org, 2015).
55. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference for Learning Representations* (2015).
56. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (2017).
57. Shao, H. et al. Rethinking controllable variational autoencoders. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 19228–19237 (IEEE, 2022).

58. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008).
59. Boyeau, P. et al. An empirical Bayes method for differential expression analysis of single cells with deep generative models. *Proc. Natl Acad. Sci. USA* **120**, e2209124120 (2023).
60. Kass, R. E. & Raftery, A. E. Bayes factors. *J. Am. Stat. Assoc.* **90**, 773–795 (1995).
61. Zhu, J., Sun, S. & Zhou, X. SPARK-X: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies. *Genome Biol.* **22**, 184 (2021).
62. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
63. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
64. Granja, J. M. et al. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nat. Genet.* **53**, 403–411 (2021).
65. Zhang, Y. et al. Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* **9**, R137 (2008).
66. Stuart, T., Srivastava, A., Madad, S., Lareau, C. A. & Satija, R. Single-cell chromatin state analysis with Signac. *Nat. Methods* **18**, 1333–1341 (2021).
67. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**, 57–74 (2012).
68. Tian, T. Spatial genomics datasets. *figshare* https://doi.org/10.6084/m9.figshare.21623148.v5 (2023).
69. Tian, T. spaVAE: spatial dependency-aware deep generative models. *Zenodo* https://doi.org/10.5281/zenodo.8407637 (2023).

Acknowledgements

The study was supported by grant R15HG012087 (Z.W.) from the National Institutes of Health (NIH), grant BK20230781 (J.Z.) from the Natural Science Foundation of Jiangsu Province, and also funded in part by an Institutional Development Fund from The Children's Hospital of Philadelphia (CHOP) and by CHOP's Endowed Chair in Genomic Research. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) through the allocation CIE170034, supported by the National Science Foundation grant number ACI1548562. The authors thank R. Cheng from the Tianjin University of Finance and Economics for assistance with the manuscript.

Author contributions

T.T. conceived the project. T.T. and J.Z. designed the method. T.T., J.Z. and X.L. designed and conducted the experiments. Z.W. and H.H. supervised the study. T.T., J.Z., X.L., Z.W. and H.H. wrote the manuscript. All authors approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

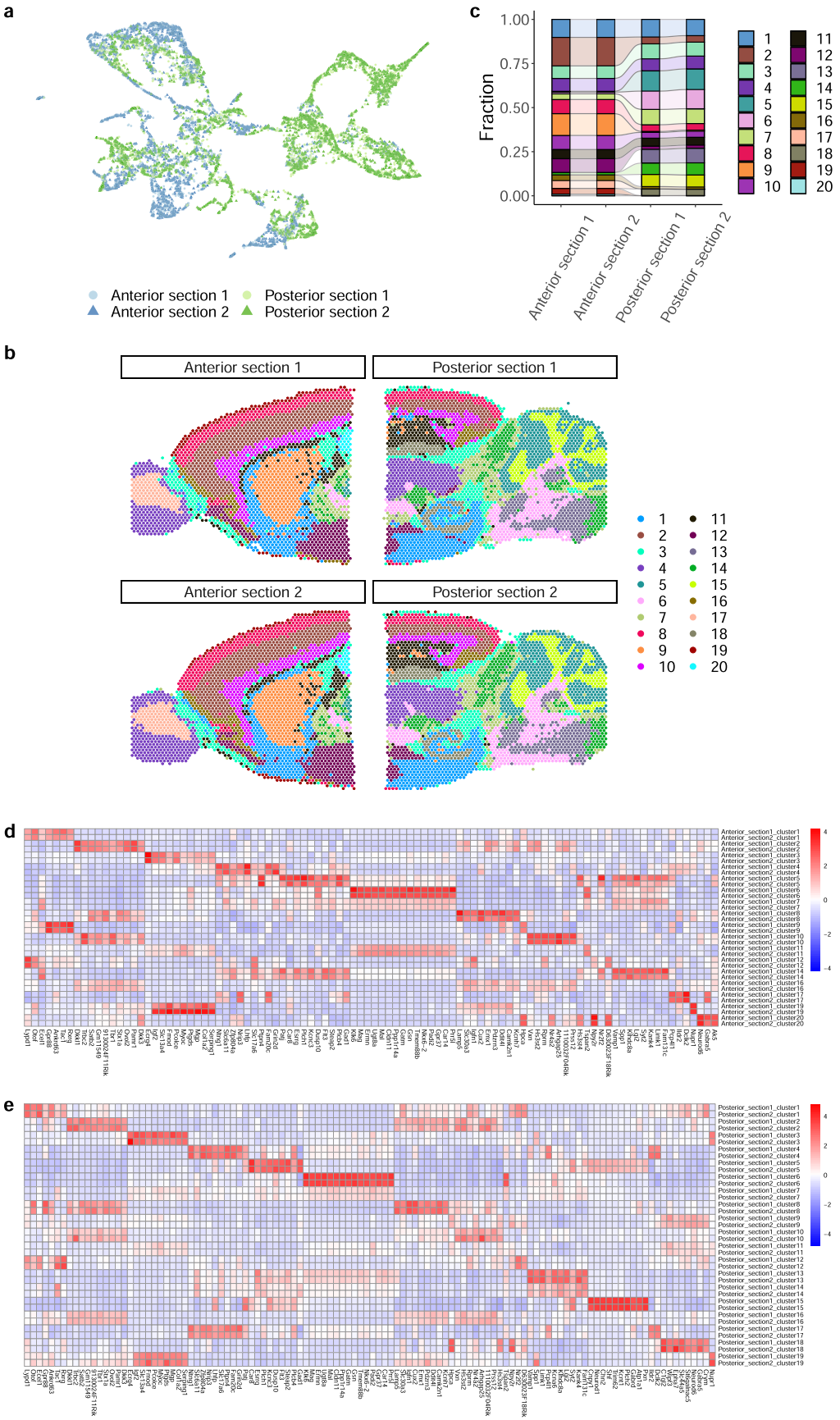
Extended data are available for this paper at https://doi.org/10.1038/s41592-024-02257-y.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-024-02257-y.

Correspondence and requests for materials should be addressed to Zhi Wei.

Peer review information *Nature Methods* thanks Ofir Lindenbaum and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available. Primary Handling Editors: Hui Hua and Lin Tang, in collaboration with the *Nature Methods* team.

https://doi.org/10.1038/s41592-024-02257-y

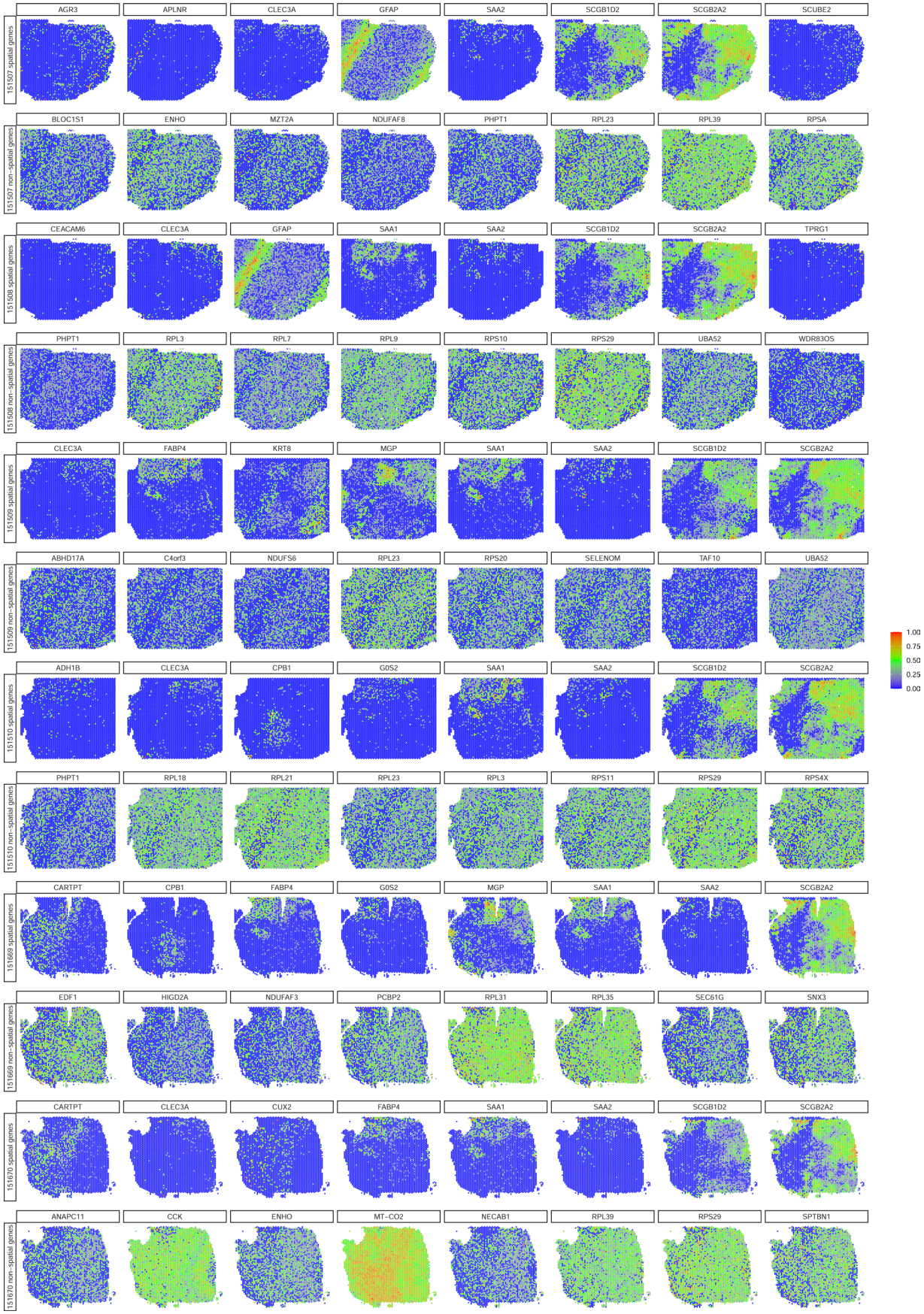


Extended Data Fig. 1 | See next page for caption.

https://doi.org/10.1038/s41592-024-02257-y

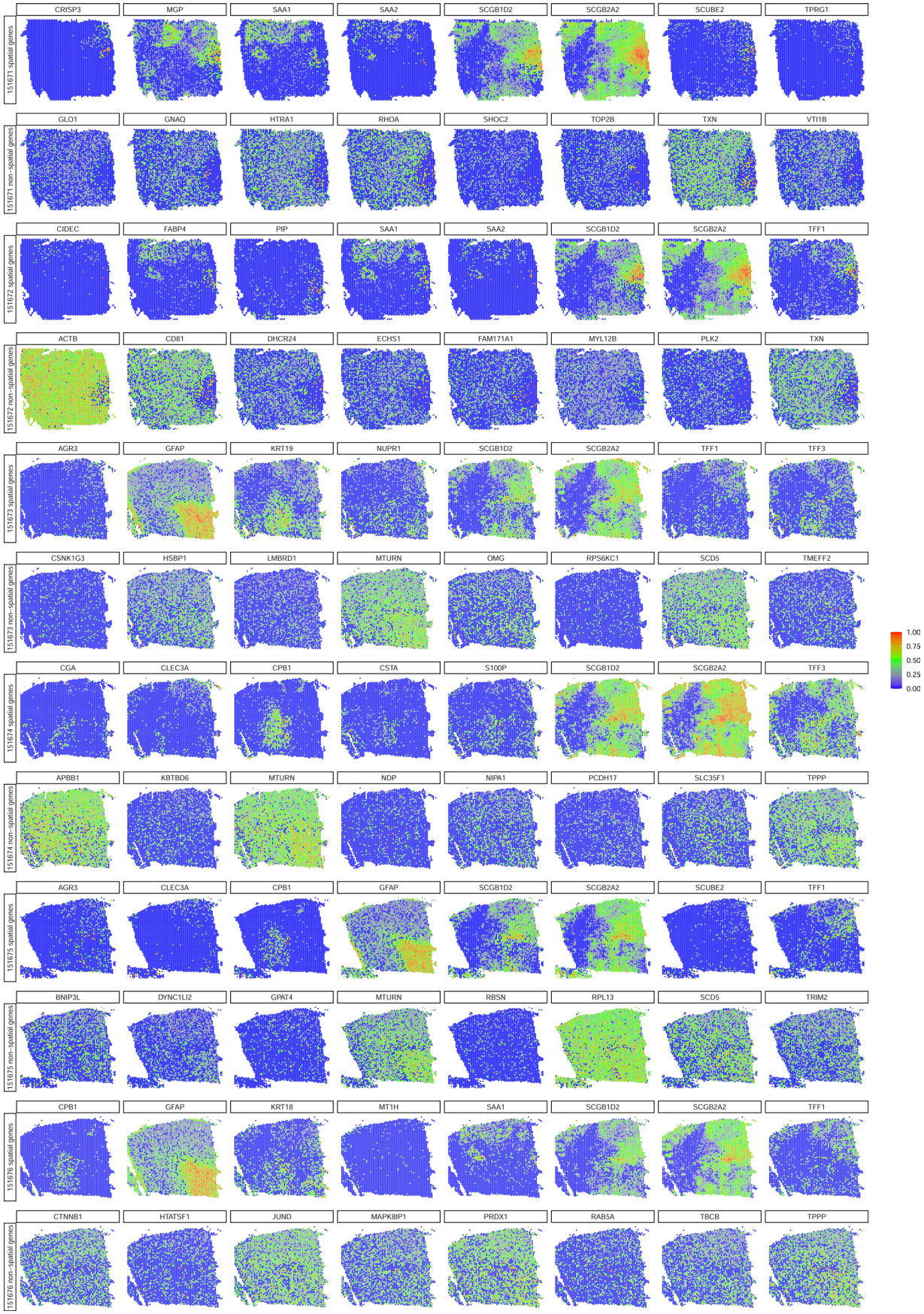
Extended Data Fig. 1 | spaVAE for integrating batches of 10X mouse anterior and posterior brains data. a, The spaVAE embedding of the mouse brain data with two regions (anterior and posterior) and two batches (section 1 and section 2), with colors and shapes denoting brain regions and batches. **b**, Clustering labels of the combined four samples, with colors denoting cluster labels. **c**, Alluvial plot of cluster proportions across different samples. **d**-**e**, Top genes identified (log fold change > 1 and Bayes factor > 10) by spaVAE within different clusters among the two sections of mouse anterior brain (**d**) and among the two sections of mouse posterior brain (**e**). Heatmaps display relative denoised averaged expression levels across the clusters.

https://doi.org/10.1038/s41592-024-02257-y



Extended Data Fig. 2 | Top spatially and nonspatially variable genes identified by spaLDVAE in the first 6 human DLPFC samples. Top 4000 highly variable genes are used for this analysis.

https://doi.org/10.1038/s41592-024-02257-y



Extended Data Fig. 3 | Top spatially and nonspatially variable genes identified by spaLDVAE in the last 6 human DLPFC samples. Top 4000 highly variable genes are used for this analysis.

nature portfolio

Corresponding author(s): Zhi Wei

Last updated by author(s): Jan 10, 2024

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	Softwares used for data preprocessing: SCANPY (https://github.com/scverse/scanpy) SPARK (https://github.com/xzhoulab/SPARK) SPARK-X (https://github.com/xzhoulab/SPARK) Described in the "Spatial genomics datasets" section of the "Methods" part of the manuscript Preprocessed spatial genomics datasets can be found at https://figshare.com/articles/dataset/Spatial_genomics_datasets/21623148
Data analysis	Described in the "Spatial genomics datasets" section of the "Methods" part of the manuscript An open-source software implementation of spaVAE, spaPeakVAE, spaMultiVAE, spaLDVAE and spaPeakLDVAE is available on Github: https://github.com/ttgump/spaVAE. It is also available at Zenodo repository: https://doi.org/10.5281/zenodo.8407637.

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

The data that support the findings of this study are available from the corresponding author upon request. Preprocessed spatial genomics datasets can be found at https://figshare.com/articles/dataset/Spatial_genomics_datasets/21623148

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	not applicable
Data exclusions	not applicable
Replication	not applicable
Randomization	not applicable
Blinding	not applicable

Reporting for specific materials, systems and methods

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

Materials & experimental systems

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

Methods

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