Using hierarchical variational autoencoders to incorporate conditional independent priors for paired single-cell multi-omics data integration

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

Thanks to recent advances in single-cell multi-omics sequencing technology such as CITE-Seq [1], scNMT-Seq [2], SNARE-Seq [3], SHARE-Seq [4] and 10X Multiome [5], it is now possible to capture multiple modalities of molecular data simultaneously at single-cell resolution. However, integration of multiple modalities and deriving biological insight from them is challenging, as each modality requires distinct data processing, normalization, modeling and interpretation [6]. To address this, various data analysis tools and computational methods have been proposed. For instance, Seurat [7], Harmony [8], Scanpy [9] and MUON [10] all provide a unified framework for quality control, data exploration and downstream analysis of single-cell data. Other methods such as scMVAE [11], Cobolt [12], scMM [13] and TotalVI [14] integrate multiple modalities of single-cell data by applying variational autoencoders to embed the observed data into a joint low-dimensional latent space. However, using a joint latent space renders it difficult to interpret the model and unravel the interactions between different modalities. Moreover, most of these published methodologies aim at direct multi-omics data integration. How to incorporate prior biological understanding of the properties of data into the existing model remains an open question in the field.

Here, we propose a novel probabilistic learning framework that explicitly incorporates the conditional independent relationships between multi-modal data using a generalized variational ladder autoencoder [15]. Specifically, our proposed framework takes as input the observed datasets and a directed acyclic graph representing the relationships between data. Then, we sequentially train each ladder based on the topological order in the provided graph to optimize the evidence lower bound of the data likelihood of each modality of the observed data. We anticipate our proposed framework could help construct flexible graphical models that reflect biological hypotheses with ease and unravel the interactions between different biological data types, such as different modalities of paired single-cell multi-omics data.

2 Methods

2.1 Problem Definition

Consider M different views (e.g. modalities) of data with the same anchor, and a directed acyclic graph $G = (V, E)$, where V is the set of vertices each representing a view of the data, and the E is the set of unweighted directed edges specifying our prior knowledge on the conditional independent relationships between different views. Our proposed method aims at learning the dependency between the latent distributions across different views explicitly based on G.

For paired single-cell multi-omics sequencing data, each view represents a single molecular assay anchored by the barcodes of cells. The provided graphical model to our method could, for example,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

be the states of chromatin accessibility that may facilitate regulation of gene expression [16] (Figure 1b).

Note that although here we showcase our proposed method in the context of paired single-cell multi-omics data, it can also be applied to single-omics data derived from single-cells or bulk tissues. For instance, single-cell expression data can be separated into multiple views representing the expression of transcription factors or miRNAs and their target genes [17]. (Figure 1c). For other potential use cases, please refer to Figure 1.

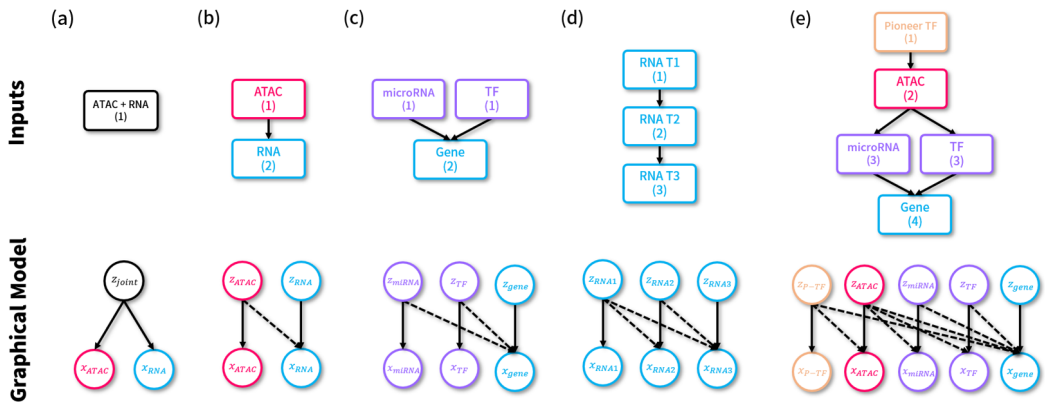

Figure 1: Examples of relational graphs and graphical models.

2.2 Probabilistic Generative Model

Inspired by a recently published method designed for multi-facet clustering in computer vision [18], we applied a generalized version of a variational ladder autoencoder to incorporate prior knowledge of the conditional independent relationships between biological data into a probabilistic learning framework. This approach allows multiple cluster assignments that conditionally separate the abstract features of each view of the data, based on a provided relational graph. This is particularly helpful to identify new clusters of cell types.

Consider M different views of the observed data x_m, where $m = 1, 2...M$, and a directed acyclic graph G with vertices $V = \{v_m \mid m = 1, 2...M\}$ and unweighted directed edges $E \subseteq V \times V$. Our proposed model approximates the conditional generative process of the observed data with:

$$
\begin{aligned}
c_m &\sim Categorical(\pi_m) \\
z_m &\sim IndependentNormal(\boldsymbol{\mu}_{c_m}, \boldsymbol{\Sigma}_{c_m}) \\
\tilde{z}_m &= ConcatParents(m) \\
\bar{z}_m &= f_m^{(r)}(z_m; \theta_m^{(r)}) \\
\hat{z}_m &= f_m^{(b)}([\bar{z}_m; \tilde{z}_m]; \theta_m^{(b)}) \\
p_m &= f_m^{(d)}(\hat{z}_m; \theta_m^{(d)}) \\
x_m &\sim Dist(p_m)
\end{aligned}
\tag{1}
$$

where c_m is the cluster assignment for the mixture of independent Gaussian distributions with corresponding mean μ_{c_m} and standard deviation $\boldsymbol{\Sigma}_{c_m}$. z_m is the latent representation of view m. $ConcatParents$ is a function concatenating the latent representations $\hat{z}_{\mathcal{P}}$ of the parents $\mathcal{P}$ of the m-th view. $f_m^{(b)}$ and $f_m^{(r)}$ are linear transformations with parameters $\theta_m^{(b)}$ and $\theta_m^{(r)}$. $f_m^{(d)}$ is the decoder neural network used to parameterize the specified data distribution. p_m is the set of parameters of user-defined data distributions $Dist$ used to compute the data likelihood.

The model is trained by optimizing the evidence lower bound (ELBO) of the data likelihood for each view sequentially based on the order of views in the topological sort of provided relational graph G

(Appendix C). For the m-th view of the data and its ancestors $\mathcal{A}$ in the provided graph G, the ELBO is:

$$\log p(x_m) \geq \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B}, \mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log(\frac{p_\theta(x_m, \mathbf{z}_\mathcal{B}, \mathbf{c}_\mathcal{B})}{q_\phi(\mathbf{z}_\mathcal{B}, \mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})})] = \mathcal{L}(x_m; \theta, \phi) \tag{2}$$

, where $q_\phi(z_m, c_m|x_m)$ is parameterized by a encoder neural network $f_m^{(e)}$, $\mathcal{B}$ is the union of ancestors $\mathcal{A}$ and m-th view of the data. For the online multi-facet clustering, we applied the single-facet VaDE trick [18, 19] sequentially to identify the cluster assignments of each data point. Intuitively, the proposed framework can be thought of as a hierarhical multi-modal variational autoencoder [20] generalized to a directed acyclic graph with mixture-of-Gaussian priors [19]. For a more detailed derivation, please refer to Appendix A and B.

2.3 Implementation Details

Previous studies have shown that *progressive training* improves the capability of the model to learn disentangled representations across layers in hierarchical variational autoencoder [18, 21]. We found that it is particularly helpful to prevent learning redundant representations from the ancestors of the view. Moreover, as shown in Method 2.2, we applied *sequential training* based on the order of topological sort of the provided graph (Appendix C). We found that this complements the progressive training and prevents the ancestors from learning representations uniquely relevant to the predecessors.

For the preliminary analysis presented in this study, we trained the model with 1000 epochs using the Adam optimizer [22] with a learning rates of $5e-4$. The dimensions for the latent distribution $dim(\mathbf{z}) = 30$. The neural architecture is shown in Appendix Table S1. To ensure the model is identifiable, we set the number of components in the mixture of independent Gaussian priors to $K = 2 \times \dim(\mathbf{z}) + 1 = 61$ [23, 24]. For a more detailed description of the implementation of the framework, please refer to Appendix D.

2.4 Datasets

For the preliminary analysis presented in this study, we used the SNARE-Seq data from the cerebral cortex of an adult mouse (accession number GSE126074) [3]. These data include views on both chromatin state and gene expression data in the same cell. We used the Bernoulli distribution to model the binarized chromatin accessibility data and the zero-inflated negative binomial distribution to model the gene expression data similar to previous studies [25, 26]. We annotated cell types using marker genes from the single-cell transcriptomic reference dataset of the mouse visual cortex from the Allen Brain Atlas [27]. We identified these genes using the *FindTransferAnchors* function in Bioconductor package Seurat (version 4.1.0) [7].

3 Preliminary Results

We hypothesize that chromatin accessibility states facilitate the regulation of gene expression, and incorporate a directed acyclic graph (Figure 1b) indicating this dependency into our proposed framework. We applied our approach to SNARE-Seq data obtained from the cerebral cortex of an adult mouse (see Method 2.4 for more detail). We found that our method is capable of identifying different clustering assignments based on the given conditional independent relationships between data (Figure 2c, 2d). In particular, our method isolates the source of variability between data types and finds a distinct cluster of Pvalb cell types in the latent representations of gene expression data conditioned on chromatin accessibility (Figure 2b). This implies that the distinction between Pvalb cells from other cell types is not specifically caused by changes in the states of chromatin accessibility, but rather by other regulatory mechanisms. Compared to the latent representations identified from other prior relational graphs that embed the two modalities into a shared latent space or two independent latent spaces, our method can highlight clusters of interest based on the biological hypothesis of interactions between data (Appendix Figure S3). The online multi-facet clustering also identify distinct clusters in an unsupervised way for Pvalb cells (Figure S4)

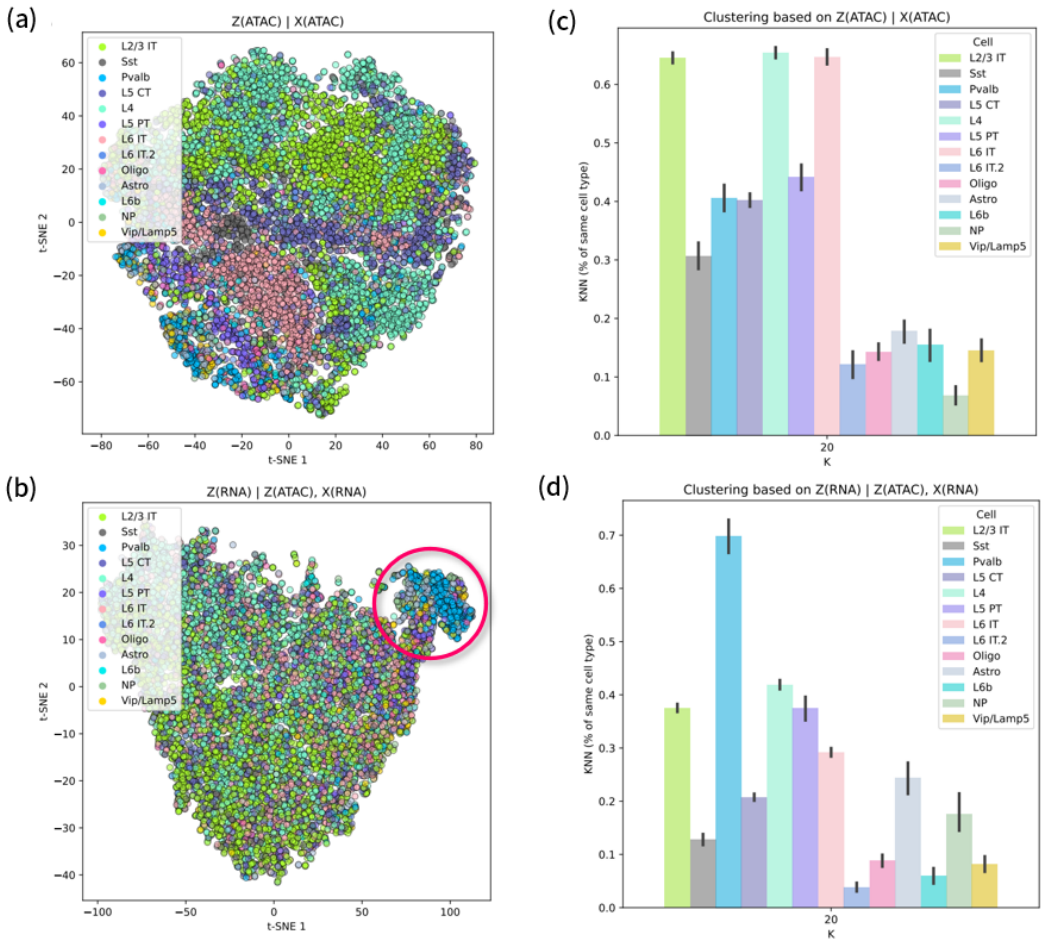

Figure 2: Multi-facet clustering for SNARE-Seq cerebral cortex of an adult mouse dataset. The left panel is the t-SNE visualization of the latent representations colored by annotated cell types. The right panel is the proportion of 20-nearest neighbors that shared the same annotated cell types.

4 Discussions and Future Works

We propose a novel probabilistic learning framework to explicitly incorporate prior biological understanding of the relationships between views of data as a directed acyclic graph into the graphical model for data integration. We showed that our method can integrate paired single-cell multi-omics datasets and identify cell clusters of interests based on the provided hypothesis. We plan to further characterize such cell clusters through gene expression analyses. To further facilitate downstream analysis, we plan to implement functionalities such as interactive figure rendering to visualize the latent representations, differential expression analysis using the Bayesian factor [25], batch correction [25], model interpretation using attributions derived from gradients [28, 29] and conditional paired single-cell multi-omics data generation. In the future, this method can be applied to other paired single-cell multi-omics datasets from, for example, 10X Multiome and CITE-Seq to discover more biological insights.

References

[1] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature methods*, 14(9):865–868, 2017.

[2] Stephen J Clark, Ricard Argelaguet, Chantriolnt-Andreas Kapourani, Thomas M Stubbs, Heather J Lee, Celia Alda-Catalinas, Felix Krueger, Guido Sanguinetti, Gavin Kelsey, John C Marioni, et al. scnmt-seq enables joint profiling of chromatin accessibility dna methylation and transcription in single cells. *Nature communications*, 9(1):1–9, 2018.

[3] Song Chen, Blue B Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature biotechnology*, 37(12):1452–1457, 2019.

[4] Sai Ma, Bing Zhang, Lindsay M LaFave, Andrew S Earl, Zachary Chiang, Yan Hu, Jiarui Ding, Alison Brack, Vinay K Kartha, Tristan Tay, et al. Chromatin potential identified by shared single-cell profiling of rna and chromatin. *Cell*, 183(4):1103–1116, 2020.

[5] Kamila Belhocine, Laura DeMare, and Olivia Habern. Single-cell multiomics: Simultaneous epigenetic and transcriptional profiling: 10x genomics shares experimental planning and sample preparation tips for the chromium single cell multiome atac+ gene expression system. *Genetic Engineering & Biotechnology News*, 41(1):66–68, 2021.

[6] Ricard Argelaguet, Anna SE Cuomo, Oliver Stegle, and John C Marioni. Computational principles and challenges in single-cell data integration. *Nature biotechnology*, 39(10):1202–1215, 2021.

[7] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck III, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[8] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[9] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018.

[10] Danila Bredikhin, Ilia Kats, and Oliver Stegle. Muon: multimodal omics analysis framework. *Genome Biology*, 23(1):1–12, 2022.

[11] Chunman Zuo and Luonan Chen. Deep-joint-learning analysis model of single cell transcriptome and open chromatin accessibility data. *Briefings in Bioinformatics*, 22(4):bbaa287, 2021.

[12] Boying Gong, Yun Zhou, and Elizabeth Purdom. Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome biology*, 22(1):1–21, 2021.

[13] Kodai Minoura, Ko Abe, Hyunha Nam, Hiroyoshi Nishikawa, and Teppei Shimamura. A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. *Cell reports methods*, 1(5):100071, 2021.

[14] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature methods*, 18(3):272–282, 2021.

[15] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017.

[16] Sandy L Klemm, Zohar Shipony, and William J Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 2019.

[17] Kevin Chen and Nikolaus Rajewsky. The evolution of gene regulation by transcription factors and micrornas. *Nature Reviews Genetics*, 8(2):93–103, 2007.

[18] Fabian Falck, Haoting Zhang, Matthew Willetts, George Nicholson, Christopher Yau, and Chris C Holmes. Multi-facet clustering variational autoencoders. *Advances in Neural Information Processing Systems*, 34:8676–8690, 2021.

[19] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[20] Jannik Wolff, Rahul G Krishnan, Lukas Ruff, Jan Nikolas Morshuis, Tassilo Klein, Shinichi Nakajima, and Moin Nabi. Hierarchical multimodal variational autoencoders. 2021.

[21] Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. *arXiv preprint arXiv:2002.10549*, 2020.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[24] Matthew Willetts and Brooks Paige. I don't need u: Identifiable non-linear ica without side information. *arXiv preprint arXiv:2106.05238*, 2021.

[25] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[26] Tal Ashuach, Daniel A Reidenbach, Adam Gayoso, and Nir Yosef. Peakvi: A deep generative model for single-cell chromatin accessibility analysis. *Cell reports methods*, 2(3):100182, 2022.

[27] Bosiljka Tasic, Zizhen Yao, Lucas T Graybuck, Kimberly A Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N Economo, Sarada Viswanathan, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, 2018.

[28] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[30] Isaac Virshup, Sergei Rybakov, Fabian J Theis, Philipp Angerer, and F Alexander Wolf. anndata: Annotated data. *bioRxiv*, 2021.

[31] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[32] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Decomposition of ELBO

For the m-th view of the data and its ancestors $\mathcal{A}$ in the provided graph G, the evidence lower bound (ELBO) of the data likelihood is shown in Equation S1. For simplicity, we use $\mathcal{B}$ to denote the union of ancestors $\mathcal{A}$ and the m-th view.

$$\begin{aligned}
\log p(x_m) &= \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log(\frac{p_\theta(x_m,\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B})}{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})})] + \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log(\frac{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}{p_\theta(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|x_m)})] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log(\frac{p_\theta(x_m,\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B})}{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})})] + D_{\mathrm{KL}}(q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})||p_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|x_m)) \qquad \text{(S1)} \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log(\frac{p_\theta(x_m,\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B})}{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})})] = \mathcal{L}(x_m;\theta,\phi)
\end{aligned}$$

, where $q_\phi(z_m, c_m|x_m)$ is parameterized by a encoder neural network $f_m^{(e)}$. Assuming independence between latent representations, $p_\theta(z_m|\mathbf{z}_\mathcal{A}) = p_\theta(z_m)$, the generative model $p_\theta(x_m, \mathbf{z}_\mathcal{B}, \mathbf{c}_\mathcal{B})$ and the posterior $q_\phi(\mathbf{z}_\mathcal{B}, \mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})$ can be further expanded:

$$\begin{aligned}
p_\theta(x_m,\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}) &= p_\theta(x_m|z_m,\mathbf{z}_\mathcal{A})p_\theta(z_m|c_m)p_\theta(c_m)p_\theta(\mathbf{z}_\mathcal{A}|\mathbf{c}_\mathcal{A})p_\theta(\mathbf{c}_\mathcal{A}) \\
&= p_\theta(x_m|z_m,\mathbf{z}_\mathcal{A})p_\theta(c_m|z_m)p_\theta(z_m)p_\theta(\mathbf{c}_\mathcal{A}|\mathbf{z}_\mathcal{A})p_\theta(\mathbf{z}_\mathcal{A}) \\
q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B}) &= q_\phi(\mathbf{z}_\mathcal{B}|\mathbf{x}_\mathcal{B})q_\phi(\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B}) \\
&= q_\phi(z_m|x_m)q_\phi(c_m|x_m)q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})
\end{aligned} \qquad \text{(S2)}$$

substitute back to Equation (S1), we can rewrite the ELBO as:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log(\frac{p_\theta(x_m|z_m,\mathbf{z}_\mathcal{A})p_\theta(c_m|z_m)p_\theta(z_m)p_\theta(\mathbf{c}_\mathcal{A}|\mathbf{z}_\mathcal{A})p_\theta(\mathbf{z}_\mathcal{A})}{q_\phi(z_m|x_m)q_\phi(c_m|x_m)q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})})] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{B},\mathbf{c}_\mathcal{B}|\mathbf{x}_\mathcal{B})}[\log p_\theta(x_m|z_m,\mathbf{z}_\mathcal{A})] - \\
&\quad \mathbb{E}_{q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(z_m|x_m)q_\phi(c_m|x_m)}D_{\mathrm{KL}}(q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})||p_\theta(\mathbf{z}_\mathcal{A})) - \\
&\quad \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(c_m|x_m)}D_{\mathrm{KL}}(q_\phi(z_m|x_m)||p_\theta(z_m)) - \\
&\quad \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(z_m|x_m)q_\phi(c_m|x_m)}D_{\mathrm{KL}}(q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})||p_\theta(\mathbf{c}_\mathcal{A}|\mathbf{z}_\mathcal{A})) - \\
&\quad \mathbb{E}_{q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})q_\phi(z_m|x_m)}D_{\mathrm{KL}}(q_\phi(c_m|x_m)||p_\theta(c_m|z_m))
\end{aligned} \qquad \text{(S3)}$$

Note that the model is trained by optimizing the ELBO of the data likelihood for each view sequentially based on the order of views in the topological sort of provided relational graph G (Appendix C). Therefore, when training the m-th view of the data, $q_\phi(\mathbf{c}_\mathcal{A}|\mathbf{x}_\mathcal{A})$ and $q_\phi(\mathbf{z}_\mathcal{A}|\mathbf{x}_\mathcal{A})$ are considered as constants (non-trainable).

B Multi-facet Clustering

To identify the optimal cluster assignment for the posterior distribution $q_\phi(c_m|x_m)$, we applied the single-facet VaDE trick [18, 19] to derive the optimal $q_\phi^*(c_m|x_m)$ that minimizes $D_{\mathrm{KL}}(q_\phi(c_m|x_m)||p_\theta(c_m|z_m))$:

$$\begin{aligned}
&\underset{q_\phi(c_m|x_m)}{\operatorname{argmin}} D_{\mathrm{KL}}(q_\phi(c_m|x_m)||p_\theta(c_m|z_m)) = \frac{\exp\left(\mathbb{E}_{q_\phi(z_m|x_m)}[\log p(c_m|z_m)]\right)}{Z(q_\phi(c_m|x_m))} \\
&, Z(q_\phi(c_m|x_m)) = \sum_{c=1}^{K} \exp\left(\mathbb{E}_{q_\phi(z_m|x_m)}[\log p(c_m|z_m)]\right)
\end{aligned} \qquad \text{(S4)}$$

, where K is the number of components in the mixture of independent Gaussian priors in view m.

C Sequential Training

As shown in Method 2.2, 2.3 and Appendix A, we apply sequential training based on the order of views in the topological sorting of the provided relational graph G (the numbers of vertices in Figure 1). Specifically, when training m-th view with its ancestor $\mathcal{A}$, we fixed the weights in $f_{\mathcal{A}}^{(e)}$, $f_{\mathcal{A}}^{(d)}$, $f_{\mathcal{A}}^{(r)}$ and $f_{\mathcal{A}}^{(b)}$. So the posterior distribution and the generative model for the ancestors remains the same. This prevents the ancestors to learn representations uniquely to the m-th view posterior distribution. Note that, the decoder of view m, $f_m^{(d)}$, takes the sampling values from the ancestors $q_\phi(\mathbf{z}_{\mathcal{A}}|\mathbf{x}_{\mathcal{A}})$, and the posterior $q_\phi(z_m|x_m)$ would be conditioned on the posterior of the ancestors. Theoretically the predecessors would naturally have more expressive decoders. To make the framework even more general, we also allow the posterior of a specific view to be conditioned only on the parents $\mathcal{P}$ instead of all the ancestors $\mathcal{A}$.

D Implementation Details

The inputs and outputs format of our proposed framework is compatible with widely used package such as AnnData [30], Scanpy and MUON in the field of computational biology (Figure S1). The created graphical model is implemented using Tensorflow 2.8.0 sequential and functional APIs [31]. Each view of the data is handled by a component module with custom data preprocessor, encoder $f_m^{(e)}$, hierarchical encoder ($f_m^{(r)}$ and $f_m^{(b)}$) and decoder $f_m^{(d)}$ (Figure S2). Note that one component can potentially take multiple views as inputs (i.e. component can be used as a standalone multi-modal variational autoencoder with a joint latent space.)

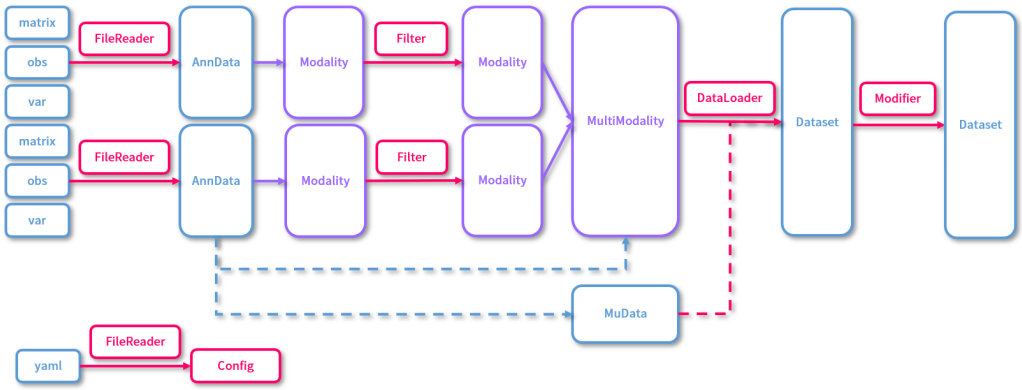

Figure S1: Examples of relational graphs and graphical models.

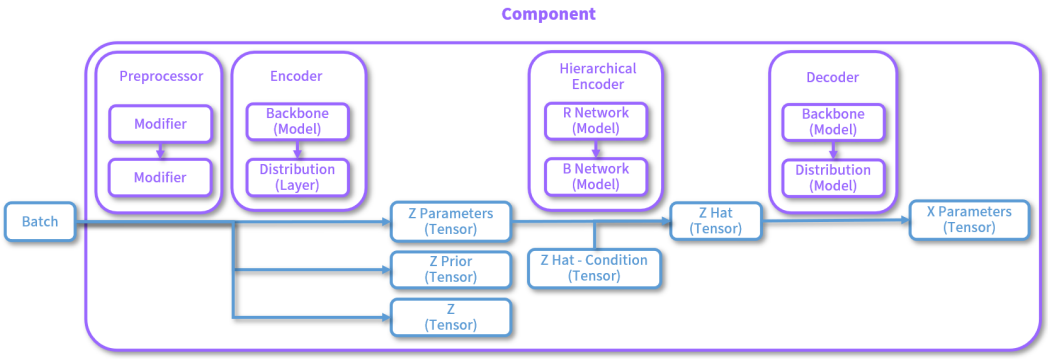

Figure S2: Examples of relational graphs and graphical models.

To facilitate the future development and broaden the applications of this framework, we provide high-level APIs as command line tools for the researchers to use, mid-level APIs that allow the

programmers to customize the model with parameters in the constructor functions, and low-level APIs for developers who wish to design their own data distribution, preprocessing steps, loss function or architecture of the neural network with modification for builder function or inheritance from the abstract classes. The default neural network architecture is shown in Table S1.

Table S1: Default neural network architecture

Name	Description
$f_m^{(b)}$	$\dim(ConcatParents(m)) \times \dim(ConcatParents(m))$ with linear activation
$f_m^{(r)}$	$\dim([z_m; \tilde{z}_m]) \times \dim([z_m; \tilde{z}_m])$ with linear activation
$f_m^{(d)}$	$\dim([z_m; \tilde{z}_m]) \times 128$ with ELU activation[1]
	128×256 with ELU activation
	256×512 with ELU activation
	$512 \times (\dim(\mathbf{x}_m) \times$ number of parameters$)$[2]
$f_m^{(e)}$	$\dim(\mathbf{x}_m) \times 1024$ with linear activation
	1024×512 with ELU activation
	512×256 with ELU activation
	256×128 with ELU activation
	$128 \times (2 \times \dim(\mathbf{z}_m))$ with ELU activation

[1] exponential linear unit [32]

[2] with specific activation functions according to the properties of parameters.

E Independent and Joint Modeling

As a comparative analysis, we apply our framework in two other different ways (1) embedding the single-cell gene expression data from the SNARE-Seq datasets into latent distributions independently from the chromatin accessibility data (Figure S3a). (2) embedding both single-cell gene expression and chromatin accessibility data into a shared joint latent space (Figure S3b). To make a fair comparison, the dimension of the latent space is set to $dim(\mathbf{z}) = 60$.

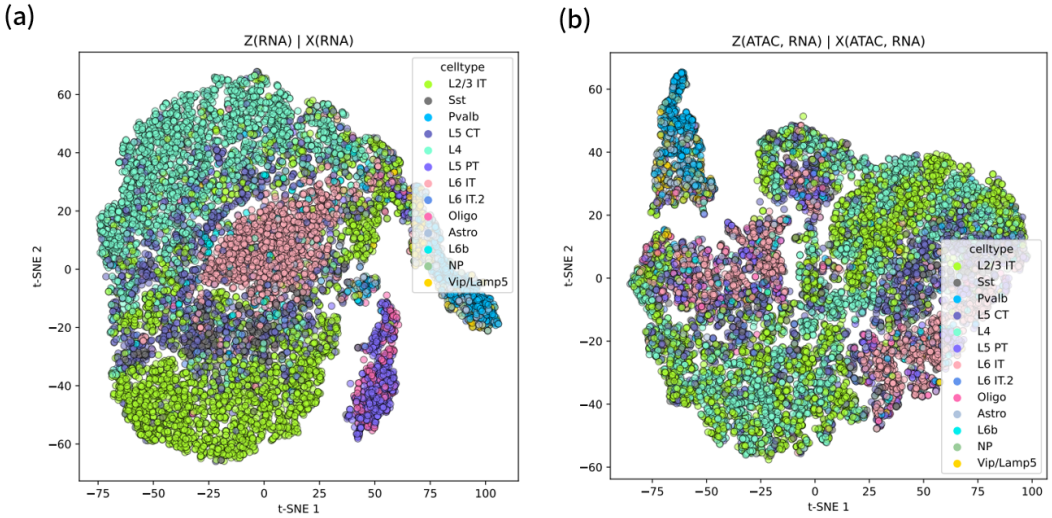

Figure S3: Independent and joint modeling

The cell clusters from independent latent space based on gene expression data mainly reflect the cell types. However, we can only conclude that the distinction between Pvalb cells and other cells is because of different expression profile. The connection between different expression profiles and chromatin accessibility is unknown. In contrast, the joint modeling also identifies a distinct clusters for Pvalb cells, but since the model embed the data into a shared latent space, it cannot isolate the

source of variability between data types. We also found that joint embedding approach requires more training epochs to reach convergence (2000 epochs). For both approaches, without the prior information on the relationships between data, it may become difficult to draw conclusions from the observed cell clusters without additional downstream analysis on the clusters.

F Online Multi-facet Clustering Analysis

Here, we show that the model is capable of learning different clustering assignment in different view. The rand index is computed by comparing with the annotated cell types. Using the latent representations on the gene expression conditionally independent from the chromatin accessibility, our model mainly assign Pvalb cells to cluster 6 in an unsupervised way.

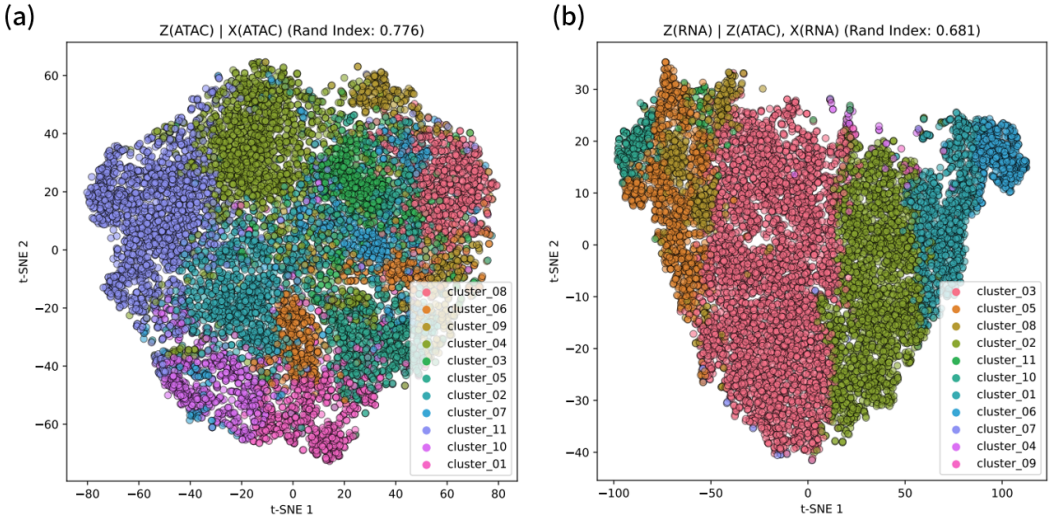

Figure S4: Result for online multi-facet clustering