

MULTI-OMICS SINGLE-CELL DATA INTEGRATION AND REGULATORY INFERENCE WITH GRAPH-LINKED EMBEDDING

Zhi-Jie Cao and Ge Gao, Nature Biotechnology (2022)

Background

The growing array of biological signals measurable at single-cell resolution is expanding rapidly.

However, methods that can capture all of these signals in the same single cell remain scarce.

Gene Expression (scRNA)

ATAC (scATAC)

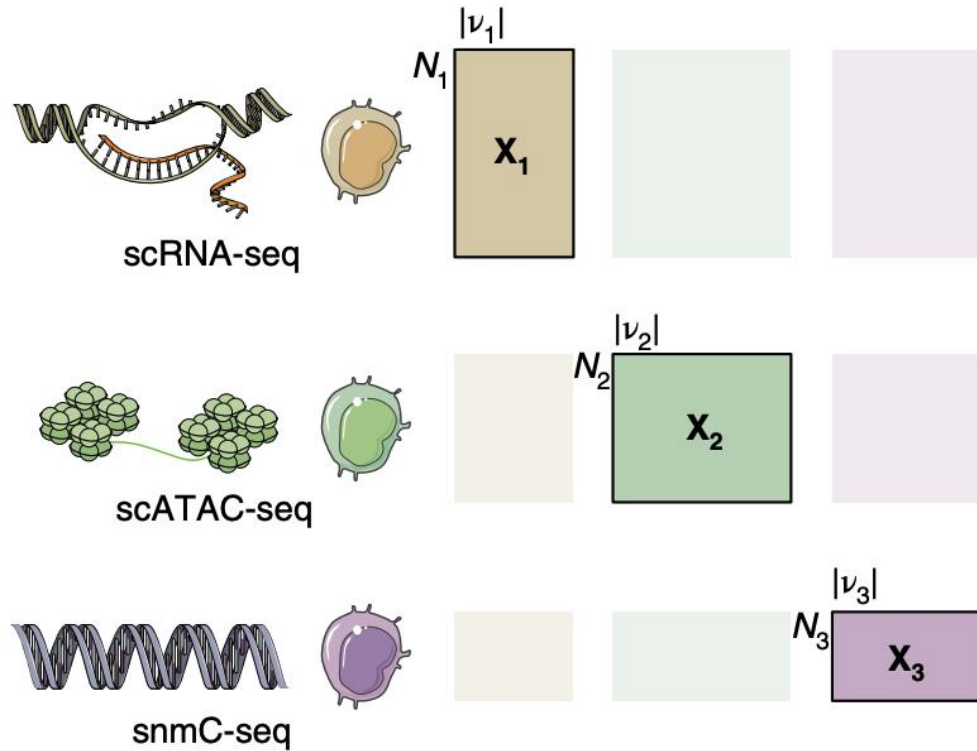
Histone Mark (scCUT&Tag)

Hi-C (scHI-C)

Methylation (snmC-seq)

10x Multiome, SHARE-seq, SNARE-seq

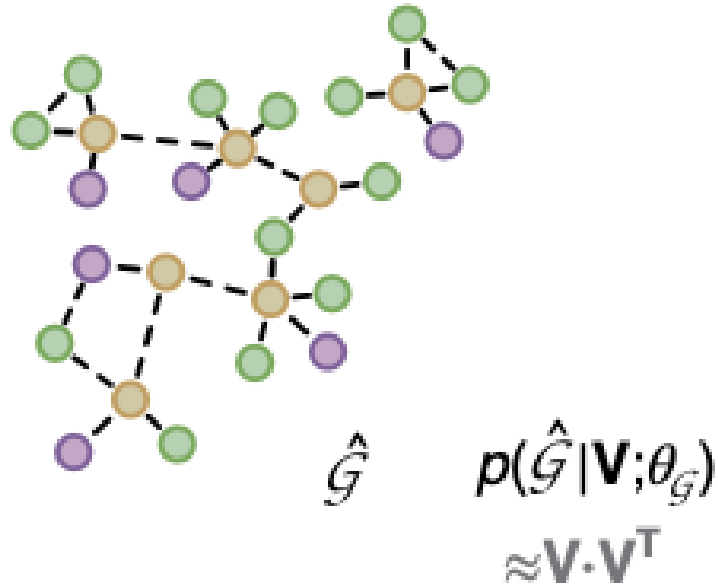
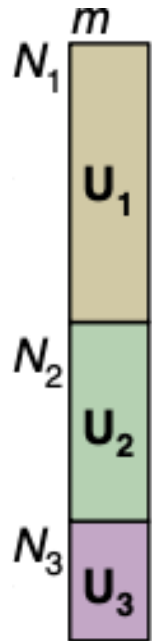
Background



Even though we got the three datasets, they are all different cells, and all different features

So, we like to do the multi-omics analysis by **Integrating** un-paired diagonal shape data.

Background



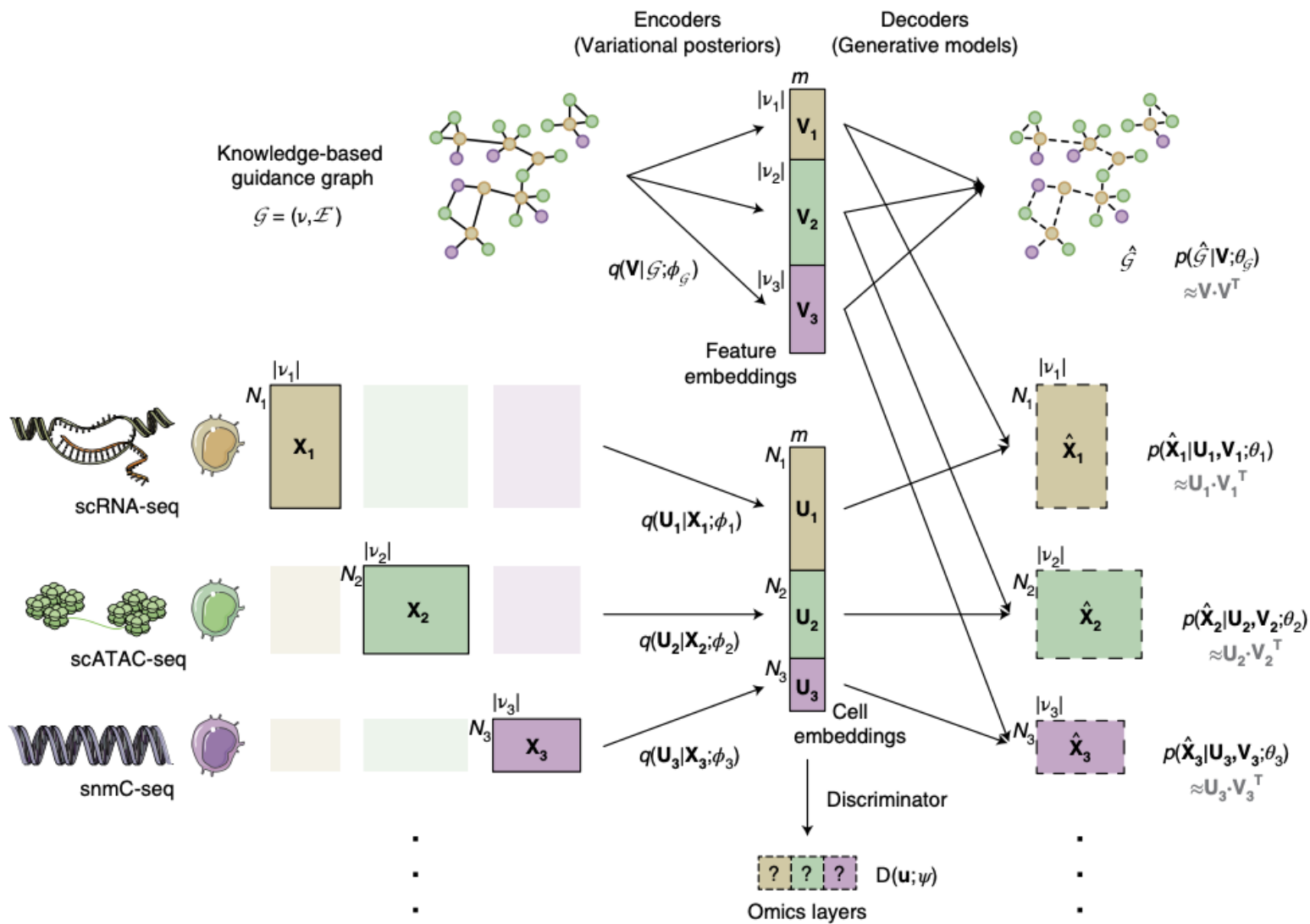
The result is

Integrated Latent Variable Across the datasets

Adjusted Feature-wise Interaction Network

Methods

GLUE (graph-linked unified embedding)

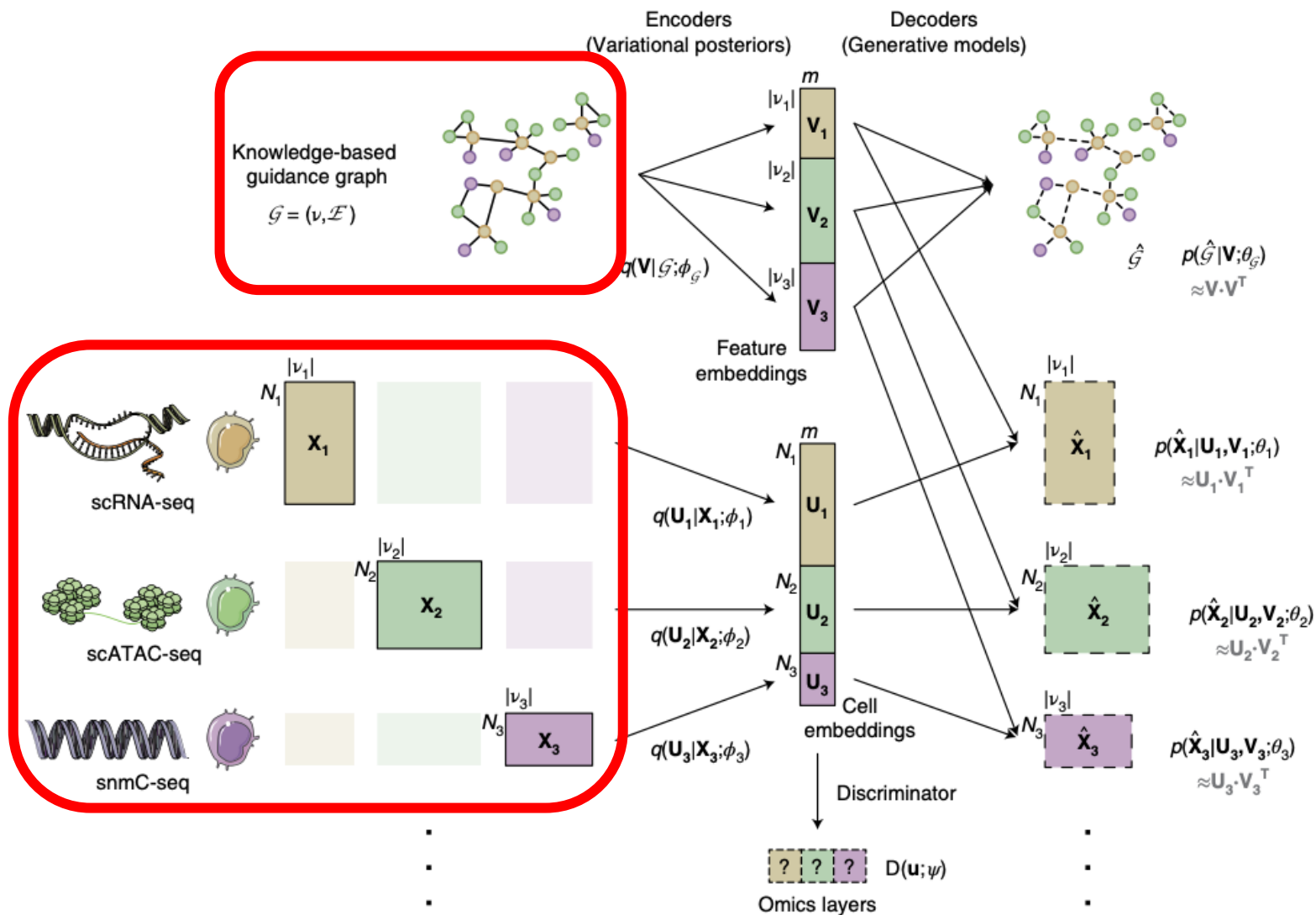


Methods

What we have

Feature-Feature Interaction Graph

Multomics Datasets



Methods

What we have

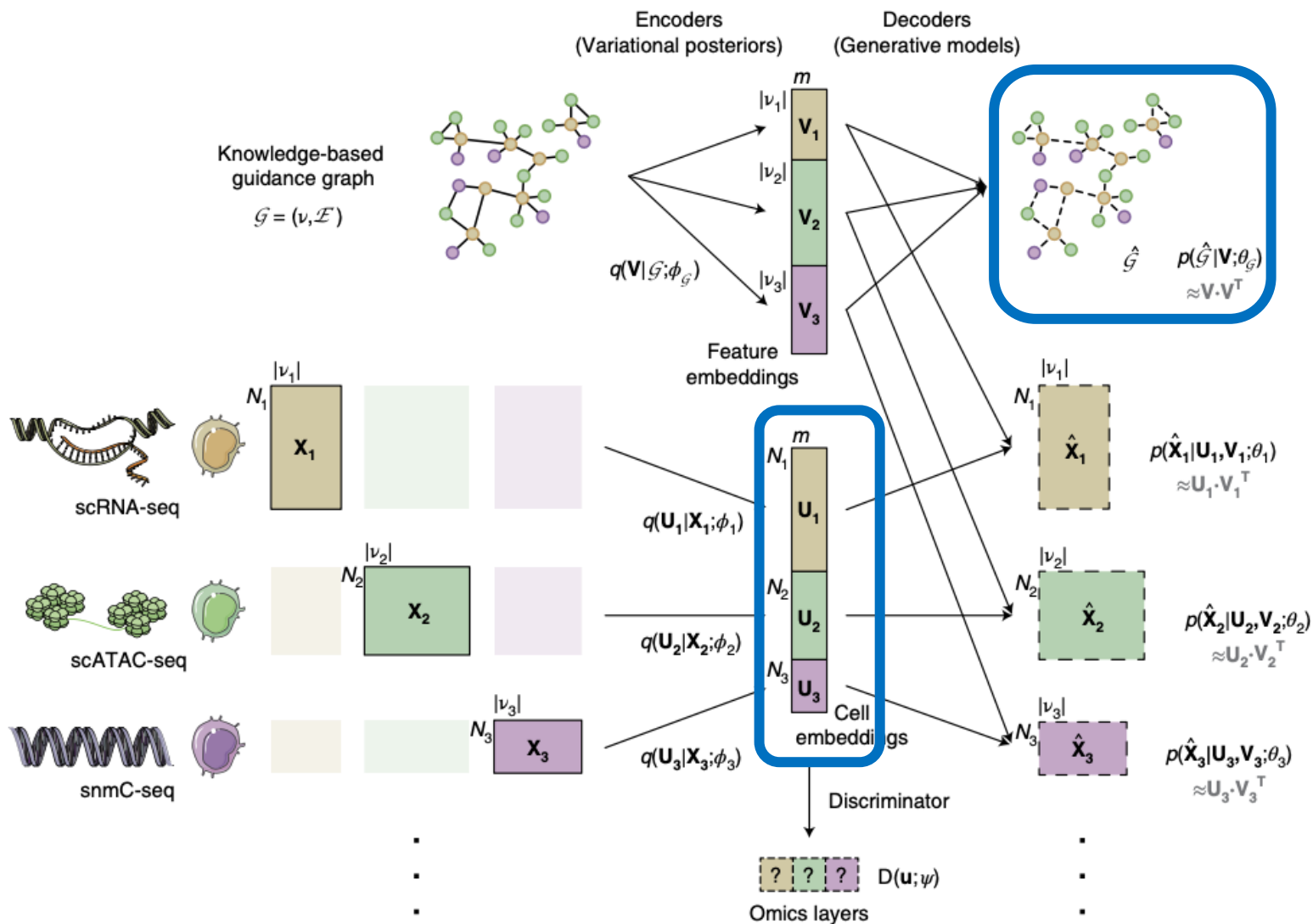
Feature-Feature Interaction Graph

Multomics Datasets

What we will get

Updated Interaction Graph

Integrated New Axis



Methods

What we have

Feature-Feature Interaction Graph

Multomics Datasets

What we will get

Updated Interaction Graph

Integrated New Axis

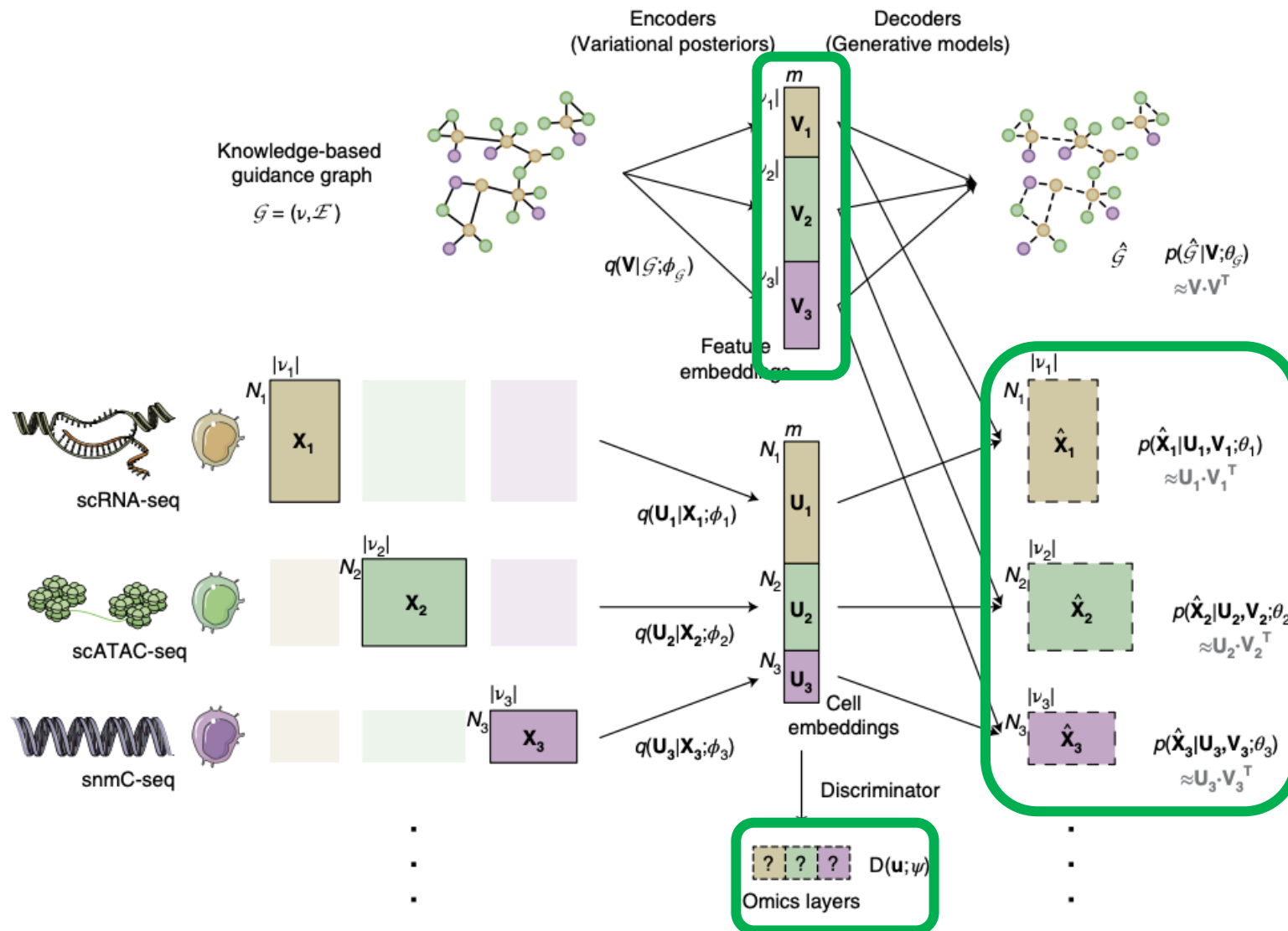
What the model uses

(Things you probably don't need to care about)

Graph embeddings

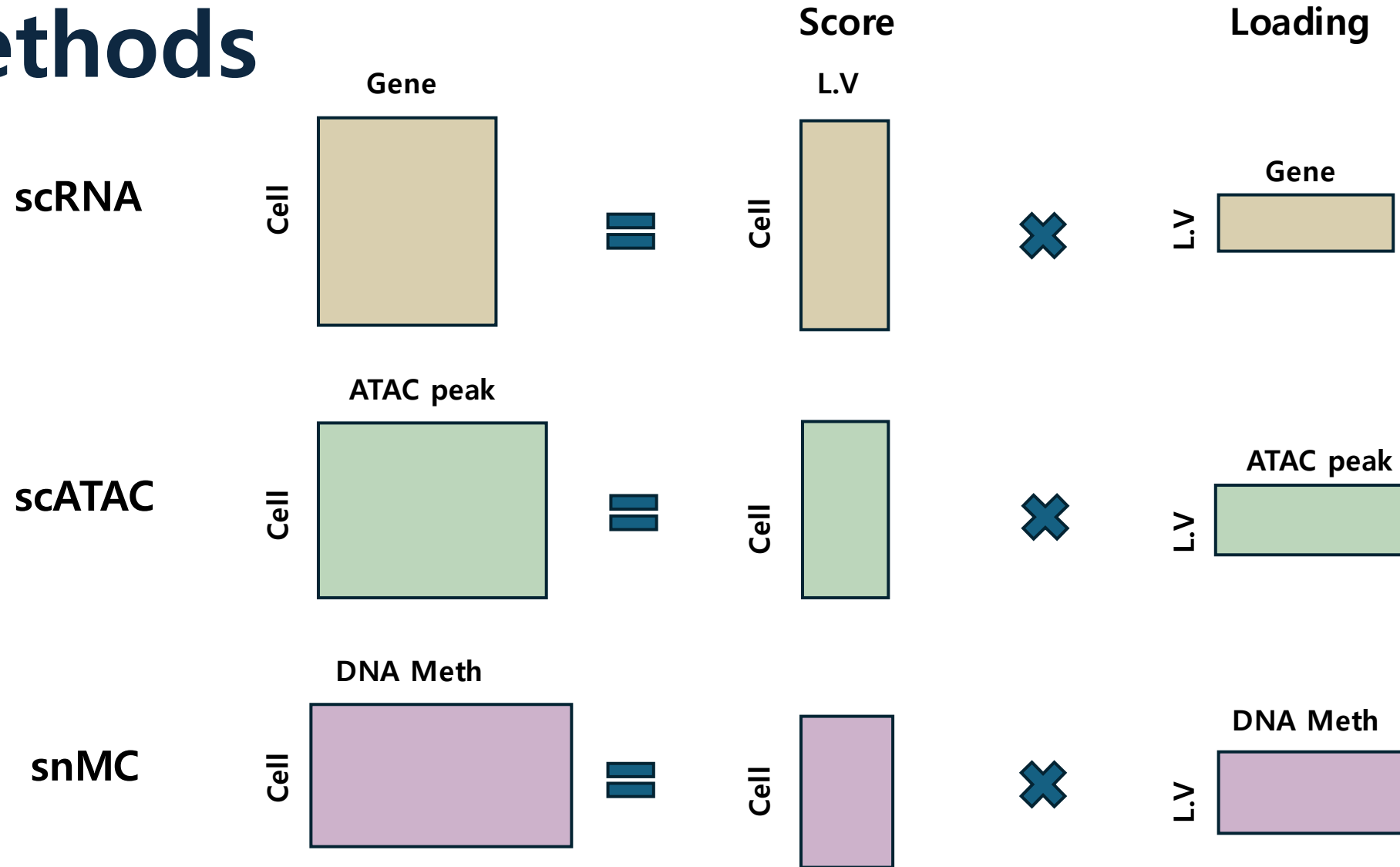
Reconstructed data

Multi-omics source prediction vectors



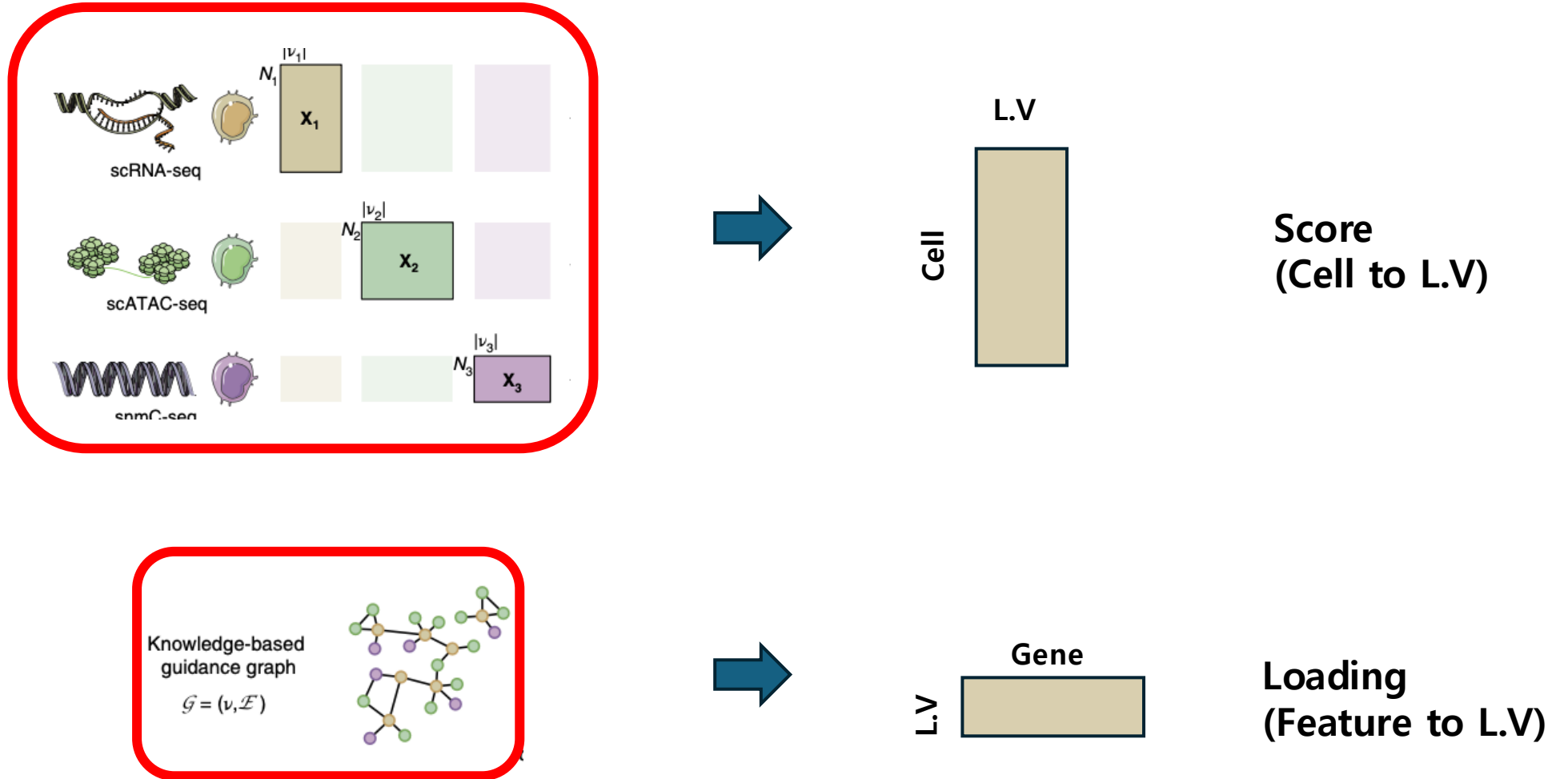
Methods

L.V = Latent Variable



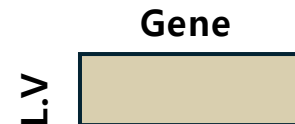
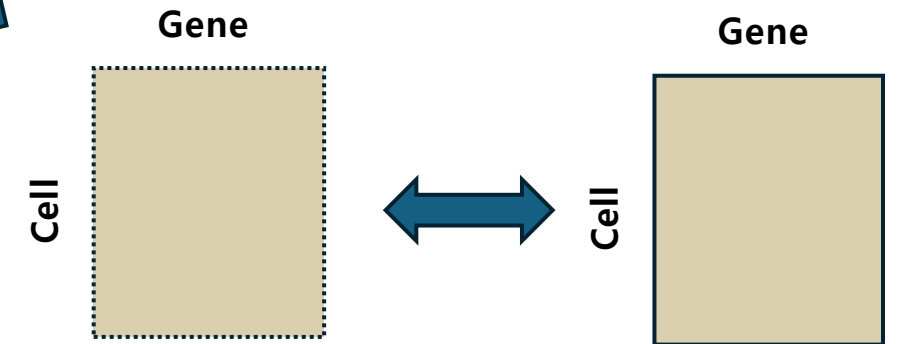
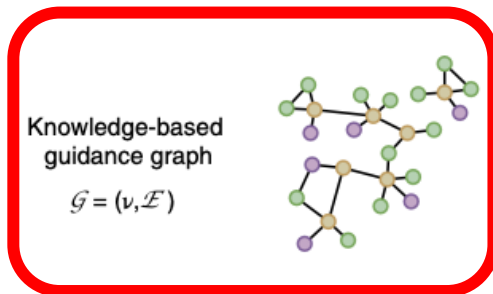
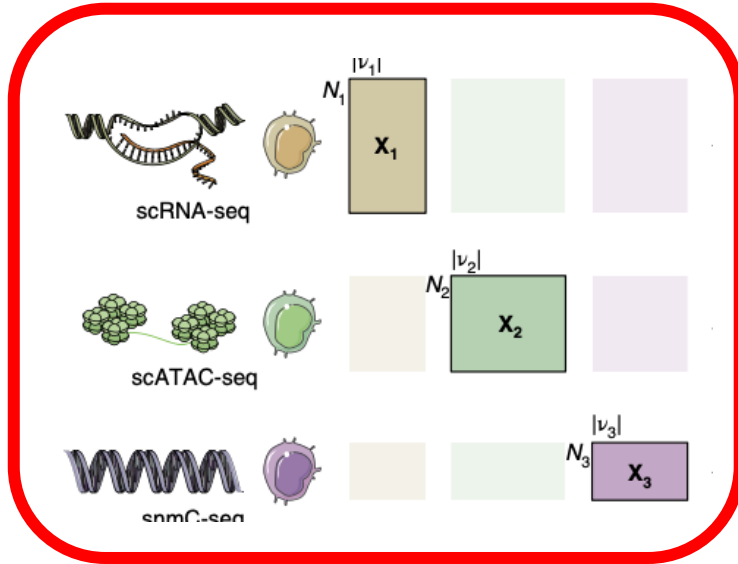
Assume all the dataset's features come from some *hidden* , *latent* variables
Each of Latent Vector connected with *Cells* and *Feature*

Methods



Encoding Process – Using the *Raw Data* to construct *Latent Space*

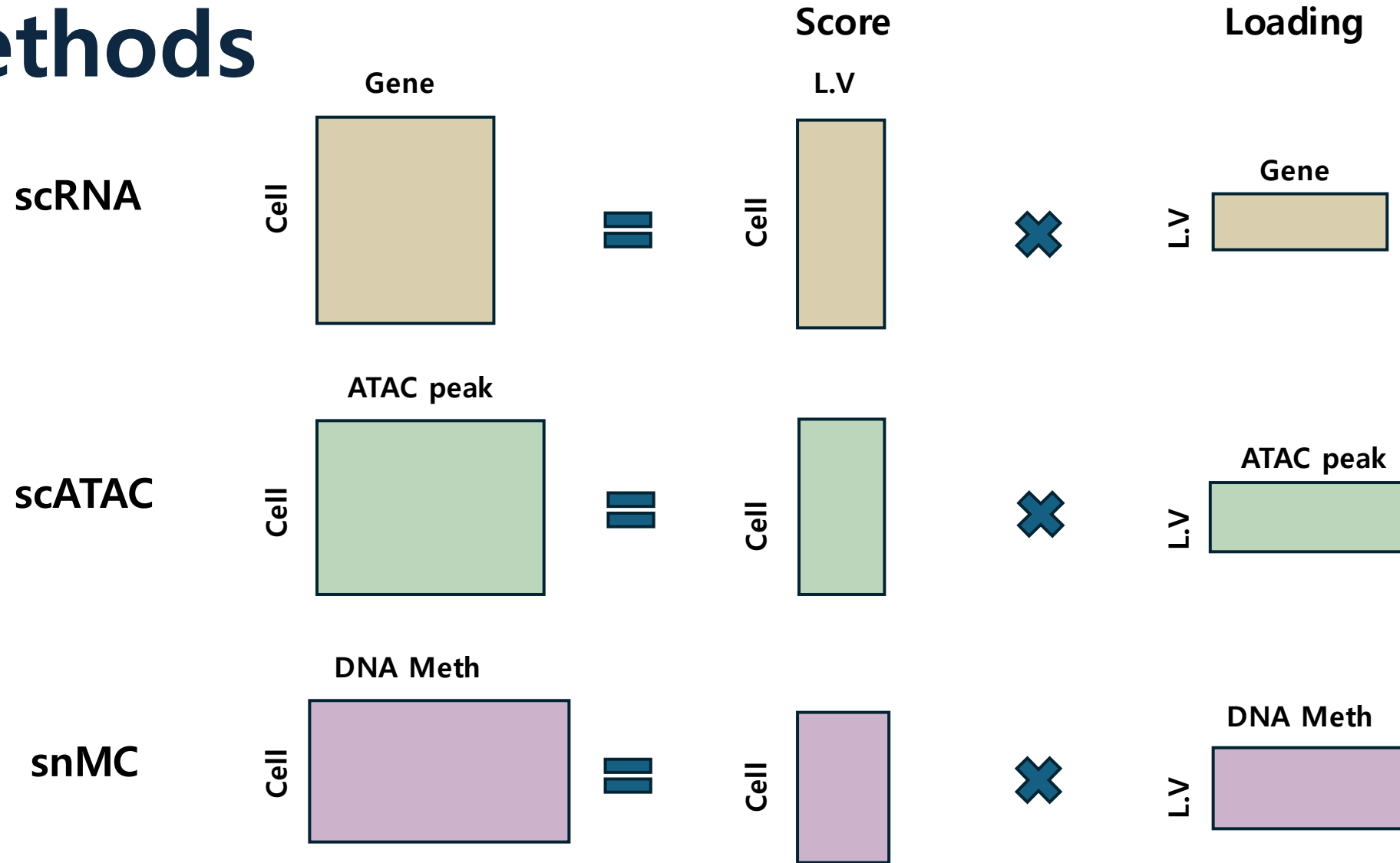
Methods



Decoding Process – Using the *Latent Space* to re-construct *the Raw Data*

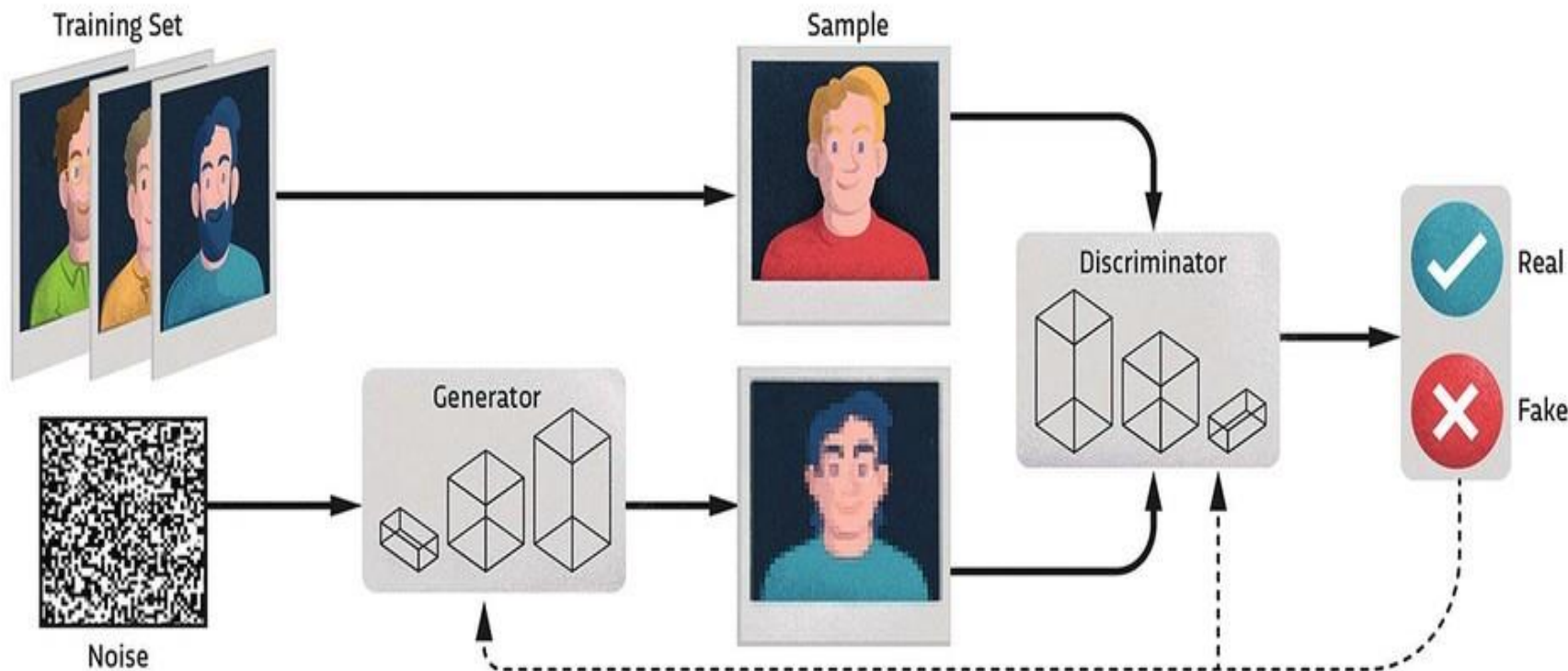
Methods

L.V = Latent Variable



Even though we reconstruct the Latent Variable somehow,
It's not that obvious that the latent variable's are truly integrating the datasets

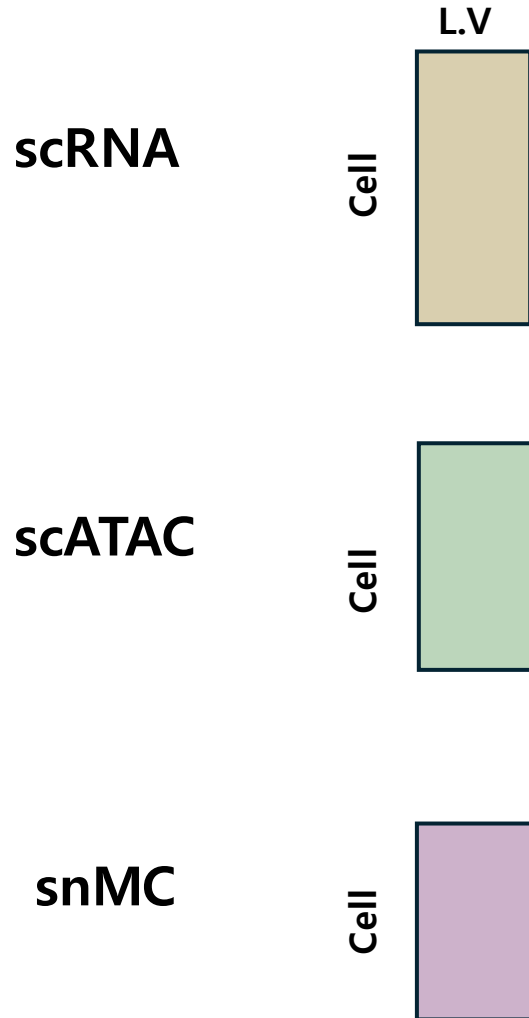
Methods



Generative Adversary Network (GAN)

To build the model making super realistic fake Figure, training the generating model by competing with Discriminating Model

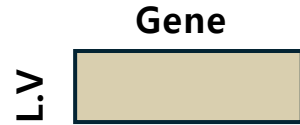
Methods



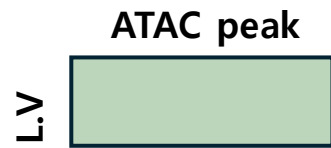
By competing the Encoder with discriminating model,
They can make in-discriminatable latent variable
(integrated variable)

Methods

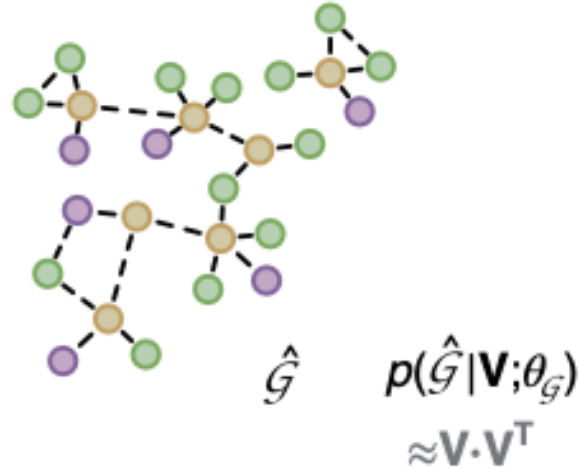
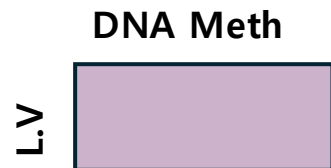
scRNA



scATAC



snMC



And finally, we reconstruct the Graph using the Loading Matrix

Methods

What we have

Feature-Feature Interaction Graph

Multomics Datasets

What we will get

Updated Interaction Graph

Integrated Latent Variable

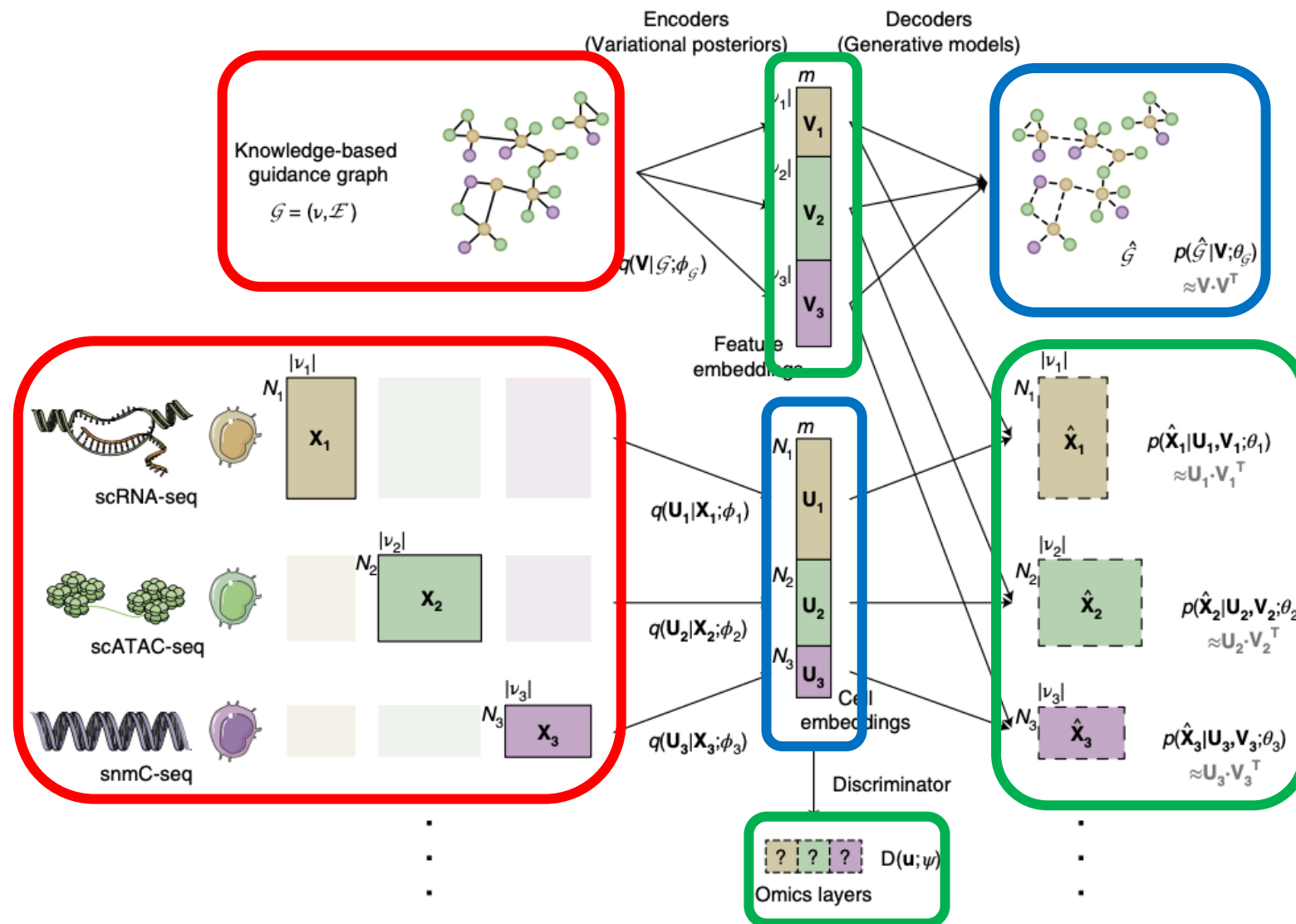
What the model uses

(Things you probably don't need to care about)

Graph embeddings

Reconstructed data

Multi-omics source prediction vectors



Methods

The graph variational posterior $q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}})$ (that is, graph encoder) is modeled as diagonal-covariance normal distributions parameterized by a graph convolutional network⁷⁰:

$$q(\mathbf{u}, \mathbf{V}|\mathbf{x}_k, \mathcal{G};\phi_k, \phi_{\mathcal{G}}) = q(\mathbf{u}|\mathbf{x}_k;\phi_k) \cdot q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}}) \quad (10)$$

$$q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}}) = \prod_{i \in \mathcal{V}} q(\mathbf{v}_i|\mathcal{G};\phi_{\mathcal{G}}) \quad (11)$$

$$q(\mathbf{v}_i|\mathcal{G};\phi_{\mathcal{G}}) = N(\mathbf{v}_i; \text{GCN}_{\mu_i}(\mathcal{G};\phi_{\mathcal{G}}), \text{GCN}_{\sigma_i^2}(\mathcal{G};\phi_{\mathcal{G}})) \quad (12)$$

where $\phi_{\mathcal{G}}$ represents the learnable parameters in the graph convolutional network (GCN) encoder.

The variational data posteriors $q(\mathbf{u}|\mathbf{x}_k;\phi_k)$ (that is, data encoders) are modeled as diagonal-covariance normal distributions parameterized by multilayer perceptron (MLP) neural networks:

$$q(\mathbf{u}|\mathbf{x}_k, \mathbf{V}_k;\phi_k) = N(\mathbf{u}; \text{MLP}_{k,\mu}(\mathbf{x}_k;\phi_k), \text{MLP}_{k,\sigma^2}(\mathbf{x}_k;\phi_k)) \quad (13)$$

$$q(\mathbf{u}|\mathbf{x}_k, \mathbf{V}_k;\phi_k) = N(\mathbf{u}; \text{MLP}_{k,\mu}(\mathbf{x}_k;\phi_k), \text{MLP}_{k,\sigma^2}(\mathbf{x}_k;\phi_k)) \quad (13)$$

where ϕ_k is the set of learnable parameters in the multilayer perceptron encoder of the k th omics layer.

Model fitting can then be performed by maximizing the following evidence lower bound:

$$\sum_{k=1}^K \mathbb{E}_{\mathbf{x}_k \sim p_{\text{data}}(\mathbf{x}_k)} \left[\mathbb{E}_{\mathbf{u} \sim q(\mathbf{u}|\mathbf{x}_k;\phi_k), \mathbf{V} \sim q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}})} \log p(\mathbf{x}_k|\mathbf{u}, \mathbf{V};\theta_k) p(\mathcal{G}|\mathbf{V};\theta_{\mathcal{G}}) \right] - \text{KL}(q(\mathbf{u}|\mathbf{x}_k;\phi_k) q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}}) \parallel p(\mathbf{u}) p(\mathbf{V})) \quad (14)$$

which can be further rearranged into the following form:

$$K \cdot \mathcal{L}_{\mathcal{G}}(\theta_{\mathcal{G}}, \phi_{\mathcal{G}}) + \sum_{k=1}^K \mathcal{L}_{\mathcal{X}_k}(\theta_k, \phi_k, \phi_{\mathcal{G}}) \quad (15)$$

where we have

$$\mathcal{L}_{\mathcal{X}_k}(\theta_k, \phi_k, \phi_{\mathcal{G}}) = \mathbb{E}_{\mathbf{x}_k \sim p_{\text{data}}(\mathbf{x}_k)} \left[\mathbb{E}_{\mathbf{u} \sim q(\mathbf{u}|\mathbf{x}_k;\phi_k), \mathbf{V} \sim q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}})} \log p(\mathbf{x}_k|\mathbf{u}, \mathbf{V};\theta_k) - \text{KL}(q(\mathbf{u}|\mathbf{x}_k;\phi_k) \parallel p(\mathbf{u})) \right] \quad (16)$$

$$\mathcal{L}_{\mathcal{G}}(\theta_{\mathcal{G}}, \phi_{\mathcal{G}}) = \mathbb{E}_{\mathbf{V} \sim q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}})} \log p(\mathcal{G}|\mathbf{V};\theta_{\mathcal{G}}) - \text{KL}(q(\mathbf{V}|\mathcal{G};\phi_{\mathcal{G}}) \parallel p(\mathbf{V})) \quad (17)$$

Below, for convenience, we denote the union of all encoder parameters as $\phi = (\bigcup_{k=1}^K \phi_k) \cup \phi_{\mathcal{G}}$ and the union of all decoder parameters as

$$\theta = (\bigcup_{k=1}^K \theta_k) \cup \theta_{\mathcal{G}}.$$

To ensure the proper alignment of different omics layers, we use the adversarial alignment strategy^{31,71}. A discriminator D with a K -dimensional softmax output is introduced, which predicts the omics layers of cells based on their embeddings $\mathbf{u}$. The discriminator D is trained by minimizing the multiclass classification cross entropy:

$$\mathcal{L}_D(\phi, \psi) = -\frac{1}{K} \sum_{k=1}^K \mathbb{E}_{\mathbf{x}_k \sim p_{\text{data}}(\mathbf{x}_k)} \mathbb{E}_{\mathbf{u} \sim q(\mathbf{u}|\mathbf{x}_k;\phi_k)} \log D_k(\mathbf{u}; \psi) \quad (18)$$

where D_k represents the k th dimension of the discriminator output and ψ is the set of learnable parameters in the discriminator. The data encoders can then be trained in the opposite direction to fool the discriminator, ultimately leading to the alignment of cell embeddings from different omics layers⁷².

The overall training objective of GLUE thus consists of:

$$\min_{\psi} \lambda_D \cdot \mathcal{L}_D(\phi, \psi) \quad (19)$$

$$\max_{\theta, \phi} \lambda_D \cdot \mathcal{L}_D(\phi, \psi) + \lambda_{\mathcal{G}} K \cdot \mathcal{L}_{\mathcal{G}}(\theta_{\mathcal{G}}, \phi_{\mathcal{G}}) + \sum_{k=1}^K \mathcal{L}_{\mathcal{X}_k}(\theta_k, \phi_k, \phi_{\mathcal{G}}) \quad (20)$$

Weighted adversarial alignment. As shown in previous work³¹, canonical adversarial alignment amounts to minimizing a generalized form of Jensen–Shannon divergence among the cell embedding distributions of different omics layers:

$$\frac{1}{K} \sum_{k=1}^K \text{KL} \left(q_k(\mathbf{u}) \parallel \frac{1}{K} \sum_{k=1}^K q_k(\mathbf{u}) \right) \quad (21)$$

where $q_k(\mathbf{u}) = \mathbb{E}_{\mathbf{x}_k \sim p_{\text{data}}(\mathbf{x}_k)} q(\mathbf{u}|\mathbf{x}_k;\phi_k)$ represents the marginal cell embedding distribution of the k th layer. Without other loss terms, equation (21) converges at perfect alignment, that is, when $q_i(\mathbf{u}) = q_j(\mathbf{u})$, $\forall i \neq j$. This can be problematic when cell type compositions differ dramatically across different layers, for example, in the cell atlas integration. To address this issue, we added cell-specific weights $w^{(n)}$ to the discriminator loss in equation (18):

$$\mathcal{L}_D(\phi, \psi) = -\frac{1}{K} \sum_{k=1}^K \frac{1}{W_k} \sum_{n=1}^{N_k} w^{(n)} \cdot \mathbb{E}_{\mathbf{u} \sim q(\mathbf{u}|\mathbf{x}_k^{(n)};\phi_k)} \log D_k(\mathbf{u}; \psi) \quad (22)$$

where the normalizer $W_k = \sum_{n=1}^{N_k} w^{(n)}$. The adversarial alignment still amounts to minimizing equation (21) but with weighted marginal cell embedding distributions

$q_k(\mathbf{u}) = \frac{1}{W_k} \sum_{n=1}^{N_k} w^{(n)} q(\mathbf{u}|\mathbf{x}_k^{(n)};\phi_k)$. By assigning appropriate weights to balance

the cell distributions across different layers, the optimum of $q_i(\mathbf{u}) = q_j(\mathbf{u})$, $\forall i \neq j$ could be much closer to the desired alignment.

$$p(\mathbf{x}_k; \theta_k) = \int p(\mathbf{x}_k | \mathbf{u}; \theta_k) p(\mathbf{u}) d\mathbf{u}$$

$$\mathcal{L}_{\mathcal{X}_k}(\phi_k, \theta_k) = \mathbb{E}_{\mathbf{x}_k \sim p_{\text{data}}(\mathbf{x}_k)} [\mathbb{E}_{\mathbf{u} \sim q(\mathbf{u}|\mathbf{x}_k;\phi_k)} \log p(\mathbf{x}_k | \mathbf{u}; \theta_k) - \text{KL}(q(\mathbf{u}|\mathbf{x}_k;\phi_k) \parallel p(\mathbf{u}))]$$

$$p(\mathbf{x}_k, \mathcal{G}; \theta_k, \theta_{\mathcal{G}}) = \int p(\mathbf{x}_k | \mathbf{u}, \mathbf{V}; \theta_k) p(\mathcal{G} | \mathbf{V}; \theta_{\mathcal{G}}) p(\mathbf{u}) p(\mathbf{V}) d\mathbf{u} d\mathbf{V}$$

$$p(\mathbf{u}) = N(\mathbf{u}; \mathbf{0}, \mathbf{I}_m)$$

$$p(\mathbf{v}_i) = N(\mathbf{v}_i; \mathbf{0}, \mathbf{I}_m), p(\mathbf{V}) = \prod_{i \in \mathcal{V}} p(\mathbf{v}_i)$$

$$\log p(\mathcal{G} | \mathbf{V}; \theta_{\mathcal{G}}) = \mathbb{E}_{i,j \sim p(i,j; w_{ij})}$$

$$[\log \sigma(s_{ij} \mathbf{v}_i^\top \mathbf{v}_j) + \mathbb{E}_{j' \sim p_{\text{ns}}(j'|i)} \log(1 - \sigma(s_{ij} \mathbf{v}_i^\top \mathbf{v}_{j'}))]$$

$$p(\mathbf{x}_k | \mathbf{u}, \mathbf{V}; \theta_k) = \prod_{i \in \mathcal{V}_k} \text{NB}(\mathbf{x}_{ki}; \boldsymbol{\mu}_i, \boldsymbol{\theta}_i)$$

$$\text{NB}(\mathbf{x}_{ki}; \boldsymbol{\mu}_i, \boldsymbol{\theta}_i) = \frac{\Gamma(\mathbf{x}_{ki} + \boldsymbol{\theta}_i)}{\Gamma(\boldsymbol{\theta}_i) \Gamma(\mathbf{x}_{ki} + 1)} \left(\frac{\boldsymbol{\mu}_i}{\boldsymbol{\theta}_i + \boldsymbol{\mu}_i} \right)^{\mathbf{x}_{ki}} \left(\frac{\boldsymbol{\theta}_i}{\boldsymbol{\theta}_i + \boldsymbol{\mu}_i} \right)^{\boldsymbol{\theta}_i}$$

$$\boldsymbol{\mu}_i = \text{Softmax}_i(\boldsymbol{\alpha} \odot \mathbf{V}_k^\top \mathbf{u} + \boldsymbol{\beta}) \cdot \sum_{j \in \mathcal{V}_k} \mathbf{x}_{kj}$$

Application

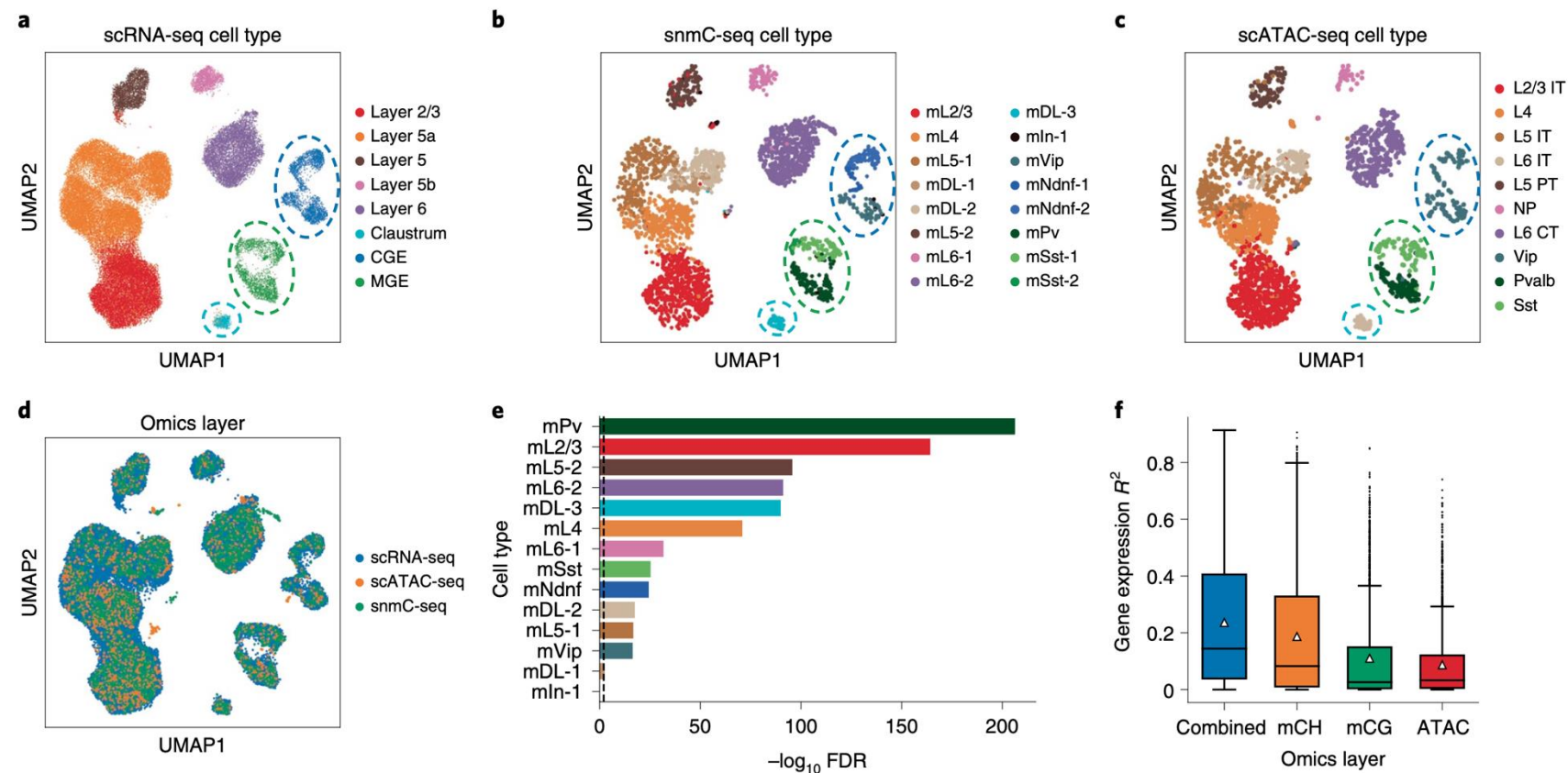
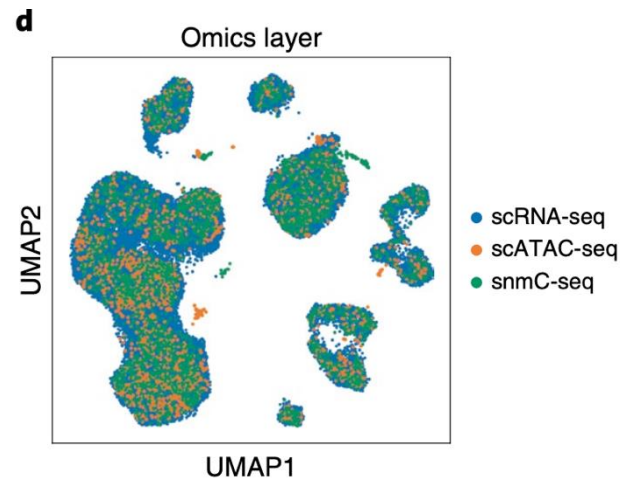
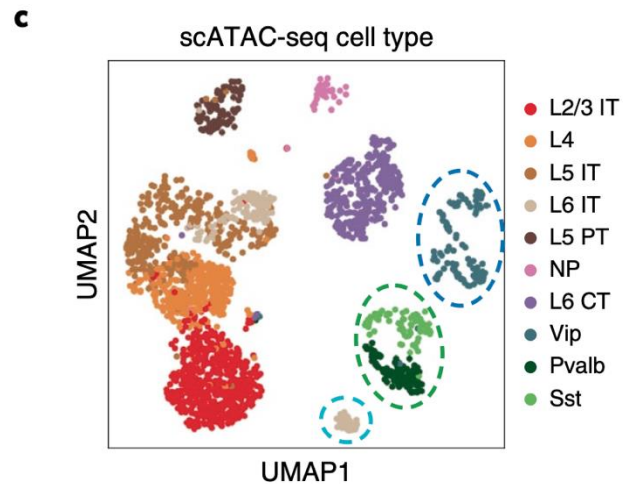
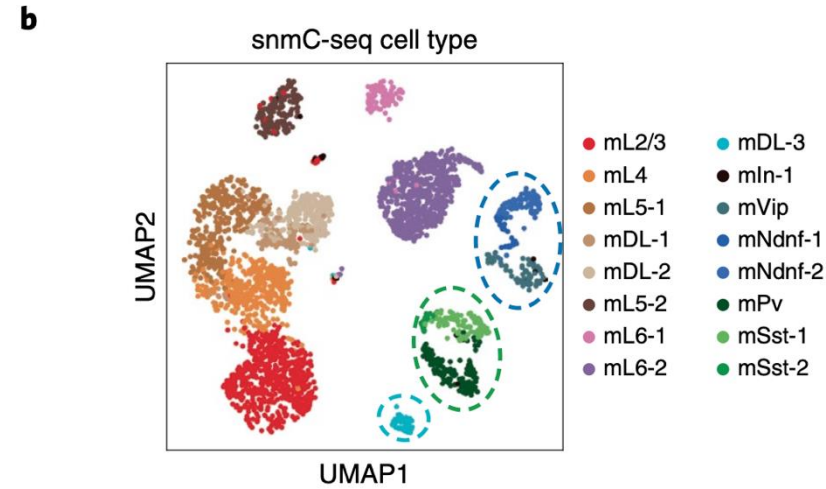
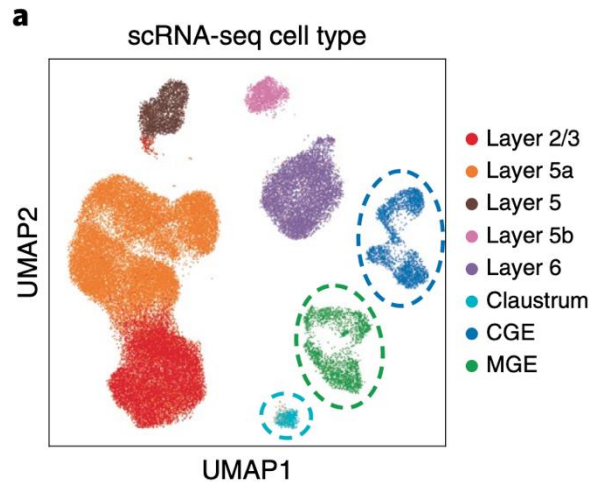


Figure3

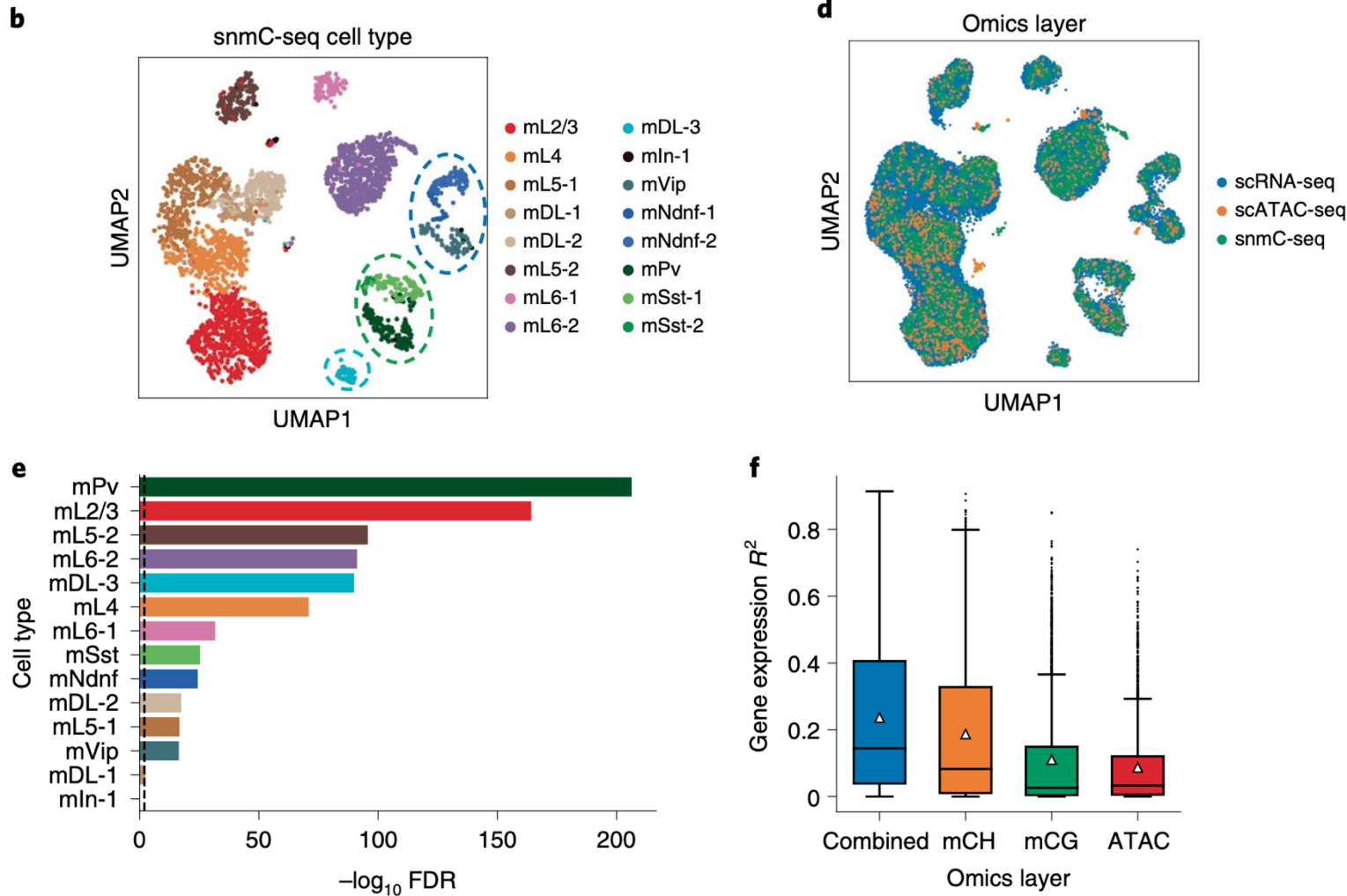
Application



Three-Omics Integrated UMAP

- a. scRNA Seq
- b. snmC-seq (DNA Methylation)
- c. scATAC-seq
- d. Multi-Omics

Application



Now, we have new cell-type labels for all the cells using Nearest-Neighborhood.

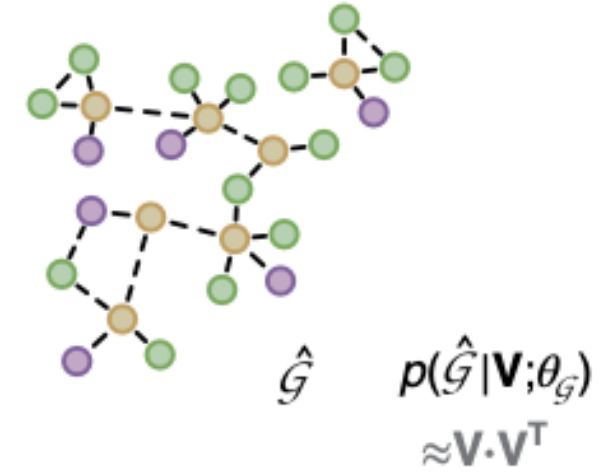
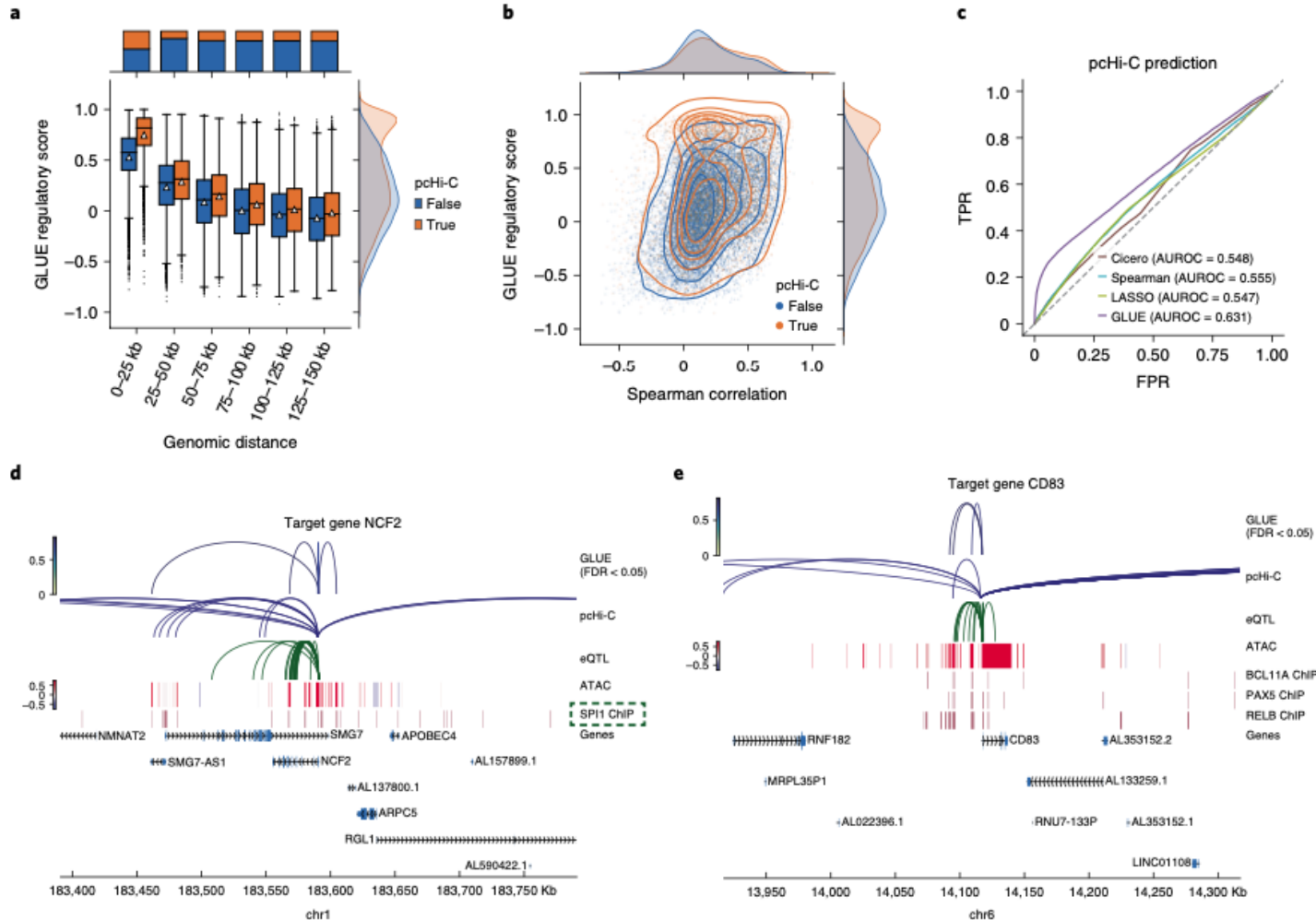
E : marker gene overlapping Test For a specific celltype

1. Converting the features into Gene
2. Getting Marker gene per omics layer
3. Using Fisher test to check the overlapped ratio

F : Regression Test

1. Make 200 subcluster using the integrated latent variable
2. Summarising the Gene Expression, ATAC signal, Methylation signal.
3. Regression 2.7k HVG with other features

Application



Getting the *peak-gene* regulatory scores using the graph

(Cosine Similarities of feature loading)

Figure4

Application

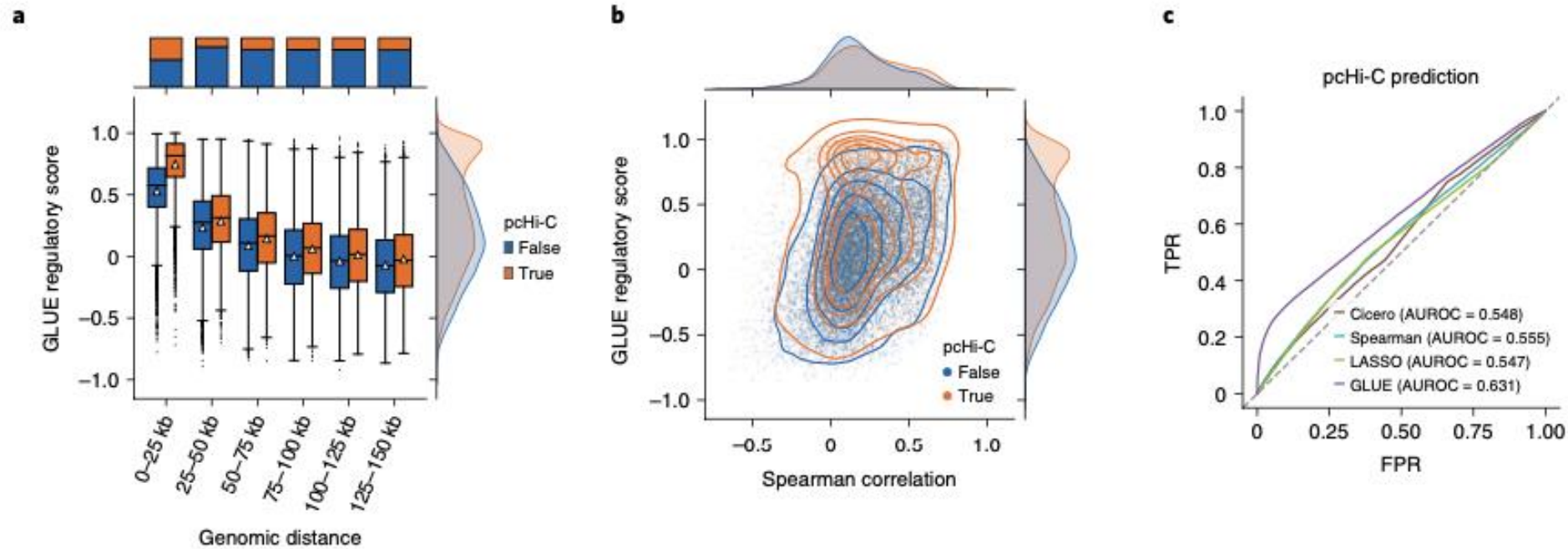
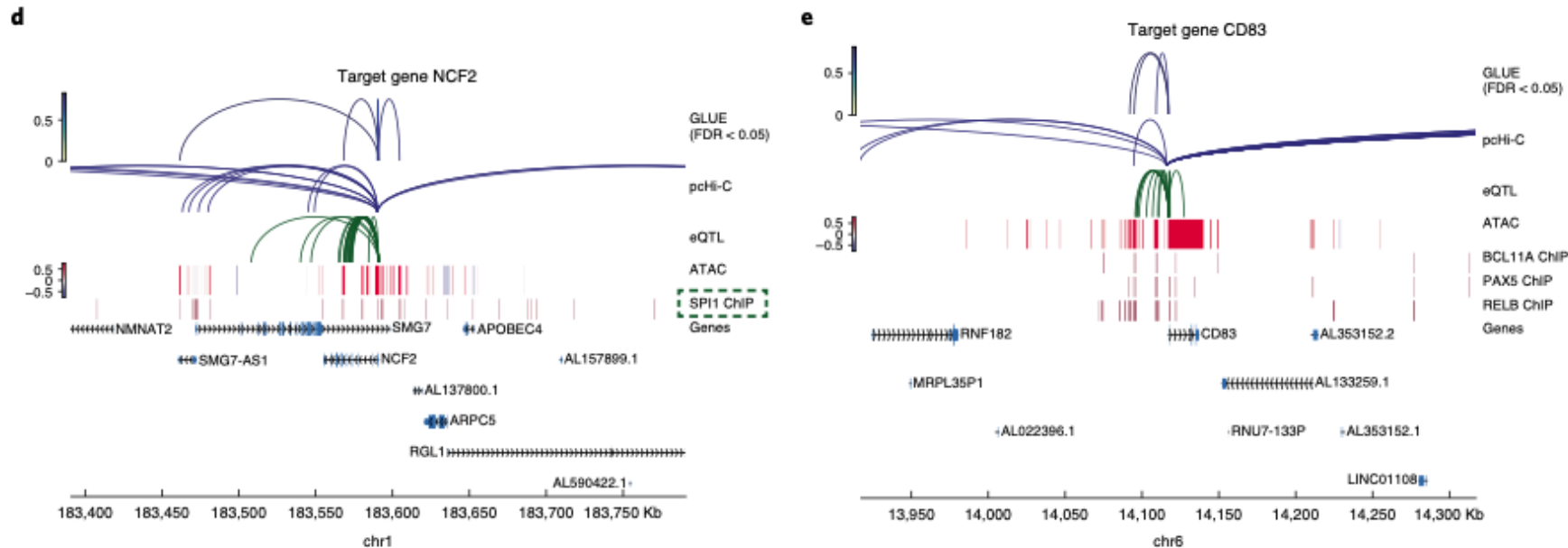


Fig A. GLUE regulatory score correlated with Genomic Distance and higher at pcHi-C True gene-peak pairs

Fig B. GLUE regulatory score correlated with empirical peak-gene correlation

Fig C. GLUE regulatory score predict better the pcHi-C label than others

Application



Case study of GLUE score

- d.** *NCF2*, a well-known immune-related gene,
GLUE identified three regulatory peaks, all of which correspond to *SPI1* binding sites.
- e.** *CD83*, a gene associated with both B cells and monocytes,
GLUE detected three regulatory peaks that correspond to binding sites of *BCL11A*, *PAX5*, and *RELB*.
BCL11A is a B-cell lineage factor, *PAX5* is a B-cell master regulator, and *RELB* belongs to the NF-κB family

Application

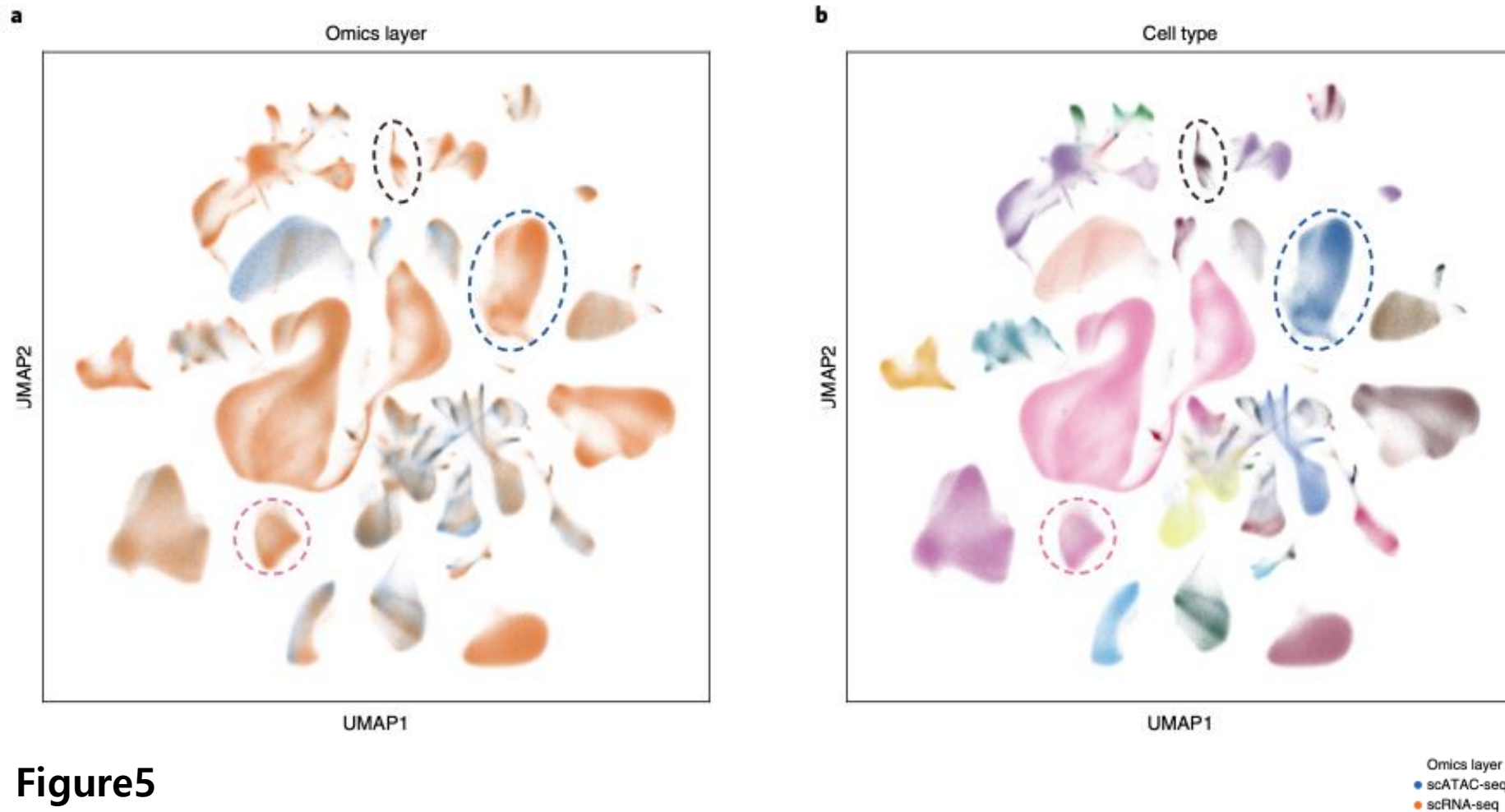


Figure5

- GLUE can handle millions cell integration (Scalability)

Benchmarking Comparison

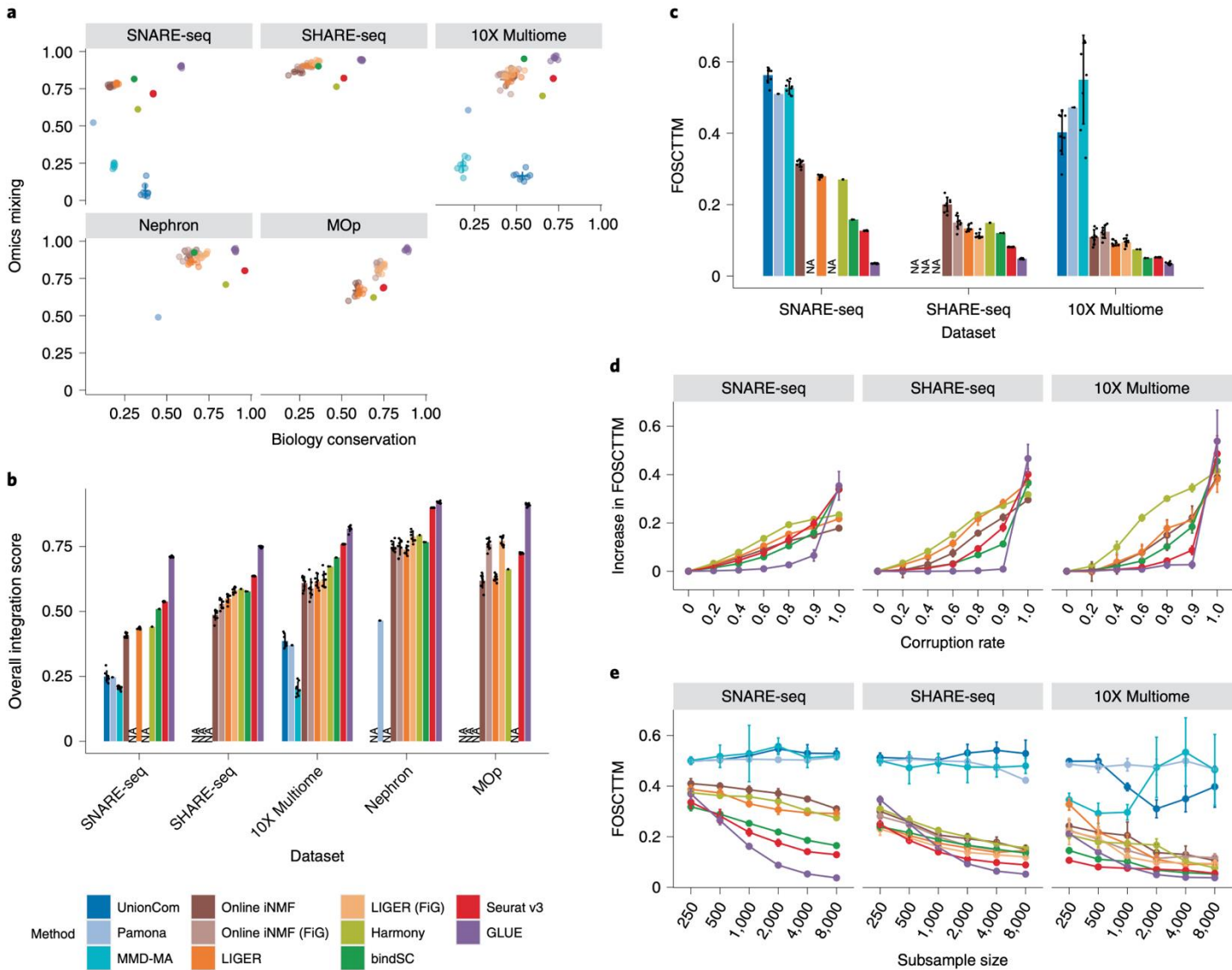
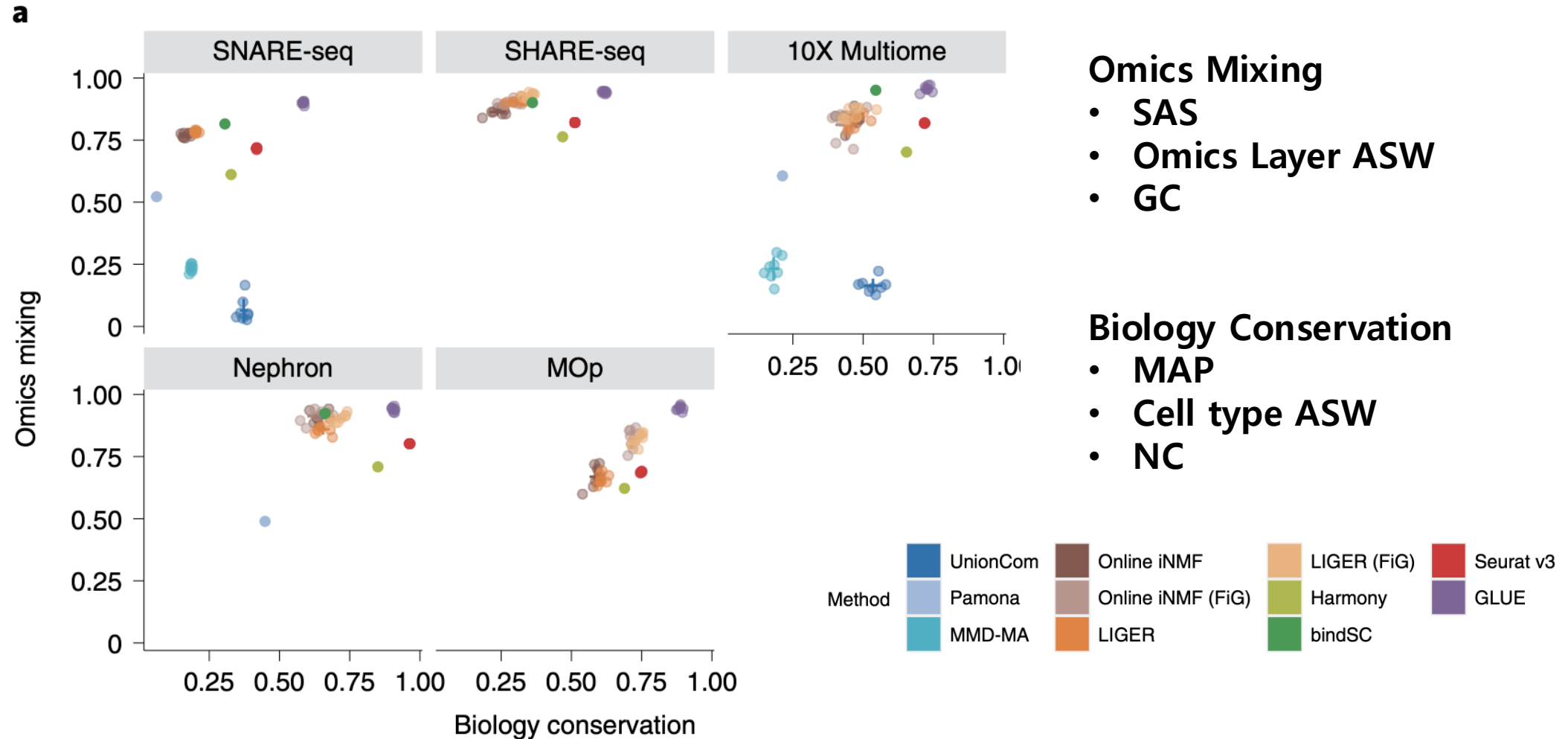


Figure2

Comparison

	List
Data Sets	SNARE-seq, SHARE-seq, 10X Multiome, Nephron, MOp
Methods	GLUE , UnionCom, Online iNMF, LIGER, Seurat v3, Pamona, MMD-MA, bindSC, Harmony
Scores	<i>Omics Mixing, Biological Conservation, Overall Integration Score, FOSCTTM</i>

Comparison



Comparison

Omics Mixing : how well the different omics layers are mixed in the integrated embedding.

SAS (Same-modality Neighbors Score) : 1-adjusted ratio of K-nearest cells coming at the same omics

$$\text{SAS} = 1 - \frac{\bar{x} - \frac{K}{N}}{K - \frac{K}{N}}$$

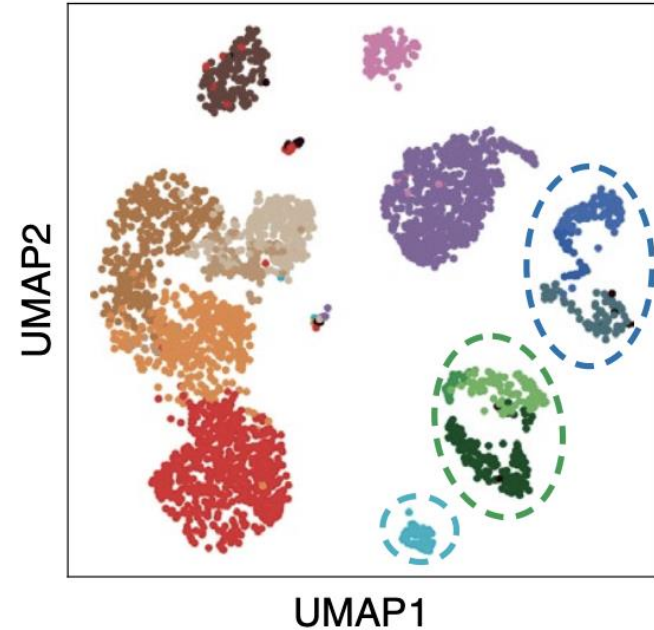
Omics Layer ASW : a measure that whether nearby cells are assigned the same omics

$$\text{omics layer ASW} = \frac{1}{M} \sum_{j=1}^M \text{omics layer ASW}_j$$

$$\text{omics layer ASW}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} 1 - \left| s_{\text{omics layer}}^{(i)} \right|$$

GC (Graph Connectivity) : a metric that evaluates how a celltype connected well (not separated)

$$\text{GC} = \frac{1}{M} \sum_{j=1}^M \frac{|\text{LCC}_j|}{N_j}$$



Comparison

Biology Conservation :how the embedding layer preserve the biological trait

- **MAP** : cells that are close to each other in the integrated latent space tend to share the same cell type.

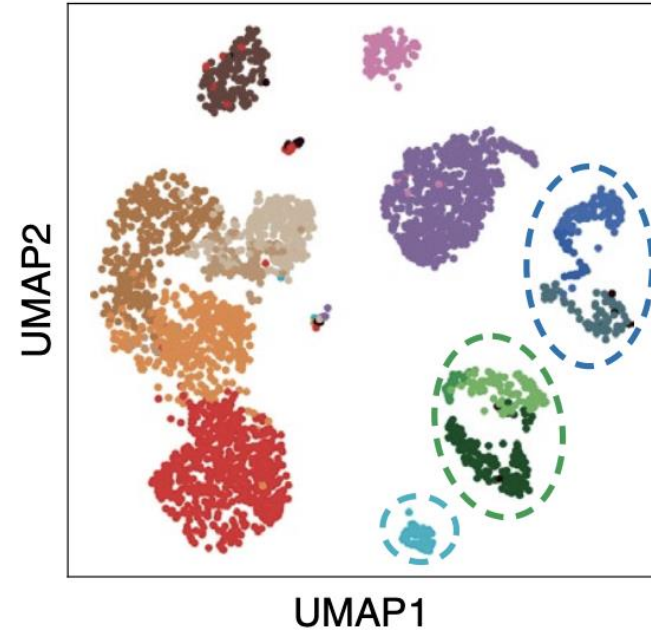
$$\text{MAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}^{(i)} \quad \text{AP}^{(i)} = \begin{cases} \frac{\sum_{k=1}^K 1_{y^{(i)}=y_k} \cdot \frac{\sum_{j=1}^k 1_{y^{(i)}=y_j}}{k}}{\sum_{k=1}^K 1_{y^{(i)}=y_k}}, & \text{if } \sum_{k=1}^K 1_{y^{(i)}=y_k} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- **Cell type ASW**: how well the integrated representation maintains clear separation among biological cell types.

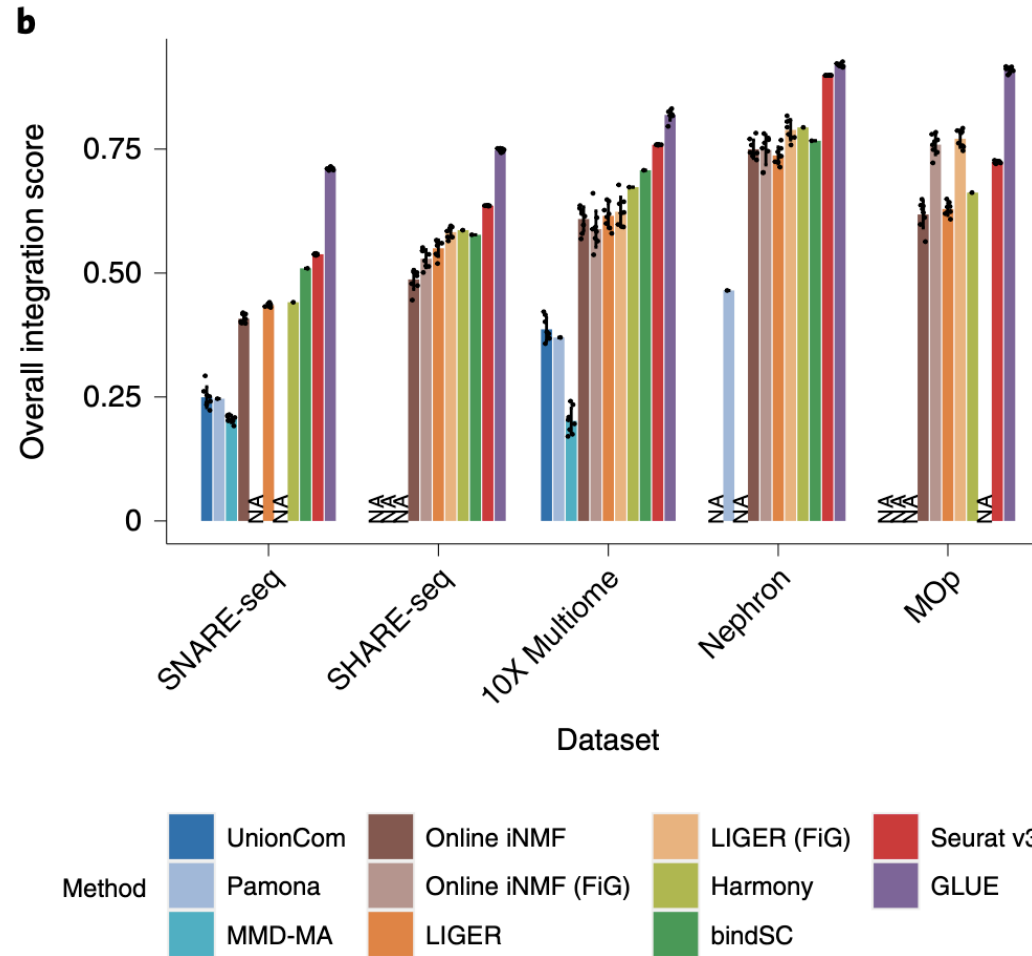
$$\text{cell type ASW} = \frac{1}{2} \left(\frac{\sum_{i=1}^N s_{\text{cell type}}^{(i)}}{N} + 1 \right)$$

- **NC (Neighbor consistency)** : a metric that evaluates whether the nearby cells are similar at single omics layer and integrated latent variables

$$\text{NC} = \frac{1}{N} \sum_{i=1}^N \frac{|\text{NNS}^{(i)} \cap \text{NNI}^{(i)}|}{|\text{NNS}^{(i)} \cup \text{NNI}^{(i)}|}$$

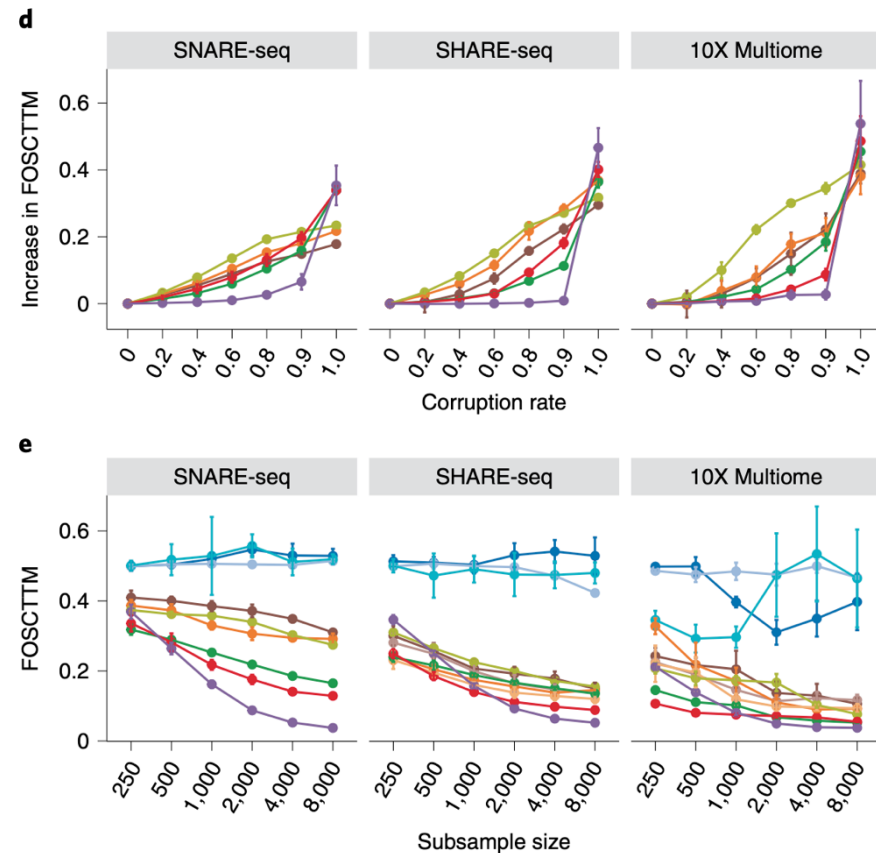
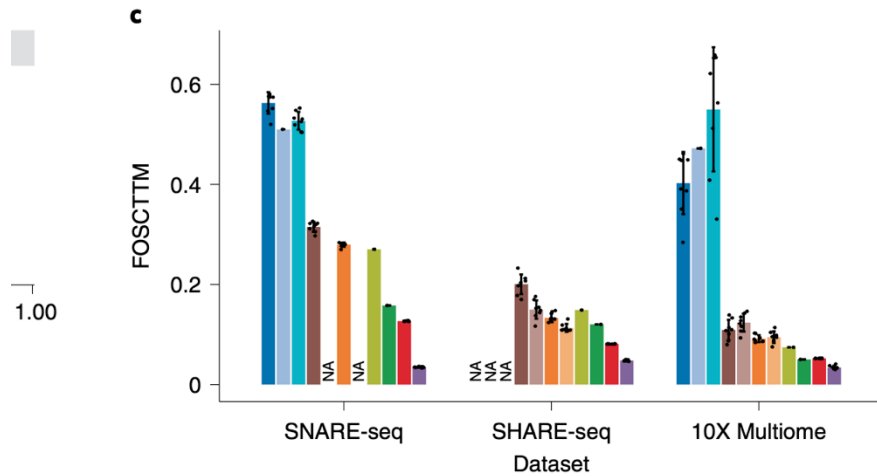


Comparison



Overall Integration Score
= 0.4 Omics Mixing +
0.6 Biology Conservation

Comparison



FOSCTTM : Fraction Of Samples Closer Than The True Match

$$\text{FOSCTTM} = \frac{1}{2N} \left(\sum_{i=1}^N \frac{n_1^{(i)}}{N} + \sum_{i=1}^N \frac{n_2^{(i)}}{N} \right)$$

$$n_1^{(i)} = |\{j | d(\mathbf{x}_j, \mathbf{y}_i) < d(\mathbf{x}_i, \mathbf{y}_i)\}|$$

$$n_2^{(i)} = |\{j | d(\mathbf{x}_i, \mathbf{y}_j) < d(\mathbf{x}_i, \mathbf{y}_i)\}|$$

Summary

GLUE can integrate the multiomics dataset even if all the cells are separated

They can make **integrated latent variables** containing all the multiomics information.

They can also make the adjusted **Feature-wise interaction graph** in a bayesian sense

They can preserve the **biological background** (Cell type) and **mixed the omics** information.

They used *Variational AutoEncoder (VAE)* , *Graph Regulization*, *Factorization Model* and *Generative Adversarial Network (GAN)*

Why I hate This!

Category	Statistical Thinking	Deep Learning Thinking
Purpose	Explain mechanisms; understand <i>why</i> things happen	Optimize performance; <i>as long as it works</i>
Structure	Clear, simple, interpretable models	Highly complex architectures (VAE + GAN + GCN + FM, etc.)
How Performance Improves	Theoretical refinement, better assumptions	More layers, more regularizers, more training tricks

Thanks!