Multimodal Deep Learning for Single-Cell Analysis

Denis Grachev
TUM
justwantpost@gmail.com

Abstract

Single-cell analysis has revolutionized our understanding of cellular heterogeneity, providing a high-resolution view of molecular features (e.g., transcriptomics, proteomics) in individual cells. However, analyzing multiple modalities simultaneously can be complex due to high dimensionality, sparsity, and batch effects. In this report, we apply multimodal deep learning to single-cell data integration, focusing on models like Multigrate, TOTALVI, and MOFA. Each model produces low-dimensional embeddings for every cell, which are then converted into a cell-cell graph. We evaluate the quality of these embeddings via graph-based metrics such as ARI, NMI, Silhouette Score, Graph Connectivity, and Isolated Labels ASW. Our findings highlight the promise of deep learning in capturing the interrelationships among diverse molecular measurements.

1 Introduction

Single-cell technologies capture gene expression, chromatin accessibility, or protein abundance at single-cell resolution, providing unprecedented insight into cellular heterogeneity. Yet these data come from different modalities, so integrative methods are needed to combine genomics, transcriptomics, proteomics, and other assays. Deep learning approaches offer flexible frameworks for learning shared representations of multimodal data, while also helping to mitigate common challenges like batch effects.

In this report, we examine a benchmark multimodal single-cell dataset, describe a preprocessing pipeline, and present three representative models—Multigrate, TOTALVI, and MOFA—for integrated analysis. After each model learns embeddings for the cells, we build a cell-cell graph to compute a variety of metrics (e.g., ARI, NMI). We compare the models' performance and discuss the trade-offs involved in single-cell data integration.

2 Data

The dataset used in this study comprises approximately 120,000 human bone marrow cells collected from 10 donors. Measurements come from two main assay combinations: (1) nuclear gene expression (GEX) plus ATAC, and (2) cellular GEX plus antibody-derived tags (ADT). Nested batch effects are introduced by sampling each donor at four different laboratory sites, thus representing both site- and donor-level variability. Each cell contains:

- **ATAC**: Accessibility of 119,254 genomic regions.
- **GEX**: Expression levels of 15,189 genes.
- **ADT**: Abundance of 134 surface proteins.

The dataset covers multiple developmental lineages, including immune populations and erythrocyte precursors. Data preprocessing and annotation were performed using standardized pipelines with

quality control, normalization, clustering, and trajectory inference. Ground-truth cell identities were assigned through expert curation and marker-based annotation.

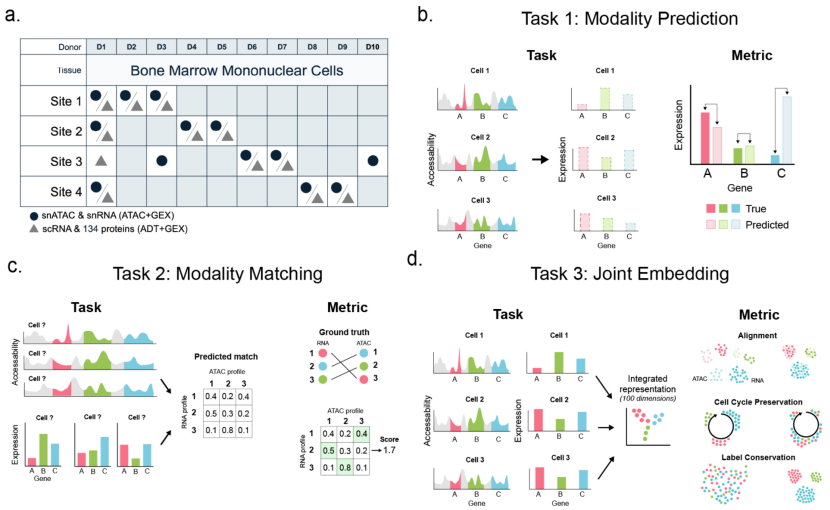


Figure 1: Overview of a multimodal single-cell reference dataset and the three principal tasks: (a) multi-site BMMC data with known annotations, (b-d) modality prediction, matching, and joint representation [5].

3 Data Preprocessing

Before training the models, we perform essential preprocessing steps to ensure consistent and comparable inputs across different modalities:

1. **Subsampling**: For demonstration or memory considerations, the data may be subsampled (e.g., to 20,000 cells) to reduce computational load. Subsampling allows rapid prototyping without qualitatively changing core results, especially when dealing with very large datasets.
2. **Splitting RNA and ADT**: The GEX features are extracted as an `AnnData` object (`rna`), while the ADT features are kept in a separate `AnnData` (`adt`). This separation is convenient for multi-omics pipelines (e.g., MuData structures).
3. **Layer Allocation**: For RNA, we copy `layers["counts"]` into `.X` so that normalization is based on raw counts. This ensures that the standard single-cell protocols (e.g., `sc.pp.normalize_total`, `sc.pp.log1p`) are applied consistently.
4. **Normalization and Log-Transform** (RNA): Total count normalization (target sum $= 10^4$) followed by a $\log(1 + \cdot)$ transform to stabilize variance and reduce skew.
5. **Highly Variable Gene (HVG) Selection**: We identify the top 2,000 HVGs, considering `Site` as a batch variable to adjust for batch-specific differences. Retaining only HVGs can improve downstream model performance by focusing on the most informative transcripts.
6. **CLR Transform (ADT)**: For the ADT features, a centered log-ratio (`clr`) normalization is often recommended, as it helps control for differences in overall protein capture efficiency. The `muon.prot.pp.clr` function performs this step, storing the result in `.X`.
7. **MuData Assembly**: We can merge the preprocessed `rna` and `adt` `AnnData` objects into a single `MuData` container (`mdata`), preserving separate modalities but allowing integrated analysis.

After these steps, the dataset is better suited for models that integrate multiple omics layers, each modality having been normalized and transformed appropriately for the model's assumptions.

4 Batch Effect Correction

Single-cell data can exhibit significant batch effects due to technical or procedural variation. If not corrected, these effects may obscure true biological signals. Common strategies include:

- **Linear Methods (e.g., ComBat, limma)**: Model batch as an additive or multiplicative term in a linear framework.
- **Nonlinear Methods (e.g., MNN, CCA)**: Align data through mutual nearest neighbors or canonical correlations to capture complex batch effects.
- **Deep Learning Methods (e.g., VAEs)**: Learn latent spaces where batch variation is separated from biological signals.

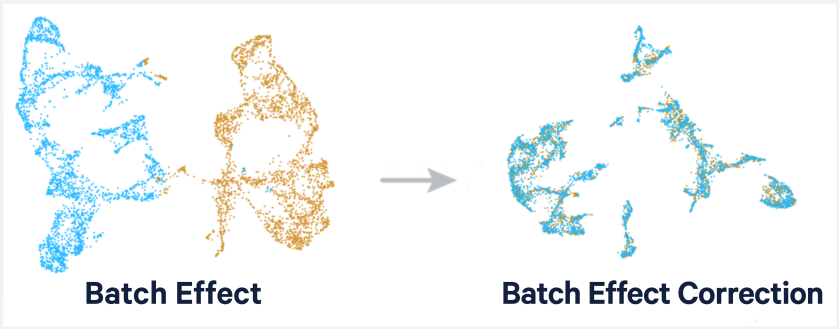


Figure 2: Diagram of batch effect (left) and corrected data (right) [1].

5 Models

We apply three models designed for multimodal single-cell data. Each model outputs a latent embedding for each cell. We then construct a *cell-cell graph* (e.g., using k-nearest neighbors) based on these embeddings, which forms the basis of the evaluation metrics.

5.1 Multigrate Architecture

Multigrate is a generative multi-view neural network capable of handling missing modalities and batch covariates [4]. It adopts a Product of Experts (PoE) approach to fuse the latent representations of different data types.

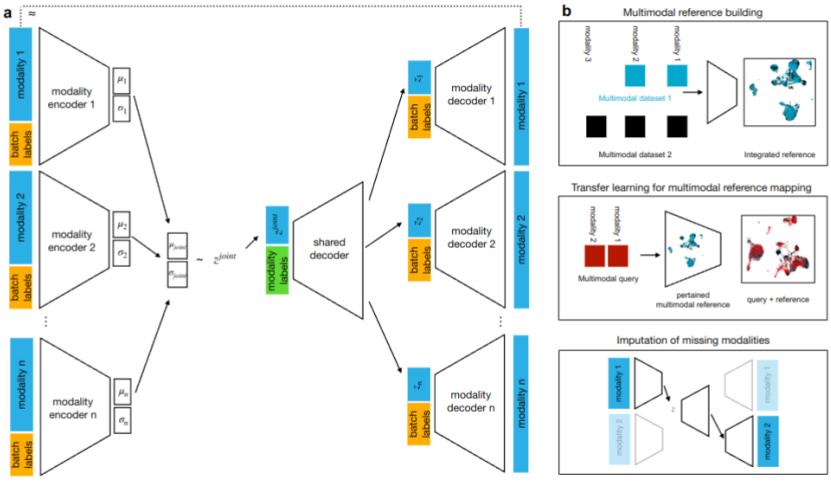


Figure 3: Multigrate architecture [4].

5.1.1 Generative Model

Let $X = \{X_i\}_{i=1}^n$ be observations from n modalities, and $S = \{S_i\}_{i=1}^n$ be study labels encoding batch/site. When a modality i is absent, the corresponding X_i is empty.

Using PoE, the approximate posterior is:

$$q_\phi(Z_{\text{joint}} \mid X, S) = \prod_{i=1}^{n} q_{\phi_i}(Z_i \mid X_i, S_i), \tag{1}$$

where $q_{\phi_i}(Z_i \mid X_i, S_i) = 1$ if modality i is missing, otherwise it is a Gaussian $\mathcal{N}(\mu_i, \sigma_i)$. The joint distribution is computed via:

$$\mu_{\text{joint}} = \left(\mu_0\sigma_0^{-1} + \sum_{i=1}^{n} m_i\mu_i\sigma_i^{-1}\right)\left(\sigma_0^{-1} + \sum_{i=1}^{n} m_i\sigma_i^{-1}\right)^{-1}, \tag{2}$$

$$\sigma_{\text{joint}} = \left(\sigma_0^{-1} + \sum_{i=1}^{n} m_i\sigma_i^{-1}\right)^{-1}, \tag{3}$$

where $m_i = 1$ if modality i is present, and μ_0, σ_0 are prior parameters.

5.1.2 Training Objective

The model minimizes:

$$L_{AE} = \alpha\, \mathbb{E}_{q_\phi(Z_{\text{joint}}|X_i,S_i)}\big[\log p_\theta(X_i \mid Z_{\text{joint}}, S_i)\big] - \eta\, D_{\text{KL}}\big(q_\phi(Z_{\text{joint}} \mid X_i, S_i) \,\|\, p_\theta(Z_{\text{joint}} \mid S_i)\big), \tag{4}$$

summed over all modalities. An MMD term aligns latent spaces across batches:

$$L_{\text{MMD}}(Z_{\text{joint}_i}, Z_{\text{joint}_j}) = \sum_{i<j} k(Z_{\text{joint}_i},\, Z_{\text{joint}_j}, \gamma),$$

where k is a multi-scale RBF kernel. The total loss is:

$$L_{\text{Multigrate}} = \sum_{i=1}^{n} L_{AE}(X_i, S_i) + \beta \sum_{i<j} L_{\text{MMD}}(Z_{\text{joint}_i}, Z_{\text{joint}_j}). \tag{5}$$

5.2 TOTALVI Architecture

TOTALVI is a deep generative model for jointly analyzing RNA and protein data [3]. It uses a VAE-like framework that accounts for batch effects and protein background noise.

5.2.1 Generative Model

Each cell n has:

$$x_n \in \mathbb{R}^G \quad \text{(RNA counts)}, \quad y_n \in \mathbb{R}^T \quad \text{(protein counts)},$$

and a one-hot batch vector s_n. The latent variable z_n follows a LogisticNormal prior, while ℓ_n accounts for RNA library size.

RNA counts follow a Gamma-Poisson mixture:

$$x_{ng} \mid z_n, \ell_n \sim \text{NegativeBinomial}(\ell_n\, \rho_{ng},\, \theta_g),$$

and proteins use a mixture distribution that handles background:

$$y_{nt} \mid z_n \sim \text{Poisson}(r_{nt}),$$

where r_{nt} depends on latent variables for protein expression and background.

5.2.2 Inference Model

Approximate posteriors:

$$q_\phi(z_n \mid x_n, y_n, s_n), \quad q_\phi(\ell_n \mid x_n, s_n), \quad q_\phi(\pi_{nt} \mid y_n),$$

are parameterized by neural networks.

5.2.3 Optimization

TOTALVI maximizes the ELBO:

$$\mathcal{L}_{\text{TOTALVI}} = \mathbb{E}_{q_\phi(z_n|x_n,y_n,s_n)}\big[\log p_\theta(x_n, y_n \mid z_n, s_n)\big] - D_{\text{KL}}\big(q_\phi(z_n \mid x_n, y_n, s_n) \,\|\, p(z_n)\big).$$

This yields latent embeddings correcting for batch and separating background signals from true protein counts.

5.3 MOFA Architecture

Multi-Omics Factor Analysis (MOFA) factorizes multi-view data into shared latent factors [2]. Each modality m is expressed as:

$$Y_m = ZW_m^T + \epsilon_m,$$

where $Z \in \mathbb{R}^{N\times K}$ is a latent factor matrix, $W_m \in \mathbb{R}^{D_m\times K}$ are modality-specific loadings, and ϵ_m is noise. Different likelihoods (Gaussian, Poisson, Bernoulli) accommodate diverse data types.

An ARD prior encourages sparsity:

$$W_{mkd} = s_{mkd}\,\tilde{W}_{mkd}, \quad s_{mkd} \sim \text{Bernoulli}(h_{mk}),$$

and inference uses a variational approach:

$$q(Z, W) \approx p(Z, W \mid Y),$$

maximizing the ELBO

$$\mathcal{L}_{\text{MOFA}} = \mathbb{E}_{q(Z,W)}[\log p(Y \mid Z, W)] - D_{\text{KL}}(q(Z, W) \,\|\, p(Z, W)).$$

6 Metrics

Once a model provides a latent embedding for each cell, we construct a graph (e.g., k-nearest neighbors) and evaluate the following:

6.1 Adjusted Rand Index (ARI)

Compares cluster assignments (from either known labels or graph-based partitions) to the embedding-based clustering, adjusted for random chance.

6.2 Normalized Mutual Information (NMI)

Measures information overlap between inferred clusters and known labels, normalized to lie between 0 and 1.

6.3 Silhouette Score

A distance-based measure of cluster separation. For each cell i, let $a(i)$ be the average distance to neighbors in the same cluster, and $b(i)$ the average distance to the nearest different cluster:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}.$$

6.4 Graph Connectivity

Examines the connectivity of the cell-cell graph itself, for instance via average shortest path length or the fraction of cells in the main connected component.

6.5 Isolated Labels ASW

Focuses on the silhouette width of rare or isolated cell types, capturing whether minority populations are well-separated in the embedding.

7 Results

We trained the three models (MOFA, TOTALVI, and Multigrate) on the processed dataset, obtained latent embeddings, and built a k-nearest-neighbors graph in each embedding space. Table 1 compares ARI, NMI, Silhouette Score, Graph Connectivity, and Isolated Labels ASW across models.

Metric	MOFA	TOTALVI	Multigrate
ARI	0.50	0.69	**0.73**
NMI	0.64	**0.79**	0.77
Silhouette Score	0.54	0.55	**0.59**
Graph Connectivity	0.79	**0.91**	0.88
Isolated Labels ASW	**0.63**	0.59	0.60

Table 1: Performance of each model, computed on a cell-cell graph built from the learned embeddings.

TOTALVI yields the highest NMI and Graph Connectivity, reflecting well-preserved local and global structure. Multigrate slightly outperforms the others on ARI, indicating strong alignment with known labels. MOFA shows competitive performance overall, especially for isolated cell populations (higher ASW).

8 Conclusion

In this report, we explored how multimodal deep learning approaches can integrate single-cell assays, ranging from gene expression to surface protein abundance. We introduced a benchmark multimodal dataset, performed a standard preprocessing pipeline (subsampling, normalization, HVG selection, CLR transform), and compared three models: Multigrate, TOTALVI, and MOFA. Each model provides a latent embedding that we transform into a cell-cell graph for computing metrics such as ARI, NMI, Silhouette Score, Graph Connectivity, and Isolated Labels ASW.

Our results illustrate how different model architectures excel in distinct metrics. For instance, TOTALVI appears to maintain high connectivity and robust gene-protein integration, while Multigrate performs especially well in aligning latent clusters to known cell annotations (high ARI). MOFA, in turn, excels at resolving certain smaller cell subsets. Overall, multimodal deep learning continues to advance our ability to decode complex cell heterogeneity by leveraging multiple types of molecular measurements within a unified framework.

References

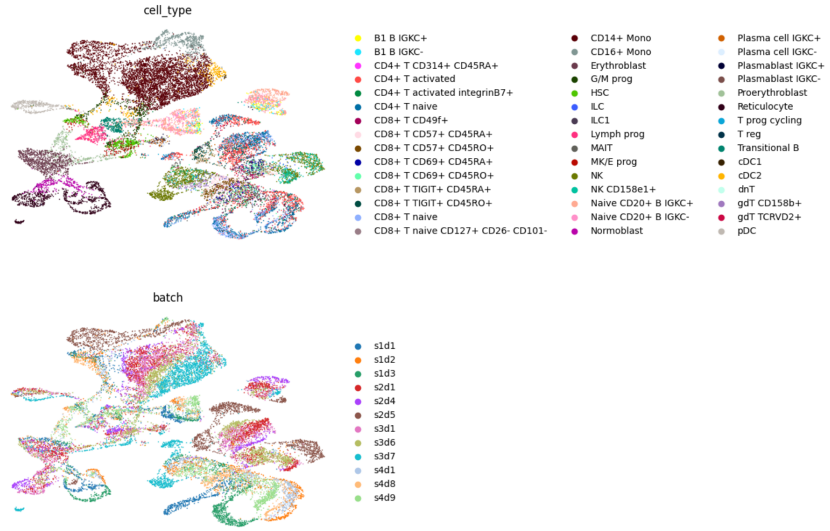
[1] 10x Genomics. Introduction to batch effect correction, 2024. Accessed: 2025-02-09.

[2] Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14(6):e8124, 2018.

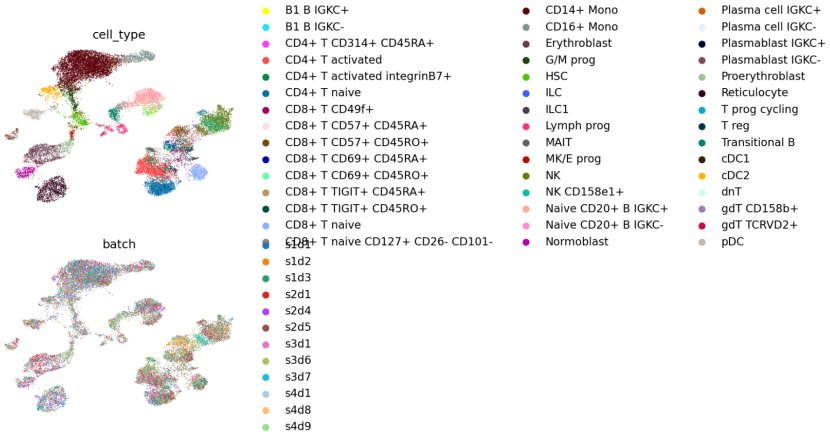
[3] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature methods*, 18(3):272–282, 2021.

[4] Mohammad Lotfollahi, Anastasia Litinetskaya, and Fabian J Theis. Multigrate: single-cell multi-omic data integration. *BioRxiv*, pages 2022–03, 2022.

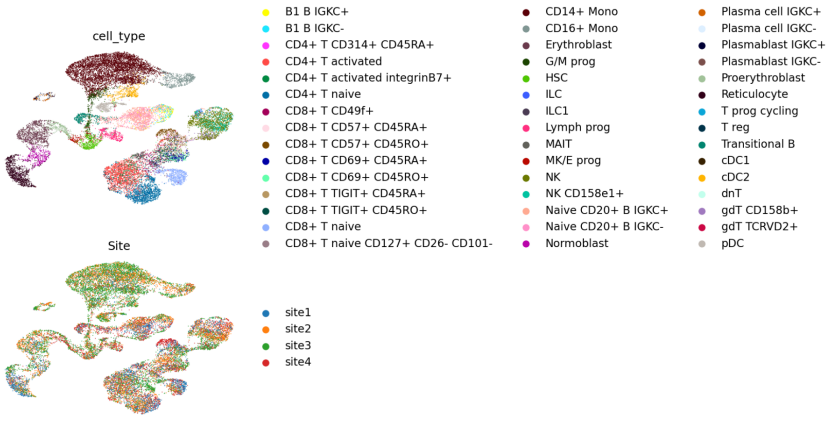
[5] MD Luecken et al. A sandbox for prediction and integration of dna, rna, and protein data in single cells. *Proc. Thirty*, pages 1–13, 2020.



(a) MOFA UMAP



(b) TOTALVI UMAP



(c) Multigrate UMAP

Figure 4: UMAP projections of the latent embeddings for (a) MOFA, (b) TOTALVI, and (c) Multigrate. Here, each plot is stacked vertically rather than side by side.