

Additional file 3

Compact and Informative Representation Learning for scRNA-seq Data Clustering with Masked Information Bottleneck

Xiaoqiang Yan, Fengshou Han, Yunpeng Wu, Zhen Tian

This PDF file includes:

Note S1 to S4:

- Note S1: Architectural extension of scMIB for multi-omic learning.
- Note S2: The optimization process of scMIB.
- Note S3: Proof procedure for transforming mutual information maximization into the problem of minimizing binary cross-entropy. Model implementation.
- Note S4: Model implementation.

Supplementary Notes

Note S1: Architectural extension of scMIB for multi-omic learning.

In adapting the architecture, we preserved the core design principles of scMIB while allowing modality-specific processing. Each omic modality was encoded by an independent encoder with the same structural design as the original scRNA-seq encoder. The latent representations produced by these encoders were then concatenated and passed through a dedicated fusion layer that learned to integrate cross-modal information into a unified embedding. All loss components that originally operated at the single-modality level, including the mask prediction loss, reconstruction loss, and information bottleneck regularization, continued to be applied independently within each modality. This design ensures that noise-aware denoising and information compression are preserved at the omic-specific level. In contrast, the semantic alignment term was elevated to the fused representation space. The contrastive objective therefore encouraged consistency between the fused latent embeddings derived from masked and clean inputs, effectively aligning multi-omic representations in a corruption-robust manner and propagating cross-modal structure to the final embedding. The resulting unified embedding was subsequently used as input to standard external clustering algorithms, such as K-means and Leiden, to obtain the final clustering results.

Note S2: The optimization process of scMIB.

Algorithm 1 : The optimization of scMIB

1. **Input:** Single-cell RNA-seq dataset X ; Number of clusters K ; Number of Iterations I_t .
 2. **Random initialization:** Randomly initialize the encoders, decoders, label MLP and perturbation predictor.
 3. **for** $epoch=1$ to I_t **do**
 4. Compute masked data $\tilde{X}$ by Eq. (4).
 5. Obtain masked representations $\tilde{Z}$ and original representations Z .
 6. Calculate compress loss $L_{compress}$ by Eq. (5).
 7. Obtain semantic labels $\tilde{S}$ and S through label MLPs.
 8. Calculate semantic alignment loss $L_{semantic}$ by Eq. (12).
 9. Obtain predicted mask $\hat{M}$ by Eq. (13).
 10. Calculate mask estimation loss L_{mask} by Eq. (16).
 11. Obtain data reconstruction by Eq. (17).
 12. Calculate reconstruction loss $L_{reconstruct}$ by Eq. (17).
 13. Obtain the total loss L_{total} by Eq. (1).
 14. Minimize L_{total} to optimize parameters by gradient descent.
 15. **end**
 16. Perform clustering on the learned representation.
 17. **Output:** The predicted label Y .
-

Note S3: Proof procedure for transforming mutual information maximization into the problem of minimizing binary cross-entropy.

Firstly, we propose a proposition: *Maximizing the mutual information between the true mask M and the predicted mask $\hat{M}$ is equivalent to minimizing the binary cross-entropy loss between M and $\hat{M}$.*

Next, we provide a proof for the proposition, where the mask estimation loss is formulated as follows:

$$L_{mask} = \max I(M; \hat{M}).$$

According to the definition of mutual information, we can rewrite it in terms of entropy:

$$I(M; \hat{M}) = H(\hat{M}) - H(\hat{M} | M).$$

Since $H(\hat{M})$ is a general entropy term, maximizing $I(M; \hat{M})$ is equivalent to minimizing the conditional entropy $H(\hat{M} | M)$:

$$\max I(M; \hat{M}) \Leftrightarrow \min H(\hat{M} | M).$$

By the definition of conditional entropy, $H(\hat{M} | M)$ is formulated as follows:

$$H(\hat{M} | M) = -\mathbb{E}_M \left[\mathbb{E}_{\hat{M} | M} [\log P(\hat{M} | M)] \right].$$

Since $\hat{m}_{ij}$ represents the probability that $m_{ij} = 1$, the likelihood of observing the true mask M under the model's predicted probabilities follows a Bernoulli likelihood function:

$$P(M | \hat{M}) = \prod_{i,j} \hat{m}_{ij}^{m_{ij}} (1 - \hat{m}_{ij})^{1-m_{ij}}.$$

Taking the logarithm, we obtain:

$$\log P(M | \hat{M}) = \sum_{i,j} \left[m_{ij} \log \hat{m}_{ij} + (1 - m_{ij}) \log (1 - \hat{m}_{ij}) \right].$$

The expectation over $\log P(\hat{M} | M)$ results in the following expression for the conditional entropy:

$$H(\hat{M} | M) = -\mathbb{E}_M \left[\sum_{i,j} \left[m_{ij} \log \hat{m}_{ij} + (1 - m_{ij}) \log (1 - \hat{m}_{ij}) \right] \right].$$

Minimizing $H(\hat{M} | M)$ is equivalent to maximizing the log-likelihood of the true mask labels M given the predicted probabilities $\hat{M}$. This leads to the binary cross-entropy (BCE) loss function:

$$L_{mask} = -\frac{1}{C \cdot G} \sum_{i,j} \left[m_{ij} \log \hat{m}_{ij} + (1 - m_{ij}) \log (1 - \hat{m}_{ij}) \right].$$

Note S4: Model implementation.

The model is implemented in Python (version 3.9.16) using PyTorch (version 1.12.0). It is trained for 80 epochs with a batch size of 256, employing the Adam optimizer with the AMSGrad variant and an initial learning rate of 0.001.

The encoder module comprises three fully connected layers, with hidden dimensions of 256 and a latent representation of 128. Each of the first two layers is followed by LayerNorm and a Mish activation function. The label MLP consists of a fully connected layer that maps the latent representation to the class-dimensional output space, followed by a Softmax activation function to normalize the outputs into a probability distribution. While the encoder and label MLP structures are shared between the masked and original data, their parameters remain independent. The perturbation predictor consists of two fully connected layers. The first layer consists of a fully connected layer followed by a Swish activation function, where the former maps the latent representation to a 256-dimensional hidden space. The second fully connected layer transforms the hidden representation into a feature space corresponding to the number of genes. For masked data, the decoder is a single fully connected layer that maps the concatenated latent representation and predicted mask to the original feature space corresponding to the number of genes. The decoder for the original data consists of three fully connected layers and non-linear activation functions. It first applies a linear transformation from the hidden representation to a 256-dimensional space, followed by LayerNorm for normalization and a Mish activation function. The second fully connected layer further maps the representation to 512 dimensions, followed by another LayerNorm and Mish activation. Finally, a linear layer projects the representation into the original feature space.

All experiments in this study were conducted on a Windows 11 machine with 64 GB of RAM and an RTX 4090 GPU (24 GB VRAM) for acceleration.