BioFormer

Ho Anh Dung - Le Thi Phuong Thao

June 2025

1 Introduction

Single-cell RNA sequencing (scRNA-seq) technologies have revolutionized transcriptomics by enabling the profiling of gene expression at single-cell resolution. However, the resulting datasets are high-dimensional, sparse, and often exhibit strong batch effects across studies, making downstream analysis and model generalization challenging. These limitations have motivated the development of robust deep learning approaches that can learn cell representations invariant to batch variation while preserving biological identity.

Recent works have explored transformer-based architectures for this purpose, including scBERT and scGPT. These models adopt masked token prediction and sequence modeling techniques inspired by natural language processing to learn representations of cellular states. scGPT, in particular, demonstrated that transformers can effectively learn gene expression patterns using tokenized, binned values with multi-task objectives.

Building on this foundation, the present work introduces **BioFormer**, a transformer encoder model specifically designed for large-scale modeling of gene expression across diverse studies. BioFormer incorporates multiple inductive biases tailored for single-cell data:

- Removal of positional embeddings to maintain gene-order invariance.
- A special representation token at the first sequence position, analogous to the [CLS] token in BERT, which is used for cell-level supervision.
- A combination of loss functions, including masked language modeling, continuous expression regression, adversarial batch prediction, and an embedding contrastive separation (ECS) loss that enforces cell-type-aware representation similarity.

The model is trained on approximately 4 million cells from 20 datasets obtained from the CELLxGENE portal. Its generalization is evaluated on 7 held-out datasets and downstream perturbation tasks. Results show that BioFormer achieves competitive clustering and batch correction performance, and serves as a strong base model for biological inference tasks.

2 Materials and Dataset

2.1 Pretraining and Test Dataset

The pretraining corpus consists of 20 publicly available single-cell RNA-seq datasets sourced from the CELLxGENE data portal[1]. These datasets cover a broad range of tissues, experimental protocols, and cell types, enabling the model to generalize across biological and technical variation. In total, the combined dataset includes approximately 4 million single-cell profiles.

To evaluate the generalization performance of BioFormer, 7 additional datasets from CELLxGENE were held out as a test set. These were chosen to maximize diversity in tissue types, sequencing protocols, and batch origin. All preprocessing steps were kept identical to ensure compatibility with the trained model.

2.2 Perturbation Dataset

To assess the model's ability to perform biological inference tasks beyond clustering, BioFormer was fine-tuned on the perturbation response prediction task using the dataset from Norman et al. (2019). This dataset consists of single-cell gene expression profiles under CRISPR-based perturbations, providing a benchmark for evaluating continuous gene response prediction. It was not used during pretraining, ensuring a fair downstream generalization evaluation.

3 Methodology

3.1 Preprocessing Pipeline

To standardize the input format, a preprocessing pipeline was applied:

- **Highly Variable Gene (HVG) selection:** Performed globally across all datasets to identify a shared vocabulary of informative genes.
- **Gene vocabulary construction:** A fixed vocabulary of 1000 genes was selected based on HVG frequency and expression consistency across datasets.
- **Expression binning:** Expression values were quantile-binned into 51 discrete levels per gene. This includes a reserved bin for zeros to handle sparsity and dropout noise.
- **Tokenization:** Each cell was represented as a sequence of 1000 tokens, where each token combines gene identity and its corresponding binned value.

[1] `https://cellxgene.cziscience.com/datasets`

3.2 Model Architecture

BioFormer is a transformer-based model tailored for single-cell gene expression modeling. Its architecture is designed to be permutation-invariant with respect to gene order and to support multiple biological objectives via distinct output heads. Figure 1 shows an overview of the full model flow.

1. Input Representation Each single cell is represented as a fixed-length sequence of 1000 tokens, where each token corresponds to a gene selected from a predefined vocabulary of highly variable genes.

For each gene position, the final token embedding is the sum of four components:

- **Gene Embedding:** Represents gene identity.
- **Value Embedding:** Encodes the quantile-binned expression level.
- **Study Embedding:** Indicates which dataset/study the cell originates from.
- **Cell Type Embedding:** Supervises training with ECS loss; used only during training.

2. Transformer Encoder Stack The embedded sequence is passed through a stack of 12 transformer encoder layers, each with 8 attention heads and hidden dimension 512. No positional encoding is used, as gene order is biologically unstructured [1]. Layer normalization, residual connections, and feedforward networks follow the original Transformer design [10].

3. Special [CLS] Token A special token, [CLS], is prepended to each input sequence. Its embedding at the final layer serves as a summary representation of the entire cell. This vector is routed to classification and contrastive heads.

4. Output Heads The model uses four parallel output heads:

- **MLM Head:** Predicts the masked binned expression tokens (cross-entropy loss).
- **Continuous Regression Head:** Predicts real-valued gene expression (MSE loss).
- **Adversarial Head:** Predicts batch/study ID from [CLS] using a gradient reversal layer.
- **ECS Head:** Applies contrastive loss over [CLS] embeddings based on cell type.

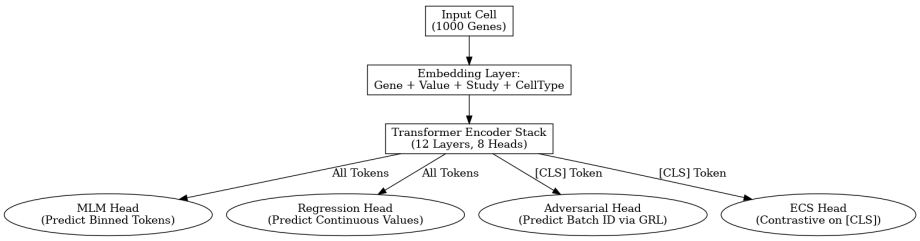

Figure 1: BioFormer architecture: gene-tokenized input flows through embedding layers, a 12-layer transformer encoder, and branches into multiple output heads.

3.3 Loss Functions

BioFormer is trained using a multi-task loss that combines four complementary objectives. Each component addresses a distinct modeling goal in single-cell expression learning. The final training loss is a weighted combination of these components.

1. Masked Language Modeling (MLM) Loss This loss is adapted from the masked token prediction used in BERT [2] and scGPT [1]. In BioFormer, gene expression is tokenized into discrete bins. A random subset of gene tokens is masked during training, and the model is trained to recover their correct binned values using contextual information.

- $\mathcal{M}$: set of masked gene positions
- x_i: true binned token (discrete bin index) for gene i
- $\hat{x}_{\setminus i}$: model input excluding the masked position i

$$\mathcal{L}_{\text{MLM}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} -\log p(x_i \mid \hat{x}_{\setminus i})$$

This loss enables BioFormer to learn gene co-expression patterns and contextual relationships across a cell's transcriptome.

2. Continuous Expression Regression (MSE) Loss To complement the discrete MLM prediction, BioFormer includes a regression head to predict real-valued expression magnitudes. This loss allows the model to retain fine-grained quantitative information.

- y_i: true log-normalized expression value for gene i
- $\hat{y}_i$: model's predicted value for gene i
- m_i: binary mask where $m_i = 1$ if gene i was masked, 0 otherwise

- N: total number of genes in the sequence

$$\mathcal{L}_{\text{cont}} = \frac{1}{\sum_i m_i} \sum_{i=1}^{N} m_i \cdot (y_i - \hat{y}_i)^2$$

This loss is only computed at masked positions and avoids penalizing unmasked or padded genes, which is essential due to sparsity in scRNA-seq data [1].

3. Adversarial Batch Prediction Loss (GRL) This component ensures that learned embeddings do not encode batch- or study-specific signals. It employs a domain-adversarial training strategy with a Gradient Reversal Layer (GRL) [3], which inverts gradients to prevent the model from predicting study labels.

- h_{CLS}: [CLS] token embedding representing the whole cell
- $D(\cdot)$: classifier predicting batch/study ID
- y_{batch}: ground truth batch label

$$\mathcal{L}_{\text{adv}} = \text{CrossEntropy}(D(\text{GRL}(h_{\text{CLS}})), y_{\text{batch}})$$

This loss encourages batch-invariant representations and improves the model's ability to generalize across datasets.

4. Embedding Contrastive Separation (ECS) Loss ECS loss is designed to improve **biological clustering** in the embedding space. It uses the [CLS] token as a cell-level summary and encourages:

- Cells of the **same type** to have similar embeddings
- Cells of **different types** to be separated by a margin τ

Let:

- h_i, h_j: [CLS] embeddings of two cells i and j
- c_i, c_j: their cell type labels
- $\delta_{c_i=c_j}$: indicator function, equals 1 if cells are of the same type, 0 otherwise
- $\delta_{c_i\neq c_j}$: indicator function, equals 1 if cells are of different types
- N: number of cells in the current mini-batch

$$\mathcal{L}_{\text{ecs}} = \frac{1}{N^2} \sum_{i,j} \left[\delta_{c_i=c_j} \cdot \|h_i - h_j\|^2 + \delta_{c_i \neq c_j} \cdot (\max(0, \tau - \|h_i - h_j\|))^2 \right]$$

This loss is inspired by contrastive learning objectives such as triplet loss [4] and supervised contrastive learning [6]. It improves the separability of cell types in the latent space and benefits clustering quality in downstream analyses.

Total Loss The overall objective is the weighted sum of all loss components:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{MLM}} \mathcal{L}_{\text{MLM}} + \lambda_{\text{cont}} \mathcal{L}_{\text{cont}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{\text{ecs}} \mathcal{L}_{\text{ecs}}$$

Here, λ coefficients control the relative contribution of each term during training and are tuned as hyperparameters.

3.4 Training Configuration

BioFormer is trained on multiple GPUs using PyTorch DistributedDataParallel (DDP) to support large-scale parallelism. Automatic mixed precision (AMP) is used to reduce memory usage and accelerate training.

Key training configurations include:

- **Number of transformer layers:** 12
- **Number of attention heads per layer:** 8
- **Hidden size:** 512
- **Feedforward expansion:** 2048
- **Dropout rate:** 0.1
- **Batch size:** 78 (adjusted based on available GPU memory)
- **Optimizer:** Adam with initial learning rate 1×10^{-4}
- **Weight decay:** 1×10^{-5}
- **Loss weights:** $\lambda_{\text{MLM}} = 1.0$, $\lambda_{\text{cont}} = 0.1$, $\lambda_{\text{adv}} = 0.1$, $\lambda_{\text{ecs}} = 0.1$
- **Training epochs:** 10

Model checkpoints were saved after each epoch. The best-performing checkpoint was selected based on validation clustering metrics. Evaluation and visualization were conducted using extracted [CLS] token embeddings and UMAP projection.

4 Evaluation Metrics

To evaluate the performance of BioFormer, both clustering fidelity and batch effect correction were assessed using standard metrics. These metrics were computed based on UMAP projections of [CLS] token embeddings followed by Leiden clustering.

4.1 Normalized Mutual Information (NMI)

NMI measures the similarity between the predicted clustering (e.g., via Leiden algorithm) and ground-truth cell type labels. It is defined as:

$$\text{NMI}(X, Y) = \frac{2 \cdot I(X; Y)}{H(X) + H(Y)}$$

where $I(X; Y)$ is the mutual information between cluster assignments X and labels Y, and $H(\cdot)$ is the entropy. NMI ranges from 0 (no agreement) to 1 (perfect match). It is widely used for evaluating clustering quality in single-cell analysis [9].

4.2 Adjusted Rand Index (ARI)

ARI compares clustering assignments to the true labels, adjusted for chance grouping. It penalizes incorrect splits and merges. Values near 1 indicate strong clustering agreement:

$$\text{ARI} = \frac{\text{RI} - \text{Expected RI}}{\max(\text{RI}) - \text{Expected RI}}$$

ARI is a robust choice for comparing unsupervised clusterings to ground truth [5].

4.3 Silhouette Score

The Silhouette Score measures intra-cluster cohesion and inter-cluster separation using cosine distance. For each point i:

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

where $a(i)$ is the mean intra-cluster distance and $b(i)$ is the mean nearest-cluster distance. Scores range from -1 to 1; higher values indicate well-defined clusters.

To standardize interpretation, the silhouette score was normalized to $[0, 1]$:

$$\text{Silhouette}_{\text{norm}} = \frac{1 + \text{Silhouette}_{\text{raw}}}{2}$$

This metric is frequently used in cluster validation studies [8].

4.4 Graph Connectivity Score

Graph connectivity quantifies how well cells of the same type are mixed across different batches or studies in the learned embedding space. It is a local measure of batch effect correction and reflects whether biologically similar cells remain neighbors despite technical differences.

This metric is based on constructing a k-nearest neighbor (kNN) graph using cosine similarity between cell embeddings. For each cell i, its k nearest neighbors $\mathcal{N}_i$ are identified. The score focuses on how many of these neighbors share the same cell type as i but originate from a different batch or study.

Formally, the Graph Connectivity Score (GC) is defined as:

$$\mathrm{GC} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} \sum_{j \in \mathcal{N}_i} 1 \left\{ \mathrm{celltype}_j = \mathrm{celltype}_i \wedge \mathrm{study}_j \neq \mathrm{study}_i \right\}$$

where:

- N is the total number of cells.
- $\mathcal{N}_i$ is the set of k nearest neighbors of cell i.
- $1\{\cdot\}$ is the indicator function, returning 1 if the condition is true, 0 otherwise.
- $\mathrm{celltype}_i$ and study_i are the cell type and dataset identifier for cell i, respectively.

This metric rewards configurations where a cell's neighborhood includes other cells of the same type that originate from different batches. A higher score implies more effective batch mixing without compromising biological identity, making it useful for evaluating integration quality in multi-study single-cell analyses [7].

4.5 Aggregate Metrics

To summarize performance, two composite scores were used:

- **AvgBio:** Mean of NMI, ARI, and normalized Silhouette Score — captures biological signal quality.
- **AvgBatch:** Graph Connectivity Score — indicates effectiveness of batch correction.

5 Results

5.1 Quantitative Evaluation

BioFormer was evaluated on seven held-out datasets from the CELLxGENE portal using the metrics described in Section 4. The results demonstrate strong performance in both clustering fidelity and moderate batch integration:

- **NMI Score:** 0.9174
- **ARI Score:** 0.7041
- **Silhouette Score:** 0.9278
- **Graph Connectivity Score:** 0.3094
- **AvgBio Score:** 0.8498
- **AvgBatch Score:** 0.3094

These metrics suggest that BioFormer successfully captures biologically relevant structures while achieving reasonable batch mixing, even without explicit domain alignment losses.

5.2 UMAP Visualization

To visualize the learned cell embeddings, UMAP projections were generated using the [CLS] token from each cell. Two plots were produced for interpretation:

1. Coloring by **Study ID** and marker shapes for **Cell Type**
2. Coloring by **Cell Type** and marker shapes for **Study ID**

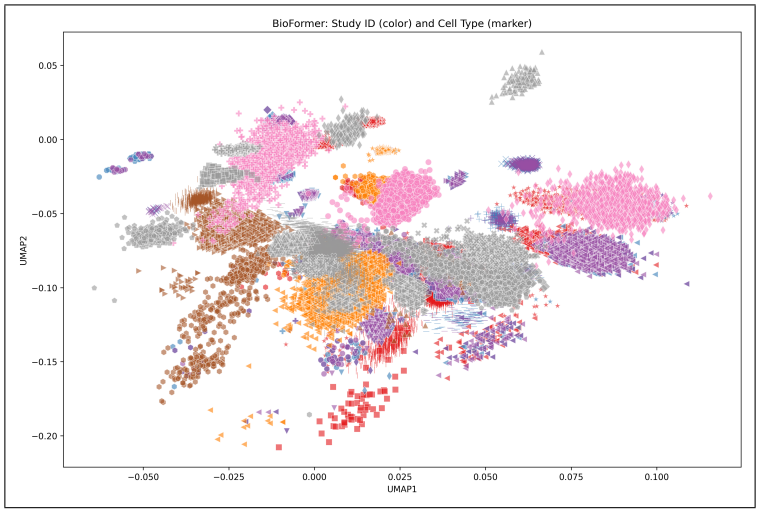

Figure 2: UMAP projection of cell embeddings colored by study ID. Marker shapes represent different cell types.

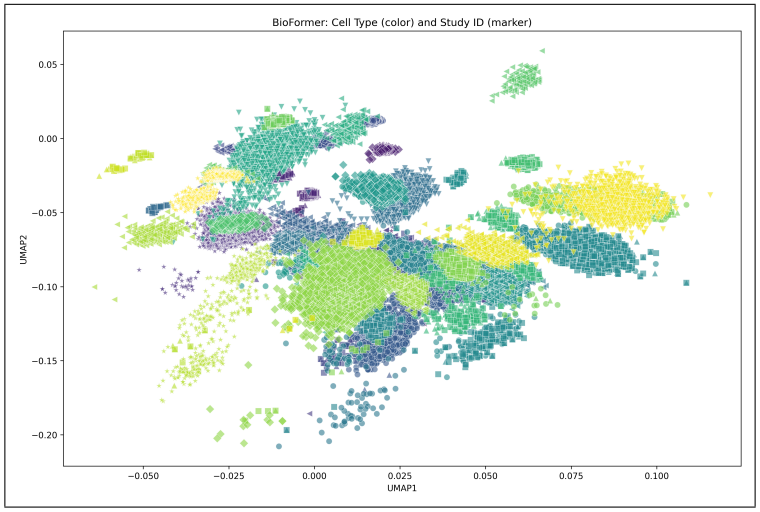

Figure 3: UMAP projection of cell embeddings colored by cell type. Marker shapes represent different studies.

These visualizations show clear biological clustering and moderate overlap across batches, reinforcing the model's ability to generalize across experimental domains.

6 Discussion and Future Work

The results demonstrate that BioFormer is capable of learning robust and biologically meaningful representations from large-scale, heterogeneous single-cell RNA-seq datasets. The use of multi-task training — combining masked prediction, continuous regression, adversarial learning, and contrastive supervision — allows the model to generalize well across both biological and technical variation.

6.1 Biological Representations

High scores on NMI (0.9174), ARI (0.7041), and normalized Silhouette (0.9278) show that BioFormer preserves fine-grained cell type distinctions. This indicates that even without relying on positional encodings or spatial transcriptomics, gene expression sequences can be effectively modeled as unordered token sequences.

The use of a dedicated [CLS] token and the ECS loss plays a critical role in ensuring that the latent embedding space reflects biologically relevant structure. By encouraging cells of the same type to cluster while separating dissimilar ones, ECS improves interpretability and stability of downstream clustering.

6.2 Batch Effect Handling

To effectively handle batch effects, BioFormer integrates multiple design choices and loss objectives that work jointly to minimize study-specific biases in cell representations.

1. Study Embeddings Each cell is annotated with a learnable **study embedding** that captures the identity of the dataset it originates from. This embedding is added to the token inputs during training, allowing the model to learn how to generalize across technical variations without overfitting to them.

2. Adversarial Training with Gradient Reversal A dedicated **adversarial batch prediction head** is attached to the [CLS] embedding. During training, a Gradient Reversal Layer (GRL) inverts the gradients passed to the shared encoder layers. This causes the encoder to learn features that are *uninformative* about the batch identity while still being predictive for the main biological objectives. This method follows the domain-adversarial strategy introduced in [3].

3. Embedding Contrastive Separation (ECS) The **ECS loss** uses the [CLS] embeddings to explicitly group cells of the same type (regardless of batch) closer together while pushing apart cells of different types. This forces the latent space to organize based on biological identity rather than technical origin.

4. Quantitative Support Quantitative results show that BioFormer achieves a Graph Connectivity Score of 0.3094, indicating that same-type cells across studies are moderately well mixed in local neighborhoods. UMAP visualizations further confirm that cell clusters form by type rather than by batch, demonstrating successful batch effect mitigation.

This combined approach — leveraging both architectural embeddings and targeted loss components — enables BioFormer to remove batch effects without sacrificing biological signal fidelity.

6.3 Limitations and Future Directions

Despite promising results, several limitations remain:

- **Binning-based tokenization:** The quantile binning method used to discretize expression values may lead to information loss, especially for low-abundance genes. Future models may benefit from adopting assay-specific token schemes as in scGPT-spatial or TranscriptFormer.
- **Scale and efficiency:** BioFormer was trained on 4 million cells, but recent datasets like Tahoe 100M present opportunities to scale further. Pretraining on larger, unified corpora may improve generalization to unseen perturbations and rare cell types.

- **Architecture alternatives:** Currently, BioFormer uses a transformer encoder backbone. Decoder-based models such as TranscriptFormer, which model autoregressive gene distributions, may offer advantages for generative or perturbation-aware tasks.
- **Perturbation modeling:** While BioFormer performed well on perturbation downstream tasks, future work can integrate perturbation tokens directly into the sequence or condition training using gene knockouts to simulate causal inference.

6.4 Applications and Extensibility

The current model can be fine-tuned or extended to various tasks, including:

- Cell type classification and annotation
- Perturbation response prediction

BioFormer provides a clean and extensible foundation for further innovations in single-cell representation learning.

Code Repository

The full implementation of BioFormer, including training scripts, evaluation pipelines, and preprocessing code, is available at:

`https://github.com/dung-h/BioFormer`

References

[1] Hao Cui, Wenhao Gao, Yuxin Xie, Zhihan Hu, Shuyi Wang, Junyu Li, Yifan Yu, Ning Zhang, Tian Zhao, Xuegong Xie, et al. scgpt: Pre-trained transformer models for single-cell omics. *bioRxiv*, 2023.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[3] Yaroslav Ganin and Victor Lempitsky. Domain-adversarial training of neural networks. 2016.

[4] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[5] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[6] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, et al. Supervised contrastive learning. In *NeurIPS*, 2020.

[7] Malte D Luecken and Fabian J Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022.

[8] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[9] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3:583–617, 2002.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, pages 5998–6008, 2017.