scCONTRAST: A CONTRASTIVE LEARNING BASED APPROACH FOR ENCODING SINGLE-CELL GENE EXPRESSION DATA

Winston Li
Department of Computer Science
Brown University
Providence, RI 02906, USA
winston_y_li@brown.edu

Ghulam Murtaza
Department of Computer Science
Brown University
Providence, RI 02906, USA
ghulam_murtaza@brown.edu

Ritambhara Singh *
Department of Computer Science and
Center for Computational and Molecular Biology
Brown University
Providence, RI 02906, USA
ritambhara@brown.edu

ABSTRACT

Single-cell RNA sequencing (scRNA-seq) captures gene expression at a individual cell resolution, which reveals critical insights into cellular diversity, disease processes, and developmental biology. However, a key challenge in scRNA-seq analysis is clustering similar cells across multiple batches, particularly when distinct sequencing protocols are used. In this work, we present scContrast, a semi-supervised contrastive learning method tailored for embedding scRNA-seq data from both plate- and droplet-based protocols into a universal representation space. By leveraging five simple augmentations, scContrast extracts biologically relevant signals from gene expression data while filtering out batch effects and technical artifacts. We trained scContrast on a subset of Tabula Muris tissues and evaluated its zero-shot performance on unseen tissues. Our results demonstrate that scContrast generalizes effectively to new tissues and outperforms the leading UCE approach in integrating scRNA-seq data from droplet- and plate-based sequencing protocols.

1 INTRODUCTION

Single-cell RNA sequencing (Jovic et al., 2022) (scRNA-seq) enables researchers to profile gene expression at the level of individual cells, permitting insights into cellular heterogeneity (Ximerakis et al., 2019), developmental processes (Qiu et al., 2022), and disease mechanisms (Mathys et al., 2019). A typical scRNA-seq analysis begins by clustering similar cells and identifying key marker genes for each cluster to map out differentiation pathways or uncover disease mechanisms (Su et al., 2022). However, clustering cells based on the full gene expression profile is particularly challenging due to the high dimensionality, sparsity, and batch effects inherent to scRNA-seq data (Bouland et al., 2023).

Existing computational methods for clustering scRNA-seq data can be broadly divided into three categories. The first category includes unsupervised approaches, which typically involve a highly variable gene selection followed by dimensionality reduction methods such as Principal Component Analysis (PCA) to get low-dimensional cell-level representations that cluster similar cells together. While such approaches work well for individual datasets, these approaches struggle to integrate data from multiple batches because they cannot account for batch effects or technical artifacts (Yu

*Corresponding Author

et al., 2024). To resolve these challenges, a second category of methods that includes supervised approaches such as scVI (Lopez et al., 2018) requires batch labels to assign similar representations to similar cells. However, these supervised approaches need to be retrained for every new integration, making them computationally expensive and impractical for large-scale datasets. More recently, inspired by the success of large foundational models such as ChatGPT (Brown et al., 2020), LLAMA (Touvron et al., 2023), and SAM (Kirillov et al., 2023), researchers have developed similar foundational models for scRNA-seq (Theodoris et al., 2023; Yang et al., 2022; Cui et al., 2024; Rosen et al., 2023). These foundational models are extremely large transformer-based architectures trained on atlas-scale datasets with a masked language modeling (MLM) objective. While these models promise to project arbitrary scRNA-seq data into a universal embedding space, it has been shown that these models struggle to effectively integrate data generated from different experimental protocols (Kedzierska et al., 2023).

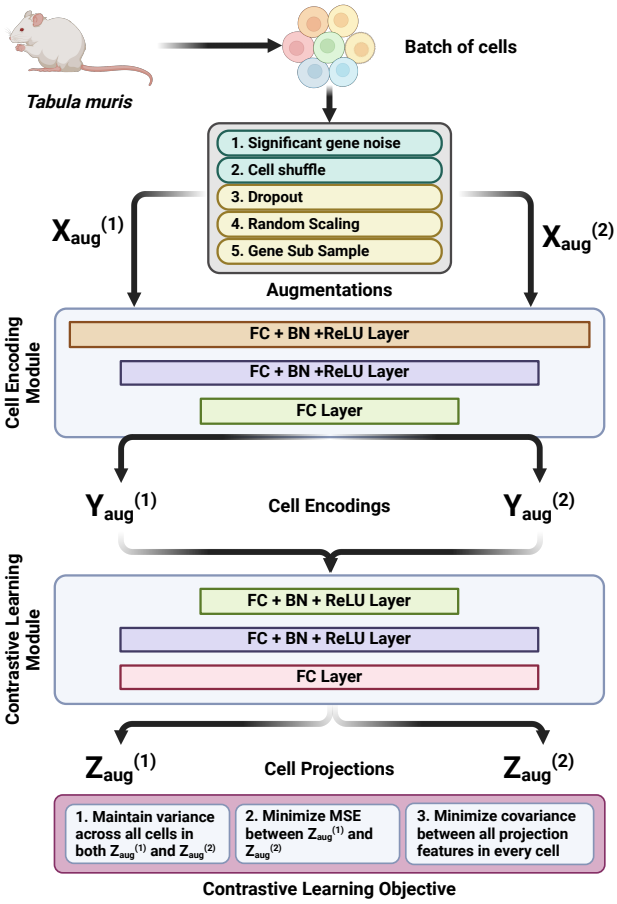

Figure 1: Overview of scContrast pipeline. First, given a batch of samples as a gene expression matrix, two augmented views are generated. Then, the views are encoded by a series of fully connected layers. Next, the encodings are projected into a higher dimension space by another series of fully connected layers. Finally, the projections are used in a contrastive learning objective that simultaneously maintain variance across samples, minimize MSE between augmented views, and minimize covariance between projection features.

This work introduces **scContrast**, summarized in Figure. 1, which is a semi-supervised contrastive learning approach designed to embed scRNA-seq data from different experimental protocols into a universal cell-level representation space. scContrast leverages a simple yet effective set of five augmentation functions, which simulate the technical artifacts, protocol biases, and batch effects characteristic of scRNA-seq assays. These augmentation functions guide scContrast to extract biologically meaningful features of single-cell gene expression profiles.

We trained scContrast on 21 out of 25 tissue samples from the Tabula Muris dataset (Consortium, 2018), which includes data from multiple donor mice sequenced using both droplet- and plate-based sequencing platforms. To simulate a true zero-shot setting, we tested scContrast on the remaining four held-out tissues. Through our evaluations, we found that scContrast generalizes well to unseen tissue samples and outperforms the state-of-the-art UCE model in integrating batches from different sequencing protocols. Specifically, scContrast scores 0.18 and 0.73 on the kBET and PCR batch integration metrics, while UCE scores 0.02 and 0.39, respectively. Thus, our findings highlight scContrast's superior zero-shot cross-protocol batch integration capabilities. However, scContrast performs slightly worse on KMeans ARI and ASW bio-conservation metrics compared to UCE, suggesting opportunities for improvement in the future.

2 METHOD

As shown in Figure 1, scContrast has three main components:

2.1 AUGMENTATIONS MODULE

The augmentations module take in a batch of cells from the Tabula Muris dataset to produce two augmented views X^1_{aug} and X^2_{aug} in a semi-supervised fashion. For each sample, the corresponding augmented views form a positive pair. This module relies on five augmentation functions that aim to perturb the gene expression data in a manner that preserves each sample's cellular identity. Such functions can be sub-divided into two categories. The first category of augmentation functions require cell-type annotations (highlighted in green in Figure. 1). By leveraging the cell-type annotations of each sample, they ensure samples from the same cell-type maintain similar representations while challenging the model to distinguish between distinct cell-types. The significant gene noise augmentation injects noise into each cell's most differentially expressed genes, which promotes robustness to expression variability in cell-type defining genes. The cell-shuffle augmentation permutes the gene expression profiles of samples of the same cell-type, which forces the model to treat similarly cells sharing an annotation.

The second category of augmentation functions does not rely on any annotations (highlighted with yellow color in the Figure. 1). These augmentation functions randomly dropout, subsample, or rescale gene expression across all cells to simulate random dropout events and sequencing protocol biases, consequently guiding the model to ignore these biases.

2.2 CELL ENCODING MODULE

The core purpose of the Cell Encoding Module is to encode cells into a low-dimensional representation space that preserves biological semantic properties such as cell-type similarities, relationships, and developmental trajectories. Unlike existing scRNA-seq foundational models such as UCE (Rosen et al., 2023), Geneformer (Theodoris et al., 2023), scGPT (Cui et al., 2024), and scBERT (Yang et al., 2022) that have transformer blocks with self-attention, we rely on simple, fully connected neural networks to extract relationships between all genes in the cell. Specifically, we have two fully connected layers with batch normalization to stabilize the learning process and ReLU to capture non-linear relationships, followed by a single fully connected layer to produce cell encodings Y^1_{aug} and Y^2_{aug}. The entire Cell Encoding Module has 5.2 million parameters, which is 10—100 times smaller than existing scRNA-seq foundational models (Theodoris et al., 2023; Rosen et al., 2023).

2.3 CONTRASTIVE LEARNING MODULE

The Contrastive Learning Module is responsible for comparing the two augmented encodings generated by the Cell Encoding Module. It is inspired by VICReg (Bardes et al., 2022) and aims to ensure **V**ariance, **I**nvariance, and **C**ovariance among each batch of projected sample encodings to further enrich latent space representation. We use the same encoder and projection head for each augmented view.

More specifically, outputs from the Cell Encoding Module are first projected into $Z^{(1)}_{aug}$ and $Z^{(2)}_{aug}$, respectively, using a lightweight, trainable projection head. Then, a loss comprising three terms are

calculated from these projected augmentations as shown in the equation below, where X and Y are $Z_{aug}^{(1)}$ and $Z_{aug}^{(2)}$, respectively:

$$\mathcal{L}_{\text{VICReg}}(X,Y) \quad = \quad \mu \underbrace{\Big(\ell_{\text{var}}(X) + \ell_{\text{var}}(Y)\Big)}_{\text{variance}} + \underbrace{\lambda \|X - Y\|_2^2}_{\text{invariance}} + \nu \underbrace{\Big(\ell_{\text{cov}}(X) + \ell_{\text{cov}}(Y)\Big)}_{\text{covariance}},$$

where the variance term is $\ell_{\text{var}}(V) = \frac{1}{d}\sum_{j=1}^{d} \max\Big(0, \gamma - \sqrt{\text{Var}(V_{:,j}) + \epsilon}\Big)$, the invariance term is $\|X - Y\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \|x_i - y_i\|_2^2$, and the covariance term is $\ell_{\text{cov}}(V) = \frac{1}{d}\sum_{i \neq j} \big[C(V)\big]_{i,j}^2$, such that $V \in \{X, Y\}$, γ is the target standard deviation, ϵ is a small constant (e.g., 10^{-4}) for numerical stability, and $C(V)$ is the empirical covariance matrix of embeddings V, which is squared and summed over all off-diagonal elements. Coefficients μ, λ, and ν are tunable hyperparameters that weigh each loss term.

2.4 Implementation details

We implemented the entire pipeline in `Python==3.9.16` using `Pytorch==2.5.1+cu118` and `Pytorch-Lightning==2.4.0`. scContrast produces cell encodings with 128 dimensions and cell projections with 256 dimensions. We use the cell encodings Y for all of our evaluations and visualizations. We use cell projections Z_{aug}^{1} and Z_{aug}^{2} to calculate the VICReg (Bardes et al., 2022) contrastive loss with a batch size of 8192. We use AdamW optimizer with default parameters to calculate gradients used to update all the weights.

3 Experimental Setup

3.1 Datasets

We obtained scRNA-seq datasets from the Tabula Muris dataset, which is a comprehensive collection of single-cell transcriptomics data generated from both droplet-based (10x Genomics (Zheng et al., 2017)) and plate-based (Smart-Seq2 (Picelli et al., 2014)) sequencing platforms for *Mus musculus* (mouse). The Tabula Muris dataset contains 25 distinct tissues and has more than $350,000$ cells. In order to evaluate the performance of scContrast on out-of-distribution data in a zero-shot setting, we randomly selected four of the twenty-five tissues for testing and kept the rest for training scContrast.

We preprocessed all datasets by first filtering out all cells with 'UKWN'/NULL annotations. Next, we normalized gene expressions of each cell by scaling the total counts to 10000 . Finally, we applied a log-transformation to minimize technical artifacts (Su et al., 2022).

3.2 Baselines

In order to comprehensively compare the performance of scContrast against existing approaches, we include:

(a) **PCA**: A fully unsupervised method that decomposes the cell-by-gene matrix into its principal components, producing low-dimensional cell representations.

(b) **scVI** (Lopez et al., 2018): A supervised batch-integration approach that explicitly requires batch annotations. We train scVI with `n_layers=3` and `n_latent=128` to match the size and shape of the Cell Encoding Module of scContrast.

(c) **UCE** (Rosen et al., 2023): A scRNA-seq foundational model with over 650 million parameters trained on an expansive corpus of scRNA-seq dataset containing over 36 million cells and spanning across eight diverse species. UCE can reportedly integrate scRNA-seq datasets with batch effects across unseen tissues and species without any finetuning. We use UCE with its default configurations without any finetuning to generate cell-level representations.

3.3 EVALUATION METRICS

We evaluate the quality of cell encodings across bio-conservation and batch integration metrics defined by the `scib-metrics` package (Luecken et al., 2021), which is commonly used for benchmarking scRNA-seq models. For the sake of conciseness, we show two out of five metrics for each category and the rest of the scores in the Appendix (Table. 2, Table. 3).

We present the following four metrics:

(a) **Bio-conservation metric ARI**: Calculates the Adjusted Rand Index from the KMeans-clustered embedding space of each model.

(b) **Bio-conservation metric ASW**: Calculates the average silhouette width with respect to cell-type.

(c) **Batch integration metric kBET**: Conducts the k-nearest-neighbor batch effect test, which rewards neighborhoods of embeddings having the same distribution of batch labels as that of the full dataset.

(d) **Batch integration metric PCR**: Performs a principal component regression comparison by comparing the explained variance before and after integration.

4 RESULTS

4.1 SCCONTRAST SUCCESSFULLY INTEGRATES GENE EXPRESSION DATA FROM PLATE- AND DROPLET-BASED SEQUENCING PLATFORMS IN UNSEEN TISSUES

scContrast is distinguished from existing approaches by its comparatively light model size, small training set, lack of explicit batch labels during training, and most importantly, its ability to generalize to unseen tissue samples in a zero-shot manner. For baselines, we choose PCA, scVI, and UCE to represent existing unsupervised, fully-supervised, and large-scale foundational model paradigms that are currently used for batch integration, respectively. For evaluation, we use four held-out tissues from Tabula Muris: skin, liver, pancreas, and limb muscle. These tissues feature diverse cellular landscapes ranging from organ-specific cells to common immune cells such (e.g. macrophages and resident variants). In order to visualize the quality of the cell representation spaces, we use UMAP (McInnes et al., 2020), which is a commonly employed visualization method. Moreover, we quantified these representations using ARI, ASW, kBET, and PCR.

As shown in Figure. 4.1, we find that both PCA and UCE struggle to integrate scRNA-seq data from droplet- and plate-based sequencing protocols. Specifically, from the UMAP visualizations, we observe that cells of the same cell-type but different sequencing protocol form very distinct and dense clusters in PCA; those of UCE tend be closer to each other, but they still do not integrate to form homogeneous clusters. In contrast, both scVI and scContrast successfully integrate across sequencing protocols: the plate-based SS2-colored embeddings (orange) overlay atop the droplet-based 10X-colored embeddings (blue), highlighting that representations of the same cell-type across both protocols homogeneously cluster together.

	Bio-Conservation		Batch Integration	
Model	KMeans ARI	ASW	kBET	PCR
PCA	0.42	<u>0.60</u>	0.00	0.00
scVI	0.40	0.56	**0.25**	**0.93**
UCE	**0.51**	**0.65**	0.02	0.39
scContrast	<u>0.43</u>	<u>0.60</u>	<u>0.18</u>	<u>0.73</u>

Table 1: Quantitative comparison of different models on bio-conservation and batch integration of unseen tissues from Tabula Muris. The best score is bolded and the second best scores are underlined.

Quantitatively, UCE performs best on the bio-conservation metrics, highlighting its capacity to cluster cells of the same type closer to each other in the cellular representation space. However, the

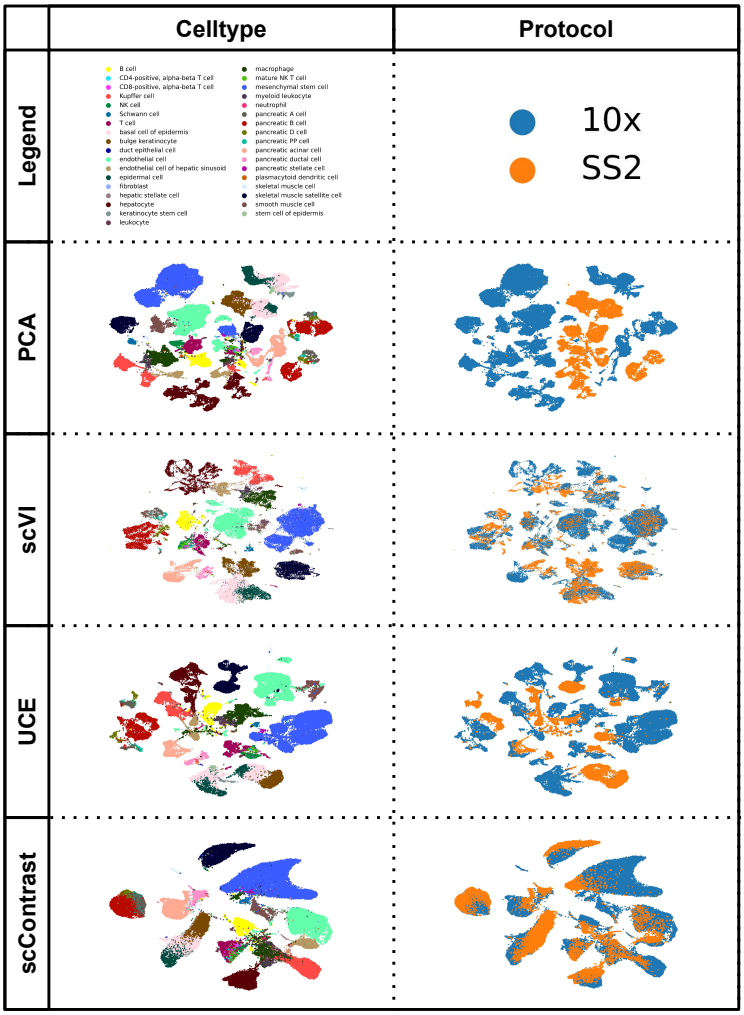

Figure 2: Qualitative comparisons of different models integrating scRNA-seq samples from plate (SS2)- and droplet (10x)-based sequencing platforms in unseen tissues in Tabula Muris

batch integration metrics reflect findings in Figure. 4.1, showing that both PCA and UCE perform worse than scContrast and scVI in integrating samples from different sequencing protocols.

While these results show that our contrastive learning-based approach is promising in learning a universal cell representation space, there is potential to further refine our approach to learn better representation spaces.

5 Discussion and Conclusion

In this work, we have demonstrated a semi-supervised contrastive learning-based approach scContrast for encoding single-cell RNA-sequencing data into a universal cell representation space. We compared our model against PCA, scVI, and UCE, which exemplify current unsupervised, fully supervised, and large foundational model approaches. Results demonstrate that PCA understandably struggles to integrate the batches, since protocol-inherent gene expression biases likely produce spurious axes of variability, which is exacerbated by PCA's reliance on covariances. More interestingly, UCE also struggles, which may reflect the limitations of a purely masked modeling approach. Finally, while scVI performs comparatively best at cross-protocol integration, it explicitly requires batch labels, which scContrast does not.

In the future, we aim to extend scContrast across three dimensions. First, we aim to develop better augmentation functions to closer simulate batch effects and technical artifacts to improve scContrast's zero-shot capabilities. Second, we plan to scale up our training datasets and explore deeper architectures that can learn richer cell representations. Lastly, we plan to investigate techniques like $\mathbb{X}$-sample Contrastive Learning (Sobal et al., 2024) to impose hierarchical cell ontologies in the representation space.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. 2022. URL https://arxiv.org/abs/2105.04906.

Gerard A. Bouland, Ahmed Mahfouz, and Marcel J. Reinders. Consequences and opportunities arising due to sparser single-cell rna-seq datasets. *Genome Biology*, 24(1), Apr 2023. doi: 10.1186/s13059-023-02933-w.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

The Tabula Muris Consortium. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, 562(7727):367–372, Oct 2018. doi: 10.1038/s41586-018-0590-4.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. Scgpt: Toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, Feb 2024. doi: 10.1038/s41592-024-02201-0.

Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo. Single-cell rna sequencing technologies and applications: A brief overview. *Clinical and Translational Medicine*, 12(3), Mar 2022. doi: 10.1002/ctm2.694.

Kasia Z. Kedzierska, Lorin Crawford, Ava P. Amini, and Alex X. Lu. *Assessing the limits of zero-Shot Foundation models in single-cell biology*, Oct 2023. doi: 10.1101/2023.10.16.561085.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

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, Nov 2018. doi: 10.1038/s41592-018-0229-2.

Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, Dec 2021. doi: 10.1038/s41592-021-01336-8.

Hansruedi Mathys, Jose Davila-Velderrain, Zhuyu Peng, Fan Gao, Shahin Mohammadi, Jennie Z. Young, Madhvi Menon, Liang He, Fatema Abdurrob, Xueqiao Jiang, and et al. Single-cell transcriptomic analysis of alzheimer's disease. *Nature*, 570(7761):332–337, May 2019. doi: 10.1038/s41586-019-1195-2.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. URL https://arxiv.org/abs/1802.03426.

Simone Picelli, Omid R Faridani, Åsa K Björklund, Gösta Winberg, Sven Sagasser, and Rickard Sandberg. Full-length rna-seq from single cells using smart-seq2. *Nature Protocols*, 9(1):171–181, Jan 2014. doi: 10.1038/nprot.2014.006.

Chengxiang Qiu, Junyue Cao, Beth K. Martin, Tony Li, Ian C. Welsh, Sanjay Srivatsan, Xingfan Huang, Diego Calderon, William Stafford Noble, Christine M. Disteche, and et al. Systematic reconstruction of cellular trajectories across mouse embryogenesis. *Nature Genetics*, 54(3):328–341, Mar 2022. doi: 10.1038/s41588-022-01018-x.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorcan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. *bioRxiv*, pp. 2023–11, 2023.

Vlad Sobal, Mark Ibrahim, Randall Balestriero, Vivien Cabannes, Diane Bouchacourt, Pietro Astolfi, Kyunghyun Cho, and Yann LeCun. $\mathbb{X}$-sample contrastive loss: Improving contrastive learning with sample similarity graphs, 2024. URL https://arxiv.org/abs/2407.18134.

Min Su, Tao Pan, Qiu-Zhen Chen, Wei-Wei Zhou, Yi Gong, Gang Xu, Huan-Yu Yan, Si Li, Qiao-Zhen Shi, Ya Zhang, and et al. Data analysis guidelines for single-cell rna-seq in biomedical studies and clinical applications. *Military Medical Research*, 9(1), Dec 2022. doi: 10.1186/s40779-022-00434-8.

Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Mantineo, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, May 2023. doi: 10.1038/s41586-023-06139-9.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Methodios Ximerakis, Scott L. Lipnick, Brendan T. Innes, Sean K. Simmons, Xian Adiconis, Danielle Dionne, Brittany A. Mayweather, Lan Nguyen, Zachary Niziolek, Ceren Ozek, and et al. Single-cell transcriptomic profiling of the aging mouse brain. *Nature Neuroscience*, 22 (10):1696–1708, Sep 2019. doi: 10.1038/s41593-019-0491-3.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. Scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, Sep 2022. doi: 10.1038/s42256-022-00534-z.

Ying Yu, Yuanbang Mai, Yuanting Zheng, and Leming Shi. Assessing and mitigating batch effects in large-scale omics studies. *Genome Biology*, 25(1), Oct 2024. doi: 10.1186/s13059-024-03401-9.

Grace X. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, and et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1), Jan 2017. doi: 10.1038/ncomms14049.

A APPENDIX

Supplementary metrics from `scib-metrics` are given in table 2 and table 3. Our scContrast model's architecture is given here here.

	Bio-Conservation				
Model	Isolated Labels	KMeans NMI	KMeans ARI	ASW Labels	cLISI kNN
PCA	**0.67**	0.76	0.42	0.60	1.00
scVI	**0.67**	0.76	0.40	0.56	1.00
UCE	**0.67**	**0.80**	**0.51**	**0.65**	1.00
scContrast	0.62	0.75	0.43	0.60	1.00

Table 2: Additional quantitative comparison of different models on bio-conservation metrics of representation spaces of cells from unseen tissues from Tabula Muris. The best scores are bolded and the second best scores are underlined.

	Batch Integration				
Model	ASW Batch	iLISI kNN	kBET	Graph Connectivity	PCR
PCA	0.54	0.00	0.00	0.76	0.00
scVI	**0.85**	**0.02**	**0.25**	**0.92**	**0.93**
UCE	0.73	0.00	0.02	0.79	0.39
scContrast	0.77	0.01	0.18	0.80	0.73

Table 3: Additional quantitative comparison of different models on batch integration metrics of representation spaces of cells from unseen tissues from Tabula Muris. The best score is bolded and the second best scores are underlined.

```
scRNASeqE_VICRegExpander(
  (encoder): scRNASeqEncoder(
    (encoder): Sequential(
      (0): Linear(in_features=20116, out_features=256, bias=True)
      (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
                       track_running_stats=True)
      (2): ReLU()
      (3): Linear(in_features=256, out_features=128, bias=True)
      (4): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True,
                       track_running_stats=True)
      (5): ReLU()
      (6): Linear(in_features=128, out_features=128, bias=True)
    )
  )
  (projector): scRNASeqProjectionHeadExpander(
    (projectionHead): Sequential(
      (0): Linear(in_features=128, out_features=256, bias=True)
      (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
                       track_running_stats=True)
      (2): ReLU()
      (3): Linear(in_features=256, out_features=256, bias=True)
      (4): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
                       track_running_stats=True)
      (5): ReLU()
      (6): Linear(in_features=256, out_features=256, bias=True)
    )
  )
  (vicreg_loss_fn): VICRegLoss()
)
```