CS230

A Universal Deep Learning Pipeline for Single Cell RNA Prediction Models

Haider Suleman
Department of Computer Science
Stanford University

Sebastian Kronmueller
Department of Biomedical Data Science
Stanford University

Abstract

Recent technological advances have led to the increased use of single cell RNA sequencing. Integration, analysis and predictive modelling of this data is challenging due to the technical bias introduced across individual studies. We implemented an extendable, universal deep learning pipeline consisting of a conditional variational autoencoder for technical bias correction and a deep learning predictive model. We leverage a transfer learning approach for the application of the pipeline to new studies. Our pipeline outperforms non-deep learning batch correction methods.

1 Introduction

In biology, cells are often seen as the fundamental units of life, and the study of cell behavior is critical to our understanding of biological processes. The gene expression profile (or transcriptome) of a cell - the RNA transcripts that have been expressed at a certain point in time and then translated into the proteins driving cell behavior - can convey important scientific information, as it reflects the underlying genomic and epigenomic factors regulating gene expression. Recent technological advances like next generation sequencing technologies and modern microfluidics have enabled researchers to generate increasingly large amounts of transcriptomic data for individual cells, often called single cell RNA sequences (or scRNA-seq). ScRNA-seq data has the potential to provide key insights into biological processes and support the discovery of new biomarkers and drug targets for diseases. However, analyzing this data effectively is challenging due to its volume (each study typically generates transcriptomic profiles for tens of thousands of cells) and its variability with respect to cell type, cell life-cycle stage and donor. Further, beyond the biological variability, there is significant technical variability in the data in practice - so-called batch-effects. The main source of these batch-effects is variation in the efficiency of RNA capture and reverse transcriptase, which can differ strongly between used technologies, instruments, workflows and operators. Figure 1 shows an overview of the typical workflow and sources of technical bias. These batch-effects make the integration of datasets challenging and drive systematic biases which need to be eliminated to generate meaningful scientific insights (see Hwang et al. 2018).

Figure 1: Typical scRNA workflow and sources of bias. Technical bias is introduced in b) and d)

We propose an extendable, universal deep learning pipeline that enables effective scRNA data integration across disparate studies, development of predictive models for biological and clinical variables, and seamless extension to unseen datasets. Our pipeline consists of a conditional variational

autoencoder (CVAE) for technical bias correction wrapped inside an approach to transfer learning known as "architectural surgery", which enables the addition of new studies without retraining of the whole model. The resulting latent representations can be used for various predictive tasks. Here, we demonstrate their usefulness for cell-type prediction. We benchmarked our pipeline against classical algorithms for batch correction and predictive modelling, and demonstrated the benefit of deep learning for each component. Figure 2 illustrates our pipeline.

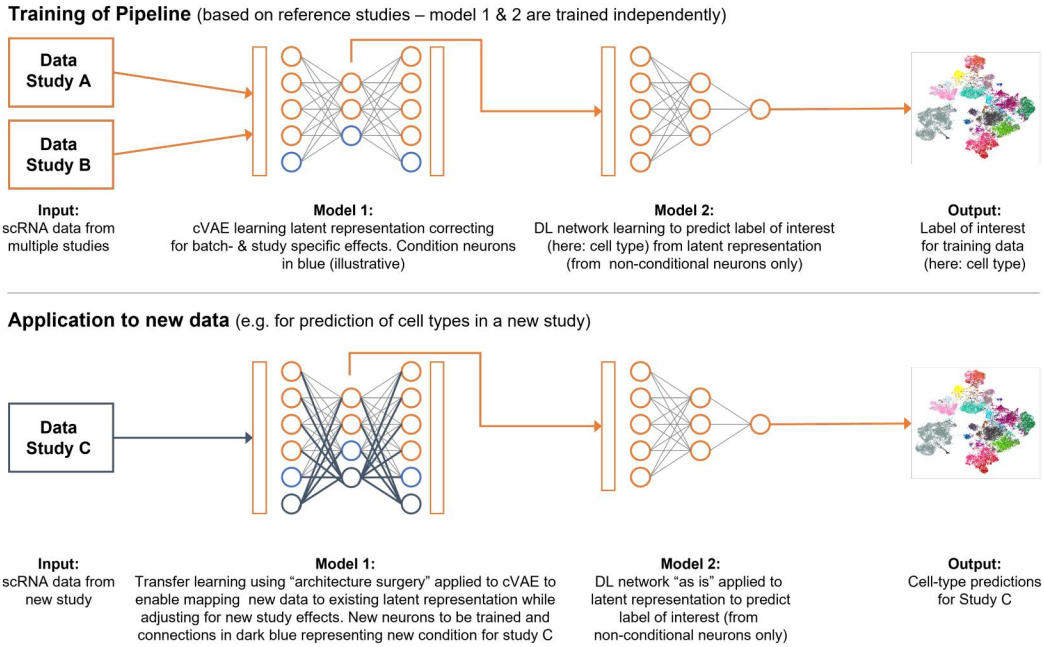

Figure 2: Illustration of pipeline and "architectural surgery" approach to transfer learning

2 Related Work

The use of deep learning models for the batch-correction and analysis of scRNA-seq data is well established (Lopez et al. 2018; M. Lotfollahi et al. 2020; Tran et al. 2020). Recently, the continuing agglomeration of massive scRNA-seq databases, such as the Human Cell Atlas (Regev et al. 2017), has sparked interest in expanding this work to the development of large pretrained models that are scalable and transferable to downstream biological tasks. The most recent, and leading, example of this is the "architectural surgery" methodology proposed by Lotfollahi et al. 2022. We contribute to the literature by implementing an end-to-end deep learning pipeline for scRNA-seq data, demonstrating its applicability to the downstream task of cell-type prediction, and quantifying the benefit of deep learning over classical methods (e.g Harmony batch-correction by Korsunsky et al. 2019) at each step of the pipeline.

3 Dataset and Features

Our dataset comprises three distinct studies containing a total of $\sim$ 330,000 individual cells from lung samples. Each is available publicly via the Human Cell Atlas: 1) Watanabe et al. 2020 - 57,000 cells from 12 donors related to COPD; 2) Chua et al. 2020 - 160,500 cells from 24 donors related to COVID-19; 3) Trump et al. 2021 - 114,800 cells from 48 donors related to COVID-19 and Hypertension. To prepare the dataset, we randomly selected 10,000 cells from each study, selected the 5000 genes with the highest variability within each study and integrated the resulting data in a single matrix of 30,000 cells with 3471 high variability genes (the intersection of the high variability genes across the studies). The Watanabe and Trump studies were chosen to be our training/validation set. The Chua data was reserved to be the "new" study which was integrated via transfer learning (architectural surgery) to the CVAE, and then used as the test set for the predictive model.

4 Methods

4.1 Conditional Variational Autoencoder (Model 1)

Let $\mathbf{x} \in \mathbb{R}^N$ represent a single cell in our dataset, where a feature x_n represents the gene expression count for gene n. Assuming D studies in our training set, $\mathbf{s} \in \mathbb{R}^D$ denotes a one-hot encoding of those studies. The latent variable $\mathbf{z} \in \mathbb{R}^K$, with $K << N$, is a low dimensional feature vector that is intended to capture the variability in $\mathbf{x}$ that is independent of the study-specific effects in $\mathbf{s}$. That is, $\mathbf{z}$ captures the biological variability between cells. We want to maximise the conditional log-likelihood:

$$\log p_\theta(\mathbf{x}|\mathbf{s}) = \log \int p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}) p_\theta(\mathbf{z}|\mathbf{s}) \mathrm{d}\mathbf{z}$$

Here, this integral is intractable, so we optimize the variational lower bound, $\mathcal{L}$, instead:

$$\mathcal{L}(\mathbf{x}, \mathbf{s}; \theta, \phi) = \log p_\theta(\mathbf{x}|\mathbf{s}) - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{s}) \| p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{s})\right) = \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{s}) \| p_\theta(\mathbf{z}|\mathbf{s})\right)$$

where the rightmost equality gives the negative of our loss function. We follow standard (C)VAE practice and assume $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{s})$ to be Gaussian, with its mean and variance predicted by a Decoder network with parameters ϕ. Additionally, the prior $p_\theta(\mathbf{z}|\mathbf{s})$ is assumed to be standard Gaussian. The distribution $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s})$ is chosen to be Negative Binomial (NB), in order to directly model the discrete-count, over-dispersed nature of gene-expression data (Love et al. 2014). The parameters of the NB distribution are predicted by an Encoder network with parameters θ.

To optimize the lower bound $\mathcal{L}$, we use analytic expressions for the Negative Binomial $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s})$, and the KL divergence between two Gaussians. The expectation $\mathbb{E}_{q_\phi}[\cdot]$ involves an intergral over $\mathbf{z}$ that is analytically intractable, so it is evaluated using Monte-Carlo sampling and the re-parametrization trick (Kingma and Welling 2014). The parameters are learned using Adam optimization.

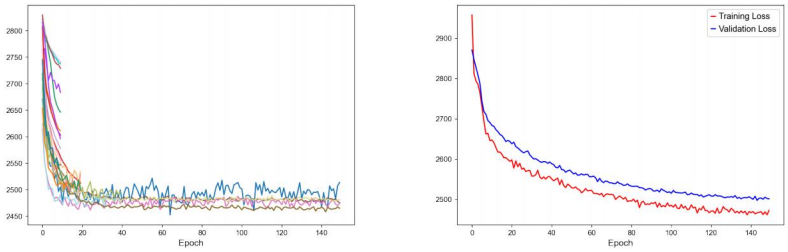

Figure 3: Hyperparameter tuning validation loss (left). Train/validation loss for optimal hyperparameters (right)

For hyperparameter tuning, we conducted a randomized search over a universe that was deemed reasonable relative to the existing literature (Lopez et al. 2018 and Lotfollahi et al. 2022). The trials were run using the Asynchronous HyperBand algorithm (Li et al. 2020), which terminates poor performing hyperparameter combinations early. Our implementation used the Ray Tune library. The left plot of Figure 3 shows the validation loss for the tuning trials.

Subsequently, the Encoder and Decoder were chosen to have three hidden layers each, with 128 hidden units in each layer. Dropout was applied to each hidden layer with a drop probability of 20%. The latent representation $\mathbf{z}$ was chosen to be 10-dimensional (i.e. $K = 10$). Maximum Mean Discrepancy (MMD) regularization was applied to the first layer of the decoder to penalize the encoder for producing output correlated with the conditions $\mathbf{s}$ (M. Lotfollahi et al. 2020). The CVAE was trained for 150 epochs - the right plot of Figure 3 shows that the validation loss stabilises towards the end of training. The implementation was in PyTorch.

4.2 Architectural surgery

Architectural surgery, proposed by (Lotfollahi et al. 2022), is a transfer learning approach specific to conditional generative models and single-cell RNA data. For training the reference CVAE model, the input layer of the Encoder is $(\mathbf{x} \circ \mathbf{s}) \in \mathbb{R}^{N+D}$ and the input layer of the Decoder is $(\mathbf{z} \circ \mathbf{s}) \in \mathbb{R}^{K+D}$, where "$\circ$" denotes concatenation. Suppose we now obtain some additional studies, $\mathbf{s}' \in \mathbb{R}^{D'}$. We would like to extend our reference CVAE to include these studies. To achieve this, we first update the the Encoder input layer to $(\mathbf{x} \circ \mathbf{s} \circ \mathbf{s}') \in \mathbb{R}^{N+D+D'}$ and the Decoder input layer to

$(\mathbf{z} \circ \mathbf{s} \circ \mathbf{s}') \in \mathbb{R}^{K+D+D'}$. The addition of the new studies $\mathbf{s}'$ introduces new parameters into the first hidden layers of the Encoder and Decoder, say $\theta_{s'}$ and $\phi_{s'}$ respectively. Then, in the fine-tuning step, we train *only* the parameters $\theta_{s'}$ and $\phi_{s'}$. That is, we keep the original θ and ϕ frozen. This is illustrated in Figure 2.

4.3 Cell-type Prediction Model (Model 2)

Our predictive model is a fully-connected network with a categorical cross-entropy loss. It takes as input the latent representation $\mathbf{z}$ and outputs a probability distribution over cell-type labels. Following some initial experimentation, we conducted a systematic, randomized hyperparameter search. We considered up to four layers with a maximum of 64 nodes and up to 2 dropout layers. The chosen model consists of three dense layers (64, 32 and 16 nodes) with ReLU activations, one dropout layer with drop probability 30% and a softmax output layer. The model was implemented via Keras and trained for 200 epochs. Figures 4 a) and b) show training/validation loss and accuracy (respectively) on the CVAE latents. Figure 4 c) shows training/validation accuracies for various hyperparameters.

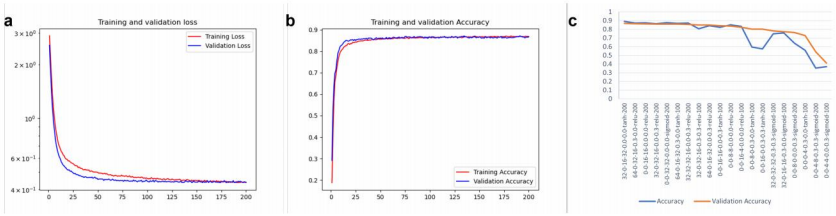

Figure 4: Train/validation a) loss b) accuracy. c) Accuracies from hyperparameter search

5 Experiments/Results/Discussion

5.1 Evaluation metrics and Benchmarks

We benchmarked the data integration performance of our CVAE first in terms of the batch correction and conservation of biological variation embedded in the latent representation $\mathbf{z}$. Batch correction was evaluated using the metrics Entropy of Batch Mixing (EBM) and Average Silhouette Width (ASW) on batch labels. Meanwhile, biological conservation was measured using Normalized Mutual Information (NMI) and ASW on cell labels. All metrics are normalized to lie in the range $[0, 1]$, with values closer to 1 indicating better batch correction or biological conservation. Roughly, all stated metrics measure the ability to create well-defined clusters by batch or cell-type. As such, the metrics have been normalized so that we penalize the ability to cluster by batch, but reward the ability to do so by cell-type. A detailed discussion of these metrics can be found in (Luecken et al. 2021). The CVAE was benchmarked against a straightforward dimensionality reduction via PCA and against Harmony (Korsunsky et al. 2019), which is an iterative, non-deep learning batch correction algorithm applied on top of PCA. In each case, the lower dimensional representation had 10 dimensions (i.e. $K = 10$).

Further, we benchmarked the predictive performance of our pipeline on the downstream task of cell-type prediction. The metric used is prediction accuracy versus ground truth labels. The benchmark models are Softmax regression and SVM.

PCA and Harmony were implemented using Scanpy (Wolf et al. 2018); EBM, ASW and NMI using scArches (Lotfollahi et al. 2022); and softmax regression and SVM using scikit-learn.

5.2 Results and Discussion

The data integration results are reported post-architectural surgery, i.e. inclusive of all three studies. The cell-type prediction results are reported both for the training/validation set and the test set.

The left sub-table of Table 1 shows that the CVAE achieves the best batch correction and outperforms Harmony in terms of bio-conservation. However, both the CVAE and Harmony achieve less bio-conservation than the PCA. This is expected, as the CVAE and Harmony will lose some biological variability as they attempt to eliminate the batch effects. The UMAP plots in Figure 5 corroborate the metrics. The right subplots show that the cell-type clusters are most compact and well-defined for the CVAE. Consider for example the ciliated cells (dark-purple dots on the right). In the CVAE

UMAP, we are able to smoothly overlay the three studies (blue, orange, green dots on the left) and thereby cluster the ciliated cells together. Harmony exhibits a similar phenomenon, albeit less clearly. PCA suffers from the batch effects and fails to group the ciliated cells together. Therefore, the CVAE seems to provide the best trade-off between batch correction and bio-conservation.

Table 1: Data integration metrics (left). Cell-type prediction accuracy (right)

	Batch Correction		**Bio-Conservation**	
	EBM	ASW	NMI	ASW
PCA	0.32	0.70	0.69	0.51
Harmony	0.48	0.76	0.6	0.47
CVAE	0.71	0.91	0.65	0.50

	Train (Trump+Watanabe)			**Test (Chua)**		
	Softmax	SVM	NN	Softmax	SVM	NN
PCA	0.88	0.92	0.89	0.67	0.69	0.71
Harmony	0.81	0.91	0.85	0.61	0.65	0.62
CVAE	0.85	0.88	0.88	0.65	0.66	0.67

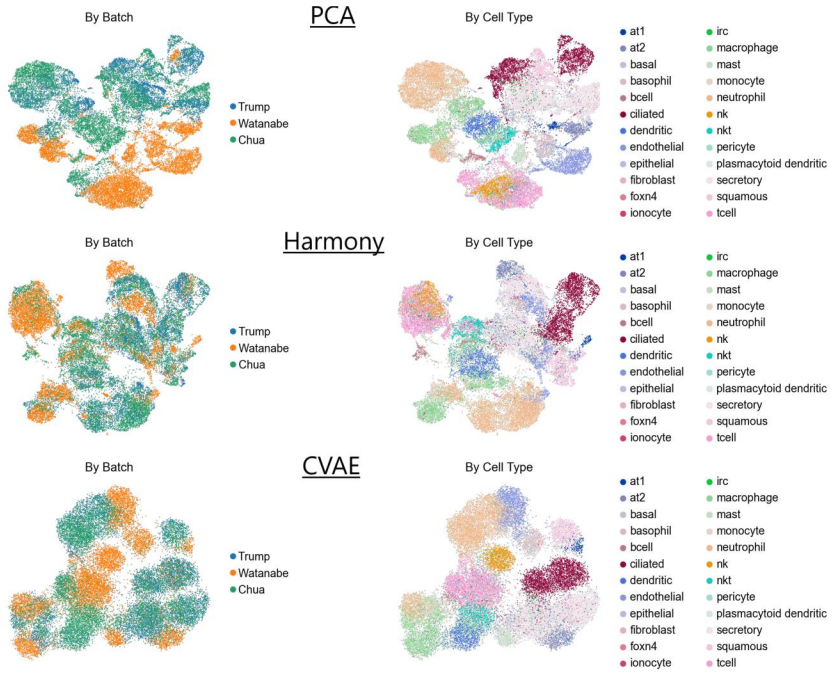

Figure 5: UMAP (McInnes and Healy 2018) of the latents. PCA (top), Harmony (middle), CVAE (bottom)

The cell-type prediction results (right sub-table of Table 1) are reflective of the data integration metrics. Consistently across all predictive models, the best test accuracy is achieved when the PCA representations are used as inputs, followed by the CVAE and then Harmony. This is rationalised by the fact that PCA was found to preserve the most biological variation. We would expect the advantage of PCA to dissipate as the pipeline is scaled to more studies, so batch correction becomes more important. Additionally, the deep learning predictive model outperforms the Softmax and SVM on the test set for the PCA and CVAE representations. We expect the deep learning model to be the biggest beneficiary from the addition of more studies to the pipeline.

6 Conclusion/Future Work

In this study we developed and benchmarked a universal deep learning pipeline for single cell RNA prediction models. The pipeline shows improved batch correction vs non-deep learning batch correction methods and can be easily applied to new studies via transfer learning. Future work includes training of the pipeline on a larger number of studies to establish a full reference baseline, and the application of the prediction models to other biological and medical tasks, e.g., disease prediction or the study of regulatory networks under perturbations.

7 Contributions

Both authors contributed equally to the conceptualization and writeup. H.S. implemented the CVAE. S.K. implemented the data extraction/processing and deep learning prediction model.

References

Chua, R.L. et al. (2020). "COVID-19 severity correlates with airway epithelium–immune cell interactions identified by single-cell analysis". In: *Nature Biotechnology*.

Hwang, B. et al. (2018). "Single-cell RNA sequencing technologies and bioinformatics pipelines". In: *Experimental and Molecular Medicine*.

Kingma, P and M Welling (2014). "Auto-encoding variational Bayes". In: *Oral presentation at the International Conference on Learning Representations*.

Korsunsky, I et al. (2019). "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nature Methods*.

Li, Liam et al. (2020). "A System for Massively Parallel Hyperparameter Tuning". In: *Third Conference on Systems and Machine Learning*.

Lopez, R et al. (2018). "Deep generative modeling for single-cell transcriptomics". In: *Nature Methods*.

Lotfollahi, M et al. (2022). "Mapping single-cell data to reference atlases by transfer learning". In: *Nature Biotechnology*.

Lotfollahi, M. et al. (2020). "Conditional out-of-distribution generation for unpaired data using transfer VAE". In: *Bioinformatics*.

Love, I et al. (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome. Biol.*

Luecken, M et al. (2021). "Benchmarking atlas-level data integration in single-cell genomics". In: *Nature Methods*.

McInnes, L and J Healy (2018). "Uniform Manifold Approximation and Projection for Dimension Reduction". In: *ArXiv e-prints*.

Regev, A et al. (2017). "The Human Cell Atlas". In: *Elife*.

Tran, H et al. (2020). "A benchmark of batch-effect correction methods for single-cell RNA sequencing data". In: *Genome Biology*.

Trump, S. et al. (2021). "Hypertension delays viral clearance and exacerbates airway hyperinflammation in patients with COVID-19". In: *Nature Biotechnology*.

Watanabe, Na. et al. (2020). "Single-cell Transcriptome Analysis Reveals an Anomalous Epithelial Variation and Ectopic Inflammatory Response in Chronic Obstructive Pulmonary Disease". In: *medRxiv*.

Wolf, F et al. (2018). "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome Biology*.