Deep generative model for harmonizing time-series scRNA-seq data

Dongshunyi (Dora) Li [1] **Hyun Woong Kim** [1]

1. Introduction

Starting from progenitor cells, cells can develop into many different types. For example, lung progenitor cells can develop into epithelial cells, muscle cells or immune cells. Various methods have been developed to reconstruct cell trajectory with time-series single cell RNA-seq (scRNA-seq) data. A popular group of probabilistic methods, for example (Ding et al., 2018), use Kalman filter or variants of Hidden Markov Models (HMM) to infer the hidden cell states at different differentiation stage. Figure 1 shows an example output, where different cell types are assigned to different states through the time course. However, these methods, originally developed to deal with data from a single study, perform poorly when applied to data mixed from different studies. Due to the cost of performing scRNA-seq experiments, a study usually only samples 1-3 time points, each with a limited cell population, during cell development. To gain a more comprehensive view of cell trajectory and to utilize information across studies, methods need to be developed to reconstruct trajectory and being robust against variations from experiments preparation and cell intrinsic biological properties (e.g., cell size, cell cycle and stochasticity in expression).

Methods specializing in integrating scRNA-seq data from different studies have been recently developed. (Lopez et al., 2018) developed a generative model explicitly accounting for different sources of variations. For each cell n, they model the latent biological variance as z_n whereas have s_n for technical confounder (e.g., different studies) and l_n for sequencing library size. The variational posterior $q(z_n|x_n, s_n)$ can be used for downstream analysis such as cell trajectory reconstruction. This method performs reasonably well for cell types having thousands of observations and sequenced at a single time point (i.e. major type), where the learned latent biological variance z_n does correlate with s_n but correlate with known cell labels. However, when

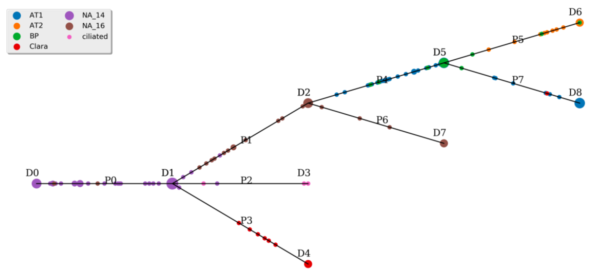

Figure 1. An example output of cell trajectory reconstruction methods. Dots are cells colored by cell labels. Figure courtesy to (Lin & Bar-Joseph, 2018)

applied to cell types sequenced with a limited sample size (e.g., 100-200 observations per cell), the method output z_n still strongly correlate with s_n whereas not correlate with known cell labels. Also, the z_n for these rare cell types are mixed and cannot be easily separated. Possible reasons accounting for this include the limited sample size of each cell type and the relatively higher variance among cells from different types or different developmental stage. Unfortunately, given the high expense of biological study, a study typically only samples 1-3 time points, only focus on a few particular cell types and usually have a limited sample size for each cell type, where current integration methods such as (Lopez et al., 2018) does not work well.

Another recently developed method (Wang et al., 2018) aims at learning the latent biological variance from the noisy scRNA-seq data by initializing auto-encoders with weights pre-trained by cells samples from the repository of Human Cell Atlas. By transferring the biological relevant knowledge from other studies, they succeeded in recovering the latent structure of some immune cells even with a very limited sample size present in a single study. Inspired by their finding, we can possibly improve the performance of aforementioned (Lopez et al., 2018) method in the time-series scRNA-seq scenario by utilizing other biological relevant information for learning the latent z_n.

In time-series scenario, time dependent information from previous time points will be inherited by the current time point, which results in auto-correlation and Granger causality. This gives the natural intuition of using the time depen-

[1]Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania. Correspondence to: Dongshunyi (Dora) Li <dongshul@andrew.cmu.edu>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

dent information from previous latent variables to help with inferring the time dependent variables at current time. This is especially suitable for cell trajectory inference where time dependent information is mostly related to biological variance whereas distinct technical variance occurs randomly at a few time points.

Here we propose a robust generative model for inferring time-dependent biological latent variables from time-series scRNA-seq data. We model the observations at different time steps similarly in (Lopez et al., 2018) and chain the time dependent latent variables by a deep Kalman filter. The latent posterior variables can be readily applied to downstream cell trajectory analysis. To further constrain the model to focus on biological time dependent variance, we propose to utilize some prior biology knowledge. For example, it is well known that one of the underlying driven mechanism of cell development is gene regulation. In methods such as (Ding et al., 2018), regulatory elements expressed at a time point are used to infer the expression of genes in a later time point, given the regulatory elements interact with the genes. The incorporation of this type of information not only facilitates the learning of biological relevant variations but also provides possibilities in inferring the regulatory mechanism behind the cell development. Knowing how the regulation changes over the development time and how it shapes cell identity enables biologists to engineer the progenitor cells to a specific destiny.

2. Related Work

The literature on time series model is vast. The most relevant one to our model is (Krishnan et al., 2017) where the authors propose a general framework by applying variational autoencoder (Kingma & Welling, 2013) to the classic Kalman filter model. The authors propose to use neural networks in place of linear transformations between variables. In their general framework, both the time dependent latent variable and the observation are represented by a single variable as in the case of classic Kalman filter. In our model, we also have a single latent biological variable for the time-dependent latent variable whereas we have a much finer representation of the observations. We disentangle it with several variables as inspired by (Lopez et al., 2018).

Various methods specializing in integrating scRNA-seq data across studies have been recently developed. A main group of methods are based on heuristic methods which mostly include low-dimensional representation and a non-linear alignment such as (Butler et al., 2018) and (Haghverdi et al., 2018). These methods do not provide probabilistic interpretation as well as flexibility in tuning parameter to avoid potential over-aligning. (Lopez et al., 2018) and its extension (Xu et al., 2019) use a generative Bayesian hierarchical model which enable tuning and probabilistic interpretation. Our model parameterizes the observation at each time point similarly as in (Lopez et al., 2018). (Xu et al., 2019) further extends (Lopez et al., 2018) by using available cell type labels to infer the latent biological variable. Both of the methods do not consider utilizing time dependent information.

Cell trajectory construction methods can be largely grouped by the framework they use (probabilistic vs deterministic) and the representation they provide (continuous vs. discrete cell assignment). Deterministic methods usually starts with a dimension reduction of the expression data, followed by a graph analysis or Guassian Process(GP) to connect cells. Probabilistic methods are mostly based on probabilistic graphical models. Among them, the one most related to our model is (Ding et al., 2018), where the authors apply Kalman filter to infer a rooted tree cell trajectory. This method uses prior knowledge on regulatory elements and gene interaction, which is also considered in our model. However, they model the observation as a single variable with zero-inflated Gaussian distribution. Also, they use linear transformations between variables.

3. Model

We present the model to harmonize time-series scRNA-seq data by enabling time-dependent passage passing with deep Kalman filter and explicitly parameterizing the distribution of observed data as suggested in (Lopez et al., 2018). We primarily focus on cell development trajectory construction and thus, the time point in our scenario is the differential stage. As Figure 2 shows, specifically, for each cell $n \in N$, we have a vector of latent variables $\vec{z}$ standing for its low-dimensional embeddings for a consecutive observed time points. We account for confounding factor with s_{tn} and the RNA library size, which is a physical property of cells, with l_{tn}. Note we have a confounder and a library size for each cell specifically for each time point. Because we may have different experimental measurements at different time points. Both confounder s_{tn} and library size l_{tn} is affected by experimental conditions.

For each gene $g \in G$ at time t, we model its dispersion as θ_{tg}. Dispersion for a gene may change over time. This is well supported by the literature that the heterogeneity of genes across cells is largely associated with the development process. Genes with high dispersion at early cell stages are different from those in late cell stages but stages closed to each other tend to share the most dispersed genes. This parameter is shared by all cell type. For each cell n and each gene g, at time t, we have w_{tng} as the mean expression value of the gene from the cell and y_{tng} as the expression value draw from a distribution with w_{tng} as the prior mean. h_{tng} is the drop-out rate. Drop-out here means the event that one observes 0 in the dataset. It is well studied that drop-

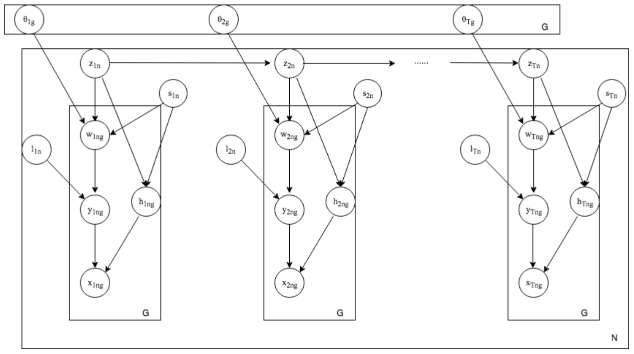

Figure 2. Graphical representation for the proposed model

out may come from both biological events and confounders such as experimental errors. Therefore, here we have h_{tng} depend on both z_{tn} and s_n. Finally, x_{tng} is the observed expression value of gene g of cell n at time t.

Similar as (Krishnan et al., 2017), we model the latent time dependent states z_t as Gaussian distributions with mean and variance approximated by neural networks with previous latent states as input. This is in line with biological applications where low-dimensional embeddings are usually assumed following Gaussian distributions. We also assume a Gamma-Poisson conjugate for negative binomial distributed expression values, as in (Lopez et al., 2018) and (Wang et al., 2018). We also have our drop-out indicator distributed as Bernoulli.

The detailed parameterization is listed as follows:

$$z_{1n} \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{1}$$

$$z_{tn} \sim \mathcal{N}(f_{za}(z_{(t-1)n}), f_{zb}(z_{(t-1)n})) \tag{2}$$

$$l_{tn} \sim \text{log normal}(l_\mu, l_\sigma^2) \tag{3}$$

$$w_{tng} \sim \text{Gamma}(f_w(z_{tn}, s_{tn}), \theta_{tg}) \tag{4}$$

$$y_{tng} \sim \text{Poisson}(l_{tn} w_{tng}) \tag{5}$$

$$h_{tng} \sim \text{Bernoulli}(f_h^g(z_{tn}, s_{tn})) \tag{6}$$

$$x_{tng} = \begin{cases} 0, & \text{otherwise} \\ y_{tng}, & \text{if } h_{tng} = 0 \end{cases} \tag{7}$$

For each differentiation stage t, we model the observation of each gene of each cell x_{tng} as an i.i.d random variable from the generative process. The generative process for time t is illustrated as follows. Note l_μ, l_σ^2 are cell specific prior estimated by the empirical statistics and f_{za}, f_{zb},f_w, f_h are all neural networks.

1. for cell $n \in N$
   (a) for time t
      i. Draw a latent presentation of this cell $z_{tn} \sim \mathcal{N}(f_{za}(z_{(t-1)n}), f_{zb}(z_{(t-1)n}))$
      ii. Choose a cell-scaling factor $l_{tn} \sim Log\mathcal{N}(l_\mu, l_\sigma^2)$
      iii. for gene $g \in G$ do
         A. choose an expression mean $w_{tng} \sim$ Gamma$(f_w(z_{tn}, s_{tn}), \theta_{tg})$
         B. choose a expression value $y_{tng} \sim$ Poisson$(l_{tn} w_{tng})$
         C. choose a dropout indicator $h_{tng} \sim$ Bernoulli$(f_h^g(z_{tn}, s_{tn}))$
         D. if dropout, then output $x_{tng} = 0$, else output y_{tng}

3.1. Derivation of Evidence Lower Bound (ELBO)

We use evidence lower bound(ELBO) of the conditional log likelihood $\log p_\theta(\vec{x} \mid \vec{s})$ as our optimization objective. Here $\vec{x}$ is the observations of all time points for this particular cell. $\vec{s}$ is the corresponding confounding factors for each time point. $\vec{l}$ is the library size. For a single cell $n \in N$, we have:

$$\begin{aligned} \log p_\theta(\vec{x} \mid \vec{s}) \geq\ & \mathrm{E}_{q(\vec{z},\vec{l}|\vec{x},\vec{s})}(\log p_\theta(\vec{x} \mid \vec{z}, \vec{l}, \vec{s})) \\ & - KL(q(\vec{z}, \vec{l} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{z}, \vec{l} \mid \vec{s})) \end{aligned}$$

The reconstruction loss $\mathrm{E}_{q(\vec{z},\vec{l}|\vec{x},\vec{s})}(\log p_\theta(\vec{x} \mid \vec{z}, \vec{l}, \vec{s}))$ can be further decomposed by using Markov dependencies:

$$\mathrm{E}_{q(\vec{z},\vec{l}|\vec{x},\vec{s})}(\log p_\theta(\vec{x} \mid \vec{z}, \vec{l}, \vec{s})) = \sum_{t=1}^{T} \mathrm{E}_{q(z_t,l_t|\vec{x},\vec{s})}(\log p_\theta(x_t|z_t, l_t, s_t))$$

We apply mean-field for $\vec{l}$ and $\vec{z}$ such that:

$$q(\vec{z}, \vec{l} \mid \vec{x}, \vec{s}) = q(\vec{z} \mid \vec{x}, \vec{s}) q(\vec{l} \mid \vec{x}, \vec{s})$$

Thus, for the KL divergence $KL(q(\vec{z}, \vec{l} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{z}, \vec{l} \mid \vec{s}))$, we have:

$$\begin{aligned} KL(q(\vec{z}, \vec{l} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{z}, \vec{l} \mid \vec{s})) =\ & KL(q(\vec{z} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{z} \mid \vec{s})) \\ & + KL(q(\vec{l} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{l} \mid \vec{s})) \end{aligned}$$

The term on time-dependent $\vec{z}$ can be further decomposed using Markov dependencies:

$$\begin{aligned} KL(q(\vec{z} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{z} \mid \vec{s})) =\ & KL(q(z_1|\vec{x}, \vec{s}) \parallel p_0(z_1|\vec{s})) \\ & + \sum_{t=2}^{T} \mathrm{E}_{q(z_{t-1}|\vec{x},\vec{s})}(KL(q(z_t|z_{t-1}, \vec{x}, \vec{s}) \parallel p(z_t \mid \end{aligned}$$

The term on $\vec{l}$ can be further decomposed using mean-field:

$$KL(q(\vec{l} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{l} \mid \vec{s})) = \sum_{t=1}^{T} KL(q(l_t|x_t, \vec{s}) \parallel p(l_t|\vec{s}))$$

3.2. Generative and recognition networks

The generative networks are composed of a emission network, i.e. $f_w(z_{tn}, s_{tn})$ and $f_h^g(z_{tn}, s_{tn})$ and a gated transition network, i.e. $f_{za}(z_{(t-1)n})$,$f_{fb}(z_{(t-1)n}))$. The emission network contains fully connected layers with a single hidden layer of 128 nodes and linear layers projected to the parameters space. s_{tn} are fitted as indicators as the input for the emission network. According to (Krishnan et al., 2017), the gated transition allows the flexibility for some dimension to have linear transformation. The gated transition contains a fully connected layer for the proposed mean, one with sigmoid for the gate, and linear ones for the parameter spaces.

The recognition networks are for inferring $q(z_t \mid z_{t-1}, \vec{x}, \vec{s})$ and $q(\vec{l} \mid \vec{x}, \vec{s})$. They are parameterized as follows:

$$q(z_t \mid z_{t-1}, \vec{x}, \vec{s}) \sim \mathcal{N}(g_{za}(z_{t-1}, \vec{x}), g_{zb}(z_{t-1}, \vec{x})) \quad (8)$$

$$q(\vec{l} \mid \vec{x}, \vec{s}) \sim \mathcal{N}(g_{la}(\vec{x}), g_{lb}(\vec{x})) \quad (9)$$

As suggested in (Krishnan et al., 2017), when inferring z_t, both observations $\vec{x}$ and the previous z_{t-1} should be taken in to account. Thus, we also use a similar 'combiner' strategy, where we combine the output from the backward RNN and a non-linear transformation of z_{t-1} to infer the parameters of the distribution of $q(z_t \mid z_{t-1}, \vec{x}, \vec{s})$. This is illustrated in Figure 3. Backward RNN suits our problem because biological data does not have many time points. We usually only have 3-10 time points. Thus, we don't need complicated architecture such as LSTM. Also, given the purpose of our model is to infer the latent states, it is desirable to use future information which gives better empirical results as suggested by many literature.

3.3. Incorporation of biological priors

In order to learn biologically relevant time dependent transition model and provide possibilities in inferring the regulatory mechanism behind the cell development we can apply some biological prior knowledge. For example, as suggested by (Ding et al., 2018), transcription factors' regulation effect is time dependent. Specifically, genes which are the targets of co-expressing transcription factors at time $t-1$ will be up or down regulated at time t. Thus, a simply linear model of the transition, as done in (Ding et al., 2018) is simply $z_t = z_{t-1} + B$ where B is the overall effect of the transcription factors. Another approach to incorporate

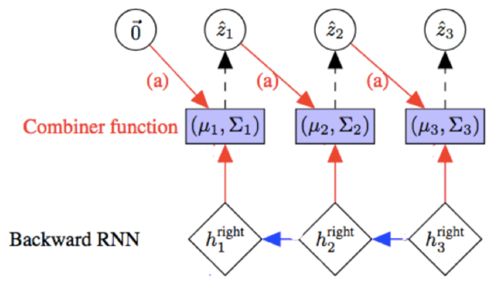

Figure 3. Illustration of the recognition network. Figure adapted from (Krishnan et al., 2017)

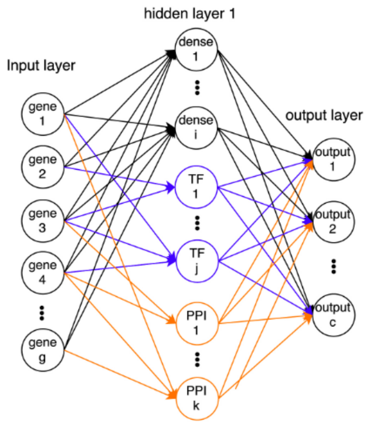

Figure 4. Illustration of incorporating biological prior in NN. Figure courtesy to (Lin et al., 2017)

biological relevant information and to enable better interpretation is to specifically design the hidden nodes in the neural networks. For example, as suggested in (Lin et al., 2017), one can use hidden nodes as transcription factors which only connect with those genes in the input layer if it regulates the genes. The hidden nodes can also be a protein-protein interaction network(PPI) where only the genes expressing proteins in the network will be connected. This is illustrated in the Figure 4.

4. Inference by stochastic backpropagation

We applied a similar strategy of optimization as in (Krishnan et al., 2017). We perform gradient ascent of the ELBO and use stochastic backpropagation to obtain Monte Carlo estimates of the true gradient (Kingma & Welling, 2013). Specifically, we implemented the training process as in Algorithm 1.

4.1. Optimization challenges

Given that we have a hierarchy of latent variables $\vec{z}$, our model may subject to a few well-known optimiza-

Algorithm 1 Learn with stochasitc gradient descent

```
0: initialize $\theta$ and $\phi$ for $p$ and $q$
1: while not converged do
1:     $X^M \leftarrow$ Random mini-batch of M data points
1:     $H^M \leftarrow$ RNN output of M data points
1:     initialize $p(z_0)$ and $q(z_0)$
2:     for $t \in T$ do
2:         estimate posterior parameters of $z_t$ and sample $z_t \sim q(z_t \mid z_{t-1}, H^M, s_t)$
2:         estimate transition parameters of $p(z_t \mid z_{t-1})$
2:         estimate posterior parameters of $l_t$ and sample $l_t \sim q(l_t \mid X^M, s_t)$
2:         estimate emission parameters of $p(x_t|z_t, l_t, s_t)$
2:         estimate KL divergences and reconstruction likelihood
3:     end for
3:     $g \leftarrow \nabla_{\theta,\phi}\mathcal{L}(X^M; (\theta, \phi))$
3:     $\theta, \phi \leftarrow$ update parameters with gradients using Adam (Kingma & Ba, 2014)
4: end while=0
```

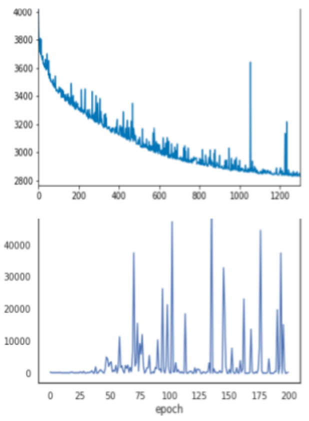

Figure 5. Compare variance in training loss of different training strategies.

tion challenges of "deep" variational autoencoders. The first challenge is the high variance in estimates and gradients, especially in estimating the KL divergence $KL(q(\vec{z} \mid \vec{x}, \vec{s}) \parallel p_0(\vec{z} \mid \vec{s}))$. We first tried to directly sample from this KL term and obtain the Monte Carlo estimate. However, this end up in large variance in loss and sometimes renders the parameters invalid for some distributions. Thus, we decided to derive the decomposed KL divergence term each of which has an analytic form. This approach largely decreased the variance. In Figure 5, the upper plot is the training loss for decomposed KL terms whereas the lower plot is the training loss using Monte Carlo sampling.

As pointed out in (Sø nderby et al., 2016), another challenge in optimization is the over-regularization effect by KL terms. We applied the ladder VAE suggested by (Sø nderby et al., 2016) by a small modification of the training procedure. Basically, instead of directly using the posterior parameters for z_t, we take the average between the estimates of recognition parameters μ_ϕ and Σ_ϕ and the estimates of generative parameters μ_θ and Σ_θ, weighted by Σ_ϕ and Σ_θ. According to the authors, this enables the interplay of bottom-up observation information and top-down prior information. The authors also suggest warming-up period where the KL terms are multiplied by a scaling factor $beta$ which is gradually increased from a value smaller than 1 (e.g. 0.1) to 1. We applied ladder VAE but according to the latent space segregation. The results are not good as the usual training process. Also, we only applied the warming-up trick for 20 epochs. One possible reason accounting for the unexpected down performance is that the datasets we used for experiments only have 3 time points and thus not might not need the tricks for deep VAE.

5. Experiments

Given that the objective of our model is to learn the time dependent latent embeddings of observed scRNA-seq data, we would like to mainly focus on experimenting with real scRNA-seq datasets and evaluating if the latent variables learned from our model 1) alleviate the effect of confounders 2) preserve the biological identity of cells. Confounders in scRNA-seq experiment scenario are mostly different 'batches'. (Note batch here is different from its meaning in 'mini-batch'). A batch normally means a different experimental condition. Biological identity is most straight forward associated with cell labels (i.e. cell types).

Due to the high expense and technical difficulties in conducting scRNA-seq experiments, high quality datasets with relative large sample size is very rare. Most datasets have sample size of about 50-200 cells. Due to this limitation, at current stage, we will only focus on 2 datasets as listed below (Table 1). However, each dataset contains a vast diversty of cell types and thus provide enough information to test the performance of our model.

Table 1. Description of the datasets

ORIGINAL STUDY	TIME POINT	NO. OF CELLS
(PLASSCHAERT ET AL., 2018)	ADULT	7898
(MONTORO ET AL., 2018)	ADULT	7193

In the following sections, we will compare and contrast the performance of our model, which will be denoted as 'DMM'

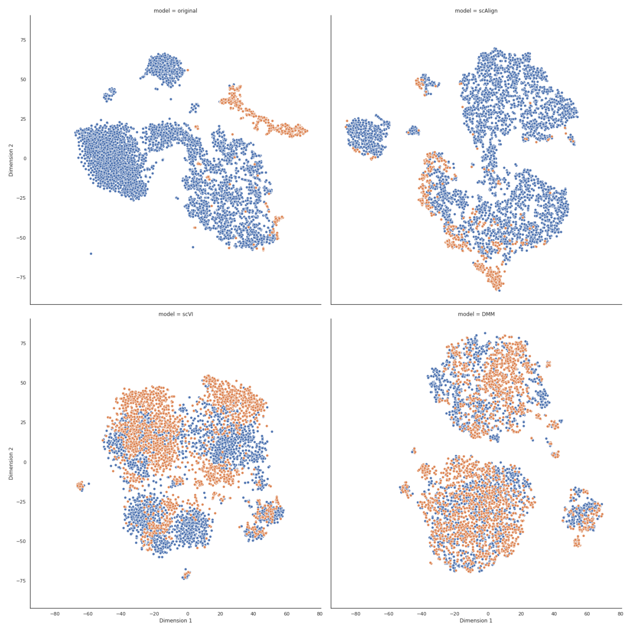
Figure 6. Cell color by batch. The more mixed the better the performance. Top left: original data space, top right: scAlign, bottom left: scVI, bottom right: DMM (our model)

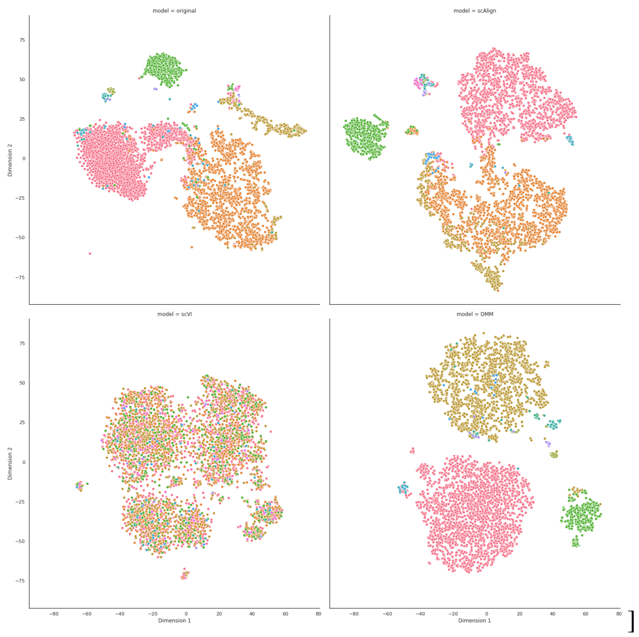
Figure 7. Cell color by cell type. The less mixed the better the performance. Top left: original data space, top right: scAlign, bottom left: scVI, bottom right: DMM (our model)

for simplification, and other 2 state-of-art methods (Lopez et al., 2018) and (Johansen & Quon, 2019). Throughout the experiments, we use a learning rate of 1e-3 and a weight-decay of 1e-3.

5.1. Visualization of latent space

As Figure 6 and 7 shows, our model(right-down) performs the best in that compared to the other models, it mixed the batches such as the counfounder's effect is removed and also keeping the identity of each cells, as can be seen from the the different colors and well-separated clusters in the t-sne plot.

5.2. Batch Entropy Mixing

Now that we have the latent space of scRNA-seq, we can test our embedding. We used two metrics to evaluate our model. Both of these metrics were introduced in (Lopez et al., 2018).

The first metric we used measured the batch effect removing power of the model. It computes the entropy of the batch labels for k nearest neighbors for 100 cells (k = 10,20,,100). We repeated such sampling 50 times, and used the mean entropy as our metric. The higher the metric, the stronger power of the batch effect removing power of the model. Note that since in the original data batch label separates the cells, if the metric is close to zero then the model failed to remove the batch effect.

We computed this metric for latent spaces of the original dataset, scAlign, scVI, and our model (DMM). For original gene space and scAlign (which outputs the re-projected space instead of its latent space) we used PCA and used 30 PCs as their latent space. Our model outperformed recent methods (Figure 8).

5.3. K-nearest neighbors purity

The model has to remove the batch effects while preserving cell identity. For the cells to be classified correctly, the cells that are close in the original data space (where cells with the same labels would be close to each other) also have to be close in the latent space of the model. Thus we used a metric that measures the similarity of the k-nearest neighbors graph of the two spaces. For each method, we computed the Jaccard index between the k-nearest neighbors graph matrix of the latent space of one batch and the k-nearest graph matrix of the latent space of that batch in the original data. Then our metric termed KNN-purity is defined as the average Jaccard index of the two batches. Again we used the latent space with rank 30 obtained by PCA for the original data space and scAlign.

Note that high KNN-purity means that the proximity between similar cells are preserved. As expected, the latent space of the original data showed the highest KNN-purity (Figure 9). Our model showed similar performance to

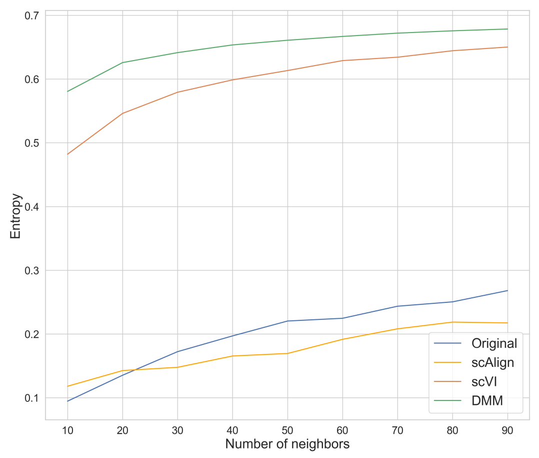

Figure 8. Batch entropy mixing of latent space of original data and 3 models. For the latent space of the original data and scAlign, we used 30 principal components.

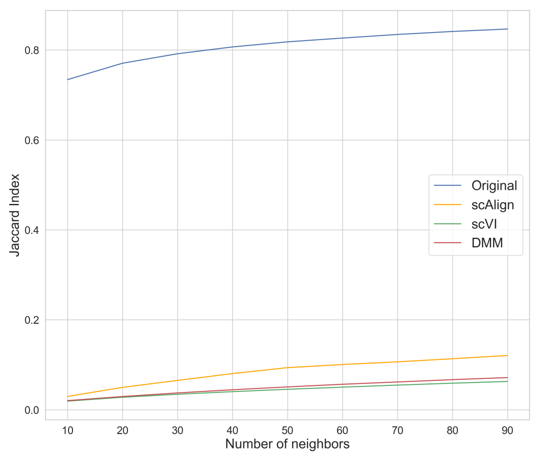

Figure 9. KNN-purity of latent space of original data and 3 models. For the latent space of the original data and scAlign, we used 30 principal components.

scVI (Lopez et al., 2018). scAlign showed the highest KNN-purity among the three models.

5.4. Cell label classifier using neural networks

To further test the latent space from our model, we tried to build a cell label classifier. Single cell RNA-seq doesn't give the true label of the cell, but researchers can infer the cell type based on the gene expression profile. For the datasets we used the group that published it provided with us with cell labels based on their classifier. We used these labels as ground truth to train our model.

The model was a simple fully connected network with 3 hidden layers of size 500 each. We used ReLU as the activation function for the three layers. The final layer was connected to the output layer whose size is the same as the number of different cell classes. This layer was followed with softmax activation for classification. We used cross-entropy loss as the loss function and trained the model with Adam optimizer (Kingma & Ba, 2014). We trained all of classifers with 10 epochs each. The model was implemented using Keras (Chollet et al., 2015).

As inputs we used the original gene space, 30 PCs of that space, reprojected space of scAlign, 30 PCs of that space, the latent space of scVI, and the latent space of our model (Figure 10). For evaluation we used 3-fold cross-validation and reported the training and test accuracy. For splitting the dataset we used stratified split that preserves the ratio of class labels using the `sklearn.model_selection.StratifiedKFold` function from the sklearn package (Pedregosa et al., 2011). Cell labels that consisted less than 1% of the entire dataset were considered as 'rare' cell types, and we computed the accuracy of prediction of these rare cell types for our test dataset. Results show that our model show high test accuracy that is comparable to the results obtained using the original data space (Figure 11). Moreover, our model showed good rare cell type classification accuracy. The latent space of the original data space using PCA (original_PC30) failed to classify rare cell types (average accuracy = 0.116). Our model achieved 0.516 accuracy that was higher than scAlign but lower than scVI.

One thing we noticed was that the two groups that provided the two datasets used different labels for the same cell type. For example, a group used 'Tuft' and the other group used 'Brush' for the same cell type. Other examples include 'PNEC' and 'Neuroendocrine' and 'Secretory' and 'Club'. We strongly recommend future users to be aware about this when merging the labels used in different experiments.

6. Conclusion and future work

We developed a deep generative model for inferring time-dependent biological latent variables from time-series scRNA-seq data. The latent variables obtained with our model can be used for various applications such as cell trajectory analysis during development or inferring cell labels.

Prior biological info has shown to improve learning, e.g.,

Dataset	number of datapoints	number of features
original	15091	16177
original_PC30	15091	30
scAlign	8999	4796
scAlign_PC30	8999	30
scVI	15091	30
dmm	8990	30

Figure 10. Description of the different datasets used to train the cell label classifier. The PC30 datasets are the first 30 PCs of the corresponding original dataset.

Dataset	Training accuracy	Test accuracy	Rare cell type accuracy
	0.992	0.974	0.783
original	0.995	0.948	0.547
	0.998	0.920	0.613
3-fold CV	**0.995**	**0.948**	**0.648**
	0.942	0.947	0.144
original_PC30	0.942	0.924	0.055
	0.957	0.902	0.148
3-fold CV	**0.947**	**0.924**	**0.116**
	0.951	0.959	0.521
scAlign	0.951	0.956	0.271
	0.959	0.939	0.594
3-fold CV	**0.954**	**0.951**	**0.462**
	0.959	0.960	0.274
scAlign_PC30	0.960	0.959	0.229
	0.969	0.942	0.362
3-fold CV	**0.963**	**0.954**	**0.288**
	0.987	0.957	0.682
scVI	0.990	0.959	0.766
	0.988	0.961	0.774
3-fold CV	**0.988**	**0.959**	**0.741**
	0.985	0.986	0.514
dmm	0.983	0.988	0.583
	0.985	0.977	0.451
3-fold CV	**0.984**	**0.984**	**0.516**

Figure 11. Training accuracy, test accuracy, and the rare cell type accuracy (in test dataset) of the cell label classifier trained with different datasets.

regulators act on different time stages. To incorporate this, we may consider either change the RNN structure such as regulators added as hidden nodes or add time dependent intervention variables.

References

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411, 2018.

Chollet, F. et al. Keras. `https://keras.io`, 2015.

Ding, J., Aronow, B. J., Kaminski, N., Kitzmiller, J., Whitsett, J. A., and Bar-Joseph, Z. Reconstructing differentiation networks and their regulation from time series single-cell expression data. *Genome research*, 28(3):383–395, 2018.

Haghverdi, L., Lun, A. T., Morgan, M. D., and Marioni, J. C. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421, 2018.

Johansen, N. and Quon, G. scalign: a tool for alignment, integration and rare cell identification from scrna-seq data. *bioRxiv*, pp. 504944, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Lin, C. and Bar-Joseph, Z. Continuous state hmms for modeling time series single cell rna-seq data. *bioRxiv*, pp. 380568, 2018.

Lin, C., Jain, S., Kim, H., and Bar-Joseph, Z. Using neural networks for reducing the dimensions of single-cell rna-seq data. *Nucleic acids research*, 45(17):e156–e156, 2017.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053, 2018.

Montoro, D. T., Haber, A. L., Biton, M., Vinarsky, V., Lin, B., Birket, S. E., Yuan, F., Chen, S., Leung, H. M., Villoria, J., Rogel, N., Burgin, G., Tsankov, A. M., Waghray, A., Slyper, M., Waldman, J., Nguyen, L., Dionne, D., Rozenblatt-Rosen, O., Tata, P. R., Mou, H., Shivaraju, M., Bihler, H., Mense, M., Tearney, G. J., Rowe, S. M., Engelhardt, J. F., Regev, A., and Rajagopal, J. A revised airway epithelial hierarchy includes cftr-expressing ionocytes. *Nature*, 560(7718):319–324, 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0393-7. URL https://doi.org/10.1038/s41586-018-0393-7.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Plasschaert, L. W., Zilionis, R., Choo-Wing, R., Savova, V., Knehr, J., Roma, G., Klein, A. M., and Jaffe, A. B. A single-cell atlas of the airway epithelium reveals the cftr-rich pulmonary ionocyte. *Nature*, 560 (7718):377–381, 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0394-6. URL https://doi.org/10.1038/s41586-018-0394-6.

Sø nderby, C. K., Raiko, T., Maalø e, L., Sø nderby, S. r. K., and Winther, O. Ladder variational autoencoders. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3738–3746. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6275-ladder-variational-autoencoders.pdf.

Wang, J., Agarwal, D., Huang, M., Hu, G., Zhou, Z., Conley, V. B., MacMullan, H., and Zhang, N. R. Transfer learning in single-cell transcriptomics improves data denoising and pattern discovery. *bioRxiv*, pp. 457879, 2018.

Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M. I., and Yosef, N. Harmonization and annotation of single-cell transcriptomics data with deep generative models. 2019.