

# RVAgene: Generative modeling of gene expression time series data

Raktim Mitra<sup>1</sup>, Adam L. MacLean<sup>1,\*</sup>

<sup>1</sup>Quantitative and Computational Biology, University of Southern California

\*macleana@usc.edu

## Abstract

Methods to model dynamic changes in gene expression at a genome-wide level are not currently sufficient for large (temporally rich or single-cell) datasets. Variational autoencoders offer means to characterize large datasets and have been used effectively to characterize features of single-cell datasets. Here we extend these methods for use with gene expression time series data. We present RVAgene: a recurrent variational autoencoder to model gene expression dynamics. RVAgene learns to accurately and efficiently reconstruct temporal gene profiles. It also learns a low dimensional representation of the data via a recurrent encoder network that can be used for biological feature discovery, and can generate new gene expression data by sampling from the latent space. We test RVAgene on simulated and real biological datasets, including embryonic stem cell differentiation and kidney injury response dynamics. In all cases, RVAgene accurately reconstructed complex gene expression temporal profiles. Via cross validation, we show that a low-error latent space representation can be learnt using only a fraction of the data. Through clustering and gene ontology term enrichment analysis on the latent space, we demonstrate the potential of RVAgene for unsupervised discovery. In particular, RVAgene identifies new programs of shared gene regulation of *Lox* family genes in response to kidney injury.

## 1 Introduction

Dynamic changes in gene expression control the transcriptional state of a cell, and are responsible for modulating cellular states and fates. Gene expression dynamics are in turn controlled by cell-internal and external signaling networks. Despite the noisiness of gene expression in single cells (Raj & Van Oudenaarden 2008), over time or over populations of cells, predictable patterns emerge. Here we address the challenge of classifying and predicting gene expression dynamics across large groups of genes.

Machine learning (and deep learning in particular) has led to recent advances in our ability to explain or predict biological phenomena (Ching et al. 2018). Deep learning modeling via autoencoders (Hinton & Salakhutdinov 2006) and variational autoencoders (Kingma & Welling 2014) has been central to progress in the field. Autoencoders learn two functions: one to encode each input data point to a low dimensional point, and another (the decoder) to reconstruct the original data point from the low dimensional representation. Variational autoencoders (VAEs) build on this architecture and instead encode input data points as distributions; VAEs are less prone to overfitting and can offer meaningful representations of biological features in the latent space (Way & Greene 2017).

Single-cell mRNA sequencing (scRNA-seq) data present appealing sources of data for deep learning models, given their size and complexity (Svensson et al. 2018). Deep learning models have been used to analyze scRNA-seq data and address a variety of challenges. Autoencoders have been developed to perform noise removal/batch correction (Deng et al. 2019, Eraslan et al. 2019, Wang et al. 2019), imputation (Talwar et al. 2018), and visualization & clustering (Lin et al. 2017). VAEs have been developed for the visualization and clustering of scRNA-seq data (Ding et al. 2018, Wang & Gu 2018), and can provide a broad framework for generative modeling of scRNA-seq data (Lopez et al. 2018): scVI can be used for batch correction, clustering, visualization, and differential expression testing.

The methods described above for single-cell data analysis by deep learning focus primarily on cell-centric tasks; here we are interested in gene-centric inference. Particularly, we are interested in characterizing dynamic changes in gene expression. These can be either changes with respect to real time or “pseudotime,” the latter

referring to the ordering of single cells along an axis describing a dynamic cell process such as development or stem cell differentiation (see methods overview in (Saelens et al. 2019)). We can interpret any scRNA-seq data as gene expression time series data, given an appropriate underlying temporal process, either in terms of real (experimental) time (low resolution: around 2 – 20 data points) or pseudotime (high resolution:  $10^3$  –  $10^6$  data points). McDowell et al. (2018) introduced a non-parametric hierarchical Bayesian method (DPGP) to model such data. Using a Gaussian process to cluster temporal gene profiles and a Dirichlet process to generate the Gaussian processes, DPGP offers powerful and intuitive means with which to cluster gene expression time series data. However, since learning Gaussian processes is equivalent to a fully agnostic search in function space, training DPGP is computationally intensive and difficult to parallelize.

Clustering relies on strong assumptions about the underlying structure of the data. Even for methods that move away from hard clustering towards probabilistic methods for cell type assignment (Jetka et al. 2018, Zhu et al. 2019), assumptions remain and under certain conditions a continuous representation of the data may be better. Here we take such an approach, and seek to find a low dimensional representation of the data, on which further analyses (including but not limited to clustering) can be performed. VAEs are an obvious choice, given their success on other scRNA-seq analysis tasks, but modeling temporal changes with a feed-forward VAE would be equivalent to a fully agnostic search, similar to learning a Gaussian process. Recurrent networks offer well-established architectures for learning sequential and temporal data, and have been successfully combined with VAEs (Fabius & van Amersfoort 2014). We use a recurrent network architecture to structure the data and reduce the computational cost.

We introduce a recurrent variational autoencoder for modeling gene dynamics from scRNA-seq data (RVAgene). RVAgene learns two functions during training, parameterized by encoder and decoder networks. The encoder network projects the training data into latent space (we use a 2 or 3 dimensions in order to visualize, though there are no inherent limits). The decoder network learns a reconstruction of training genes from their latent representation. RVAgene facilitates clustering of other characterization of gene profiles in the latent space. By sampling points from the latent space and decoding them, RVAgene provides means to generate new gene expression time series data, drawn from the biological process that was encoded. Overall, RVAgene serves as a multipurpose generative model for exploring gene expression time-series data.

The remainder of the paper is structured as follows: we next present methodological details and development of RVAgene. We produce a synthetic gene expression time-series dataset with innate cluster structure, and demonstrate the accuracy of RVAgene on these data. We then explore two biological datasets with RVAgene: a scRNA-seq dataset on stem cell differentiation, where gene expression changes are studied over pseudotime; and a bulk RNA-seq dataset describing dynamic responses to kidney injury. We conclude by discussing key features and limitations of RVAgene, and recent advances in machine learning that will pave the way for future work in these directions.

## 2 Methods

We develop a recurrent variational autoencoder to model gene expression dynamics (RVAgene). Here we briefly describe the methods underpinning variational autoencoders, and present the implementation of RVAgene.

### 2.1 Variational inference and variational autoencoders

In the most general setting of a Bayesian model, we seek to learn the latent variables  $\mathbf{z}$  that best characterize some data  $\mathbf{x}$ . Given a generative process that draws latent variables from a prior distribution,  $p(\mathbf{z})$ , and a likelihood of the data observed that is given by  $p(\mathbf{x}|\mathbf{z})$ , then the posterior probability is given by Bayes rule:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int_z p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz}. \quad (1)$$

The denominator is often intractable, making it difficult to estimate  $p(\mathbf{z}|\mathbf{x})$ . Markov Chain Monte Carlo methods provide means to estimate posterior probability distributions. An alternative method to estimate hard-to-compute probability distributions is Variational Inference (VI) (Hoffman et al. 2013), which starts from the assumption that the posterior can be approximated by a distribution  $q(\mathbf{z})$  from the family  $\mathcal{Q}$ . VI then amounts to an optimization

problem to find the  $q^*$  that minimizes the Kullback–Leibler (KL) divergence between the approximation and the true posterior:

$$q^*(\mathbf{z}) = \operatorname{argmin}_{q(\mathbf{z}) \in \mathcal{Q}} \text{KL}(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})). \quad (2)$$

Much recent effort has gone into solving VI problems in different settings (Zhang et al. 2018, Ingraham & Marks 2017, Bouchard-Côté & Jordan 2010). VI can be framed as solving an optimization problem over function families: neural networks are popular candidates for representing and learning complex functions. VI was incorporated into autoencoders (Kingma & Welling 2014) to create the architecture of a variational autoencoder (VAE). A VAE consists of an encoder network to approximate  $p(\mathbf{z}|\mathbf{x})$  through a function  $q_{\mathbf{x}}(\mathbf{z})$ , and a decoder network  $p(\mathbf{x}|\mathbf{z})$  (Fig. 1A). Conceptually, the encoder solves an inference problem: approximating the posterior distribution  $p(\mathbf{z}|\mathbf{x})$  as some  $q_{\mathbf{x}}^*(\mathbf{z})$ , while the decoder solves a reconstruction problem: defining a generative process for  $p(\mathbf{x}|\mathbf{z})$ , given the latent variables. The VAE posterior is modeled by a multivariate normal  $\mathcal{N}(\mu, \Sigma)$  of the same dimension as  $\mathbf{z}$ . Training then comes down to minimizing two objective functions. For the encoder network, which should learn a “well distributed” latent space, minimize the KL divergence:  $\text{KL}(\mathcal{N}(\mu, \Sigma) || \mathcal{N}(0, \mathbf{I}))$ . For the decoder network, which should reconstruct the inputs  $\mathbf{x}$  from the latent space, minimizing either an  $L1$  or  $L2$  objective function with respect to  $\hat{\mathbf{x}}$  is appropriate. The use of KL-divergence and an  $L2$  objective solves the VI formulation of Eq. 2 (Kingma & Welling 2014), however, an  $L1$  objective may be preferred in practice, e.g. in cases where we want to suppress the effects of outliers on the structure of  $\mathbf{z}$  (Botchkarev 2018).

## 2.2 RVAgene: A recurrent variational autoencoder to model gene expression dynamics

Following the VAE architecture, RVAgene consists of an encoder and a decoder network with a reparameterization step in between. To incorporate the knowledge that we are modeling temporal data, recurrent neural networks offer an ideal architecture to use for both the encoder and the decoder networks. Recurrent and VAE networks have been successfully combined elsewhere, e.g. for textual (Nallapati et al. 2016) and time series data (Malhotra et al. 2015).

The architecture of RVAgene is based on Fabius & van Amersfoort (2014). An input sequence (i.e. gene)  $x \in \mathbf{x}$ ,  $x = (x_1, x_2, \dots, x_t, \dots, x_T)$  is encoded using a recurrent function described by a long short-term memory (LSTM) unit. LSTM units are the state-of-the-art in recurrent architectures, since they are robust against the vanishing gradient problem for longer sequences, unlike other recurrent units (see details in Hochreiter & Schmidhuber (1997)). We encode  $x$  in the following manner:

$$h_{t+1}^{enc} = \text{LSTM}(W_{enc}^T h_t^{enc} + W_{inp}^T x_t + b_{enc}), \quad (3)$$

where ( $W_{enc}$ ,  $W_{inp}$  and  $b_{enc}$ ) are network weight parameters, and the hidden states  $h_t$  represent information shared over timepoints in the LSTM. The dimension of the  $h_t$  (and  $W_{enc}$ ) is given by a hyperparameter (“hidden-size”). The encoded  $h_{t+1}$  are used to parametrize the posterior mean and variance from  $x$ , with mean  $\mu_z$  and diagonal covariance  $\sigma_z$  as:

$$\begin{aligned} \mu_z &= W_{\mu}^T h_{T+1}^{enc} + b_{\mu} \\ \log(\sigma_z) &= W_{\sigma}^T h_{T+1}^{enc} + b_{\sigma}. \end{aligned} \quad (4)$$

We then use the reparameterization trick described in Kingma & Welling (2014) to sample  $z$  from the distribution:

$$z = \mu_z + \epsilon \sigma_z, \quad (5)$$

where, for known  $\epsilon$ , backpropagation through the sampling step is possible while training the network.

For the decoder network, the first state  $h_1$  is calculated from  $z$ , and the recurrent formulation follows by reconstructing  $x$  as  $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_t, \dots, \hat{x}_T)$ , thus:

$$\begin{aligned} h_1^{dec} &= \text{sigm}(W_z^T z + b_z) \\ h_{t+1}^{dec} &= \text{LSTM}(W_{dec}^T h_t^{dec} + W_{out}^T \hat{x}_t + b_{dec}) \\ \hat{x}_t &= \text{sigm}(W_{out}^T h_t^{dec} + b_{out}), \end{aligned} \quad (6)$$

where  $\text{sigm}(u) = \frac{1}{1+e^{-u}}$  is the sigmoid activation function, and  $(W_i, b_i)$  are the network weight parameters. A schematic diagram of the network is shown in Fig. 1A, which can now be trained using backpropagation, to minimize the objective function:

$$\mathcal{L}(\theta, x) = D_{KL}(\mathcal{N}(\mu_z, \Sigma_z) || \mathcal{N}(\mathbf{0}, \mathbf{I})) + |x - \hat{x}|, \quad (7)$$

where  $\mu_z$  and  $\Sigma_z = \text{diag}(\sigma_z)$  are calculated from  $x$  by the encoder. 90

To evaluate the accuracy of RVAgene, we need an appropriate error measure. For each gene in the test set, we calculate the  $L1$  reconstruction error between generated data  $\hat{x}$  and true data  $x$ , averaged over all time points. We normalize the data to lie in  $[0, 1]$  to avoid skewing the error by differences in gene expression magnitudes. Thus we define:

$$\text{Reconstruction error}(x, \hat{x}) = \frac{1}{T} \sum_t |s(\hat{x})_t - s(x)_t|, \text{ where } s(x) = \frac{x}{\sum_{t=1}^T x_t}. \quad (8)$$

## 2.3 Generating synthetic gene expression time series data

To test RVAgene, we generate a synthetic time series dataset. Six clusters each containing 20 genes are simulated, where for each cluster  $c$ , the mean gene expression time series  $Y_c = (y_{c1}, y_{c2}, \dots, y_{ct})$  was generated using addition or convolution and rescaling of two random sinusoidal functions of the form  $k_1 \sin(k_2 t)$ , where  $k_1, k_2$  are randomly chosen positive integers. Trajectories of cluster members were then generated by sampling from the multivariate normal  $\mathcal{N}(Y_c, \Sigma_c)$ . We model  $\Sigma_c$  as the positive definite matrix  $\alpha Y_c Y_c^T$ , where  $\alpha$  is a scaling factor, we use:  $\alpha = 1/|Y_c Y_c^T|$ . As defined,  $\Sigma_c$  will describe nonzero correlations for all pairs of time points,  $(t_i, t_j)$ . This is unrealistic, so we set to 0 the entries of  $\Sigma_c$  for which column and row indices have a difference of more than some threshold  $T$  (we used  $T = 50$ ), reflecting the fact that correlations between time points are lost over larger time windows (temporal correlations are local). Note that under this condition,  $\Sigma_c$  is no longer necessarily positive definite. The multivariate Gaussian sampler `numpy.random.multivariate_normal()` implemented in `numpy` (Harris et al. 2020) was used to sample from this augmented  $\Sigma_c$ . 91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102

## 3 Results

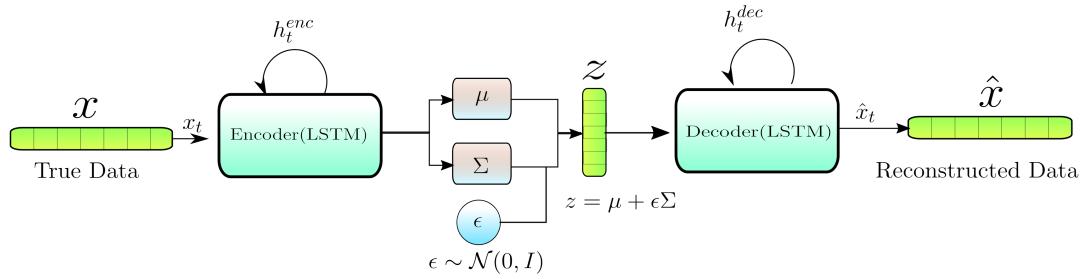
### 3.1 RVAgene can accurately and efficiently reconstruct temporal profiles from synthetic data

We generated a dataset of 120 genes using convolutions of sinusoidal functions (see Methods) to test the ability of RVAgene (Fig. 1A) to learn and predict noisy nonlinear temporal profiles. An RVAgene model was trained on all 120 genes from 6 clusters with a hidden size of 70 and a 3 dimensional latent space. The model was trained for 400 epochs, after which the average batch objective  $\mathcal{L}$  function indicates convergence (Fig. 1B), producing a three-dimensional latent space representation (Fig. 1C). K-means clustering on the latent space ( $k=6$ ) identified well-separated clusters (Fig. 1D). 103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124

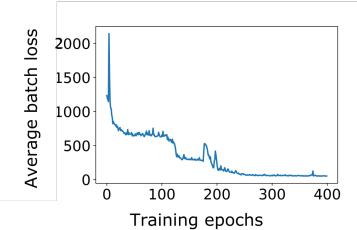
RVAgene modeling followed by k-means clustering on the latent space identified 6 clusters with perfect fidelity between predicted and true clusters. One might reasonably ask, why use a neural network for this task? Simpler dimensionality reduction methods (e.g. PCA, t-SNE, or a non-variational autoencoder) would also find the correct solution. RVAgene has the advantage over these methods that the underlying structure of the latent space leads to interpretability. A point in reduced PCA or t-SNE space that does not overlap with a data point is not interpretable. Traditional autoencoders lack regularity in the latent space, i.e. even for a representation with arbitrary accuracy (a reconstruction error of zero), decoding a point that does not correspond to a training data point can result in nonsensical generated data, even if the decoded point is arbitrarily close to a training data point. Variational Autoencoders remedy this by learning a regularised or smoother distribution on the latent space. In this sense, the KL-divergence term in the VAE loss function can be thought of as a regulariser. This property enables RVAgene to generate new gene expression dynamics by decoding points from different regions of the latent space, having properties similar to clusters nearby to those points.

To demonstrate the generative properties of the RVAgene latent space, we sample points from multivariate Normal distributions, centered on the empirical mean of each cluster with variance of 0.4, i.e.  $\mathcal{N}(\mu_c, 0.4I)$ , 125

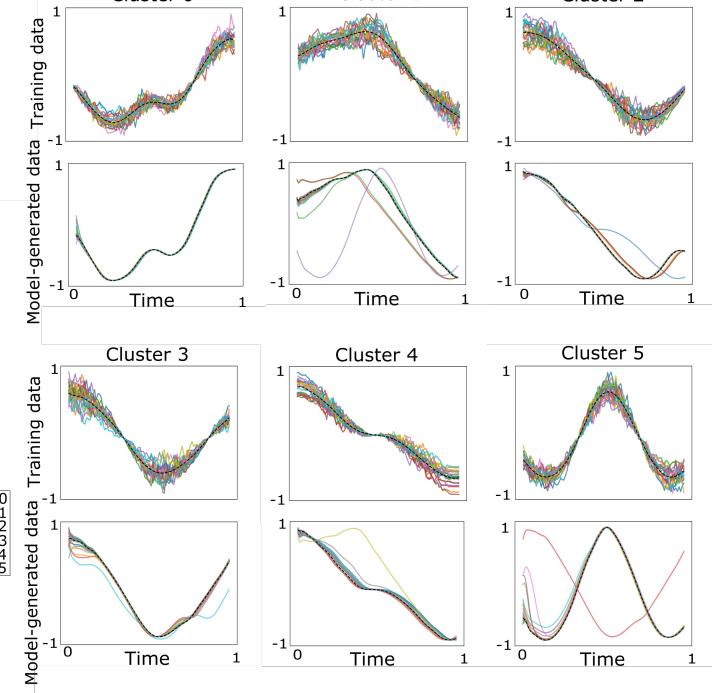
A



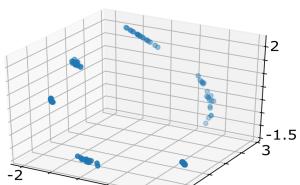
B



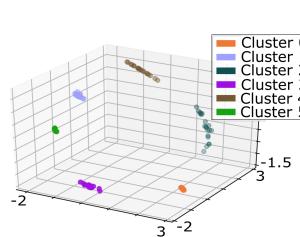
E



C



D



**Figure 1. Unsupervised representation learning with RVAgene using synthetic data.** (A) Schematic diagram of the RVAgene model. (B) Average loss function  $\mathcal{L}$  as over duration of training. (C) Latent space representation learnt by RVAgene model after training. (D) Clusters detected by  $k$ -means clustering on the latent space, with  $k = 6$  (E) First and third rows show input training data used (20 simulated genes in each of six clusters); cluster means shown in black. Second and fourth rows show the model-generated data, obtained by sampling and decoding points from the latent space; decoded cluster empirical means shown in black.

where  $\mu_c$  is the empirical mean of the cluster and  $\mathbf{I}$  is the identity matrix in  $\mathbb{R}^3$ . Corresponding to each cluster, we sample 20 points in the latent space, and use the decoder network to generate new time series data (Fig. 1E). Most of the points sampled generate trajectories that belong to the correct cluster. Moreover, we identify cases where corresponding to transitions between clusters. For example, some points sampled near Cluster 2 generate trajectories that are similar to members of Cluster 4, and vice versa. This makes sense due to the similarity between the temporal profiles of Clusters 2 and 4. A similar correspondence is observed between Clusters 1 and 5. We also observe some generated trajectories that display intermediate profiles between two or more clusters: the decoder function learnt by RVAge is smooth, and gives rise to meaningful representations of points across regions of the latent space.

RVAge offers additional functionality as a tool for removing noise the data. RVAge model reconstructions generated by decoding points in the latent space denoise the data, with trajectories that are smooth relative to the input data (Fig. 1E). Similar neural network approaches have been proposed to denoise from single-cell data, e.g. using a deep count autoencoder (Eraslan et al. 2019). RVAge provides data denoising as a by-product of its primary functionality: modeling dynamic changes in gene expression.

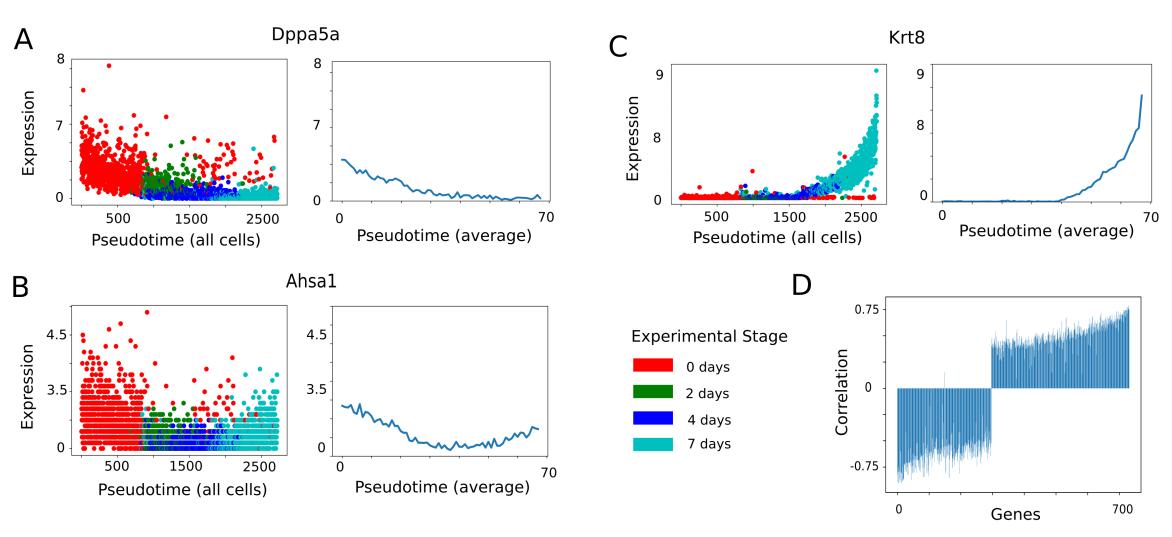
Similar to other VAE-based tools for the analysis of single-cell data, RVAge is efficient and scalable for use with large datasets. We compared the performance of RVAge with a Bayesian nonparametric approach for the analysis of gene expression time series data (Dirichlet Process Gaussian Process (McDowell et al. 2018)). Using either CPU or GPU computing, the time and memory gains are substantial, enabling the analysis of larger datasets than would otherwise be possible (Fig. S1). Analysis of RVAge using simulated temporal data highlights the ability of such an architecture as means to study and generate gene expression dynamics. It enables learning of an unsupervised representation space, on which post-processing (e.g. unsupervised clustering) can be performed, as well as data denoising, and the generation of new time series data from arbitrary points in the latent space.

It is inevitably challenging to include sufficient dimensionality and variation in synthetic datasets to accurately capture biological processes such as those we observe in experimental datasets. Thus, in the subsequent two sections, we test the capabilities of RVAge on two whole-genome biological datasets: embryonic stem cell differentiation, and kidney injury response. As we will see, in these cases it may not be possible to characterize the latent space by simple (e.g. k-means) clustering; we need to use other means to gain insight into the features of the latent space.

### 3.2 Characterization of embryonic stem cell differentiation with RVAge

We applied RVAge to model gene expression dynamics during embryonic stem cell (ESC) differentiation. Klein et al. (2015) identified 732 differentially expressed genes over the time course of mouse ESC differentiation following leukemia inhibitory factor (LIF) withdrawal. Data is gathered at four time points: 0, 2, 4, and 7 days after LIF withdrawal. (Table S2 in Klein et al. (2015)). We ordered the data (2717 single cells) using diffusion pseudotime (DPT), which provides robust methods for the reconstruction of single-cell temporal processes (Haghverdi et al. 2016). The root cell was randomly sampled from the initial time point (Fig. 2A-C). The inferred pseudotime is highly correlated with the experimental time points, giving confidence that true biological processes are represented over the DPT pseudotime. The gene expression dynamics over pseudotime show considerable variability among cells. To smooth the data, we apply a moving window average, over windows of length 40, to give 68 time points after smoothing (Fig. 2A-C). We fit linear regression models to the smoothed pseudotime profiles of each gene (Fig. S2), and see that for the majority of genes the correlation coefficients are  $> 0.5$  (Fig. 2D), with a clear distinction between the up- and down-regulated genes over pseudotime.

An RVAge model was trained on the data with a two-dimensional latent space, on which genes are classified based on their correlation coefficients (Fig. 3A). Two distinctive characteristics emerge: a) the two groups (up- and down-regulated genes) are well-separated in the latent space, and b) the two groups merge and overlap at some point, illustrating the continuity of the latent space, as discussed above. We compared the results of RVAge with DPGP, an unsupervised approach for gene expression time series clustering (McDowell et al. 2018). DPGP is a hierarchical Bayesian model that estimates the number of clusters along with the cluster membership, although to do so it has considerably higher resource consumption (Fig. S1). To assess the correspondence between methods, genes clustered by DPGP (Fig. S3) were projected onto the RVAge latent space (Fig. 3B). Of the 12 clusters detected by DPGP, the four largest can be characterized by their up- and down-regulation profiles



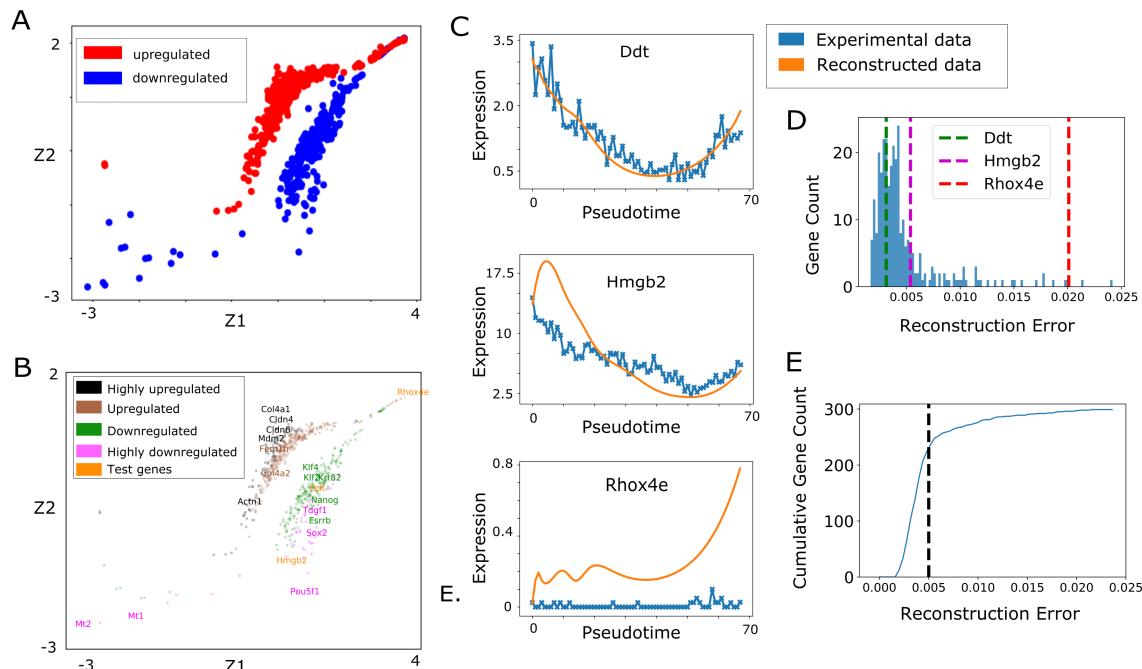
**Figure 2. Analysis of embryonic stem cell differentiation gene expression dynamics in single cells.** (A-C) Pseudotemporal ordering of 2717 single cells (data from (Klein et al. 2015)), calculated using DPT; three example genes shown: Dppa5a, Ahsa1, and Krt8. Gene expression values given as  $\log_2(\text{counts}+1)$  for all cells (left), and for sliding window average (right). (D) Pearson correlation coefficient between gene expression and time for 732 differentially expressed genes.

over pseudotime. On the RVAgene latent space, we find that genes sampled from each of the DPGP clusters appear close together, and moreover, are represented on a spectrum from upregulation to downregulation (Fig. 3B). The goals of RVAgene and DPGP are to some degree complementary: DPGP characterizes gene expression profiles discretely with no need for prior information, while RVAgene characterizes profiles with a continuous representation, that can explain smooth changes in patterns.

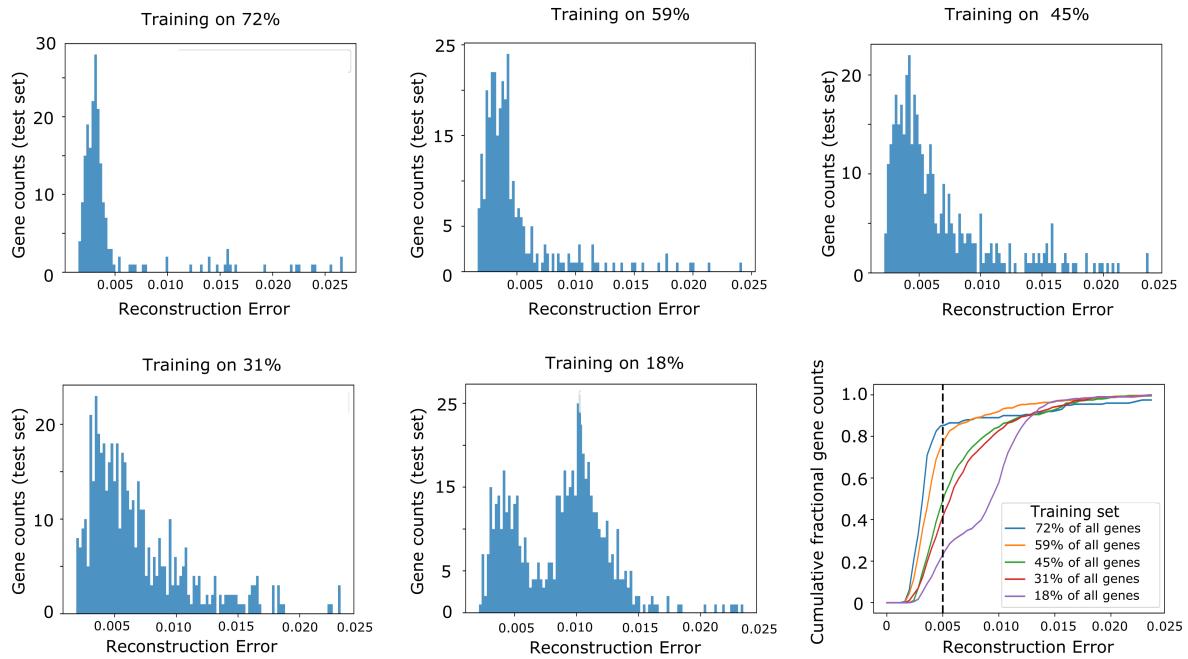
To assess model predictions for individual genes, we kept aside 300 genes for testing and trained RVAgene on the remaining 432 genes. Predicted gene expression profiles are shown for three reconstructed genes, chosen to sample across the spectrum of reconstruction errors (Fig. 3C). The prediction for *Ddt*, which has a reconstruction error near the mode (Fig. 3D), shows very high accuracy. The prediction for *Hmgb2*, which has twice the reconstruction error, still broadly captures the temporal profile but with lesser accuracy. Finally we show the prediction for *Rhox4e*, a gene that was sampled from the long tail of the reconstruction error distribution, i.e. does not well match the data. Comparing these three examples with the full distribution of reconstruction errors (Fig. 3D), we see that the large majority of genes lie to the left of *Hmgb2*, i.e. have better-than-moderate prediction accuracy. The reconstruction error of *Hmgb2* is close to 0.005, which we use as a cut off for “well-reconstructed” genes, based on analysis of individual gene predictions. The cumulative reconstruction error distribution reiterates this point: 230 out of 300 genes (77%) have a reconstruction error  $\leq 0.005$  (Fig. 3E); we can conclude that the majority of test genes were faithfully reconstructed by the model.

RVAgene accurately reconstructed most gene profiles using only  $\sim 60\%$  of the data for training (Fig. 4), likely due to co-regulation of gene expression programs. This led to a question: what is the smallest training gene set that can be used to accurately reconstruct gene dynamics? We subset the data randomly into train/test sets and trained separate RVAgene models on each. We found that reconstruction errors slowly increase as the size of the training set decreases, but not until the training set was as low as 18% of the data did the reconstruction errors significantly increase (Fig. 4). Analysis of the cumulative distribution of reconstruction errors across all groups found that RVAgene predicts the majority of gene temporal profiles well (defined as below a reconstruction error of 0.005) if  $\geq 45\%$  of the data is used for training. The successful prediction of gene expression dynamics de novo using small subsets of the data suggests widespread co-regulation of gene expression programs during embryonic stem cell differentiation, as found in previous work (Jang et al. 2017).

177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203



**Figure 3. Accurate reconstruction of embryonic stem cell differentiation dynamics with RVAgene.** (A) The 2D latent space learnt by an RVAgene model trained on 732 gene profiles over pseudotime, showing clear separation between upregulated and downregulated genes. (B) Comparison of RVAgene and DPGP. The four largest clusters from DPGP are plotted on the RVAgene latent space: temporal expression patterns (from highly upregulated to highly downregulated) are in close agreement between methods. (C) Comparison of experimental data and predictions. Model-generated reconstructions of three genes from the test set not used in training: Ddt, Hmgb2, and Rho4e. Expression values are  $\log_2(\text{counts}+1)$ . (D) Distribution of average  $L_1$  reconstruction errors for the 300 genes used in the test set. Genes plotted in C are marked. (E) Cumulative distribution of reconstruction error: 77% of genes (230/300) have reconstruction error less than 0.005.



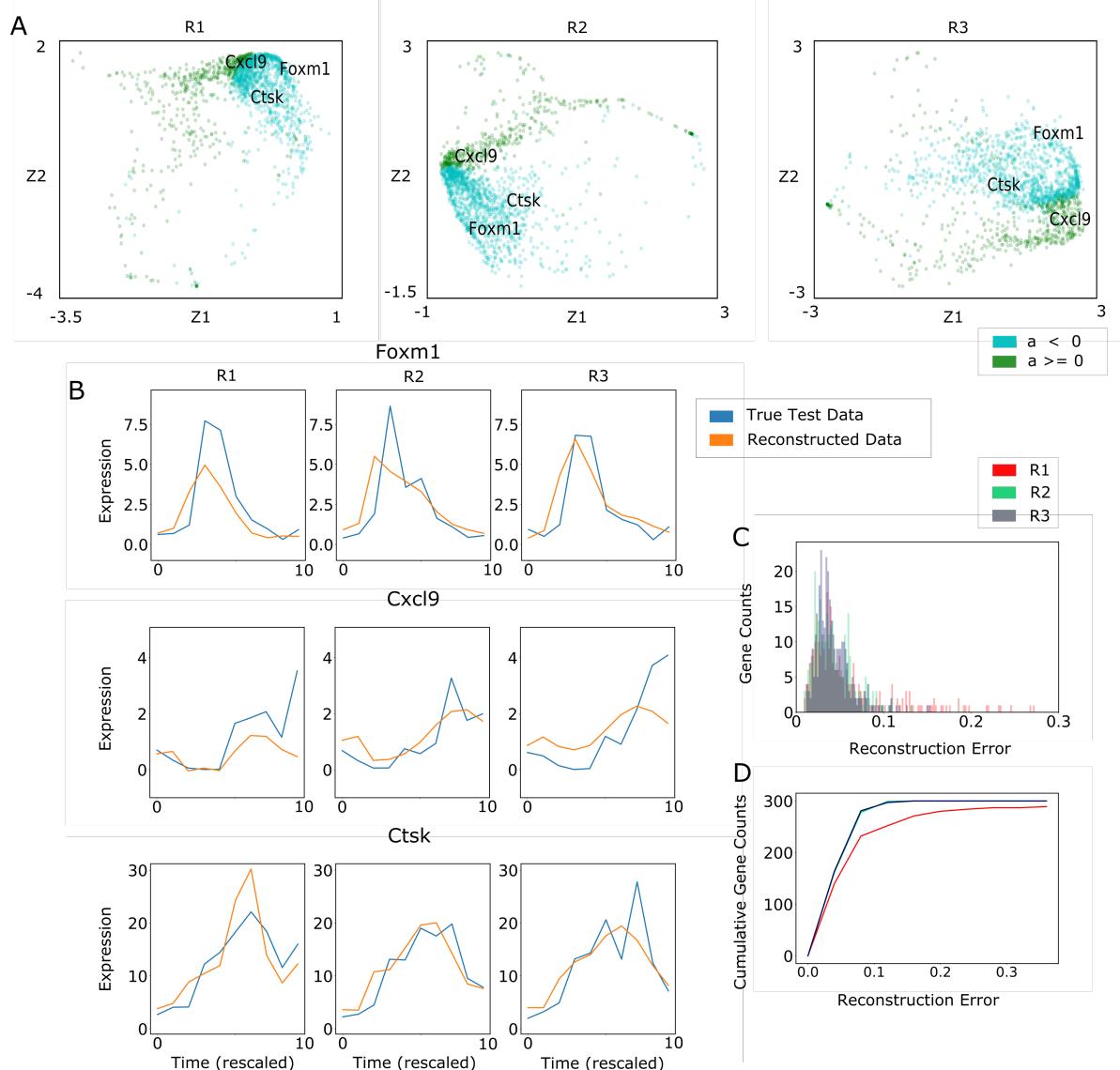
**Figure 4. Accuracy of RVAgene reconstructions for different train/test group sizes.** Distributions of reconstruction errors on randomly sampled sets of test genes, where the full data were split into test groups of: 200 genes (train on 72%), 300 genes (train on 59%), 400 genes (train on 45%), 500 genes (train on 31%), and 600 genes (train on 18%). Cumulative fractional distribution of reconstruction errors (cumulative count/test set size) for all groups.

### 3.3 RVAgene can classify and predict gene expression dynamics in response to kidney injury

We investigated gene expression dynamics in the murine kidney by applying RVAgene to a dataset that describes gene expression profiles before, during, and after a kidney injury (Liu et al. 2017). The dataset is temporally rich, with a total of ten bulk samples over twelve months. Since in this case no single-cell information is available, we cannot order samples by pseudotime to smooth the data. Moreover, the temporal gene expression profiles described in Liu et al. (2017) display more complex dynamics than for the previous dataset (Klein et al. 2015), and are not readily separable by linear patterns of up- and down-regulated genes (cf. Fig. 3A). Thus, below, we must consider nonlinear models in order to characterize the temporal patterns observed.

The data consist of one initial timepoint ( $t = 0$ ) before the injury event (an ischemia/reperfusion injury model) and nine subsequent time points ( $t = 1$  to 10) following the injury (48 hours, 72 hours, 7 days, 14 days, 28 days, 6 months and 12 months). We note that the timepoints are not uniformly spaced, which is not taken into account in RVAgene, which only models the broad temporal trend (see Discussion). From an initial list of 1927 differentially expressed genes measured over the time course in three biological replicates, we removed putative/predicted and non-protein coding genes, retaining a list of 1713 genes as input to the model.

We ran RVAgene separately for each of three biological replicates. Independent replicates & independently trained models provide additional means with which to test the reproducibility of these methods. For each replicate, RVAgene was trained with a two-dimensional latent space and a hidden size of 10, on the full set of genes over 200 epochs: found to be sufficient for the convergence of  $\mathcal{L}$  (see Methods for further details). We fit linear regression models to the temporal gene profiles (Fig. S4) and found that linear fits rarely described the gene temporal profiles well (most correlation coefficients had values close to zero), not did they identify separate clusters in the latent space. Normalizing the data to lie in  $[0, 1]$  improved our ability to discriminate clusters in the latent space (Fig. S4C), but came at the expense of a significant loss of information, as the variance captured in the latent space was dramatically reduced. The absence of evidence for linear correlations could



**Figure 5. Accurate reconstruction of kidney injury response gene dynamics with RVAgene.** (A) Latent space representations of RVAgene models trained separately on three independent replicates (R1-R3); classified by quadratic fit coefficient  $a$ . (B) Model generation of gene dynamics for genes not used in training: *Foxm1*, *Cxcl9* and *Ctsk*. (C) Histograms of reconstruction errors for RVAgene models trained on R1-R3 (truncated). (D) Cumulative distribution of reconstruction errors.

indicate expression dynamics that are uncorrelated with time, but could of course also indicate more complicated (nonlinear) gene expression dynamics, which are explored below.

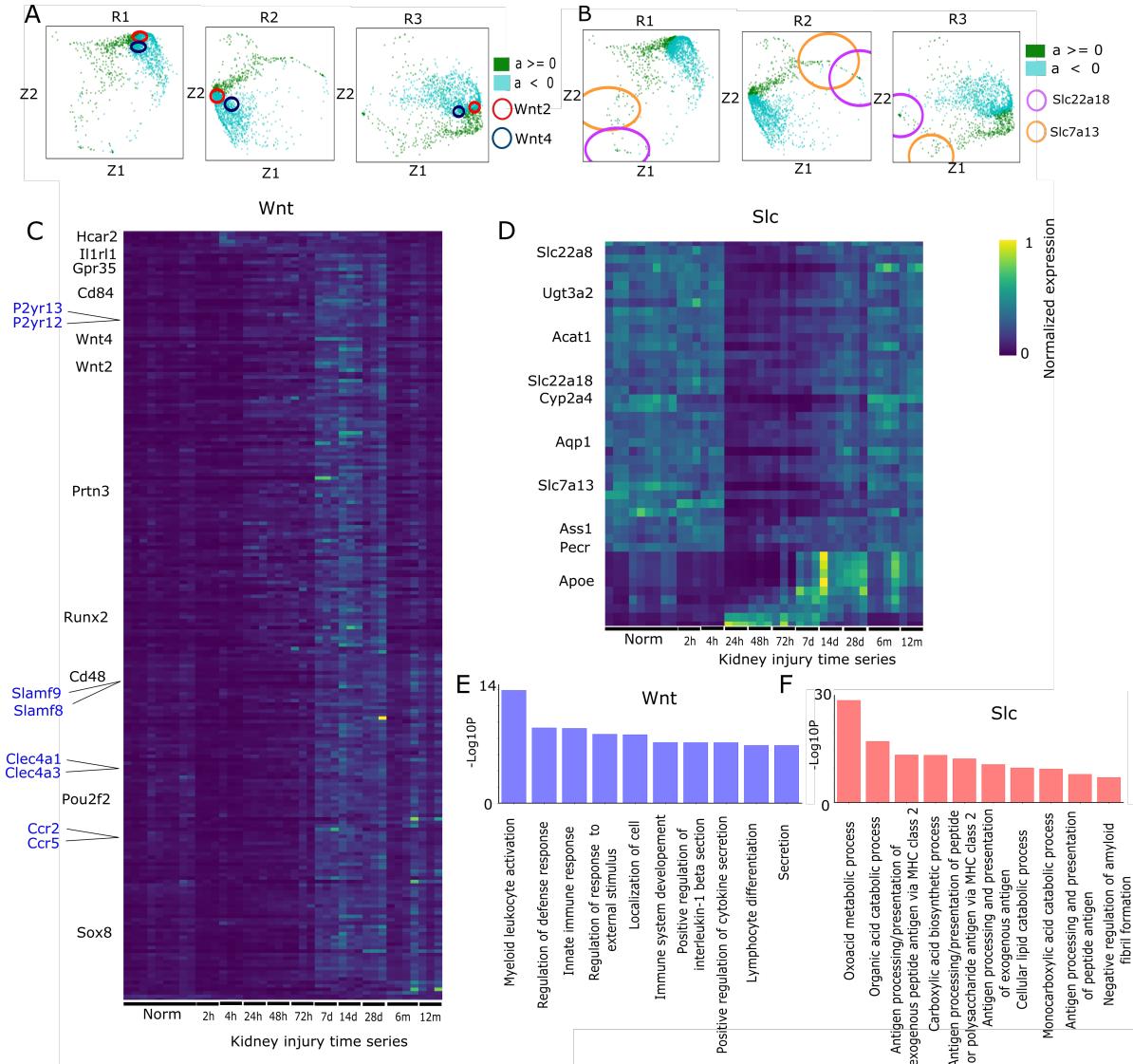
To study nonlinear gene expression dynamics, we fit a 2nd degree polynomial, i.e. we fit the temporal trajectory of each gene  $x$  to:  $x = at^2 + bt + c$ , where  $a, b, c$  are constants (Fig. S5). We hypothesized that this function could adequately describe the transient dynamics observed by Liu et al. (2017) for most genes in response to the kidney injury. Thus, we classified genes into one of two groups,  $a < 0$ : convex (up-down pattern), 1200 genes; and  $a \geq 0$ : concave (down-up pattern), 512 genes. In the latent space, the separation of these two groups is clearly visible for each replicate (Fig. 5A). Moreover, the classification is in agreement with Liu et al. (2017), where the majority of differentially expressed genes are upregulated transiently. To explore the ability of RVAgene to generate gene expression dynamic profiles de novo, we kept aside 300 randomly sampled genes for testing, and trained RVAgene models on the remaining genes for each of the three replicates. Independently for each model, we then generated dynamic profiles for the test genes. Three genes sampled randomly from the test set are plotted in Fig. 5B. Of particular note, for each of genes, the model-generated data captures the temporal patterns while displaying a higher degree of similarity across replicates than the experimental data itself. This illustrates that the model is neither under- nor overfitting, but capturing the underlying biological patterns while sufficiently accounting for the noise. Reconstruction errors are comparable across the three replicates, albeit with slightly higher overall errors in replicate 1 (Fig. 5C-D). Overall, the reconstruction errors are higher than for the previous section (averaging over many pseudotemporal time points allowed us to significantly reduced the noise).

To investigate in more depth the features that are captured in the RVAgene latent space, we studied three distinct areas of the latent space of each model. Areas were defined using three gene groups, chosen simply based on their co-location in distinct regions of the latent space: 1) a *Wnt* group consisting of family members *Wnt2* & *Wnt4*; 2) an *Slc* group consisting of family members *Slc7a13* & *Slc22a18*; and 3) a *Sdc1* group, consisting of only *Sdc1*. For each group, we characterized neighboring genes by defining a circular neighborhood around each gene in the group, with radius  $r$  (depending on the local density, the radius was varied, giving:  $r^2 = 1$  for *Slc*,  $r^2 = 0.3$  for *Sdc*,  $r^2 = 0.05$  for *Wnt*). We then took all genes inside this radius for each replicate, and found the intersection of genes over the three replicates (Fig. 6A-B). We analyzed the intersection gene set for each group by studying their temporal profiles and their gene ontology (GO) term associations. Each group was characterized by a strikingly clear temporal profile. The *Sdc1* and *Wnt* groups both show transient upregulation, over different timescales: the *Sdc1* group is upregulated from 24 hours post-injury until 14-28 days post-injury (fast response) (Supplementary Fig. S6B), whereas the *Wnt* group is upregulated at 7 days post-injury until 28 days post-injury (slow response) (Fig. 6C). In contrast, the *Slc* group is downregulated at 24 hours post-injury, and remains suppressed until 7-28 days post-injury (Fig. 6D).

Analysis of GO biological process terms enriched in each gene group further highlighted the power of the latent space for biological discovery. The fast response (*Sdc1*) group was characterized by upregulation of programs related to apoptosis, stress response, wound healing and chemotaxis, i.e. the first responders to the site of injury (Fig. S6C). In addition all five *Lox* genes comprising the GO term “peptidyl-lysine oxidization” were found in this group. This is consistent with the oxidative stress resulting from the renal ischemia-reperfusion injury that was performed. However, distinct factors regulate the *Lox* family genes, as can be partly observed by their subtle differences in temporal profile (Fig. S6D). Their co-location in the latent spaces of all three models thus highlights the potential use of RVAgene for discovery of complex temporal regulatory events from gene expression data.

The slow response (*Wnt*) group was primarily characterized by immune response processes, including leukocyte activation, platelet aggregation, and various cytokine-mediated pathways including *IL-1* and *IL-33* (Fig. 6E). Notably, the *Wnt* group identifies multiple gene orthologs (Fig. 6C) with very similar profiles: likely evidence of shared temporal regulation. This illustrates once again (as for the *Lox* genes above) the potency of RVAgene for the discovery of temporally co-regulated genes.

Finally, the *Slc* group of genes shows a transiently down-regulated pattern between 24 hours and 7-28 days, although some gene in this group deviate from this pattern (Fig. 6D). GO term enrichment identifies the positive regulation of metabolic processes (Fig. 6F). The downregulation of metabolic programs during the response to kidney injury is agreement with the findings of Liu et al. (2017). Notably, this metabolism-sensitive group contains many genes that also display sexually dimorphic expression, primarily in specific regions of the proximal tubule (Ransick et al. 2019), thus independently identifying the well-established (though under-studied) interplay



**Figure 6. RVAgene latent space captures biological processes driving concordant gene expression changes.** (A) Z-plots for replicates R1-R3 with local neighborhoods of Wnt2 and Wnt4 marked (circles). (B) As in A, for Slc family members Slc22a18 and Slc7a13. (C) Heatmap of expression changes over time course of injury for the Wnt neighborhood genes in the intersection of R1-R3. Selected genes marked (black), as well as ortholog gene pairs (blue). (D) As in C, for Slc neighborhood genes. (E) Histogram of  $-\log_{10} p$  values of gene ontology terms for biological processes terms associated with the Wnt neighborhood (gene set in C). (F) As in E, with the Slc neighborhood (gene set in D).

between sex differences and injury responses in the kidney (Neugarten et al. 2000).

In summary, unsupervised analysis of groups of genes co-located in the latent spaces of RVAgenc finds: 1) high similarity between temporal gene profiles of genes nearby in latent space, and 2) clear biological signatures represented by these groups of nearby genes, in strong agreement with prior knowledge (Liu et al. 2017). Moreover, the latent spaces of RVAgenc models can be used to predict programs of temporal co-regulation.

## 4 Discussion

We have presented RVAgenc, a recurrent variational autoencoder for generative modeling of gene expression time series data. Through its encoder network, RVAgenc provides means to visualize and classify gene expression dynamic profiles, which can lead to the discovery of biological processes. Through its decoder network, RVAgenc provides means to generate new gene expression dynamic profiles by sampling points from the latent space, and can accurately predict gene dynamics in complex biological data. As a by-product, the model produces smoothed outputs, which can be used for denoising gene expression time series data. RVAgenc is efficient on temporally-rich whole genome datasets, in comparison to current methods used (Gaussian processes).

RVAgenc can be used to discover structure in the data, such as gene profile clusters. Popular methods for clustering gene profiles such as Bayesian hierarchical clustering (Cooke et al. 2011) or DPGP (McDowell et al. 2018) detect the number of clusters in the data by fitting a hyperparameter  $\alpha$ , the concentration parameter of the governing Dirichlet process (Ferguson 1973). Although unsupervised, inevitably, the choice of  $\alpha$  affects the number of clusters output. Visualizing the data first with RVAgenc can give an idea whether the data favor clustering or a continuous representation. Thus analysis in RVAgenc can guide the setting of the hyperparameter  $\alpha$  in DPGP and similar methods. In the case of ESC differentiation, DPGP predicts 12 clusters (Fig. S3), yet most have very few members and many share similar patterns. The RVAgenc latent space for this dataset finds two major divisions in the data, and orders the largest DPGP clusters along a spectrum (Fig. 3B), suggesting that DPGP might be overfitting the data. Indeed, the two methods can be used complementarily: RVAgenc for high-level structure discovery and DPGP for clustering. We note however that DPGP does not scale well with large datasets and thus cannot always be used (Fig. S1).

The latent space of an RVAgenc model encodes useful information about biological features, and in that sense provides biologically interpretable representations of the data. However, the representation is not interpretable in the sense that the components of the latent space do not have a physical meaning nor are they necessarily independent. Recent methods have tackled this issue of interpretability, by either modifying the loss function to make components independent (Higgins et al. 2016) or substituting linear functions in parts of the VAE (Svensson et al. 2020, Ainsworth et al. 2018). These methods have clear advantages regarding the analysis and interpretation of features in the latent space. In future work, decoding an RVAgenc model with a linear function (Svensson et al. 2020) could facilitate additional discovery and improve our ability to gain insight into dynamic biological processes through the analysis of the latent space.

Dynamic changes in gene expression underlie essential cell processes. As such, modeling gene expression changes can also facilitate downstream analysis tasks, including gene regulatory network (GRN) inference. Inferring gene regulatory networks from single-cell data is challenging (Chen & Mar 2018), particularly due to cell-cell heterogeneity and high levels of noise. Several recent approaches to GRN inference use differential equations in their formulation to model gene expression changes (Ma et al. 2020, Aubin-Frankowski & Vert 2020, Matsumoto et al. 2017). RVAgenc could either supplement such methods directly by providing denoised input data, or could be used to replace the differential equation-based components of these methods (which are notoriously difficult to parameterize) with a generative model of gene expression dynamics.

RVAgenc is currently agnostic of irregular time intervals between consecutive points in a time series, i.e. it standardizes the time interval. This is not usually a concern for single-cell data, since with pseudotime information we can choose appropriate time intervals. However, in other cases, such as in response to kidney injury (Liu et al. 2017), standardizing time intervals distorts the dynamic profiles. Since RVAgenc seeks to describe broad temporal patterns, we do not see this as a critical issue, though it would be desirable to generalize the model. A simple way to model irregularly spaced time points would be to augment the data through interpolation, though this is difficult without making strong assumptions about the (generally unknown) noise model. Gaussian process models (McDowell et al. 2018, Hensman et al. 2013) can take irregular data as input, although (as noted above)

are not efficient enough to run on large datasets. An alternative approach would be to modify the recurrent network architecture to take time points explicitly as input values, this would enable modeling of irregular or asynchronous data (Wu et al. 2018).

RVAgene models in discrete time steps. There is no simple modification to the recurrent network structure that allows for prediction on continuously valued time. However, a recent development: neural ordinary differential equations (ODEs) (Chen et al. 2018), enables modeling of time series data with continuous timepoints. Chen et al. (2018) describe a generative latent ODE architecture similar to that of RVAgene, except that in their case the recurrent decoder network is replaced by a neural ODE decoder network. Chen et al. (2018) demonstrate accurate results using synthetic data, however when we applied the method to the ESC single-cell differentiation dataset (Klein et al. 2015), the neural ODE network was found to converge very slowly and was overall underfit (Fig. S7). The latent ODE method used by Chen et al. (2018) does not address the challenge of modeling asynchronous/irregularly spaced data, but this has been more recently addressed (Rubanova et al. 2019). These new models may well lead to future improvements in network architectures, although it seems that computational progress is needed before they can be successfully applied to complex biological systems.

In the current work, the prior on latent space used throughout was a unit spherical Normal, appropriate for exploratory data analysis where we have no further knowledge about structure in the latent space. However, given more information, e.g. that the data is contains  $k$  clusters, a different prior on the latent space might be more appropriate. A multi-modal prior – such as a Gaussian Mixture Model (GMM) prior – would permit structured (multi-modal) representations. However, the KL-divergence for an arbitrary GMM is not tractable; approximation (Hershey & Olsen 2007) or numerical computation would be necessary. Moreover, there is a greater problem: mixture models contain discrete parameters and VAE models are ill-suited for the optimization of discrete parameters (Dilokthanakul et al. 2016), thus directly replacing the Normal prior of a VAE with a GMM is not feasible. A workaround to this problem is presented in (Dilokthanakul et al. 2016), however implementing this for a recurrent model architecture remains an open problem.

The points raised above offer much scope for future work. These include the design of new latent space models with informative priors, modeling irregular time series data, and modeling in continuous time. Developments in some of these areas (Chen et al. 2018), while promising, tend to rely on training data with relatively low levels of noise: far from the reality of most biological data. Thus it seems highly likely to be beneficial for both machine learning and biology to develop new neural network architectures in light of biological data.

## Acknowledgements

We thank A.P. McMahon for valuable discussions and comments on the manuscript. R.M. gratefully acknowledges support from a USC Viterbi Fellowship.

## Data Availability

The synthetic data used for evaluation of RVAgene are available at: <https://github.com/maclean-lab/RVAgene>. Additional data used in the manuscript are available from the Gene Expression Omnibus: ESC differentiation (GEO accession GSE65525) and kidney injury (GEO accession GSE98622).

## Software Availability

RVAgene is available in Python released under an MIT license: <https://github.com/maclean-lab/RVAgene>.

## Supplementary Information

Supplementary figures are available for download.

## References

Ainsworth, S. K., Foti, N. J., Lee, A. K. & Fox, E. B. (2018), oi-vae: Output interpretable vaes for nonlinear group factor analysis, in 'International Conference on Machine Learning', pp. 119–128.

Aubin-Frankowski, P.-C. & Vert, J.-P. (2020), 'Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference', *Bioinformatics* p. btaa576.

Botchkarev, A. (2018), 'Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology', *arXiv preprint arXiv:1809.03006* .

Bouchard-Côté, A. & Jordan, M. I. (2010), Variational inference over combinatorial spaces, in 'Advances in Neural Information Processing Systems', pp. 280–288.

Chen, S. & Mar, J. C. (2018), 'Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data', *BMC Bioinformatics* 19(1), 232.

Chen, T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. (2018), Neural ordinary differential equations, in 'Advances in neural information processing systems', pp. 6571–6583.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M. et al. (2018), 'Opportunities and obstacles for deep learning in biology and medicine', *Journal of The Royal Society Interface* 15(141), 20170387.

Cooke, E. J., Savage, R. S., Kirk, P. D., Darkins, R. & Wild, D. L. (2011), 'Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements', *BMC bioinformatics* 12(1), 399.

Deng, Y., Bao, F., Dai, Q., Wu, L. F. & Altschuler, S. J. (2019), 'Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning', *Nature methods* 16(4), 311–314.

Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K. & Shanahan, M. (2016), 'Deep unsupervised clustering with gaussian mixture variational autoencoders', *arXiv preprint arXiv:1611.02648* .

Ding, J., Condon, A. & Shah, S. P. (2018), 'Interpretable dimensionality reduction of single cell transcriptome data with deep generative models', *Nature communications* 9(1), 1–13.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. (2019), 'Single-cell rna-seq denoising using a deep count autoencoder', *Nature communications* 10(1), 1–14.

Fabius, O. & van Amersfoort, J. R. (2014), 'Variational recurrent auto-encoders', *arXiv preprint arXiv:1412.6581* .

Ferguson, T. S. (1973), 'A bayesian analysis of some nonparametric problems', *The annals of statistics* pp. 209–230.

Haghverdi, L., Buettner, M., Wolf, F. A., Buettner, F. & Theis, F. J. (2016), 'Diffusion pseudotime robustly reconstructs lineage branching', *Nature methods* 13(10), 845.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J. et al. (2020), 'Array programming with numpy', *Nature* 585(7825), 357–362.

Hensman, J., Lawrence, N. D. & Rattray, M. (2013), 'Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters', *BMC bioinformatics* 14(1), 252.

Hershey, J. R. & Olsen, P. A. (2007), Approximating the kullback leibler divergence between gaussian mixture models, in '2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07', Vol. 4, IEEE, pp. IV–317.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. & Lerchner, A. (2016), 'beta-vae: Learning basic visual concepts with a constrained variational framework'.

Hinton, G. E. & Salakhutdinov, R. R. (2006), 'Reducing the dimensionality of data with neural networks', *science* **313**(5786), 504–507.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Hoffman, M. D., Blei, D. M., Wang, C. & Paisley, J. (2013), 'Stochastic variational inference', *The Journal of Machine Learning Research* **14**(1), 1303–1347.

Ingraham, J. & Marks, D. (2017), Variational inference for sparse and undirected models, in 'International Conference on Machine Learning', PMLR, pp. 1607–1616.

Jang, S., Choubey, S., Furchtgott, L., Zou, L.-N., Doyle, A., Menon, V., Loew, E. B., Krostag, A.-R., Martinez, R. A., Madisen, L. et al. (2017), 'Dynamics of embryonic stem cell differentiation inferred from single-cell transcriptomics show a series of transitions through discrete cell states', *elife* **6**, e20487.

Jetka, T., Nienaltowski, K., Filippi, S., Stumpf, M. P. & Komorowski, M. (2018), 'An information-theoretic framework for deciphering pleiotropic and noisy biochemical signaling', *Nature communications* **9**(1), 1–9.

Kingma, D. P. & Welling, M. (2014), 'Auto-encoding variational bayes', *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings arXiv:1312.6114* .

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A. & Kirschner, M. W. (2015), 'Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells', *Cell* **161**(5), 1187–1201.

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

Liu, J., Kumar, S., Dolzhenko, E., Alvarado, G. F., Guo, J., Lu, C., Chen, Y., Li, M., Dessing, M. C., Parvez, R. K. et al. (2017), 'Molecular characterization of the transition from acute to chronic kidney injury following ischemia/reperfusion', *JCI insight* **2**(18).

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. (2018), 'Deep generative modeling for single-cell transcriptomics', *Nature methods* **15**(12), 1053–1058.

Ma, B., Fang, M. & Jiao, X. (2020), 'Inference of gene regulatory networks based on nonlinear ordinary differential equations', *Bioinformatics* .

Malhotra, P., Vig, L., Shroff, G. & Agarwal, P. (2015), Long short term memory networks for anomaly detection in time series, in 'Proceedings', Vol. 89, Presses universitaires de Louvain, pp. 89–94.

Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M. S. H., Ko, S. B. H., Gouda, N., Hayashi, T. & Nikaido, I. (2017), 'SCODE: An efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation', *Bioinformatics* **33**(15), 2314–2321.

McDowell, I. C., Manandhar, D., Vockley, C. M., Schmid, A. K., Reddy, T. E. & Engelhardt, B. E. (2018), 'Clustering gene expression time series data using an infinite gaussian process mixture model', *PLoS computational biology* **14**(1), e1005896.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. et al. (2016), 'Abstractive text summarization using sequence-to-sequence rnns and beyond', *arXiv preprint arXiv:1602.06023* .

Neugarten, J., Acharya, A. & Silbiger, S. R. (2000), 'Effect of Gender on the Progression of Nondiabetic Renal Disease: A Meta-Analysis', *Journal of the American Society of Nephrology* **11**(2), 319–329.

Raj, A. & Van Oudenaarden, A. (2008), 'Nature, nurture, or chance: stochastic gene expression and its consequences', *Cell* **135**(2), 216–226.

Ransick, A., Lindström, N. O., Liu, J., Zhu, Q., Guo, J.-J., Alvarado, G. F., Kim, A. D., Black, H. G., Kim, J. & McMahon, A. P. (2019), 'Single-Cell Profiling Reveals Sex, Lineage, and Regional Diversity in the Mouse Kidney', *Developmental Cell* 51(3), 399–413.e7.

Rubanova, Y., Chen, T. Q. & Duvenaud, D. K. (2019), Latent ordinary differential equations for irregularly-sampled time series, in 'Advances in Neural Information Processing Systems', pp. 5321–5331.

Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. (2019), 'A comparison of single-cell trajectory inference methods', *Nature biotechnology* 37(5), 547–554.

Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. (2020), 'Interpretable factor models of single-cell rna-seq via variational autoencoders', *Bioinformatics* 36(11), 3418–3421.

Svensson, V., Vento-Tormo, R. & Teichmann, S. A. (2018), 'Exponential scaling of single-cell rna-seq in the past decade', *Nature protocols* 13(4), 599–604.

Talwar, D., Mongia, A., Sengupta, D. & Majumdar, A. (2018), 'Autoimpute: Autoencoder based imputation of single-cell rna-seq data', *Scientific reports* 8(1), 1–11.

Wang, D. & Gu, J. (2018), 'Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder', *Genomics, proteomics & bioinformatics* 16(5), 320–331.

Wang, J., Agarwal, D., Huang, M., Hu, G., Zhou, Z., Ye, C. & Zhang, N. R. (2019), 'Data denoising with transfer learning in single-cell transcriptomics', *Nature methods* 16(9), 875–878.

Way, G. P. & Greene, C. S. (2017), 'Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders', *BioRxiv* p. 174474.

Wu, S., Liu, S., Sohn, S., Moon, S., Wi, C.-i., Juhn, Y. & Liu, H. (2018), 'Modeling asynchronous event sequences with rnns', *Journal of biomedical informatics* 83, 167–177.

Zhang, C., Bütepage, J., Kjellström, H. & Mandt, S. (2018), 'Advances in variational inference', *IEEE transactions on pattern analysis and machine intelligence* 41(8), 2008–2026.

Zhu, L., Lei, J., Klei, L., Devlin, B. & Roeder, K. (2019), 'Semisoft clustering of single-cell data', *Proceedings of the National Academy of Sciences* 116(2), 466–471.