Published in IEEE BIBM 2019, San Diego, CA, USA, 18-21 November

Recurrent Neural Network for Gene Regulation Network Construction on Time Series Expression Data

Yue Zhao*, Pujan Joshi, Dong-Guk Shin*
Computer Science and Engineering Department
University of Connecticut
Storrs,Connecticut 06269
Email: yue.2.zhao@uconn.edu

***Abstract*—We propose a new way of exploring potential transcription factor targets in which the Recurrent Neural Network (RNN) is used to model time series gene expression data. Once the training of the RNN is completed, inference is performed through feeding the RNN artificially constructed signals. These artificial signals emulate the original gene expression data and the transcriptional factor of interest is set to be zero constantly to model the knockout state of the transcription factor. The predicted expression patterns of the other genes from the RNN are then used to measure the likelihood that the gene is regulated by the knocked out transcriptional factor. After repeating the same process for each gene as Transcription Factor in the dataset, we construct a gene regulation network with edge weights assigned. We demonstrate the effectiveness of our model by comparing our method with existing popular approaches. The result shows that our RNN method can identify transcription factor targets with higher accuracies than most of existing approaches. Overall, our RNN model trained on time series gene expression data can be useful for discovering transcription factor targets as well as building a gene regulation network.**

***Keywords*—Recurrent Neural Network; Gene Regulation Network; Modeling and Simulation**

I. Introduction

Single-cell RNA-Seq (scRNA-Seq) can now quantify the expression of individual cells and it allows scientists to analyze transcriptome differences among cells [1]. Single cell RNA-seq data can be used for cell type identification and cell lineage estimation. Specifically, one can reconstruct the differentiation process by identifying the degree of differentiation of each cell according to timeline [2]. Moreover, correlations of gene expression can be calculated with high accuracy if scRNA-Seq can distinguish the states of individual cells. Especially, the accurate co-expression pattern of each cell type can even reveal the key regulatory factors for lineage programming [3]. Expression dynamics in pseudo-time and accurate relationships among genes have been inferred from scRNA-Seq data [4]. Gene regulatory network (GRN) inference has been performed since pseudo-time can be regarded as interpreting time-dependency information. The key issue is how to convert discovery of gene regulation network from temporal gene expression data. There exist multiple approaches aiming to reconstruct GRN from time-series data in which many assumes the time series as stationary. For this reason, higher resolution inference cannot be performed. Our proposal is using RNN to tackle this particular limitation. In section II, we review the existing approaches for GRN construction. In section III, we focus on methodology illustration. In Section IV, we prove the significance of this approach by comparing the accuracy of discovering GRN against existing approaches with three independent single cell data sets. Finally, in section V, a conclusion is given together with multiple future extension ideas of our method.

II. Related Work

An Boolean network-based algorithms for inferring GRN from single-cell data has been given by [5]. Meanwhile ordinary differentiation equations (ODEs) have been used to describe regulatory network and expression dynamics. Although several ODE-based network-inference algorithms have been proposed [6] [7], most of them assume the time series is steady thus are not suitable for the differentiation case. Several ODE-based algorithms that infer GRNs such that the observed expression dynamics can be reconstructed from the optimized ODE [8]. SCODE, a most recent approach based on ODE was proposed [4]. Previously, we presented a series of methods in which the pathway route is used to discover potential transcriptional targets and extend pathways [9] [10] [11] [12] [13].

III. Methods

A. Framework overview

Given the time series data $\mathbf{x}_t, t = 1 \dots T$, a recurrent neural network (RNN) is fitted to explain the time series thoroughly. After the information of the biological process gets occupied by RNN through training, artificial time series signal will be used as input to the RNN. In the artificial time series signal, a chosen Transcription Factor (TF) is set to be zero (knocked out) while other genes have the same values as original training data. And the prediction of other genes from the RNN will be used to determine if this gene is being regulated by the

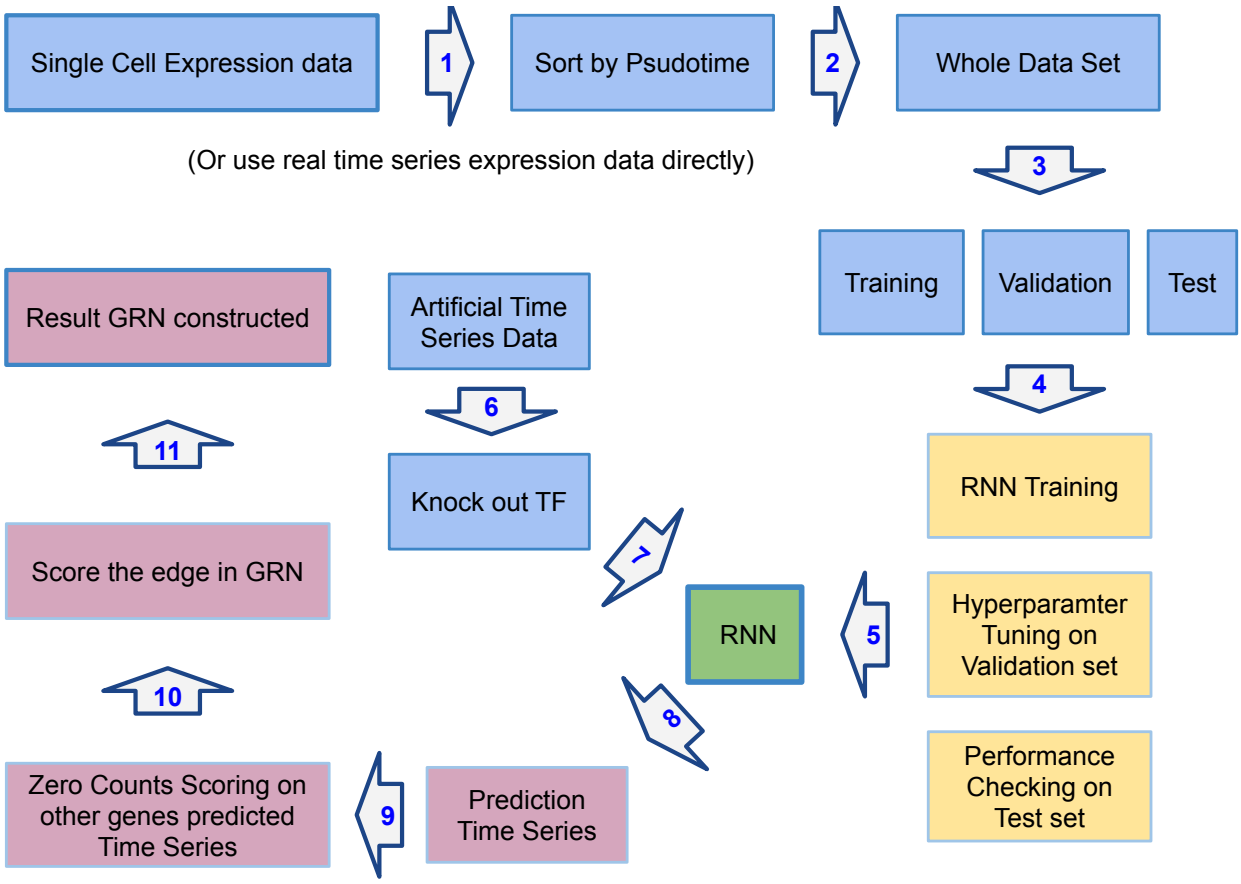

Fig. 1. Workflow overview

knocked out TF by considering how different the predicted expression values are from the original dataset. The workflow is displayed in Figure.1.

B. Recurrent Neural Network

The Recurrent Neural Network architecture is given as follows. The vanilla architecture is used.

$$\begin{aligned} \mathbf{s_t} &= \mathbf{g}(\mathbf{W}_a \begin{bmatrix} \mathbf{s}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_a), t = 1 \dots T \\ \mathbf{y}_t &= \mathbf{g}(\mathbf{W}_y \mathbf{s}_t + \mathbf{b}_y), t = 1 \dots T \end{aligned} \tag{1}$$

where $\mathbf{s}_t$ is the hidden state memorizing the information from all previous time points before time t. The initial state $\mathbf{s}_0 = \mathbf{0}$. Function g is the nonlinear transformation function Rectified Linear Unit (ReLU). Here we specifically select ReLU as the nonlinear activation function because the range of ReLU greatly matches the values of gene expression. In this way, gene suppression is well represented by 0 output from the ReLU. The $\mathbf{W}_a$ and $\mathbf{W}_y$ are weight matrices shared by each time point.

The whole sequence data was firstly divided into training, validation and testing as shown in Fig.2. The training dataset is then fed into the RNN in batches. We use one batch data, $x_1, x_2, \dots, x_m$ to predict the next data point, x_{m+1}. The error of each prediction is then accumulated and used as the Loss for training as shown in (2). The loss function in (2) consists of L2 norm of distance between y_t and x_{t+1} such that the prediction of time point t becomes the data value of the next time point $t+1$, x_{t+1}. Regularization term in the loss function is introduced to penalize $\mathbf{W}_a$ so as to force a sparse structure in the weight matrix, thus only the key regulator genes in the previous steps can affect the next time step values. Other weight matrix is not regularized since they do not use information from $\mathbf{x}_t$ directly. The L1 loss coefficient α is chosen such that the RNN has the best performance on the validation dataset, i.e. the validation dataset has the smallest loss defined by (2). Test set loss is checked in order to have the model have a good prediction power on test data set and avoid overfitting. The loss on validation and test data is calculated in the same way as on the training data, as is illustrated in Fig.2.

$$L = \sum_{t=m-1}^{T-1} ||\mathbf{x}_{t+1} - \mathbf{y}_t||_2 + \alpha ||\mathbf{W}_a||_1 \tag{2}$$

where m is the batch size. Gradient calculation is done by backpropagation through time [14]. And Adam optimization [15] is used to update the weights.

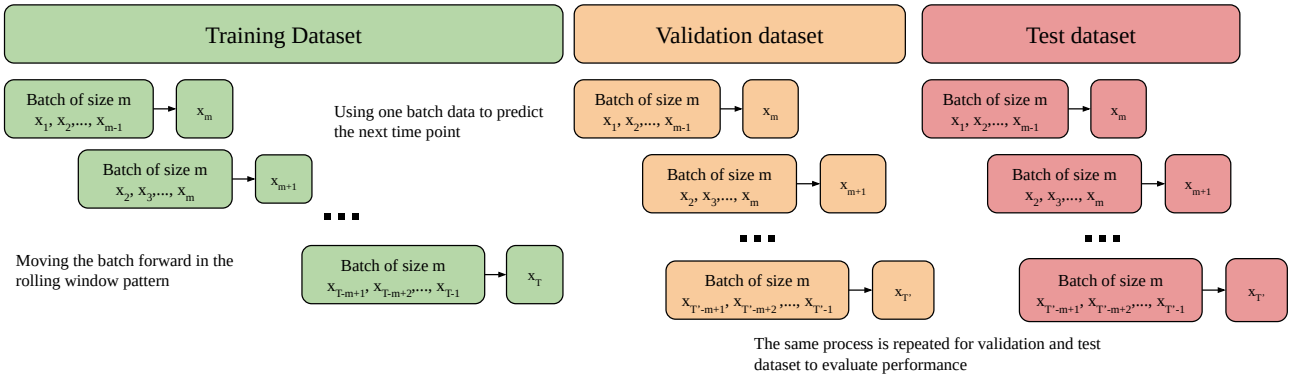

Fig. 2. Data Partition Illustrution: For each batch, the last time point in the batch is predicted based on the previous $m-1$ data points in the batch. T stands for the size of training data while T' is the validation and test data size

After getting a trained RNN, certain biological process in the given time series data is captured by the model. We can further use this RNN as a simulator of the biological process and study the gene behavior at each time step. Please note that this is only one approach to train the RNN, other approaches can also be considered as long as the RNN can be trained properly.

C. Inference by Artificial Signals with a Transcription Factor Knockout

In this subsection, we use the trained RNN to construct a gene regulation network. Initially we assume that the biological mechanism has been learned by the RNN. Then we artificially wipe out one certain TF expression to have expression patterns in its corresponding targets change. More or less, we are performing a gene 'knockout' experiment on each TF and check the responses of other genes through the trained RNN prediction.

Here we generate artificial signals from the original dataset by setting a chosen TF g_i to be 0, meaning the TF g_i is fully knocked out at each time point. The artificially modified data is then fed into the RNN in batches where the batch size is same as the one used during training. The RNN prediction for each gene j is collected and denoted as $\hat{\mathbf{y}}_j$ where $\hat{\mathbf{y}}_j$ is a vector with the size of the number of predictions made. The predicted value at each time point t, will be denoted as $\hat{y}_{jt}$

Different expression patterns should be observed for the knockout TF direct target as long as the RNN truely captures the biological process. Ideally, the direct target gene should be highly affected and the expression levels should be much lower than the original expression in data if the TF is functioning properly and the TF is up regulating this target. On the other hand, if the TF is down regulating the target, high expression of the target is expected due to the knock-out. When we feed the artificial signal where only the TF is "knocked out", the predicted expression level of each gene can be used to measure how likely the gene is regulated by the knocked out TF. Here we use the score in (3) to rank the weight of edge from g_i to g_j in Gene Regulation Network since the more zeros (high expression value that is larger than β) are observed in the predicted expression of g_j, the more likely that g_j is up (down) regulated by the knocked out TF g_i.

$$Score_{ij} = max(\sum_t I(\hat{y}_{jt} = 0), \sum_t I(\hat{y}_{jt} > \beta)) \text{ given } \mathbf{x}_i = \mathbf{0} \tag{3}$$

After repeating the knockout process for each gene in the dataset, we finish constructing a weighted GRN for this time series dataset. In the next section, we compare our approach against several existing methods to show the significance of our proposed framework.

IV. Comparison Study

Our goal is to compare our method against SCODE [4], which can successfully construct regulatory network using single cell time series data. In this study, we will directly compare the result of our work with the results from SCODE and other existing regulatory network discovery approaches. This study shows that our RNN approach is capable of detecting regulatory relationships more accurately than most existing approaches. Our implementation of RNN is done in Python (Tensorflow) [16].

A. Data Preparation

All data sets are single cell RNA-seq data with pseudo-time provided by [4]. In order to compare the result against the one in [4], we use exactly the same dataset provided in the SCODE paper, which contains 100 genes for each dataset. By sorting the cells by pseudo-time estimated by Monocle [2], we end up having three time series data sets. Each data set is used to train an RNN separately. The discovered regulation network for each dataset is compared against the result shown in [4].

1) Data 1: It contains 456 mouse ES cells in the process of transformation to primitive endoderm cells. The first time-course scRNA-Seq dataset analyzed was derived from primitive endoderm (PrE) cells differentiated from mouse ES cells [17].

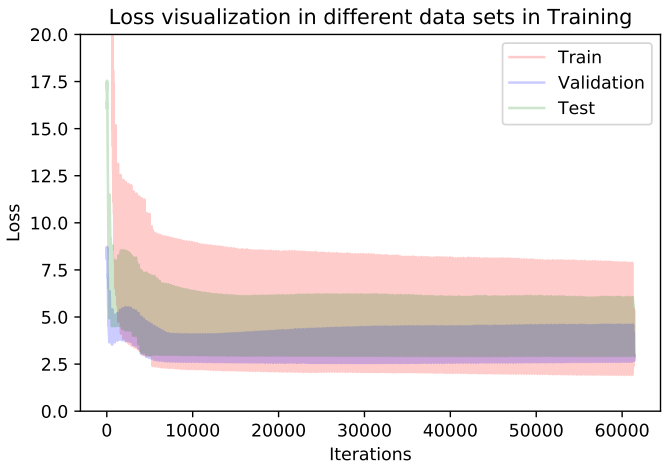

Fig. 3. Loss visualization during RNN training: The y axis represents loss defined by (2), while the x axis represents training iterations using one batch of data. L shape curve on test set indicates that the model is not overfitting

2) Data 2: It contains 405 cell mouse embryonic fibroblast cells differentiating into myocytes. This second dataset was derived from scRNA-Seq data obtained to examine direct reprogramming of mouse embryonic fibroblast (MEF) cells destined to myocytes [18].

3) Data 3: It contains 758 human ES cells differentiating into definitive endoderm cells. The third dataset was a scRNA-Seq time course derived from definitive endoderm (DE) cells differentiated from human ES cells [19].

In this study, we scale the expression of each gene to $(0, 10)$. This speeds up the convergence of the RNN training and converts the expression level to relative expression levels. We did not pick the common choice of $(0, 1)$ because we observe serious gradient vanishing if the small values are used. Use of small values causes a great number of nodes to remain silenced throughout the time course.

B. Hyperparameter Setting and Tuning

In this study, we use the same setting of the hyperparameters for all three datasets. We use batch of 10 continuous data points due to the limitation of data size. The dataset is firstly divided into training, validation and test data sets, where validation and test dataset have $T' = 20$ data points each. Regularization coefficient α in (2) is set to be 0.01 by using the criterion of smallest loss on validation set. Due to the time limitation, hyperparameter tuning is not done for each dataset. Rather a uniformed setting is used and it provides a great performance. The RNN uses hidden state $\mathbf{s}_t$ of the size 2000. This setting allows the model to have an adequate amount of memory to save information from previous time points. This setting without regularization may encounter overfitting, but the good performance on test data, illustrated in Fig.5 appears to show that the model is not overfitting. This figure contains the visualization of real data value with predicted values for the genes in Data 2. Three genes were shown due to the space limitation. And we also show all genes version for Data 3 for completeness. The red line represents the predicted value while the blue line represents the true value. The last 40 data points are 20 validation data points with 20 test data points. We can see that the trained RNN fits test data set very well suggesting a good generality of the RNN model. On the other hand, for overfitting models, the loss for test dataset would get higher as the training continues on for more epochs. This is not observable in Figure.3. The training lasts for 200 epochs. The learning rate is set to be 0.0001. We use threshold $\beta = 9$ in (3) here because the maximum expression value is 10 after scaling.

TABLE I
AUROC FOR EACH EXISTING METHODS

Data	RNN	SCODE	lm	msgps	Cor	GENIE3	Jump3
Data 1	**0.620**	0.536	0.480	0.510	0.505	0.474	0.504
Data 2	**0.587**	0.581	0.489	0.516	0.492	0.472	0.492
Data 3	**0.578**	0.523	0.480	0.499	0.524	0.522	0.501

C. Result

The Area Under Receiver Operating Characteristic Curve (AUROC) is calculated the same way as SCODE [4]. The corresponding reference GRN (provided by [4]) for each dataset is compared against the constructed weighted GRN. The AUROC is calculated by converting the predicted GRN to a binary classification problem on the edges. If there is a directed edge e_{ij} in reference network, then e_{ij} is labeled as 1. Otherwise it is labeled as zero. The weight calculated by (3) of edges in the constructed GRN can be regarded as the likelihood of predicting the edge to be 1. After sorting the weight and the true labels of each edge, True Positive rate and False Positive rate are calculated by selecting a certain threshold, where all the edges with weight score higher than the threshold are predicted to be 1. ROC curve, which is visualized in Fig.6, is generated by selecting different thresholds and visualizing the True Positive rates and False Positive rates. The AUROC is used as the performance measure and is compared with the result of other existing approaches provided [4]. The results are displayed in Table.I. The top most performance is in bold for each dataset. We can see that our RNN approach outperforms 10% better than almost all the methods across all three datasets. The comparison study shows that our RNN approach can provide a better performance than these existing methods.

V. Conclusion

In this work, we introduced a novel way to construct gene regulation network using time series expression data. Our method is compared with several other popular methods and we show that our RNN method is capable of reconstructing reference gene regulation network more accurately. Due to time limitation, we used identical hyper parameter setting for all three datasets. The result could even improve if more proper hyperparameter is set for each dataset.

Actually more potential benefits can be provided by this approach. Imagine a wet lab scientists have limited budget to

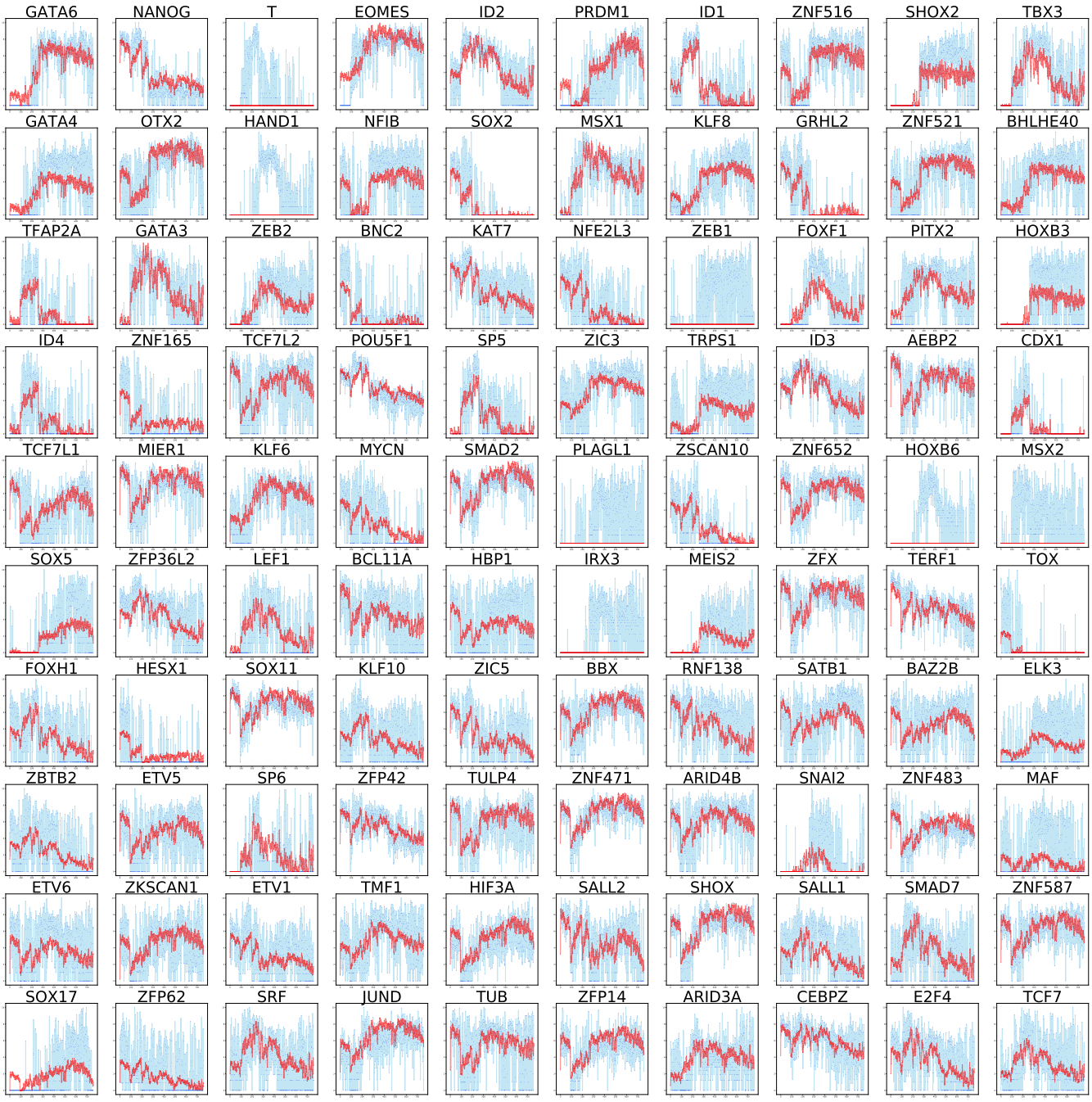

Fig. 4. Prediction vs Real visualization for Data 3: each graph represents one gene. The blue line represents the real data values and the red lines represents predicted value by the RNN. We can see that for most genes, the RNN fits well. However, due to uniform hyperparameter setting, some genes did get biased prediction.

find certain potential targets of a transcription factor, if time series expression data in the same context is available, then this approach will help make good decision on which gene we should spend the money on. Also, as we discussed earlier in the article, this approach may also support co-regulation relationship discovery by feeding multiple TF artificial expression data in. This interesting and exciting work will be performed in the future. We anticipate that our work can be further extended by LSTM since LSTM is proven to be easier to train than the vanilla RNN. Also alternative options for the score measure defined in (3) can be attempted, such as statistical tests [20], since the key idea is to measure the difference between predicted time series against original time series. Applying our method to the analysis of other single cell RNA-seq data sets and real-time time series datasets is another promising future research direction.

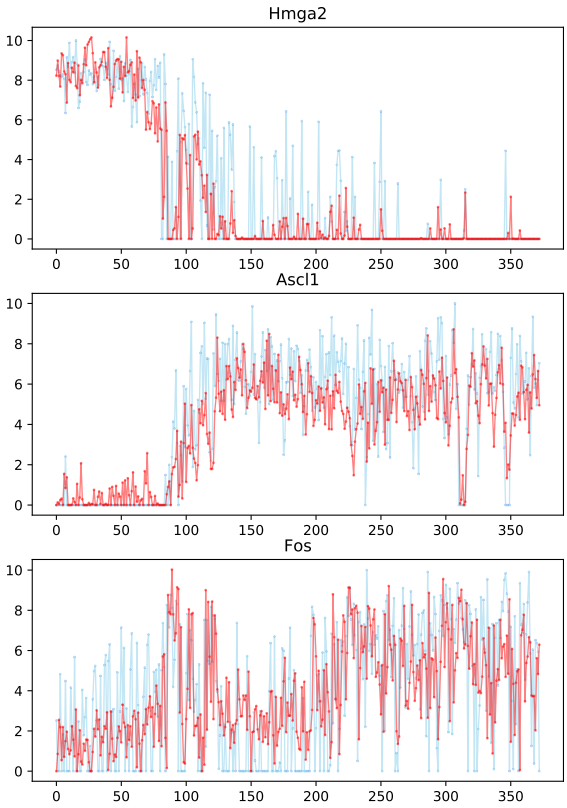

Fig. 5. Prediction vs Real visualization after RNN training

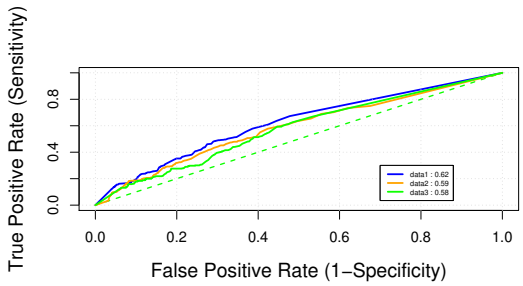

Fig. 6. ROC curve for the three datasets

REFERENCES

[1] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann, "The technology and biology of single-cell rna sequencing," *Molecular cell*, vol. 58, no. 4, pp. 610–620, 2015.

[2] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature biotechnology*, vol. 32, no. 4, p. 381, 2014.

[3] C. Pina, J. Teles, C. Fugazza, G. May, D. Wang, Y. Guo, S. Soneji, J. Brown, P. Edén, M. Ohlsson *et al.*, "Single-cell network analysis identifies ddit3 as a nodal lineage regulator in hematopoiesis," *Cell reports*, vol. 11, no. 10, pp. 1503–1510, 2015.

[4] H. Matsumoto, H. Kiryu, C. Furusawa, M. S. Ko, S. B. Ko, N. Gouda, T. Hayashi, and I. Nikaido, "Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation," *Bioinformatics*, vol. 33, no. 15, pp. 2314–2321, 2017.

[5] C. Y. Lim, H. Wang, S. Woodhouse, N. Piterman, L. Wernisch, J. Fisher, and B. Göttgens, "Btr: training asynchronous boolean models using single-cell expression data," *BMC bioinformatics*, vol. 17, no. 1, p. 355, 2016.

[6] D. Di Bernardo, T. S. Gardner, and J. J. Collins, "Robust identification of large genetic networks," in *Biocomputing 2004*. World Scientific, 2003, pp. 486–497.

[7] T. S. Gardner, D. Di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, 2003.

[8] M. Bansal, G. D. Gatta, and D. Di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics*, vol. 22, no. 7, pp. 815–822, 2006.

[9] Y. Zhao, T. H. Hoang, P. Joshi, S.-H. Hong, and D.-G. Shin, "Deep pathway analysis incorporating mutation information and gene expression data," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2016, pp. 260–265.

[10] Y. Zhao, T. H. Hoang, P. Joshi, S.-H. Hong, C. Giardina, and D.-G. Shin, "A route-based pathway analysis framework integrating mutation information and gene expression data," *Methods*, vol. 124, pp. 3–12, 2017.

[11] Y. Zhao, S. Piekos, T. Hoang, and D.-G. Shin, "A framework using topological pathways for deeper analysis of transcriptome data," in *Bioinformatics Research and Applications: 14th International Symposium, ISBRA 2018, Beijing, China, June 8–11, 2018, Proceedings*. ISBRA, 2018.

[12] Y. Zhao, "An extension of deep pathway analysis: A pathway route analysis framework incorporating multi-dimensional cancer genomics data," in *International Symposium on Bioinformatics Research and Applications*. Springer, 2018, pp. 113–124.

[13] T. H. Hoang, Y. Zhao, Y. Lam, S. Piekos, Y.-C. Han, C. Reilly, P. Joshi, S.-H. Hong, C. O. Sung, C. Giardina *et al.*, "Biotarget: A computational framework identifying cancer type specific transcriptional targets of immune response pathways," *Scientific reports*, vol. 9, no. 1, p. 9029, 2019.

[14] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural networks*, vol. 1, no. 4, pp. 339–356, 1988.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[17] D. Shimosato, M. Shiki, and H. Niwa, "Extra-embryonic endoderm cells derived from es cells induced by gata factors acquire the character of xen cells," *BMC developmental biology*, vol. 7, no. 1, p. 80, 2007.

[18] B. Treutlein, Q. Y. Lee, J. G. Camp, M. Mall, W. Koh, S. A. M. Shariati, S. Sim, N. F. Neff, J. M. Skotheim, M. Wernig *et al.*, "Dissecting direct reprogramming from fibroblast to neuron using single-cell rna-seq," *Nature*, vol. 534, no. 7607, p. 391, 2016.

[19] L.-F. Chu, N. Leng, J. Zhang, Z. Hou, D. Mamott, D. T. Vereide, J. Choi, C. Kendziorski, R. Stewart, and J. A. Thomson, "Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm," *Genome biology*, vol. 17, no. 1, p. 173, 2016.

[20] J. Serra and J. L. Arcos, "An empirical evaluation of similarity measures for time series classification," *Knowledge-Based Systems*, vol. 67, pp. 305–314, 2014.