

RESEARCH

Open Access

scDLC: a deep learning framework to classify large sample single-cell RNA-seq data



Yan Zhou¹, Minjiao Peng¹, Bin Yang¹, Tiejun Tong², Baoxue Zhang³ and Niansheng Tang^{4*}

Abstract

Background: Using single-cell RNA sequencing (scRNA-seq) data to diagnose disease is an effective technique in medical research. Several statistical methods have been developed for the classification of RNA sequencing (RNA-seq) data, including, for example, Poisson linear discriminant analysis (PLDA), negative binomial linear discriminant analysis (NBLDA), and zero-inflated Poisson logistic discriminant analysis (ZIPLDA). Nevertheless, few existing methods perform well for large sample scRNA-seq data, in particular when the distribution assumption is also violated.

Results: We propose a deep learning classifier (scDLC) for large sample scRNA-seq data, based on the long short-term memory recurrent neural networks (LSTMs). Our new scDLC does not require a prior knowledge on the data distribution, but instead, it takes into account the dependency of the most outstanding feature genes in the LSTMs model. LSTMs is a special recurrent neural network, which can learn long-term dependencies of a sequence.

Conclusions: Simulation studies show that our new scDLC performs consistently better than the existing methods in a wide range of settings with large sample sizes. Four real scRNA-seq datasets are also analyzed, and they coincide with the simulation results that our new scDLC always performs the best. The code named “scDLC” is publicly available at <https://github.com/scDLC-code/code>.

Keywords: Single-cell RNA sequencing, Deep learning, Classifier

Background

The development of RNA sequencing (RNA-seq) has enabled unprecedented insight into the dynamics of gene expression [1–7]. In contrast to microarray data, next-generation sequencing data improve the specificity and sensitivity of gene expression and have been increasingly popular in biological and medical research, such as detecting differentially expressed genes and identifying which type of diseases a new patient belongs to with gene expression. In recent years, a single-cell RNA-sequencing (scRNA-seq), allowing sequencing to be conducted on the level of single cells, has become another standard tool in biological and medical studies [8–12]. The scRNA-

seq data analysis not only discovers new cell types, but also reveals the deep regulatory networks [13–16]. Among them, cell type identification is an important task in scRNA-seq data analysis [17]. In a general way, we identify cell types with unsupervised clustering within scRNA-seq data and then do the manual annotation based on a set of known marker genes [18]. In practice, we rarely know the number of clusters in advance, and the annotation of clusters is also somewhat subjective [19]. This may lead to bias in the analysis of the better characterised cell types. In contrast, supervised learning methods can identify the cell types more accurately and also reduce the bias associated with marker gene selection in cell type annotation.

For the classification of RNA-seq data, several statistical methods have been developed [20, 21], in particular for the bulk RNA-seq experiments. Poisson and negative binomial distributions are two most commonly used distributions to model the discrete RNA-seq data.

Yan Zhou and Minjiao Peng are shared first authorship.

*Correspondence: nstang@ynu.edu.cn

⁴Yunnan Key Laboratory of Statistical Modeling and Data Analysis, Yunnan University, Kunming, China

Full list of author information is available at the end of the article



© The Author(s). 2022 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

Witten [22] assumed that the RNA-seq data follow a Poisson distribution and proposed the Poisson linear discriminant analysis (PLDA). Dong et al. [23] took into account the overdispersion of the RNA-seq data and proposed the negative binomial linear discriminant analysis (NBLDA). Note also that RNA-seq data may have excess zeros, especially when the sequence depth is not enough. Zhou et al. [24] further proposed the zero-inflated Poisson logistic discriminant analysis (ZIPLDA) with a point mass at zero when classifying RNA-seq data.

Nowadays, scRNA-seq data have been increasingly used to identify cell types and disease states for new patients. Yet to the best of our knowledge, there are still relatively few methods in the literature to classify scRNA-seq data despite the enormous potential of scRNA-seq data. Generally, low sequencing depths cause high noise levels and a large fraction of so-called “dropout” events in scRNA-seq data; and moreover, classification methods for bulk RNA-seq data may cause unacceptably large misclassification rates for scRNA-seq data. Especially for scRNA-seq data with relative large sample sizes, they may follow a more complex mixed distribution. Most existing classification methods for RNA-seq data require a certain distribution assumption, and they may fail in improving the classification accuracy for scRNA-seq data with large sample sizes. Alquicira-Hernandez et al. [25] developed a novel classification method based on singular value decomposition and a support vector machine model for scRNA-seq data. Zhao et al. [26] reviewed the existed classification tools for scRNA-seq data. Lin et al. [27] proposed a scClassify method by using a distance weighted kNN classifier. More recently, Wang and Li [28] proposed a scale-invariant deep-neural-network classifier (SINC) method which is based on deep neural-network (DNN) to classify scRNA-seq data. Their method provides a new way to dig more information for large sample size and also a novel thinking of scale invariant for next generation sequencing data. From another perspective, however, we note that the SINC method does not consider the dependency between the feature genes so that the settings may not be very realistic.

In this paper, we consider a deep learning classifier (scDLC) to identify cell types for large sample scRNA-seq data, which is based on the two-layer long short-term memory recurrent neural networks (LSTMs). The deep learning classifier can learn scRNA-seq data without the need of a distribution assumption. What's more, the scDLC method considers the dependency between the feature genes in the process of classification. LSTMs [29] is a special kind of recurrent neural network which can learn long-term dependencies of a sequence. For scRNA-seq data, scDLC can automatically learn each sample of the class as a gene sequence.

Our scDLC framework for identifying cell types in scRNA-seq data can be summarized as four steps. For the first fully connected layer, the gene sequences of a sample are mapped to a larger dimension. The first step aims to enlarge the information of gene sequence and make the class difference more obvious. Second, the output of the first fully connected layer is taken as the input of the two-layer long short-term memory network layer, and the weights of all gates are estimated by network calculation in each class. Third, we reduce the output dimensions to the number of classes in the second fully connected layer. Lastly, the outputs of the second fully connected layer are transformed to a probability distribution with a softmax function. In the process of training, we compare the probability of each class to the observation and estimate the optimization parameters under the cross-entropy loss function.

To summarize the main advantages of the scDLC framework for classifying scRNA-seq data with large sample sizes, we note that scDLC is applicable to all scRNA-seq data no matter what the underlying distribution is. Moreover, scDLC has the capacity to capture the difference information of gene sequence from different classes, which is another key reason why it can perform the best compared to the existing competitors. In **Methods**, we propose the framework of scDLC and further describe the estimation of parameters in details. In Simulation studies, we conduct simulation studies to evaluate the performance of the new classifier and compare it with existing methods. In Application to Real Data, we apply the proposed method to analyze four real scRNA-seq datasets to demonstrate its usefulness in practice. We then conclude the paper in Discussion with some discussion and future directions.

Results

We propose a deep learning framework (scDLC) based on the LSTMs model to classify scRNA-seq data. The details of the scDLC model have been shown in **Methods**. To validate the performance of proposed method, we consider simulation studies and real data analysis. All the R scripts that analysed the data have been uploaded at github, which could be accessible at <https://github.com/scDLC-code/scDLC>.

Simulation studies

In this section, we evaluate the performance of the proposed scDLC method via simulation studies. To generate scRNA-seq read count data, we apply the Splatter Bioconductor package [30] that is known to be simple, reproducible and well-documented. While for comparison, we also consider seven other methods including PLDA, NBLDA, ZIPLDA, the support vector machines (SVM), scPred, scClassify and the SINC method.

Simulation design

In each experiment, we generate n samples for the training set and another n samples for the test set. We first consider the binary classification with $K = 2$. Study 1 investigates the effect of different sample sizes for the binary classification. We fix the proportions of differentially expressed genes $DE = 0.5$, the probability of excess zeros $p_{zero} = 0.2$, and consider the gene number $g = 100, 200, 300$ and 400 . We then compute the misclassification rates of all methods with different sample sizes ranging from 100 to 900. In Study 2, we evaluate the performance of all methods when the proportions of differentially expressed genes are 0.2, 0.3, 0.4, 0.5, 0.6 and 0.7 with fixed sample size $n = 200, 300, 400$ and 500 . In addition, we set the probability of excess zeros $p_{zero} = 0.2$ and the gene number $g = 100$. In Study 3, we test the performance of all methods with the different probability of excess zeros, including $p_{zero} = 0.1, 0.2, 0.3, 0.4, 0.5$ and 0.6 . For other settings, we let the gene number $g = 100$, the sample size $n = 200, 300, 400$ and 500 , and 40% of genes be differentially expressed.

For the multiple classification with $K = 3$, we also conduct three studies to evaluate the performance of the different methods. In Study 4, we evaluate the effect of different sample sizes with three classes. All other parameters are kept the same as those in the binary classification except for the sample sizes. We set $n = 300, 400, 500$ and 600 for three classes in Studies 5 and 6, respectively.

Simulation results

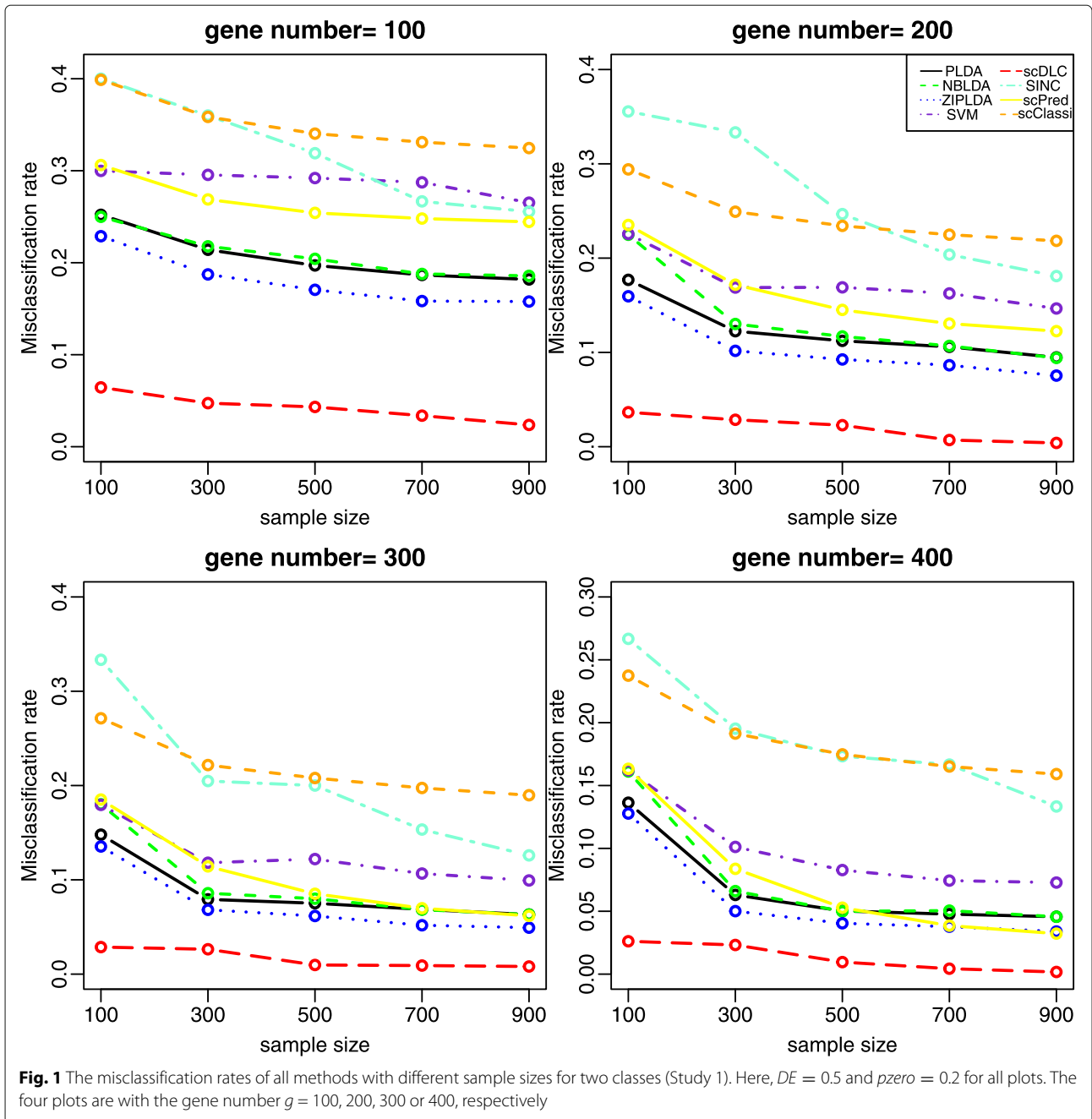
With 1000 simulations for each experiment, we report the average misclassification rates for the binary classification in Figs. 1-2 and Supplementary Fig. S1, respectively. The results for the multiple classification are presented in Supplementary Figs. S2-S4. Figure 1 shows that the misclassification rates of all the considered methods decrease as the sample size increases. It is also evident that scDLC performs much better than the other methods in all cases. Figure 2 shows that the misclassification rates of all methods are decreased with an increasing number of differentially expressed genes, and meanwhile scDLC shows its superiority over the other methods. From Supplementary Fig. S1, we note that an increasing probability of excess zeros will yield a higher misclassification rate and the proposed method again outperforms the other methods in all settings.

Supplementary Figs. S2 to S4 display the simulation results for the multiple classification with $K = 3$. They coincide with the conclusions made for the binary comparison, and in particular, scDLC always performs the best. Moreover, we note that SINC does not perform well when the number of selected feature genes is small, and so it can only be recommended for large number of selected feature genes.

Application to real data

To further evaluate the performance of the different classifiers, we also analyze six scRNA-seq datasets which are from National Center for Biotechnology Information Search database (NCBI, <https://www.ncbi.nlm.nih.gov/>). The six datasets are summarized in Table 1. The first dataset GSE99933 was released in Furlan et al. [31]. It is used to demonstrate that large numbers of chromaffin cells arise from peripheral glial stem cells. This dataset has two classes, including 384 samples recombining at E12.5 and 384 samples recombining at E13.5. The second dataset GSE123454 illustrates the high information content of nuclear RNA for characterization of cellular diversity in brain tissues [32]. This dataset includes 463 samples from single nuclei and 463 samples from matched single cells with measurements on 42003 genes. The third dataset GSE113069 is a testament to the diversity of subiculum pyramidal cells from the hippocampus [33]. It contains three classes, each with 345, 422, 423 samples, respectively. The fourth dataset GSE84133 Baron1 was created by Baron et al. [34], and was further analyzed by the deep-neural-network classifier SINC [28]. Baron1 contains all major cell groups from the first human donor, excluding those with less than 20 cells. It contains nine classes, each with 110, 51, 236, 872, 214, 120, 130, 70 and 92 samples, respectively. The last two datasets are large sample datasets which contain tens of thousands of cells. Specifically, the fifth dataset GSE107585 was used to reveal potential cellular targets of kidney disease [35]. It came from healthy mouse kidneys, containing total 43745 cells for all fifteen classes, each with 26482, 8544, 1729, 1581, 1308, 1001, 870, 643, 549, 313, 235, 228, 110, 78 and 74 samples, respectively. The sixth dataset PBMC can be downloaded from the Single Cell Portal with accession numbers SCP424 in https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data [36]. The dataset was from human organism that contains 31021 cells for all thirteen classes, each with 7805, 6437, 4391, 3529, 2881, 2197, 1466, 908, 620, 372, 203, 149, 52, and 11 samples, respectively.

We assess the performance of our proposed scDLC method with seven baseline methods, including three traditional classifiers based on the Bayesian, scPred, scClassify, SVM and SINC methods. We apply the AUC score, which is the area surrounded by the coordinate axis under the ROC curve [37], to measure the performance of the classifiers. We randomly draw 40 to 450 of the samples to build the training set, and regard the rest as the test set. In real data, the majority of genes are not differentially expressed and they are irrelevant for class distinction. For example, we observe in Fig. 2 that the large rate of feature genes for class distinction will improve the accuracy of the classifiers. Thus to improve the rate of feature genes, we follow Zhou et al. [24] to select the top p feature genes



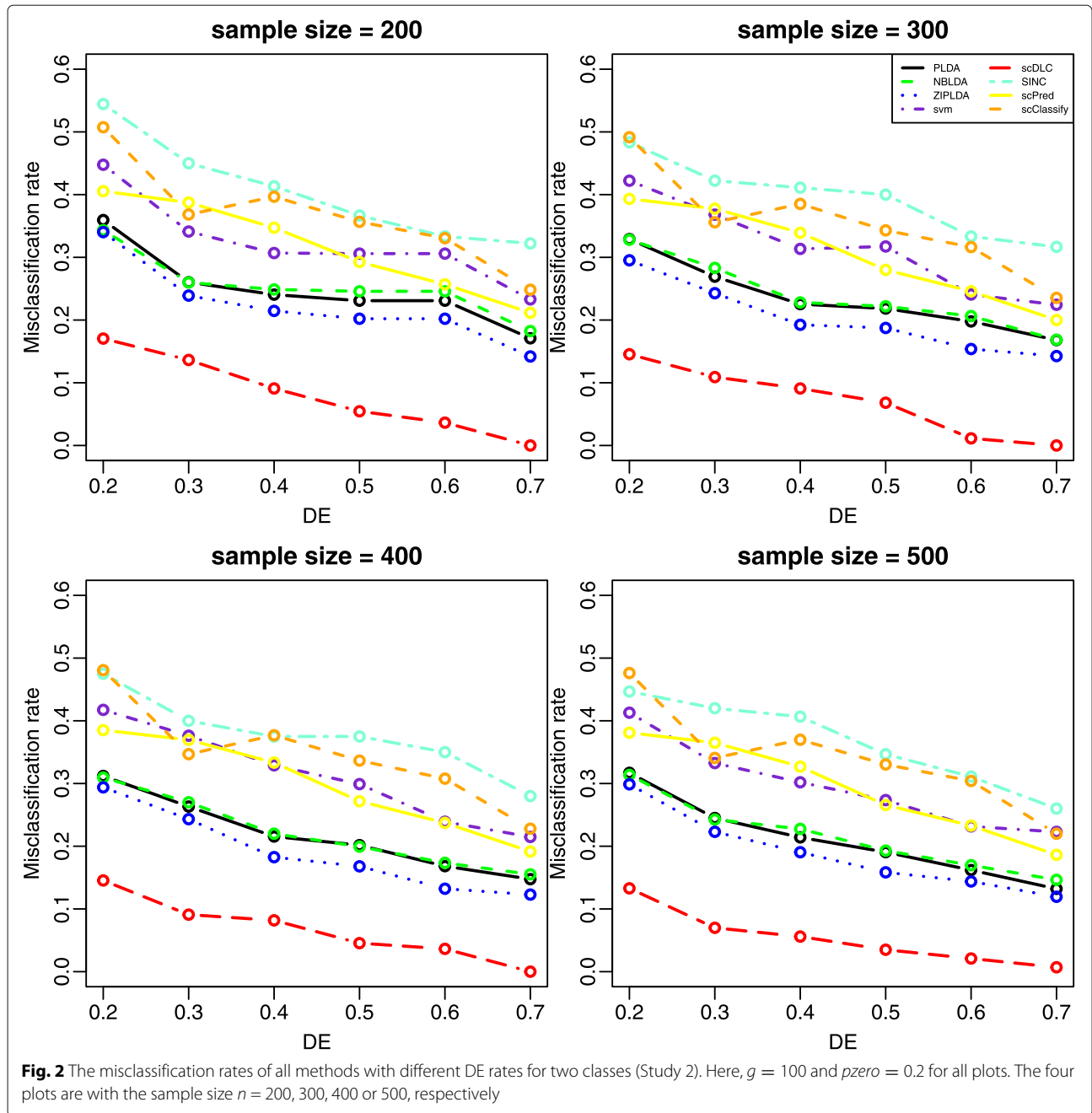
from the training set using the BW method. Specifically for the j th gene, the BW value is defined as the ratio of the sum of squares between groups (BSS) to that within groups (WSS) as follows:

$$BW(j) = \frac{\sum_{k=1}^K \sum_{i=1}^{n_k} (\bar{x}_{k,j} - \bar{x}_{\cdot,j})^2}{\sum_{k=1}^K \sum_{i=1}^{n_k} (x_{kij} - \bar{x}_{\cdot,j})^2}, \tag{1}$$

where $\bar{x}_{\cdot,j} = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} \sum_{i=1}^{n_k} x_{kij}$ is the averaged expression values across all samples, $\bar{x}_{k,j} = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{kij}$ is the averaged expression value across samples belonging to

class k , and K is the number of classes. Moreover, without loss of generality, we retain the top $p = 100$ feature genes from each simulation as the inputs of the first layer of scDLC. We further repeat all the experiments 100 times and calculate the average AUC scores. We also present their respective boxplots in Fig. 3 with the AUC scores. From the boxplots, it is evident that our proposed scDLC outperforms the baseline methods for all four datasets.

Next, we compare the performance of all classifiers with different sizes of training samples. Figure 4 shows the AUC scores of the eight methods with different sizes



of training samples for the first four real datasets with small sample size. The number of feature genes is fixed at 100 and the training sample size varies from 40 to 450. From Fig. 4, although the AUC scores of the proposed method are not outstanding when the training sample size is smaller than 50, it is still the best classifier on the whole. In particular, when the training sample size is larger than 100, our scDLC is consistently better than all other methods. As shown in Figs. 3 and 4, ScPred is comparable to scDLC for GSE99933 and GSE123454 datasets and they are both better than the other methods, which contain

Table 1 Details of the four scRNA-seq datasets

Datasets	Sample size	No. of classes	No. of genes
GSE99933	768	2	23420
GSE123454	926	2	42003
GSE113069	1190	3	23218
GSE84133(Baron1)	1895	9	20126
GSE107585	43745	15	16272
PBMC	31021	13	29669

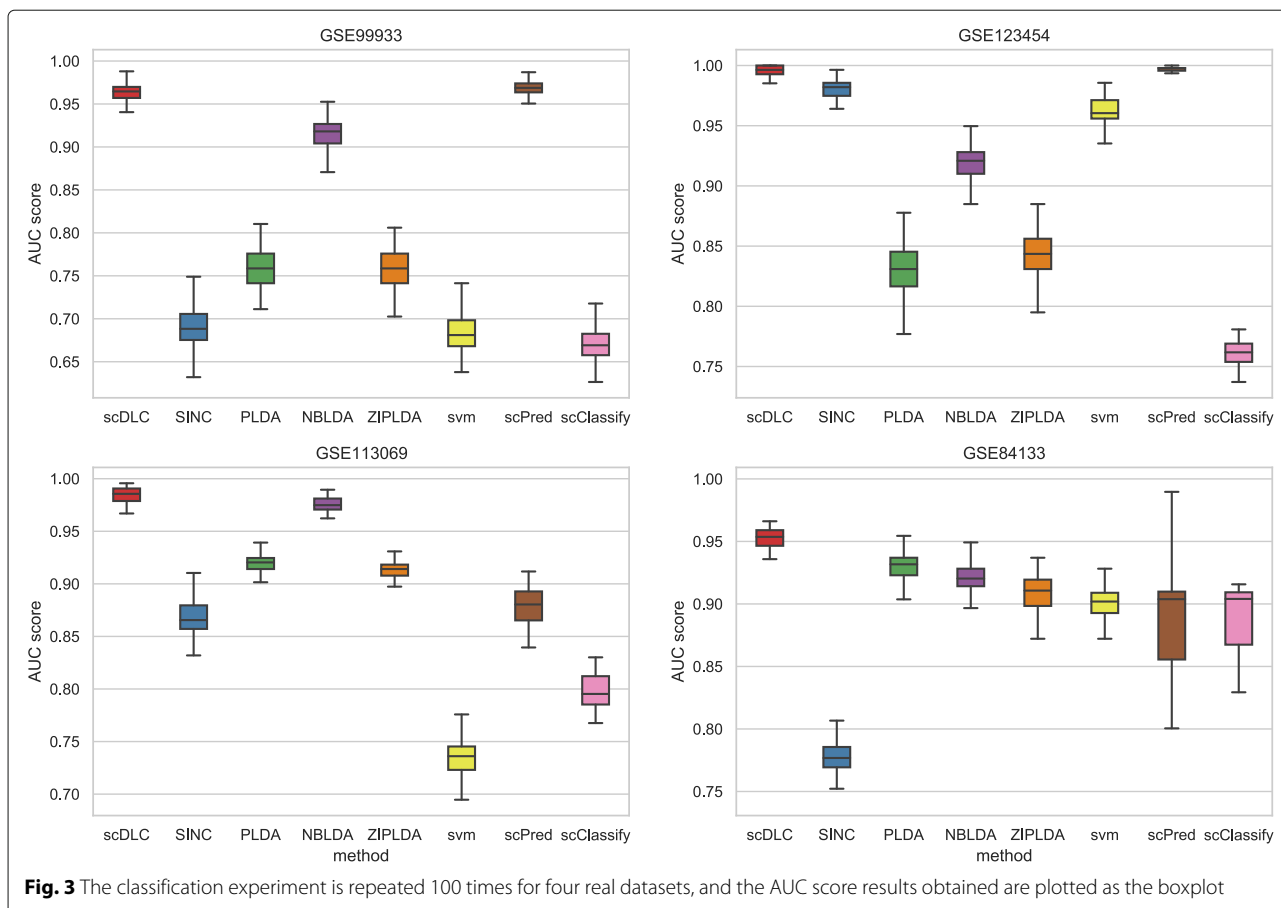


Fig. 3 The classification experiment is repeated 100 times for four real datasets, and the AUC score results obtained are plotted as the boxplot

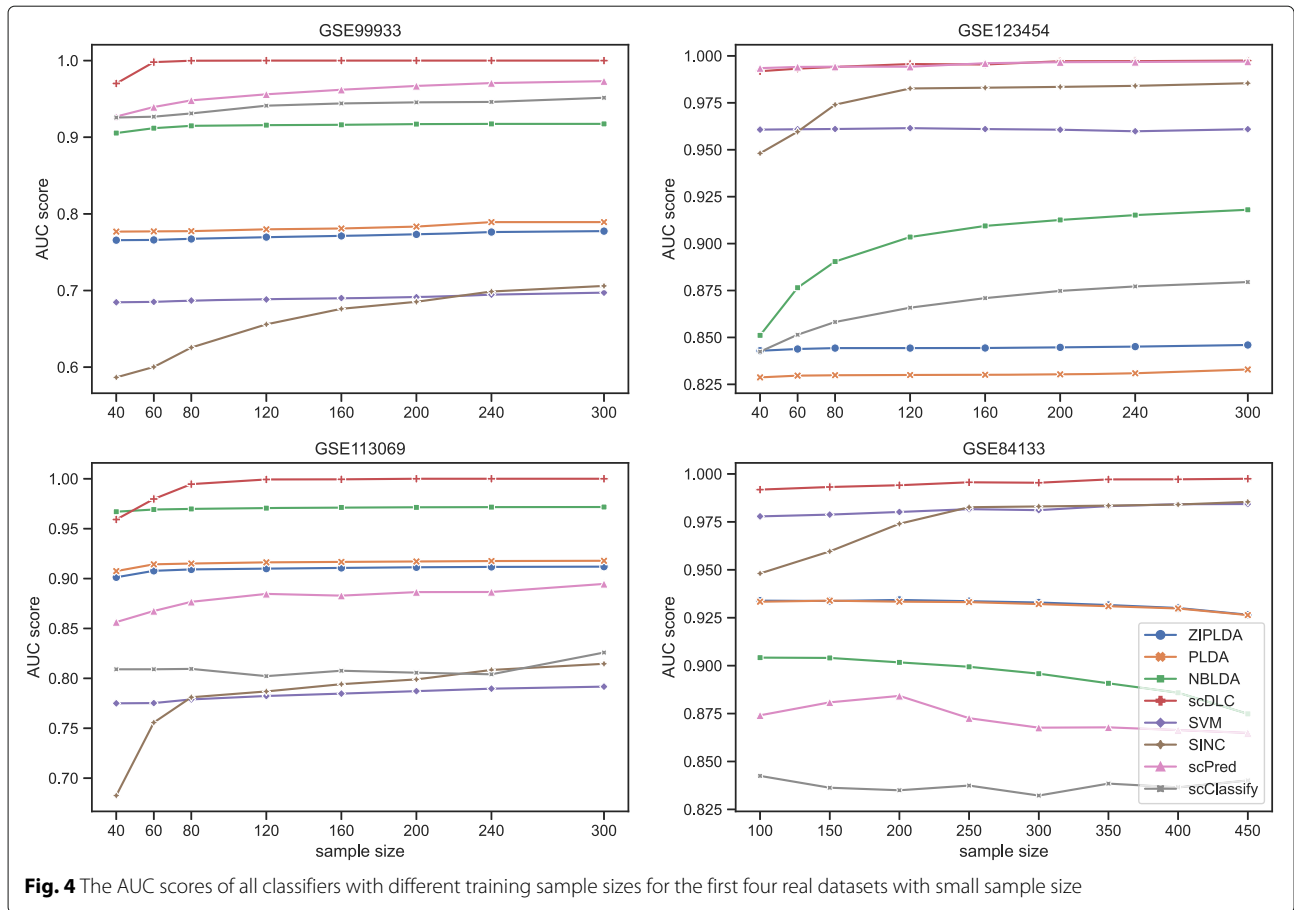
only two cell types. Figure 5 shows the AUC scores of the eight methods with different sizes of training samples for the last two real datasets with large sample. The number of feature genes is fixed at 100 and the training sample size varies from 1200 to 12000. From Fig. 5, the proposed method outperforms the existing methods for large training sample in the two real datasets. The AUC scores of SVM are less than those of our scDLC but much higher than the other methods.

Finally, we consider the performance of each classifier under different selected feature genes. Specifically, we use 70% of the dataset as the training set and the rest as the test set. According to the degree of differential expression, the top 20 to 100 genes are selected to test the performance of each classification method. Figure 6 and Supplementary Figs. S5-S7 show the AUC scores of the eight methods with different selected feature genes. For the GSE123454 and GSE99933 datasets in Fig. 6 and Supplementary Fig. S5, the scPred method is comparable to the scDLC method and much better than the other methods. However, NBLDA is comparable to the scDLC method in Supplementary Fig. S7. In Supplementary Figs. S6 and S7, we observe a similar result that the scDLC

method outperforms the other methods in the GSE84133 and GSE113069 datasets. The four Figures show that the comparison results of the classifiers are relatively consistent under different choices of the selected genes and the proportion. Finally, it is noteworthy that the AUC scores of scDLC are not affected much by the number of feature genes.

Discussion

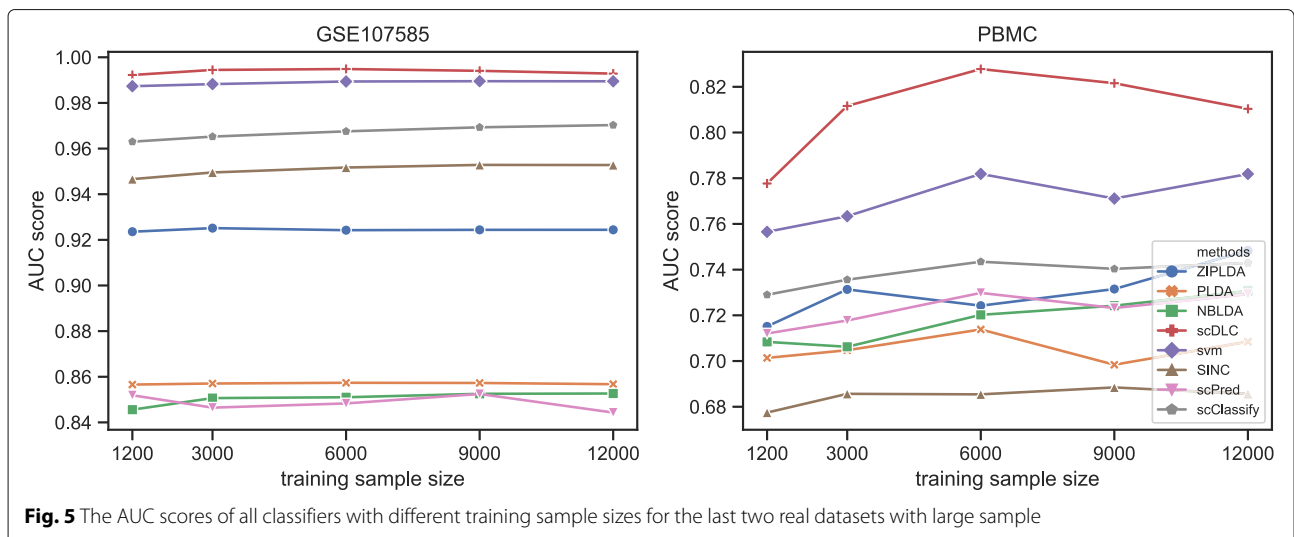
The single-cell RNA sequencing (scRNA-seq) technology has been increasingly used in molecular diagnosis of clinical diseases. In this paper, we proposed a deep learning framework with two layers of LSTMs, namely scDLC, to classify large sample scRNA-seq data. The innovation of scDLC is mainly manifested in two aspects. Firstly, compared to the existing discriminant rules, our new method does not require a distribution assumption so that it can be widely applied in practice. Secondly, our scDLC also amplifies the features of the selected genes through the first fully connected layer. It is thus beneficial to improve the classification accuracy and stability of the model, and meanwhile our scDLC can be trained with less computer resource using only the top selected feature genes.

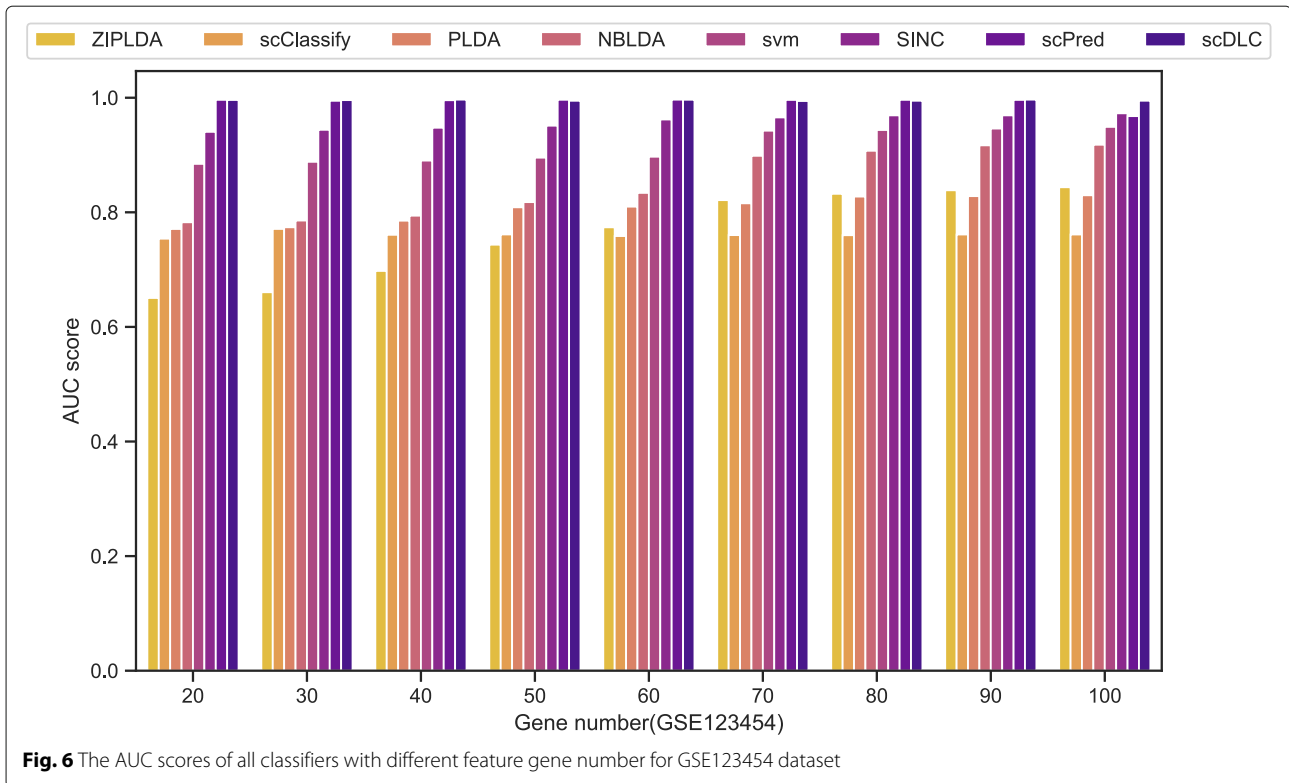


To evaluate the performance of our new classifier, we considered both the binary classification and the multiple classification. Simulation results show that our deep learning method can sufficiently capture the difference information of classes in gene sequences, and that it performs much better than, or at least as well as, the existing

competitors in a wide range of settings with large sample sizes. We also analyzed six real scRNA-seq datasets, including both small and large sample sizes, and they all support that our new scDLC always performs the best.

As a future work, we will study from the network structure level why scDLC can efficiently capture class





differences from gene sequences, and we expect that understanding the mechanism can bring deep insights to gene expression and regulation. Moreover, it can also be interesting to extend deep learning techniques to conduct in-depth research in precision medicine such as neonatal genetic disease-related gene screening.

Methods

We first review the framework of long short-term memory recurrent neural networks (LSTMs), and then introduce a new workflow of the deep learning classifier (scDLC) for large sample size scRNA-seq data.

Hochreiter and Schmidhuber [29] proposed a recurrent neural network with long short-term memory network. This network has a great performance to solve the sequential data related learning problem. LSTMs can effectively capture both short-term and long-term time dependence. Sak et al. [38] showed that the long short-term memory network is effective for acoustic modeling. Marchi et al. [39] proposed a bidirectional LSTMs for audio onset detection. Due to the gate mechanism, LSTMs solves the problem of gradient vanishing which cannot be overcome by the simple recurrent neural network. The early LSTMs was refined and popularized by many people in the following work. The structure of this model was further improved by Graves et al. [40] based on the previous research [41, 42]. The core idea of the LSTMs is several non-linear gating units that control information retention and forgetting, as well as a memory cell that can maintain

its state over time. As shown in Fig. 7, it includes a single cell, two tanh activation blocks and three gates (input gate, forget gate, output gate). The input gate controls the input information and whether the input will be read. The forget gate controls the internal state information and whether the current cell value is forgotten. The output gate controls the output information and whether new cell values are output. The input of the three gates is the output of the previous time and the input of the current time. The activation function of three gates is the sigmoid function. Let x_t , h_t and C_t denote the input value, the output value and the cell state at time t , respectively. Let b denote the bias term, and W denote the weight matrix. Let also f , i and o denote the forget gate, the input gate and the output gate, respectively. The recurrent process of LSTMs can be expressed as follows:

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
 C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 h_t &= o_t * \tanh(C_t),
 \end{aligned}
 \tag{2}$$

where \tilde{C}_t is a vector of new candidate values, $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function, and $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ is the tanh function. In addition, “ $*$ ” represents the matrix

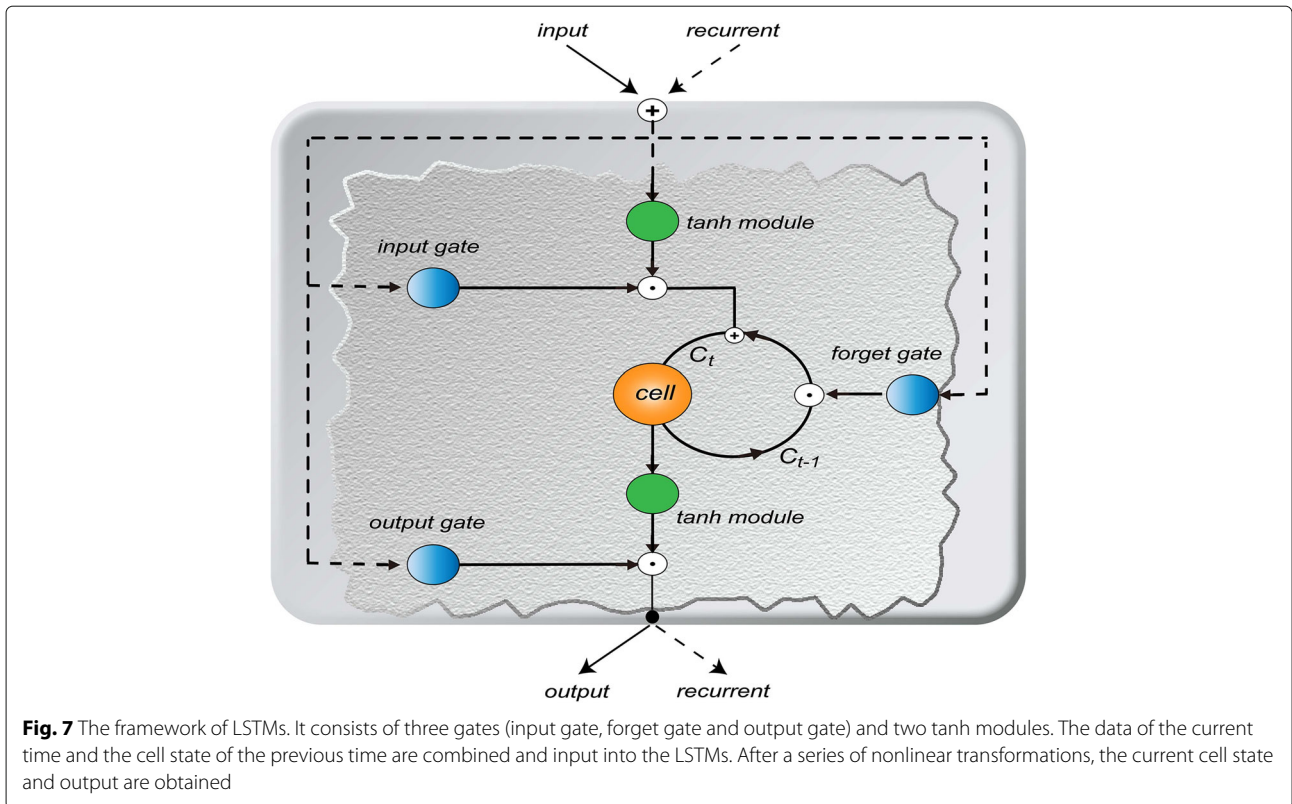


Fig. 7 The framework of LSTMs. It consists of three gates (input gate, forget gate and output gate) and two tanh modules. The data of the current time and the cell state of the previous time are combined and input into the LSTMs. After a series of nonlinear transformations, the current cell state and output are obtained

multiplication and “*” represents the multiplication with scalars.

Deep learning classifier for scRNA-seq data

The scDLC framework is shown in Fig. 8, which includes two fully connected layers and a two-layer LSTMs. The fully connected layers are located at the first layer and the last layer, respectively. After the model training, it results in a scRNA-seq data classifier. Inputting a gene sequence sample into scDLC, the probability that the gene sequence sample belongs to each class will be obtained. Finally, we identify which class the sample belongs to based on the probability vector.

Fully connected layers: Each node of the fully connected layer is connected to all nodes of the previous layer. It can synthesize the extracted features through the rectified linear unit (*ReLU*) activation function. The function of the first fully connected layer in scDLC is to amplify the information of the gene sequence and make the class difference more obvious. This layer can greatly improve the accuracy of discrimination. The *ReLU* activation function is

$$a = \max(0, Wx + b), \quad (3)$$

where x is the input vector, W is the weight matrix, b is the bias vector, and a is the activation vector which is the output of the fully connected layer. Using the *ReLU*

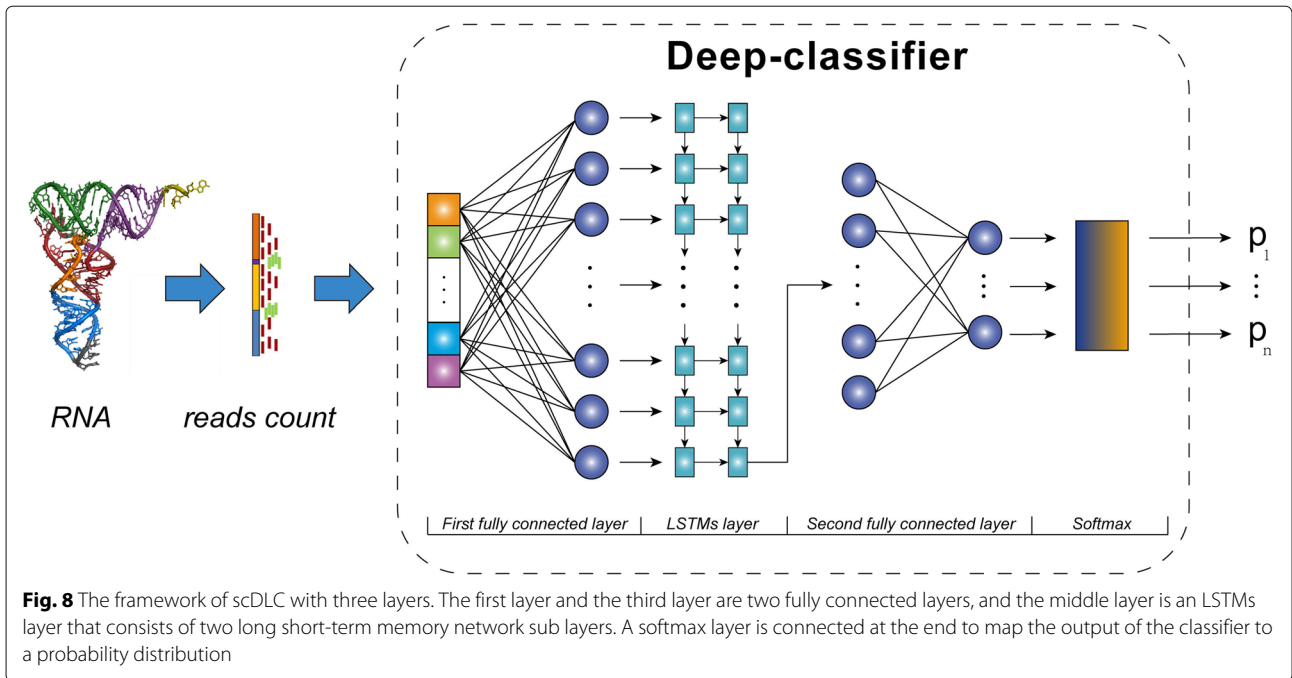
activation function in the network can make the classifier perform better. At the end of the model, we map the output of the second fully connected layer to the probability distribution of the class through a *softmax* function as

$$\text{softmax}(y_c) = \frac{e^{y_c}}{\sum_{j=1}^M e^{y_j}}, \quad (4)$$

where M is the number of classes.

LSTMs layer: In the LSTMs layer, we take two LSTMs sublayers to learn data. The horizontal connection between sublayers means that the output h of the first sublayer is entered into the second sublayer as input. The vertical connection means that the cell state C of the previous time is transferred to the next time in the same sublayer. The output of this layer will be used as the input to the second fully connected layer. The forward recursions of this layer refer to the formulas in (Fig. 7).

The trainable parameters (all weights and biases) in this deep model are denoted as θ . The partial derivatives $\partial L / \partial \theta$ of the loss function L with respect to any trainable parameter in the network can be calculated by the back propagation algorithm [43]. We further take the cross entropy as the loss function since it can well describe the difference between the true probability distribution and the predicted probability distribution. To be specific, we define the loss function as



$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^M y_{ic} \log(p_{ic}), \tag{5}$$

where N is the sample size, M is the number of classes, y_{ic} is an indication variable which is 1 if class c is the same as the class of the sample or otherwise 0, and p_{ic} represents the prediction probability that sample i belongs to class c .

The gradient descent method is a widely used optimization algorithm in machine learning. We use a mini-batch gradient descent algorithm (MBGD) [44] to train our model. For a set of training samples, MBGD does not use all the training samples to calculate the real gradient of the target, but instead calculates the gradient of a small batch samples. We then minimize the loss function by updating the trainable parameter θ . According to the MBGD algorithm, the rule for updating is as follows:

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}, \tag{6}$$

where η is the learning rate. In order to avoid fluctuation in the later stage of training, we further set the learning rate decay exponentially during the training. That is

$$\tilde{\eta} = \eta e^{\gamma/s}, \tag{7}$$

where $\tilde{\eta}$ is the learning rate after decay, γ is the decay rate, and s is the global step. The exponential-decay learning rate means that the learning rate is correlated with the number of training times, and it will decline exponentially with the increase of training times. Here, r is the decay rate, s is the global step, the maximum learning rate is set to $max_lr = 0.005$, the minimum learning rate is set to $min_lr = 0.001$, epoch is the training times, x is the

sample size in the total training set, $batch_size$ represents the sample size in a batch, then decay rate is computed with $r = \log(max_lr/min_lr)/(epoch * x/batch_size)$. Then the learning rate after decay can be obtained according to the calculated $decay_rate$.

Hyperparameter settings

To implement the proposed scDLC, it is further needed to determine the hyperparameters in the model. Note that the hyperparameters are the configuration outside the model, and their values cannot be estimated from the data. Appropriate hyperparameters can greatly improve the performance of the model. According to the test of different hyperparameter combinations, we set the following parameters that can yield a good performance for the classification.

hidden size=64: The parameter represents the size of the hidden state of LSTMs and we set it as 64.

batch size=11: For the number of samples in a batch, we randomly choose 11 samples throughout the simulations.

grad clip=5: To stabilize the network in the process of training, we set the threshold as 5 for the gradient to control the weight update within a certain range.

train keep prob=0.3: To prevent overfitting, we let the train keep probability equal to 0.3, which means that only 30% of the information will be used in the next time.

initial learning rate=0.005: For the appropriate learning rate that can make the objective function converge to a local minimum at a suitable time, we set the initial learning rate as 0.005. Since the learning rate will decline with training, we further set the minimum learning rate as 0.001.

Abbreviations

scDLC: Deep learning classifier for large sample scRNA-seq data; LSTMs: Long short-term memory recurrent neural networks; scRNA-seq: Single-cell RNA sequencing; FDR: False discovery rate; ROC: Receiver operating characteristic; AUC: Area under the curve

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12864-022-08715-1>.

Additional file 1: Supplementary figures and tables. This file contains related figures and tables for simulated and real datasets.

Acknowledgements

Our sincere thanks go to the editor and two reviewers for their valuable comments and helpful suggestions that have led to substantial improvements of the article.

Authors' contributions

YZ and NT conceived the idea. BZ and BY processed the data and conducted simulation and real dataset experiments. YZ and TT wrote the manuscript. YZ, MP, BZ, TT and NT revised the manuscript. All authors read and approved the final manuscript.

Funding

Yan Zhou's research was supported by the National Natural Science Foundation of China (Grant No. 12071305, 11871390 and 11871411), Natural Science Foundation of Guangdong Province of China under grant 2020B1515310008, Project of Educational Commission of Guangdong Province of China under grant 2019KZDZX1007. Niansheng Tang's research was supported by the National Natural Science Foundation of China (Grant No. 11731011). Tiejun Tong's research was supported by the General Research Fund (HKBU12303918), the National Natural Science Foundation of China (1207010822), and the Initiation Grant for Faculty Niche Research Areas (RC-FNRA-IG/20-21/SCI/03) of Hong Kong Baptist University.

Availability of data and materials

The datasets are from National Center for Biotechnology Information Search database (NCBI, <https://www.ncbi.nlm.nih.gov/>). The first dataset GSE99933 was released in [31]. The second dataset GSE123454 illustrates the high information content of nuclear RNA for characterization of cellular diversity in brain tissues [32]. The third dataset GSE113069 is a testament to the diversity of subiculum pyramidal cells from the hippocampus [33]. The fourth dataset GSE84133 Baron1 was created by [34]. The fifth dataset GSE107585 was released in [35]. The sixth dataset PBMC can be downloaded from the Single Cell Portal with accession numbers SCP424 [36]. All the R scripts that analysed the data are available at <https://github.com/scDLC-code/scDLC>. Additional supporting Figures and Tables are included as Additional files.

Declarations

Ethics approval and consent to participate

Not applicable. Humans, animals or plants have not been directly used in this study.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹College of Mathematics and Statistics, Institute of Statistical Sciences, Shenzhen Key Laboratory of Advanced Machine Learning and Applications, Shenzhen University, Shenzhen, China. ²Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. ³School of Statistics, Capital University of Economics and Business, Beijing, China. ⁴Yunnan Key Laboratory of Statistical Modeling and Data Analysis, Yunnan University, Kunming, China.

Received: 3 September 2021 Accepted: 21 June 2022

Published online: 12 July 2022

References

- Mardis ER. Next-Generation DNA sequencing methods. *Annu Rev Genomics Hum Genet.* 2008;9(1):387–402.
- Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet.* 2009;10(1):57–63.
- Morozova O, Hirst M, Marra MA. Applications of new sequencing technologies for transcriptome analysis. *Annu Rev Genomics Hum Genet.* 2009;10(1):135–51.
- Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods.* 2008;5(7):621–8.
- Nagalakshmi U, Wang Z, Waern K, Shou C, Raha D, Gerstein M, et al. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science.* 2008;320(5881):1344–9.
- Wilhelm BT, Landry JR. RNA-Seq-quantitative measurement of expression through massively parallel RNA-sequencing. *Methods.* 2009;48(3):249–57.
- Ozsolak F, Milos PM. RNA sequencing: advances, challenges and opportunities. *Nat Rev Genet.* 2011;12(2):87–98.
- Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, et al. mRNA-Seq whole-transcriptome analysis of a single cell. *Nat Methods.* 2009;6(5):377–82.
- Picelli S, Björklund ÅK, Faridani OR, Sagasser S, Winberg G, Sandberg R. Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nat Methods.* 2013;10(11):1096–8.
- Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell.* 2015;161(5):1187–201.
- Macosko EZ, Basu A, Satija R, Nemeshe J, Shekhar K, Goldman M, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell.* 2015;161(5):1202–14.
- Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. *Nat Commun.* 2017;8(1):14049.
- Darling EM, Guilak F. A neural network model for cell classification based on single-cell biomechanical properties. *Tissue Eng A.* 2008;14(9):1507–15.
- Ding B, Zheng L, Zhu Y, Li N, Jia H, Ai R, et al. Normalization and noise reduction for single cell RNA-seq experiments. *Bioinformatics.* 2015;31(13):2225–7.
- Diaz A, Liu SJ, Sandoval C, Pollen A, Nowakowski TJ, Lim DA, et al. SCell: integrated analysis of single-cell RNA-seq data. *Bioinformatics.* 2016;32(14):2219–20.
- Miao Z, Deng K, Wang X, Zhang X. DEsingle for detecting three types of differential expression in single-cell RNA-seq data. *Bioinformatics.* 2018;34(18):3223–4.
- Trapnell C. Defining cell types and states with single-cell genomics. *Genome Res.* 2015;25:1491–8.
- Wang B, Zhu J, Pierson E, Ramazzotti D, Batzoglou S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods.* 2017;14:414–6.
- Grün D, Oudenaarden A. Design and analysis of single-cell sequencing experiments. *Cell.* 2015;163:799–810.
- Tan KM, Petersen A, Witten D. Classification of RNA-seq data. Statistical analysis of next generation sequencing data. Cham: Springer; 2014, pp. 219–46.
- Zhou Y, Wang J, Zhao Y, et al. Discriminant Analysis and Normalization Methods for Next-Generation Sequencing Data. *New Frontiers of Biostatistics and Bioinformatics.* Cham: Springer; 2018, pp. 365–84.
- Witten DM. Classification and clustering of sequencing data using a Poisson model. *Ann Appl Stat.* 2011;5(4):2493–518.
- Dong K, Zhao H, Tong T, Wan X. NB-LDA: negative binomial linear discriminant analysis for RNA-Seq data. *BMC Bioinformatics.* 2016;17(1):369.
- Zhou Y, Wan X, Zhang B, Tong T. Classifying next-generation sequencing data using a zero-inflated Poisson model. *Bioinformatics.* 2018;34(8):1329–35.
- Alquicira-Hernandez J, Sathé A, Hanlee PJ, Nguyen Q, Powell JE. scPred: accurate supervised method for celltype classification from single-cell RNA-seq data. *Genome Biol.* 2019;20:264.

26. Zhao X, Wu S, Fang N, Sun X, Fan J. Evaluation of single-cell classifiers for single-cell RNA sequencing data sets. *Brief Bioinforma.* 2020;21(5):1581–95.
27. Lin Y, et al. scClassify: sample size estimation and multiscale classification of cells using single and multiple reference. *Mol Syst Biol.* 2020;16:e9389.
28. Wang C, Li J. SINC: a scale-invariant deep-neural-network classifier for bulk and single-cell RNA-seq data. *Bioinformatics.* 2020;36(6):1779–84.
29. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997;9(8):1735–80.
30. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* 2017;18(1):174.
31. Furlan A, Dyachuk V, Kastri ME, Calvo-Enrique L, Abdo H, Hadjab S, et al. Multipotent peripheral glial cells generate neuroendocrine cells of the adrenal medulla. *Science.* 2017;357(6346):eaal3753.
32. Bakken TE, Hodge RD, Miller JA, Yao Z, Nguyen TN, Aebermann B, et al. Single-nucleus and single-cell transcriptomes compared in matched cortical cell types. *PLoS ONE.* 2018;13(12):e0209648.
33. Cembrowski MS, Wang L, Lemire AL, Copeland M, DiLisio SF, Clements J, et al. *eLife.* 2018;7:e37701.
34. Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.* 2016;3(4):346–60.
35. Park J, Shrestha R, Qiu CX, Kondo A, et al. Single-cell transcriptomics of the mouse kidney reveals potential cellular targets of kidney disease. *Science.* 2018;360(6390):758–63.
36. Ding JR, Adiconis X, Simmons SK, Kowalczyk MS, et al. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nat Biotechnol.* 2020;38:737–746.
37. Lobo JM, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models. *Glob Ecol Biogeogr.* 2008;17(2):145–51.
38. Sak H, Senior AW, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014. <https://research.google/pubs/pub43905.pdf>.
39. Marchi E, Ferroni G, Eyben F, et al. Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2014. p. 2164–8.
40. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Int Joint Conf Neural Netw.* 2005;18:602–10.
41. Gers FA, Schmidhuber JA, Cummins FA. Learning to forget: continual prediction with LSTM. *Neural Comput.* 2000;12(10):2451–71.
42. Gers FA, Schmidhuber J. Recurrent nets that time and count. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol 3. IEEE; 2000. p. 189–94. <https://ieeexplore.ieee.org/abstract/document/861302>.
43. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature.* 1986;323:533–6.
44. Ruder S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:160904747.* 2016.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

