Artificial Intelligence in the Life Sciences 1 (2021) 100017

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

AutoGenome: An AutoML tool for genomic research

Denghui Liu[a,1], Chi Xu[a,1], Wenjun He[a,1], Zhimeng Xu[a], Wenqi Fu[a], Lei Zhang[a], Jie Yang[a], Zhihao Wang[e], Bing Liu[e], Guangdun Peng[b,c], Dali Han[d], Xiaolong Bai[a], Nan Qiao[a,*]

[a] Lab of Health Intelligence, Huawei Technologies Co., Ltd, China
[b] CAS Key Laboratory of Regenerative Biology and Guangdong Provincial Key Laboratory of Stem Cell and Regenerative Medicine, Guangzhou Institutes of Biomedicine and Health, Chinese Academy of Sciences, Guangzhou, China
[c] Guangzhou Regenerative Medicine and Health Guangdong Laboratory, Guangzhou, China
[d] Key Laboratory of Genomic and Precision Medicine, Beijing Institute of Genomics, Chinese Academy of Sciences, Beijing, China
[e] BioBank, The First Affiliated Hospital of Xi'an Jiaotong University, Shaanxi 710061, China

ARTICLE INFO

Keywords:
Autogenome
Automl
Residual fully-connected neural network
Genomic
Deep learning

ABSTRACT

Deep learning has achieved great successes in traditional fields like computer vision (CV), natural language processing (NLP), speech processing, and more. These advancements have greatly inspired researchers in genomics and made deep learning in genomics an exciting and popular topic. The convolutional neural network (CNN) and recurrent neural network (RNN) are frequently used to solve genomic sequencing and prediction problems, and multiple layer perception (MLP) and auto-encoders (AE) are frequently used for genomic profiling data like RNA expression data and gene mutation data. Here, we introduce a new neural network architecture-the residual fully-connected neural network (RFCN)-and describe its advantage in modeling genomic profiling data. We also incorporate AutoML algorithms and implement AutoGenome, an end-to-end, automated deep learning framework for genomic studies. By utilizing the proposed RFCN architecture, automatic hyper-parameter search, and neural architecture search algorithms, AutoGenome can automatically train high-performance deep learning models for various kinds of genomic profiling data. To help researchers better understand the trained models, AutoGenome can assess the importance of different features and export the most critical features for supervised learning tasks and the representative latent vectors for unsupervised learning tasks. We expect AutoGenome will become a popular tool in genomic studies.

1. Introduction

Over the past few decades, the emergence of high-throughput sequencing technology has revolutionized biomedical research, and the continuous development of different methods has generated vast amounts of omics data, providing comprehensive information for all kinds of genomic studies, ranging from general genomics to specialized subfields. For instance, in general genomics, microarray [1] and next-generation DNA-Seq [2] are widely used to identify genome-wide copy number variations (CNVs), single-nucleotide polymorphisms (SNPs), and DNA mutations. In the epigenomics area, MeDip-Seq [3] and BS-Seq [4] are used to profile DNA methylations, and ChIP-Seq is used to identify chromatin associated proteins [5]. In transcriptomics, microarray [1] and RNA-Seq [6] are used to quantify whole RNA expression profiles. In proteomics, LC-MS [7] and ICAT [8] are used to analyze protein complexes and quantify proteins. In metabolomics, MNR [9] and mass spectrometry [10] are used to profile metabolic markers. These techniques and methods have been widely used in biomedical researches [11,12], and many bioinformatics tools have been developed for data analysis.

Deep learning has proven to be very effective in computer vision [13,14], natural language processing [15], speech processing [16–18], and other traditional fields. By using a well-designed, deep, stacked neural network architecture, low-, middle-, and high-level features can be extracted automatically and combined to make end-to-end predictions. Inspired by the successes of using deep learning in traditional fields, researchers are now actively designing neural network architectures to leverage deep learning in the field of genomic study (Fig. 1a).

The convolutional neural network [14] (CNN) and recurrent neural network [16] (RNN) are widely used neural network architectures. CNN can effectively extract multiple-dimension spatial features from 1-D to N-D data [14,19,20], and RNN can capture long short-term features [21] from time series data. By treating DNA/RNA/protein sequences as image or sequential data, CNN and RNN can also be used to model

* Corresponding author.
E-mail address: qiaonan3@huawei.com (N. Qiao).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.ailsci.2021.100017
Received 1 October 2021; Received in revised form 18 November 2021; Accepted 18 November 2021
Available online 22 November 2021

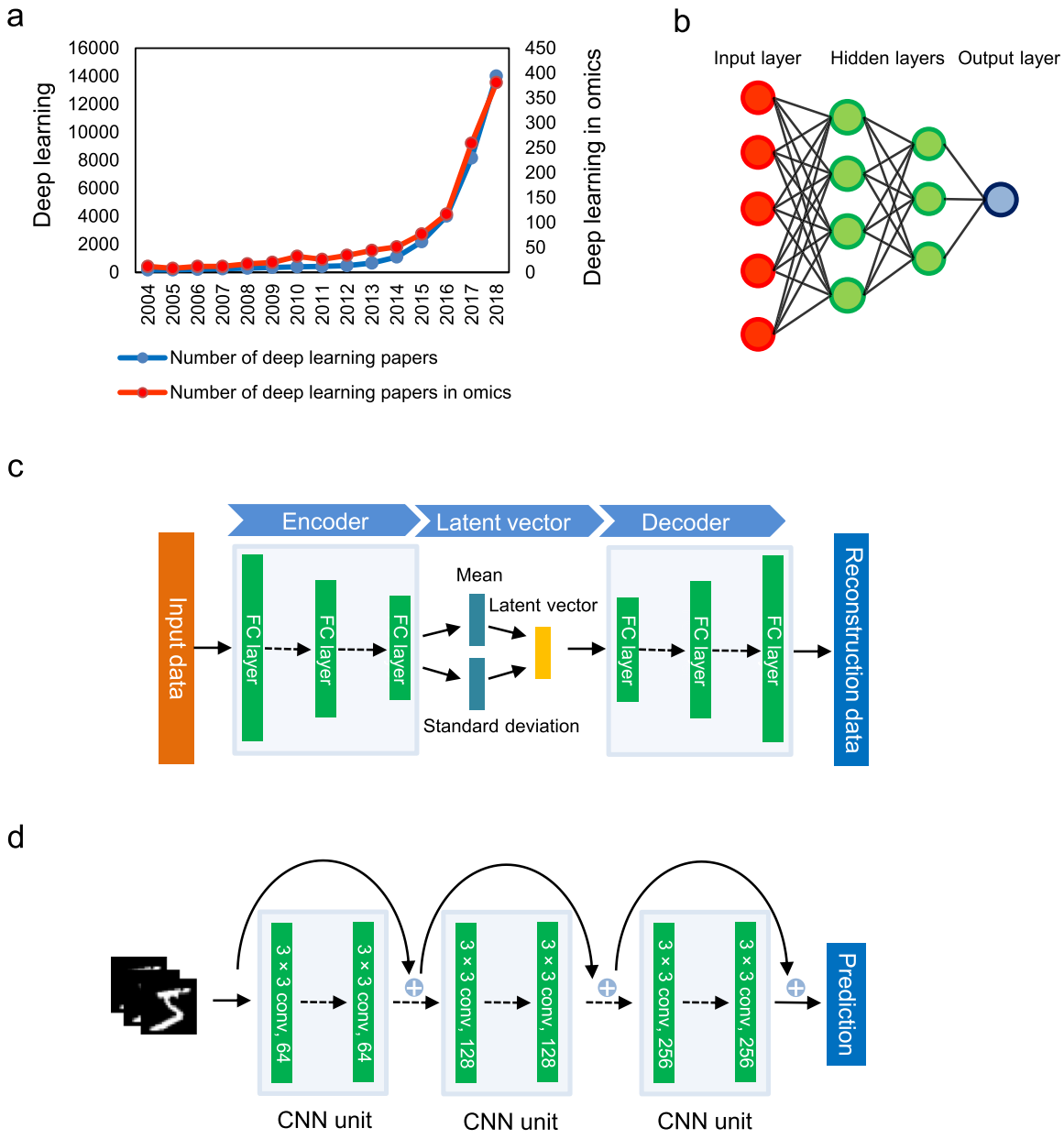

Fig. 1. Introduction to deep learning neural network. (a) Publication trends of deep learning papers vs. deep learning papers in genomics. The data is from https://apps.webofknowledge.com. (b) Multiple layer perception (MLP) architecture. Information from the previous layer is used as input for the next layer. (c). Variational autoencoder (VAE) architecture. The Encoder compresses raw data into latent vectors by decreasing number of neurons in each layer. The Decoder reconstructs the raw data from latent vectors. (d). Residual convolutional network architecture. The residual network involves skip-connections. The basic unit in the network is convolutional layers.

genome sequencing problems like variant discovery [22], DNA motif finding [23], sgRNA on-target site prediction [24,25], protein-protein-interaction prediction [26,27], and drug design [28–30].

However, CNN and RNN cannot be used for non-sequence data like genomic profiling data, where RNA expression values, gene mutation status, and gene copy number variations from whole genomes are profiled, because there are no spatial or temporal relationships in the data. For genomic profiling data, the underlying mechanism is the gene regulatory pathway/network: multiple genes can interact with one another (to activate, inhibit, or enhance gene expression) and compose a structured hierarchical network to regulate the biological functions [31], which is the key to modeling genomic profiling data through deep learning.

Several studies were conducted using multiple layer perception (MLP), autoencoders (AE) or variational autoencoders (VAE) for supervised or unsupervised tasks on genomic profiling data (Fig. 1b–c), during which researchers observed improvements in comparison with traditional machine learning algorithms [32–34]. However, these methods also have shortcomings. As deep learning tasks in computer vision show the importance of network depth [14,35,36], MLP and VAE face the vanishing gradient problem [37,38], which stands in the way of training deeper neural networks. Most of the published research articles about using deep learning in genomics have less than four hidden layers (Fig. S4a).

In machine learning, hyper-parameter selection and network architecture selection usually require specialized knowledge, and machine learning methods are very challenging for people who are not experts to use. The development of automated machine learning (AutoML) aims to simplify these processes and democratize AI for all. Automatic model

selection, feature selection, hyper-parameter search, and neural architecture search are commonly used in AutoML [39–43]. However, in genomics, most studies are still limited to handcrafted DNN structures and parameter space.

To address these challenges and further facilitate genomic studies using deep learning, we propose AutoGenome, an end-to-end AutoML framework for genomic study. In AutoGenome, we propose the RFCN and its variants and validate that they outperform MLP and VAE. We also use hyper-parameter search and the efficient network architecture search [41] (ENAS) algorithm in AutoGenome to enable automatic search for brand-new neural network architectures. We will show that AutoGenome can be easily programmed to train customized deep learning models, evaluate their performances, and interpret the results for genomic profiling data.

2. Results

RFCN outperforms MLP in genomic profiling data modeling

Looking back at the success of deep learning in the ImageNet challenge [44], we can see that the increasing network depths played an important role [14,35,36]. However, simply increasing the network depth can cause the vanishing gradient problem [37,38]. A residual network was then proposed to resolve this problem by adding a shortcut at each layer, achieving smooth gradient backpropagation (Fig. 1d). ResNets [45], HighwayNets [46] and DenseNets [47] are major variants of the residual network. The residual network also enhances feature propagation and encourages feature reuse.

In order for the residual network to work for genomic profiling data, we propose an RFCN. Unlike previous residual networks [45,47], which use the convolution layer as the basic unit, the RFCN uses the fully-connected layer as the basic unit, and each layer may connect to the next layer directly (Fig. 2a left, Path 1), through a skip connection (Fig. 2a left, Path 2), or through a branch (Fig. 2a left, Path 3). The basic fully-connected (FC) unit is shown in Fig. 2a right. The FC unit comprises a sequence of a fully-connected layer, batch normalization layer, and activation function layer.

By replacing the CNN unit with the FC unit in the standard form of ResNet/DenseNet, we introduce the RFCN variants: RFCN-ResNet and RFCN-DenseNet (Fig. 2b and c). In RFCN-ResNet, each FC unit takes the summation of the previous unit's output and input as its own input (Fig. 2b). In RFCN-DenseNet, each FC unit takes the concatenation of all the outputs from previous units as its input (Fig. 2c). To simplify the use of RFCN-ResNet and RFCN-DenseNet for researchers, we have implemented the algorithms in AutoGenome, which can be used to search for the best RFCN-ResNet/RFCN-DenseNet structures and hyper-parameters automatically (see Methods).

To assess the performance of RFCN-ResNet and RFCN-DenseNet, we designed a series of experiments on two independent datasets and compared the performances with XGBoost [48] (a popular machine learning algorithm) and AutoKeras [39] (a popular AutoML tool). In the first experiment, we used gene expression profiles from 8127 samples, corresponding to 24 tumor types from the TCGA [49] dataset, to perform a pan-cancer classification task; somatic mutation profiles from 6186 TCGA samples (24 tumor types) were also used to perform pan-cancer classification independently. The second experiment used the gene expression profiles from 10,000 single mouse cells, covering 10 different embryonic developmental stages (downloaded from a public dataset [50]) to perform a classification task. After preprocessing the datasets (details in Methods), we trained different models with XGBoost, AutoKeras, RFCN-ResNet, and RFCN-DenseNet separately and evaluated their performances with an independent test set. The results showed that for pan-cancer classification tasks using gene expression profiles, the RFCN-ResNet and RFCN-DenseNet achieved 95.9% and 95.7% accuracy, respectively, outperforming XGBoost and AutoKeras by 4.8% and 4.9%, respectively (Fig. 2c); with only genomic somatic mutation profiles, the RFCN-ResNet and RFCN-DenseNet achieved 64.1% and 60.1% accuracy, respectively, outperforming XGBoost and AutoKeras by 5.7% and 19.1%, respectively (Fig. S1a). In the second experiment, RFCN-ResNet and RFCN-DenseNet achieved 95.9% and 96.5% accuracy, respectively, outperforming XGBoost and AutoKeras by 6.1% and 5.5%, respectively (Fig. 2d). These results showed that the proposed RFCN-ResNet and RFCN-DenseNet performed significantly better than MLP-based architectures (from AutoKeras) in genomic profiling data modeling. Furthermore, AutoGenome is also very efficient, as it can search for the best model faster compared with XGBosst and AutoKeras in the same task (Supplementary Table 1).

Randomly-wired RFCN is a promising architecture in genomic research

Based on the definition of ResNet and DenseNet, the skip-connection structures in RFCN-ResNet and RFCN-DenseNet are fixed. Most neural network architectures are still handcrafted and therefore might not be the best choice for addressing various kinds of scientific problems. As such, we propose another RFCN variant, the randomly-wired residual fully-connected neural network (RRFCN), which adopts neural architecture search (NAS) to generate and search for the best residual fully-connected neural architecture for any given genomic profiling problem. With RRFCN, researchers can search for brand-new RFCN architectures to address their scientific problems.

Utilizing the existing NAS optimization methods [40–43], we integrated the efficient neural architecture search [41] (ENAS) algorithm into AutoGenome to search for the RRFCN. ENAS can significantly accelerate training by forcing all the child models to share weights [41]. The RRFCN search space is represented as a single directed acyclic graph (DAG, Fig. 2e left). This way, an RRFCN architecture can be realized by taking a subgraph from the DAG (the subgraph is represented as the red path in Fig. 2e left, and the corresponding architecture is shown in Fig. 2e right). The building block for RRFCN is the FC unit illustrated in Fig. 2a right, and the search space is described in Methods.

We also assessed the performance of RRFCN using the abovementioned two datasets. For the pan-cancer classification task, the RRFCN had the highest accuracy compared with XGBoost, AutoKeras, RFCN-ResNet, RFCN-DenseNet for both gene expression and gene mutation data (the accuracy was 96.3% for TCGA gene expression profiles and 68.1% for TCGA somatic mutation profiles, Fig. 2d left, Fig. S1a). The best RRFCN architectures from both datasets had six hidden layers and four skip connections, but the skip connections connected differently (Figs. 3a, S1b); both RRFCN architectures were brand-new neural networks for genomic research. For the single mouse cell experiment, the RRFCN achieved an accuracy of 96.3%, ranking the second highest, slightly lower than RFCN-ResNet (Fig. 2d right). The best RRFCN architecture had seven hidden layers and seven skip connections (Fig. S2b), which was also a novel neural network architecture in genomic research.

Explaining predictions for RFCN models

Many researchers believe deep learning models are black boxes [51–53] that are difficult to explain. To help researchers understand how our models work, we use a popular method called SHapley Additive exPlanations [54] (SHAP) in AutoGenome. Given a deep learning model, SHAP will calculate the marginal contribution of each feature to the overall predictions, which is referred to as the SHAP value [54]. AutoGenome can visualize the feature importance of each gene to the predicted classes (Fig. 3b, d), or the SHAP values distribution of each gene to the predicted classes (Fig. 3c, e; S3), which provides meaningful insights about the deep learning models.

For the pan-cancer classification task with gene expression profiles, the top most important genes for each cancer type are visualized by AutoGenome (Figs. 3b, c; S3a, b; Supp. Table 2). The top gene list is corroborated by many research articles in genomics. For example, the

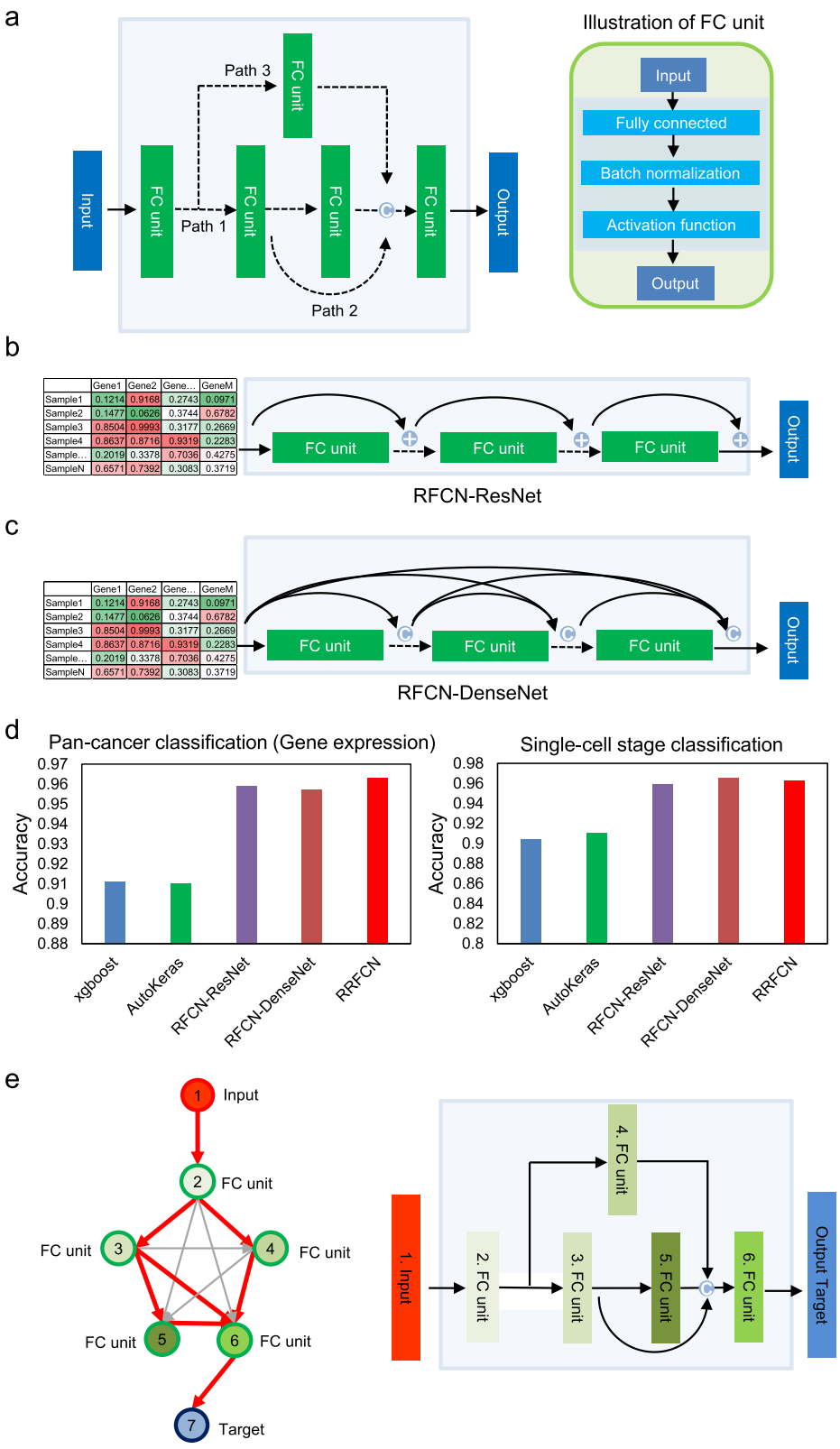

Fig. 2. Residual fully-connected neural network (RFCN). (a) We introduced fully-connected (FC) units into residual networks and named it RFCN. Each FC unit contains sequential layers of FC, batch normalization, and the activation function. We built three variants for RFCN: RFCN-ResNet (b), RFCN-DenseNet (c), and randomly-wired RRFCN (e). (b) In RFCN-ResNet, for each FC unit, it takes the summation of the previous unit's output and input as its input. "+" indicates the summation operation. (c) In RFCN -DenseNet, for each FC unit, it takes the concatenation of all the outputs from previous units as its input. "C" indicates the concatenation operation. (d) The performance of the three RFCN variants compared with traditional methods (XGBoost and AutoKeras) for pan-cancer type classification and single-cell stage classification based on gene expression profiles. The y-axis indicates top-1 classification accuracy. (e) In RRFCN, ENAS searches for the optimal subgraph from a large directed acyclic graph (DAG) as the final best model. The arrows indicate all the potential connections among nodes, among which the red ones indicate the searched optimal subgraph connections (left). A searched optimal subgraph corresponds to a neural network (right).

top ranked gene for all 24 cancer types is *RPS27* (Fig. 3b), which has been found to be highly expressed in various human cancers [55,56]. The pseudogenes *PA2G4P4* and *H3F3C* were reported to be functional in many cancers [57,58] (Fig. 3b). In cervical and endocervical cancers (CESC), a long non-coding RNA gene, *MALAT1*, is ranked top 3 among all 8449 genes (Fig. 3c), and its high expression predicts a poor prognosis of cervical cancer [59]. In lung adenocarcinoma (LUAD), *ST3GAL5* is ranked 9th (Fig. S3a), and this gene is involved in the modulation of cell proliferation and maintenance of fibroblast morphology, and has been known to be associated with *in situ* pulmonary adenocarcinoma [60]. In lung squamous cell carcinoma (LUSC), *BMS1P20* is ranked 8th (Fig. S3b), and is known to be associated with lung cancer [61]. The full list of important genes for each cancer type is shown in Supp. Table 2, and those results offer candidates of potential biomarkers for cancers.

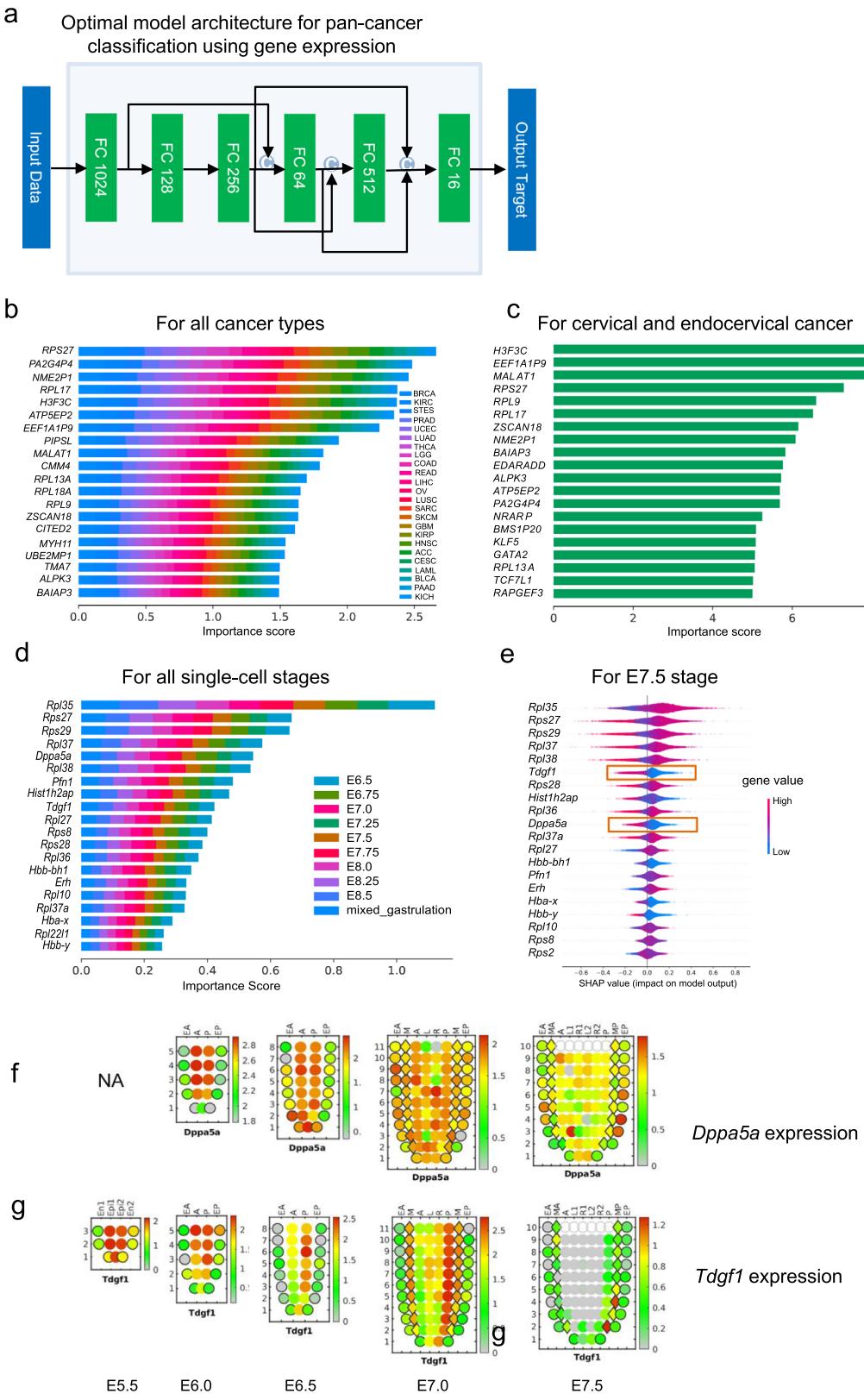

Fig. 3. Feature importance analysis for explaining predictions of RFCN. (a) The optimal model architecture for pan-cancer classification using gene expression profiles. The top 20 genes ranked by feature importance scores from gene expression-based cancer type classification for (b) all cancer types and (c) a specified cancer, cervical or endocervical cancer. The top-20 genes ranked by feature importance scores from gene expression-based single-cell classification for (d) all single-cell stages and (e) E7.5 stage. e) For the E7.5 stage, high- and low-gene expressions are in purple and blue, respectively. The red rectangles indicate marker genes for early embryonic development. (f) *Dppa5a* and (g) *Tdgf1* gene expressions from E5.5 to E7.5 stages by Corn Plot defined in the Peng et al. [72]. Corn Plot is used to visualize the dynamic gene expression change during embryonic development, where x-axes represent the embryonic region, while the y-axes represent the spatial layer on the embryo. Red or Green colors means the genes is highly or lowly expressed at the temporal/spatial point (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

We also analyzed the top genes for the somatic-mutation-profile-based pan-cancer classification. *TP53* and *PICK3CA* are ranked 1st in the contribution for the prediction of ovarian serous cystadenocarcinoma (OV) and breast invasive carcinoma (BRCA), respectively, showing a positive effect to the model output with positive SHAP values for their mutated status in most samples (Fig. S3c-d). It is consistent with *TP53* and *PICK3CA* being most frequently mutated in OV and BRCA among the 24 cancer types, with the frequency of 87.34% and 32.41%, respectively (276 OV patients with *TP53* mutations compared with 316 total OV patients, and 318 BRCA patients with *PIK3CA* mutation compared with 981 total BRCA patients). Both *TP53* and *PICK3CA* mutations were reported to be potential diagnostic and prognostic biomarkers for ovarian cancer [62] and breast cancer [63], respectively.

For the single cell embryonic development stage classification task, the important genes for each development stage were also visualized by AutoGenome (Fig. 3d–e; Supp. Table 3). Many ribosomal genes were at the top of the list, consistent with a previous finding that ribosomal genes played an important role in embryonic development and stem cell differentiation [64,65]. From the SHAP value distributions for different genes per development stage, researchers can better understand how the genes contribute to different development stages. For example, the top ranked gene, *Rpl35* (Fig. 3d–e), was reported to be important during early embryogenesis [66,67], and the *Rpl35* SHAP distribution in development stage E7.5 (Fig. 3e) showed that a high *Rpl35* expression value would predict E7.5 but a low *Rpl35* expression value would not. *Dppa5a* and *Tdgf1* were also on the top list and were reported to regulate early embryonic development and pluripotency [68–71]. By looking into their SHAP distribution in development stage E7.5 (Fig. 3e), it was clear that a low expression would predict E7.5, but a high expression would not. We validated the biological significance of SHAP distribution in another independent dataset [72]. Corn Plot [73] is used to visualize the dynamic gene expression change during embryonic development, where *Dppa5a* and *Tdgf1* had relatively high expression values in E6.0, E6.5, and E7.0, but decreased significantly in E7.5 (Fig. 3f, g). This dynamic change trend is consistent with feature importance distribution (Fig. 3e), that a lower expression of *Dppa5a* and *Tdgf1* would predict E7.5, a high expression would not.

Residual fully-connected VAE outperforms traditional VAE in omics data

RFCN-ResNet, RFCN-DenseNet, and RRFCN implemented in AutoGenome can be used for supervised learning tasks like regression and classification. We have extended RFCN to unsupervised learning by proposing a new neural network architecture, the residual fully-connected variational auto-encoder (Res-VAE, Fig. 4a) by adding skip connections to the traditional variational auto-encoder [74] (VAE) architecture. In the traditional VAE architecture (Fig. 1c), the encoder compresses input into latent vectors by decreasing the number of neurons in each subsequent layer, and the decoder reconstructs the input data from the latent vectors. The Res-VAE adds skip-connections to both the encoder and the decoder to enhance feature propagation and encourage feature reuse for each module. By minimizing the sum of the reconstruction loss and Kullback–Leibler divergence (KLD) loss, Res-VAE learns representative features from the data. The representative features are stored in the latent vectors and can be used for further analysis [75–77].

To assess the performance of Res-VAE, we designed an experiment using a new single-cell RNA-seq dataset provided by the 10X platform [78] and compared the result of Res-VAE with PCA, T-SNE, and VAE (Fig. 4b). In general, the unsupervised learning of PCA, T-SNE, VAE and Res-VAE aims at mapping high dimensional gene expression data (16,653 genes in total, see Methods) to a lower dimensional space. For a fair comparison, the first two dimensions from PCA and T-SNE results were visualized in Fig. 4c. For VAE and Res-VAE, the number of latent variables were set to 128, which meant that in the encoder module, the expression values of 16,653 genes were compressed into 128 variables. T-SNE was then used to map the 128 latent variables into a 2-dimensional space, also shown in Fig. 4c. The colors in Fig. 4 indicates the cell types, and the Res-VAE results have clearer boundaries between different cell types and tight clusters for each cell type. The Davies-Bouldin score [79] is usually used to measure the quality of clustering, and a lower Davies-Bouldin score indicates a better clustering. The results showed that Res-VAE had the lowest Davies-Bouldin score, indicating that all the cell type clusters were very compact (Fig. 4b).

AutoGenome - an AutoML tool for genomic research

Researchers may encounter various challenges when applying deep learning in their research, especially when it comes to choosing a good framework and architecture, preparing data, and setting the hyper-parameters. AutoML aims to address these challenges by combining advanced technologies like automated data cleaning [80], automated feature engineering [81], hyper parameter optimization [82], and neural architecture search [40–43]. Some AutoML tools like AutoKeras, Auto-sklearn, and H2O AutoML have been developed, but currently, they are only able to search for MLP, CNN, or RNN-based architectures.

We have developed a new AutoML tool, called AutoGenome, for genomic research to enable researchers to easily perform end-to-end learning with the best cutting-edge neural network architectures. When gene matrix data from genomic profiling is provided for supervised learning tasks, AutoGenome can automatically search for the best MLP architectures, the proposed RFCN-ResNet architectures, RFCN-DenseNet architectures, and RRFCN architectures. After AutoGenome finds the best model, the evaluation confusion matrix is also provided. Finally, AutoGenome can calculate and visualize the feature importance scores and SHAP value distributions, which researchers can use to investigate further into and explain the models (Fig. 5a). For unsupervised learning tasks, AutoGenome can automatically search for the best Res-VAE architectures and, after finding the best model, the latent variable matrix and reconstruction matrix are also provided for further analysis (Fig. 5a).

With AutoGenome, researchers only need to execute five lines of code to perform the whole analysis (Fig. 5b). AutoGenome also provides the expert mode for experienced deep learning researchers. All the parameters in AutoGenome can be manually changed by modifying the configuration file, which is well organized in JSON format. In addition, the hyper-parameter search space can also be changed to match users' favorite sets (Fig. 5b).

3. Methods

AutoGenome automl tool

AutoGenome is built on Tensorflow [83] and implemented as a Python package. The parameters required for training are specified in the JSON configuration file. It can be saved, revised and loaded to a temporary workspace. The JSON configuration file comprises the following modules. (1) Data loader module: The user can specify the paths to the training, evaluation and test datasets, or set the number of threads and capacity when loading data. (2) Model trainer module: In this module, the user can specify the general training parameters, like batch size, number of GPUs to utilize, learning rate range, and so on. (3) Loss function module: The user can specify the loss calculation method. Now we support "cross entropy" for classification tasks, "mean-square error" for regression tasks, and "area under curve" for imbalanced binary classification tasks. 4) Search mode module: There are five modes that can be selected: MLP, FC-ResNet, FC-DenseNet, ENAS for supervised learning, and Res-VAE for unsupervised learning. Each mode has a different search space that will be described later in detail. 5) Best model save path and reload module: The user can specify the path to save the best model for sharing or future reloading.

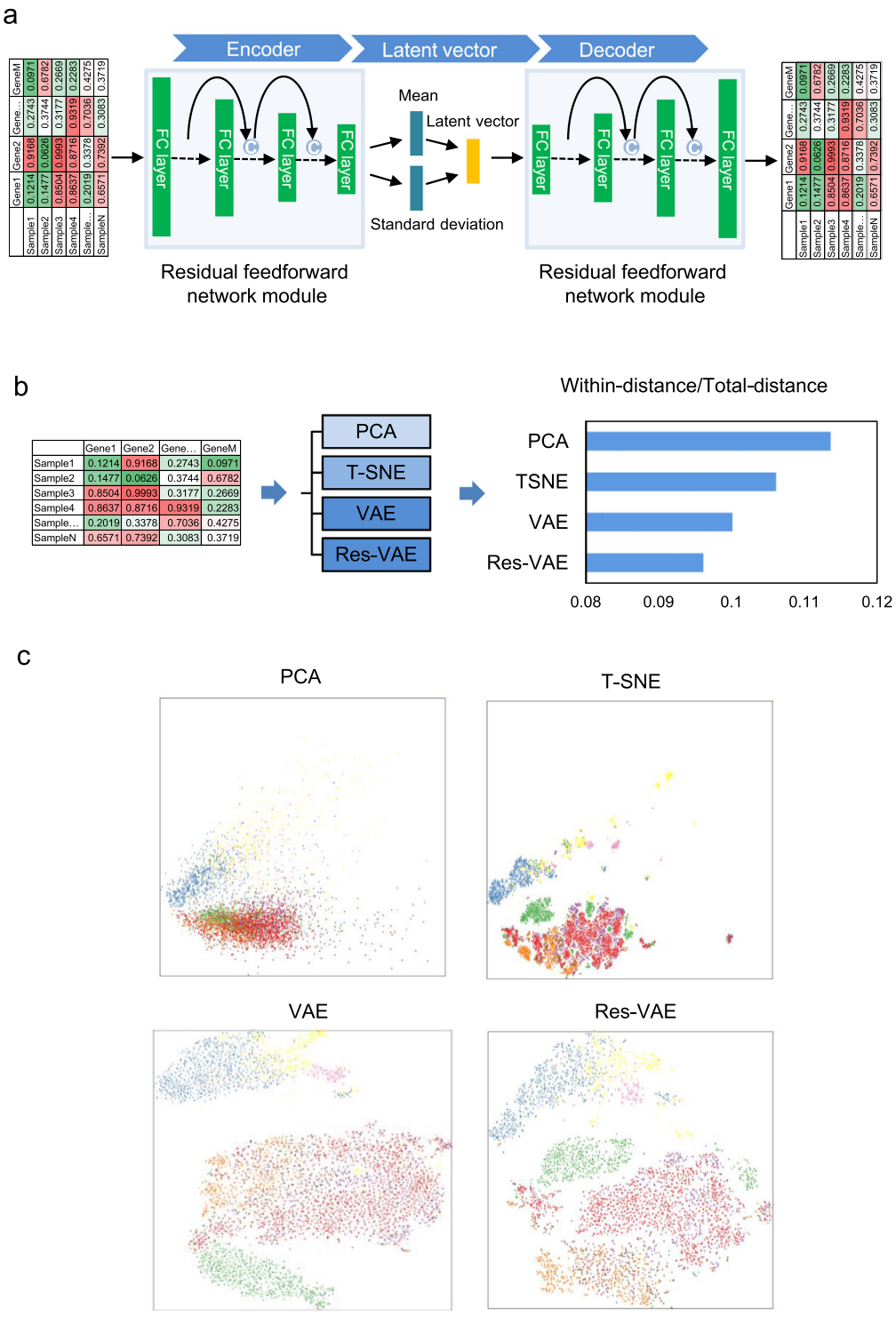

Fig. 4. Residually fully-connected VAE. (a) Res-VAE compresses the raw data into latent vectors in the Encoder process, and reconstructs the raw data from latent vectors in the Decoder process. Skip-connections are involved in both the Encoder and Decoder parts. "C" indicates that the integration method of information from different layers is concatenation. (b) Within-distance/Total-distance between single-cell categories using their gene expression profiles for comparison of Res-VAE and other three unsupervised machine learning methods. (c) 2D visualization for single cell classifications after processing by Res-VAE and other three methods.

Hyper-parameter search

The hyper-parameter search method refers to the previously described approach [84]. The general hyper-parameters in the search space are learning rate, total batch size, momentum, weight decay, number of layers in neural networks and number of neurons in each layer. The specific search space for each mode is as follows:

MLP search space

(1) The number of neurons in each layer. The default value is [8, 16, 32, 64, 128, 256, 512, 1024]. (2) The drop-out ratio of the first layer compared with the input layer, selected from [0.6, 0.8, 1.0]

RFCN-ResNet search space. (1) The number of blocks for ResNet. The default value is [1, 2, 3, 5, 6]. (2) The number of neurons in each layer.

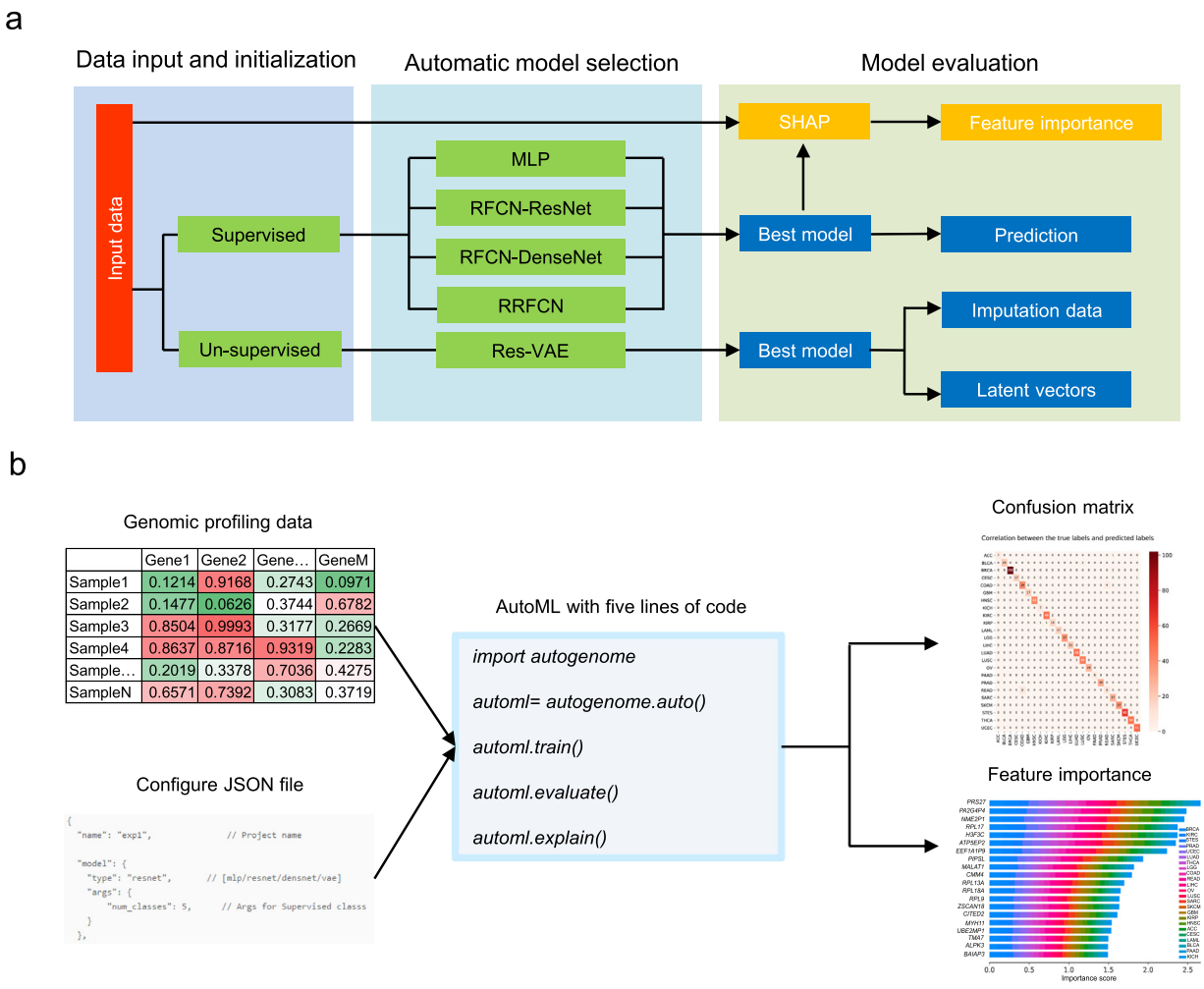

Fig. 5. Overview of AutoGenome. (a) The process of automatically building a model by using AutoGenome involves three steps: data input and initialization (left), automatic model selection (middle), and model evaluation (right). There are two types of tasks – supervised and un-supervised tasks, and five model structures – MLP, RFCN-ResNet, RFCN-DenseNet, RRFCN and Res-VAE. (b) Input data and codes to implement AutoGenome. AutoGenome takes a genomic profiling data matrix and a JSON configuration file as input and outputs the model accuracy, confusion matrix, and feature importance scores for explaining predictions. All these processes take only five lines of code for AutoGenome.

The default value is [8, 16, 32, 64, 128, 256, 512]. (3) The drop-out ratio of the first layer compared with the input layer, selected from [0.6, 0.8, 1.0]

RFCN-densenet search space

(1) The blocks structure for DenseNet. The default value is [2, 3, 4,3, 4, 5]. (2) The growth rate of neurons in each block. The default value is [8, 16, 32, 64, 128, 256, 512]. (3) The drop-out ratio of the first layer compared with the input layer, selected from [0.6, 0.8, 1.0]

Res-VAE search space

(1) The number of layers, selected from [3, 4, 5]. (2) The drop-out ratio of the first layer compared with the input layer, selected from [0.6, 0.8, 1.0]. (3) The number of neurons in the first layer, selected from [4096, 2048, 1024, 512, 256, 128]. (4) The neuron number decay ratio of the next layer compared with the previous layer, selected from [0.8, 0.6, 0.5].

ENAS search

We improved the original ENAS [42] methods in two ways to adjust them to our framework. (1) We altered the original convolutional layer into a fully-connected layer to adjust the input of modeling of genomic data. (2) We combined the output of different layers through concatenation. The number of layers in this mode should be fixed. The default number is 6. And the number of neurons in the first layer should be set in the JSON file (the default number is 2048). The search space is as follows. (1) The number of neurons from the 2nd layer to the last layer, selected from [16, 32, 64, 128, 256, 512, 1024, 2048]. (2) The connection relationship between different layers.

Feature importance estimation

We incorporated the SHAP [54] package into AutoGenome to estimate the importance of different features. When calling the function "autogenome.explain()", AutoGenome takes the best model and raw data as input for the SHAP module, with the "GradientExplainer" mode specified. The SHAP module automatically returns the SHAP value of each feature for each sample. The sum of the absolute SHAP values from each class is also calculated and an importance score is returned for each feature in each class.

Dataset preprocessing

All the datasets used in our study are public data. The data was downloaded, and then preprocessed. AutoGenome automatically splits

the dataset into training, validation and test sets with a proportion of 8:1:1. Detailed procedures of each case study are as follows:

Experiment 1 – pan-cancer classification. We downloaded gene expression profiles and somatic mutation profiles for pan-cancer patients' tumor samples covering 28 human cancer types from The Cancer Genome Atlas (TCGA [https://gdac.broadinstitute.org/]) database. Log2-transformed Transcripts Per Million (TPM) was used to represent gene expression values. Somatic mutation profiles were extracted from TCGA mutation annotation files and represented by 0 or 1, indicating a gene is mutated or not in a sample. To avoid misleading classification, we filtered cancer types with a precise definition and removed four cancer types, KIPAN, STAD, GBMLGG, and COADREAD, that were included in the sample. Thus, 24 cancer types remained for analysis. Then, the gene expression values of the 24 cancer types were processed by zero-one scaling in a gene-wise manner and input for model building.

Experiment 2 - Single mouse cell classification. We used a dataset of single mouse cells [50] to perform a classification task. The gene expression matrix was downloaded as described in https://github.com/MarioniLab/EmbryoTimecourse2018. As the sample size was very large (more than 100,000 single cells), we randomly selected 10,000 single cells for the classification task. The expression matrix we used contained 22,018 genes for 10,000 single cells. These single cells belonged to 10 predefined cell types, each having 1000 cells. The expression matrix was subjected to 0,1 scale within features before being fed to AutoGenome.

Experiment 3 - 10X PBMC single-cell RNA-seq. The dataset was provided by the 10X platform [78]. We downloaded the processed expression matrix and cell labels from https://github.com/ttgump/scDeepCluster/tree/master/scRNA-seq%20data. This expression matrix contains 16,653 expressed genes for 4271 single cells. These single cells belong to 8 predefined cell types, which correspond to the colors in Fig. 4c. The expression matrix was subjected to 0,1 scale within features before being fed to AutoGenome.

XGBoost and autokeras

We selected "gbtree" in XGBoost as the 'booster' to train the model, with the following parameters generated from a grid-like searched parameters: 'objective' : "multi : softmax", 'gamma' : "0.1″, 'max_depth' : "6″. And then trained models by these parameters.

We chose "MlpModule" in AutoKeras for genomics data modeling, with the parameters "loss = classification_loss" and "metric = Accuracy" and then searched 4 or 6 h by AutoKeras to determine the most suitable model for current task.

PCA and T-SNE

The "PCA" and "TSNE" functions from "sklearn" [85] were used for unsupervised learning from genomic data. After dimensionality reduction, the first two dimensions were used for further visualization. We then calculated the pair-wise Euclidean distance of the sample in the first two dimensions. As the raw data contains golden standard labels, we can calculate the Davies-Bouldin score to evaluate the performance of dimensionality reduction. "sklearn" was used to calculate the Davies-Bouldin score.

Data availability

All the datasets used in our study are public data. The dataset used for pan-cancer classification is from TCGA. Single-cell classification data is from accessions: Atlas: E-MTAB-6967 and the processed data is downloaded following the instructions at https://github.com/MarioniLab/EmbryoTimecourse2018. 10X PBMC single-cell RNA-Seq was provided by 10X platform and we downloaded the processed expression matrix and cell labels from https://github.com/ttgump/scDeepCluster/tree/master/scRNA-seq%20data.

Software availability

The software website could be accessed from http://autogenome.com.cn/, which contains the software introduction, installation, and softwawre usage. The protocols for specific experiments are also provided as notebook examples on the website.

4. Discussion

Genomic profiling data are crucial for biomedical researches, and they are growing rapidly. Hence machine learning methods are being used by researchers to predict and interpret genomic data. However, many of the methods also have shortcomings, and new neural network architectures, which mirror the recent progresses in computer vision, natural language processing, and speech processing, are needed to overcome challenges in this area. Compared with the widely used MLP architecture, the advantages of the RFCN architecture are significant: a. RFCN can be used to train deeper neural networks; b. RFCN can enhance feature propagation and encourage the reuse of features; c. RRFCN can be used to generate new RFCN architectures. AutoGenome has made it easy for researchers to use the RFCN in their research.

After reviewing several genomic research publications, we found that most of the published MLP-based neural networks have less than 4 layers (Fig. S4a). We have summarized the depth (number of hidden layers) of the best model for RFCN architectures in experiments 1 and 2 (Supp. Table 4). The depth of RFCN-ResNet and RRFCN are around 6 to 7, and the depth of RFCN-DenseNet is around 11. RFCN architectures can be used to train neural networks with more hidden layers. We also looked into the correlations between the depth of RRFCN and accuracy (Fig. S4b), and found that the model delivered the highest accuracy when the depth was around 6 to 7. Further increasing the depth would not improve the performance. This might be relevant to the complexity of the problems as well as the underlying gene regulatory pathway/network.

In our experiments, RFCN architectures were proven to have better performances than MLP architectures, but for the three RFCN variants (RFCN-ResNet, RFCN-DenseNet, and RRFCN), the performances were similar (Fig. 2d). Compared with RRFCN, RFCN-ResNet and RFCN-DenseNet are easier to train and the architectures are easier to understand (Fig. 2a–c, e).

AutoGenome is open source and freely available to everyone. Notebook examples are also provided so researchers can get started with AutoGenome quickly.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The authors declare no competing interests.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ailsci.2021.100017.

CRediT authorship contribution statement

Denghui Liu: Investigation, Software, Writing – original draft, Writing – review & editing. **Chi Xu:** Investigation, Software, Writing – original draft, Writing – review & editing. **Wenjun He:** Investigation, Software, Writing – review & editing. **Zhimeng Xu:** Investigation, Software, Writing – review & editing. **Wenqi Fu:** Writing – original draft, Writing – review & editing. **Lei Zhang:** Investigation, Software, Writing – review & editing. **Jie Yang:** Writing – original draft, Writing – review & editing. **Zhihao Wang:** Writing – original draft, Writing – review &

editing. **Bing Liu:** Writing – original draft, Writing – review & editing. **Guangdun Peng:** Writing – original draft, Writing – review & editing. **Dali Han:** Writing – original draft, Writing – review & editing. **Xiaolong Bai:** Writing – original draft, Writing – review & editing. **Nan Qiao:** Visualization, Writing – original draft, Writing – review & editing.

References

[1] Taub FE, DeLeo JM, Thompson EB. Sequential comparative hybridizations analyzed by computerized image processing can identify and quantitate regulated RNAs. DNA Mary Ann Liebert Inc 1983;2:309–27.
[2] Shendure J, Ji H. Next-generation DNA sequencing. Nat Biotechnol 2008;26:1135–45.
[3] Weber M, et al. Chromosome-wide and promoter-specific analyses identify sites of differential DNA methylation in normal and transformed human cells. Nat Genet 2005;37:853–62.
[4] Frommer M, et al. A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands. Proc Natl Acad Sci USA 1992;89:1827–31.
[5] Johnson DS, Mortazavi A, Myers RM, Wold B. Genome-wide mapping of *in vivo* protein-DNA interactions. Science 2007;316:1497–502.
[6] Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 2009;10:57–63.
[7] Moco S, et al. A liquid chromatography-mass spectrometry-based metabolome database for tomato. Plant Physiol 2006;141:1205–18.
[8] Colangelo CM, Williams KR. Isotope-coded affinity tags for protein quantification. Methods Mol Biol Clifton NJ 2006;328:151–8.
[9] Reo NV. NMR-based metabolomics. Drug Chem Toxicol 2002;25:375–82.
[10] Dettmer K, Aronov PA, Hammock BD. Mass spectrometry-based metabolomics. Mass Spectrom Rev 2007;26:51–78.
[11] Gallo Cantafio ME, et al. From single level analysis to multi-omics integrative approaches: a powerful strategy towards the precision oncology. High Throughput 2018;7.
[12] Manzoni C, et al. Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences. Brief Bioinform 2018;19:286–302.
[13] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. ArXiv:150602640 Cs (2015).
[14] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. in Advances in neural information processing systems 25, Pereira F, Burges CJC, Bottou L, Weinberger KQ, (editors) 1097–105 (Curran Associates, Inc., 2012).
[15] Collobert R, et al. Natural language processing (almost) from scratch. ArXiv:11030398 Cs (2011).
[16] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. (2013).
[17] Xiong W, et al. Achieving human parity in conversational speech recognition. ArXiv:161005256 Cs Eess (2016).
[18] Sak H, Senior A, Rao K, Beaufays F. Fast and accurate recurrent neural network acoustic models for speech recognition. ArXiv:150706947 Cs Stat (2015).
[19] Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Imaging 2018;9:611–29.
[20] Valen DAV, et al. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. PLOS Comput Biol 2016;12:e1005177.
[21] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80.
[22] Poplin R, et al. A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol 2018;36:983.
[23] Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res 2016;26:990–9.
[24] Abadi S, Yan WX, Amar D, Mayrose I. A machine learning approach for predicting CRISPR-Cas9 cleavage efficiencies and patterns underlying its mechanism of action. PLoS Comput Biol 2017;13:e1005807.
[25] Listgarten J, et al. Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. Nat Biomed Eng 2018;2:38–47.
[26] Hashemifar S, Neyshabur B, Khan AA, Xu J. Predicting protein–protein interactions through sequence-based deep learning. Bioinformatics 2018;34:i802–10.
[27] Zhang L, Yu G, Xia D, Wang J. Protein–protein interactions prediction based on ensemble deep neural networks. Neurocomputing 2019;324:10–19.
[28] Subramanian A, et al. A next generation connectivity map: L1000 platform and the First 1,000,000 profiles. Cell 2017;171 1437-1452.e17.
[29] Duvenaud DK, Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, et al. Convolutional networks on graphs for learning molecular fingerprints. In: Advances in neural information processing systems, 28. Curran Associates, Inc.; 2015. p. 2224–32.
[30] Hirohara M, Saito Y, Koda Y, Sato K, Sakakibara Y. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. BMC Bioinform 2018;19:526.
[31] Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 2000;28:27–30.
[32] Lin C, Jain S, Kim HY, Bar-Joseph Z. Using neural networks to improve single cell RNA-seq data analysis. in (2017).
[33] Jabeen A, Ahmad N, Raza K. Machine learning-based state-of-the-art methods for the classification of RNA-Seq data. bioRxiv 120592 (2017), doi:10.1101/120592.
[34] Urda D, Montes-Torres J, Moreno F, Franco L, Jerez J. Deep learning to analyze rna-seq gene expression data. In: Advances in computational intelligence; 2017. p. 50–9. doi:10.1007/978-3-319-59147-6_5.
[35] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv:14091556 Cs (2014).
[36] Szegedy C, et al. Going deeper with convolutions. ArXiv:14094842 Cs (2014).
[37] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. ArXiv:12115063 Cs (2012).
[38] Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. IEEE; 2001.
[39] Jin H, Song Q, Hu X. Auto-keras: an efficient neural architecture search system. ArXiv:180610282 Cs Stat (2018).
[40] Cai H, Chen T, Zhang W, Yu Y, Wang J. Efficient architecture search by network transformation. In: Proceedings of the thirty-second AAAI conference on artificial intelligence; 2018.
[41] Pham H, Guan MY, Zoph B, Le QV, Dean J. Efficient neural architecture search via parameter sharing. ArXiv:180203268 Cs Stat (2018).
[42] Zoph B, Le QV. Neural architecture search with reinforcement learning. ArXiv:161101578 Cs (2016).
[43] Elsken T, Metzen JH, Hutter F. Neural architecture search: a survey. ArXiv:180805377 Cs Stat (2018).
[44] Deng J, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of the 2009 IEEE conference on computer vision and pattern recognition; 2009. p. 248–55. doi:10.1109/CVPR.2009.5206848.
[45] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR); 2016. p. 770–8. doi:10.1109/CVPR.2016.90.
[46] Srivastava RK, Greff K, Schmidhuber J. Highway Networks. ArXiv:150500387 Cs (2015).
[47] Huang G, Liu Z, Maaten L vd, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR); 2017. p. 2261–9. doi:10.1109/CVPR.2017.243.
[48] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 16 785–94 (2016) doi:10.1145/2939672.2939785.
[49] Weinstein JN, et al. The cancer genome atlas pan-cancer analysis project. Nat Genet 2013;45:1113–20.
[50] Pijuan-Sala B, et al. A single-cell molecular map of mouse gastrulation and early organogenesis. Nature 2019;566:490–5.
[51] Kim B, Khanna R, Koyejo OO, Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R. Examples are not enough, learn to criticize! Criticism for Interpretability. In: Advances in neural information processing systems, 29. Curran Associates, Inc.; 2016. p. 2280–8.
[52] Doshi-Velez F, Wallace B, Adams R. Graph-Sparse L.D.A.: A topic model with structured sparsity. ArXiv:14104510 Cs Stat (2014).
[53] Kim B, Rudin C, Shah J. The bayesian case model: a generative approach for case-based reasoning and prototype classification. ArXiv:150301161 Cs Stat (2015).
[54] Lundberg SM, Lee SI, Guyon I, et al. A unified approach to interpreting model predictions. Advances in neural information processing systems 2017;30:4765–74.
[55] Dutton-Regester K, et al. A highly recurrent RPS27 5'UTR mutation in melanoma. Oncotarget 2014;5:2912–17.
[56] Huang CJ, et al. Ribosomal protein S27-like in colorectal cancer: a candidate for predicting prognoses. PLoS ONE 2013;8:e67043.
[57] Li Y, et al. A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data. BMC Genom 2017;18.
[58] Li Y, et al. A comprehensive genomic pan-cancer classification using the cancer genome atlas gene expression data. BMC Genom 2017;18:508.
[59] Yang L, Bai HS, Deng Y, Fan L. High MALAT1 expression predicts a poor prognosis of cervical cancer and promotes cancer cell growth and invasion. Eur Rev Med Pharmacol Sci 2015;19:3187–93.
[60] Mathow D, et al. Zeb1 affects epithelial cell adhesion by diverting glycosphingolipid metabolism. EMBO Rep 2015;16:321–31.
[61] Lyu L, et al. Integrative analysis of the lncRNA-associated ceRNA network reveals lncRNAs as potential prognostic biomarkers in human muscle-invasive bladder cancer. Cancer Manag Res 2019;11:6061–77.
[62] Zhang Y, Cao L, Nguyen D, Lu H. TP53 mutations in epithelial ovarian cancer. Transl Cancer Res 2016;5:650–63.
[63] Mukohara T. PI3K mutations in breast cancer: prognostic and therapeutic implications. Breast Cancer Targets Ther 2015;7:111–23.
[64] Zahn LM. Ribosomes regulate stem cell fate. Science 2015;347:1214.
[65] Sharma E, Blencowe BJ. Orchestrating ribosomal subunit coordination to control stem cell fate. Cell Stem Cell 2018;22:471–3.
[66] Jiang N, Hu L, Liu C, Gao X, Zheng S. 60S ribosomal protein L35 regulates β-casein translational elongation and secretion in bovine mammary epithelial cells. Arch Biochem Biophys 2015;583:130–9.
[67] Lau TP, et al. Pair-wise comparison analysis of differential expression of mRNAs in early and advanced stage primary colorectal adenocarcinomas. BMJ Open 2014;4:e004930.
[68] Miharada K, Sigurdsson V, Karlsson S. Dppa5 improves hematopoietic stem cell activity by reducing endoplasmic reticulum stress. Cell Rep 2014;7:1381–92.
[69] Qian X, Kim JK, Tong W, Villa-Diaz LG, Krebsbach PH. DPPA5 supports pluripotency and reprogramming by regulating NANOG turnover. Stem Cells Dayt Ohio 2016;34:588–600.
[70] Azizi H, Asgari B, Skutella T. Pluripotency potential of embryonic stem cell-like cells derived from mouse testis. Cell J 2019;21:281–9.

[71] Miyoshi N, et al. TDGF1 is a novel predictive marker for metachronous metastasis of colorectal cancer. Int J Oncol 2010;36:563–8.
[72] Peng G, Cui G, Ke J, Jing N. Using single-cell and spatial transcriptomes to understand Stem cell lineage specification during early embryo development. Annu Rev Genom Hum Genet 2020;21:163–81.
[73] Peng G, et al. Molecular architecture of lineage allocation and tissue organization in early mouse embryo. Nature 2019;572:528–32.
[74] An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. SNU Data Mining Center; 2015.
[75] Higgins I, et al. beta-VAE: learning basic visual concepts with a constrained variational framework ICLR; 2017.
[76] Way GP, Greene CS. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. In: Proceedigs of the pacific symposium on biocomputing, 23; 2018. p. 80–91.
[77] Wang D, Gu J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. Genom Proteom Bioinform 2018;16:320–31.
[78] Zheng GXY, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;8:14049.
[79] Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 1979:224–7 PAMI-1.
[80] Chuck C, et al. Statistical data cleaning for deep learning of automation tasks from demonstrations. In: Proceedings of the 2017 13th IEEE conference on automation science and engineering (CASE); 2017. p. 1142–9. doi:10.1109/COASE.2017.8256258.
[81] Khurana U, Turaga D, Samulowitz H, Parthasrathy SCognito. Automated feature engineering for supervised learning. In: Proceedings of the 2016 IEEE 16th international conference on data mining workshops (ICDMW); 2016. p. 1304–7. doi:10.1109/ICDMW.2016.0190.
[82] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res 2012;13:281–305.
[83] Abadi M, et al. TensorFlow: a system for large-scale machine learning. ArXiv:160508695 Cs (2016).
[84] Smith LN. A disciplined approach to neural network hyper-parameters: part 1 – learning rate, batch size, momentum, and weight decay. ArXiv:180309820 Cs Stat (2018).
[85] Pedregosa F, et al. Scikit-learn: machine learning in python. J Mach Learn Res 2011;12:2825–30.