Briefings in Bioinformatics, 2025, **26(1)**, bbae639
https://doi.org/10.1093/bib/bbae639
Problem Solving Protocol

A self-attention-driven deep learning framework for inference of transcriptional gene regulatory networks

Yong Liu[1,‡], Le Zhong[1,‡], Bin Yan[2,‡], Zhuobin Chen[3,‡], Yanjia Yu[1], Dan Yu[2], Jing Qin[3,*], Junwen Wang[2,4,*]

[1]College of Electronic Information, Guangxi Minzu University, 188 East University Road, Nanning, Guangxi, 530006, China
[2]Division of Applied Oral Sciences & Community Dental Care, Faculty of Dentistry, The University of Hong Kong, 34 Hospital Road, Hong Kong SAR, China
[3]School of Pharmaceutical Sciences (Shenzhen), Shenzhen Campus of Sun Yat-sen University, 66 Gongchang Road, Shenzhen, Guangdong, 518107, China
[4]Department of Quantitative Health Sciences, Center for Individualized Medicine, and Mayo Clinic Comprehensive Cancer Center, Mayo Clinic, 13400 E Shea Blvd, Scottsdale, AZ, 85259, United States

*Corresponding authors. Junwen Wang, Faculty of Dentistry, The University of Hong Kong, 34 Hospital Road, Hong Kong SAR, China. E-mail: junwen@hku.hk; Jing Qin, School of Pharmaceutical Sciences (Shenzhen), Shenzhen Campus of Sun Yat-sen University, 66 Gongchang Road, Shenzhen, Guangdong, China. E-mail: qinj29@mail.sysu.edu.cn
‡Yong Liu, Le Zhong, Bin Yan, and Zhuobin Chen contributed equally to this work.

Abstract

The interactions between transcription factors (TFs) and the target genes could provide a basis for constructing gene regulatory networks (GRNs) for mechanistic understanding of various biological complex processes. From gene expression data, particularly single-cell transcriptomic data containing rich cell-to-cell variations, it is highly desirable to infer TF–gene interactions (TGIs) using deep learning technologies. Numerous models or software including deep learning–based algorithms have been designed to identify transcriptional regulatory relationships between TFs and the downstream genes. However, these methods do not significantly improve predictions of TGIs due to some limitations regarding constructing underlying interactive structures linking regulatory components. In this study, we introduce a deep learning framework, DeepTGI, that encodes gene expression profiles from single-cell and/or bulk transcriptomic data and predicts TGIs with high accuracy. Our approach could fuse the features extracted from Auto-encoder with self-attention mechanism and other networks and could transform multihead attention modules to define representative features. By comparing it with other models or methods, DeepTGI exhibits its superiority to identify more potential TGIs and to reconstruct the GRNs and, therefore, could provide broader perspectives for discovery of more biological meaningful TGIs and for understanding transcriptional gene regulatory mechanisms.

Keywords: self-attention mechanism; deep learning; transcription factor-gene interaction; gene regulatory networks; single-cell transcriptomic data

Introduction

Regulation of gene expression in eukaryotic cells is accomplished by different complex mechanisms, a set of interactions between transcription factors (TFs) and the downstream target genes in specific ways (activation or inhibition). The regulatory interactions between TFs and genes provide primary information for reconstruction of gene regulatory networks (GRNs) that underlie various biological processes and functions. Therefore, identification of the TF–gene interactions (TGIs) could assist in mechanistic understanding of gene regulations. Recently advanced single-cell sequencing technologies allow us to explore alterations of gene expression in single cell levels. The previous studies suggest that complex interrelations between TFs and genes linking regulatory factors to coordinate gene expression profiling could be proposed as a model for predicting TGIs and for inferring GRNs.

A variety of traditional models or approaches have been developed for integrative analysis of gene expression profiling and for unraveling a transcriptional regulatory relationship from bulk transcriptomic data, for example, GENIE3 [1], WGCNA [2], and ARACNE [3], etc. In contrast, PIDC [4], PPCOR [5], and SCENIC [6] can use single-cell-based transcriptomic data for modelling pairwise mutation information between genes and then compute the ratio between the unique component and the mutation. However, these correlation-based methods cannot distinguish the direction of regulatory relationships because of the symmetry of the correlation information. Machine learning–based GENIE3, GRNBoost2, and SCENIC utilize algorithms, such as Gradient Boosting Machine and Random Forest, to predict TGIs and infer GRNs as regression or classification problems in large scale, which require intensive computation, thus limiting the scope of applications of these algorithms.

Different from these traditional methods above, deep learning–based models can assemble not only gene expression profiles but also known regulatory factor–gene interactions and organism/tissue- or cell-related information [7–9]. In general, they rely on classic deep learning algorithms, like artificial neural network (ANN), convolutional neural network (CNN), or gate recurrent unit, for inferring regulatory relationships between TFs

Received: June 11, 2024. **Revised:** October 15, 2024. **Accepted:** November 25, 2024

and genes [10–12]. For example, CNN for coexpression encodes gene expression data into matrices and then extracts spatial features from it, which integrates prior knowledge and other information contained in expression data during the training of models [9]. Limited by such encoding, the CNN methods cannot efficiently capture the correlation information contained in transcriptomic data. An ANN-based gene network embedding assembles both expression data and gene interaction topology into embedded representations and then predicts gene regulatory interactions. However, it cannot extract effective features from the embedding vector limited to the ANN [13]. Most recently, a hybrid deep learning method, DGRNS, encodes the gene expression with Pearson correlation coefficient (PCC) and then extracts time-dependent characteristics from the correlation vector with gate recurrent unit. Finally, it employs CNNs to learn spatial features from the matrices to reconstruct GRNs [14]. However, this method relies on time-related information, which affects its performance in processing data not in time series.

To overcome the limitations described above, we newly devised a deep learning framework DeepTGI (Deep learning method for TF–Gene Interaction) that can integrate Auto-encoder with self-attention mechanism and other networks using correlation vectors of gene expression profile. By analysis of bulk or single-cell transcriptomic data, our model can identify potential interactions between TFs and genes with biological meanings. We carried out comprehensive evaluations on performance of DeepTGI by comparative analyses of various transcriptomic datasets from different cell lineages and species with other methods. The results demonstrate the advantages of DeepTGI in holding high accuracy to identify regulatory relationships between TFs and target genes and to infer GRNs and thus could provide a better solution for understanding of transcriptional gene regulation.

Data and methods

Data collection

In this study, six mRNA expression datasets of mouse or human extracted from the National Center of Biotechnology Information/Gene Expression Omnibus database were used for evaluation, comparison, and application of models (Supplementary Table S1). There are also different sets of TFs selected for analyzing each dataset due to their involvement in the experiments. These data include the following three sets.

1) Mouse embryonic stem cells (mESCs) were collected from three bulk transcriptomic data GSE16375, GSE31381, and GSE26520. A total of 222 TFs in three datasets were included in TGI prediction analysis.

2) Single-cell RNA-seq (scRNA-seq) data GSE103221 of different time-point reprogramming processes from mouse-induced pluripotent stem cells (miPSCs) [15]. As previously described [13], the scRNA-seq raw data were normalized, and the normalized counts of all genes were log2-transformed. The scRNA-seq data show dynamic change of gene expression during 8 days from 0 to 8.

3) To compare different cell lineages/tissues or species, four scRNA-seq datasets similar to those proposed by Zhao *et al.* [14] were used, including two mouse hematopoietic stem cell (mHSC) datasets from two different lineages, progenitor cell transitions to erythroid (mHSC-E) or to lymphoid (mHSC-L), and two human datasets, of human embryonic stem cells (hESCs) derived from 758 cells along the process of differentiation into definitive endoderm cells, and of hepatocyte-like lineage (hHEPs) from human induced pluripotent stem cells differentiating to hepatocyte-like cells in 2D culture.

Our method requires the known TGIs as input training data. When trained with mESC and miPSC datasets, we constructed the gold-standard GRNs of mESCs by using our ChIP-Array [16, 17]. This method uses ChIP-seq/chip of a TF combined with gene expression change under perturbation of the same TF in mESCs to identify all targets of the TF. Each target of a TF in the network is evidenced by the *in vivo* cell-type-specific binding sites of the TF on its promoter/enhancer and the expression change in the perturbation experiment of the TF, which is well accepted as a true target of this TF. According to the true TGIs, the consequential gold standard networks can be built. Based on the TGIs predicted by using bulk or single-cell transcriptomic data and the gold-standard GRNs, common genes are selected from all three datasets for further training, testing, and evaluation. The gold standard networks corresponding to two mouse datasets mHSC-E and mHSC-L and two human datasets hESC and hHEP are also obtained from BEELINE [18].

DeepTGI overview

In this study, we introduce DeepTGI, a self-attention mechanism–driven deep learning framework that aims to decode regulatory relations between TFs and genes and to reconstruct GRNs. Figure 1 illustrates the schematic diagram of DeepTGI consisting of three main steps. Firstly, single-cell and/or bulk transcriptomic data are used as input. DeepTGI calculates and sets up PCC similarity matrices by comparative analysis of expression profile between TFs and genes in the datasets. The model then extracts vector representations of TFs and genes within the similarity matrices and subsequently concatenates these vectors in a structured manner. Secondly, DeepTGI fuses features from various transcriptomic data through three network approaches, Auto-encoder, 2DCNN-1DCNN, and Siamese. Auto-encoders can perform representation learning on messenger RNA (mRNA) expression data and simultaneously fuse features from single-cell and/or bulk data. The 2DCNN-1DCNN can learn the spatial features contained in the transcriptomic data, and the Siamese network can learn similar characteristics between TFs and target genes. Finally, after fusing the three features, multihead attention modules are applied to learn important information from the latent vectors. The final linear layers are utilized to predict all possible TGIs, and the potential interactions of main TFs and the target genes would provide a basis for reconstruction of GRNs.

Feature representation

Feature representation is essential for model construction. Here, we used both bulk and scRNA-seq data to train the models, which can learn more features from various datasets. The feature of each gene (including TFs) corresponds to a set of its expressions in all samples. To acquire the information containing internal correlations, we calculated PCCs of mRNA expression between TFs and genes, respectively, and two similarity matrices were generated. The PCC is a measure of the linear relationship between TFs and genes, enabling quantification of correlation degrees. The bulk and single-cell data generate two $N \times N$ similarity matrices, respectively, where N is the number of genes. The PCC of a pair of genes/TFs, gene/TF i and j is calculated as follows:

$$\rho_{i,j} = \frac{\mathrm{Cov}(i,j)}{\sigma_i \sigma_j}$$

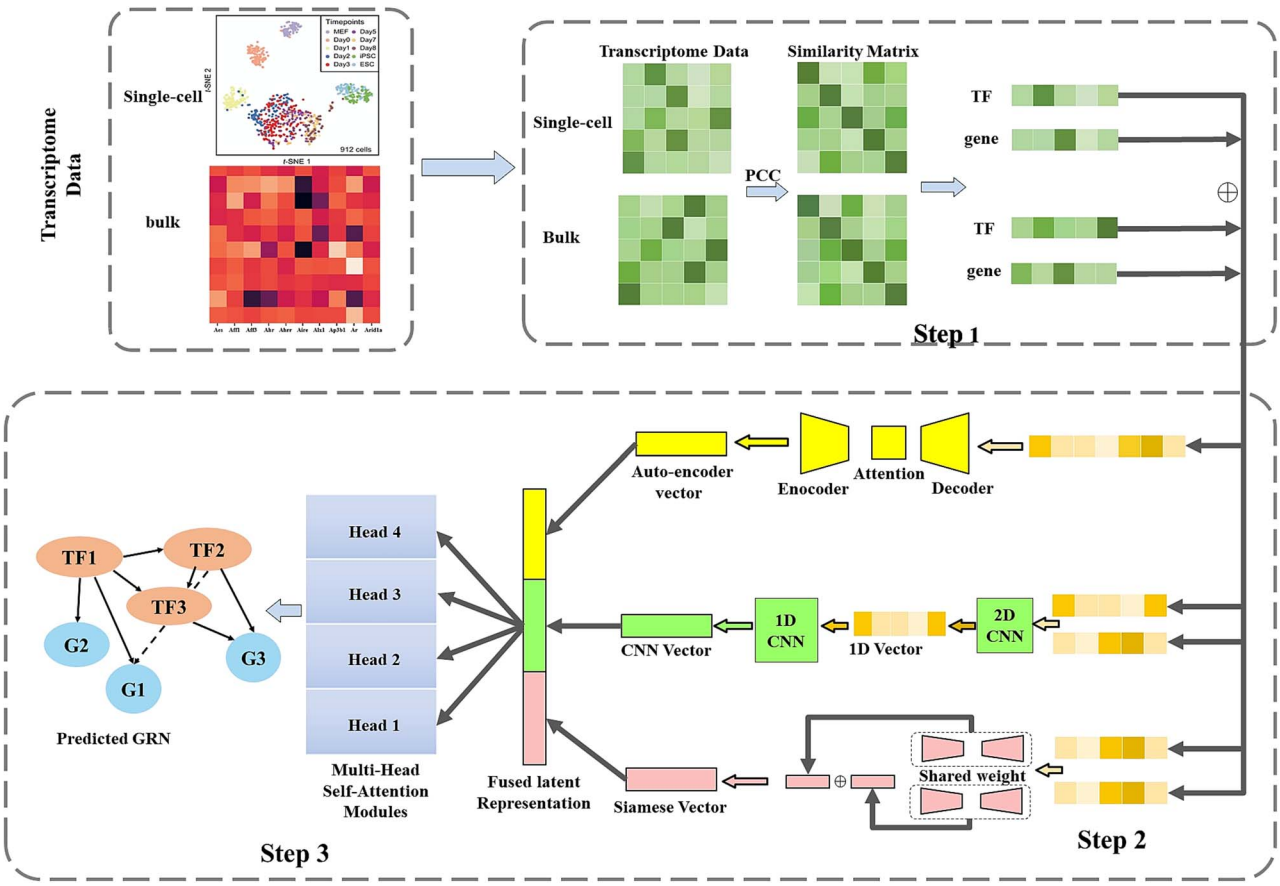

Figure 1. Overview of the self-attention-driven deep learning framework, DeepTGI. This model system establishes the GRNs according to interactions between TFs and genes in biological systems. The DeepTGI algorithm starts with profiling gene expressions to learn vector representations of TFs and their downstream genes within the PCC similarity matrix and to obtain a latent vector representation for TF–gene pairs. DeepTGI employs a fully connected layer to predict potential TGIs.

where Cov(i,j) is the covariance of the pair of gene/TF i and j and σi and σj is the standard deviation of their expression, respectively. Each row of the similarity matrix represents the degree of similarity between a TF and all other genes, and this information will be used for subsequent feature extraction.

The single-cell datasets are sparse and have high dimensionality. As input data, such high sparsity may reduce the performance of the model. In our method, PCCs play as the features to further decrease the sparsity of the input and enrich the features.

Feature extraction

DeepTGI first uses three machine learning models to extract different features from transcriptomics data, namely, Auto-encoders, 2DCNN-1DCNN, and Siamese networks. In machine learning, Auto-encoder is an unsupervised learning model, which extracts the features from the input data. The output is a feature representation of the input data.

Auto-encoder with self-attention mechanism

The Auto-encoder contains two parts: an encoder that encodes the input and extracts features from it and a decoder that aims to decode the features and acquire the fused representation of the input. In DeepTGI, the input data comprise two TF features and two gene features derived from single-cell and bulk transcriptomics, respectively. Auto-encoder is a decent way to fuse those features into one latent representation for feature extraction. A high dimensionality can lead to extremely inferior performance of the model under certain circumstances. Thus, the Auto-encoder represents an appropriate module for dimension reduction and feature extraction [19]. The self-attention mechanism is derived from attention mechanism and is widely applied in multiple tasks, which can reduce dependence of models on internal information and make models capture useful features from the internal correlation of data [20]. Moreover, the self-attention mechanism empowers models to focus on the critical information contained in internal correlations. We thus add a self-attention layer into the Auto-encoder that can make models extract critical information from gene expression profiles. In addition, an extra output of the Auto-encoder can be an auxiliary loss for improving the performance of the model.

2DCNN-1DCNN

The CNN is a deep learning algorithm that extracts features from the input. DeepTGI uses two types of CNNs: two-dimensional CNN (2DCNN) and one-dimensional CNN (1DCNN) to process single-cell or bulk transcriptomic datasets. To extract the spatial characteristics of transcriptomic data, the similarity vectors of TFs and genes go into 1DCNN vectors, and then, they are combined into a 2D matrix and become a 2DCNN that extracts their combined two-bit space features. The output of the 2DCNN represents a 1D vector as the latent feature vector. This vector is further inputted into the 1DCNN to obtain the final spatial latent vector that is finally fed into the feature fusion modules.

Siamese networks

A Siamese network contains two or more identical subnetworks used to find similarities between inputs by comparing feature vectors [21]. The twin network extracts the output by sharing weights into the similar characteristics of the dataset. Here, mRNA expression similarity matrices of TFs and genes are fed into two branches of the Siamese network: branch1 and branch2. Each branch of the twin network adopts the same structure, including a multihead attention mechanism layer, linear layer, and layer normalization. Through the attention mechanism, the model would extract important similar features. Specifically, each input vector is composed of a TF vector and a gene vector that are concatenated to form a vector, which is then fed into branch1 and branch2, respectively. After processing, the outputs of branch1 and branch2 are combined to form a feature vector that brings similar features extracted from the two datasets and is finally fused to the feature fusion modules for further fusion processing.

Feature fusion

Feature fusion is completed by the integration of multiple stages in the DeepTGI algorithm. We used the encoder structure of the transformer to perform latent feature fusion. Through three different gene fusion methods and network structures, Auto-encoder, 2DCNN-1DCNN, and Siamese network, we can obtain three different latent vectors of TF–gene pairs. Then, we element-wise add the three latent vectors to get the four hidden vectors through a multihead attention mechanism. Finally, we concatenate these four hidden vectors as the new features of the TF–gene pair and transform such features into the encoder structure of the transformer (Fig. 1). Additionally, the multihead attention mechanism is used afterward to assign different weights to the fused features based on their importance, enhancing the final representation.

Multihead attention mechanism

Self-attention mechanism, a supervised deep learning model, calculates the attention score of the input with each other. The model can detect the important features that we should pay attention to. By contrast, the multihead attention mechanism is a variant of sell-attention. Different from self-attention, multi-head attention collects different attention scores of the input by deploying multiple attention heads [22]. After obtaining the latent feature extracted by the Auto-encoder, the information contained in the four expressions' similarity vectors contribute differently to TGI prediction since some features extracted by the Auto-encoder are redundant and less important [23]. The multihead attention modules can help models to identify features that are critical to the model and give those features with high weights for inferring GRNs. Since gene expression profiles are highly dimensional, we calculate the PCCs to decrease the sparsity of expression data, which makes the latent vector become more informative for the model to learn features from it. Here, we apply a multihead attention mechanism to predict TGIs that can learn essential features from the latent similarity vector. Recognizing the increased complexity and diversity of features within the gene expression data, we have expanded the number of modules in the multi-head attention mechanism from two to four. This adjustment allows for a more nuanced and comprehensive analysis of the latent similarity vectors, significantly enhancing their capability to identify pivotal TGIs in GRNs. The augmented equation for the multihead attention, accommodating this enhancement, is defined as follows:

$$\mathrm{Q_i} = \mathrm{X} \bullet \mathrm{W_i^Q}$$

$$\mathrm{K_i} = \mathrm{X} \bullet \mathrm{W_i^K}$$

$$V_i = X \bullet W_i^V$$

$$\mathrm{head_i} = \mathrm{softmax}\left(\frac{Q_i \bullet K_i^T}{\sqrt{d_k}}\right)$$

$$X_{\mathrm{MH\_attn}} = \mathrm{Concat}\ (\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_n)$$

where X is the feature vector attained from the Auto-encoder, and W_i^Q, W_i^k, and W_i^v are the ith weight matrices. Q_i, K_i, *and* V_i are the ith matrices obtained from the linear transformation of X, respectively.

Residual module & layer normalization module

Two modules, residual and layer normalization, are used in DeepTGI. The residual module can partially prevent the problem of gradient disappearance in neural networks and enables DeepTGI's network to be designed deeper. Also, by standardizing the output of features, layer normalization can stabilize the internal covariant shift of the model and at the same time speed up the convergence speed of the model during training. In our model, layer normalization is utilized in the output of the self-attention layer and of the feed-forward neural network.

Loss function

We infer GRNs as a prediction task. Each TF–gene pair corresponds to two results, interact or not interact, so the prediction task is a binary classification. The datasets used in the analysis are all balanced. The binary cross entropy loss is applied to train the model, and the formulas are listed as follows:

$$BCE\ loss = -\frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} y_i \cdot \log(p_i) + \log(1-p_i)\cdot(1-y_i)$$

where y_i and p_i represent the label and prediction corresponding to the ith data, and N is the number of the training sample. Besides, mean squared error loss is also employed to be an auxiliary loss that regularizes the extra output of the Auto-encoder.

$$MSE\ loss = \frac{1}{\mathrm{N}}\sum_{i=1}^{n}(y_i - p_i)^2$$

Data augmentation

Data augmentation aims to improve performance of a model in the process of training as it increases the number of training samples when the model tries to learn more information from the limited original data without substantially enlarging the size of the dataset [24]. Mixup aims to increase the size of samples by mixing different samples from the training data. Thus, the augmented dataset can improve the generalization of the model and increase the robustness in the process of training. The formula for the Mixup is listed as follows:

$$\lambda = \mathrm{Beta}(\alpha, \beta)$$

$$\mathrm{Batch\_data}_x = \lambda \bullet \mathrm{Batch}_{x1} + (1-\lambda) \bullet \mathrm{Batch}_{x2}$$

$$\mathrm{Batch\_data}_y = \lambda \bullet \mathrm{Batch}_{y1} + (1-\lambda) \bullet \mathrm{Batch}_{y2}$$

where Batch_data_x is a set of batch data augmented by the Mixup method, and Batch_data_y is a set of labels corresponding to the augmented batch data. λ is the mixing coefficient that is calculated from the Beta distribution with the specific hyper-parameter α and β. $\mathbf{Batch}_{x1}$ is a set of batch data; $\mathbf{Batch}_{x2}$ is the shuffled batch samples derived from the $\mathbf{Batch\_data}_{x1}$. $\mathbf{Batch}_{y1}$ and $\mathbf{Batch}_{y2}$ are the corresponding labels of the $\mathbf{Batch}_{x1}$ and $\mathbf{Batch}_{y2}$.

Training and validation of models

We conducted training and testing of model performance by using different cell sizes/cell lineages and in different species. The positive samples are TF–gene pairs whose regulatory relationship is present in the gold standard network and gives label "1." By contrast, in the negative samples, the regulatory relationship is not in the standard network and gives label "0." Here, all the datasets are constructed as balanced as there are no reliable negative samples in gene regulatory inference, and the negative samples may be the false negative samples.

We then apply 5-fold cross-validation to evaluate the performance of models. All the metrices are calculated as the mean value in five folds. Each dataset is divided into a training set and testing set according to the ratio of 4:1, and the positive sample and negative sample are consistent with the whole dataset. As hyper-parameters play an important role in the training process, we introduce the parameter settings. The optimizer is the RAdam (Rectified Adam Optimizer) optimizer, and the activation function is Gaussian error linear unit. For the setting of network layers, every fully connected layer is followed by an activation layer, a batch normalization layer, and a dropout layer. The dropout rate is 0.3, and the learning rate is 1e-5. All the predicted results provided in the figures are based on the testing set. The models mentioned in this study are trained and tested on the Linux platform Ubuntu-20.04 with configurations Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz Graphics Processing Unit (GPU) @ Nvidia Quadro RTX 8000.

According to all the predicted scores, an optimal threshold is calculated to determine the regulatory relationship of the TF–gene pair, and then, the predicted GRN can be constructed by comparing the predicted scores to the threshold. The threshold is computed from the Youden index [25]. As the datasets we constructed are all balanced, the number of positive samples is equal to negative samples. Thus, the data for training and testing, which are randomly sampled from the datasets, are balanced too. We calculate six metrices with TPR (true positive rate), FPR (false positive rate), TNR (true negative rate), and FNR (false negative rate), and then use AUROC (area under a receiver operating characteristic curve), AUPR (area under precision–recall curve), ACC (Accuracy), and F1-scores to make a comprehensive evaluation of the performance of the model. The ROC curve is drawn by TPR and FPR, whereas the PR curve is according to Recall and Precision. These corresponding indicators are defined as the following equations:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Similar to the AUPR, F1-score is also an assessment metric constructed based on the Precision and Recall values, calculated by the following formula:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

ACC is the percentage of samples that the model can predict accurately from the total input samples, calculated as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

TP (true positive) refers to the predicted results existing in both the predicted TGIs and the golden standard networks, but FP (false positive) to those in the predicted but not in the golden standard networks. TN (true negative) represents those neither in the predicted nor in the gold standard and FN (false negative) for those not in the predicted but in the gold standard.

Statistical analysis

We calculated differences of the metrices between DeepTGI and other models/methods by using ANOVA and *t*-test. *P*-values $\leq .05$ indicate significant differences of the metrices. We used DAVID online software (https://david.ncifcrf.gov) to analyze the enrichment of Gene Ontology biological processes and Kyoto Encyclopedia of Genes and Genomes pathways among the target genes of TFs predicted by DeepTGI. Based on Fisher's exact test, statistical scores were then assigned to significantly enriched biological functions with *P*-values $\leq .01$. We calculated PCCs between expression values of TFs and genes. The *P*-values of PCCs $\leq .05$ are considered significant.

Results

Evaluating performance of DeepTGI by using bulk/single-cell transcriptomics data

To evaluate the performance of our DeepTGI system, two types of real mRNA expression data derived from bulk and single-cell experiments were divided into training and testing sets. The training sets were subjected to input, and the information contained in the gene expression was trained with the known TGIs. Then, we predicted new interactions from the unknown part. The predicted result could be used to evaluate the performance of the model.

We carried out the training–testing processes by using bulk and single-cell transcriptomic data, separately and together. Six metrices AUROC, AUPR, ACC, F1-score, Recall, and Precision were calculated for comparative analysis of different methods. Among unsupervised methods, GENIE3, ARACNE, and WGCNA utilized the expression profile of bulk transcriptomic data, while PPCOR, PIDC, and SCENIC used scRNA-seq data. We also compared the recently reported DGRNS, a supervised and deep learning method to infer GRNs from long short-term memory and CNN modules [14]. When either bulk or single-cell data were involved, the DeepTGI method held the highest levels of all six metrices with significant differences (Table 1). DGRNS achieved the second highest levels, supporting the robustness of deep learning algorithms in prediction of TGIs.

Next, we calculated TPR, FPR, TNR, and FNR using the combined data from bulk and single-cell experiments. DeepTGI obtained a higher TPR and TNR than other methods DGRNS, SCENIC, and GENIE3. On the contrary, the corresponding FPR and FNR of DeepTGI are the lowest (Supplementary Fig. S1A and B).

Table 1. Performance comparison of DeepTGI with other methods for prediction of TGIs in mESCs/miPSCs.

Transcriptome data types	Method	Testing set					
		AUROC	ACC	F1-score	Recall	Precision	AUPR
Bulk	GENIE3	0.53	0.52	0.51	0.51	0.53	0.53
	ARACNE	0.50	0.49	0.03	0.01	0.63	0.55
	WGCNA	0.54	0.54	0.50	0.48	0.54	0.54
	DeepTGI	**0.93**	**0.86**	**0.86**	**0.88**	**0.84**	**0.92**
Single cell	PPCOR	0.50	0.50	0.35	0.31	0.52	0.51
	PIDC	0.51	0.51	0.43	0.38	0.52	0.51
	SCENIC	0.51	0.51	0.35	0.27	0.53	0.52
	DGRNS	0.63	0.59	0.59	0.59	0.59	0.62
	DeepTGI	**0.94**	**0.87**	**0.87**	**0.90**	**0.85**	**0.91**
Combined bulk and single cell	**DeepTGI**	**0.94**	**0.87**	**0.88**	**0.90**	**0.85**	**0.93**

The bold values indicate the results obtained from DeepTGI.

Table 2. Comparison of feature fusion with different approaches for prediction of TF-gene interactions (TGIs) in miPSCs by using DeepTGI.

Fusion approaches	AUROC	ACC	F1-score	Recall	Precision	AUPR
CNN	0.50	0.51	0.53	0.59	0.51	0.50
CNN + Siamese	0.92	0.85	0.86	0.88	0.84	0.90
CNN + Siamese + Auto-encoder	0.93	0.86	0.86	0.90	0.83	0.91
CNN + Siamese + Auto-encoder + multihead attention	**0.94**	**0.87**	**0.88**	**0.91**	**0.84**	**0.92**

The bold values indicate the results obtained from a model including all four approaches.

About half of its prediction scores of TPR obtained from single-cell data were distributed to a range of 0.9 to 1.0 (Supplementary Fig. S1C). Thus, we gained a higher-level ROC curve generated by DeepTGI than other methods (Fig. 2A). Similarly, DeepTGI achieved the highest level of Precision–Recall curve (Fig. 2B). This finding supports the power of DeepTGI to identify new TGIs from both bulk and single-cell data. In addition, we measured the binary cross entropy loss that can describe how many losses cross different datasets during the training epochs of models. Supplementary Fig. S1D delineates the trajectory of training loss across bulk or single-cell transcriptomics alone and a combination of both during DeepTGI processing. DeepTGI converged to a loss below 0.007 on all three datasets before 80 epochs. Especially, when single-cell data were incorporated, the convergence rate was much faster than that from bulk data.

The DeepTGI framework is equipped with four network approaches to extract features obtained from transcriptomic data. We investigated the effect of Auto-encoder, 2DCNN-1DCNN, Siamese, and multihead attention by ablation experiments (Table 2). The result indicates that the integration of all four components could achieve the highest metrices. If two or three networks are combined, the performance of TGI prediction slightly declines. However, all metrices decrease if only the CNN was used.

Because there are a large number and different sets of TFs included in TGI prediction, we examined the effect of different numbers of TFs on the evaluation. The order of top number TFs is defined according to their target gene numbers by descending order. For example, the top 40 TFs represent those that TFs have the largest number of the predicted target genes from 40 of all TFs. In Table 3, we observed an increase in most of the metrices as the number of top TFs becomes larger, like AUROC, ACC, and Recall. However, for AUPR and Precision, their levels slightly decrease.

It is well known that Pou5f1, Sox2, and Nanog are three key TFs to maintain the pluripotency and stemness of embryonic stem cells [26]. By focusing on the three TFs with biological significance, as shown in Supplementary Table S2 and Fig. 2C, we predicted a total of 284 and 216 target genes of Pou5f1 by DeepTGI and DGRNS, respectively. Both of the results suggested Pou5f1 as the top one hub TF in the mESC network. Other methods like SCENIC and GENIE3 only predicted 124 and 134 targets of Pou5f1 (Fig. 2C), which is much less than DeepTGI or DGRNS. Noticeably, >99% of target genes of Pou5f1 predicted by the other methods can be covered by the prediction of DeepTGI. There were 140 and 126 target genes of Sox2 and Nanog predicted by DeepTGI that are more than twice as much as the predictions of SCENIC and GENIE3. Similar to Pou5f1 target genes, DeepTGI can capture >98% of Sox2 or Nanog target genes predicted by other methods (Fig. 2C). This result suggests that DeepTGI holds higher sensitivity than other methods, as well as its capability to recover GRN of biological meanings. We next calculated the correlation of TGIs predicted by DeepTGI from different cell numbers and found a slow increase in ACC of the predictions when the cell number is >200 (Fig. 3C). Moreover, we randomly selected a certain number of cells to reach the final numbers of the 17 groups. We also randomly sampled the single-cell transcriptomic data. In each group, sampling was repeated 10 times, and DeepTGI predicted 10 sets of TGIs. The predicted ACC from randomly selected cells increased as the cell number increases, indicating that our approach captures more accurate TGIs as cell number increases (Supplementary Fig. S2).

Validation and evaluation of DeepTGI under different factors

Through a 5-fold cross-validation by using the mESCs/miPSCs combined dataset, DeepTGI shows stable and high AUROC and ACC levels on every fold (Fig. 3A), indicating a similar performance among different folds. It is expected that many factors may affect the performance of deep learning models, for example, the number of cells from single-cell data and input features. Here, we examined the effect of cell numbers and input features on

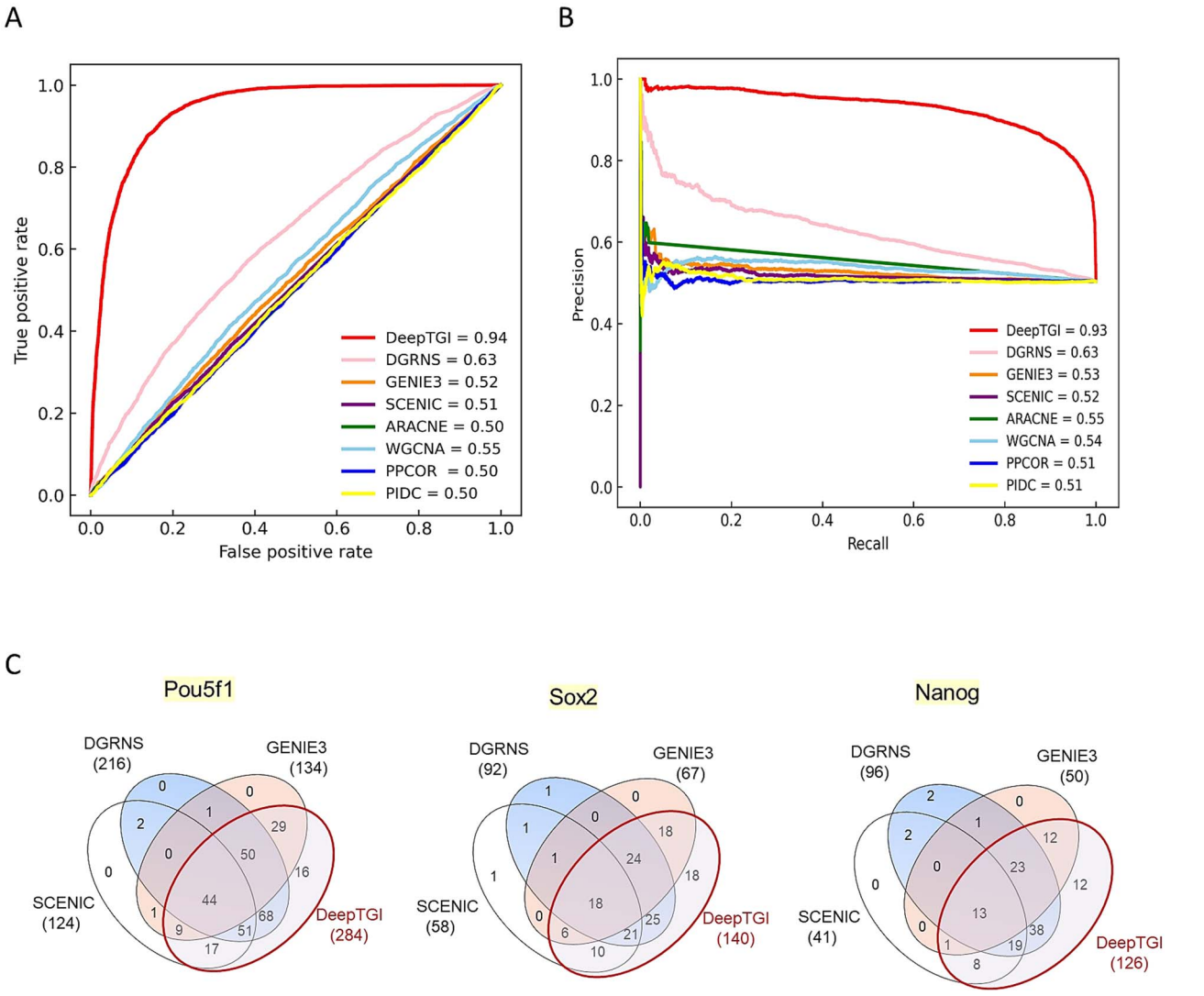

Figure 2. Comparisons of performance between DeepTGI and other benchmarking algorithms or methods. The combination of bulk transcriptomics of mESCs and single-cell transcriptomics of miPSCs were used as input data. (A) ROC curves and AUROC. (B), Precision–recall curves and AUPR. (C) Venn diagram of the predicted target genes of three key TFs Pou5f1, Sox2, and Nanog by using DeepTGI, DGRNS, SCENIC, and GENIE3.

Table 3. Performance comparison of different numbers of top TFs used to predict TF-gene interactions (TGIs)[a].

Number of TFs included[b]	AUROC	ACC	F1-score	Recall	Precision	AUPR
Top 40 TFs	0.70	0.69	0.81	0.70	0.96	0.97
Top 60 TFs	0.75	0.73	0.83	0.74	0.95	0.96
Top 80 TFs	0.78	0.75	0.84	0.76	0.94	0.95
Top 100 TFs	0.80	0.75	0.83	0.75	0.94	0.95
All TFs	0.94	0.87	0.87	0.90	0.85	0.91

[a]TGIs were predicted by DeepTGI using mESC-miPSC combined data. [b]Number of TFs from all TFs were included in calculation of the metrices.

the DeepTGI function. Figure 3B displays an increasing trend of ACC when the number of cells increased from 10 to all, forming 17 groups containing 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, and all cells, regardless of whether the input feature is expression profiles or PCCs. DeepTGI shows stable and satisfactory performance >0.8 when the cell number is >200, although the ACC drops when the cell numbers are small. Using PCCs as input features instead of single-cell expression profiles significantly reduces the sparsity of data matrices, which is known to be detrimental to the performance of deep learning models. In Fig. 3B, the ACC derived from DeepTGI using PCCs were compared with using expression value without PCCs as input features. In general, DeepTGI achieved a higher ACC when using PCCs in all experiments of different cell numbers.

We next calculated the correlation of TGIs predicted by DeepTGI from different cell numbers and found a slow increase in ACC of the predictions when the cell number is >200 (Fig. 3C). Moreover, we randomly selected a certain number of cells to reach the final numbers of the 17 groups. We also randomly sampled

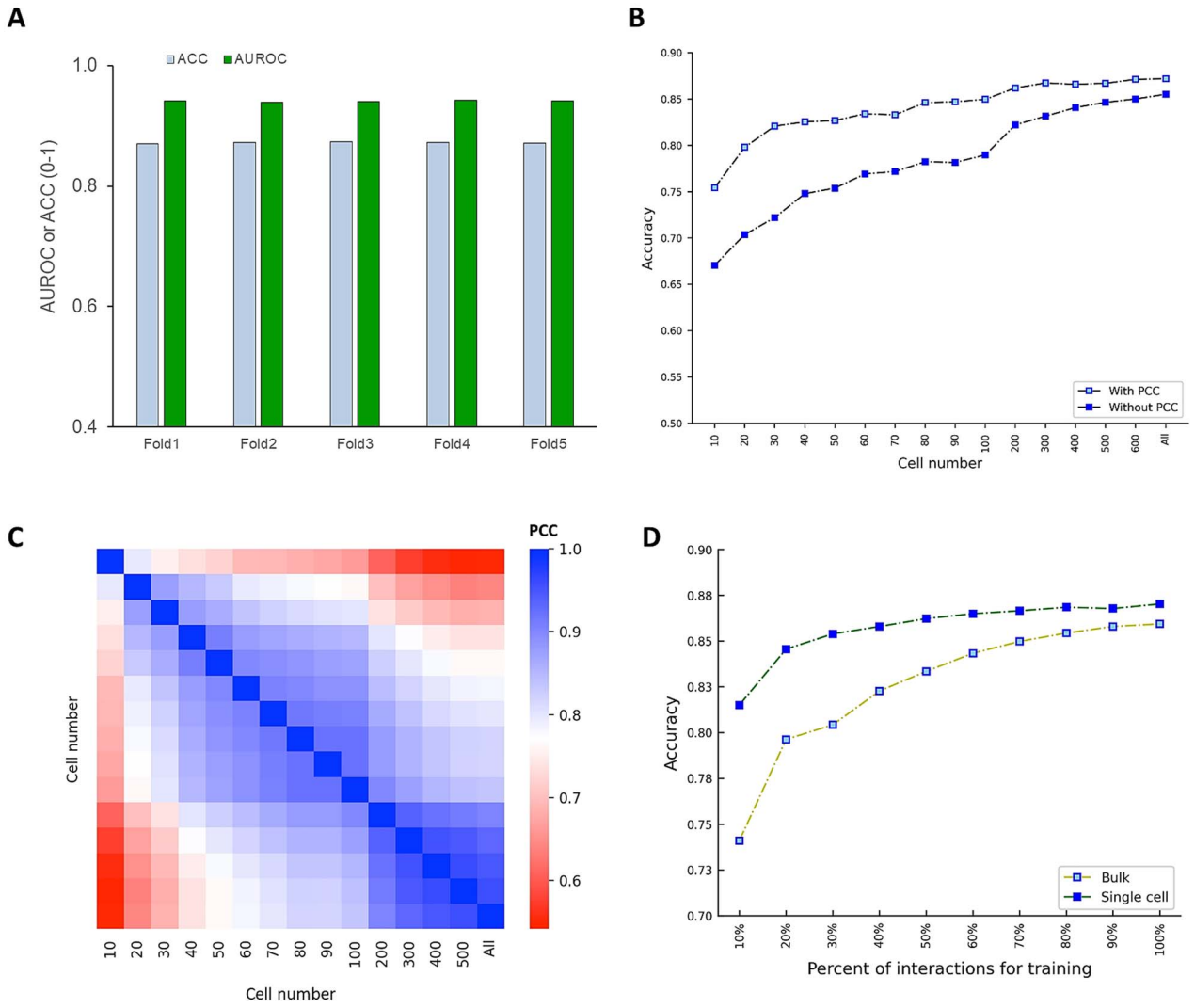

Figure 3. Validation for the robustness of the newly developed DeepTGI algorithm. (A) Through a 5-fold cross-validation. The consistency in AUROC and ACC across different partitions between training and testing sets. (B) Comparison of performance of DeepTGI across varying cell counts within single-cell transcriptomic data and under inclusion or noninclusion of PCCs in the model. (C) A heatmap of PCC regulatory networks from different cell count groupings, further emphasizing the influence of cell quantity on prediction accuracy. (D) Randomly picked up 10 sets, from 10%, 20%, 30%, . . . to 100% of the known TGIs, and kept the TGIs for testing the same as the previous set.

the single-cell transcriptomic data. In each group, sampling was repeated 10 times, and DeepTGI predicted 10 sets of TGIs. The predicted ACC from randomly selected cells increases as the cell number increases, indicating that our approach captures more accurate TGIs as the cell number increases (Supplementary Fig. S2).

Furthermore, we examined the robustness of DeepTGI when different numbers of known TGIs were selected for model training. We randomly took 1 from 10 sets, i.e., 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the known interactions and meanwhile kept the TGIs for testing the same as the previous set. A slight increase of ACC from 0.81 to 0.87 was observed as percentages of the known interactions increased in single-cell data, while, for bulk data, from 0.74 to 0.85 (Fig. 3D). It is possible that single-cell data contain more factors such as diversity of gene expression in different cells, cell numbers, or cell type characteristics and so achieves higher ACC than using bulk data.

In addition, the multihead attention mechanism may increase computational complexity and running times of DeepTGI. We examined the effect of sample sizes on processing time and compared the times to run different models. Even with GPU acceleration, the processing time increases as the dataset size grows (Supplementary Table S3A). From Supplementary Table S3B, we found a large variety of running times among different methods. Comparatively, DeepTGI does not require the longest time. Moreover, there are substantial computational demands for DeepTGI based on Supplementary Table S3C, including a total of 124 840 395 floating point operations. Also, the model comprises 108 284 328 parameters, which not only affect memory usage but also increase processing times.

Performance in different biological systems

The findings described above indicate the effectiveness of DeepTGI in inferring regulatory interactions between TFs and genes in mESCs/miPSCs. To further evaluate its performance in different species and cell lineages, we analyzed scRNA-seq data from two mouse mHSC-E or mHSC-L and two human hESCs and hHEPs by DeepTGI and obtained the target gene number of top TFs (Supplementary Table S4). Supplementary Table S5 shows the predicted target genes of the top 20 TFs from the

four datasets. We compared the result between DeepTGI and other methods DGRNS, PIDC, GENIE3, and PPCOR. Considering different numbers of TFs included, respectively, in these datasets, we also examined the effects of all or half of the TFs on the prediction. The 50% of TFs were chosen based on their target gene numbers in descending order. DeepTGI displays the highest levels of AUROC and ACC in all these human and mouse datasets whatever different amounts of TFs are included (Fig. 4A and B). Especially in hESC, DeepTGI produced AUROC (0.93–0.94) and ACC (0.86–0.89) with at least 30% higher than other methods, respectively.

Regarding deep learning–based methods, DeepTGI outperforms DGRNS on all four datasets and achieves better metrices than DGRNS. However, DGRNS is better than all other traditional methods, which further demonstrates that the deep learning model could extract more effective information and can be applied to predict TGIs and infer GRNs.

Application of DeepTGI to construct gene regulatory networks

To further demonstrate the capability of DeepTGI in the inference of TGIs, we trained the model with 50% of total interactions and then made predictions for the entire dataset to explore hub TFs and potential regulations. The predictions of TF–gene pairs that are higher than 0.5 are considered for further investigation, and the inferred GRNs are visualized. We first organized the DeepTGI predicted TGIs and built a GRN corresponding to reprogramming of miPSCs. Overall, as shown in Fig. 5A, these top 26 TFs were considered as hub TFs, and the GRN controlled by the TFs related to stem cell differentiation can be observed. A set of 151 interactions in the inferred network can be confirmed by the cell type–specific network. The target genes of the top TFs are found significantly associated with biological processes and signaling pathways of pluripotency, stem cell differentiation, cell fate determination & commitment, cell differentiation, Wnt, MAPK, and Bmp pathways, etc. (Fig. 5B). Thus, we connected the main TFs that play critical roles in pluripotency and stemness with the downstream target genes predicted by our model (Fig. 5C). It is noticed that three key TFs Pou5f1, Sox2, and Nanog form a core GRN by cooperating with other TFs or factors, like Klf4, Suz12, etc., and would modulate the major biological functions and signaling pathways that are beneficial for pluripotency and stemness maintenance.

Similarly, we constructed GRNs of mHSC-E and mHSC-L (Supplementary Fig. S3), where the target genes of the top 30 TFs were associated with cell cycle, apoptosis, and cell proliferation, but with DNA replication for mHSC-E and immune & inflammation responses, MAPK for mHSC-L. DeepTGI can identify hub TFs, recover known regulations, and explore potential TGIs based on available information, which offers essential biological information for researchers to explore the GRNs in the massive data. Moreover, we described human GRNs from hESC and hHEP datasets (Supplementary Fig. S4A), respectively. Among the top 30 TFs, the GRN of hESC is significantly associated with stem cell development, such as stem cell pluripotency, Polycomb complex, chromatin remodeling & development, etc. GRNs of hHEPs are related to cell cycles, DNA replication, RNA splicing, etc. (Supplementary Fig. S4B). There are many predictions that are not confirmed by the biological experimental means yet, but these predicted regulatory factor–gene interrelations provide essential information for the regulatory network and are necessary for further research.

Discussion

The newly developed DeepTGI shows its power for prediction of TGIs and reconstruction of GRNs. The remarkable performance of DeepTGI is possibly because of its multisource information fusion strategy. This approach integrates multisource input data into the model. As scRNA-seq data are sparse and noisy, DeepTGI converts the original expression profile into a series of PCC vectors. Combined information of single-cell and bulk data allows the model to simultaneously process different gene expression profiles at individual cell and tissue levels. As a result, DeepTGI thereby will discover more potential and truer TGIs. Moreover, in terms of feature extraction, DeepTGI effectively fuses key information from transcriptomics data by using various approaches Auto-encoder, 2DCNN-1DCNN, and Siamese networks. The combination of this information enables DeepTGI to comprehensively capture the characteristics of transcriptomic data from multiple resources. Finally, the multihead attention mechanism introduced in DeepTGI further optimizes the feature fusion from the three networks. By processing multiple attention "heads" in parallel, the mechanism could focus on different aspects of different features separately and assign appropriate weights to them. This designing not only improves the efficiency of feature fusion but also enables the model to focus on more information when making predictions, thus improving the accuracy and efficiency of predictions. The application of the multihead attention mechanism makes the model more flexible when processing complex and high-dimensional data.

As a supervised method, DeepTGI encodes the correlation vectors with Auto-encoder and applies a self-attention mechanism that extracts critical features from the latent vectors. It makes effective predictions of regulatory interactions and the formed GRNs. We have evaluated DeepTGI with comprehensive experiments on different expression datasets involving different scales, cell types, and species. The comparative analysis on the performance of various unsupervised and supervised methods shows that DeepTGI achieves the best results in all cases, demonstrating its superiority in predicting the TF–gene relationship and capturing more potential interactions. For the GRNs inferred by DeepTGI, many interactions can be validated from existing gold standard networks, such as GRNs inferred from miPSC data. These identified potential interactions provide a broader perspective for the discovery of more gene–gene relationships that may form more biologically meaningful networks.

Deep learning-derived methods show their advantage of TGI identification and GRN reconstruction because they can integrate multiple information of gene expression data and known gene interaction, such as gate recurrent unit [10–12] and DGRNS [14]. Our result indicates that DeepTGI performs better than DGRNS in predicting TGIs. It is possibly because of DeepTGI's optimization of its overall architecture design, such as the addition of the self-attention layer and the improvement of the feature fusion strategy. This not only improves the efficiency of the algorithm in processing data but also enhances its adaptability to different biological backgrounds and experimental conditions. The optimization of this design ensures the model with good generalization ability and prediction accuracy on different datasets, proving the effectiveness and practicality of DeepTGI in a variety of biological application scenarios. However, there are substantial computational demands for DeepTGI, which not only affect memory usage but also contribute to increasing processing times. For example, the multihead attention mechanisms in DeepTGI can obtain the latent features extracted from multisources; however,

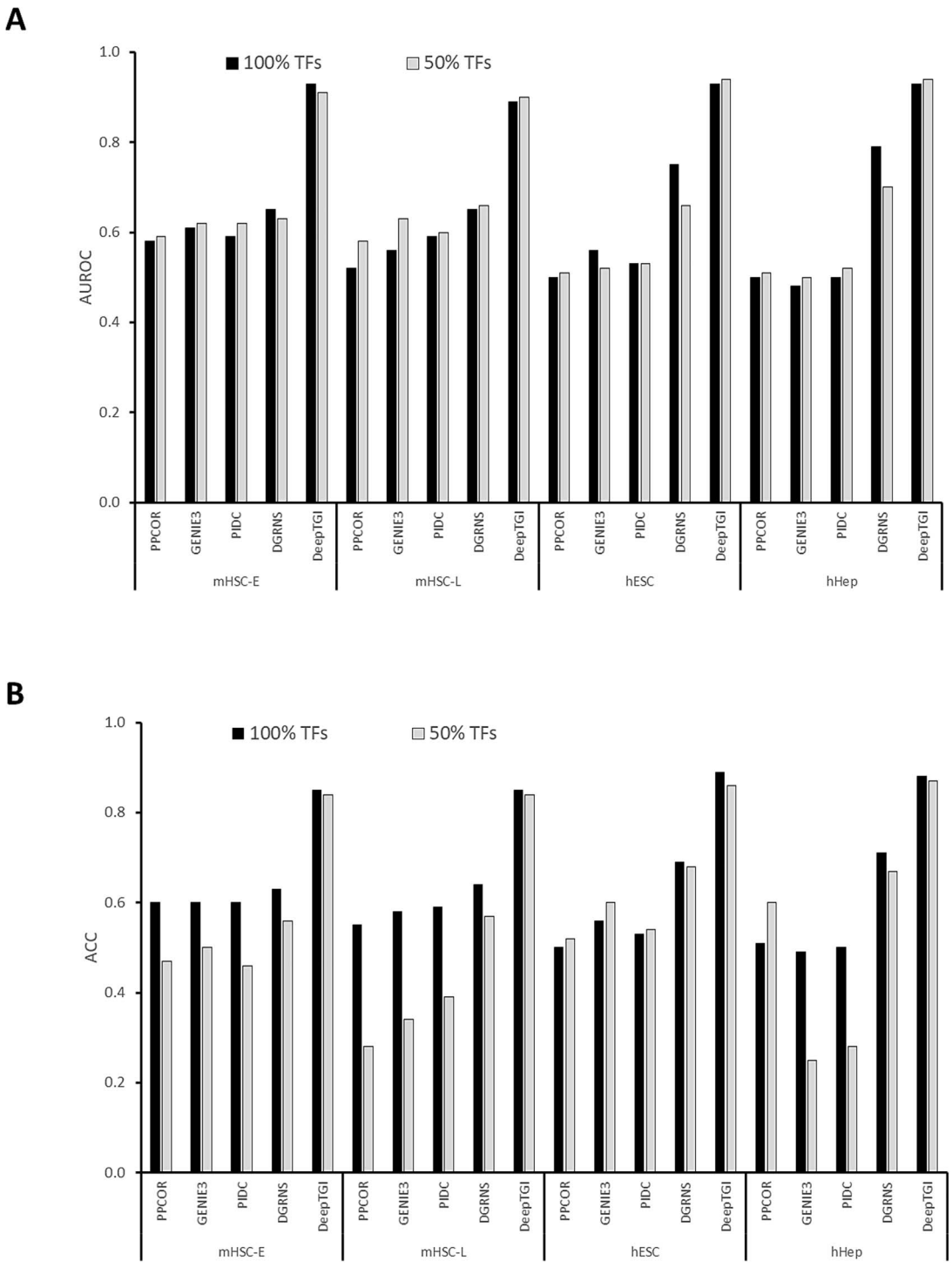

Figure 4. Comparison of AUROC and ACC values achieved by DeepTGI with other supervised or unsupervised methods. All 100% or 50% of TFs were included in the calculation of AUROC (A) and ACC (B), respectively. Four datasets represent different scales and cell lineages of human and mouse (see Supplementary Table S1).

it would increase computational complexity and extend running time. This challenges its scalability when applied to analyze large-scale datasets. The increased complexity necessitates careful consideration of computational resources, particularly in contexts involving extensive single-cell RNA profiles and transcriptional gene regulation.

In conclusion, the excellent performance of the DeepTGI algorithm describes predicting TGI and inferring GRNs stems from its series of innovations in multisource information fusion, advanced feature extraction technology, and the application of multihead attention mechanisms. Overall, DeepTGI is more accurate than the unsupervised and other deep learning methods and exhibits better performance in large-scale network inference. This method can incorporate crucial information contained in the gene expression profile and prior knowledge of the known interactions to make more effective TGI prediction and GRN inference. These advantages not only deepen our understanding of the complexity of GRNs but also provide new perspectives and powerful tools for

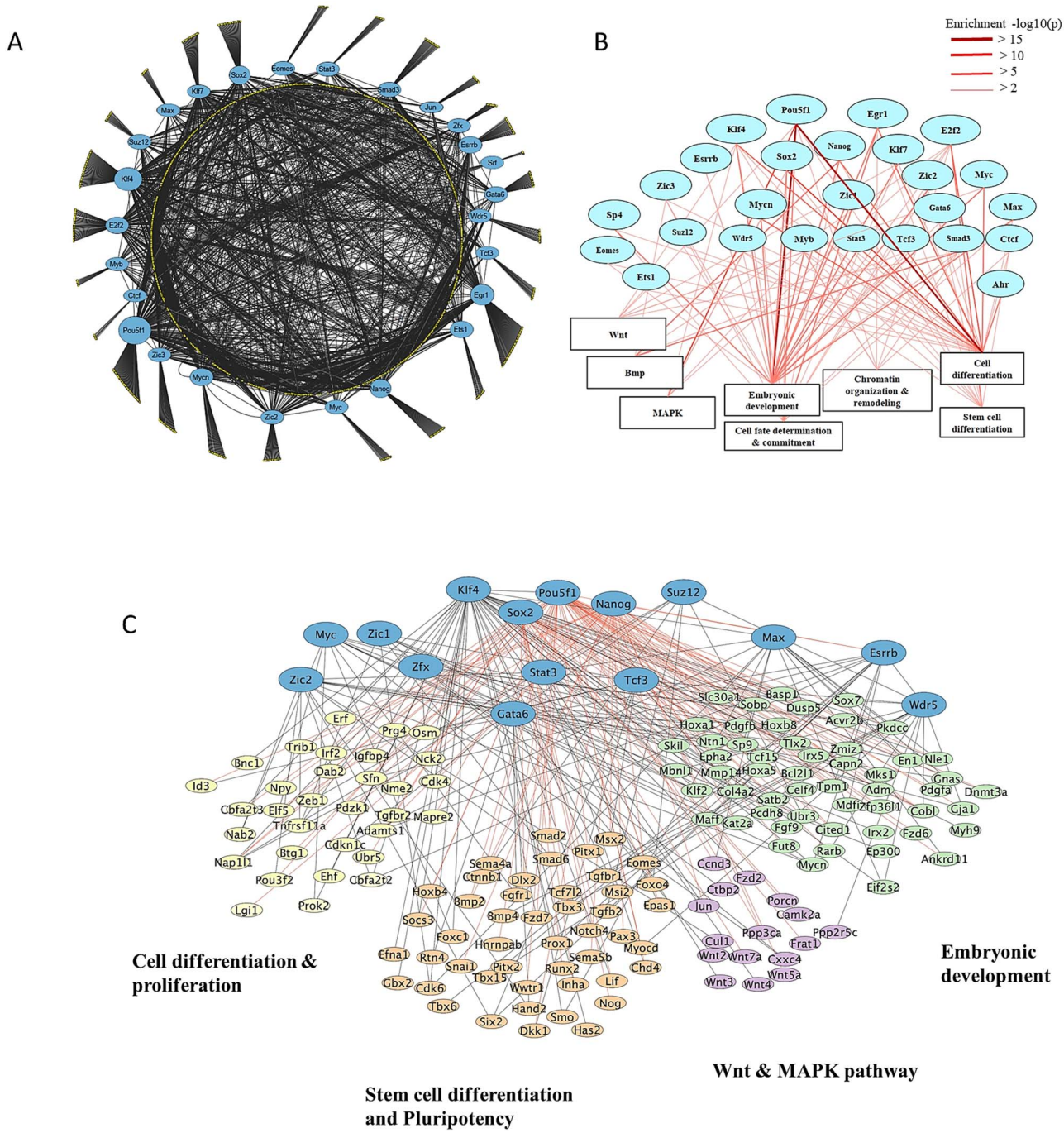

Figure 5. GRNs of miPSCs inferred by DeepTGI. (A) The GRN controlled by top 26 TFs during reprogramming of miPSCs. (B) The GRN controlled by the 26 TFs associated with biological processes and signaling pathways of pluripotency and stemness. The red edges with varying darkness and thickness represent enrichment (−log2 *P*-values) of overlaps between TF target genes and biological processes/pathways based on hypergeometric distribution. In (A, B), the node size of a TF corresponds to the number of TF target genes. (C) A GRN represents regulatory interactions among major TFs and targets that regulate pluripotency and stemness. The red edges indicate the TGIs of three pluripotency TFs Pou5f1, Sox2, and Nanog.

research and applications in the field of bioinformatics. However, there are still some factors that might limit the efficiency of our model because of various datasets from broad types of single-cell experiments, high computational demands, and complexity of regulatory mechanisms. Thus, in the future study, we will improve the model by coordinating other learning algorithms to establish a broad deep learning system that can process larger and diverse biomedical data for constructing GRNs. It is expected that the developed model system can integrate multiple regulatory layers from NEAT-seq [27], ATAC-seq [28], ChIP-seq, and the gene expression profile. We believe that this would extend the function of DeepTGI or other models and to capture TF–target GRNs efficiently.

Key Points

- Introduce a novel deep learning model system with superior ability to predict interactions between transcription factors and target genes and infer gene

regulatory networks from bulk and single-cell transcriptomic data.
- Integrate multiple vectors with self-attention mechanism and transform multi-head attention modules to capture representative features.
- Provide broader perspectives for the discovery of more biological meaningful networks and for understanding gene regulatory mechanisms.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Acknowledgements

We thank HKU ITS, GXUN, and SYSU-SZ for the computing facilities (HPC/AI) provided.

Funding

J.W. was supported by collaborative research fund of Hong Kong Research Grants Council [C7015-23G], seed funding for collaborative research [2207101590] and basic research [2201101499] from the University of Hong Kong, and startup/matching funds [207051059, 6010309] from Faculty of Dentistry, the University of Hong Kong (after 1 August 2022), and the National Institutes of Health of USA [R01LM013438, before 1 August 2022]. J.Q. was supported by the National Natural Science Foundation of China [32170655, 12071306], Shenzhen Science and Technology Program [No. 202206193000001, 20220817122906001], and the Natural Science Foundation of Guangdong Province [2024A1515011210].

Author contributions

Yong Liu: Conceptualization, Methodology, Project Administration, Resources, Supervision, Writing - Original Draft, Writing - Review & Editing; Le Zhong: Data Curation, Formal Analysis, Investigation, Methodology, Software, Visualization, Writing - Original Draft, Writing - Review & Editing; Bin Yan: Data Curation, Formal Analysis, Investigation, Methodology, Writing - Original Draft, Writing - Review & Editing; Zhuobin Chen: Data Curation, Formal Analysis, Resources, Validation, Writing - Review & Editing; Yanjia Yu: Data Curation, Formal Analysis, Investigation, Validation, Writing - Original Draft, Writing - Review & Editing; Dan Yu: Formal Analysis, Investigation, Writing - Review & Editing; Jing Qin: Data Curation, Funding Acquisition, Project Administration, Resources, Supervision, Writing - Original Draft, Writing - Review & Editing; Junwen Wang: Conceptualization, Funding Acquisition, Methodology, Project Administration, Resources, Supervision, Writing - Original Draft, Writing - Review & Editing.

Code and data availability

A Python implementation of the DeepTGI framework is available at https://github.com/wanglabhku/DeepTGI. We are committed to supporting and facilitating scientific inquiry and innovation. To ensure the integrity and reproducibility of our research, we are prepared to share the code and data with researchers in the field following a review of the request's purpose and scope. Interested parties are encouraged to contact the corresponding author to discuss the terms of access. This approach is in alignment with our ethical responsibility to prevent potential misuse while promoting a collaborative scientific environment.

Ethics approval and consent to participate

Not applicable.

References

1. Huynh-Thu VA, Irrthum A, Wehenkel L. *et al.* Inferring regulatory networks from expression data using tree-based methods. *PloS One* 2010;**5**:e12776. https://doi.org/10.1371/journal.pone.0012776.
2. Langfelder P, Horvath S. WGCNA: An R package for weighted correlation network analysis. *BMC Bioinformatics* 2008;**9**:1–13. https://doi.org/10.1186/1471-2105-9-559.
3. Margolin AA, Nemenman I, Basso K. *et al.* ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 2006;**7**:1–15. https://doi.org/10.1186/1471-2105-7-S1-S7.
4. Chan TE, Stumpf MP, Babtie AC. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Systems* 2017;**5**:251, 251–267.e3. https://doi.org/10.1016/j.cels.2017.08.014.
5. Kim S. Ppcor: An R package for a fast calculation to semi-partial correlation coefficients. *Communications for Statistical Applications and Methods* 2015;**22**:665. https://doi.org/10.5351/CSAM.2015.22.6.665.
6. Aibar S, González-Blas CB, Moerman T. *et al.* SCENIC: Single-cell regulatory network inference and clustering. *Nat Methods* 2017;**14**:1083–6. https://doi.org/10.1038/nmeth.4463.
7. Razaghi-Moghadam Z, Nikoloski Z. Supervised learning of gene-regulatory networks based on graph distance profiles of transcriptomics data. *NPJ Systems Biology and Applications* 2020;**6**:21. https://doi.org/10.1038/s41540-020-0140-1.
8. Shu H, Zhou J, Lian Q. *et al.* Modeling gene regulatory networks using neural network architectures. *Nature Computational Science* 2021;**1**:491–501. https://doi.org/10.1038/s43588-021-00099-8.
9. Yuan Y, Bar-Joseph Z. Deep learning for inferring gene relationships from single-cell expression data. *Proc Natl Acad Sci* 2019;**116**:27151–8. https://doi.org/10.1073/pnas.1911536116.
10. Angermueller C, Parnamaa T, Parts L. *et al.* Deep learning for computational biology. *Mol Syst Biol* 2016;**12**:878. https://doi.org/10.15252/msb.20156651.
11. Eraslan G, Avsec Z, Gagneur J. *et al.* Deep learning: New computational modelling techniques for genomics. *Nat Rev Genet* 2019;**20**:389–403. https://doi.org/10.1038/s41576-019-0122-6.
12. Jin S, Zeng X, Xia F. *et al.* Application of deep learning methods in biological networks. *Brief Bioinform* 2021;**22**:1902–17. https://doi.org/10.1093/bib/bbaa043.
13. Kc K, Li R, Cui F. *et al.* GNE: A deep learning framework for gene network inference by aggregating biological information. *BMC Syst Biol* 2019;**13**:1–14. https://doi.org/10.1186/s12918-019-0694-y.
14. Zhao M, He W, Tang J. *et al.* A hybrid deep learning framework for gene regulatory network inference from single-cell transcriptomic data. *Brief Bioinform* 2022;**23**:bbab568. https://doi.org/10.1093/bib/bbab568.
15. Guo L, Lin L, Wang X. *et al.* Resolving cell fate decisions during somatic cell reprogramming by single-cell RNA-Seq. *Mol Cell* 2019;**73**:e817. https://doi.org/10.1016/j.molcel.2019.01.042.

16. Qin J, Li MJ, Wang P. *et al.* ChIP-Array: Combinatory analysis of ChIP-seq/chip and microarray gene expression data to discover direct/indirect targets of a transcription factor. *Nucleic Acids Res* 2011;**39**:W430–6. https://doi.org/10.1093/nar/gkr332.
17. Wang P, Qin J, Qin Y. *et al.* ChIP-Array 2: Integrating multiple omics data to construct gene regulatory networks. *Nucleic Acids Res* 2015;**43**:W264–9. https://doi.org/10.1093/nar/gkv398.
18. Pratapa A, Jalihal AP, Law JN. *et al.* Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat Methods* 2020;**17**:147–54. https://doi.org/10.1038/s41592-019-0690-6.
19. Kingma DP, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 2013.
20. Devlin J, Chang M-W, Lee K. *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018.
21. Liu X, Gao W, Li R. *et al.* One shot ancient character recognition with siamese similarity network. *Sci Rep* 2022;**12**:14820. https://doi.org/10.1038/s41598-022-18986-z.
22. Ashish V. Attention is all you need. *Advancs in Neural Information Processing Systems* 2017;**30**:I.
23. Turki T, Taguchi Y. SCGRNs: Novel supervised inference of single-cell gene regulatory networks of complex diseases. *Comput Biol Med* 2020;**118**:103656. https://doi.org/10.1016/j.compbiomed.2020.103656.
24. Cubuk ED, Zoph B, Mane D. *et al. Autoaugment: Learning Augmentation Strategies from Data*. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019. pp. 113–123. https://doi.org/10.48550/arXiv.1805.09501.
25. Youden WJ. Index for rating diagnostic tests. *Cancer* 1950;**3**:32–5. https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.
26. Li MJ, Sham PC, Wang J. FastPval: A fast and memory efficient program to calculate very low P-values from empirical distribution. *Bioinformatics* 2010;**26**:2897–9. https://doi.org/10.1093/bioinformatics/btq540.
27. Lin X, Jiang S, Gao L. *et al.* MultiSC: A deep learning pipeline for analyzing multiomics single-cell data. *Brief Bioinform* 2024;**25**:bbae492. https://doi.org/10.1093/bib/bbae492.
28. Lin Y, Wu T-Y, Wan S. *et al.* scJoint integrates atlas-scale single-cell RNA-seq and ATAC-seq data with transfer learning. *Nat Biotechnol* 2022;**40**:703–10. https://doi.org/10.1038/s41587-021-01161-6.

Briefings in Bioinformatics, 2025, **26(1)**, bbae639
https://doi.org/10.1093/bib/bbae639
Problem Solving Protocol