Gene Regulatory Network Inference through Link Prediction using Graph Neural Network

S. Ganeshamoorthy[1], L. Roden[2], D. Klepl[1] and F. He[1]

1. Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry, UK
2. Centre for Sport, Exercise and Life Sciences, Coventry University, Coventry, UK
{ganeshamos, klepld}@uni.coventry.ac.uk, {ad1301, fei.he}@coventry.ac.uk

I. Introduction

Gene Regulatory Networks (GRNs) depict the causal regulatory interactions between transcription factors (TFs) and their target genes [2], where TFs are proteins that regulate gene transcription. GRN plays a vital role in explaining gene function, which helps to identify and prioritize the candidate genes for functional analysis [3]. Currently, high-dimensional transcriptome datasets are produced from high-throughput sequencing techniques, such as microarray and RNA-Seq. These techniques can capture the differences in the expression of thousands of genes at once. Through these wet-lab experiments, studying the interconnections among a large number of genes or TFs at a network level is challenging [4]. Therefore, one of the important topics in computational biology is the inference of GRNs from high-dimensional gene expression data through statistical and machine learning approaches [2].

There is a vast literature on using machine learning and statistical methods to reconstruct GRN from gene expression data [2, 4]. Classical machine learning methods perform poorly on non-Euclidean objects, such as networks or graphs. More recently, deep learning techniques have been extended to graph-based learning approaches [2, 5]. The graph-based model identifies complex interconnections within a network instead of only learning two ends of a relationship [1, 2], which makes it a suitable method for GRN inference and has advantages over classical statistical inference approaches. Graph Neural Network (GNN) is one of the emerging graph-based methods [5]. This technique can learn node embeddings by aggregating information from topological neighbourhoods [2]. One of the GNN applications is link prediction, which can predict missing links between two nodes in a network [1].

The GRN inference problem can be handled through link prediction, i.e. predicting missing links between TFs and genes using the known links in the partially constructed network [1]. Adopting an autoencoder or variational autoencoder for predicting links has helped to attain great performance transductively. Although there are many types of architectures, graph convolution is popular among them [6]. Generally, the Graph Autoencoder(GAE) is based on a Graph Convolutional Network (GCN). The study conducted by Kipf and Welling [7] has designed a GAE along with a Variational Graph Autoencoder (VGAE), which is the probabilistic variant of the former. Relating to this, graph convolution based on the GAE model has recently been implemented to predict missing links in *E. coli* and yeast synthetic networks from the GeneNetWeaver dataset [5]. From this study, it is proved that VGAE performs better than GAE in GRN link prediction. On the other hand, to achieve superior accuracy in gene expression prediction, a one-dimensional Convolutional Neural Network (1D-CNN) is employed as a feature extraction technique [10]. Therefore, In this study, we propose a GCN-based graph autoencoder model with a 1D-CNN feature extraction module to enhance the performance of the GRN inference.

II. Data

The data used in this study is the gene-expression microarray data sets from *E. coli* in a Dialogue on Reverse Engineering Assessment and Methods (DREAM) project, more specifically, DREAM5 challenge [8]. *E. coli* gold standard network from the DREAM5 dataset consists of 4511 genes, 2066 verified interactions, and 805 samples. This dataset is used as a benchmark for training and testing the proposed method.

III. METHOD

The proposed GRN inference method can be implemented in two steps: 1) Feature Extraction using a 1D convolution neural network (1D-CNN) layer and 2) the GRN inference (link prediction) using GCN-based GAE or VGAE.

Feature Extraction using 1D-CNN: A graph can be represented as $G = (V, E)$; V represents a set of nodes, and E is a set of edges between the nodes. In a GRN, genes are the nodes, and edges are the relationships between the genes. To construct a GAE, this study uses gene expression as a node feature to learn the relationship in the graph. To improve the performance of link prediction using graph autoencoder, a feature learning approach has been implemented in the gene expression of each node. To learn the features, a 1D-CNN is used. This is shown in Figure 1(a), where the raw gene expression matrix is the input, and the output is a learned feature matrix. This extracted feature matrix is then used as the input of the GCN-based GAE (Figure 1(b)). This feature extraction module consists of two convolutional blocks. Each block consists of a 1D convolution (kernel size = 5, stride = 3), batch normalisation, ReLU activation function and a maximum pooling layer (kernel size = 5, stride = 2). The convolution layer is designed to extract the local features, and batch normalization is used to reduce overfitting. Since convolution is a linear operation, to activate nonlinearity, a ReLU activation function is used. Moreover, max-pooling is adopted to reduce dimensionality. The dropout layer (p = 0.2) is used to prevent overfitting.

GRN inference with GCN-based graph autoencoders: Autoencoder is a neural network consisting of both an encoder and a decoder. The function of the encoder is to project the input data into a low-dimensional latent space while the decoder reconstructs the input from the latent embedding. GAE architecture offers several advantages over other graph-based methods. Firstly, it has the capability to map the graph data into low-dimensional space. Secondly, it is one of the most effective graph embedding techniques among other graph-based approaches, which also helps to control the computational cost [9].

GCN-based graph autoencoder is illustrated in the bottom part of Figure 1, which has three main components: the input, which contains the node feature matrix and the adjacency matrix (Figure 1(b)), GCN-based Autoencoder (Figure 1(c)), and the output as the reconstructed (predicted) GRN (Figure 1(d)). Figure 1(b) demonstrates the input of a GCN-based graph autoencoder which contains an adjacency matrix A and node feature matrix X. The adjacency matrix indicates the presence of the corresponding link in the network by a matrix of booleans (0's and 1's). We propose two methods to obtain the node feature matrix. In the first method, we use the original gene expression (without feature learning), and in the second method, a 1D-CNN is used for feature learning which is further discussed in section 3.1. The feature matrix is the output of this model, and it is used as the input for the GCN-based graph autoencoder.

Figure 1(c) depicts GCN-based graph autoencoder methods. Two GCN-based graph autoencoder architectures are utilised in this study which are the GAE and VGAE. GAE is an extension of autoencoder to graphs. The encoder is a GCN and outputs a latent vector Z, where $Z = GCN(A, X)$. The decoder is executed through the inner product among latent vectors (Z) with a sigmoid activation function ($\sigma(x) = {}^{1}/_{1+e^{-x}}$) to reconstruct the adjacency matrix through learning the similarity of each node inside Z and outputs adjacency matrix $\hat{A}$, where $\hat{A} = \sigma(ZZ^T)$.

VGAE is an extension of GAE based on the variational auto-encoder [7]. The encoder of VGAE is shown in equation 1, where Z_i is the probabilistic version of the latent variable. Here μ_i is the matrix of the mean vectors, $\mu = GCN_\mu(A, X)$ and σ is the matrix of standard deviation vectors, $log\ \sigma = GCN_\sigma(A, X)$.

$$q(Z|A,X) = \prod_{i=1}^{N} q(z_i|A,X),\ where\ q(z_i|A,X) = \mathcal{N}(z_i|\mu_i, diag(\sigma^2)) \tag{1}$$

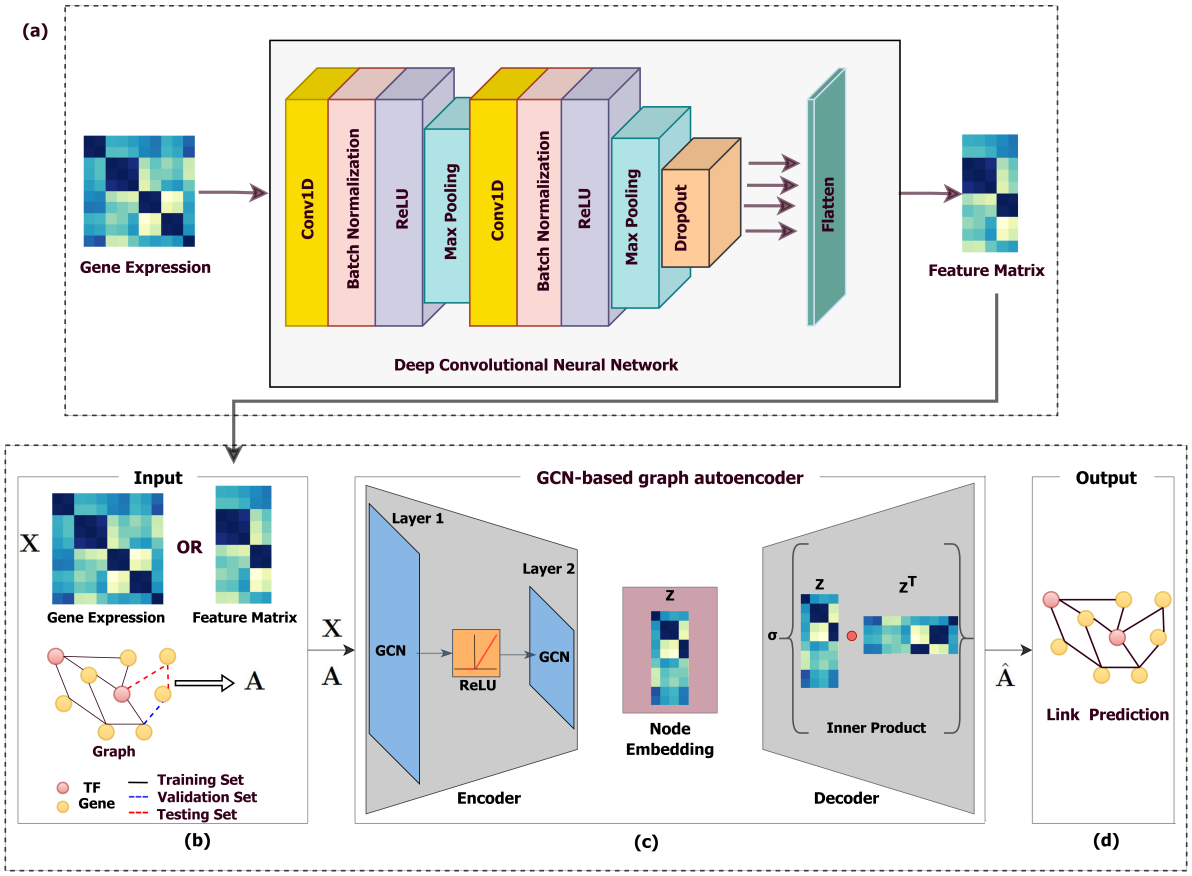

Figure 1. The overall architecture of a GCN-based graph autoencoder with 1D-CNN feature learning. The framework includes (a) a 1D-CNN-based feature learning, (b) The graph links (connections) are divided into a training set (80%), validation set (10%) and testing set (10%). Input for GCN-based graph autoencoder is adjacency matrix A and node feature matrix X where A is converted from the training set of graph links, and X is either raw Gene Expression or Feature matrix from the output of 1-D CNN (c) a GCN-based graph autoencoder (Encoder, Node Embedding, Decoder), and (d) the output is the predicted graph.

The decoder of VGAE is constructed similarly to before. The inner product among latent variables is given in equation 2, where A_{ij} is the elements of A and σ is the sigmoid activation function.

$$p(A|Z) = \prod_{i=1}^{N}\prod_{J=1}^{N} p(A_{ij}|z_i,z_j),\ where\ p(A_{ij}=1|z_i,z_j) = \sigma(z_i^T z_j) \tag{2}$$

Finally, equation 3 provides the function used to optimize the variational parameters. Here, $KL[q(.)||p(.)]$ is the Kulback-Leibler (KL) divergence among $q(.)$ and $p(.)$. $p(Z)$ is a Gaussian prior in the KL divergence, i.e. $p(Z) = \prod_i p(z_i) = \prod_i \mathcal{N}(z_i|0,I))$.

$$\mathscr{L} = E_{q(Z|A,X)}[log\ p(A|Z)]\ -\ KL[q(Z|A,X)\ ||\ p(Z)] \tag{3}$$

IV. Results and Conclusions

In this study, the GRN inference using GCN-based graph autoencoders has been conducted and compared using the following combination of methods. GAE method with 1D-CNN feature extractor module, GAE without feature learning, VGAE with 1D-CNN feature learning and VGAE method with 1D-CNN feature learning.

Table 1. Evaluation of proposed methods

Model	AUC	Precision
GAE	0.7000	0.7000
GAE + 1D CNN	0.7544	0.8093
VGAE	0.8193	0.8484
VGAE + 1D CNN	0.8343	0.8836

The results of all the described models are listed in Table 1. To evaluate the performance of these models, the study uses Area Under the Curve (AUC) and precision. It is evident that 1D-CNN has enhanced the performance of GCN-based graph autoencoders. Moreover, VGAE performs better than GAE because of its probabilistic nature.

In summary, this study has explored link prediction in GRN by using graph neural network methods. We demonstrate that VGAE performs better than GAE and including a 1D-CNN feature learning module leads to an increased performance of GCN-based GAE and VGAE. Future studies might explore other feature learning methods and how those methods impact on link prediction of GNN models.

REFERENCES

[1] G. Muzio, L. O'Bray, and K. Borgwardt, "Biological network analysis with deep learning," *Briefings in Bioinformatics*, vol. 22, no. 2, pp. 1515–1530, 2021.

[2] J. Wang, A. Ma, Q. Ma, D. Xu, and T. Joshi, "Inductive inference of gene regulatory network using supervised and semi-supervised graph neural networks," *Computational and structural biotechnology journal*, vol. 18, pp. 3335–3343, 2020.

[3] K. Mochida, S. Koda, K. Inoue, and R. Nishii, "Statistical and machine learning approaches to predict gene regulatory networks from transcriptome datasets," *Frontiers in Plant Science*, vol. 871, p.1770, 2018.

[4] N. Patel and J. T. Wang, "Semi-supervised prediction of gene regulatory networks using machine learning algorithms," *Journal of biosciences*, vol. 40, no. 4, pp. 731–740, 10 2015.

[5] B. Teji, J. K. Das, S. Roy, and D. Bhandari, "Predicting Missing Links in Gene Regulatory Networks Using Network Embeddings: A Qualitative Assessment of Selective Embedding Techniques," *Lecture Notes in Networks and Systems*, vol. 431, pp. 143–154, 2022.

[6] Y. Long, M. Wu, Y. Liu, Y. Fang, C. K. Kwoh, J. Chen, J. Luo, and X. Li, "Pre-training graph neural networks for link prediction in biomedical networks," *Bioinformatics*, vol. 38, no. 8, pp. 2254–2262, 4 2022.

[7] T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders," 11 2016.

[8] D. Marbach et al., "Wisdom of crowds for robust gene network inference," *Nature Methods 2012 9:8*, vol. 9, no. 8, pp. 796–804, 7 2012.

[9] Y. Wang, B. Xu, M. Kwak, and X. Zeng, "A Simple Training Strategy for Graph Autoencoder," *ACM International Conference Proceeding Series*, pp. 341–345, 2 2020.

[10] V. Chaubey, M. S. Nair and G. N. Pillai, "Gene Expression Prediction Using a Deep 1D Convolution Neural Network," *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 1383-1389

Gene Regulatory Network Inference through Link Prediction using Graph Neural Network

S. Ganeshamoorthy[1], L. Roden[2], D. Klepl[1] and F. He[1]

1. Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry, UK

2. Centre for Sport, Exercise and Life Sciences, Coventry University, Coventry, UK

INTRODUCTION

- Gene Regulatory Networks (GRNs) depict the causal regulatory interactions between transcription factors and their target genes.
- GRN helps to identify and prioritize the candidate genes for functional analysis.
- Recently, high-dimensional transcriptome datasets are produced from high-throughput sequencing techniques, such as microarray and RNA-Seq.
- These techniques can capture the differences in the expression of thousands of genes at once.
- Through these wet-lab experiments, studying the interconnections among a large number of genes or transcription factors at a network level is challenging.
- Therefore, one of the important topics in computational biology is the inference of GRNs from high-dimensional gene expression data through statistical and machine learning approaches.

OBJECTIVE

- The Gene Regulatory Network inference problem handling through link prediction using graph convolution network (GCN) based graph autoencoder (GAE) model.
- Additionally, one-dimensional convolution neural network (1D-CNN) feature extraction module is used to enhance the performance of the GRN inference.

DATA

- E. coli in a Dialogue on Reverse Engineering Assessment and Methods (DREAM) project (DREAM5 challenge).
- E. coli gold standard network from the DREAM5 dataset consists of 4511 genes, 2066 verified interactions, and 805 samples.

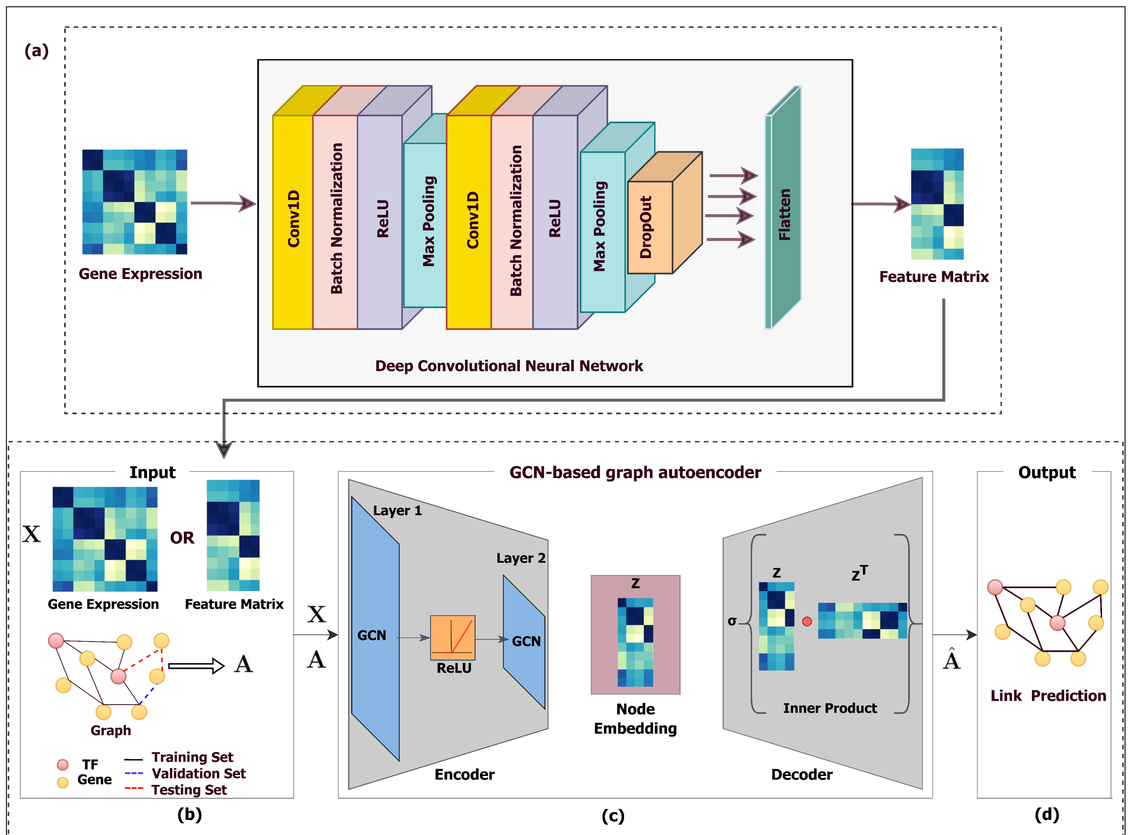

Figure 1. The overall architecture of a GCN-based graph autoencoder with 1D-CNN feature learning. The framework includes (a) a 1D-CNN-based feature learning, (b) The graph links (connections) are divided into a training set (80%), validation set (10%) and testing set (10%). Input for GCN-based graph autoencoder is adjacency matrix A and node feature matrix X where A is converted from the training set of graph links, and X is either raw Gene Expression or Feature matrix from the output of 1-D CNN (c) a GCN-based graph autoencoder (Encoder, Node Embedding, Decoder), and (d) the output is the predicted graph.

Model	AUC	Precision
GAE	0.7000	0.7000
GAE + 1D CNN	0.7544	0.8093
VGAE	0.8193	0.8484
VGAE + 1D CNN	**0.8343**	**0.8836**

Table 1. Evaluation of proposed methods

METHODOLOGY

- A graph can be represented as **G = (V,E).** **V** represents a set of nodes, and **E** is a set of edges between the nodes.
- In Gene Regulatory Network,

  Nodes: Genes

  Edges: Relationship between genes
- The proposed GRN inference method can be implemented in two steps.

Feature Extraction using 1D-CNN layer

Input : Gene expression

Deep Convolution Neural Network : Two convolution blocks (1D Convolution layer, Batch Normalisation, ReLU activation function, max pooling)

Output : Feature Matrix

Link Prediction using GCN-based GAE

Input : Feature Matrix or Gene Expression **(X),** Adjacency Matrix **(A)**

GCN-based graph autoencoder : Encoder (GCN) , Node Embedding (Z) , Decoder

Output : Predicted GRN

- Variational graph autoencoder (VGAE) is an extension of GAE based on the variational auto-encoder.

RESULT & CONCLUSION

- To evaluate the performance of these models, the study uses Area Under the Curve (AUC) and precision.
- It is evident that 1D-CNN has enhanced the performance of GCN-based graph autoencoders. Moreover, VGAE performs better than GAE because of its probabilistic nature.
- Future studies might explore other feature learning methods and how those methods impact on link prediction of GNN models.