2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)

Graph Neural Networks for Z-DNA prediction in Genomes

Artem Voytetskiy
Bioinformatics Laboratory
HSE University
Moscow, Russia
voytetskiyartem@gmail.com

Alan Herbert
Bioinformatics Laboratory
HSE University
Moscow, Russia
InsideOutBio
Charlestown, MA, USA
alan.herbert@insideoutbio.com

Maria Poptsova
Bioinformatics Laboratory
HSE University
Moscow, Russia
mpoptsova@hse.ru

2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-6819-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIBM55620.2022.9995451

Abstract— **Deep learning methods have been successfully applied to the tasks of predicting functional genomic elements such as histone marks, transcriptions factor binding sites, non-B DNA structures, and regulatory variants. Initially convolutional neural networks (CNN) and recurrent neural networks (RNN) or hybrid CNN-RNN models appeared to be the methods of choice for genomic studies. With the advance of machine learning algorithms other deep learning architectures started to outperform CNN and RNN in various applications. Graph neural network (GNN) applications improved the prediction of drug effects, disease associations, protein-protein interactions, protein structures and their functions. The performance of GNN is yet to be fully explored in genomics. Earlier we developed DeepZ approach in which deep learning model is trained on information both from sequence and omics data. Initially this approach was implemented with CNN and RNN but is not limited to these classes of neural networks. In this study we implemented the DeepZ approach by substituting RNN with GNN. We tested three different GNN architectures – Graph Convolutional Network (GCN), Graph Attention Network (GAT) and inductive representation learning network GraphSAGE. The GNN models outperformed current state-of the art RNN model from initial DeepZ realization. Graph SAGE showed the best performance for the small training set of human Z-DNA ChIP-seq data while Graph Convolutional Network was superior for specific curaxin-induced mouse Z-DNA data that was recently reported. Our results show the potential of GNN applications for the task of predicting genomic functional elements based on DNA sequence and omics data.**

Availability and implementation—The code is freely available at https://github.com/MrARVO/GraphZ.

Keywords—Z-DNA, flipons, non-B DNA, Recurrent Neural Network, Graph Neural Network, Graph Convolutional Network Graph Attention Network, GraphSAGE

I. Introduction

Graph Neural Network approaches are applicable to any data that can be represented as a graph, as shown by previous applications of GNN that leverage many bioinformatic datasets available. The most successful implementations are for prediction of drug-target, ligand-protein or protein-protein interactions (see [1] for review).

Later graph representation learning has been extended to genomics. A graph neural representation of RNA-protein interactions based on an adjacency matrix containing base-pairing probabilities was used to predict RNA structure [2]. A Graph Convolutional Network used the k-mer co-occurrence and relationships binding to learn the underlying k-mer graph to successfully model DNA-protein interactions [3]. A Graph Attention Network (GAN) was applied to Hi-C data and gene co-expression networks to produce a graph representation of the gene contact network to predict chromatin organization in cells and reveal importance of gene contacts with other genes in the neighborhood [4].

The application of GNN to single cell RNA-seq analysis has been highly successful in learning the cell–cell relationships that drive biological responses. In this approach, the gene expression matrix from scRNA-Seq experiments is first processed with graph autoencoder network such as that proposed in [5]. The topological graph embeddings are then recovered to reconstruct the cell graph [6].

Graph neural network for protein function prediction was implemented in DeepGraphGO that used combination of protein sequence and high-order protein network data [7]. The GNN approach combining sequence 3D-genome organization with other information also allowed better prediction of epigenetic state than current methods based on graph convolutional networks which learn node representations from local sequence and long-range interactions [8]. The incorporation of patient omics data into the graph representation learning model allowed prediction of disease outcome by a multi-level attention GNN [9].

Here we pioneer the use of graph neural networks for the task of prediction Z-DNA regions, called Z-flipons [10], which have the potential to regulate many cellular processes [11] and that are involved in mendelian disease [12]. We tested performances of several graph neural network architectures - Graph Convolutional Network (GCN) [13, 14], Graph Attention Network (GAT) [15, 16] and GraphSAGE [17]. We evaluated the representation of omics data by different aggregation functions implemented in different GNN architectures and discuss the advantages of each for different classes of data and the most appropriated benchmarks to use.

II. Methods

A. Data

We took the same data as it was used in the original paper of DeepZ [18]: ChIP-seq data from Shin et al. for human genome [19] that identified 385 regions of Z-DNA. After filtering for black-listed regions, we interrogated a dataset of 1054 omics features to find those predictive of other Z-

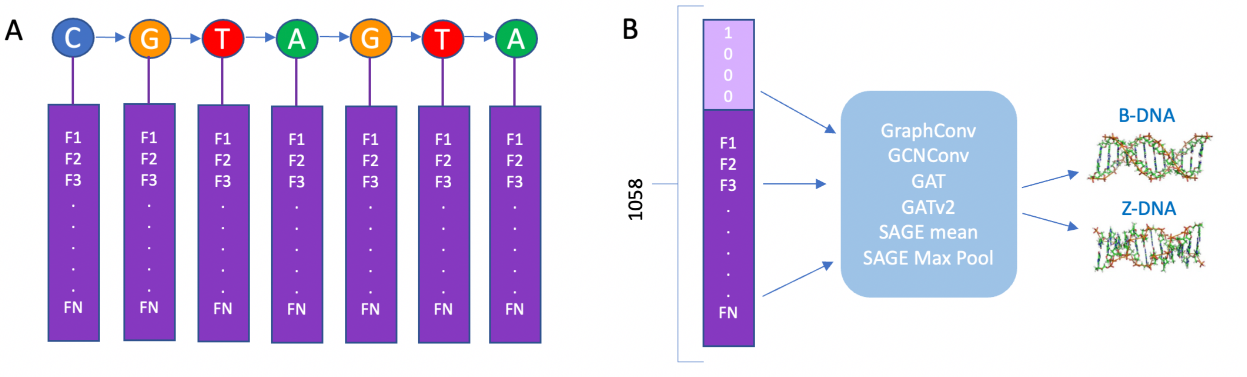


Fig. 1 DeepZ with GNN model. (A) DNA sequence with omics features represented as vectors. (B) Schematics of DeepZ approach with different GNN architectures.

DNA forming regions in the genome (see full list in [18]). For the mouse genome we took the same data as =was used for DeepZ in [20]. We also used ChIP-seq data prepared with an anti-Z-DNA antibody from mouse embryonic fibroblasts that were treated with curaxin to induce Z-DNA. In total, 1957 regions and 874 omics features were analyzed (see full list in [20]). Schema of DNA sequence and omics data representation is presented in Figure 1.

B. *Training*

The whole genome was split into 5000 bp segments, and each segment was classified according to presence/absence of Z-DNA. We choose a fixed interval to avoid centering on the Z-DNA in the region so that Z-DNA regions are randomly distributed over 5000 bp intervals. The approach prevents the model from just learning boundary features that could result in target leakage. To further avoid overfitting, we took a region at least one order of magnitude larger than the real Z-DNA region, considering ChIP-seq signals, which could be several hundreds of nucleotides large.

The negative class was composed from randomly selected segments without Z-DNA. We tested several ratios of negative to positive classes. Training and test sets were split in the proportion of 4:1 with the same ratio of class balance. The training process took 25 epochs, and the hyperparameters were selected according to f1-score. With this approach we select the model with the optimal values of precision and recall, which does not suffer from overfitting.

We used 3- and 4-layers models as designated in Tables 1-4 because the performance of these models was better than of other configurations. We also trained models on data sets with different ratio of positive and negative classes. For example, model termed as 5N was trained on a data set that had 5 times more elements of negative class than the positive.

C. *Graph neural networks*

Schematics of DeepZ approach with different GNN architectures is presented in Figure 1B. All models were implemented with torch geometric that include realization of different types of layers of GNN with RMSprop optimizer and NLLLoss loss function.

Graph Convolutional Network was implemented as GCNConv [13] and GraphConv [14].

Graph Attention Network was implemented as GATConv [16] and GATv2Conv [15]. GAT architecture uses hidden layers with attention mechanisms. These layers assign to neighbors of a node individual weights that are taking into consideration during aggregation. The difference between two sub-types is that GATConv uses static attention mechanism and GATv2Conv – dynamic.

GraphSAGE was implemented as SAGEConv [17] with two different aggregation schemes: mean and max pooling. In GraphSAGE architecture instead of separate vectors for every node the training is performed for aggregation functions. Each function can gather information of different distances from the current node. This approach helps to predict nodes that represent Z-DNA forming regions not seen during the training.

D. *Benchmark model*

DeepZ model was run as described in [18] for human genome data set [19] and as described in [20] for mouse genome data set [20]. Human data set comprised 1054 omics features (see full list in [18]) and mouse data set comprised 874 omics features (see full list in [20]).

III. Results and Discussion

The results of the models' performance on the test sets and for whole-genome prediction for human and mouse genome are presented in Tables 1-4. L signifies number of layers and N stands for the ratio between positive and negative classes.

When working with models to predict non-B DNA structures, one should understand the specificity of the experimental data used for training. Flipons (Z-DNA, qadruplexes, triplexes) are dynamical structures that are assembled in order to launch different genetic programs and disassembled after they performed their function [10]. Any genome-wide experiments detect only fraction of all possible flipons. A machine learning model trained on an experimental data set will overcome this problem by recognizing a number of additional real non-B DNA structures with similar features but that were not captured by a particular experiment. The model's precision therefore will be low as many of the predictions lack experimental validation. This is more noticeable on whole-genome predictions but also can be significant for the test set if that data is noisy and includes many false positive inputs, as often is the case for ChIP-Seq data. In these situations recall will also be low.

TABLE I. MOUSE GENOME. PREDICTION ON THE TEST SET.

Model	Prec.	Recall	F1	ROC-AUC	PR-AUC	Nu Param
GCNConv						
3L 5N	0.578	0.374	0.454	0.868	0.459	395902
3L 7N	0.590	0.316	0.412	0.827	0.414	395902
4L 5N	0.558	0.395	0.462	0.854	0.447	456102
GraphConv						
3L 5N	**0.578**	**0.404**	**0.475**	**0.873**	**0.486**	791302
3L 7N	0.550	0.419	0.475	0.865	0.475	791302
4L 5N	0.621	0.375	0.468	0.864	0.484	911502
4L 7N	0.588	0.387	0.467	0.852	0.468	911502
GAT						
3L 4N	0.649	0.341	0.448	0.882	0.502	396906
3L 5N	0.63	0.323	0.427	0.866	0.469	396906
3L 7N	0.585	0.361	0.447	0.87	0.453	396906
4L 5N	0.616	0.357	0.452	0.872	0.468	457506
GAT v2						
3L 5N	0.598	0.358	0.448	0.866	0.468	792808
3L 7N	0.596	0.349	0.441	0.868	0.464	792808
4L 5N	0.567	0.388	0.461	0.856	0.446	913608
SAGE mean						
3L 5N	0.588	0.362	0.448	0.868	0.460	791302
3L 7N	0.536	0.398	0.457	0.872	0.434	791302
4L 7N	0.595	0.353	0.443	0.868	0.452	911502
SAGE max pooling						
3L 5N	0.566	0.408	0.474	0.872	0.476	791302

TABLE II. MOUSE GENOME. WHOLE-GENOME PREDICTION.

Model	Prec.	Recall	F1
RNN (DeepZ)			
	0.400	0.160	0.230
GCNConv			
3L 5N	0.224	0.387	0.284
3L 7N	0.288	0.323	0.304
4L 5N	0.229	0.421	0.297
GraphConv			
3L 5N	**0.190**	**0.441**	**0.266**
3L 7N	0.118	0.443	0.264
4L 5N	0.253	0.416	**0.315**
4L 7N	0.203	0.422	0.274
GAT			
3L 4N	0.232	0.354	0.280
3L 5N	0.288	0.342	**0.312**
3L 7N	0.210	0.370	0.273
4L 5N	0.217	0.380	0.275
GAT v2			
3L 5N	0.238	0.377	0.292
3L 7N	0.239	0.362	0.288
4L 5N	0.234	0.416	0.299
SAGE mean			
3L 5N	0.246	0.382	0.299
3L 7N	0.215	0.411	0.283
4L 7N	0.232	0.372	0.286
SAGE max pooling			
3L 5N	0.174	0.432	0.248

TABLE III. HUMAN GENOME. PREDICTION ON THE TEST SET.

Model	Prec.	Recall	F1	ROC-AUC	PR-AUC	Nu Param
GCNConv						
3L 5N	0.660	0.348	0.456	0.851	0.496	579802
4L 5N	0.621	0.407	0.491	0.868	0.472	650002
GraphConv						
3L 4N	0.708	0.393	0.509	0.878	0.560	1159002
3L 5N	0.642	0.472	0.544	0.812	0.462	1159002
4L 5N	**0.541**	**0.554**	**0.547**	**0.834**	**0.519**	1299202
GAT						
3L 3N	0.657	0.392	0.491	0.881	0.529	581006
3L 4N	0.707	0.374	0.489	0.893	0.553	581006
3L 5N	0.669	0.381	0.486	0.848	0.499	581006
4L 5N	0.650	0.429	0.517	0.88	0.533	651606
GAT v2						
3L 5N	0.668	0.380	0.484	0.831	0.517	1160808
4L 5N	0.654	0.388	0.487	0.868	0.492	1301608
SAGE mean						
3L 5N	0.699	0.383	0.495	0.833	0.548	1159002
4L 5N	0.660	0.446	0.532	0.861	0.526	1299202

TABLE IV. HUMAN GENOME. WHOLE-GENOME PREDICTION.

Model	Precision	Recall	F1
RNN (DeepZ)			
	0.110	0.270	**0.160**
GCNConv			
3L 5N	0.068	0.547	0.121
4L 5N	0.037	0.603	0.070
GraphConv			
3L 4N	0.059	0.660	0.109
3L 5N	0.030	0.757	0.060
4L 5N	**0.016**	**0.832**	**0.032**
GAT			
3L 3N	0.054	0.600	0.099
3L 4N	0.067	0.604	0.120
3L 5N	0.062	0.626	0.112
4L 5N	0.042	0.706	0.079
GAT v2			
3L 5N	0.060	0.593	0.109
4L 5N	0.053	0.586	0.097
SAGE mean			
3L 5N	0.069	0.622	**0.124**
4L 5N	0.043	0.731	0.082

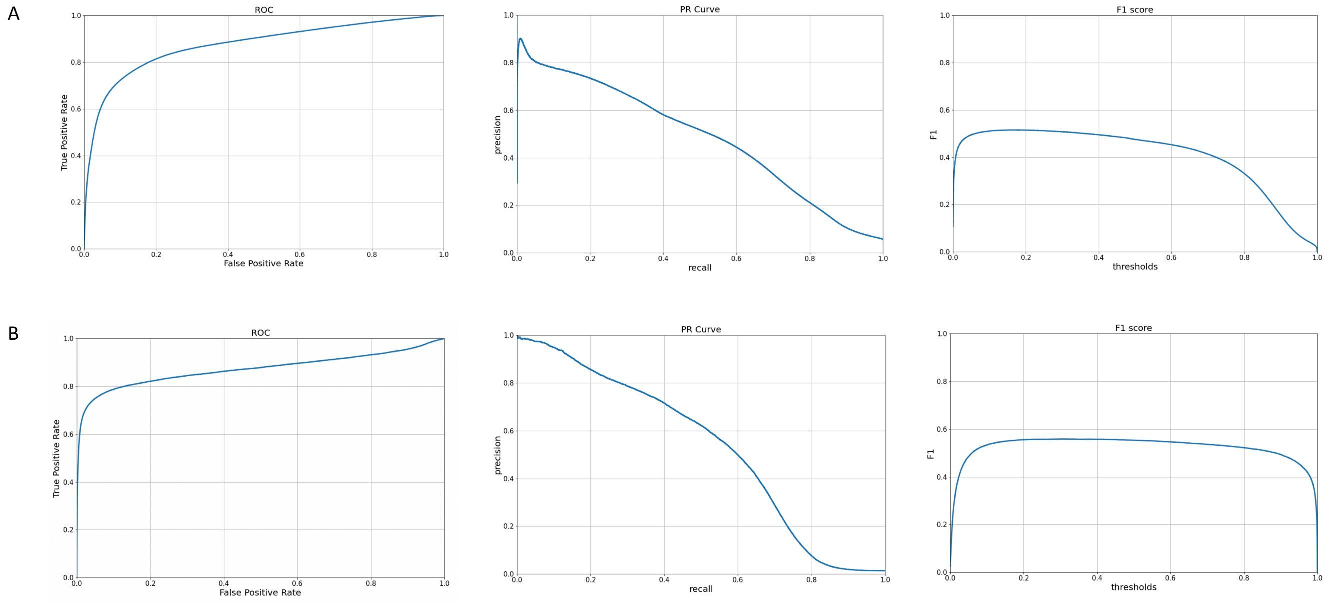


Fig. 2. ROC, PR and F1 Curves for the best GNN models. A. Mouse genome: GraphConv 3L 5N. B.Human Genome: GraphConv 4L 5N.

Having these caveats in mind, when choosing the best model, we consider all model's performance metrics cumulatively paying most attention to the F1 following by recall, PR AUC, and ROC AUC.

The best predictions on the test sets are Graph Convolutional Networks (GraphConv) that have highest F1 followed by high PR AUC, ROC AUC in both human and mouse genomes. The ROC, PR and F1 Curves for the best GNN models are given in Figure 2.

When we look for the performance of the models selected at the test sets for whole-genome predictions, Graph Convolutional Networks show the highest performance for recall both for the mouse genome with the recall 2.75 higher than RNN (an increase from 0.16 to 0.441) and for the human genome with the recall 3.08 higher (an increase from 0.270 to 0.832). Low precision, and hence low F1, on the whole-genome predictions are due to considerations described above – many novel predictions are present at the genome-wide level that were not experimentally detected in the cell line tested.

When considering F1 metrics, i.e. both precision and recall, the Graph SAGE (F1=0.124) architecture outperforms the Graph Convolutional (F1=0.121) on the human data set.

Graph Attention Networks have almost the same performance as Graph SAGE (F1=0.120 vs F1=0.124) for the human genome and as Graph Convolutional does for the mouse genome (F1=0.315 vs F1=0.312).

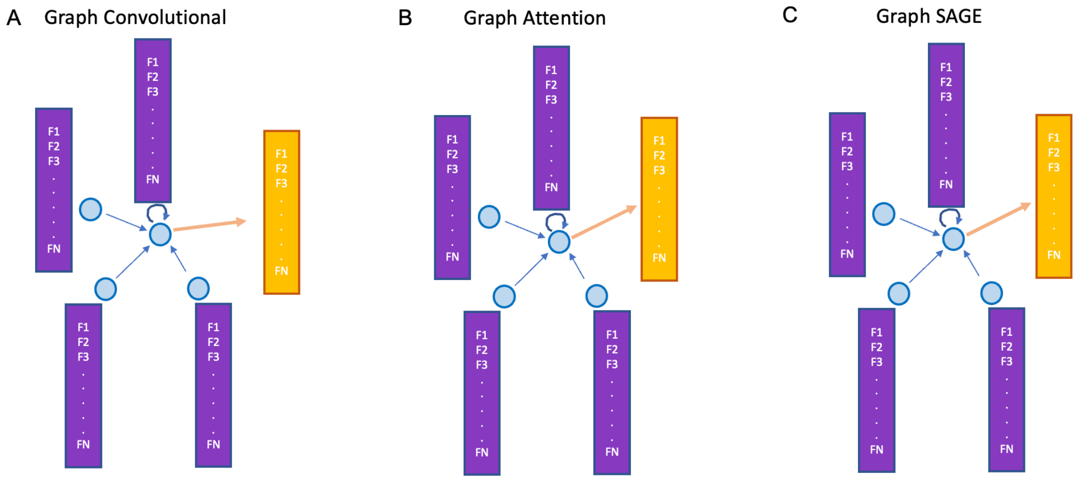
Overall, when comparing original RNN implementation with GNN implementation, we see the recall increases by a factor of ~3. Overall, the low F-metric at the whole-genome level is explained by the incomplete experimental data due to dynamic nature of Z-DNA formation with only a subset of all possible functional Z-flipons captured from the cell type used in the experiment. However, in individual experiments, DeepZ and various GNN models predict true functional Z-DNA regions that are present in the training set, yielding considerably higher recall metrics for both human and mouse genomes (Tables 1-4).

The higher precision of the original RNN implementation compared to any GNN models for human data sets results in a higher F1, but this approach may also decrease the recall as controlling the false positive rate can also increase the frequency of false negatives. As a result, the fit with experimental data is diminished. It is of more use to select the model that is better at predicting the experimental data, i.e. model with the highest recall. The model's validity can be further confirmed with additional experimental datasets that sample different parts of the genome. If the model is sound, the new data with lead to higher precision and F1 scores.

The increase in performance of GNNs over RNNs and CNNs can be explained by the ability of GNN models to take into account omics features of the neighboring nodes that are encoded as node weights – making use of information that the other approaches cannot. In our case, the features selected are from experimental genome-wide data sets for histone marks, transcription factors, chromatin accessibility and even biophysical energy of dinucleotide flipping from B- to Z-conformations.

In our models DNA sequence is represented as a graph with each nucleotide placed at a node and then assigned node-specific omics data features (Figure 1). The advantage of graph neural network over convolution neural network is that the model learns features by aggregating features from neighboring nodes, each with many experimentally derived attributes. This approach allows extraction of relevant features from the many high dimensional datasets available.

The three GNN architectures differ from each other by the way they learn features of neighboring nodes (Figure 3). Graph Convolutional Network is a convolutional neural network on a graph in which, instead of applying a convolution kernel, the algorithm first aggregates features from neighboring and current nodes, and then encodes them depending on the result of the aggregation. As an aggregation function, GCN uses per element average and then makes a weighted sum. The distinctive property of Graph Attention Network is that feature aggregation is realized using an



$$F_i = \sigma \sum_{i=1}^{N_j} c_{ij} W F_j$$

$$F_i = \sigma \sum_{i=1}^{N_j} a_{ij} W F_j$$

$$a_{ij} = \frac{exp\left(attention(F_i, F_j)\right)}{\sum_{k=1}^{N_i} exp\left(attention(F_i, F_k)\right)}$$

Aggregation Function F_i, which is based on the mean or max pooling, is trained together with the model.

Fig. 3. Aggregations functions for different types of GNN. A. Graph Convolutional Network B. Graph Attention Network. C. GraphSage.

attention mechanism. In the original publication, the attention mechanism is implemented as an additional layer of neural network that allows aggregation of data for each node using weights, thus signifying an importance of each node for its neighbors. This approach takes into account properties of neighboring nodes and improve overall results. GraphSAGE is an inductive representation learning network with an aggregation function that can use different models: average, pooling, or LSTM and an update function with concatenation. The advantage of GraphSAGE is its ability to learn from small amount of data.

When comparing a number of parameters needed for the different architectures of CNN, RNN, and GNN (Tables 3 and Figure 2 in [18]), the best Graph Convolutional models requires an order of magnitude less (around 1,3 million) versus 13 million for the best RNN and 10 million for the best CNN architectures.

IV. Conclusion

Here for the first time we applied Graph Neural Networks to the task of predicting non-B DNA structures. We tested DeepZ approach with GNN deep learning model instead of RNN. We built and tested three major types of graph neural network modes – two types of Graph Convolutional Networks, two types of Graph Attention Networks and inductive representation learning network GraphSAGE. All three networks outperformed RNN from the initial DeepZ implementation by increasing recall in 2.3-2.6 times for whole-genome prediction. Graph Convolutional Network showed overall higher performance when compared to Graph Attention Networks and GraphSAGE, while GraphSAGE appeared better for small data sets. The presented study demonstrated potential of replacing CNN or RNN with GNN in genomic applications especially for those that rely on high dimensional omics data for training. We explored the strength and weakness of three types of GNN architectures. The advantage of GNN is that it aggregate features from neighboring nodes. There is potential for further improvement of GNN architecture by incorporating long-range interactions between DNA nodes into the graph representation, by using different weighing schemes that capture the correlation between features of adjacent nodes and the use of L1 metrics to reduce the model sizes. Our approach is not restricted to non-B DNA structure predictions but can be extended to any functional genomic elements of interest.

Acknowledgment

We thank Nazar Bekanazarov for providing DeepZ training data for mouse and human genomes. The publication was supported by the grant for research centers in the field of AI provided by the Analytical Center for the Government of the Russian Federation (ACRF) in accordance with the agreement on the provision of subsidies (identifier of the agreement 000000D730321P5Q0002) and the agreement with HSE University No. 70-2021-00139.

References

[1] X. M. Zhang, L. Liang, L. Liu, and M. J. Tang, “Graph Neural Networks and Their Current Applications in Bioinformatics,” *Front Genet,* vol. 12, pp. 690049, 2021.

[2] Z. Yan, W. L. Hamilton, and M. Blanchette, “Graph neural representational learning of RNA secondary structures for predicting RNA-protein interactions,” *Bioinformatics,* vol. 36, no. Supplement_1, pp. i276-i284, 2020.

[3] Y. Guo, X. Luo, L. Chen, and M. Deng, "Dna-gcn: Graph convolutional networks for predicting dna-protein binding." pp. 458-466.

[4] [K. Zhang, C. Wang, L. Sun, and J. Zheng, “Prediction of Gene Co-expression from Chromatin Contacts with Graph Attention Network,” *Bioinformatics*, 2022.

[5] [M. Ciortan, and M. Defrance, “GNN-based embedding for clustering scRNA-seq data,” *Bioinformatics,* vol. 38, no. 4, pp. 1037-1044, 2022.

[6] [J. Wang, A. Ma, Y. Chang, J. Gong, Y. Jiang, R. Qi, C. Wang, H. Fu, Q. Ma, and D. Xu, “scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses,” *Nature communications,* vol. 12, no. 1, pp. 1-11, 2021.

[7] R. You, S. Yao, H. Mamitsuka, and S. Zhu, “DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction,” *Bioinformatics,* vol. 37, no. Supplement_1, pp. i262-i271, 2021.

[8] J. Lanchantin, and Y. Qi, “Graph convolutional networks for epigenetic state prediction using both sequence and 3D genome data,” *Bioinformatics,* vol. 36, no. Suppl_2, pp. i659-i667, Dec 30, 2020.

[9] X. Xing, F. Yang, H. Li, J. Zhang, Y. Zhao, M. Gao, J. Huang, and J. Yao, “Multi-level attention graph neural network based on co-expression gene modules for disease diagnosis and prognosis,” Bioinformatics, vol. 38, no. 8, pp. 2178-2186, 2022.

[10] A. Herbert, “ALU non-B-DNA conformations, flipons, binary codes and evolution,” *R Soc Open Sci,* vol. 7, no. 6, pp. 200222, Jun, 2020.

[11] A. Herbert, “A genetic instruction code based on DNA conformation,” *Trends in Genetics,* vol. 35, no. 12, pp. 887-890, 2019.

[12] A. Herbert, “Z-DNA and Z-RNA in human disease,” *Commun Biol,* vol. 2, pp. 7, 2019.

[13] T. N. Kipf, and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[14] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks." pp. 4602-4609.

[15] S. Brody, U. Alon, and E. Yahav, “How attentive are graph attention networks?,” *arXiv preprint arXiv:2105.14491*, 2021.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.

[17] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems,* vol. 30, 2017.

[18] N. Beknazarov, S. Jin, and M. Poptsova, “Deep learning approach for predicting functional Z-DNA regions using omics data,” *Sci Rep,* vol. 10, no. 1, pp. 19134, Nov 5, 2020.

[19] S.-I. Shin, S. Ham, J. Park, S. H. Seo, C. H. Lim, H. Jeon, J. Huh, and T.-Y. Roh, “Z-DNA-forming sites identified by ChIP-Seq are associated with actively transcribed regions in the human genome,” *DNA Research,* vol. 23, no. 5, pp. 477-486, 2016.

[20] T. Zhang, C. Yin, A. Fedorov, L. Qiao, H. Bao, N. Beknazarov, S. Wang, A. Gautam, R. M. Williams, and J. C. Crawford, “ADAR1 masks the cancer immunotherapeutic promise of ZBP1-driven necroptosis,” *Nature*, pp. 1-9, 2022.