Multi-omics data integration by generative adversarial network

Ahmed K T, et al.
Bioinformatics, 2022, 38(1): 179-186.

Presenter: Zhe LIU
Bioinformatics and Biostatistics Lab
March 28, 2024

Outline

1 Background

2 Methods

3 Experiments and Results

4 Conclusion

Background

Limitation of the existing methods

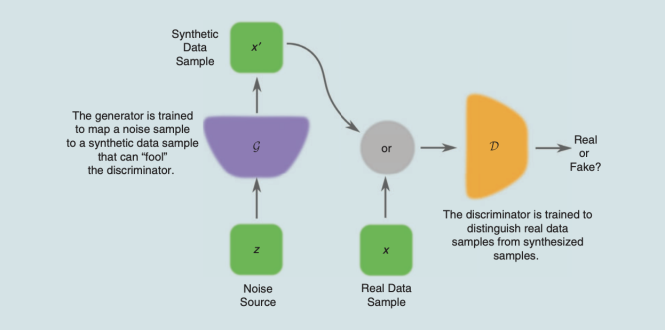
- Phenotypes depend on molecular profiles and interaction network across genomic, epigenomic, transcriptomic, proteomic and metabolomic levels.

- Integrating the interaction network into multi-omics data analysis will capture the regulatory effect and establish a better correlation with the phenotype.

- Most of the existing multi-omics integration methods did not consider the relations across different biological layers, so the power of high-throughput technologies cannot be fully utilized.

Methods

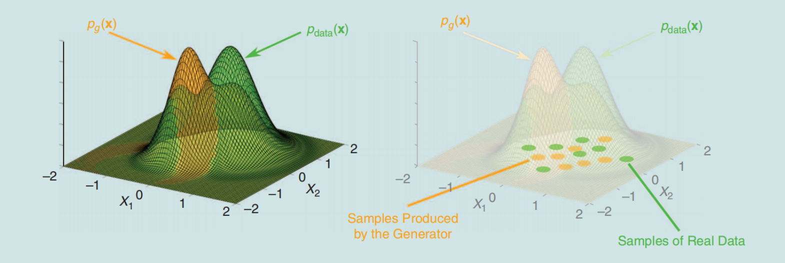
Methods: GANs model[1](composition and task)



- The generator is trained to map a noise sample to a synthetic data sample that can "fool" the discriminator.
- The discriminator is trained to distinguish real data samples from synthesized samples.

[1] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

Methods: GANs model(working principle)

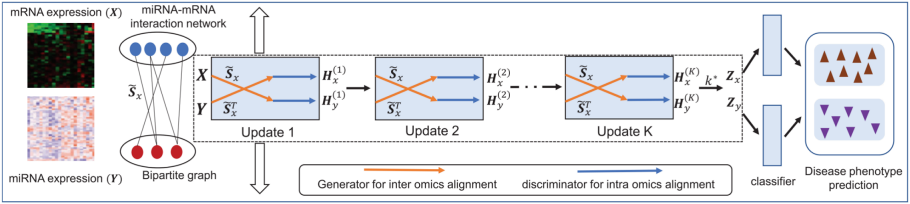


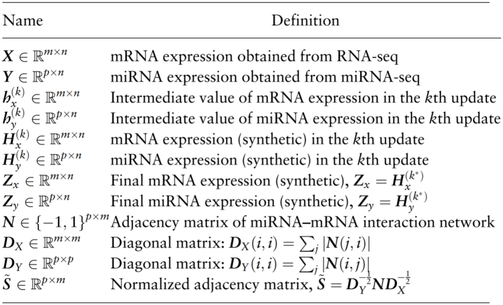
- GANs learn through computing similarity between the distribution of a candidate model $P_g(x)$ and that of real data $P_{\text{data}}(x)$.
- The generator is encouraged to produce a distribution of samples $P_g(x)$ to match that of real data $P_{\text{data}}(x)$.
- The training process involves an adversarial dynamic between the generator and the discriminator. This process continues through multiple iterations, with the two continually improving in their respective tasks.

Methods: The principle of the omicsGAN model

- The principle of the omicsGAN framework is based on Generative Adversarial Networks (GANs).
- Takes one omics data and its network interacting with another omics as the input of the generator.
- Another omics data is used as input for the discriminator.
- Generate synthetic data containing both omics data and their interaction information through adversarial training.

Methods: Notations for omicsGAN



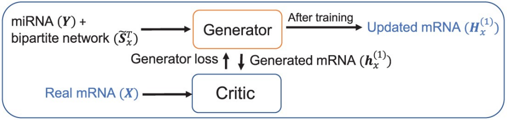


Name	Definition
$X \in \mathbb{R}^{m\times n}$	mRNA expression obtained from RNA-seq
$Y \in \mathbb{R}^{p\times n}$	miRNA expression obtained from miRNA-seq
$h_x^{(k)} \in \mathbb{R}^{m\times n}$	Intermediate value of mRNA expression in the kth update
$h_y^{(k)} \in \mathbb{R}^{p\times n}$	Intermediate value of miRNA expression in the kth update
$H_x^{(k)} \in \mathbb{R}^{m\times n}$	mRNA expression (synthetic) in the kth update
$H_y^{(k)} \in \mathbb{R}^{p\times n}$	miRNA expression (synthetic) in the kth update
$Z_x \in \mathbb{R}^{m\times n}$	Final mRNA expression (synthetic), $Z_x = H_x^{(k^*)}$
$Z_y \in \mathbb{R}^{p\times n}$	Final miRNA expression (synthetic), $Z_y = H_y^{(k^*)}$
$N \in \{-1,1\}^{p\times m}$	Adjacency matrix of miRNA–mRNA interaction network
$D_X \in \mathbb{R}^{m\times m}$	Diagonal matrix: $D_X(i,i) = \sum_j \lvert N(j,i)\rvert$
$D_Y \in \mathbb{R}^{p\times p}$	Diagonal matrix: $D_Y(i,i) = \sum_j \lvert N(i,j)\rvert$
$\tilde{S} \in \mathbb{R}^{p\times m}$	Normalized adjacency matrix, $\tilde{S} = D_Y^{-\frac{1}{2}} N D_X^{-\frac{1}{2}}$

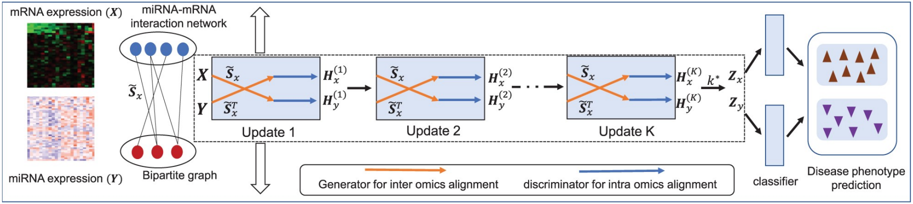
- m: the number of mRNAs(X)
- p: the number of miRNAs(Y)
- n: the number of samples
- K: the total number of updates

Methods: Overview of the framework

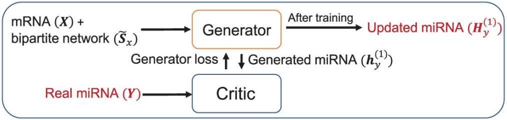
(b). generation of an updated mRNA feature set (update 1)



miRNA (Y) +
bipartite network ($\tilde{S}_x^T$)
Generator
After training
Updated mRNA ($H_x^{(1)}$)
Generator loss
Generated mRNA ($h_x^{(1)}$)
Real mRNA (X)
Critic



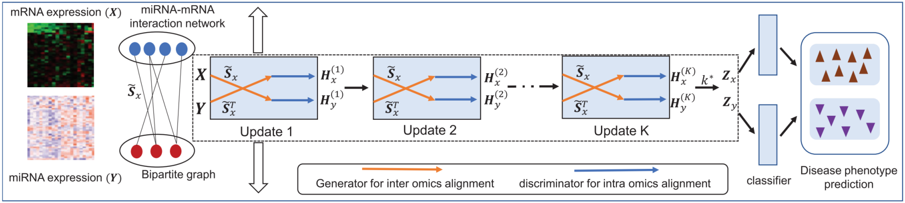
(a). Deep learning-based integration of multi-omics dataset to predict cancer phenotype



mRNA (X) +
bipartite network ($\tilde{S}_x$)
Generator
After training
Updated miRNA ($H_y^{(1)}$)
Generator loss
Generated miRNA ($h_y^{(1)}$)
Real miRNA (Y)
Critic

(c). generation of an updated miRNA feature set (update 1)

Methods: Overview of the framework



(a). Deep learning-based integration of multi-omics dataset to predict cancer phenotype

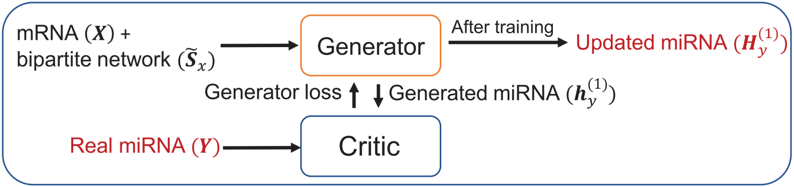
- Figure 1a represents kth update which contains two Wasserstein GANs (wGANs) for two omics data.
- Each generator generates a synthetic data and considered as the updated omics data from that box.
- A classification model is then applied on the new feature sets to predict the disease phenotype.

Methods: Why wGANs?

- GANs are well-known for their training instability, with issues such as mode collapse and vanishing or exploding gradients.
- WGANs[2] aims to improve the training process of GANs by introducing the Wasserstein distance as the optimization objective and imposing a Lipschitz constraint.

[2] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017).

Methods: Update of synthetic datasets


mRNA ($\boldsymbol{X}$) +
bipartite network ($\tilde{\boldsymbol{S}}_x$)
Generator
After training
Updated miRNA ($\boldsymbol{H}_y^{(1)}$)
Generator loss
Generated miRNA ($\boldsymbol{h}_y^{(1)}$)
Real miRNA ($\boldsymbol{Y}$)
Critic

- For each update, an intermediate value for mRNA expression is first generated from the generator using miRNA expression and normalized adjacency matrix.

$$h_x^{(k)} = G(H_y^{(k-1)}, \tilde{S}^T) \tag{1}$$

$$h_y^{(k)} = G(H_x^{(k-1)}, \tilde{S}) \tag{2}$$

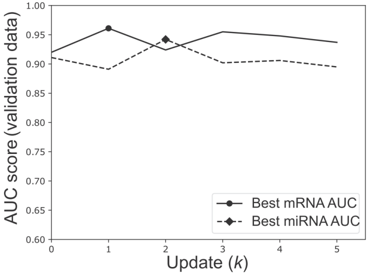
- The intermediate mRNA expression value $h_x^{(k)}$ along with the input mRNA expression value $H_x^{(k-1)}$ are then passed through a critic

$$loss_x = D_{loss}(h_x^{(k)}, H_x^{(k-1)}) \tag{3}$$

$$loss_y = D_{loss}(h_y^{(k)}, H_y^{(k-1)}) \tag{4}$$

Methods: Determination of the number of updates

- All updated datasets are sequentially fed into the SVM classifier.
- The updated datasets(k^*) with the highest AUC score for validation samples was selected as the final synthetic output.
- After the K_{th} update, get the final synthetic datasets $Z_x = H_x^{(k*)}$ and $Z_y = H_y^{(k*)}$



Methods: GANs model in this framework

- In multi-omics study, instead of random noise, introduce information from one omics data

- GANs model forces the distribution of the first omics data toward the second, ensuring the integration of information from both omics data in the generated samples.

- Fuse the interaction network in the model through the generator following the works of Kipf and Welling[3] [Appendix 3]

[3] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

Methods: Architecture and training of GANs model

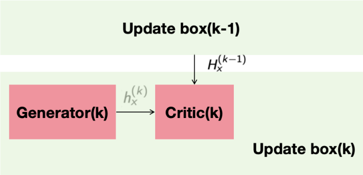
- Generators in each wGAN are three layers fully connected neural network that generates a dataset based on one omics data and the normalized adjacency matrix following the equations:

$$h_x^{(k)} = (ReLU(ReLU(\tilde{S}^T H_y^{(k-1)} W^{(0)}) W^{(1)}) W^{(2)} \quad (5)$$

$$h_y^{(k)} = (ReLU(ReLU(\tilde{S} H_x^{(k-1)} W^{(0)}) W^{(1)}) W^{(2)} \quad (6)$$

- Critic assigns values to the obtained intermediate representation $h_x^{(k)}$(small) and input dataset $H_x^{(k-1)}$(large). Objective function for training the critic is:

$$Min\ L_C = C(h_x^{(k)}) - C(H_x^{(k-1)}) \quad (7)$$



Methods: Architecture and training of GANs model

- Generator tries to produce synthetic data that will fool the critic into thinking it as real. Objective function for training the generator is:

$$Min\ L_G = -C(h_x^{(k)}) + \alpha \| h_x^{(k)} - X \|_2 \tag{8}$$

- α: coefficient to control the weight put on the two terms
- An L2-norm is added to further steer the updated dataset toward the original mRNA expression.

Methods: Evaluation methods

1. **Cancer outcome classification model:** SVM is implemented as a classifier for all experiments.
2. **Survival prediction model:** A Cox proportional hazards model with Elastic Net penalty is applied, maximizing the following log-likelihood function

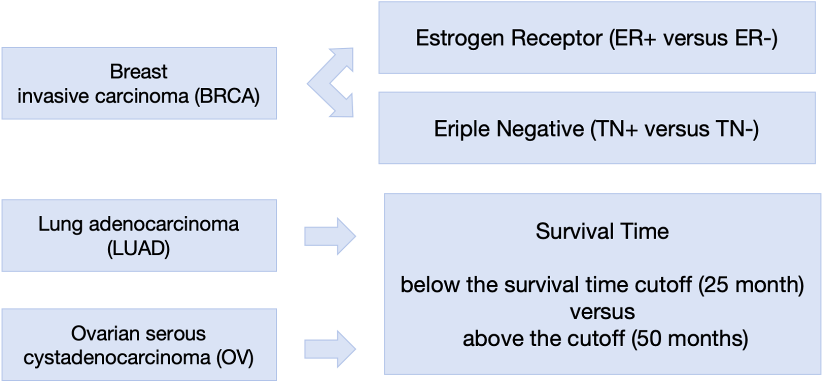
$$\log L(\beta) - \alpha \left(r \sum_{i=1}^{m} |\beta_i| + \frac{1-r}{2} \sum_{i=1}^{m} \beta_i^2 \right) \tag{9}$$

   - L(β): partial likelihood of the Cox model
   - α: hyper-parameter that controls the amount of shrinkage
   - r: relative weight of the L1-norm and L2-norm penalties
   - β_i: coefficient for the i_{th} genomic feature

Experiments and Results

Database and their usage in Experiments

- Three types of TCGA data are used for classification tasks



Breast invasive carcinoma (BRCA)

Estrogen Receptor (ER+ versus ER-)

Eriple Negative (TN+ versus TN-)

Lung adenocarcinoma (LUAD)

Ovarian serous cystadenocarcinoma (OV)

Survival Time

below the survival time cutoff (25 month)
versus
above the cutoff (50 months)

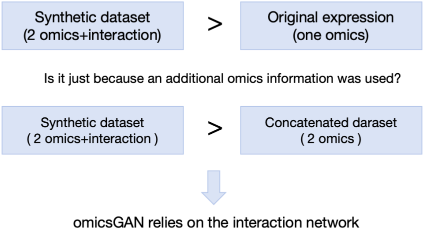
Experiments

Performed three experiments to evaluate the performance of omicsGAN and the quality of its generated synthetic data:

1. Compare outcome prediction power of the real and synthetic datasets
    - 1.1 classify clinical variables of cancer patients
    - 1.2 number of significant features identified in each dataset
2. Explore the impact of an accurate interaction network on the prediction power of synthetic datasets
3. Compare the cancer patient's overall survival prediction using real and synthetic datasets

Result 1.1: omicsGAN improved cancer outcome prediction

- Synthetic datasets achieved better average classification results than original expression for phenotype predictions across all three cancer types.
- Synthetic dataset always outperforms the concatenated dataset, indicating omicsGAN relies on the interaction network to generate synthetic data with better predictive signal.



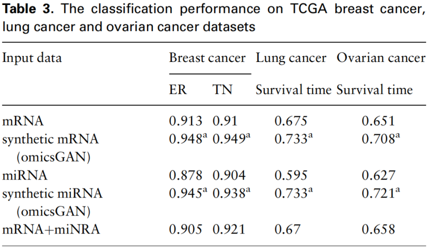


Table 3. The classification performance on TCGA breast cancer, lung cancer and ovarian cancer datasets

Input data	Breast cancer		Lung cancer	Ovarian cancer
	ER	TN	Survival time	Survival time
mRNA	0.913	0.91	0.675	0.651
synthetic mRNA (omicsGAN)	0.948[a]	0.949[a]	0.733[a]	0.708[a]
miRNA	0.878	0.904	0.595	0.627
synthetic miRNA (omicsGAN)	0.945[a]	0.938[a]	0.733[a]	0.721[a]
mRNA+miNRA	0.905	0.921	0.67	0.658

Result 1.2: omicsGAN enriches the significant features

- Performed t-test on the expression datasets with different clinical variables
- Except for ovarian cancer, all other significant features have increased compared to the original miRNA expression datasets.
- Therefore, omicsGAN enriches the features of synthetic datasets with better predictive signatures that results into improved cancer outcome prediction.

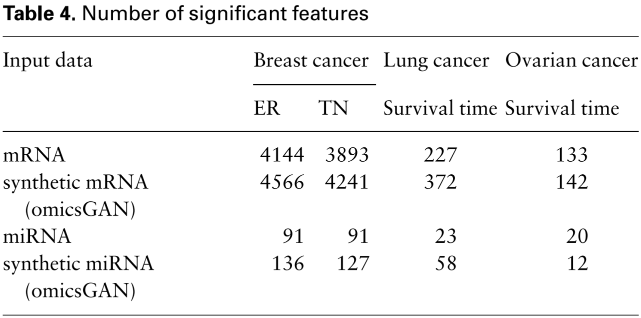
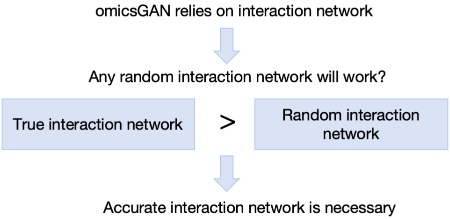


Table 4. Number of significant features

Input data	Breast cancer		Lung cancer	Ovarian cancer
	ER	TN	Survival time	Survival time
mRNA	4144	3893	227	133
synthetic mRNA (omicsGAN)	4566	4241	372	142
miRNA	91	91	23	20
synthetic miRNA (omicsGAN)	136	127	58	12

Result 2: Impact of interaction network on cancer outcome prediction



omicsGAN relies on interaction network

Any random interaction network will work?

True interaction network > Random interaction network

Accurate interaction network is necessary

- Replace the true interactioin network with 10 different randomized networks, the performances of synthetic datasets decrease

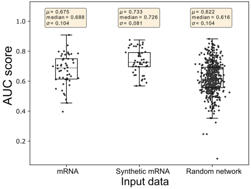


Figure: Prediction results of the survival time on lung cancer patients using mRNA expression or the synthetic one.

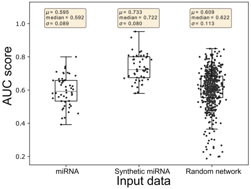


Figure: Prediction results of the survival time on lung cancer patients using miRNA expression or the synthetic one.

Result 3: omicsGAN improved survival prediction

- The Cox model evaluates the correlation between patient's overall survival and genomic features
- The patient survival predictions were improved using synthetic omics profiles compared to the original expressions.
- P-values clearly demonstrate a strong additional prognostic power of the synthetic omics profiles.

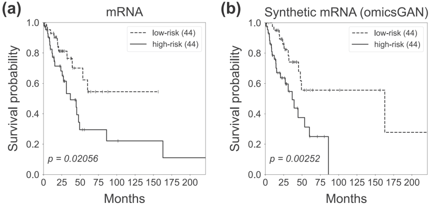


Figure: Survival prediction on lung cancer patients with mRNA profiles.

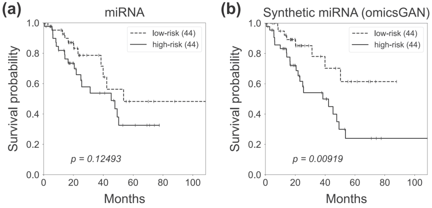


Figure: Survival prediction on lung cancer patients with miRNA profiles.

Result 4: framework can work with varying set multi-omics data

- They also designed another experiment using TF–gene interaction network to evaluate whether omicsGAN can show similar improvement in integrating other omics data and their interaction network.
- Both the synthetic TF and target gene expression performed better in classifying the lung cancer patients based on their survival time than the original TF, gene expression and concatenated TF and gene expression.

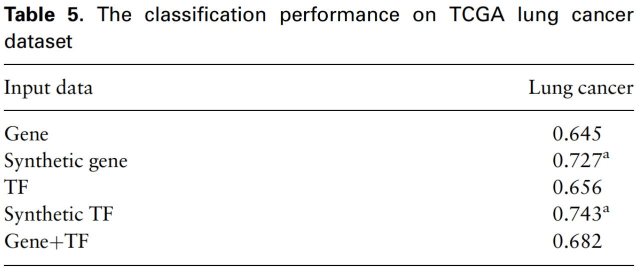


Table 5. The classification performance on TCGA lung cancer dataset

Input data	Lung cancer
Gene	0.645
Synthetic gene	0.727[a]
TF	0.656
Synthetic TF	0.743[a]
Gene+TF	0.682

Conclusion

Conclusion

1. omicsGAN not only gathers information from two omics data, but also functionally incorporate their biological interaction into the integration.
2. Synthetic data generated from omicsGAN has better discriminative power on cancer outcome classification and cancer patients survival prediction compared to the original omics datasets.

References

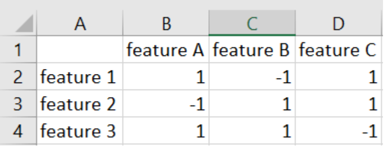
[1] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[2] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017).

[3] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

Appendix 1: Adjacency matrix of miRNA-mRNA interaction network

- miRNA–mRNA interaction network was obtained from TargetScanHuman.
- A modified adjacency matrix represented the interaction network, where each interaction was valued as -1 to imitate that miRNA negatively regulates the expression of the targeted mRNA and no interaction was valued as 1.
- Interaction netowrk should be in first omics data by second omics data format. First column should be the feature names of first omics data and first row is the feature names of second omics data.



	A	B	C	D
1		feature A	feature B	feature C
2	feature 1	1	-1	1
3	feature 2	-1	1	1
4	feature 3	1	1	-1

Appendix 2: Normalized adjacency matrix

- WHY? Normalization ensures that the influence of each node's neighbors is weighted based on their degree when aggregating their neighbor information.
- Normalized adjacency matrix: $\tilde{S} = D_X^{-\frac{1}{2}} S D_Y^{-\frac{1}{2}}$
- $D_X(i,i) = \sum_j |N(j,i)|$
  - $D_X(i,i)$ is the in-degree of node i, i.e., the total number of edges connecting to node i.
  - The in-degree of node i: sum of elements in the i-th column
- $D_Y(i,i) = \sum_j |N(i,j)|$
  - $D_Y(i,i)$ is the out-degree of node i, i.e., the total number of edges originating from node i.
  - The out-degree of node i: sum of the elements in the i-th row

Appendix 3: Fuse the interaction network

The layer-wise propagation rule in a Graph Convolutional Network (GCN)[4]:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{10}$$

- $H^{(l)}$: the activation matrix at layer l.
- $\tilde{A}$: Adjacency matrix
- $\tilde{D}$: Degree matrix of $\tilde{A}$, where $\tilde{D}_{ii}$ is the degree of node i.
- $W^{(l)}$ is the trainable weight matrix at layer l.
- $\sigma(\cdot)$ is the activation function, such as the ReLU function.

GCNs can effectively propagate information on graph-structured data, enabling each node's representation to encompass not only its own attributes but also the information from its neighboring nodes.

[4] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

Appendix 3: Fuse the interaction network

- In the interaction network, due to the directional regulation of miRNA on mRNA, this adjacency matirx is a directed graph.
- In a directed graph, it is necessary to normalize the in degree and out degree separately

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}^T\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{11}$$

- Computation of the new feature representation $H_i^{(l+1)}$ for node i by aggregating the features $H_j^{(l)}$ of all its neighboring nodes j, using the normalized adjacency matrix $\tilde{A}$ and the degree matrix $\tilde{D}$.
- This method ensures that the neighboring influence of each node is weighted based on its in and out degrees during feature propagation.

Thank you