OXFORD

Briefings in Bioinformatics, 2025, **26(3)**, bbaf273
https://doi.org/10.1093/bib/bbaf273
Problem Solving Protocol

Gene expression inference based on graph neural networks using L1000 data

Tae Hyun Kim[1,‡], Harim Kim[2,‡], Hyunjin Hwang[3,‡], Shinwhan Kang[3,‡], Kijung Shin[3], Inwha Baek[1,2,4,*,§]

[1]Department of Regulatory Science, Graduate School, Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun District, Seoul 02447, South Korea
[2]College of Pharmacy, Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun District, Seoul 02447, South Korea
[3]Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science & Technology, 85 Hoegi-ro, Dongdaemun District, Seoul 02455, South Korea
[4]Institute of Regulatory Innovation through Science (IRIS), Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun District, Seoul 02447, South Korea

[*]Corresponding author. College of Pharmacy, Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun District, Seoul 02447, South Korea. E-mail: ibaek@khu.ac.kr
[‡]Equally contributed.
[§]Inwha Baek is an assistant professor in the College of Pharmacy at Kyung Hee University. She received her Ph.D. in Biological and Biomedical Sciences in 2021 from Harvard University. Her research interests are gene expression regulation and epigenetics. Kijung Shin is an associate professor in the Kim Jaechul Graduate School of AI and the School of Electrical Engineering at Korea Advanced Institute of Science & Technology. He received his Ph.D. in Computer Science in 2019 from Carnegie Mellon University. His research interests are data mining, graph algorithms, and network science.

Abstract

Gene expression profiles can serve as proxies for cellular states and provide valuable insights into the discovery of functional connections across diverse cellular contexts. A cost-effective method called L1000 has been developed to generate gene expression profiles for over a million different conditions. Since gene expression inference of this method relies on linear regression, nonlinear regression methods, including deep learning models, have been assessed. However, these approaches process gene expression data as a vector structure, motivating us to investigate whether nonlinear models based on a graph structure are more effective in capturing the relationships between genes underlying gene expression profiles. In this work, we show that the graph neural network (GNN) model with genes as nodes outperforms both linear and nonlinear non-GNN models in predicting gene expression values and expression-based gene rankings. Importantly, our GNN model requires ∼10-fold less information than other models to achieve comparable performance. A strategic selection of input features, or incorporating an organ feature, from which the gene expression data are derived, further improves gene expression inference performance of the GNN model. Additionally, we evaluate the cross-platform generality of gene expression inference. Our study demonstrates that the transformation of RNA expression data into a graph structure effectively captures nonlinear correlations between genes, thereby enabling highly accurate and efficient prediction of gene expression profiles.

Keywords: gene expression inference; graph neural network; transcriptome

Introduction

Gene expression profiles, such as transcriptomic profiles, are indicative of cellular states. Multiple studies have demonstrated that similarities and differences in the cellular effects of pharmacological treatments, genetic perturbations, and disease states can be assessed through the gene expression profiles they induce [1–4]. The National Institutes of Health (NIH)–funded Library of Integrated Network-Based Cellular Signatures (LINCS) consortium generated a catalog of 473 647 gene expression signatures from 42 080 perturbagens [3]. This extensive catalog has served as a valuable reference for discovering functional connections between perturbagens based on their effects on gene expression.

To generate a large catalog of gene expression signatures, the LINCS project developed a high-throughput and low-cost gene expression profiling method known as L1000 [3]. This platform is a fluorescence- and hybridization-based assay that directly measures the expression of only 978 landmark transcripts. The expression values of the remaining 11 350 transcripts are inferred using linear regression (LR). This approach of inferring a full transcriptome from low-cost landmark transcript data has enabled the generation of gene expression profiles for over a million different conditions. While these inferred gene expression signatures are a valuable resource, the LR-based inference method is limited in its ability to capture nonlinear correlations within gene expression profiles [5–7]. Additionally, the LR model used in the LINCS project was calculated from separate datasets of gene expression profiles generated using microarrays [3], which further led to low prediction accuracy.

RNA-sequencing (RNA-seq) has emerged as an alternative to microarrays, with researchers having generated millions of RNA-seq datasets [8]. Inferring RNA-seq values from L1000 expression data would be more beneficial as it allows for better comparison and integration with a vast array of publicly available RNA-seq data. This motivated us to develop a gene expression inference algorithm that could potentially generate an improved catalog of

Received: January 9, 2025. **Revised:** April 11, 2025. **Accepted:** May 19, 2025

gene expression profiles in RNA-seq expression values with higher accuracy from cost-effective L1000 data.

Several attempts have been made to improve the inference model, including the gene inference model contest hosted by the NIH LINCS Consortium in 2016 [9]. Among the models submitted to the contest, the *k*-nearest neighbors (KNN) regression approach showed a notable improvement in inference accuracy [9]. Since then, several studies have applied deep learning–based methods to transcriptomic data (reviewed in [10]). One such example is the deep learning method for gene expression inference (D-GEX) which uses a multilayer feedforward neural network [11]. However, many deep learning models have yet to be extensively evaluated for the gene inference problem. Importantly, it remains unanswered which deep learning architecture is the most effective for estimating transcriptomic profiles.

Here, we explored three different nonlinear models, multilayer perceptron (MLP), Swin Transformer, and graph neural networks (GNNs), and compared their inference performance with that of the baseline LR model. The KNN model was included as an additional control. Most studies represent gene expression profiles for each cell simply as high-dimensional vectors, with each dimension corresponding to a specific gene, without explicitly modeling the relationships between genes [11–14]. We hypothesized that a graph structure with genes as nodes might be more effective than a vector structure for reflecting gene–gene correlations [15]. To test this, we examined GNN and non-GNN models, representing the graph and vector models, respectively. Recent studies have begun to apply GNNs to transcriptomic data in both classification and regression tasks [16, 17]. To the best of our knowledge, this is the first study to test an edge-attentive GNN model specialized for inferring gene expression values without incorporating any external data for graph construction. The non-GNNs tested were MLP and SwinIR. SwinIR was selected because gene expression inference can be conceptually framed as image restoration, which involves generating a high-resolution estimate of an image from a subset of data [18]. Each model was trained on paired L1000 and RNA-seq data to obtain a model for predicting gene expression profiles from the L1000 expression.

We evaluated inference performance in two components of transcriptomic profiles, the gene expression values for each gene and differential expression represented as gene ranking across various cellular contexts. The GNN model outperforms other models in inferring both RNA-seq values and differential expression-based gene rankings. Interestingly, the GNN model requires ∼10% of the input information to match the inference performance of the LR model using full input information. These results suggest that a graph data structure effectively captures gene–gene correlations, thereby inferring gene expression profiles with high accuracy and efficiency. Our GNN model offers an opportunity for cost-effective profiling of gene expression. This will ultimately benefit the clinical and pharmaceutical fields by enabling high-throughput assessment of gene expression patterns in patients and facilitating drug repurposing based on gene expression signatures. Lastly, our work serves as a framework for applying GNNs to regression tasks in biological data.

Methods

Overview of gene expression inference and data preparation

Our goal is to predict the full transcriptome in RNA-seq values utilizing the L1000 platform (Fig. 1A). We used L1000 expression values of 970 landmark transcripts as input data and RNA-seq expression values of 12 320 transcripts, including both landmark and non-landmark transcripts, as output data. These input and output data correspond to L1000 data levels 3a and 3b of the NIH LINCS project [3]. To identify a nonlinear model that effectively captures gene–gene correlations underlying gene expression, we evaluated GNN, MLP, and SwinIR models, with LR and KNN models included as controls.

We obtained L1000 and RNA-seq data from GEO accession number GSE92743, which includes the data used for the gene inference contest hosted by the NIH LINCS Consortium. The datasets include 3176 tissue samples with quantile-normalized $\log_2$-RNA expression levels measured using both L1000 and RNA-seq platforms. For all the methods, we used the first 2500 paired samples (∼80%) for training, the next 500 paired samples (∼15%) for validation, and the remaining 176 paired samples (∼5%) for testing.

Model evaluation

Assume there are N training datasets, L landmark transcripts, and T non-landmark genes: the training dataset is expressed as $\{x_i, y_i\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^L$ denotes the L1000 values of landmark transcripts and $y_i \in \mathbb{R}^{L+T}$ denotes the ground-truth RNA-seq values of both landmark and non-landmark transcripts in the ith dataset. Our goal is to find the functional mapping $F : x_i \in \mathbb{R}^L \rightarrow y_i \in \mathbb{R}^{L+T}$ that fits $\{x_i, y_i\}_{i=1}^{N}$ (Fig. 1A).

We use the sum of squared errors (SE), Spearman's correlation coefficients (SCCs), and Pearson correlation coefficient (PCC) to evaluate the inference performance for each dataset i,

$$SE_i = \sum_{g=1}^{L+T} \left(y_{i,g} - \hat{y}_{i,g}\right)^2,$$

$$SCC_i = \rho\left(y_i, \hat{y}_i\right) = 1 - \frac{6\sum_{g=1}^{L+T} {d_g}^2}{(L+T)\left((L+T)^2 - 1\right)}, \text{ and}$$

$$PCC_i = \frac{\sum_{g=1}^{L+T}\left(y_{i,g} - \bar{y}_{i,g}\right)\left(\hat{y}_{i,g} - \bar{\hat{y}}_{i,g}\right)}{\sqrt{\sum_{g=1}^{L+T}\left(y_{i,g} - \bar{y}_{i,g}\right)^2 \sum_{g=1}^{L+T}\left(\hat{y}_{i,g} - \bar{\hat{y}}_{i,g}\right)^2}}$$

where $\hat{y}_{i,g}$ is the inferred RNA-seq value for gene g in the ith dataset and d_g is the difference between the two ranks of the inferred and the measured expression values of gene g. When evaluating the LR model used in the L1000 project (LINCS LR), the ComBat batch correction was performed to adjust for cross-platform differences [19].

The overall error is defined as the average of the root of SE over N' test datasets. The overall SCC and PCC are defined as the average SCC or PCC over N' test datasets, respectively, as follows:

$$\textit{Overall error} = \frac{\sum_{i=1}^{N'} \sqrt{SE_i}}{N'},$$

$$\textit{Overall SCC} = \frac{\sum_{i=1}^{N'} SCC_i}{N'}, \text{ and}$$

$$\textit{Overall PCC} = \frac{\sum_{i=1}^{N'} PCC_i}{N'}.$$

These metrics were used to evaluate the predictive performance of different models. We adapted the gene-level recall (R_{gene}) metric, as defined in the LINCS project to select well-inferred genes [3]. A minor modification was including landmark transcripts in the inferred expression vectors when generating a null distribution of SCCs. The SCC null distribution was generated by

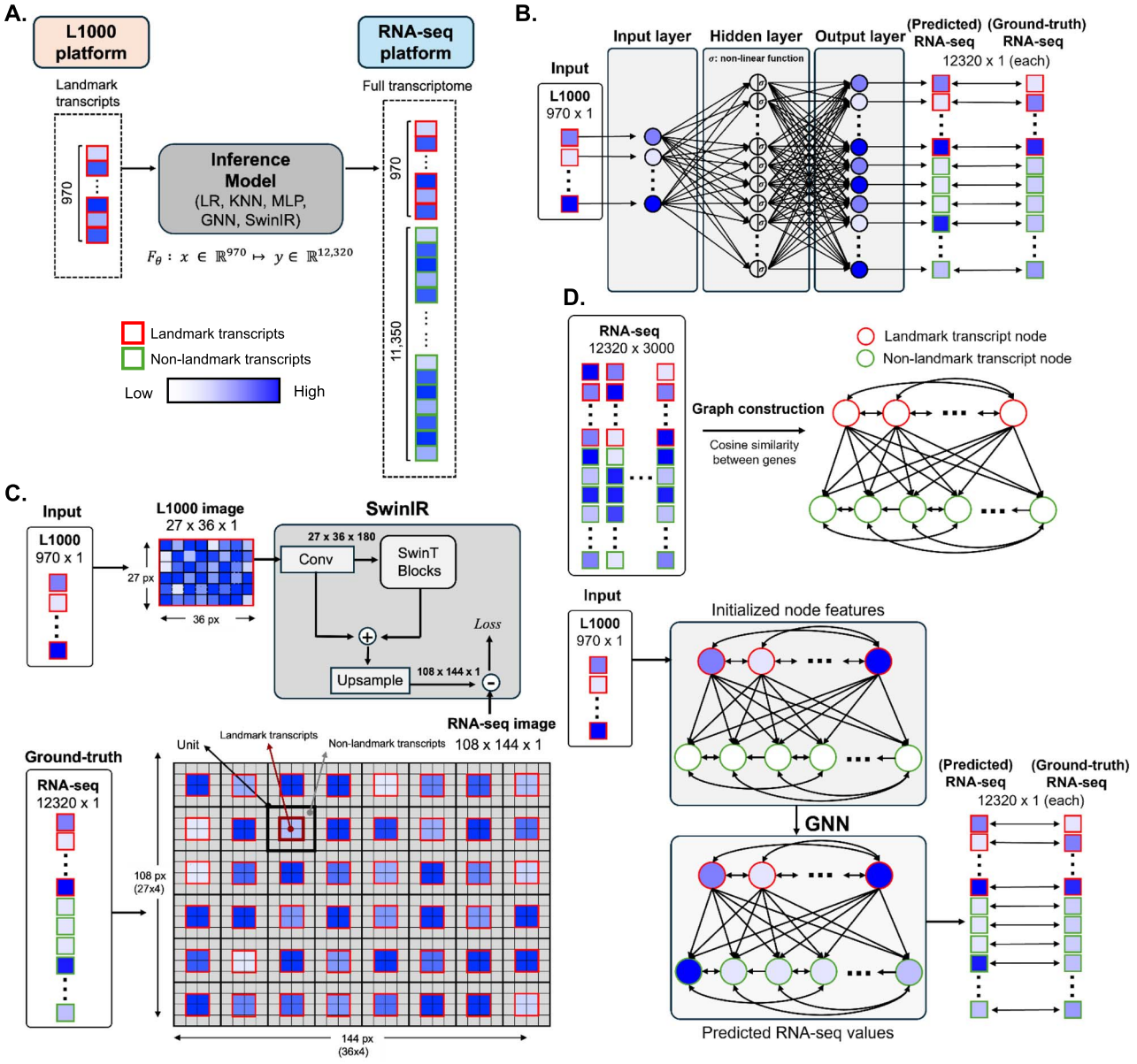

Figure 1. Gene expression inference structure and model schematics. (A) Inferring RNA-seq values of 12 320 transcripts from L1000 expression values of 970 landmark transcripts. (B) Three-layered multilayer perceptron with ReLU activation. (C) SwinIR. Both input and output gene expression vectors are reshaped into two-dimensional images for SwinIR training. (D) Graph consisting of landmark and non-landmark transcript nodes. Each landmark transcript node forms outgoing edges to all other nodes. Edges between non-landmark transcript nodes are determined based on cosine similarity between genes (Boxes and circles indicates individual transcripts and nodes, respectively, with distinct outlines used to differentiate landmark transcripts from non-landmark transcripts; expression values are shown using a graded scale to represent relative intensity).

comparing each gene in the ground-truth RNA-seq datasets with every non-matched gene in the inferred datasets,

$$\rho_{null} = \left\{ \rho\left(r_g, \hat{r}_{g'}\right) \middle| \forall g \in L+T; \forall g' \in L+T; g \neq g' \right\},$$

where r_g denotes an N'-dimensional vector comprising the RNA-seq values of gene g in N' test datasets, namely, $r_g = \left(y_{1,g}, y_{2,g}, \ldots, y_{N',g}\right)$. Similarly, $\hat{r}_{g'}$ denotes a N'-dimensional vector of the inferred expression values of gene g'; $\hat{r}_{g'} = \left(\hat{y}_{1,g'}, \hat{y}_{2,g'}, \ldots, \hat{y}_{N',g'}\right)$. If the self-correlation coefficient between the RNA-seq measured and inferred expression values of gene g, $\rho\left(r_g, \hat{r}_g\right)$ is greater than the 95th percentile of ρ_{null}, then gene g is considered to be a well-inferred gene. These well-inferred genes and landmark transcripts are subjected to a gene ranking correlation analysis that evaluates the inference accuracy regarding expression level-based gene ranking. Additionally, a gene ranking correlation coefficient is calculated for all inferred genes (12 320 genes) as an additional evaluation metric. For these analyses, we generated normalized expression profiles for each gene across the test datasets using a robust z-scoring procedure as used in the LINCS project:

$$z_{i,g} = \frac{y_{i,g} - median\left(r_g\right)}{MAD\left(r_g\right) * 1.4826}$$

where $z_{i,g}$ is the robust z-score of gene g in ith test dataset, MAD is the median absolute deviation of the vector r_g, and the factor of 1.4826 is a scale factor that allows MAD to be used as a consistent estimator of scale for normally distributed data.

This results in a vector of robust z-scores for all genes in each profile. To compare models, we calculated the SCCs of the robust z-score vectors between the ground-truth RNA-seq values and the inferred expression values for each test dataset. The SCCs are then averaged across all test datasets, and we define the averaged SCC as a gene ranking correlation coefficient.

Linear regression

The LINCS program performed gene expression inference using the LR model trained on the microarray datasets [3]. To acquire the LR model that takes L1000 inputs and predicts RNA-seq values, we trained the LR model on the paired datasets generated from both the L1000 and RNA-seq platforms. Specifically, we used LinearRegression model from scikit-learn library [20] and 2500 pairs of L1000 and RNA-seq values to find a weight matrix W that minimizes the residual sum of squares between the observed and the inferred values.

k-Nearest neighbor regression

The k-nearest neighbor (KNN) regression is a nonparametric method employed for predicting continuous target values based on the proximity of the input data [21]. The input vector is a 970-dimensional vector of the L1000 values of the 970 landmark transcripts. The corresponding target vector is the RNA-seq values of all transcripts. The method begins by computing the distances between a given query L1000 vector and other L1000 vectors in the training datasets using an appropriate distance metric. Based on the distances, it identifies the k L1000 vectors (neighbors) in the training datasets that are nearest to the query vector. Subsequently, the RNA-seq expression value is inferred by computing a weighted average of the target vectors of these nearest samples, with weights determined by the inverse of the distances between the input vector and k-nearest vectors. We used KNeighborsRegressor model from scikit-learn library [20]. We tested values of k at 3, 5, 7, and 9 and set k to 5 that yielded the best gene expression performance. The Manhattan distance (L1) metric was used to find the nearest neighbors and assign weights among them.

Multilayer perceptron

We used an MLP consisting of a single hidden layer with the rectified linear units (ReLU) activation. The input dimension is 970 containing L1000 values, and each of the following layers has a dimension of 12 320 (Fig. 1B). The mean absolute error (MAE) loss function, the Adam optimizer [22], and early stopping based on the validation SCC were adopted. The learning rate was selected from the set {1e-3, 5e-4, 1e-4}, and the weight decay coefficient was set to 1e-6. The hyperparameters were chosen based on the validation correlation.

SwinIR

SwinIR is an image restoration neural network based on the Transformer architecture, utilizing sliding windows [18]. It consists of three layers: shallow feature extraction, deep feature extraction, and image reconstruction. The shallow feature extraction uses a convolutional neural network (CNN), while the deep feature extraction utilizes the Swin Transformer. The default structure of the decoder is a CNN. To apply SwinIR to the gene inference problem, the gene expression values were reshaped into an image format (Fig. 1C). First, each expression value was normalized by dividing it by the maximum value to fit within the [0, 1] range. The normalized expression values from the 970 landmark transcripts were arranged into a 27 × 36 matrix. Similarly, the 12 320 RNA-seq values were arranged into a 108 × 144 matrix, with each dimension being four times greater than that of the L1000 matrix (27 × 36). This configuration results in the RNA-seq matrix containing a total of 972 units, with each unit consisting of 4 × 4 pixels. The 2 × 2 pixels in the center of each unit were filled with RNA-seq values of the landmark transcripts. The spatial distribution of landmark transcripts in the matrices was consistent between the L1000 and RNA-seq data. Specifically, if the landmark transcript g is assigned to (i, j) pixel of the L1000 matrix, the corresponding landmark transcript is assigned to the center of the (i, j) unit of the RNA-seq matrix, covering $(4i-2, 4j-2)$, $(4i-1, 4j-2)$, $(4i-2, 4j-1)$, and $(4i-1, 4j-1)$. The remaining 12 pixels in each unit were randomly filled with non-landmark transcripts. Any empty pixels in the L1000 and RNA-seq matrices were filled with 0. For a given L1000 matrix input ($27 \times 36 \times 1$, $H \times W \times Cin$ where H, W, Cin are the matrix height and width and input channel number, respectively), we use a 3 × 3 convolutional layer to extract the shallow feature $27 \times 36 \times 180$ ($H \times W \times C$ where C is the number of feature channels). We then extract the deep feature using six consecutive residual Swin Transformer blocks (RSTB) and a 3 × 3 convolutional layer. Each RSTB comprises six Swin Transformer layers with a patch size of 24 and six attention heads, alternately shifting the window with a size of 4. The training patch size and window size are set to smaller numbers than a default setting since the L1000 and RNA-seq data sizes are smaller than typical image data. Next, an MLP with a fully connected layer and a GELU activation function is used for further feature transformations. At the end of each RSTB, a 3 × 3 convolutional layer with the output channel number of 180 was applied. To upsample the RNA-seq matrix, PixelShuffle with an upscale factor of 2 is applied twice. The last 2D convolutional layer is applied to obtain a final RNA-seq matrix with dimensions $108 \times 144 \times 1$ ($H \times W \times C$) (Fig. 1C, "Upsample"). For training, we used the MAE loss function, Adam optimizer [22], and a multistep learning rate starting from 2.0e-4, which diminished by half at 250 000 iterations. The total number of trainable parameters was 11 857 789.

Graph neural network

Based on the cosine similarity between the expression values from 3000 RNA-seq datasets, we constructed a graph consisting of 12 320 nodes with two node types: landmark transcript nodes and non-landmark transcript nodes (Fig. 1D). For 11 350 non-landmark transcript nodes, each node has incoming directed edges from the k most similar (top-k) and k least similar (bottom-k) non-landmark nodes, with k set to 50. This graph structure ensures that both positively and negatively correlated nodes connect to each node. Additionally, directed edges from 970 landmark transcript nodes to all other nodes were constructed to ensure the information flow from landmark transcripts. We assigned the L1000 values of the 970 landmark transcripts as scalar features to their corresponding nodes while non-landmark transcript nodes were set to 0. This feature was concatenated with a one-hot vector of dimension 12 320 to enable the model to distinguish between each node. To infer the target values for each node (i.e. RNA-seq expression values), we developed an edge-attentive GNN model. Similar to conventional GNN models, our model iteratively aggregated information from the connected nodes, including the node itself. Specifically, the feature vector of each node was first projected into a d-dimensional embedding using a trainable projection matrix, where d represents the hidden dimension. Subsequently, in each layer, the embedding was updated as the weighted

sum of neighboring node embeddings. A key distinction of our model is that, in each layer, we associated each edge with a layer-specific trainable weight, treated as a free parameter without any constraint. That is, while most existing GNN models with attention mechanisms constrain edge weights to be non-negative (e.g. by normalizing them to the range between 0 and 1), our model allowed edge weights to take on both positive and negative values, enabling the model to capture a broader spectrum of relationships between nodes. This flexibility is especially important in our biological context, where both positive and negative correlations between genes carry meaningful signals. In our experiments, we used a four-layered GNN with a hidden dimension d of 64, followed by a fully connected regressor. For training, we used the MAE loss function, the Adam optimizer [22], and early stopping based on validation SCC. The learning rate was selected from the set {1e-3, 5e-4, 1e-4}, and the weight decay coefficient was set to 1e-6. The total number of trainable parameters was 56 284 193.

Results

Performance in inferring gene expression values

The LINCS project infers gene expression values from the L1000 expression using an LR model fitted on microarray data [3]. As RNA-seq has emerged as an alternative to microarrays and as the standard for transcriptomic profiling, we explored models capable of effectively inferring RNA-seq values from L1000 expression. To best capture nonlinear correlations present in gene expression for better gene expression inference performance, we investigated a broad range of deep learning models, focusing on evaluating whether transforming transcriptomic data into a graph is more effective than a vector structure. We selected the GNN and two non-GNN models, MLP and SwinIR (Fig. 1). We included the SwinIR architecture, as we viewed the gene expression inference problem as analogous to an image restoration problem, where low- and high-resolution images conceptually correspond to the expression values of the landmark transcripts and the full transcriptome, respectively. We considered landmark transcripts akin to regions centered on a local feature in an image, as they are the centroid genes of clusters in a reduced eigenspace where correlated genes are grouped together. Among the deep learning–based restoration methods, SwinIR was chosen for its impressive performance with fewer parameters [18]. The LR and KNN models were used as controls for comparison. Each model was trained on 2500 L1000 and RNA-seq paired samples. Overall errors, overall SCCs, and overall PCCs for each trained model are used to evaluate the inference performance. As a baseline, the LR model from the LINCS project was assessed, exhibiting an overall SCC of 0.8627 with an SD of 0.0442, similar to the previously reported performance [3]. Our baseline LR model trained on paired L1000 and RNA-seq data shows an overall error of 3.691 (SD = 0.451) and an overall SCC of 0.9702 (SD = 0.0062), significantly outperforming the LR model trained on the microarray data. The KNN model shows better performance than the LR, with an overall error of 3.310 (SD = 0.818) and an overall SCC of 0.9775 (SD = 0.0104). Interestingly, not all the nonlinear models outperform the baseline LR model. The performance of MLP was comparable to that of the LR model, whereas SwinIR performed worse than the LR model (Table 1). One possible explanation for SwinIR's poor performance is that the arrangement of gene locations in the image format is suboptimal. It might be necessary to perform extensive permutations or use prior biological knowledge to determine the best arrangement of gene locations. The GNN has the best performance, with an overall error of 2.809 with an SD of 0.535 and an overall SCC of 0.9836 with an SD of 0.0059. Overall PCCs are correlated with the overall SCCs, with the highest overall PCC of 0.9852 (SD = 0.0062) for the GNN model and the lowest overall PCC of 0.9537 (SD = 0.0137) for the SwinIR model. Altogether, the GNN model demonstrates the best performance in inferring RNA-seq values from L1000 expression (Table 1 and Fig. 2). This suggests that a graph structure with genes as nodes is more effective than a vector structure for capturing gene–gene correlations underlying gene expression profiles.

Performance in inferring differential gene expression

Gene expression profiles represent cell states involving differential expression between genes. The relative expression patterns are reflected in the gene rankings based on their expression values. We assessed the ability of the models to infer gene rankings by calculating the gene ranking correlation coefficients. Specifically, we first identified well-inferred genes through a gene-level recall analysis for each model. The number of well-inferred genes in each model correlates with the overall SCC, ranging from 10 475 genes for the SwinIR (85%) to 11 867 genes for the GNN model (96%) (Table 2). A set of well-inferred genes was examined for gene ranking correlation. The GNN model exhibits the highest gene ranking correlation coefficient of 0.8673 (SD = 0.0580), followed by the MLP model where the gene ranking correlation coefficient is 0.8197 with an SD of 0.0767. MLP exhibits a higher gene ranking correlation coefficient than KNN (Table 2), suggesting that it is more effective at predicting differential gene expression rather than absolute expression values. Consistent with the overall error and overall SCC and PCC, the SwinIR model shows the lowest gene ranking correlation coefficient of 0.7329 with an SD of 0.1002. The LINCS LR model exhibits poor performance in inferring gene rankings, with <70% of transcripts being scored as well-inferred genes and a gene ranking correlation coefficient of 0.5158 (SD = 0.1176). The gene ranking correlation coefficients for all 12 320 genes correlate with those for the well-inferred genes (Table 2). When we applied a stricter criterion for well-inferred genes—specifically, a gene-level recall greater than the 99.9th percentile of a null distribution—we were able to select 10 650 genes with a self-correlation coefficient >0.7 in the GNN model (Supplementary Table S1). We expect that these 10 650 genes could provide accurate differential gene expression across different cellular states. Overall, the GNN model reliably infers RNA-seq values from the L1000 data and effectively retains the information of expression-based gene ranking.

To further improve GNN's performance in gene expression inference, we incorporated organ information as an additional feature before regressing into RNA-seq values. This feature addition reduces the overall error from 2.809 (SD = 0.535) to 2.741 (SD = 0.514) and increases the overall SCC from 0.9836 (SD = 0.0059) to 0.9844 (SD = 0.0053). The overall PCC increases from 0.9852 (SD = 0.0062) to 0.9859 (SD = 0.0057). All these metrics show a significant improvement with the addition of the organ feature (P value <.001). It also improves gene ranking prediction by increasing both the number of well-inferred genes and the gene ranking correlation coefficient (Supplementary Table S2).

Performance with a reduced number of landmark transcripts

We investigated whether <970 landmark transcripts are sufficient for deep learning models to infer the full transcriptome. We randomly selected 1/2, 1/3, and 1/9 of the 970 landmark transcripts,

Table 1. Performance in inferring gene expression values

Model	Number of landmark transcripts	Overall error	Overall SCC	Overall PCC
LR (LINCS)	978	N/A	0.8627 ± 0.0442	0.8657 ± 0.0438
Constant model	970	16.603 ± 0.015	N/D	N/D
LR		3.691 ± 0.451	0.9702 ± 0.0062	0.9749 ± 0.0064
KNN ($k = 5$)		3.310 ± 0.818	0.9775 ± 0.0104	0.9787 ± 0.0113
MLP		3.611 ± 0.782	0.9743 ± 0.0091	0.9753 ± 0.0114
SwinIR		4.953 ± 0.712	0.9544 ± 0.0112	0.9537 ± 0.0137
GNN		**2.809 ± 0.535**	**0.9836 ± 0.0059**	**0.9852 ± 0.0062**

Note: The models were trained using paired L1000 and RNA-seq data. The models were tested using 176 test datasets. The performance of the models was analyzed using overall error, overall Spearman correlation coefficient (SCC), and overall Pearson correlation coefficient (PCC). The overall error of the LR model used in the LINCS project was not evaluated due to platform-dependent discrepancies between trained data (microarray) and ground-truth RNA-seq data. The constant model has the expression values of all 12 320 genes set to the mean value of the ground-truth RNA-seq values in each dataset. The GNN model reduces the prediction error by 97% compared to the constant model. **Bold** indicates the best performance. LINCS, Library of Integrated Network-Based Cellular Signatures; LR, linear regression; KNN, k-nearest neighbor regression; MLP, multilayer perceptron; GNN, graph neural network; N/A, not applicable; N/D, not determined

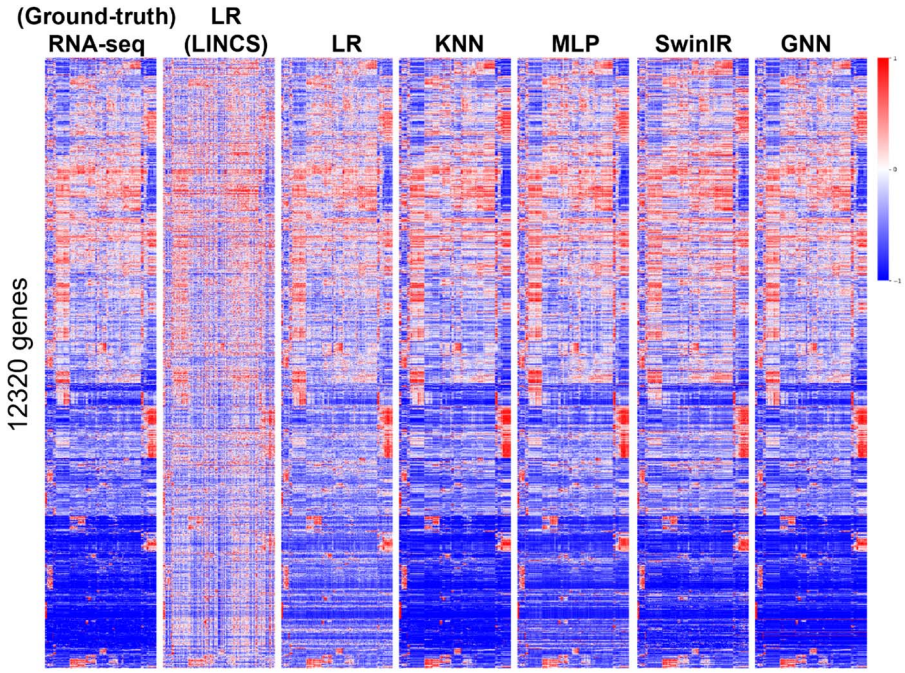

Figure 2. Heatmaps of ground-truth gene expression values from 176 test datasets (leftmost) and the inferred gene expression values from the models. Each row is an individual transcript, and each column corresponds to a separate test dataset. The order of 12 320 transcripts and 176 samples is organized through hierarchical clustering based on ground-truth RNA-seq data using seaborn [28]. The inferred values were min–max scaled for plotting (LINCS: Library of Integrated Network-Based Cellular Signatures, LR: linear regression, KNN: k-nearest neighbor regression, MLP: multilayer perceptron, GNN: graph neural network).

Table 2. Performance in inferring gene rankings

Model	Number of landmark transcripts	Number of well-inferred genes	Gene ranking correlation coefficient (well-inferred genes)	Gene ranking correlation coefficient (all 12 320 genes)
LR (LINCS)	978	7831	0.5158 ± 0.1176	0.3988 ± 0.1134
Constant model	970	616	N/D	N/D
LR		11 237	0.7959 ± 0.0731	0.7657 ± 0.0768
KNN ($k = 5$)		11 861	0.8102 ± 0.1043	0.8036 ± 0.1048
MLP		11 479	0.8197 ± 0.0767	0.8078 ± 0.0782
SwinIR		10 475	0.7329 ± 0.1002	0.6868 ± 0.1023
GNN		**11 867**	**0.8673 ± 0.0580**	**0.8612 ± 0.0586**

Note: The performance of the models was analyzed using the numbers of well-inferred genes and gene ranking correlation coefficient. If the gene-level recall value is greater than the 95th percentile of a null distribution, the corresponding gene is considered to be a well-inferred gene. The constant model has the expression values of all 12 320 genes set to the mean value of the ground-truth RNA-seq values in each dataset. **Bold** indicates the best performance. LINCS, Library of Integrated Network-Based Cellular Signatures; LR, linear regression; KNN, k-nearest neighbor regression; MLP, multilayer perceptron; GNN, graph neural network; N/D, not determined

Table 3. Performance in inferring gene expression profiles using randomly selected 108 landmark transcripts

Model	Number of landmark transcripts	Overall error	Overall SCC	Overall PCC	Number of well-inferred genes	Gene ranking correlation coefficient (well-inferred genes)	Gene ranking correlation coefficient (all 12 320 genes)
LR	108	4.733 ± 0.793	0.9557 ± 0.0128	0.9576 ± 0.0144	10 994	0.7112 ± 0.1208	0.6784 ± 0.1242
KNN ($k = 5$)		3.671 ± 1.085	0.9724 ± 0.0164	0.9730 ± 0.0179	11 749	0.7743 ± 0.1445	0.7664 ± 0.1447
MLP		4.129 ± 1.017	0.9678 ± 0.0133	0.9672 ± 0.0165	11 547	0.7806 ± 0.1075	0.7706 ± 0.1083
SwinIR		6.217 ± 0.991	0.9293 ± 0.0245	0.9261 ± 0.0271	9527	0.6790 ± 0.1438	0.5966 ± 0.1386
GNN		**3.242 ± 0.779**	**0.9790 ± 0.0096**	**0.9798 ± 0.0105**	**11 784**	**0.8310 ± 0.0912**	**0.8244 ± 0.0915**

Note: Randomly selected 108 landmark transcripts are listed in Supplementary Table 3. The GNN model reduces the prediction error by 96% compared to the constant model. **Bold** indicates the best performance. LINCS, Library of Integrated Network-Based Cellular Signatures; LR, linear regression; KNN, k-nearest neighbor regression; MLP, multilayer perceptron; GNN, graph neural network

corresponding to 486, 324, and 108 landmark transcripts, respectively. As the number of landmark transcript inputs decreases, the overall error increases and the overall SCC and PCC decrease for all models (Table 3, Supplementary Tables S3–S4). However, both the GNN and KNN models using only 10% of the 970 landmark transcripts outperform the baseline LR model, which used all 970 landmark transcripts. The overall error of the GNN model is 3.242 with an SD of 0.779, and that of the KNN model is 3.671 with an SD of 1.085. Their overall SCCs remain above 0.97. For the overall PCC, the GNN model shows 0.9798 (SD = 0.0105), which is greater than that of baseline LR model using 970 landmark transcripts. We further examined how well the GNN and KNN models, trained on 10% of landmark transcripts, preserve gene ranking information. The GNN model shows a gene ranking correlation coefficient of 0.8310 (SD = 0.0912) for well-inferred genes and 0.8244 (SD = 0.0915) for all inferred genes, which are higher than those of non-GNN models using all 970 landmark transcripts (Tables 2 and 3). The KNN model consistently shows poor performance in gene ranking prediction. These results suggest that the graph structure effectively leverages gene–gene relationships compared to non-graph structures.

Next, we hypothesized that a strategic selection of landmark transcripts could further optimize the GNN model. We used LASSO regression [23] and greedy forward selection methods [24] to select 108 landmark transcript inputs as listed in Supplementary Table S5. Greedy forward selection is an iterative method for choosing a subset of features [24]. At each step, it incrementally adds the feature that produces the greatest reduction in overall validation error. Nine transcripts overlap between the random and LASSO selection methods: *RGS1*, *SERPINE1*, *TMEM2*, *WFS1*, *ICAM3*, *NPDC1*, *PSMB8*, *PTK2B*, and *HSPB1*. Five transcripts overlap between the random and greedy forward selection methods: *CSNK1A1*, *SERPINE1*, *SPDEF*, *TMEM2*, and *TSKU*. There are 42 overlapping transcripts between the LASSO and greedy forward selection methods. For all tested models, the greedy forward selection method resulted in higher inference performance compared to the random and LASSO selection methods (Table 4 and Supplementary Table S6). When the number of landmark transcript inputs was reduced to an extremely low number, specifically to 10 or even 1, the performance of the GNN model was more significantly affected by the choice of feature selection methods employed (Supplementary Table S7). Notably, the GNN model using 108 landmark transcripts selected using the greedy forward selection exhibited performance comparable to the GNN model using all 970 landmark transcripts (Table 4). These results suggest that the L1000 assay for these 108 landmark transcripts, followed by GNN-based inference, could generate the full transcriptome in a more cost-effective manner.

Assessment of cross-platform performance

Some gene expression inference studies have tested the cross-platform performance by using input data from different platforms. For example, the LINCS project trained an inference model using microarray data but generated gene expression profiles on L1000 data as input [3]. Similarly, the D-GEX model was trained on microarray data and tested on both microarray and RNA-seq data [11]. To assess the input cross-platform generality of our GNN model, we performed gene expression inference on RNA-seq input using the GNN model trained with the L1000 input. It shows better performance than non-GNN models in inferring expression values, with an overall error of 4.271 (SD = 0.891), compared to the KNN and baseline LR models, which show overall errors of 9.239 (SD = 1.764) and 9.298 (SD = 0.582), respectively. Additionally, an overall SCC and PCC for the GNN model are higher than those of non-GNN models (Supplementary Table S8). However, the GNN model does not outperform non-GNN models in predicting gene rankings (Supplementary Table S8). Interestingly, the KNN model exhibits worse cross-platform performance in gene ranking inference than the LR model. Since the GNN model with RNA-seq input does not perform well as it does with the training input (L1000 data), particularly in inferring differential expression between genes, platform-dependent differences in RNA-seq versus L1000 input seem to be unavoidable. It might require data preprocessing for effective cross-platform applications of the GNN model.

Discussion

Transcriptome information on gene expression levels and rankings is critical for estimating cellular states. The LINCS project developed full transcriptomes of millions of cellular contexts by predicting the full gene expression profiles from a subset of genes known as landmark genes. Expression values for 970 landmark transcripts were obtained using a cost-effective L1000 platform. Despite efforts to enhance inference performance with methods such as KNNs, neural networks, and generative models, further improvement is needed to develop a model capable of accurately inferring RNA-seq values from L1000 expression without data preprocessing. Additionally, it is unclear whether fewer than 970 landmark transcripts are sufficient to achieve the current prediction performance. In this study, we comprehensively assessed various models, including the baseline LR, MLP, KNN, SwinIR, and

Table 4. Performance in inferring gene expression profiles using LASSO regression or greedy forward selection of landmark transcripts

Model	Number of landmark transcripts	Selection method	Overall error	Overall SCC	Overall PCC	Gene ranking correlation coefficient (well-inferred genes)	Gene ranking correlation coefficient (all 12 320 genes)
GNN	108	LASSO	3.195 ± 0.719	0.9793 ± 0.0088	0.9805 ± 0.0097	0.8301 ± 0.0838 (11 995)	0.8248 ± 0.0840
		Greedy forward	**2.874 ± 0.574**	**0.9830 ± 0.0063**	**0.9844 ± 0.0068**	**0.8589 ± 0.0631** (11 990)	**0.8537 ± 0.0633**

Note: If the gene-level recall value is greater than the 95th percentile of a null distribution, the corresponding gene is considered to be a well-inferred gene. The GNN model using the greedy forward selection method reduces the prediction error by 97% compared to the constant model. **Bold** indicates the best performance. GNN, graph neural network

GNN. The GNN model exhibited the best performance in predicting both expression values and gene ranking, suggesting its ability to effectively leverage nonlinear gene correlations underlying gene expression profiles compared to other models. In contrast, SwinIR, one of the state-of-the-art deep learning models for image restoration tasks, performed even worse than the baseline LR model did. In the SwinIR model, genes were randomly assigned to the RNA-seq matrices. We expect that strategic assignment based on prior knowledge could potentially improve its performance.

Most GNN studies in the biological domain have focused on classification tasks, such as molecular interaction prediction and metastasis prediction in cancer [25, 26]. The inference performance of the GNN model in our work suggests its superior performance in capturing gene–gene relationships and its potential application to other regression tasks, such as predicting gene copy numbers, protein expression levels, or metabolite concentrations. Furthermore, ∼10% of the input information was sufficient for the GNN model to achieve the prediction performance of non-GNN models that use full input information. This further confirms better performance of the GNN model over others. The greedy forward selection method enhances the performance of the GNN when fewer landmark transcripts are used. The GNN model is an effective method for inferring gene expression profiles with fewer inputs, and offers higher accuracy in gene expression values and differential expression than non-GNN models. The generation of a full transcriptome using L1000 expression values from fewer landmark transcripts will enhance the cost-effectiveness compared to the current method. We believe the superior performance of the GNN model is attributed to its ability to process data features through a rich graph structure that encompasses interactions between non-landmark transcripts, which are not explicitly modeled in other models. To assess the impact of crosstalk among non-landmark transcripts, we conducted an ablation study by varying the number of edges between non-landmark transcripts in the graph. While increased non-landmark crosstalk does not always lead to better performance, more non-landmark crosstalk appears to be necessary when input data becomes limited (Supplementary Table S9).

While the GNN model demonstrates powerful performance in gene expression inference, its effectiveness largely depends on the graph structure. Designing an optimal graph requires extensive refinement and careful engineering, especially when explicit supervision for the graph structure is unavailable. Dense graph structures and deeper GNN layers often lead to increased computational costs and memory demands, although they can improve model expressiveness. Like other deep learning models, GNN faces interpretability challenges that must be addressed to improve its applicability in the biomedical field. Although there is a growing body of work proposing various explainability models for GNNs [27], understanding how predictions are made remains challenging due to the interactions between nodes and edges. Furthermore, it is not straightforward to translate edge connections and weights into biological implications.

Key Points

- Deep learning models with various architectures were evaluated for predicting gene expression profiles.
- The graph neural network (GNN) model outperforms non-GNN models in inferring both gene expression values and differential gene expression.
- The GNN model shows comparable inference performance to the linear regression model with ∼10 times less information.
- Our work demonstrates that transforming RNA expression data into a graph structure is effective for capturing nonlinear correlations between genes, providing a framework for applying GNNs to regression tasks in biological data.

Supplementary data

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

Conflict of interest: None declared.

Funding

This work was supported by a grant from the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) [RS-2023-00278378] to I.B.; a grant from Kyung Hee University (KHU-20233256) to I.B.; and an Institute of Information & Communication Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2024-00438638, EntireDB2AI: Foundations and Software for Comprehensive Deep Representation Learning and Prediction on Entire Relational Databases] to K.S.

Data availability

The paired L1000 and RNA-seq data used in this study are available in Gene Expression Omnibus (GEO) with the accession number GSE92743. The code for the GNN, MLP, and SwinIR

models used for gene expression inference and performance evaluation is available at https://github.com/HyunjinHwn/GeneExpressionGNN.

References

1. Hughes TR, Marton MJ, Jones AR. *et al.* Functional discovery via a compendium of expression profiles. *Cell* 2000;**102**:109–26. https://doi.org/10.1016/S0092-8674(00)00015-5
2. Lamb J, Crawford ED, Peck D. *et al.* The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 2006;**313**:1929–35. https://doi.org/10.1126/science.1132939
3. Subramanian A, Narayan R, Corsello SM. *et al.* A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* 2017;**171**:e1417. https://doi.org/10.1016/j.cell.2017.10.049
4. Li T, Tong W, Roberts R. *et al.* Deep learning on high-throughput transcriptomics to predict drug-induced liver injury. *Front Bioeng Biotechnol* 2020;**8**:562677. https://doi.org/10.3389/fbioe.2020.562677
5. Tripathi YM, Chatla SB, Chang YI. *et al.* A nonlinear correlation measure with applications to gene expression data. *PloS One* 2022;**17**:e0270270. https://doi.org/10.1371/journal.pone.0270270
6. Xiong H. Non-linear tests for identifying differentially expressed genes or genetic networks. *Bioinformatics* 2006;**22**:919–23. https://doi.org/10.1093/bioinformatics/btl034
7. Saint-Antoine MM, Singh A. Network inference in systems biology: recent developments, challenges, and applications. *Curr Opin Biotechnol* 2020;**63**:89–98. https://doi.org/10.1016/j.copbio.2019.12.002
8. Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. *Nat Rev Genet* 2019;**20**:631–56. https://doi.org/10.1038/s41576-019-0150-2
9. Blasco A, Endres MG, Sergeev RA. *et al.* Advancing computational biology and bioinformatics research through open innovation competitions. *PloS One* 2019;**14**:e0222165. https://doi.org/10.1371/journal.pone.0222165
10. Cheng Y, Xu SM, Santucci K. *et al.* Machine learning and related approaches in transcriptomics. *Biochem Biophys Res Commun* 2024;**724**:150225. https://doi.org/10.1016/j.bbrc.2024.150225
11. Chen Y, Li Y, Narayan R. *et al.* Gene expression inference with deep learning. *Bioinformatics* 2016;**32**:1832–9. https://doi.org/10.1093/bioinformatics/btw074
12. Jeon M, Xie Z, Evangelista JE. *et al.* Transforming L1000 profiles to RNA-seq-like profiles with deep learning. *BMC Bioinformatics* 2022;**23**:374. https://doi.org/10.1186/s12859-022-04895-5
13. Lu J, Chen M, Qin Y. Drug-induced cell viability prediction from LINCS-L1000 through WRFEN-XGBoost algorithm. *BMC Bioinformatics* 2021;**22**:13. https://doi.org/10.1186/s12859-020-03949-w
14. Magnusson R, Tegner JN, Gustafsson M. Deep neural network prediction of genome-wide transcriptome signatures - beyond the black-box. *NPJ Syst Biol Appl* 2022;**8**:9. https://doi.org/10.1038/s41540-022-00218-9
15. Zhou J, Cui GQ, Hu SD. *et al.* Graph neural networks: a review of methods and applications. *AI Open* 2020;**1**:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001
16. McDermott MBA, Wang J, Zhao WN. *et al.* Deep learning benchmarks on L1000 gene expression data. *IEEE/ACM Trans Comput Biol Bioinform* 2020;**17**:1846–57. https://doi.org/10.1109/TCBB.2019.2910061
17. Yan F, Jiang L, Chen D. *et al.* Reinventing gene expression connectivity through regulatory and spatial structural empowerment via principal node aggregation graph neural network. *Nucleic Acids Res* 2024;**52**:e60. https://doi.org/10.1093/nar/gkae514
18. Liang JY, Cao JZ, Sun GL. *et al.* SwinIR: image restoration using Swin transformer. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW 2021)* 2021;1833–44. 10.1109/ICCVW54120.2021.00210 https://doi.org/10.48550/arXiv.2108.10257
19. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* 2007;**8**:118–27. https://doi.org/10.1093/biostatistics/kxj037
20. Pedregosa F, Varoquaux G, Gramfort A. *et al.* Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;**12**:2825–30.
21. Taunk K, De S, Verma S. *et al.* A brief review of nearest neighbor algorithm for learning and classification. *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICCS)* 2019;**2019**:1255–60.
22. Diederik P, Kingma JB. Adam: a method for stochastic optimization. *3rd International Conference for Learning Representations* (ICLR). 2015. https://arxiv.org/pdf/1412.6980
23. Tibshirani R. Regression shrinkage and selection via the Lasso. *J R Stat Soc Ser B Stat Method* 1996;**58**:267–88.
24. Isabelle Guyon AeE. An introduction to variable and feature selection. *J Mach Learn Res* 2003;**3**:1157–82.
25. Chereda H, Bleckmann A, Menck K. *et al.* Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. *Genome Med* 2021;**13**:42. https://doi.org/10.1186/s13073-021-00845-7
26. Muzio G, O'Bray L, Borgwardt K. Biological network analysis with deep learning. *Brief Bioinform* 2021;**22**:1515–30. https://doi.org/10.1093/bib/bbaa257
27. Kakkad JJ, Jannu J, Sharma K, et al. *A Survey on Explainability of Graph Neural Networks*. IEEE Computer Society, 2023. 47.
28. Waskom ML. Seaborn: statistical data visualization. *J Open Source Softw* 2021;**6**:3021. https://doi.org/10.21105/joss.03021.

Briefings in Bioinformatics, 2025, **26(3)**, bbaf273
https://doi.org/10.1093/bib/bbaf273
Problem Solving Protocol